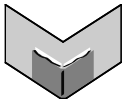
The Journal of Machine Learning Research
Volume 16
Print-Archive Edition

Pages 1305–2610



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Volume 16
Print-Archive Edition

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this volume are articles published electronically in JMLR in 2015.

JMLR is abstracted in ACM Computing Reviews, INSPEC, and Psychological Abstracts/PsycINFO.

JMLR is a publication of Journal of Machine Learning Research, Inc. For further information regarding JMLR, including open access to articles, visit http://www.jmlr.org/.

JMLR Print-Archive Edition is a publication of Microtome Publishing under agreement with Journal of Machine Learning Research, Inc. For further information regarding the Print-Archive Edition, including subscription and distribution information and background on open-access print archiving, visit Microtome Publishing at http://www.mtome.com/.

ISSN 1532-4435 (print)
ISSN 1533-7928 (online)

JMLR Editorial Board

Journal of Machine Learning Research

Volume 16, 2016

Journal of Machine Learning Research 16 (2015) 1305-1344 Submitted 12/13; Revised 12/14; Published 8/15

Learning Equilibria of Games via Payoff Queries

John Fearnley John.Fearnley@liverpool.ac.uk
Martin Gairing Gairing@liverpool.ac.uk
Ashton Building, Ashton Street, University of Liverpool, United Kingdom

Paul W. Goldberg Paul.Goldberg@cs.ox.ac.uk
Wolfson Building, Parks Road, University of Oxford, United Kingdom

Rahul Savani Rahul.Savani@liverpool.ac.uk
Ashton Building, Ashton Street, University of Liverpool, United Kingdom

Editor: Vahab Mirrokni

Abstract

A recent body of experimental literature has studied *empirical game-theoretical analysis*, in which we have partial knowledge of a game, consisting of observations of a subset of the pure-strategy profiles and their associated payoffs to players. The aim is to find an exact or approximate Nash equilibrium of the game, based on these observations. It is usually assumed that the strategy profiles may be chosen in an on-line manner by the algorithm. We study a corresponding computational learning model, and the query complexity of learning equilibria for various classes of games. We give basic results for exact equilibria of bimatrix and graphical games. We then study the query complexity of approximate equilibria in bimatrix games. Finally, we study the query complexity of exact equilibria in symmetric network congestion games. For directed acyclic networks, we can learn the cost functions (and hence compute an equilibrium) while querying just a small fraction of pure-strategy profiles. For the special case of parallel links, we have the stronger result that an equilibrium can be identified while only learning a small fraction of the cost values.

Keywords: query complexity, bimatrix game, congestion game, equilibrium computation, approximate Nash equilibrium

1. Introduction

Suppose that we have a game G with a known set of players, and known strategy sets for each player. We want to design an algorithm to solve G, where the algorithm can only obtain information about G via *payoff queries*. In a payoff query, the algorithm proposes pure strategies for the players, and is told the resulting payoffs. The general research issue is to identify bounds on the number of payoff queries needed to find an equilibrium, subject to the assumption that G belongs to some given class of games.

A general motivation for this topic is the observation that many data sets are generated by economic or competitive agents (for example, transactions on financial or housing markets, or data on competitive sports). In attempting to learn from such data sets, it seems natural to model the data-generating process in game-theoretic terms. To some extent, the work in *agent-based modelling* takes this approach: artificial selfish agents are simulated, and a general objective is to replicate various economic phenomena and behaviour observed in practice. We believe that there is considerable future potential to study data

sets through the game-theoretic lens in this way. This has already been successfully applied in the AI literature on adversarial security games, where for example, Yang et al. (2013) apply existing models of bounded rationality of an opponent, so as to improve competitive performance in an artificial online game. Nguyen et al. (2013) develop an adversary-based model (SUQR), and shows that SUQR's performance (using parameters learned from real-world data) improves over previous work that does not model adversaries; an extension of SUQR has been deployed in the context of fishery protection (Brown et al., 2014).

Suppose we have a detailed computational simulation of a game, and we want to check whether it gives rise to behaviour that corresponds with real-world observations. A key observation is that it is not too hard to take such a simulation, and feed into it some chosen behaviour of the (simulated) players, check their payoffs, and (with a bit more effort) check on whether players have a profitable deviation. From this, we get to the challenge of searching for an equilibrium of the game (ideally a Nash equilibrium; failing that, search for something weaker, like an approximate equilibrium). In terms of the theoretical model that is studied in this paper, our choice of which behaviour to simulate corresponds to the choices of payoff queries. Below, we discuss some of the literature in this setting.

1.1 Motivation for the Payoff-Query Model

Given a game, especially one with many players, it is unreasonable to assume that anyone maintains an explicit representation of its payoff function, even if the game in question has a concise representation. However, in practice, a reasonable modelling assumption is that given, say, a strategy profile for the players, we can determine their payoffs, or some estimate of the payoffs. We are interested in algorithms that find Nash equilibria using a sequence of queries, where a query proposes a strategy profile and gets told the payoffs. We would like to know under what conditions an algorithm can find a solution based on knowledge of some but not all of the game's payoffs, which is particularly important when there are many players, and the number of pure-strategy profiles is large. This kind of challenge (where you get observations of profile/payoff-vector pairs, and you want to find an approximate equilibrium, as opposed to the unobserved payoffs) has been the subject of experimental work (Vorobeychik et al., 2007; Wellman, 2006; Jordan et al., 2008; Duong et al., 2009), where Jordan et al. (2008) focuses on the case (highly relevant to this work) where the algorithm selects a sequence of pure profiles and gets told the resulting payoffs. *In this paper, we introduce the study of payoff-query algorithms from the algorithmic complexity viewpoint.*[1] We are interested in upper and lower bounds on the query complexity of classes of games.

From the theoretical perspective, we are studying a constrained class of algorithms for computing equilibria of games. The study of such constraints—especially when they lead to lower bounds or impossibility results—informs us about the approaches that a successful algorithm needs to apply. In the context of equilibrium computation, other kinds of constraint include *uncoupled* algorithms for computing equilibria (Hart and Mas-Colell, 2003, 2006), communication-constrained algorithms (Hart and Mansour, 2010; Daskalakis et al.,

1. The first discussion of this query model (that we are aware of) appears in a 2009 blog article by Noam Nisan: `https://agtb.wordpress.com/2009/11/19/the-computational-complexity-of-pure-nash/`. It also mentions *best-reply queries*, which deserve further attention in the context of adversarial security games.

2010; Goldberg and Pastink, 2012), and *oblivious* algorithms (Daskalakis and Papadimitriou, 2009). Of course, the restriction to polynomial-time algorithms is the best-known example of such a constraint. Based on the algorithms and open problems identified in this paper, we find this to be a compelling motivation for the further study of the payoff-query model. There are various related kinds of query models that are suggested by the payoff queries studied here, which may also be of similar theoretical interest; we discuss these in Section 6.

1.2 Games and Query Models

In this paper we introduce the study of payoff-queries for strategic-form games. We also consider two models of concisely represented games: *graphical games* (Kearns et al., 2001), where players are nodes in a given graph and the payoff of a player only depends on the strategies of its neighbors in the graph, and *symmetric network congestion games* (Fabrikant et al., 2004), where the strategy space of the players corresponds to the set of paths that connect two nodes in a network.

For a strategic-form game, we assume that initially the querying algorithm only knows n, the number of players, and k, the number of pure strategies that each player has.

Definition 1 *A* payoff query *to a strategic-form game G selects a pure-strategy profile* `s` *for G, and is given as response, the payoffs that G's players derive from* `s`.

There are k^n pure-strategy profiles in a game, and one could learn the game exhaustively using this many payoff queries. We are interested in algorithms that require only a small fraction of this trivial upper bound on the number of queries required.

For our results on symmetric network congestion games, we assume that initially the algorithm only knows the number of players n, and the set of pure strategies, given by a graph and the common origin/destination pair. In this paper, we will consider two different query models, which are described in the following definition.

Definition 2 *For a symmetric congestion game with m pure strategies and n players, a* query *is a tuple $q = (q_1, q_2, \ldots, q_m)$, where for each pure strategy $i = 1, 2, \ldots, m$, we have that $q_i \in \{0, 1, 2, \ldots, n\}$ is the number of players assigned to i under the query. In response to the query q, the querier learns the costs of each pure strategy under the assigned loads. Let $Q = \sum_{1 \leq i \leq m} q_i$. We consider two different types of queries:*

- *In a* normal-query*, we require that $Q = n$;*
- *in an* under-query*, we require that $Q < n$.*

Normal-queries correspond to the query model that we use for strategic-form games. For a congestion game, m, which is the number of paths from the origin to the destination in a graph, may be exponential. While we defined a query for congestion as a tuple of length m, both normal-queries and under-queries require at most n positions of this tuple to be non-zero, so the query can be specified succinctly. We use under-queries in our query algorithm for games played on directed acyclic graphs. We feel that under-queries are a reasonable query model for congestion games, because we can ask some players to refrain from playing when we conduct our query.

Definition 3 *The* payoff query complexity *of a class of games $\mathcal{G}$, with respect to some solution concept such as exact or approximate Nash equilibrium, is defined as follows. It is the smallest N such that there is some algorithm $\mathcal{A}$ that, given N payoff queries to any game $G \in \mathcal{G}$ (where initially none of the payoffs of G are known) can find a solution of G.*

The definition imposes no computational bound on the algorithm $\mathcal{A}$. It is to some extent inspired by the work on query-based learning initiated by Angluin (1987, 1988), in the context of computational learning theory. Note that $\mathcal{A}$ may select the queries in an on-line manner, so queries can depend on the responses to previous queries.

1.3 Overview of Results

We study a variety of different settings. In Section 3, we consider bimatrix games. Our first result is a lower bound for computing an exact Nash equilibrium: in Theorem 4, we show that computing an exact Nash equilibrium in a $k \times k$ bimatrix game has payoff query complexity k^2, even for zero-sum games. In other words, we have to query every pure strategy profile.

We then turn our attention to approximate Nash equilibria, where we obtain some more positive results. With the standard assumption that all payoffs lie in the range $[0, 1]$, we show that, when $2 \leq i \leq k-1$, the payoff query complexity of computing a $(1 - \frac{1}{i})$-approximate Nash equilibrium is at most $2k - i + 1$ (Theorem 5) and at least $k - i + 1$ (Theorem 7.) We also observe that, when $\epsilon \geq 1 - \frac{1}{k}$, no payoff queries are needed at all, because an ϵ-Nash equilibrium is achieved when both players mix uniformly over their pure strategies.

The query complexity of computing an approximate Nash equilibrium when $\epsilon < \frac{1}{2}$ appears to be a challenging problem, and we provide an initial lower bound in this direction in Theorem 13: we show that the payoff query complexity of finding a ϵ-approximate Nash equilibrium for $\epsilon = \mathcal{O}(\frac{1}{\log k})$ is $\Omega(k \cdot \log k)$. This gives an interesting contrast with the $\epsilon \geq \frac{1}{2}$ case. Whereas we can always compute a $\frac{1}{2}$-approximate with $2k - 1$ payoff queries, there exists a constant $\epsilon < \frac{1}{2}$ for which this is not the case, as shown in Corollary 14.

Having studied payoff query complexity in bimatrix games, it is then natural to look for improved payoff query complexity results in the context of "structured" games. In particular, we are interested in *concisely represented* games, where the payoff query complexity may be much smaller than the number of pure strategy profiles. As an initial result in this direction, in Section 4 we consider graphical games, where we show (Theorem 15) that for graphical games with constant degree d, a Nash equilibrium can be found with a polynomial number of payoff-queries. This algorithm works by discovering every payoff in the game, however unlike bimatrix games, this can be done without querying every pure strategy profile.

Finally, we focus on two different models of congestion games. In Section 5.1, we consider the case of *parallel links*, where the game has a origin and destination vertex, and m parallel links between them. We show both lower and upper bounds for this setting. If n denotes the number of players, then we obtain a $\log(n) + m$ payoff query lower bound (Theorem 17), which applies to both query models. We obtain an upper bound of $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)} + m\right)$ normal-queries (Theorem 26). Note that there are $n \cdot m$ different

payoffs in a parallel links game, and so our upper bound implies that you do not need to discover the entire payoff function in order to solve a parallel links game.

In Sections 5.2, 5.3, 5.4, we consider the more general case of symmetric network congestion games on directed acyclic graphs. We show that if the game has m edges and n players, then we can find a Nash equilibrium using $m \cdot n$ payoff queries (Theorem 38). The algorithm discovers every payoff in the game, but it only queries a small fraction of the pure strategy profiles.

2. Related Work

In Section 2.1 we review some very recent work on the payoff query complexity of related game-theoretic solution concepts. In Section 2.2 we review the experimental work that motivated this paper. Finally, in Section 2.3 we discuss the relationship with work that analyzes *best-response dynamics* in a game-theoretic context.

2.1 Payoff Query Complexity

A preliminary version of this paper appeared at the ACM conference on Electronic Commerce (Fearnley et al., 2013). Work that has appeared subsequently has studied query complexity bounds for general multi-player games, where the main parameter of interest is the number of players n, who usually just have a small number of pure strategies. Hart and Nisan (2013) obtain an exponential in n lower bound on the query complexity of finding an exact correlated equilibrium of a general n-player game. Note that any lower bounds for correlated equilibria apply immediately to Nash equilibria, since Nash equilibria are a more restrictive solution concept. For *approximate* correlated equilibria, no-regret learning dynamics can be simulated by a randomized payoff query algorithm, so that the query complexity of approximate correlated equilibria is polynomial in the number of players (Babichenko and Barman, 2013; Hart and Nisan, 2013). Goldberg and Roth (2014) studied the dependence in more detail, obtaining upper and lower bounds that are logarithmic in n. However, randomness is needed: Babichenko and Barman (2013) show that finding an exact correlated equilibrium in an n-player games using a deterministic querying strategy requires exponentially many queries in n. This result is strengthened by Hart and Nisan (2013), where it is shown that deterministic querying strategies require exponentially many queries to find even a $\frac{1}{2}$-approximate correlated equilibrium.

Approximate *well-supported* Nash equilibria are another approximate solution concept that have been studied in the context of strategic form games (Kontogiannis and Spirakis, 2010; Fearnley et al., 2012). Babichenko (2014) has shown that finding a 10^{-8}-well supported Nash equilibrium in an n-player game requires exponentially many queries in n. The query complexity of computing an ϵ-approximate Nash equilibrium (that need not be well-supported) for constant ϵ remains open, although Goldberg and Roth (2014) show that it is polynomial if the unknown game can be specified concisely. These negative results for n-player games motivate the consideration of more structured classes of games, such as congestion games, which we study in this paper.

Finally, Fearnley and Savani (2014) have continued the study of query complexity for bimatrix games that was initiated in this paper. In particular, they show that randomized payoff query algorithms can achieve better approximation ratios: there is a randomized

algorithm for finding a $(\frac{3-\sqrt{5}}{2}+\epsilon)$-Nash equilibrium in a bimatrix game using $O(\frac{k \cdot \log k}{\epsilon^2})$ payoff queries, and there is a randomized algorithm for finding a $(\frac{2}{3}+\epsilon)$-WSNE in a bimatrix game using $O(\frac{k \cdot \log k}{\epsilon^4})$ payoff queries. They also provide lower bounds for finding well-supported Nash equilibria in bimatrix games: finding an ϵ-well-supported Nash equilibrium requires $k-1$ payoff queries for any $\epsilon < 1$, even in win-lose games, and finding a $\frac{1}{3k}$-well-supported Nash equilibrium requires $\Omega(k^2)$ payoff queries, even in win-lose constant-sum games.

2.2 Experimental Work

In *empirical game-theoretic analysis* (Wellman, 2006; Jordan et al., 2010), a game is presented to the analyst via a set of observations of strategy profiles (usually, pure) and their corresponding payoffs. This set of profiles/payoff-vector pairs is called an *empirical game*. In some settings the strategy profiles are randomly generated, but it is typically feasible to obtain observations via the payoff queries we study here. The *profile selection problem* (Jordan et al., 2008) is the challenge of choosing helpful strategy profiles. The *strategy exploration problem* (Jordan et al., 2010) is the special case of finding the best way to limit the search to a small subset of a large set of strategies.

Jordan et al. (2008) envisage a setting where a game (called a *base game*) has a corresponding *game simulator*, an implementation in software, which is amenable to payoff queries; a more general scenario allows the observed payoffs to be sampled from a distribution associated with the strategy profile. The distribution is sometimes considered to be due to a noise process, and called the *noisy payoff model* in Jordan et al. (2008). In this paper we just consider deterministic payoffs, the "revealed payoff model" in Jordan et al. (2008). As noted in Vorobeychik et al. (2007), a profile can be repeatedly queried to sample from the distribution of payoffs, and thus get an estimate of the expected values. The two interacting challenges are to identify helpful queries, and to use them to find pure-strategy profiles that have low regret (where *regret* refers to the largest incentive to deviate, amongst the players.)

Vorobeychik et al. (2007) study the *payoff function approximation task*, in which a game belongs to a known class, and there is a "*regression*" challenge to determine certain parameters; the information about the game consists of a random sample of pure profiles and resulting payoff vectors. However, success is measured by the extent that the players' predicted behaviour is close to the behaviour associated with the true payoffs, rather than how well the true payoff functions are estimated.

Work on specific classes of multi-player games includes the following. Duong et al. (2009) studies algorithms for learning graphical games; we consider a graphical game learning algorithm in Section 4. Jordan et al. (2008) apply payoff-query learning to various kinds of games generated by GAMUT (Nudelman et al., 2004), including a class of congestion games. Vorobeychik et al. (2007) investigate a first-price auction and also a scheduling game, where payoffs are described via a finite random sample of profile/payoff vector pairs. Earlier, Sureka and Wurman (2005) study search for pure Nash equilibria of strategic-form games (mostly with 5 players and 10 pure strategies).

Most of the experimental work (e.g., Sureka and Wurman 2005; Jordan et al. 2008; Duong et al. 2009) uses *local search*, in which profiles that get queried are typically very similar (differing in just one player's strategy) from previously queried profiles. Jordan

et al. (2008) experiment with local-search type algorithms in which when a player has the incentive to deviate, the tested profile is updated with that deviation. Sureka and Wurman (2005) study search for pure equilibria via best-response dynamics while maintaining a tabu list, introduced to reduce the risk of cycles.

2.3 Best-Response Dynamics and Local Search

There is a large body of literature that studies best- and better-response dynamics for classes of potential games, and gives bounds on the number of steps required for convergence to pure-strategy equilibria. These dynamics relate to the payoff query model since they work by exploring the space of pure profiles, and receiving feedback consisting of payoffs. The difference is that they purport to model a decentralized process of selfish behaviour by the players, while the payoff query model envisages a centralized algorithm that is less constrained. In this section, we discuss some of the relevant literature.

Local search processes in that each pure profile is obtained from the previous one by letting a single player move have been studied extensively in the literature. Bounds on the convergence of deterministic best-response dynamics were considered in Even-Dar et al. (2003) and Feldmann et al. (2003). Gairing and Savani (2010, 2011) showed polynomial convergence of better-response dynamics for certain *hedonic games*. The better-response dynamics considered by Goldberg (2004) is the basic randomized local search algorithm, and bounds are obtained for its convergence to exact equilibrium. The work in Bei et al. (2013) shows that a Nash equilibrium of a bimatrix game can be found using a polynomial number of better-response queries. Chien and Sinclair (2011) study another local search, the *ϵ-Nash dynamics*, and its convergence to approximate equilibria. Gairing et al. (2010) employ controlled local search dynamics (where a sequence of players moves simultaneously) to compute pure Nash equilibria. Other papers (e.g., Fischer et al. 2006; Berenbrink et al. 2007) analyze strongly-distributed dynamics in which multiple players can move in the same time step; consequently the dynamics is not a local search. However, these dynamical systems could all be simulated by payoff query algorithms in which at each step, at most nk queries are made to determine the change in payoffs available to players as a result of unilateral deviations. This paper begins to answer the question: how much better could a payoff query algorithm do, if it were not subject to that constraint?

Finally, Alon et al. (2011) consider payoff-query algorithms for finding the costs of paths in graphs. They consider *weight discovery protocols* where the aim is to determine the costs of edges, and *shortest path discovery protocols* where the aim is to find a shortest path. The latter objective is more similar to what we consider, since it can avoid the need to learn the entire payoff function; also a shortest path is an equilibrium strategy for the one-player case of a network congestion game.

3. Bimatrix Games

In this section, we give bounds on the payoff-query complexity of computing approximate Nash equilibria of bimatrix games. A *bimatrix game* is a pair (R, C) of two $k \times k$ matrices: R gives payoffs for the *row player*, and C gives payoffs for the *column player*. We use $[n]$ to denote the set $\{1, 2, \ldots, n\}$. A *mixed strategy* is a probability distribution over $[k]$. A

mixed strategy profile is a pair $\mathsf{s} = (\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is a mixed strategy for the row player, and $\mathbf{y}$ is a mixed strategy for the column player.

Let $\mathsf{s} = (\mathbf{x}, \mathbf{y})$ be a mixed strategy profile in a $k \times k$ bimatrix game (R, C). We say that a row $i \in [k]$ is a *best response* for the row player if $R_i \cdot \mathbf{y} = \max_{j \in [k]} R_j \cdot \mathbf{y}$. We say that a column $i \in [k]$ is a best response for the column player if $(\mathbf{x} \cdot C)_i = \max_{j \in [k]} (\mathbf{x} \cdot C)_j$. We define the row player's *regret* under $\mathsf{s} = (\mathbf{x}, \mathbf{y})$ as the difference between the payoff of a best response and the payoff that the row player obtains under s. More formally, the regret that the row player suffers under s is:

$$\max_{j \in [k]} (R_j \cdot \mathbf{y}) - \mathbf{x} \cdot R \cdot \mathbf{y}.$$

Similarly, the column player's regret is defined to be:

$$\max_{j \in [k]} ((\mathbf{x} \cdot C)_j) - \mathbf{x} \cdot C \cdot \mathbf{y}.$$

We say that s is a *mixed Nash equilibrium* if both players have regret 0 under s. An *ϵ-Nash equilibrium* is an approximate solution concept: for every $\epsilon \in [0, 1]$, we say that s is an ϵ-Nash equilibrium if both players suffer regret at most ϵ under s.

We begin with the following simple observation: there are no query-efficient algorithms for finding *exact* Nash equilibria, even in zero-sum games. The following theorem shows that, in order to find an exact Nash equilibrium, we must query all $k \times k$ pure strategy profiles.

Theorem 4 *The payoff query complexity of finding an exact Nash equilibrium of a zero-sum $k \times k$ bimatrix game is k^2.*

Proof Consider a generalized version of matching pennies, where the column player pays 1 to the row player whenever both players choose the same strategy, otherwise the row player pays 1 to the column player. Note that this is a zero-sum game, and that it has a unique Nash equilibrium, namely when both players randomize uniformly over their strategies. Now suppose each payoff in the game is perturbed by a small quantity, in such a way as to maintain the zero-sum property. For small perturbations, there will still be a unique fully-mixed equilibrium profile, but it can only be known exactly if all the payoffs are known exactly. Thus, we cannot find an exact Nash equilibrium in a zero-sum bimatrix game without querying all $k \times k$ pure strategy profiles. ■

Theorem 4 implies that we cannot devise query-efficient algorithms for finding exact Nash equilibria. This naturally raises the question of whether there are query-efficient algorithms for finding *approximate* Nash equilibria, and we continue by presenting results on this topic. From now on, we will assume that all payoffs lie in the range $[0, 1]$, which is a standard assumption when finding approximate Nash equilibria.

Our first result is an upper bound. The work of Daskalakis, Mehta, and Papadimitriou (Daskalakis et al., 2009b) gives a simple algorithm for finding a $\frac{1}{2}$-Nash equilibrium. We adapt their algorithm to prove the following result.

Theorem 5 *Let i be chosen such that $2 \leq i \leq k-1$. The payoff query complexity of finding a $(1-\frac{1}{i})$-approximate equilibrium of a $k \times k$ bimatrix game is at most $2k-i+1$.*

Proof We begin by querying all k pure profiles where the row player plays row 1. This allows us to find the column player's best response to row 1. Without loss of generality, we can assume that this is column 1. Now query column 1 against rows 2 through $k-i+2$. Note that we have made a total of $2k-i+1$ queries. Let row b be a row that maximizes the row player's payoff against column 1, among those that we have queried. Let $B = \{1, b\} \cup [k-i+3, k]$. We propose the following mixed strategy profile s: the column player plays column 1 with probability 1, and the row player mixes uniformly over the strategies in B. Note that the row player is mixing between i rows, and thus plays each of them with probability $\frac{1}{i}$.

We claim that s is a $(1-\frac{1}{i})$-approximate Nash equilibrium. Let R and C be the actual payoff matrices for the row and column player, respectively. Note that the row player's best response to column 1 is either b, or one of the strategies between $k-i+3$ and k. Call this row j, and observe that $j \in B$. The row player's regret can be expressed as

$$\begin{aligned} R_{j,1} - \sum_{\ell \in B} \frac{1}{i} \cdot R_{\ell,1} &= (1-\frac{1}{i}) \cdot R_{j,1} - \sum_{\ell \in B \setminus \{j\}} \frac{1}{i} \cdot R_{\ell,1} \\ &\leq (1-\frac{1}{i}) \cdot R_{j,1} \\ &\leq (1-\frac{1}{i}). \end{aligned}$$

Let j' be a pure best response of the column player under s. Observe that, since column 1 is a best response against row 1, we have that $C_{1,j'} - C_{1,1} \leq 0$. The column player's regret can be expressed as:

$$\begin{aligned} \sum_{\ell \in B} \frac{1}{i} \cdot C_{\ell,j'} - \sum_{\ell \in B} \frac{1}{i} \cdot C_{\ell,1} &= \sum_{\ell \in B} \frac{1}{i} \cdot (C_{\ell,j'} - C_{\ell,1}) \\ &\leq \sum_{\ell \in B \setminus \{1\}} \frac{1}{i} \cdot (C_{\ell,j'} - C_{\ell,1}) \\ &\leq \sum_{\ell \in B \setminus \{1\}} \frac{1}{i} \\ &= 1 - \frac{1}{i}. \end{aligned}$$

Thus we have shown that both players suffer regret at most $1-\frac{1}{i}$. ■

Note that, when $i = 2$, the algorithm of Theorem 5 finds a $\frac{1}{2}$-Nash equilibrium using the same technique as the algorithm from Daskalakis et al. (2009b). For $i > 2$, our algorithm uses fewer payoff queries in exchange for a worse approximation. When $i = k-1$, our algorithm uses $k+2$ payoff queries in order to find a $(1-\frac{1}{k-1})$-Nash equilibrium. It turns out that, for $\epsilon \geq 1-\frac{1}{k}$, we do not need to make any payoff queries at all: an ϵ-Nash equilibrium

is obtained when both players play the uniform distribution over their strategies, because both players must place at least $\frac{1}{k}$ of their probability on a pure best response.

We now turn our attention to lower bounds. We complement the result of Theorem 5 by showing lower bounds for finding $(1-\frac{1}{i})$-Nash equilibria, when i is in the range $2 \leq i \leq k-1$. First, we prove an auxiliary lemma.

Lemma 6 *Suppose that every payoff query that is made by the algorithm returns 0 for both players. Let i be chosen such that $2 \leq i \leq k-1$, and let s be a $(1-\frac{1}{i})$-Nash equilibrium. Any column that receives no queries must be assigned at least $\frac{1}{i}$ probability by s.*

Proof Suppose, for the sake of contradiction, that c is a column that received no queries, and that c is assigned strictly less than $\frac{1}{i}$ probability by s. We construct a column player matrix C as follows:

$$C_{j,j'} = \begin{cases} 1 & \text{if } j' = c, \\ 0 & \text{otherwise.} \end{cases}$$

Since c received no queries, C is consistent with all queries that have been made. Note that the column player's payoff under s is strictly less than $\frac{1}{i}$, and that the payoff of playing c as a pure strategy is 1. Thus, the column player's regret is strictly greater than $1-\frac{1}{i}$, which contradicts the fact that s is a $(1-\frac{1}{i})$-Nash equilibrium ■

Now we can show our lower bound.

Theorem 7 *Let i be chosen such that $2 \leq i \leq k-1$. The payoff query complexity of finding a $(1-\frac{1}{i})$-approximate Nash equilibrium of a $k \times k$ bimatrix game is at least $k-i+1$.*

Proof Assume that all payoff queries return 0 for both players. Suppose, for the sake of contradiction, that an algorithm makes fewer than $k-i+1$ payoff queries, and then outputs s as a $(1-\frac{1}{i})$-Nash equilibrium. It follows that there must be at least i columns that have received no payoff queries at all, and without loss of generality, we can assume that these are columns 1 through i. By Lemma 6, we know that s must assign exactly $\frac{1}{i}$ probability to each of the columns 1 through i. Since there are k rows, there is at least one row r that receives probability at most $\frac{1}{k}$ under s. We construct a row player payoff matrix R as follows:

$$R_{j,j'} = \begin{cases} 1 & \text{if } j = r \text{ and } 1 \leq j' \leq i, \\ 0 & \text{otherwise.} \end{cases}$$

Since columns 1 through i were not queried, R is consistent with all queries that have been made so far. The row player's payoff under s is at most $\frac{1}{k}$. On the other hand, the row player would receive payoff 1 for playing r as a pure strategy. Thus, the row player's regret is at least:

$$1 - \frac{1}{k} > 1 - \frac{1}{i}.$$

This contradicts the fact that s is a $(1-\frac{1}{i})$-Nash equilibrium. ■

As a consequence of the previous two theorems, when $2 \leq i \leq k-1$, we have that the payoff query complexity of finding a $(1-\frac{1}{i})$-Nash equilibrium lies somewhere in the range $[k-i+1, 2k-i+1]$. Determining the precise payoff query complexity for this case is an open problem.

So far, we have only considered ϵ-Nash equilibria with $\epsilon \geq \frac{1}{2}$. Of course, the most interesting challenge is to determine the payoff query complexity for values of $\epsilon < \frac{1}{2}$. By our previous results, we know that the payoff query complexity for finding a $\frac{1}{2}$-Nash equilibrium is $\mathcal{O}(k)$, and the payoff query complexity for finding a 0-Nash equilibrium is $\mathcal{O}(k^2)$, but we do not know how the payoff query complexity behaves as we vary ϵ between 0 and $\frac{1}{2}$.

Our final result in this section will be to show a lower bound for $\epsilon = \mathcal{O}(\frac{1}{\log k})$. We will show that finding a $\mathcal{O}(\frac{1}{\log k})$-Nash equilibrium requires $\Omega(k \log k)$ payoff queries. This establishes that there are some positive values of ϵ, for which computing an ϵ-Nash equilibrium is asymptotically harder than computing a $\frac{1}{2}$-Nash equilibrium.

We will use the following class of bimatrix games, which have been previously used in Theorem 1 of Feder et al. (2007).

Definition 8 *Let $\mathcal{G}_\ell$ be the class of strategic-form games where the column player has ℓ pure strategies and the row player has $\binom{\ell}{\ell/2}$ pure strategies (where we assume ℓ is even). Let $G_\ell \in \mathcal{G}_\ell$ be the win-lose constant-sum game in which each row of the row player's payoff matrix has $\frac{\ell}{2}$ 1's and $\frac{\ell}{2}$ 0's, all rows being distinct. The column player's payoffs are one minus the row player's payoffs.*

It is well-known that every zero-sum game has a unique *value*, which is the payoff that both players can guarantee for themselves, independent of what the other player does. The value of each game $G_\ell \in \mathcal{G}_\ell$ is $\frac{1}{2}$ since either player can obtain payoff $\frac{1}{2}$ by using the uniform distribution over their pure strategies. Our first lemma shows that, if the column player deviates from this by placing too much probability on a single column, then the row player can take advantage and increase his payoff.

Lemma 9 *Suppose that in game $G_\ell \in \mathcal{G}_\ell$, the column player places probability $\alpha > 1/\ell$ on some column. Then the row player can obtain a payoff strictly greater than $\frac{1}{2} + \frac{\alpha}{2} - \frac{1}{2\ell}$.*

Proof Let j be a column that the column player plays with probability α. Let R_j be the set of rows where the row player obtains payoff 1 against column j. Suppose the row player plays the uniform distribution over rows in R_j. When the column player plays j, the row player receives payoff 1. Let $j' \neq j$ be a column, and consider the payoffs to the row player where j' intersects R_j. A fraction $\frac{\ell/2-1}{\ell-1}$ of these entries pay the row player 1, while a fraction $\frac{\ell/2}{\ell-1}$ pay the row player 0. Consequently whenever the column player plays $j' \neq j$, the row player's expected payoff is $\frac{\ell/2-1}{\ell-1}$. Thus with probability α the row player receives payoff 1, and with probability $1-\alpha$ he receives payoff $\frac{\ell/2-1}{\ell-1}$. Thus, the payoff to the row

player is

$$\begin{aligned}\alpha + (1-\alpha)\frac{\ell/2-1}{\ell-1} &= \frac{1}{2} + \frac{1}{2}\alpha - \frac{1-\alpha}{2(\ell-1)} \\ &> \frac{1}{2} + \frac{1}{2}\alpha - \frac{1-1/\ell}{2(\ell-1)} \\ &= \frac{1}{2} + \frac{1}{2}\alpha - \frac{1}{2\ell}\ ,\end{aligned}$$

which completes the proof. ■

We now use the bound from the previous lemma to show that, in an approximate Nash equilibrium for G_ℓ, the column player cannot place too much probability on any individual column.

Corollary 10 *Let $\alpha > \frac{1}{k}$, and let $\epsilon = \frac{1}{4}(\alpha - \frac{1}{\ell})$. In every ϵ-Nash equilibrium of $G_\ell \in \mathcal{G}_\ell$, the column player plays each individual column with probability at most α.*

Proof Suppose, for the sake of contradiction, that there is an ϵ-Nash equilibrium s in which that the column player assigns column j probability strictly greater than α. Then, by Lemma 9, the row player's payoff is strictly greater than $\frac{1}{2} + \frac{\alpha}{2} - \frac{1}{2\ell}$, and therefore the row player's payoff in s must be strictly greater than:

$$\frac{1}{2} + \frac{\alpha}{2} - \frac{1}{2\ell} - \epsilon = \frac{1}{2} + \epsilon.$$

Therefore, the column player obtains payoff strictly less than $\frac{1}{2} - \epsilon$. Since the value of G_ℓ is $\frac{1}{2}$, the column player's regret in s is strictly greater than ϵ, and therefore s is not an ϵ-Nash equilibrium. ■

We can now provide a lower bound for the payoff query complexity of finding an approximate Nash equilibrium for the games in $\mathcal{G}_\ell$.

Lemma 11 *For any $\epsilon < \frac{1}{12}$, and any even $\ell \geq 8$, the payoff query complexity of finding an ϵ-Nash equilibrium for the games in $\mathcal{G}_\ell$ is at least $\frac{1}{2} \cdot \binom{\ell}{\ell/2} \cdot (\frac{1}{16\epsilon+4/\ell})$.*

Proof Let $\mathcal{A}$ be a payoff query algorithm for finding an ϵ-Nash equilibrium, and, for the sake of contradiction, suppose that $\mathcal{A}$ makes fewer than $\frac{1}{2} \cdot \binom{\ell}{\ell/2} \cdot (\frac{1}{16\epsilon+4/\ell})$ many payoff queries when processing G_ℓ. Let s be the mixed strategy profile that $\mathcal{A}$ outputs for G_ℓ. By Corollary 10, we know that no column in s is assigned more than $\alpha = 4\epsilon + \frac{1}{\ell}$ probability. We also know that in s, the row player's payoff is at most $\frac{1}{2} + \epsilon$, since s is an ϵ-Nash equilibrium of a constant-sum game with value $\frac{1}{2}$. Since $\mathcal{A}$ made fewer than $\frac{1}{2} \cdot \binom{\ell}{\ell/2} \cdot (\frac{1}{16\epsilon+4/\ell})$ payoff queries, at least half of the rows received fewer than $(\frac{1}{16\epsilon+4/\ell})$ queries. Since $\ell \geq 8$, this implies that there are at least $\frac{1}{2} \cdot \binom{8}{4} = 45$ such rows. Thus, there is one such row, call it r, that is played with probability strictly less than $\frac{1}{12}$ in s.

Since $\mathtt{s}$ assigns at most α probability to each column, the total amount of probability that $\mathtt{s}$ assigns to the queried portion of r is at most $\alpha(\frac{1}{16\epsilon+4/\ell}) = \frac{1}{4}$. Now suppose that we modify G_ℓ by replacing all un-queried entries of r with payoffs of 1 for the row player. Call this new game G'_ℓ. Note that $\mathcal{A}$ outputs the same strategy profile $\mathtt{s}$ for both G_ℓ and G'_ℓ.

Let p be the payoff to the row player of playing $\mathtt{s}$ in G_ℓ, and let p' be the payoff to the row player of playing $\mathtt{s}$ in G'_ℓ. Since r is played with probability less than $\frac{1}{12}$ we have:

$$\begin{aligned} p' &\leq p + \frac{1}{12} \\ &\leq \frac{7}{12} + \epsilon \end{aligned}$$

However, the row player's best response payoff is at least $\frac{3}{4}$ in G'_ℓ, so we have:

$$p' \geq \frac{3}{4} - \epsilon$$

Therefore, we can conclude that:

$$\begin{aligned} \frac{7}{12} + \epsilon &\geq \frac{3}{4} - \epsilon \\ 2\epsilon &\geq \frac{2}{12}. \end{aligned}$$

However, this is impossible because $\epsilon < \frac{1}{12}$. ■

Finally, we can extend the lower bound to square bimatrix games.

Lemma 12 *For $k \times k$ bimatrix games, the payoff query complexity of finding an ϵ-Nash equilibrium, for $\epsilon \leq \frac{1}{8}$, is at least $k \cdot (\frac{1}{32/\log k + 64\epsilon})$.*

Proof Let k' be the largest number of the form $\binom{\ell}{\ell/2}$ that is smaller than k. We have $k' \geq k/4$ and $\ell \geq \log k/2$. By Lemma 11, the number of payoff queries needed to find an ϵ-Nash equilibrium for games in $\mathcal{G}_k$ is at least:

$$\begin{aligned} \binom{\ell}{\ell/2} \cdot \Big(\frac{1}{16\epsilon + 4/\ell}\Big) &= k'\Big(\frac{1}{4/\ell + 16\epsilon}\Big) \\ &\geq \frac{k}{4}\Big(\frac{1}{4/\ell + 16\epsilon}\Big) \\ &\geq \frac{k}{4}\Big(\frac{1}{8/\log(k) + 16\epsilon}\Big) \\ &= k\Big(\frac{1}{32/\log(k) + 64\epsilon}\Big). \end{aligned}$$

The games in $\mathcal{G}_\ell$ can be written down as a $k \times k$ game, by duplicating rows and columns. Note that these operations preserve approximate equilibria. ■

By taking $\epsilon \in \mathcal{O}(\frac{1}{\log k})$ in the previous lemma, we arrive at our final theorem.

Theorem 13 *For $k \times k$ bimatrix games, the payoff query complexity of finding a ϵ-Nash equilibrium for $\epsilon \in \mathcal{O}(\frac{1}{\log k})$, is $\Omega(k \cdot \log k)$.*

Recall, from Theorem 5, that we can always find a $\frac{1}{2}$-Nash equilibrium using $2k-1$ payoff queries. The following corollary of Lemma 12 shows that there are some constant values of ϵ that require more payoff queries.

Corollary 14 *There is a constant value of $\epsilon > 0$ for which finding an ϵ-Nash equilibrium of a $k \times k$ bimatrix game requires strictly more than $2k-1$ payoff queries.*

Proof Consider, for example, setting $\epsilon = \frac{1}{512}$ in Lemma 12. Then, for the family of games in $\mathcal{G}_l$ with $l > 2^{256}$, we have a lower bound of

$$k \cdot \left(\frac{1}{\frac{32}{\log k} + 0.0064} \right) > k \cdot \frac{1}{0.125 + 0.125} = 4 \cdot k,$$

on the number of payoff queries. ■

An interesting question that remains is whether one can a show a superlinear lower bound on the number of payoff queries required for a constant ϵ.

4. Graphical Games

In this section, we give a simple payoff query-based algorithm for graphical games. In a n-player *graphical game* (Kearns et al., 2001) the players lie at the vertices of a degree-d graph, and a player's payoff is a function of the strategies of just himself and his neighbors. If every player has k pure strategies, then the number of payoff values needed to specify such a game is $n \cdot k^{d+1}$ which, in contrast with strategic-form games, is polynomial (assuming d is a constant).

Previously, Duong et al. (2009) have carried out experimental work on payoff queries for graphical games. They compare a number of techniques; the algorithm we give here is polynomial-time but would likely be less efficient in practice. Similar to Duong et al. (2009), we assume the underlying graph G is unknown, and we want to induce the structure of G, and corresponding payoffs.

Theorem 15 *For constant d, the payoff query complexity of degree d graphical games is polynomial.*

Proof Algorithm 1 constructs a directed graph G for the (initially unknown) game, along with the payoff function. G is the "affects graph" (Goldberg and Papadimitriou, 2006) in which a directed edge (p', p) has the meaning that the behaviour of p' may affect p's payoff. Note that in Step 2, $|S| < (n \cdot k)^{d+1}$. In a degree-d graphical game, any player p's payoffs may be affected by his own strategy, and the strategies of at most d neighbours p' for which edges (p', p) exist. The existence of edge (p', p) is equivalent to the existence of strategy profiles s, s' that differ only in p''s strategy and p's payoff. This is what Algorithm 1 checks for. Finally, when the edges, and hence neighborhoods of the graph game have been found,

Algorithm 1 GraphicalGames

Initialize graph G's vertices to be the player set, with no edges
Let S be the set of pure profiles in which at least $n-(d+1)$ players play 1.
Query each element of S.
for all players p, p' **do**
    if $\exists \mathsf{s}, \mathsf{s}' \in S$ that differ only in p's payoff and p''s strategy **then**
        add directed edge (p, p') to graph
    end if
end for
for all players p **do**
    Let N_p be p's neighborhood in G
    Use elements of S to find p's payoffs as a function of strategies of N_p
end for

it is simple to read off each player's payoff matrix from the data in Step 3. ■

Algorithm 1 learns the entire payoff function with polynomially many queries, but there are a couple of important caveats. First, although the payoff query complexity is polynomial, the computational complexity is probably not polynomial, since it is PPAD-complete to actually compute an approximate Nash equilibrium for graphical games (Daskalakis et al., 2009a). Second, while Algorithm 1 avoids querying all of the exponentially-many pure-strategy profiles, it works in a brute-force manner that learns the entire payoff function. It is natural to prefer algorithms that find a solution without learning the entire game, such as those that we give for Theorem 5 and Theorem 26.

5. Congestion Games

In this section, we give bounds on the payoff-query complexity of finding a pure Nash equilibrium in symmetric network congestion games. A congestion game is defined by a tuple $\Gamma = (N, E, (S_i)_{i \in N}, (f_e)_{e \in E})$. Here, $N = \{1, 2, \ldots, n\}$ is a set of n players and E is a set of resources. Each player chooses as her *strategy* a set $s_i \subseteq E$ from a given *set of available strategies* $S_i \subseteq 2^E$. Associated with each resource $e \in E$ is a non-negative, non-decreasing function $f_e : \mathbb{N} \mapsto \mathbb{R}^+$. These functions describe *costs* (latencies) to be charged to the players for using resource e. An outcome (or strategy profile) is a choice of strategies $\mathsf{s} = (s_1, s_2, ..., s_n)$ by players with $s_i \in S_i$. For an outcome s define $n_e(\mathsf{s}) = |i \in N : e \in s_i|$ as the number of players that use resource e. The *cost* for player i is defined by $c_i(\mathsf{s}) = \sum_{e \in s_i} f_e(n_e(\mathsf{s}))$. A *pure Nash equilibrium* is an outcome s where no player has an incentive to deviate from her current strategy. Formally, s is a pure Nash equilibrium if for each player $i \in N$ and $s_i' \in S_i$, which is an alternative strategy for player i, we have $c_i(\mathsf{s}) \leq c_i(\mathsf{s}_{-i}, s_i')$. Here (s_{-i}, s_i') denotes the outcome that results when player i changes her strategy in s from s_i to s_i'.

In a *network congestion game*, resources correspond to the edges in a directed multigraph $G = (V, E)$. Each player i is assigned an origin node o_i, and a destination node d_i. A strategy for player i consists of a sequence of edges that form a directed path from o_i to

d_i, and the strategy set S_i consists of all such paths. In a *symmetric* network congestion game all players have the same origin and destination nodes. We write a symmetric network congestion game as $\Gamma = (N, V, E, (f_e)_{e \in E}, o, d)$, where collectively V, E, o, and d succinctly define the strategy space $(S_i)_{i \in N}$. We consider two types of networks, directed acyclic graphs, and the special case of parallel links. We assume that initially we only know the number of players n and the strategy space. The latency functions are completely unknown initially. As discussed in Section 1.2, we use several different querying models for congestion games.

5.1 Parallel Links

In this section, we consider congestion games on m parallel links. We present a lower bound and an upper bound on the query complexity of finding an exact pure equilibrium of these games. To simplify the presentation of the algorithmic ideas of our upper bound we introduce a stronger type of query that we call an *over-query*. Recall from Definition 2 that for a query $q = (q_1, q_2, \ldots, q_m)$, we denote by Q the total number of players used in the query, i.e., $Q = \sum_{1 \leq i \leq m} q_i$.

Definition 16 *An* over-query *is a query with $n < Q \leq mn$.*

First, we present a simple lower bound. Then, we present an algorithm, Algorithm 2, that uses over-queries. Finally, we extend Algorithm 2 to Algorithm 3, which uses only normal queries.

5.1.1 Lower Bound

In the following construction, we show that, if there are two links, the querier can do no better than performing binary search in order to find an equilibrium, which gives a lower bound of $\log(n)$ many queries.

Theorem 17 *A querier must make $\log(n)$ queries to determine a pure equilibrium of a symmetric network congestion game played on parallel links.*

Proof We fix a graph G with two parallel links e_1 and e_2, and we fix the cost of e_2 so that $f_{e_2}(i) = 1$ for all $i \in N$. We consider functions f_{e_1} that only return costs of 0 or 2. Since f_{e_1} is non-decreasing, this implies that it will be a step function with a single step. We say that the step is at location $i \in N$ if $f_{e_1}(j) = 0$ for all $j \leq i$, and $f_{e_1}(j) = 2$ for all $j > i$. The precise location of the step will be decided by an adversary, in response to the queries that are received.

The adversary's strategy maintains two integers ℓ and u with $\ell < u$, and initially the adversary sets $\ell = 0$ and $u = n$. Intuitively, for all values below ℓ the adversary has fixed f_{e_1} to 0, and for all values above u the adversary has fixed f_{e_1} to 2. The range of values between u and ℓ are yet to be fixed, and all values in this range could potentially be the location of the step.

Suppose that the adversary receives the query s. The adversary will respond with a pair (c_1, c_2), where c_1 is the cost of e_1, and c_2 is the cost of e_2. The adversary uses the following strategy:

- If $n_{e_1}(\mathsf{s}) \leq \ell$, then the adversary responds with $(0, 1)$. If $n_{e_1}(\mathsf{s}) \geq u$, then the adversary responds with $(2, 1)$.
- If $n_{e_1}(\mathsf{s}) < \frac{u+\ell}{2}$, that is, if $n_{e_1}(\mathsf{s})$ is closer to ℓ than it is to u, then the adversary sets $\ell = n_{e_1}(\mathsf{s})$, and responds with $(0, 1)$.
- If $n_{e_1}(\mathsf{s}) \geq \frac{u+\ell}{2}$, that is, if $n_{e_1}(\mathsf{s})$ is closer to u than it is to ℓ, then the adversary sets $u = n_{e_1}(\mathsf{s})$, and responds with $(2, 1)$.

Note that, if there exists an i with $\ell < i < u$, then the querier cannot correctly determine the Nash equilibrium. This is because the step could be at location i, or it could be at location $i - 1$. In the former case, the unique Nash equilibrium assigns i players to e_1 and $n - i$ players to e_2, and in the latter case the unique Nash equilibrium assigns $i - 1$ players to e_1 and $n - i + 1$ players to e_2. By construction, the adversary's strategy ensures that, in response to each query, the gap between u and ℓ may decrease by at most one half. Thus, the querier must make $\log(n)$ queries to correctly determine the Nash equilibrium. ■

Consider a one-player game with m links. Clearly, we can solve this game with a single over-query, but it requires m normal-queries. Thus we have the following:

Corollary 18 *If over-queries are not allowed, then* $\log(n) + m$ *queries are required to determine a pure equilibrium of a symmetric network congestion game played on parallel links.*

5.1.2 Upper Bound

In the rest of the section, we provide an upper bound, by constructing a payoff query algorithm that finds a pure Nash equilibrium using $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)} + m\right)$ normal-queries. In order to simplify the presentation, we first present an algorithm that makes use of over-queries; later we show how this can be translated into an algorithm that uses only normal-queries.

Our algorithm is based on an algorithm from Gairing et al. (2008). Before we present the full algorithm, we give an overview of the techniques by describing a simplified version of the algorithm. The basic idea is to group the players into blocks, where all players in a block must play on the same link. In each round of the algorithm, we maintain the property that the blocks are in equilibrium: no block of players can collectively deviate in order to reduce their latency. Initially, we place all of the players into a single block, and then in each round of the algorithm, we split each block into smaller blocks, and compute a new equilibrium for the smaller block size. Eventually, the block size will be reduced to 1, and we recover a Nash equilibrium for the congestion game.

In this simplified overview, we will assume that the number of players n is equal to 2^i for some $i \in \mathbb{N}$, and in each round we will split each block in half. Our full algorithm will be more complicated, because it must deal with an arbitrary number of players, and it will split each block into more than two pieces.

At the start of the algorithm, we place all n players into a single block. In order to find an equilibrium for this block, we simply have to find the link $i \in [m]$ that minimizes $f_i(n)$. We can do this with a single over-query $q = (n, n, \ldots, n)$.

Now suppose that we have found an equilibrium s for block size δ. We split each block into two equal-sized pieces, and our task is to transform s into an equilibrium for block size $\delta/2$ by moving blocks between the links. The key observation is that no link can receive two or more blocks of size $\delta/2$, because this would contradict the fact that s is an equilibrium for block size δ. So, when we move blocks between the links, we know that each link can receive at most one block, and therefore each link can lose at most $m-1$ blocks. We can make a single over-query in order to discover the cost of adding one block of $\delta/2$ players to each link: we simply query $p = (n_1(\mathsf{s}) + \delta/2, n_2(\mathsf{s}) + \delta/2, \ldots, n_m(\mathsf{s}) + \delta/2)$. On the other hand, we also need to determine how many blocks each link loses, and a naive approach would use m queries. We now describe a method that uses only $\log^2(m)$ under-queries.

Suppose that we guess that q, where $0 \leq q \leq m$, is the number of blocks that move. We give an algorithm that verifies whether this guess is correct. Let c be the $(q+1)$th smallest cost returned by the query p. For each link i, we determine q_i, which is the number of $\delta/2$-sized blocks that would want to move to a link with cost c. This can be done by binary search, in parallel for all links, using $\log(m)$ many under-queries. There are three possible outcomes:

- If $\sum_{i=1}^{m} q_i = q$, then our guess was correct, and exactly q blocks move.
- If $\sum_{i=1}^{m} q_i < q$, then our guess was too high, and fewer than q blocks move.
- If $\sum_{i=1}^{m} q_i > q$, then our guess was too low, and more than q blocks move.

Thus, to determine exactly how many blocks move between the links, we can use a nested binary search approach: in the outer level we guess how many blocks move, and in the inner level we use the above method to determine if our guess was too high or too low.

Therefore, we have a method for constructing an equilibrium with block size $\delta/2$ from an equilibrium with block size δ using $\log^2(m)$ many queries. Since we start with block size n, and we halve the block size in every round, this gives us an algorithm that finds a Nash equilibrium using $\log(n) \cdot \log^2(m)$ many payoff queries.

In the rest of this section, we formalize this approach, and we deal with the issues that were ignored in this high level overview. In particular, we present an algorithm that works for any number of players n, and we obtain a slightly better query complexity by splitting each block into $\log(m)$ many pieces in each round.

5.1.3 The Algorithm With Over-Queries

The algorithm ParallelLinks is depicted in Algorithm 2. We will show how this algorithm can be implemented with $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)}\right)$ queries. The integer k is a parameter to the algorithm that determines the block size: in each round we consider blocks of size k^t for some t. To deal with the fact that n may not be an exact power of k, the algorithm will maintain a special link a. This link is defined to be the link upon which all n players are placed at the start of the algorithm. Since every subsequent step of the algorithm only moves players in blocks of size k^t for some t, link a will be the only link where the number of players is not a multiple of the block size.

We start by formalizing the notion of an equilibrium with respect to a certain block size. For a congestion game Γ, an integer δ, and a special link a we define a δ-equilibrium as follows:

Definition 19 (δ-equilibrium) *A strategy profile* s *is* δ*-equilibrium if* $\delta | n_i(\mathsf{s})$ *for all* $i \in [m] \setminus \{a\}$*, and for all links* $i, j \in [m]$ *with* $n_i(\mathsf{s}) \geq \delta$ *we have* $f_i(n_i(\mathsf{s})) \leq f_j(n_j(\mathsf{s}) + \delta)$*.*

Intuitively, we can think of a δ-equilibrium s as a Nash equilibrium in a transformed game where the players (of the original game) are partitioned into blocks of size δ and each block represents a player in the transformed game, and the remaining (n mod δ) players are fixed to link a.

We start with an informal description of algorithm ParallelLinks. On Line 1 we initialize the algorithm by using one over-query to find the cheapest link a, and assigning all n players to link a. Note that a is the special link, as discussed earlier. The algorithm then works in $T + 1$ phases, where $T = \lfloor \frac{\log(n)}{\log(k)} \rfloor$. Each phase is one iteration of the **for**-loop. The **for**-loop is governed by a variable t, which is initially T and decreases by 1 in each iteration. Within any iteration, the algorithm uses the function RefineProfile to transform a k^{t+1}-equilibrium into a k^t-equilibrium.

Recall, from the overview, that when $k = 2$, we observed that each link can receive at most one block when we transform a 2^{t+1}-equilibrium into a 2^t-equilibrium. In the following lemma, we establish a similar property for the case where $k \neq 2$: each link can receive at most $2k$ blocks. Intuitively, one might expect each link to receive at most k blocks, but the extra factor of two here arises due to the special link a, which was not considered in our simplified overview.

Lemma 20 *We can convert a* k^{t+1}*-equilibrium* s *into a* k^t*-equilibrium* s' *by moving at most* $2k$ *blocks of* $\delta = k^t$ *players to any individual link and at most* km *blocks of* δ *players in total.*

Proof Since s is k^{t+1}-equilibrium, we have $f_i(n_i(\mathsf{s})) \leq f_j(n_j(\mathsf{s}) + k^{t+1})$ for all $i \in [m] \setminus \{a\}, j \in [m]$. Moreover, either (a) $f_a(n_a(\mathsf{s})) \leq f_j(n_j(\mathsf{s}) + k^{t+1})$ for all $j \in [m]$ or (b) $n_a(\mathsf{s}) < k^{t+1}$. In case (a), this implies that each link $j \in [m]$ can in total receive at most k blocks of size $\delta = k^t$ from links $i \in [m]$. In case (b), this implies that each link $j \in [m]$ can in total receive at most k blocks of size $\delta = k^t$ from links $i \in [m] \setminus \{a\}$. Moreover, since $n_a(\mathsf{s}') < k^{t+1}$, we can move at most k blocks of size $\delta = k^t$ from link a. In either case, in total we move at most km blocks. All links receive and lose players only in multiples of $\delta = k^t$, which ensures that $k^t | n_i(\mathsf{s}')$ for all $i \in [m] \setminus \{a\}$ is maintained. ■

RefineProfile determines the number of blocks q which have to be moved by binary search on q in $[0, km]$. Since, by Lemma 20, each link receives at most $2k$ blocks of players, we spend $2k$ over-queries to determine the cost function values $f_i(n_i(\mathsf{s}) + r \cdot \delta)$ for all integers $r \leq 2k$ and all links $i \in [m]$. We define Q as the multi-set of these cost function values and $C_{min}(q)$ as the $(q+1)$-th smallest value in Q. Intuitively, $C_{min}(q)$ is the cost of the $(q+1)$-th block of players that we would move. We use $C_{min}(q)$ to find out how many blocks of players q_i we need to remove from each link $i \in [m]$ so that on each link $i \in [m]$ the cost is at most $C_{min}(q)$ or we can't remove any further blocks as there are less than δ players

Algorithm 2 ParallelLinks

$a \leftarrow \arg\min_{i \in [m]} f_i(n)$ ▷ 1 over-query
initialize strategy profile s by putting all players on link a
$T \leftarrow \lfloor \frac{\log(n)}{\log(k)} \rfloor$
for $t = T, T-1, \ldots, 1, 0$ **do**
    $\delta \leftarrow k^t$
    $\mathsf{s} \leftarrow$ RefineProfile$(\mathsf{s}, \delta, 0, km)$
end for
return s

function RefineProfile$(\mathsf{s}, \delta, q_{min}, q_{max})$
    $q \leftarrow \lfloor \frac{q_{min}+q_{max}}{2} \rfloor$
    Parallel for all links $i \in [m]$
        Query for costs $f_i(n_i(\mathsf{s}) + r\delta)$ for all integer $1 \leq r \leq 2k$ ▷ $2k$ queries
    EndParallel
    $Q \leftarrow$ the ordered multiset of $2km$ non-decreasing costs from the above queries
    $C_{min}(q) \leftarrow (q+1)$-th smallest element of Q
    $p_i \leftarrow$ number of times $i \in [m]$ contributes a cost to the q smallest elements of Q
    Parallel for all links $i \in [m]$
        if $f_i(n_i(\mathsf{s}) - \lfloor \frac{n_i(\mathsf{s})}{\delta} \rfloor \cdot \delta) > C_{min}(q)$ **then** ▷ 1 query; only relevant for link a
            $q_i \leftarrow \lfloor \frac{n_i(\mathsf{s})}{\delta} \rfloor$
        else (using binary search on $q_i \in [0, \min\{km, \lfloor \frac{n_i(\mathsf{s})}{\delta} \rfloor\}]$)
            $q_i \leftarrow \min\{q_i : f_i(n_i(\mathsf{s}) - q_i\delta) \leq C_{min}(q)\}$ ▷ $\log(km)$ queries
        end if
    EndParallel
    if $\sum_{i \in [m]} q_i = q$ **then**
        modify s by removing q_i and adding p_i blocks of δ players to every link $i \in [m]$
        return s
    else if $\sum_{i \in [m]} q_i < q$ **then**
        return RefineProfile$(\mathsf{s}, \delta, q_{min}, q-1)$
    else $(\sum_{i \in [m]} q_i > q)$
        return RefineProfile$(\mathsf{s}, \delta, q+1, q_{max})$
    end if
end function

assigned to it (which can only happen on link a). By Lemma 20, we need to remove at most km blocks of players in total. Therefore, we can determine $q_i \in [0, \min\{km, \lfloor \frac{n_i(\mathsf{s})}{\delta} \rfloor\}]$ by binary search in parallel on all links, with $\mathcal{O}(\log(km))$ under-queries. Now, if $\sum_{i=1}^{m} q_i = q$, we can construct a k^t-equilibrium by removing q_i and adding p_i blocks of δ players to link $i \in [m]$; note that for every $i \in [m]$, either $q_i = 0$ or $p_i = 0$. If $\sum_{i=1}^{m} q_i \neq q$, our guess for q was not correct and we have to continue the binary search on q.

The algorithm maintains the following invariant:

Lemma 21 RefineProfile$(\mathsf{s}, \delta, 0, km)$ *returns a δ-equilibrium.*

Proof Observe that $\delta = k^t$. In the first iteration of the **for**-loop $t = T$ and RefineProfile$(\mathsf{s}, \delta, 0, km)$ gets a n-equilibrium as input, which is also a k^{T+1}-equilibrium as all players are assigned to link a and $k^{T+1} > n$. So to prove the claim, it suffices to show that RefineProfile$(\mathsf{s}, k^t, 0, km)$ returns a k^t-equilibrium if s is a k^{t+1}-equilibrium. For the s returned by RefineProfile and the q in its returning call, we have $f_i(n_i(\mathsf{s})) \leq C_{min}(q) \leq f_i(n_i(\mathsf{s}) + \delta)$ for all $i \in [m] \setminus \{a\}$. The left inequality follows from line 21 of the algorithm. The right inequality follows from the definition of $C_{min}(q)$ as the $(q+1)$-th smallest element in Q in line 15 of the algorithm. For link a, we have $f_a(n_a(\mathsf{s})) \leq C_{min}(q) \leq f_a(n_a(\mathsf{s}) + \delta)$ or we have $f_a(n_a(\mathsf{s})) > C_{min}(q)$ and $n_a(\mathsf{s}) < \delta$, where the first case follows from lines 21 and 15 as before, and the second case corresponds to line 18. Noting that RefineProfile maintains that for the returned s we have $\delta | n_i(\mathsf{s})$ for all $i \in [m] \setminus \{a\}$, as it only moves blocks of size δ, the claim follows. ■

We now give the payoff query complexity of RefineProfile. We split our analysis into over-queries and non-over-queries (i.e., under-queries or normal-queries), because we will later show how the over-queries made by our algorithm can be translated into a sequence of non-over-queries.

Lemma 22 RefineProfile$(\mathsf{s}, \delta, 0, km)$ *can be implemented to make $2k$ over-queries and $\mathcal{O}(\log^2(km))$ non-over-queries.*

Proof Note that, as long as δ is not changed, the queries made on line 12 are the same for each pair of q_{min} and q_{max}. Therefore, we can perform these $2k$ over-queries when we first call RefineProfile$(\mathsf{s}, \delta, 0, km)$, and reuse these values during each recursive call. For each value of q in the binary search, we make $\mathcal{O}(\log(km))$ under-queries to determine the q_i's in parallel for all links $i \in [m]$. The binary search on q adds a factor $\log(km)$ to give $\mathcal{O}(\log^2(km))$ under-queries in total. ■

Using Lemmas 21 and 22 we can prove the following.

Theorem 23 *Algorithm* ParallelLinks *returns a pure Nash equilibrium and can be implemented with $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)}\right)$ queries, of which $2k \cdot \frac{\log n}{\log\log m}$ are over-queries.*

Proof In the last iteration of the **for**-loop, we have $\delta = 1$, so Lemma 21 implies that s is a pure Nash equilibrium. To find the best link in line 1 of the algorithm, we need one

over-query. For any $k \geq 2$, the algorithm does $T + 1 = \mathcal{O}\left(\frac{\log(n)}{\log(k)}\right)$ iterations of the **for**-loop. In each iteration we do $\mathcal{O}(\log^2(km))$ under-queries and $2k$ over-queries. Choosing $k = \Theta(\log(m))$ yields the stated upper bound. ■

5.1.4 Using Only Normal-Queries

We now show how Algorithm 2 can be implemented without the use of over-queries. Before doing so, we remark that in the parallel links setting, we can also avoid using under-queries.

Lemma 24 *If a parallel links congestion game has at least two links, then every under-query can be translated into two normal-queries.*

Proof Suppose that the game has $m \geq 2$ links, and let $q = (i_1, i_2, \ldots, i_m)$ be an under-query. Let $n' = \sum_{j=1}^{m} i_j$ be the total number of players used by q. We define the following queries:

$$q_1 = (i_1 + n - n', i_2, \ldots, i_m),$$
$$q_2 = (i_1, i_2, \ldots, i_m + n - n').$$

Clearly both q_1 and q_2 are normal-queries. Query q_1 tells us the cost of links 2 through m under q, and query q_2 tells us the cost of link 1 under q. ■

We now turn our attention to over-queries. The following lemma gives a general method for translating over-queries into non-over-queries.

Lemma 25 *Suppose we have a parallel links game with m links and n players. Let $q = (i_1, i_2, \ldots, i_m)$ be an over-query, and define $n' = \sum_{j=1}^{m} i_j$. We can translate q into a sequence of $\mathcal{O}(n'/n)$ non-over-queries.*

Proof Consider the following greedy algorithm: find the smallest index b such that $\sum_{1 \leq k \leq b} i_k \leq n$ and assign links 1 through b to query q_1. Set $i_1 = i_2 = \cdots = i_b = 0$, and repeat. Clearly each query that we generate during this algorithm is a non-over-query.

Let $q_1, q_2, \ldots, q_l$ be the sequence of non-over-queries generated by the above algorithm for some $l \in \mathbb{N}$. For each j, let n_j be the total number of players used by q_j, and observe that $\sum_{1 \leq j \leq l} n_j = n'$. Furthermore, for each j, let $r_j = n - n_j$ be the total number of players *not* used by q_j. Due to the nature of our algorithm, for every $j > 1$ we must have $r_{j-1} < n_j$, since the first link assigned to q_j would not fit in q_{j-1}. Thus, we have:

$$\begin{aligned}\sum_{1 \leq j \leq l} r_j &< \sum_{2 \leq j \leq l} n_j + r_1 \\ &< n' + n.\end{aligned}$$

Since the total number of queries in the sequence is l, we can argue that:

$$\begin{aligned} l &= \frac{1}{n} \sum_{1 \le j \le l} (n_j + r_j) \\ &< \frac{n' + n' + n}{n} \\ &= 1 + \frac{2n'}{n}. \end{aligned}$$

Thus, our greedy algorithm generates at most $\mathcal{O}(n'/n)$ non-over-queries. ■

In order to optimize the number of non-over queries we have to adjust Algorithm 2 slightly, because with $k = \Theta(\log(m))$ in early iterations of the for loop, i.e., when T is large, the number of players used in the over queries in line (12) is large and applying Lemma 25 would yield to a total of $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)} + m\log(m)\right)$ non-over queries. In contrast, we will now show that our adjusted Algorithm 3 can be implemented to do at most $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)} + m\right)$ non-over queries. The main idea is to divide the block size by 2 until the number of players in a block is small enough and then switch to $k = \Theta(\log(m))$.

Algorithm 3 ParallelLinks avoiding over-queries

1: $a \leftarrow \arg\min_{i \in [m]} f_i(n)$ ▷ 1 over-query
2: initialize strategy profile s by putting all players on link a
3: $T \leftarrow \left\lfloor \frac{\log(n/m)}{\log(k)} \right\rfloor$
4: $T_0 \leftarrow$ largest t such that $k^T 2^t < n$
5: **for** $t = T_0, T_0 - 1, \ldots, 1$ **do**
6: $\quad \delta \leftarrow k^T 2^t$
7: $\quad \mathsf{s} \leftarrow$ RefineProfile$(\mathsf{s}, \delta, 0, 2m)$
8: **end for**
9: **for** $t = T, T - 1, \ldots, 1, 0$ **do**
10: $\quad \delta \leftarrow k^t$
11: $\quad \mathsf{s} \leftarrow$ RefineProfile$(\mathsf{s}, \delta, 0, km)$
12: **end for**
13: **return** s

To initialize our algorithm, we make an over-query that uses $m \cdot n$ players. By Lemma 25, we can translate this into $\mathcal{O}(m)$ non-over-queries.

In each iteration of the first **for**-loop with value t, by Lemma 22, we make $\mathcal{O}(1)$ over-queries. Each of these uses at most $n + m \cdot 4 \cdot k^T 2^t$ players. By Lemma 25, these can be simulated by $\mathcal{O}(1 + \frac{mk^T 2^t}{n})$ non-over-queries. Summing up over all iterations and using the definition of T_0, we can argue that all over-queries of the first **for**-loop can be simulated by

$$\sum_{t=1}^{T_0} \mathcal{O}\left(1 + \frac{mk^T 2^t}{n}\right) = \mathcal{O}(T_0) + \mathcal{O}\left(\frac{mk^T 2^{T_0}}{n}\right) = \mathcal{O}(m)$$

non-over-queries.

In each iteration of the second **for**-loop with value t, by Lemma 22, we make make $2k$ over-queries that each use at most $n + m \cdot 2k \cdot k^t$ players. By Lemma 25, these can be simulated by $\mathcal{O}(\frac{mk^{t+1}}{n})$ non-over-queries. Summing up over all iterations, we can argue that all over-queries of the second **for**-loop can be simulated by

$$\begin{aligned}\sum_{t=0}^{\lfloor \frac{\log(n/m)}{\log(k)} \rfloor} \mathcal{O}\left(\frac{mk^{t+1}}{n}\right) &= \mathcal{O}\left(\frac{m}{n} \cdot k^{\frac{\log(n/m)}{\log(k)}+1}\right) \\ &= \mathcal{O}\left(\frac{m}{n} \cdot k^{\frac{\log(n)-\log(m)+\log(k)}{\log(k)}}\right) \\ &= \mathcal{O}\left(\frac{m}{n} \cdot k^{\frac{\log(n)}{\log(k)}}\right) \\ &= \mathcal{O}(m)\end{aligned}$$

non-over-queries.

Combining this discussion with Theorem 23, we get the following result:

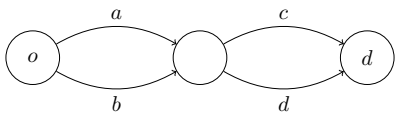
Theorem 26 *Algorithm 3 returns a pure Nash equilibrium and can be implemented with* $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)} + m\right)$ *queries.*

The upper bound in Theorem 26 should be contrasted with the lower bound of $\log(n)+m$ (Corollary 18).

5.2 Symmetric Network Congestion Games on Directed Acyclic Graphs

In this section, we consider symmetric network congestion games on directed acyclic graphs. Throughout this section, we consider the game $\Gamma = (N, V, E, (f_e)_{e \in E}, o, d)$, where (V, E) is a directed acyclic graph (DAG). We use the $\prec$ relation to denote a topological ordering over the vertices in V. We assume that, for every vertex $v \in V$, there exists a path from o to v, and there exists a path from v to d. If either of these conditions does not hold for some vertex v, then v cannot appear on an o-d path, and so it is safe to delete v.

We provide an algorithm that discovers a cost function for each edge. One immediate observation is that we can never hope to find the actual cost functions. Consider the following one-player congestion game.



If we set $f_a(1) = f_b(1) = 1$ and $f_c(1) = f_d(1) = 0$, then all o-d paths have cost 1. However, we could also achieve the same property by setting $f_a(1) = f_b(1) = 0$ and setting $f_c(1) = f_d(1) = 1$. Thus, it is impossible to learn the actual cost functions using payoff queries.

To deal with this issue, we introduce the notion of an *equivalent* cost function: two cost functions are said to be equivalent if they assign the same cost to every strategy profile. We show that, while it is impossible to find the actual cost function via payoff queries, we can use payoff queries to find an equivalent cost function.

Our algorithm proceeds inductively over the number of players in the game. For the base case, we give an algorithm that finds an equivalent cost function f' such that $f'_e(1)$ is defined for every edge e. This corresponds to learning all the costs in a one-player congestion game played on Γ. Then, for the inductive step, we show how the costs for an i-player game can be used to find the costs in an $i+1$ player game. That is, we use the known values of $f'_e(j)$ for $j \leq i$ to find the cost of $f'_e(i+1)$ for every edge e. Therefore, at the end of the algorithm, we have an equivalent cost function f' for an n-player game on Γ, and we can then apply a standard congestion game algorithm (Fabrikant et al., 2004) in order to solve our game.

Unlike our work on parallel links, in this section we will not use over-queries at all. In each inductive step, when we are considering an i-player congestion game, we will make queries that use exactly i players. Thus, in the first $n-1$ rounds we will use under-queries, and in the final round we will use normal-queries. For the sake of brevity, in this section we will use the word "query" to refer to both normal and under-queries.

As a shorthand for defining queries, we use notation of the form $\mathsf{s} \leftarrow (1 \mapsto p, 3 \mapsto q)$. This example defines s to be a four-player query that assigns 1 player to p and 3 players to q, where p and q are paths from the origin to the destination in a symmetric network congestion game. We use Query(s) to denote the outcome of querying s. It returns a function c_{s}, which gives the cost of each strategy when s is played.

5.2.1 Preprocessing

Our algorithm requires a preprocessing step. We say that edges e and e' are *dependent* if visiting one implies that we must visit the other. More formally, e and e' are dependent if, for every o-d path p, we either have $e, e' \in p$, or we have $e, e' \notin p$. We preprocess the game to ensure that there are no pairs of dependent edges. To do this, we check every pair of edges e and e', and test whether they are dependent. If they are, then we *contract* e', i.e., if $e' = (v, u)$, then we delete e', and set $v = u$. The following lemma shows that this preprocessing is valid, and therefore, from now on, we can assume that our congestion game contains no pair of dependent edges.

Lemma 27 *There is an algorithm that, given a congestion game Γ, where (V, E) is a DAG, produces a game Γ' with no pair of dependent edges, such that every Nash equilibrium of Γ' can be converted to a Nash equilibrium of Γ. The algorithm and conversion of equilibria take polynomial time and make zero payoff queries. Moreover, payoff queries to Γ' can be trivially simulated with payoff queries to Γ.*

Proof Our algorithm will check, for each pair of edges $e = (v, u)$ and $e' = (v', u')$, whether e and e' are dependent. This is done in the following way. Note that if $v = v'$, then e and e' cannot possibly be dependent. Thus, we can assume without loss of generality that $v \prec v'$. The algorithm performs two checks:

- Delete e and verify that there is no path from o to v'.
- Delete e' and verify that there is no path from u to d.

The first check ensures that every path that uses e' must also use e. The second check ensures that every path that uses e must also use e'. Thus, if both checks are satisfied,

then e and e' are dependent. On the other hand, if one of the checks is not satisfied, then we can construct an o-d path that uses e and not e', or a path that uses e' and not e, which verifies that e and e' are not dependent.

Whenever the algorithm finds a pair of edges $e, e' \in E$ that are dependent, it contracts e'. More formally, if $e' = (v, u)$, then the algorithm constructs a new congestion game $\Gamma' = (N, V', E', (f'_e)_{e \in E'}, o, d)$ where $V' = V \setminus \{u\}$, and E' contains:

- every edge $(w, x) \in E$ with and $w \neq u$, and
- an edge (v, x) for every edge $(u, x) \in E$.

Note that E' does not contain e'. Moreover, we define the cost functions f' as follows. For each edge $e'' \neq e$, we set $f'_{e''}(i) = f_{e''}(i)$ for all i. For the edge e, we define $f'_e(i) = f_e(i) + f_{e'}(i)$ for all i.

We argue that this operation is correct. Since e and e' are dependent, we have that, for every strategy profile s, and for every o-d path p:

$$\sum_{e'' \in p} f'_{e''}(i) = \sum_{e'' \in p} f_{e''}(i).$$

Therefore, we can easily translate every Nash equilibrium of Γ' into a Nash equilibrium for Γ. Moreover, every payoff query for Γ' can be translated into a payoff query for Γ by adding the edge e' where appropriate.

Thus, the algorithm constructs a sequence of games $\Gamma_1, \Gamma_2, \ldots$, where each game Γ_{i+1} is obtained by contracting an edge in Γ_i. Moreover, the Nash equilibria for Γ_{i+1} can be translated to Γ_i, which implies that the algorithm is correct. This algorithm can obviously be implemented in polynomial time. Moreover, since the algorithm only inspects structural properties of the graph, it does not make any payoff queries. ■

5.2.2 Equivalent Cost Functions

As we have mentioned, we cannot hope to find the actual cost function of Γ using payoff queries. To deal with this, we introduce the following notion of equivalence.

Definition 28 (Equivalence) *Two cost functions f and f' are equivalent if for every strategy profile $\mathsf{s} = (s_1, s_2, \ldots, s_n)$, we have $\sum_{e \in s_i} f_e(n_e(\mathsf{s})) = \sum_{e \in s_i} f'_e(n_i(\mathsf{s}))$, for all i.*

Clearly, the Nash equilibria of a game cannot change if we replace its cost function f with an equivalent cost function f'.

We say that $(f'_e)_{e \in E}$ is a *partial* cost function if for some $e \in E$ and some $i \leq n$, $f'_e(i)$ is undefined. We say that f'' is an *extension* of f' if f'' is a partial cost function, and if $f''_e(i) = f'_e(i)$ for every $e \in E$ and $i \leq n$ for which $f'_e(i)$ is defined. We say that f'' is a *total extension* of f' if f'' is an extension of f', and if $f''_e(i)$ is defined for all $e \in E$ and all $i \leq n$.

Definition 29 (Partial equivalent cost function) *Let f be a cost function. We say that f' is a partial equivalent of f if f' is a partial cost function, and if there exists a total extension f'' of f' such that f'' is equivalent to f.*

Our goal is to find a total equivalent cost function by learning the costs one edge at a time. Thus, our algorithm will begin with a partial cost function f^0 such that $f^0_e(i)$ is undefined for all $e \in E$ and all $i \leq n$. Since it is undefined everywhere, it is obvious that f^0 is a partial equivalent of f. At every step of the algorithm, we will take a partial equivalent cost function f' of f, and produce an extension f'' of f', such that f'' is still a partial equivalent of f. This guarantees that, when the algorithm terminates, the final cost function is equivalent to f.

5.3 The One-Player Case

For the one player case, our algorithm is relatively straightforward. The algorithm proceeds iteratively by processing the vertices according to their topological order, starting from the origin vertex o, and moving towards the destination vertex d. Each time we process a vertex k, we determine the cost of every incoming edge (u, k). There are two different cases: the case where $k \neq d$, and the case where $k = d$. For the latter case, we will observe that, once we know the cost of every edge other than the incoming edges to d, we can easily find the cost of the incoming edges to d.

The former case is slightly more complicated. When we consider a vertex $k \neq d$, it turns out that we cannot find the actual costs for the incoming edges at k. Instead, we can use payoff queries to discover the difference in cost between each pair of incoming edges, and therefore, we can find the cheapest incoming edge e to k. We proceed by fixing the cost of e to be 0. Once we have done this, we can then set the cost of each other incoming edge e' according to the difference between the cost of e and the cost of e', which we have already discovered. We prove that this approach is correct by showing that it yields a partial equivalent cost function.

We now formally describe our algorithm. The algorithm begins with the partial cost function f^0. The algorithm processes vertices iteratively according to the topological ordering $\prec$. Suppose that we are in iteration $a + 1$ of the algorithm, and that we are processing a vertex $k \in V$. We have a partial equivalent cost function f^a such that $f^a_e(1)$ is defined for every edge $e = (v, u)$ with $u \prec k$, for some vertex k. We then produce a partial equivalent cost function f^{a+1} such that $f^{a+1}_e(1)$ is defined for every edge $e = (v, u)$ with $u \preceq k$. We now consider the two cases.

5.3.1 The $k \neq d$ Case

We use the procedure shown in Algorithm 4 to process k. Lines 1 through 3 simply copy the old cost function f^a into the new cost function f^{a+1}. This ensures that f^{a+1} is an extension of f^a. The algorithm then picks an arbitrary k-d path p. The loop on lines 5 through 10 compute the function t, which for each incoming edge $e = (v, k)$, gives the cost $t(ep)$ of allocating one player to ep. Note, in particular, that the value of the expression $\sum_{e' \in p'} f^a_{e'}(1)$ is known to the algorithm, because every vertex visited by p' has already been processed. The algorithm then selects e' to be the edge that minimizes t, and sets the cost of e' to be 0. Once it has done this, lines 13 through 15 compute the costs of the other edges relative to e'.

When we set the cost of e' to be 0, we are making use of equivalence. Suppose that the actual cost of e' is $c_{e'}$. Setting the cost of e' to be 0 has the following effects:

Algorithm 4 ProcessK

Input: A partial equivalent cost function f^a, such that $f^a_e(1)$ is defined for all edges (v, u) with $u \prec k$.

Output: A partial equivalent cost function f^{a+1}, such that $f^{a+1}_e(1)$ is defined for all edges (v, u) with $u \preceq k$.

1: **for all** e for which $f^a_e(1)$ is defined **do**
2: $\quad f^{a+1}_e(1) \leftarrow f^a_e(1)$
3: **end for**
4: $p \leftarrow$ an arbitrary k-d path
5: **for all** $e = (v, k) \in E$ **do**
6: $\quad p' \leftarrow$ an arbitrary o-v path
7: $\quad \mathsf{s} \leftarrow (1 \mapsto p'ep)$
8: $\quad c_{\mathsf{s}} \leftarrow \text{Query}(\mathsf{s})$
9: $\quad t(ep) \leftarrow c_{\mathsf{s}}(p'ep) - \sum_{e' \in p'} f^a_{e'}(1)$
10: **end for**
11: $e' \leftarrow$ edge $e = (v, k)$ that minimizes $t(ep)$
12: $f^{a+1}_{e'}(1) \leftarrow 0$
13: **for all** $e = (v, k) \in E$ with $e \neq e'$ **do**
14: $\quad f^{a+1}_e(1) \leftarrow t(ep) - t(e'p)$
15: **end for**

- Every incoming edge at k has its cost reduced by $c_{e'}$.
- Every outgoing edge at k has its cost increased by $c_{e'}$.

This maintains equivalence with the original cost function, because for every path p that passes through k, the total cost of p remains unchanged. The following lemma formalizes this and proves that f^{a+1} is indeed a partial equivalent cost function.

Lemma 30 *Let $k \neq d$ be a vertex, and let f^a be a partial equivalent cost function such that $f^a_e(1)$ is defined for all edges $e = (v, u)$ with $u \prec k$. When given these inputs, Algorithm 4 computes a partial equivalent cost function f^{a+1} such that $f^{a+1}_e(1)$ is defined for all edges $e = (v, u)$ with $u \preceq k$.*

Proof It can be verified that the algorithm assigns a cost to $f^{a+1}_e(1)$ for every edge $e = (v, u)$ with $u \preceq k$. To complete the proof of the lemma, we must show that f^{a+1} is a partial equivalent cost function. Since f^a is a partial equivalent cost function, there must exist a total extension of f^a that is equivalent to f. Let f' denote such an extension. We use f' to construct f'', which is a total extension of f^{a+1} that is equivalent to f.

Let $e = (v, k)$ be an incoming edge at k. We begin by deriving a formula for $t(ep)$, which is computed on line 9. Note that, since f' is equivalent to f, we have $c_{\mathsf{s}}(p'ep) = \sum_{e' \in p'ep} f'_{e'}(1)$. Note also that $f'_{e'}(1) = f^a_{e'}(1)$ for every edge $e' \in p'$. Therefore, we have the

following:

$$\begin{aligned} t(ep) &= c_{\mathsf{s}}(p'ep) - \sum_{e' \in p'} f^a_{e'}(1) \\ &= \sum_{e' \in p'ep} f'_{e'}(1) - \sum_{e' \in p'} f'_{e'}(1) \\ &= \sum_{e' \in ep} f'_{e'}(1). \end{aligned}$$

For each edge $e = (v, k)$ with $e \neq e'$, line 14 sets:

$$\begin{aligned} f^{a+1}_e(1) &= t(ep) - t(e'p) \\ &= \sum_{e' \in ep} f'_{e'}(1) - \sum_{e' \in e'p} f'_{e'}(1) \\ &= f'_e(1) - f'_{e'}(1). \end{aligned}$$

Note also that line 12 sets:

$$f^{a+1}_{e'}(1) = 0 = f'_{e'}(1) - f'_{e'}(1).$$

Hence, we can conclude that $f^{a+1}_e(1) = f'_e(1) - f'_{e'}(1)$ for every incoming edge $e = (v, k)$.

We construct the total cost function f'' as follows. For every edge $e = (v, u)$, and every $i \leq n$, we set:

$$f''_e(i) = \begin{cases} f'_e(i) - f'_{e'}(1) & \text{if } u = k, \\ f'_e(i) + f'_{e'}(1) & \text{if } v = k, \\ f'_e(i) & \text{otherwise.} \end{cases}$$

Since we have shown that $f^{a+1}_e(1) = f'_e(1) - f'_{e'}(1)$ for every incoming edge $e = (v, k)$, we have that $f''_e(1)$ is a total extension of f^{a+1}.

We must now show that f''_e and f are equivalent. We will do this by showing that f'' and f' are equivalent. Let $\mathsf{s} = (s_1, s_2, \ldots, s_n)$ be an arbitrarily chosen strategy profile. If s_i does not visit k, then we have:

$$\sum_{e \in s_i} f''_e(n_e(\mathsf{s})) = \sum_{e \in s_i} f'_e(n_e(\mathsf{s})).$$

On the other hand, if s_i does visit k, then it must use exactly one edge (v, u) with $u = k$, and exactly one edge (v, u) with $v = k$. Therefore, we have:

$$\begin{aligned} \sum_{e \in s_i} f''_e(n_e(\mathsf{s})) &= \sum_{e \in s_i} f'_e(n_e(\mathsf{s})) - f'_{e'}(1) + f'_{e'}(1) \\ &= \sum_{e \in s_i} f'_e(n_e(\mathsf{s})). \end{aligned}$$

Therefore, f'' is equivalent to f', which also implies that it is equivalent to f. Thus, we have found a total extension of f^{i+1} that is equivalent to f, as required. ■

5.3.2 The $k = d$ Case

When the algorithm processes d, it will have a partial cost function f^a such that $f^a_e(1)$ is defined for every edge $e = (v, u)$ with $u \neq d$. The algorithm is required to produce a partial cost function f^{a+1} such that $f^{a+1}_e(1)$ is defined for all $e \in E$. We use Algorithm 5 to do this. Lines 1 through 3 ensure that f^{a+1} is equivalent to f^a. Then, the algorithm loops

Algorithm 5 ProcessD

Input: A partial equivalent cost function f^a, such that $f^a_e(1)$ is defined for all edges $e = (v, u)$ with $u \prec d$.

Output: A partial equivalent cost function f^{a+1}, such that $f^a_e(1)$ is defined for all edges $e \in E$.

1: **for all** e for which $f^a_e(1)$ is defined **do**
2: $\quad f^{a+1}_e(1) \leftarrow f^a_e(1)$
3: **end for**
4: **for all** $e = (v, d) \in E$ **do**
5: $\quad p \leftarrow$ an arbitrary o-v path
6: $\quad \mathsf{s} \leftarrow (1 \mapsto pe)$
7: $\quad c_{\mathsf{s}} \leftarrow \text{Query}(\mathsf{s})$
8: $\quad f^{a+1}_e(1) \leftarrow c_{\mathsf{s}}(pe) - \sum_{e' \in p} f^a_{e'}(1)$
9: **end for**

through each incoming edge $e = (v, d)$, and line 8 computes $f^{a+1}_e(1)$. Note, in particular, that $f^a_{e'}(1)$ is defined for every edge $e' \in p$, and thus the computation on line 8 can be performed. Lemma 31 shows that Algorithm 5 is correct.

Lemma 31 *Let $k \neq d$ be a vertex, and let f^a be a partial equivalent cost function defined for all edges (v, u) with $u \prec d$. When given these inputs, Algorithm 5 computes a partial equivalent cost function f^{a+1}.*

Proof Since f^a is a partial equivalent cost function, there must exist a cost function f' that is an extension of f^a, where f' is equivalent to f. We show that f' is also an extension of f^{a+1}.

Let $e = (v, d)$ be an incoming edge at d. Consider line 8 of the algorithm. Note that, since f' is equivalent to f, we have $c_{\mathsf{s}}(pe) = \sum_{e' \in pe} f'_{e'}(1)$. Furthermore, since f' is an extension of f^{a+1}, we have $f^a_{e'}(1) = f'_{e'}(1)$ for every $e' \in p$. Therefore, we have:

$$\begin{aligned} f^{a+1}_e(1) &= c_{\mathsf{s}}(pe) - \sum_{e' \in p} f^a_{e'}(1) \\ &= \sum_{e' \in pe} f'_{e'}(1) - \sum_{e' \in p} f'_{e'}(1) \\ &= f'_e(1). \end{aligned}$$

We also have $f^{a+1}_e(1) = f'_e(1)$ for every edge $e = (v, u)$ with $u \prec d$, and we have shown that $f^{a+1}_e(1) = f'_e(1)$ for every edge $e = (v, u)$ with $u = d$. Therefore f' is an extension of f^{a+1}, which implies that f^{a+1} is a partial equivalent cost function. ■

The algorithm makes exactly $|E|$ payoff queries in order to find the one-player costs. When Algorithm 4 processes a vertex k, it makes exactly one query for each incoming edge (v, k) at k. The same property holds for Algorithm 5. This implies that, in total, the algorithm makes $|E|$ queries.

5.4 The Many-Player Case

In this section, we will assume that we have a partial equivalent cost function f^a such that $f^a_e(j)$ is defined whenever $j \leq i$. We will give an algorithm that goes through a sequence of iterations and produces a partial cost function $f^{a'}$, such that $f^{a'}_e(j)$ is defined whenever $j \leq i + 1$.

The algorithm for the many-player case proceeds in a similar fashion to the algorithm for the one-player case. The algorithm is still iterative, and it still processes vertices according to their topological order, starting from the origin o, and moving towards the destination d. In this algorithm, when we process a vertex k, we will discover, for each incoming edge e to k, the cost of placing $i + 1$ players on e.

However, there is an additional complication. Our technique for discovering the cost of placing $i + 1$ players on the incoming edge at k requires two edge disjoint paths from k to d, but there is no reason at all to assume that two such paths exist. We say that an edge e is a bridge between two vertices v and u, if every v-u path contains e. Furthermore, if we fix a vertex $k \in V$, then we say that an edge e is a k-bridge if e is a bridge between k-d. The following lemma can be proved using the max-flow min-cut theorem and is a variant of Menger's theorem.

Lemma 32 *Let v and u be two vertices. There are two edge disjoint paths between v and u if, and only if, there is no bridge between v and u.*

Proof Let (V, E) be a graph, and let $v, u \in V$ be two vertices. We construct a network flow instance where every edge $e \in E$ has capacity 1, and we ask for the maximum flow between v and u. Since each edge has capacity 1, we have that the maximum flow between v and u is greater than 1 if, and only if, there are two edge-disjoint paths between v and u. Moreover, by the max-flow min-cut theorem, the maximum flow from v to u is greater than 1 if and only if there is no bridge between v and u. ■

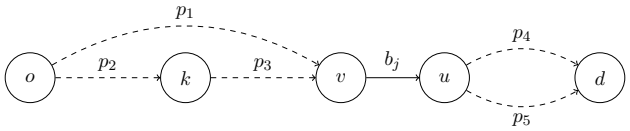
As a consequence of Lemma 32, we can only process k if there are no k-bridges. To resolve this, before attempting to process k, we first use a separate algorithm to determine the cost of placing $i + 1$ players on each k-bridge. After doing this, we can then find two k-d paths that are edge disjoint *except for k bridges*. This, combined with the fact that we know the cost of placing $i+1$ players on each k-bridge, is sufficient to allow us to process k.

The remainder of this section will proceed as follows. We first describe our algorithm for finding the costs of the k bridges. After doing so, we then describe our algorithm for processing k.

5.4.1 Bridges

Given a vertex k, we show how to determine the cost of the k-bridges. Let b_1, b_2, ..., b_m denote the list of k-bridges sorted according to the topological ordering $\preceq$. That is, if $b_1 = (v_1, u_1)$, and $b_2 = (v_2, u_2)$, then we have $v_1 \prec v_2$, and so on. Our algorithm is given a partial cost function f^a, such that $f^a_e(j)$ is defined for all $j \leq i$, and returns a cost function f^{a+1} that is an extension of f^a where, for all ℓ, we have that $f^{a+1}_{b_\ell}(i+1)$ is defined.

Our algorithm processes the k-bridges in reverse topological order, starting with the final bridge b_m. Suppose that we are processing the bridge $b_j = (v, u)$. We will make one payoff query to find the cost of b_j, which is described by the following diagram.



The dashed lines in the diagram represent paths. They must satisfy some special requirements, which we now describe. The paths p_4 and p_5 must be edge disjoint, apart from k-bridges. The following lemma shows that we can always select two such paths.

Lemma 33 *For each k-bridge $b_j = (v, u)$, there exists two paths p_4 and p_5 from u to d such that $p_4 \cap p_5 = \{b_{j+1}, b_{j+2}, \ldots b_m\}$.*

Proof Note that for each ℓ, there cannot exist a bridge between b_ℓ and $b_{\ell+1}$. Therefore, we can apply Lemma 32 to argue that there must exist two edge-disjoint paths between b_ℓ and $b_{\ell+1}$ For the same reason, we can find two edge-disjoint paths between b_m and d. To complete the proof, we simply concatenate these paths. ■

On the other hand, the paths p_1, p_2, and p_3 must satisfy a different set of constraints, which are formalized by the following lemma.

Lemma 34 *Let $b_j = (v, u)$ be a k-bridge, let p_2 be an arbitrarily chosen o-k path. There exists an o-k path p_1 and a k-v path p_3 such that: p_1 and p_3 are edge disjoint; and if p_1 visits k, then p_2 and p_1 use different incoming edges for k.*

Proof We show how p_1 and p_3 can be constructed. This splits into two cases, and we begin by considering the bridges b_j with $j > 1$. Due to our preprocessing from Lemma 27, b_j and b_{j-1} cannot be dependent. Note that every o-d path that uses b_{j-1} must also use b_j. Therefore, there must exist an o-d path p that uses b_j and not b_{j-1}. We fix p_1 to be the prefix of p up to the point where it visits b_j. Let p'_3 be an arbitrarily selected path from k to b_{j-1}. Note that p_1 cannot share an edge with p'_3, because otherwise p_1 would be forced to visit b_{j-1}.

We now show how p'_3 can be extended to reach b_j without intersecting p_1. Since there are no bridges between b_{j-1} and b_j, we can apply Lemma 32 to obtain two edge-disjoint paths q and q' from b_{j-1} to b_j. If one of these paths does not intersect with p_1, then we are done. Otherwise suppose, without loss of generality, that p_1 intersects with q before it intersects with q'. We create a path p'_1 that follows p_1 until the first intersection with q, and

follows q after that. Since q and q' are disjoint, the paths p'_1 and p'_3q' satisfy the required conditions.

Now we consider the bridge b_1. If k has at least two incoming edges, then we can apply Lemma 32 to find two edge disjoint paths from k to b_1, and we can easily construct p_1 and p_3 using these paths. Otherwise, let e be the sole incoming edge at k. Since e and b_1 are not dependent, we can find a path p_1 from o to b_1 which does not use e, and we can use the same technique as we did for $j > 1$ to find a path p_3 from k to b_1 that does not intersect with p_1. ■

Algorithm 6 FindKBridges(k)

Input: A vertex k, and a partial equivalent cost function f^a, such that $f^a_e(j)$ is defined for every $j \leq i$.

Output: A partial equivalent cost function f^{a+1}, such that f^{a+1} is an extension of f^a, and f^{a+1}_e is defined for every e that is a k bridge.

1: **for all** e and j for which $f^a_e(j)$ is defined **do**
2: $\quad f^{a+1}_e(j) \leftarrow f^a_e(j)$
3: **end for**
4: **for** $j = \mathrm{m}$ to 1 **do**
5: $\quad p_4, p_5 \leftarrow$ paths chosen according to Lemma 33
6: $\quad p_1, p_2, p_3 \leftarrow$ paths chosen according to Lemma 34
7: $\quad \mathsf{s} \leftarrow (1 \mapsto p_1 b_j p_4, i \mapsto p_2 p_3 b_j p_5)$
8: $\quad c_{\mathsf{s}} \leftarrow \mathrm{Query}(\mathsf{s})$
9: $\quad f^{a+1}_{b_j}(i+1) \leftarrow c_{\mathsf{s}}(p_1 b_j p_4) - \sum_{e \in p_1} f^{a+1}_e(n_e(\mathsf{s})) - \sum_{e \in p_4} f^{a+1}_e(n_e(\mathsf{s}))$
10: **end for**

Algorithm 6 shows how the cost of placing $i+1$ players on each of the k-bridges can be discovered. Note that on line 9, since s assigns one player to p_1, we have $n_e(\mathsf{s}) = 1$ for every $e \in p_1$. Therefore, $f^{a+1}_e(n_e(\mathsf{s}))$ is known for every edge $e \in p_1$. Moreover, for every edge $e \in p_4$, we have that $n_e(\mathsf{s}) = i+1$ if e is a k-bridge, and we have $n_e(\mathsf{s}) = 1$, otherwise. Since the algorithm processes the k-bridges in reverse order, we have that $f^{a+1}_e(n_e(\mathsf{s}))$ is defined for every edge $e \in p_4$. The following lemma shows that line 9 correctly computes the cost of b_j.

Lemma 35 *Let k be a vertex, and let f^a be a partial equivalent cost function, such that $f^a_e(j)$ is defined for every $j \leq i$. Algorithm 6 computes a partial equivalent cost function f^{a+1}, such that f^{a+1} is an extension of f^a, and f^{a+1}_e is defined for every e that is a k-bridge.*

Proof It can be verified that the algorithm constructs a partial cost function f^{a+1} that is an extension of f^a, where f^{a+1}_e is defined for every e that is a k-bridge. We must show that f^{a+1} is partially equivalent to f. Since f^a is partially equivalent to f, there exists some total cost function f' that is an extension of f^a, such that f' is equivalent to f. We will show that f' is also an extension of f^{a+1}.

We will do so inductively. The inductive hypothesis is that $f^{a+1}_e(i+1) = f'_e(i+1)$ for every $e = b_l$ with $\ell > j$. The base case, where $j = m$, is trivial, because there are no k-bridges b_l with $\ell > m$. Now suppose that we have shown the inductive hypothesis for

some j. We show that $f_{b_j}^{a+1}(i+1) = f'_{b_j}(i+1)$. Let s be the strategy queried when the algorithm considers b_j.

Consider an edge $e \in p_1$. By Lemma 34, we have that $n_e(\mathsf{s}) = 1$. By assumption, we have that $f_e^{a+1}(1) = f_e^a(1)$ for every edge e, and therefore $f_e^{a+1}(n_e(\mathsf{s})) = f'_e(n_e(\mathsf{s}))$ for every edge $e \in p_1$.

Now consider an edge $e \in p_4$. By Lemma 33, we have that $n_e(\mathsf{s}) = 1$ whenever e is not a k-bridge, and we have $n_e(\mathsf{s}) = i+1$ whenever e is a k-bridge. Therefore, by the inductive hypothesis, we have that $f_e^{a+1}(n_e(\mathsf{s})) = f'_e(n_e(\mathsf{s}))$ for every $e \in p_4$.

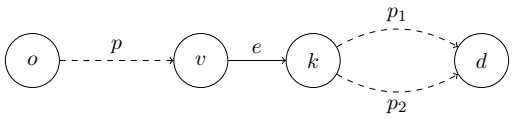
Since f' is equivalent to f, we have that $c_{\mathsf{s}}(p_1 b_l p_3) = \sum_{e \in p_1 b_l p_3} f'_e$. Therefore, line 9 sets:

$$\begin{aligned} f_{b_j}^{a+1}(i+1) &= c_{\mathsf{s}}(p_1 b_j p_4) - \sum_{e \in p_1} f_e^{a+1}(n_e(\mathsf{s})) - \sum_{e \in p_4} f_e^{a+1}(n_e(\mathsf{s})) \\ &= \sum_{e \in p_1 b_j p_4} f'_e(n_e(\mathsf{s})) - \sum_{e \in p_1} f'_e(n_e(\mathsf{s})) - \sum_{e \in p_4} f'_e(n_e(\mathsf{s})) \\ &= f'_{b_j}(n_e(\mathsf{s})) = f'_{b_j}(i+1). \end{aligned}$$

Thus, the algorithm correctly sets $f_{b_j}^{a+1}(i+1) = f'_{b_j}(i+1)$. ■

5.4.2 Incoming Edges of k

We now describe the second part of the many-player case. After finding the cost of each k-bridge, we find the cost of each incoming edge at k. The following diagram describes how we find the cost of $e = (v, k)$, an incoming edge at k .



The path p is an arbitrarily chosen path from o to v. The paths p_1 and p_2 are chosen according to the following lemma.

Lemma 36 *There exist two k-d paths p_1, p_2 such that every edge in $p_1 \cap p_2$ is a k-bridge.*

Proof Let b_1 be the first k-bridge. By Lemma 32 there exists edge disjoint paths from k to b_1. The proof can then be completed by applying Lemma 33. ■

Algorithm 7 shows how we find the cost of putting $i+1$ players on each edge e that is incoming at k. Apart from the consideration of k-bridges, this algorithm uses the same technique as Algorithm 4. Consider line 9. Note that every vertex in p is processed before k is processed, and therefore $f_{e'}^{a+1}(i+1)$ is known for every $e' \in p$. Moreover, for every edge $e' \in p_1$, we have that $n_{e'}(\mathsf{s}) = i+1$ if e' is a k-bridge, and we have $n_{e'}(\mathsf{s}) = 1$ otherwise. In either case, the $f_{e'}^{a+1}(n_{e'}(\mathsf{s}))$ is known for every edge $e' \in p_1$. The following lemma show that line 9 correctly computes $f_e^{a+1}(i+1)$.

Algorithm 7 MultiProcessK

Input: A vertex k, and a partial equivalent cost function f^a, such that $f^a_e(j)$ is defined for all $e \in E$ when $j \leq i$, all $e = (v, u)$ with $u \prec k$ when $j = i + 1$, and all k-bridges when $j = i + 1$.

Output: A partial equivalent cost function f^a, such that $f^a_e(j)$ is defined for all $e \in E$ when $j \leq i$, and for all $e = (v, u)$ with $u \preceq k$ when $j = i + 1$.

1: **for all** e and j for which $f^a_e(j)$ is defined **do**
2: $\quad f^{a+1}_e(j) \leftarrow f^a_e(j)$
3: **end for**
4: **for all** $e = (v, k) \in E$ **do**
5: $\quad p \leftarrow$ an arbitrary o-v path
6: $\quad p_1, p_2$ paths chosen according to Lemma 36
7: $\quad \mathsf{s} \leftarrow (1 \mapsto pep_1, i \mapsto pep_2)$
8: $\quad c_{\mathsf{s}} \leftarrow \text{Query}(\mathsf{s})$
9: $\quad f^{a+1}_e(i+1) \leftarrow c_{\mathsf{s}}(pep_1) - \sum_{e' \in p} f^{a+1}_{e'}(i+1) - \sum_{e' \in p_1} f^{a+1}_{e'}(n_{e'}(\mathsf{s}))$.
10: **end for**

Lemma 37 *Let k be a vertex, and let f^a be a partial equivalent cost function, such that $f^a_e(j)$ is defined for all $e \in E$ when $j \leq i$, all $e = (v, u)$ with $u \prec k$ when $j = i + 1$, and all k-bridges when $j = i + 1$. Algorithm 7 produces a partial equivalent cost function f^{a+1}, such that $f^{a+1}_e(j)$ is defined for all $e \in E$ when $j \leq i$, and for all $e = (v, u)$ with $u \preceq k$ when $j = i + 1$.*

Proof It can be verified that the algorithm constructs a partial cost function f^{a+1} that is defined for the correct parameters. We must show that f^{a+1} is partially equivalent to f. Note that f^{a+1} is an extension of f^a. Since f^a is partially equivalent to f, there exists some total cost function f' that is an extension of f^a, such that f' is equivalent to f. We will show that f' is also an extension of f^{a+1}.

Let $e = (v, k)$ be an incoming edge at k. We will show that $f^{a+1}_e(i+1) = f'_e(i+1)$. Let s be the strategy that the algorithm queries while processing e. Since f' is equivalent to f, we have that $c_{\mathsf{s}}(pep_1) = \sum_{e' \in pep_1} f'_{e'}(n_{e'}(\mathsf{s}))$. For every edge $e' \in p_1$, we have $n_{e'}(\mathsf{s}) = i + 1$. Since every vertex w visited by p satisfies $w \prec k$, for every $e' \in p_1$ we must have $f^{a+1}_{e'}(n_{e'}(\mathsf{s})) = f^a_{e'}(n_{e'}(\mathsf{s})) = f'_{e'}(n_{e'}(\mathsf{s}))$. For every edge $e' \in p_1$, we have $n_{e'}(\mathsf{s}) = 1$ if e' is not a k-bridge, and we have $n_{e'}(\mathsf{s}) = i + 1$ if e' is a k-bridge. In either case, we have that $f^{a+1}_{e'}(n_{e'}(\mathsf{s})) = f^a_{e'}(n_{e'}(\mathsf{s})) = f'_{e'}(n_{e'}(\mathsf{s}))$ for every edge $e' \in p_1$. Therefore, line 9 sets:

$$\begin{aligned} f^{a+1}_e(i+1) &= c_{\mathsf{s}}(pep_1) - \sum_{e' \in p} f^{a+1}_{e'}(n_{e'}(\mathsf{s})) - \sum_{e' \in p_1} f^{a+1}_{e'}(n_{e'}(\mathsf{s})) \\ &= \sum_{e \in pep_1} f'_{e'}(n_{e'}(\mathsf{s})) - \sum_{e' \in p} f'_{e'}(n_{e'}(\mathsf{s})) - \sum_{e' \in p_1} f'_{e'}(n_{e'}(\mathsf{s})) \\ &= f'_e(n_e(\mathsf{s})) = f'_e(i+1). \end{aligned}$$

Therefore, for each incoming edge $e = (v, k)$, we have that $f^{a+1}_e(i+1) = f'_e(i+1)$. Hence, f' is an extension of f^{a+1}, which implies that f^{a+1} is partially equivalent to f. ■

5.4.3 Query Complexity

We argue that the algorithm can be implemented so that the costs for $(i+1)$ players can be discovered using at most $|E|$ many payoff queries. Every time Algorithm 6 discovers the cost of placing $i+1$ players on a k-bridge, it makes exactly one payoff query. Every time Algorithm 7 discovers the cost of an incoming edge (v,k), it makes exactly one payoff query. The key observation is that the costs discovered by Algorithm 6 do not need to be rediscovered by Algorithm 7. That is, we can modify Algorithm 7 so that it ignores every incoming edge (v,k) that has already been processed by Algorithm 6. This modification ensures that the algorithm uses precisely $|E|$ payoff queries to discover the edge costs for $i+1$ players. This gives us the following theorem.

Theorem 38 *Let Γ be a symmetric network congestion game with n-players played on a DAG with $|E|$ edges. The payoff query complexity of finding a Nash equilibrium in Γ is at most $n \cdot |E|$.*

6. Conclusions and Further Work

We first consider open questions in the setting of payoff queries, which has been the main setting for the results in this paper. We then consider alternative query models.

6.0.1 Open Questions Concerning Payoff Queries

In the context of strategic-form games, there are a number of open problems. In Theorem 13, we show a super-linear lower bound on the payoff query complexity when ϵ is allowed to depend on k. Can we prove a super-linear lower bound for a constant ϵ? Is there a deterministic algorithm that can find an ϵ-Nash equilibrium with $\epsilon < \frac{1}{2}$ without querying the entire payoff matrices? Fearnley and Savani (2014) achieve $\epsilon < \frac{1}{2}$ with the use of randomization, but doing so with a deterministic algorithm appears to be challenging. Finally, when $2 \leq i \leq k-1$, we have shown that the payoff query complexity of finding a $(1-\frac{1}{i})$-Nash equilibrium lies somewhere in the range $[k-i+1, 2k-i+1]$. Determining the precise payoff query complexity for this case is an open problem.

For congestion games, our lower bound of $\log n + m$ arises from a game with two parallel links and a one-player game with m links. The upper bound of $\mathcal{O}\left(\log(n) \cdot \frac{\log^2(m)}{\log\log(m)} + m\right)$ is a poly-logarithmic factor off from this lower bound, with the factor depending on m. Can this factor be improved? It seems unlikely that the dependence of this factor on m can be completely removed, in which case, in order to provide tight bounds, a single lower bound construction that depends simultaneously on n and m would be necessary.

For symmetric network congestion games on DAGs it is unclear whether the payoff query complexity is sub-linear in n. Non-trivial lower and upper bounds for more general settings, such as asymmetric network congestion games (DAG or not) or general (non-network) congestion games would also be interesting.

6.0.2 Other Query Models

We have defined a payoff query as given by a *pure* (not mixed) profile s, since that is of main relevance to empirical game-theoretic modelling. Furthermore, if s was a mixed

profile, it could be simulated by sampling a number of pure profiles from s and making the corresponding sequence of pure payoff queries. An alternative definition might require a payoff query to just report a single specified player's payoff, but that would change the query complexity by a factor at most n.

Our main results have related to exact payoff queries, though other query models are interesting too. A very natural type of query is a *best-response query*, where a strategy s is chosen, and the algorithm is told the players' best responses to s. In general s may have to be a mixed strategy; it is not hard to check that pure-strategy best response queries are insufficient; even for a two-player two-action game, knowledge of the best responses to pure profiles is not sufficient to identify an ϵ-Nash equilibrium for $\epsilon < \frac{1}{2}$. Fictitious Play (Fudenberg and Levine 1998, Chapter 2) can be regarded as a query protocol that uses best-response queries (to mixed strategies) to find a Nash equilibrium in zero-sum games, and essentially a 1/2-Nash equilibrium in general-sum games (Goldberg et al., 2013). We can always synthesize a pure best-response query with $n(k-1)$ payoff queries. Hence, for questions of polynomial query complexity, payoff queries are at least as powerful as best-response queries. Are there games where best-response queries are much more useful than payoff queries? If k is large then it is expensive to synthesize best-response queries with payoff queries. The DMP-algorithm (Daskalakis et al., 2009b) finds a $\frac{1}{2}$-Nash equilibrium via only two best-response queries, whereas Theorem 5 notes that $\mathcal{O}(k)$ payoff queries are needed.

A *noisy* payoff query outputs an observation of a random variable taking values in $[0, 1]$ whose expected value is the true payoff. Alternative versions might assume that the observed payoff is within some distance ϵ from the true payoff. Noisy query models might be more realistic, and they are suggested by by the experimental papers on querying games. However in a theoretical context, one could obtain good approximations of the expected payoffs for a profile s, by repeated sampling. It would interesting to understand the power of different query models.

Acknowledgments

We would like to thank Michael Wellman for interesting discussions on this topic, and Milind Tambe for discussions on its relationship with adversarial security games. This work was supported by ESRC grant ESRC/BSB/09, and EPSRC grants EP/K01000X/1, EP/J019399/1, EP/H046623/1, and EP/L011018/1.

References

N. Alon, Y. Emek, M. Feldman, and M. Tennenholtz. Economical graph discovery. In *Proc. of ICS*, pages 476–486, 2011.

D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Y. Babichenko. Query complexity of approximate Nash equilibria. In *Proc. of STOC*, 2014.

Y. Babichenko and S. Barman. Query complexity of correlated equilibrium. *CoRR*, abs/1306.2437, 2013.

X. Bei, N. Chen, and S. Zhang. On the complexity of trial and error. In *Proc. of STOC*, pages 31–40, 2013.

P. Berenbrink, T.K. Friedetzky, L.A. Goldberg, P.W. Goldberg, and R. Martin. Distributed selfish load balancing. *SIAM Journal on Computing*, 37:1163–1181, 2007.

M. Brown, W.B. Haskell, and M. Tambe. Addressing scalability and robustness in security games with multiple boundedly rational adversaries. In *Proc. of GameSec*, 2014.

S. Chien and A. Sinclair. Convergence to approximate Nash equilibria in congestion games. *Games and Economic Behavior*, 71(2):315–327, 2011.

C. Daskalakis and C.H. Papadimitriou. On oblivious PTAS's for Nash equilibrium. In *Proc. of STOC*, pages 75–84, 2009.

C. Daskalakis, P.W. Goldberg, and C.H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, May 2009a.

C. Daskalakis, A. Mehta, and C.H. Papadimitriou. A note on approximate Nash equilibria. *Theoretical Computer Science*, 410(17):1581–1588, 2009b.

C. Daskalakis, R. Frongillo, C.H. Papadimitriou, G. Pierrakos, and G. Valiant. On learning algorithms for Nash equilibria. In *Proc. of SAGT*, pages 114–125, Oct 2010.

Q. Duong, Y. Vorobeychik, S. Singh, and M. Wellman. Learning graphical game models. In *Proc. of IJCAI*, pages 116–121, 2009.

E. Even-Dar, A. Kesselmann, and Y. Mansour. Convergence time to Nash equilibria. In *Proc. of ICALP*, pages 502–513, 2003.

A. Fabrikant, C.H. Papadimitriou, and K. Talwar. The complexity of pure Nash equilibria. In *Proc. of STOC*, pages 604–612, 2004.

J. Fearnley and R. Savani. Finding approximate Nash equilibria of bimatrix games via payoff queries. In *Proc. of EC*, pages 657–674, 2014.

J. Fearnley, P.W. Goldberg, R. Savani, and T.B. Sørensen. Approximate well-supported Nash equilibria below two-thirds. In *Proc. of SAGT*, pages 108–119, 2012.

J. Fearnley, M. Gairing, P.W. Goldberg, and R. Savani. Learning equilibria of games via payoff queries. In *Proc. of EC*, pages 397–414, 2013.

T. Feder, H. Nazerzadeh, and A. Saberi. Approximating Nash equilibria using small-support strategies. In *Proc. of EC*, pages 352–354, 2007.

R. Feldmann, M. Gairing, T. Lücking, B. Monien, and M. Rode. Nashification and the coordination ratio for a selfish routing game. In *Proc. of ICALP*, pages 514–526, 2003.

S. Fischer, H. Räcke, and B. Vöcking. Fast convergence to Wardrop equilibria by adaptive sampling methods. In *Proc. of STOC*, pages pp. 653–662, 2006.

D. Fudenberg and D.K. Levine. *The Theory of Learning in Games*. MIT Press, 1998.

M. Gairing and R. Savani. Computing stable outcomes in hedonic games. In *Proc. of SAGT*, pages 174–185, 2010.

M. Gairing and R. Savani. Computing stable outcomes in hedonic games with voting based deviations. In *Proc. of AAMAS*, pages 559–566, 2011.

M. Gairing, T. Lücking, M. Mavronicolas, B. Monien, and M. Rode. Nash equilibria in discrete routing games with convex latency functions. *Journal of Computer and System Sciences*, 74:1199–1225, 2008.

M. Gairing, T. Lücking, M. Mavronicolas, and B. Monien. Computing Nash equilibria for scheduling on restricted parallel links. *Theory Comput. Syst.*, 47(2):405–432, 2010.

P.W. Goldberg. Bounds for the convergence rate of randomized local search in a multiplayer, load-balancing game. In *Proc. of PODC*, pages 131–140, 2004.

P.W. Goldberg and C.H. Papadimitriou. Reducibility among equilibrium problems. In *Proc. of STOC*, pages 61–70, 2006.

P.W. Goldberg and A. Pastink. On the communication complexity of approximate Nash equilibria. In *Proc. of SAGT*, pages 192–203, 2012.

P.W. Goldberg and A. Roth. Bounds for the query complexity of approximate equilibria. In *Proc. of EC*, pages 639–656. ACM, June 2014.

P.W. Goldberg, R. Savani, T.B. Sørensen, and C. Ventre. On the approximation performance of fictitious play in finite games. *International Journal of Game Theory*, 42(4): 1059–1083, 2013.

S. Hart and Y. Mansour. How long to equilibrium? The communication complexity of uncoupled equilibrium procedures. *Games and Economic Behavior*, 69:107–126, 2010.

S. Hart and A. Mas-Colell. Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, 93(5):1830–1836, 2003.

S. Hart and A. Mas-Colell. Stochastic uncoupled dynamics and Nash equilibrium. *Games and Economic Behavior*, 57(2):286–303, 2006.

S. Hart and N. Nisan. The query complexity of correlated equilibria. In *Proc. of SAGT*, 2013.

P.R. Jordan, Y. Vorobeychik, and M.P. Wellman. Searching for approximate equilibria in empirical games. In *Proc. of AAMAS*, pages 1063–1070, 2008.

P.R. Jordan, L.J. Schvartzman, and M.P. Wellman. Strategy exploration in empirical games. In *Proc. of AAMAS*, pages 1131–1138, 2010.

M. Kearns, M. Littman, and S. Singh. Graphical models for game theory. In *Proc. of UAI*, pages 253–260, 2001.

S.C. Kontogiannis and P.G. Spirakis. Well supported approximate equilibria in bimatrix games. *Algorithmica*, 57(4):653–667, 2010.

T.H. Nguyen, R. Yang, A. Azaria, S. Kraus, and M. Tambe. Analyzing the effectiveness of adversary modeling in security games. In *Proc. of AAAI*, 2013.

E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proc. of AAMAS*, pages 880–887, 2004.

A. Sureka and P.R. Wurman. Using tabu best-response search to find pure strategy Nash equilibria in normal form games. In *Proc. of AAMAS*, pages 1023–1029, 2005.

Y. Vorobeychik, M.P. Wellman, and S. Singh. Learning payoff functions in infinite games. *Machine Learning*, 67:145–168, 2007.

M.P. Wellman. Methods for empirical game-theoretic analysis. In *Proc. of AAAI*, pages 1552–1555, 2006.

R. Yang, C. Kiekintveld, F. Ordóñez, M. Tambe, and R. John. Improving resource allocation strategies against human adversaries in security games: An extended study. *Artificial Intelligence*, 195:440–469, 2013.

Journal of Machine Learning Research 16 (2015) 1345-1390 Submitted 12/13; Revised 2/15; Published 8/15

Rationality, Optimism and Guarantees in General Reinforcement Learning

Peter Sunehag* SUNEHAG@GOOGLE.COM
Marcus Hutter MARCUS.HUTTER@ANU.EDU.AU
Research School of Computer Science (RSISE BLD 115)
The Australian National University, ACT 0200, Canberra Australia

Editor: Laurent Orseau

Abstract

In this article,[1] we present a top-down theoretical study of general reinforcement learning agents. We begin with rational agents with unlimited resources and then move to a setting where an agent can only maintain a limited number of hypotheses and optimizes plans over a horizon much shorter than what the agent designer actually wants. We axiomatize what is rational in such a setting in a manner that enables optimism, which is important to achieve systematic explorative behavior. Then, within the class of agents deemed rational, we achieve convergence and finite-error bounds. Such results are desirable since they imply that the agent learns well from its experiences, but the bounds do not directly guarantee good performance and can be achieved by agents doing things one should obviously not. Good performance cannot in fact be guaranteed for any agent in fully general settings. Our approach is to design agents that learn well from experience and act rationally. We introduce a framework for general reinforcement learning agents based on rationality axioms for a decision function and an hypothesis-generating function designed so as to achieve guarantees on the number errors. We will consistently use an optimistic decision function but the hypothesis-generating function needs to change depending on what is known/assumed. We investigate a number of natural situations having either a frequentist or Bayesian flavor, deterministic or stochastic environments and either finite or countable hypothesis class. Further, to achieve sufficiently good bounds as to hold promise for practical success we introduce a notion of a class of environments being generated by a set of laws. None of the above has previously been done for fully general reinforcement learning environments.

Keywords: reinforcement learning, rationality, optimism, optimality, error bounds

1. Introduction

A general reinforcement learning environment returns observations and rewards in cycles to an agent that feeds actions to the environment. An agent designer's aim is to construct an agent that accumulates as much reward as possible. Ideally, the agent should maximize a given quality measure like e.g., expected accumulated reward or the maximum accumulated reward that is guaranteed with a certain given probability. The probabilities and expectation should be the actual, i.e., with respect to the true environment. Performing this task

∗. The first author is now at Google - DeepMind, London UK

1. This article combines and extends our conference articles (Sunehag and Hutter, 2011, 2012a,b, 2013, 2014) and is further extended by (Sunehag and Hutter, 2015) covering stochastic laws.

well in an unknown environment is an extremely challenging problem (Hutter, 2005). Hutter (2005) advocated a Bayesian approach to this problem while we here introduce optimistic agents as an alternative.

The Bayesian approach to the above task is to design an agent that approximately maximizes the quality measure with respect to an a priori environment chosen by the designer. There are two immediate problems with this approach. The first problem is that the arbitrary choice of a priori environment, e.g., through a prior defining a mixture of a hypothesis class, substantially influences the outcome. The defined policy is optimal by definition in the sense of achieving the highest quality with respect to the a priori environment, but its quality with respect to other environments like the true one or a different mixture, might be much lower. The second problem is that computing the maximizing actions is typically too hard, even approximately. We will below explain how a recent line of work attempts to address these problems and see that the first problem is partially resolved by using information-theoretic principles to make a "universal" choice of prior, while the second is not resolved. Then we will discuss another way in which Bayesian methods are motivated which is through rational choice theory (Savage, 1954).

The optimistic agents that we introduce in this article have the advantage that they satisfy guarantees that hold regardless of which environment from a given class is the true one. We introduce the concept of a class being generated by a set of laws and improve our bounds from being linear in the number of environments to linear in the number of laws. Since the number of environments can be exponentially larger than the number of laws this is of vital importance and practically useful environment classes should be such that its size is exponential in the number of laws. We will discuss such guarantees below as well as the mild modification of the classical rationality framework required to deem an optimistic agent rational. We also explain why such a modification makes sense when the choice to be made by an agent is one in a long sequence of such choices in an unknown environment.

Information-theoretic priors and limited horizons. Hutter (2005) and Veness et al. (2011) choose the prior, which can never be fully objective (Leike and Hutter, 2015), through an information-theoretic approach based on the code length of an environment by letting environments with shorter implementations be more likely. Hutter (2005) does this for the universal though impractical class of all lower semi-computable environments while Veness et al. (2011) use a limited but useful class based on context trees. For the latter, the context tree weighting (CTW) algorithm (Willems et al., 1995) allows for efficient calculation of the posterior. However, to optimize even approximately the quality measure used to evaluate the algorithm for the actual time-horizon (e.g., a million time steps), is impossible in complex domains. The MC-AIXI-CTW agent in Veness et al. (2011), which we employ to illustrate the point, uses a Monte-Carlo tree search method to optimize a geometrically discounted objective. Given a discount factor close to 1 (e.g., 0.99999) the effective horizon becomes large (100000). However, the tree search is only played out until the end of episode in the tasks considered in Veness et al. (2011). Playing it out for 100000 time steps for each simulation at each time step would be completely infeasible. When an agent maximizes the return from a much shorter horizon than the actual, e.g., one game instead of a 1000 games of PacMan, the exploration versus exploitation dilemma shows up. If the environment is fully known, then maximizing the return for one episode is perfect. In an unknown environment such a strategy can be a fatal mistake. If the expected return

is maximized for a shorter working horizon, i.e., the agent always exploits, then it is likely to keep a severely sub-optimal policy due to insufficient exploration. Veness et al. (2011) addressed this heuristically through random exploration moves.

Our agent framework. In Section 3, we introduce a framework that combines notions of what is considered desirable in decision theory with optimality concepts from reinforcement learning. In this framework, an agent is defined by the choice of a decision function and a hypothesis-generating function. The hypothesis-generating function feeds the decision function a finite class of environments at every time step and the decision function chooses an action/policy given such a class. The decision-theoretic analysis of rationality is used to restrict the choice of the decision function, while we consider guarantees for asymptotic properties and error bounds when designing the hypothesis-generating function.

All the agents we study can be expressed with an optimistic decision function but we study many different hypothesis-generating functions which are suitable under different assumptions. For example, with a domination assumption there is no need to remove environments, it would only worsen the guarantees. Hence a constant hypothesis-generating function is used. If we know that the environment is in a certain finite class of deterministic environments, then a hypothesis-generating function that removes contradicted environments but does not add any is appropriate. Similarly, when we have a finite class of stochastic but non-dominant environments that we assume the truth belongs to, the hypothesis-generating function should not add to the class but needs to exclude those environments that have become implausible.

If we only know that the true environment is in a countable class and we choose an optimistic decision function, the agent needs to have a growing finite class. In the countable case, a Bayesian agent can still work with the whole countable class at once (Lattimore, 2014), though to satisfy the desired guarantees that agent (BayesExp) was adjusted in a manner we here deem irrational. Another alternative adjustment of a Bayesian agent that is closer to fitting our framework is the Best of Sampled Set (BOSS) algorithm (Asmuth et al., 2009). This agent samples a finite set of environments (i.e., hypothesis-generation) from the posterior and then constructs an optimistic environment by combining transition dynamics from all those environments in the most optimistic manner and then optimize for this new environment (optimistic decision). This is an example of an agent that uses what we refer to as environments constructed by combining laws, though BOSS belongs in the narrow Markov Decision Process setting, while we here aim for full generality.

Rationality. In the foundations of decision theory, the focus is on axioms for rational preferences (Neumann and Morgenstern, 1944; Savage, 1954) and on making a single decision that does not affect the event in question but only its utility. The single decision setting can actually be understood as incorporating sequential decision-making since the one choice can be for a policy to follow for a period of time. This latter perspective is called normal form in game theory. We extend rational choice theory to the full reinforcement learning problem. It follows from the strictest version of the axioms we present that the agent must be a Bayesian agent. These axioms are appropriate when an agent is capable of optimizing the plan for its entire life. Then we loosen the axioms in a way that is analogous to the multiple-prior setting by Gilboa and Schmeidler (1989), except that ours enable optimism instead of pessimism and are based on a given utility function. These more permissive

axioms are suitable for a setting where the agent must actually make the decisions in a sequence due to not being able to optimize over the full horizon. We prove that optimism allows for asymptotic optimality guarantees and finite error bounds not enjoyed by a realist (expected utility maximizer).

Guarantees. In the field of reinforcement learning, there has been much work dedicated to designing agents for which one can prove asymptotic optimality or sample complexity bounds. The latter are high probability bounds on the number of time steps where the agent does not make a near optimal decision (Strehl et al., 2009). However, a weakness with sample complexity bounds is that they do not directly guarantee good performance for the agent. For example, an agent who has the opportunity to self-destruct can achieve subsequent optimality by choosing this option. Hence, aiming only for the best sample complexity can be a very bad idea in general reinforcement learning. If the environment is an ergodic MDPs or value-stable environment (Ryabko and Hutter, 2008) where the agent can always recover, these bounds are more directly meaningful. However, optimizing them blindly is still not necessarily good. Methods that during explicit exploration phases, aim at minimizing uncertainty by exploring the relatively unknown, can make very bad decisions. If one has an option offering return in the interval $[0, 0.3]$ and another option has return in the interval $[0.7, 0.8]$ one should have no interest in the first option since its best case scenario is worse than the worst case scenario of the other option. Nevertheless, some devised algorithms have phases of pure exploration where the most uncertain option is chosen. On the other hand, we will argue that one can rationally choose an option with return known to be in $[0.2, 0.85]$ over either. Assuming uniform belief over those intervals, the latter option is, however, not strictly rational under the classical axioms that are equivalent to choosing according to maximum subjective expected utility. We will sometimes use the term weakly rational for the less strict version of rationality considered below.

Here we consider agents that are rational in a certain decision-theoretic sense and within this class we design agents that make few errors. Examples of irrational agents, as discussed above, are agents that rely on explicit phases of pure exploration that aim directly at excluding environments while a category of prominent agents instead rely on optimism (Szita and Lörincz, 2008; Strehl et al., 2009; Lattimore and Hutter, 2012). Optimistic agents investigate whether a policy is as good as the hypothesis class says it might be but not whether something is bad or very bad. We extend these kinds of agents from MDP to general reinforcement learning and we deem them rational according to axioms presented here in Section 2.

The bounds presented here, like discussed above, are of a sort that the agent is guaranteed to eventually act nearly as well as possible given the history that has been generated. Since the risk of having all prospects destroyed cannot be avoided in the fully general setting, we have above argued that the bounds should be complemented with a demand for acting rationally. This does of course not prevent disaster, since nothing can. Hutter (2005) brings up a heaven and hell example where either action a_1 takes the agent to hell (min reward forever) and a_2 to heaven (max reward forever) or the other way around with a_2 to hell and a_1 to heaven. If one assumes that the true environment is safe (Ryabko and Hutter, 2008) as in always having the same optimal value from all histories that can occur, this kind of bounds are directly meaningful. Otherwise, one can consider an agent that is first pessimistic and rules out all actions that would lead to disaster for some environment

in its class and then takes an optimistic decision among the remaining actions. The bounds then apply to the environment class that remains after the pessimist has ruled out some actions. The resulting environments might not have as good prospects anymore due to the best action being ruled out, and in the heaven and hell example both actions would be ruled out and one would have to consider both. However, we repeat: there are no agents that can guarantee good outcomes in general reinforcement learning (Hutter, 2005).

The bounds given in Section 5 have a linear dependence on the number of environments in the class. While this rate is easily seen to be the best one can do in general (Lattimore et al., 2013a), it is exponentially worse than what we are used to from Markov Decision Processes (MDPs) (Lattimore and Hutter, 2012) where the linear (up to logarithms) dependence is on the size of the state space instead. In Section 5.2 we introduce the concept of laws and environments generated by sets of laws and we achieve bounds that are linear in the number of laws instead of the number of environments. All environment classes are trivially generated by sets of laws that equal the environments but some can also be represented as generated by exponentially fewer laws than there are environments. Such environment classes have key elements in common with an approach that has been heuristically developed for a long time, namely collaborative multi-agent systems called Learning Classifier Systems (LCS) (Holland, 1986; Hutter, 1991; Drugowitsch, 2007) or artificial economies (Baum and Durdanovic, 2001; Kwee et al., 2001). Such systems combine sub-agents that make recommendations and predictions in limited contexts (localization), sometimes combined with other sub-agents' predictions for the same single decision (factorization). The LCS family of approaches are primarily model-free by predicting the return and not future observations while what we introduce here is model-based and has a dual interpretation as an optimistic agent, which allows for theoretical guarantees.

Related work. Besides the work mentioned above, which all use discounted reward sums, Maillard et al. (2011); Nguyen et al. (2013); Maillard et al. (2013) extend the UCRL algorithm and regret bounds (Auer and Ortner, 2006) from undiscounted MDPs to problems where the environments are defined by combining maps from histories to states with MDP parameters as in Hutter (2009b); Sunehag and Hutter (2010). Though Maillard et al. (2011, 2013) study finite classes, Nguyen et al. (2013) extend their results by incrementally adding maps. Their algorithms use undiscounted reward sums and are, therefore, in theory not focused on a shorter horizon but on average reward over an infinite horizon. However, to optimize performance over long horizons is practically impossible in general. The online MDP with bandit feedback work (Neu et al., 2010; Abbasi-Yadkori et al., 2013) aims at general environments but limited to finitely many policies called experts to choose between. We instead limit the environment class in size, but consider any policies.

Outline. We start below with notation and background for general reinforcement learning and then in Section 2 we introduce the axioms for rational and rational optimistic agents. In Section 3 we introduce an agent framework that fits all the agents studied in this article and we make the philosophy fully explicit. It consists of two main parts, rational decision functions (Section 3.1) and hypothesis-generating functions (Section 3.2) that given a history delivers a class of environments to the decision function. In Section 4 we show the importance of optimism for asymptotic optimality for a generic Bayesian reinforcement learning agent called AIXI and we extend this agent to an optimistic multiple-prior agent

with stronger asymptotic guarantees. The required assumption is the a priori environments' dominance over the true environment and that at least one a priori environment is optimistic for the true environment.

In Section 5 and Section 6 we continue to study optimistic agents that pick an optimistic hypothesis instead of an optimistic a priori distribution. This is actually the very same mathematical formula for how to optimistically make a decision given a hypothesis class. However, in this case we do not assume that the environments in the class dominate the truth and the agent, therefore, needs to exclude environments which are not aligned with observations received. Instead of assuming dominance as in the previous section, we here assume that the truth is a member of the class. It is interesting to notice that the only difference between the two sections, despite their very different interpretations, is the assumptions used for the mathematical analysis. In Section 5.2 we also show that understanding environment classes as being generated by finite sets of partial environments that we call laws, allows for error bounds that are linear in the number of laws instead of in the number of environments. This can be an exponential improvement.

In earlier sections the hypothesis-generating functions either deliver the exact same class (except for conditioning the environments on the past) at all times or just remove implausible environments from an initial class while in Section 7 we consider hypothesis-generating functions that also add new environments and exhaust a countable class in the limit. We prove error bounds that depend on how fast new environments are introduced. Section 8 contains the conclusions. The appendix contains extensions of various results.

We summarize our contributions and where they can be found in the following list:

- Axiomatic treatment of rationality and optimism: Section 2.
- Agent framework: Section 3
- Asymptotic results for AIXI (rational) and optimistic agents using finite classes of dominant stochastic environments: Section 4
- Asymptotic and finite error bounds for optimistic agents with finite classes of deterministic (non-dominant) environments containing the truth, as well as improved error rates for environment classes based on laws: Section 5
- Asymptotic results for optimistic agents with finite classes of stochastic non-dominant environments containing the truth: Section 6
- Extensions to countable classes: Section 7.
- Extending deterministic results from smaller class of conservative optimistic agents to larger class of liberal optimistic agents: Appendix A
- Extending axioms for rationality to countable case: Appendix B
- A list of important notation can be found in Appendix C

General reinforcement learning: notation and background. We will consider an agent (Russell and Norvig, 2010; Hutter, 2005) that interacts with an environment through performing actions a_t from a finite set $\mathcal{A}$ and receives observations o_t from a finite set $\mathcal{O}$ and rewards r_t from a finite set $\mathcal{R} \subset [0,1]$ resulting in a history $h_t := a_0 o_1 r_1 a_1, ..., o_t r_t$. These sets can be allowed to depend on time or context but we do not write this out explicitly. Let $\mathcal{H} := \epsilon \cup (\mathcal{A} \times \cup_n (\mathcal{O} \times R \times \mathcal{A})^n \times (\mathcal{O} \times \mathcal{R}))$ be the set of histories where ϵ is the empty history and $\mathcal{A} \times (\mathcal{O} \times R \times \mathcal{A})^0 \times (\mathcal{O} \times \mathcal{R}) := \mathcal{A} \times \mathcal{O} \times \mathcal{R}$. A function $\nu : \mathcal{H} \times \mathcal{A} \to \mathcal{O} \times \mathcal{R}$ is called a deterministic environment. A function $\pi : \mathcal{H} \to \mathcal{A}$ is called a (deterministic) policy or an agent. We define the value function V based on geometric discounting by $V_\nu^\pi(h_{t-1}) = \sum_{i=t}^\infty \gamma^{i-t} r_i$ where the sequence r_i are the rewards achieved by following π from time step t onwards in the environment ν after having seen h_{t-1}.

Instead of viewing the environment as a function $\mathcal{H} \times \mathcal{A} \to \mathcal{O} \times \mathcal{R}$ we can equivalently write it as a function $\mathcal{H} \times \mathcal{A} \times \mathcal{O} \times \mathcal{R} \to \{0,1\}$ where we write $\nu(o, r|h, a)$ for the function value. It equals zero if in the first formulation (h, a) is not sent to (o, r) and 1 if it is. In the case of stochastic environments we instead have a function $\nu : \mathcal{H} \times \mathcal{A} \times \mathcal{O} \times \mathcal{R} \to [0,1]$ such that $\sum_{o,r} \nu(o, r|h, a) = 1 \ \forall h, a$. The deterministic environments are then just a degenerate special case. Furthermore, we define $\nu(h_t|\pi) := \Pi_{i=1}^t \nu(o_i r_i|a_i, h_{i-1})$ where $a_i = \pi(h_{i-1})$. $\nu(\cdot|\pi)$ is a probability measure over strings, actually one measure for each string length with the corresponding power set as the σ-algebra. We define $\nu(\cdot|\pi, h_{t-1})$ by conditioning $\nu(\cdot|\pi)$ on h_{t-1} and we let $V_\nu^\pi(h_{t-1}) := \mathbb{E}_{\nu(\cdot|\pi,h_{t-1})} \sum_{i=t}^\infty \gamma^{i-t} r_i$ and $V_\nu^*(h_{t-1}) := \max_\pi V_\nu^\pi(h_{t-1})$.

Examples of agents: AIXI and Optimist. Suppose we are given a countable class of environments $\mathcal{M}$ and strictly positive prior weights w_ν for all $\nu \in \mathcal{M}$. We define the a priori environment ξ by letting $\xi(\cdot) = \sum w_\nu \nu(\cdot)$ and the AIXI agent is defined by following the policy

$$\pi^* := \arg\max_\pi V_\xi^\pi(\epsilon) \tag{1}$$

which is its general form. Sometimes AIXI refers to the case of a certain universal class and a Solomonoff style prior (Hutter, 2005). The above agent, and only agents of that form, satisfies the strict rationality axioms presented first in Section 2 while the slightly looser version we present afterwards enables optimism. The optimist chooses its next action after history h based on

$$\pi^\circ := \arg\max_\pi \max_{\xi \in \Xi} V_\xi^\pi(h) \tag{2}$$

for a set of environments (beliefs) Ξ which we in the rest of the article will assume to be finite, though results can be extended further.

2. Rationality in Sequential Decision-Making

In this section, we first derive the above introduced AIXI agent from rationality axioms inspired by the traditional literature (Neumann and Morgenstern, 1944; Ramsey, 1931; Savage, 1954; deFinetti, 1937) on decision-making under uncertainty. Then we suggest weakening a symmetry condition between accepting and rejecting bets. The weaker condition only says that if an agent considers one side of a bet to be rejectable, it must be prepared to accept the other side but it can accept either. Since the conditions are meant for sequential decision and one does not accept several bets at a time, considering both sides of a bet to

be acceptable is not necessarily vulnerable to combinations of bets that would otherwise cause our agent a sure loss. Further, if an outcome is only revealed when a bet is accepted, one can only learn about the world by accepting bets. What is learned early on can lead to higher earnings later. The principle of optimism results in a more explorative agent and leads to multiple-prior models or the imprecise probability by Walley (2000). Axiomatics of multiple-prior models has been studied by Gilboa and Schmeidler (1989); Casadesus-Masanell et al. (2000). These models can be understood as quantifying the uncertainty in estimated probabilities by assigning a whole set of probabilities. In the passive prediction case, one typically combines the multiple-prior model with caution to achieve more risk averse decisions (Casadesus-Masanell et al., 2000). In the active case, agents need to take risk to generate experience that they can learn successful behavior from and, therefore, optimism is useful.

Bets. The basic setting we use is inspired by the betting approach of Ramsey (1931); deFinetti (1937). In this setting, the agent is about to observe a symbol from a finite alphabet and is offered a bet $x = (x_1, ..., x_n)$ where $x_i \in \mathbb{R}$ is the reward received for the outcome i.

Definition 1 (Bet) *Suppose that we have an unknown symbol from an alphabet with m elements, say $\{1, ..., m\}$. A bet (or contract) is a vector $x = (x_1, ..., x_m)$ in $\mathbb{R}^m$ where x_j is the reward received if the symbol is j.*

In our definition of decision maker we allow for choosing neither accept nor reject, while when we move on to axiomatize rational decision makers we will no longer allow for neither. In the case of a strictly rational decision maker it will only be the zero bet that can, and actually must, be both acceptable and rejectable. For the rational optimist the zero bet is always accepted and all bets are exactly one of acceptable or rejectable.

Definition 2 (Decision maker, Decision) *A decision maker (for bets regarding an unknown symbol) is a pair of sets $(Z, \tilde{Z}) \subset \mathbb{R}^m \times \mathbb{R}^m$ which defines exactly the bets that are acceptable (Z) and those that are rejectable ($\tilde{Z}$). In other words, a decision maker is a function from $\mathbb{R}^m$ to $\{$accepted,rejected,either,neither$\}$. The function value is called the decision.*

Next we present the stricter version of the axioms and a representation theorem.

Definition 3 (Strict rationality) *We say that $(Z, \tilde{Z})$ is strictly rational if it has the following properties:*

1. **Completeness**: $Z \cup \tilde{Z} = \mathbb{R}^m$
2. **Symmetry**: $x \in Z \iff -x \in \tilde{Z}$
3. **Convexity of accepting**: $x, y \in Z,\ \lambda, \gamma > 0 \Rightarrow \lambda x + \gamma y \in Z$
4. **Accepting sure profits**: $\forall k\ x_k > 0\ \Rightarrow x \in Z \setminus \tilde{Z}$

Axiom 1 in Definition 3 is really describing the setting rather than an assumption. It says that the agent must always choose at least one of accept or reject. Axiom 2 is a symmetry condition between accepting and rejecting that we will replace in the optimistic setting. In the optimistic setting we still demand that if the agent rejects x, then it must accept $-x$ but not the other way around. Axiom 3 is motivated as follows: If $x \in Z$ and $\lambda \geq 0$, then $\lambda x \in Z$ since it is simply a multiple of the same bet. Also, the sum of two acceptable bets should be acceptable. Axiom 4 says that if the agent is guaranteed to win money it must accept the bet and cannot reject it.

The following representation theorem says that a strictly rational decision maker can be represented as choosing bets to accept based on if they have positive expected utility for some probability vector. The same probabilities are consistently used for all decisions. Hence, the decision maker can be understood as a Bayesian agent with an a priori environment distribution. In Sunehag and Hutter (2011) we derived Bayes rule by showing how the concepts of marginal and conditional probabilities also come out of the same rational decision-making framework.

Theorem 4 (Existence of probabilities, Sunehag&Hutter 2011) *Given a rational decision maker, there are numbers $p_i \geq 0$ that satisfy*

$$\{x \mid \sum x_i p_i > 0\} \subseteq Z \subseteq \{x \mid \sum x_i p_i \geq 0\}. \tag{3}$$

Assuming $\sum_i p_i = 1$ makes the numbers unique probabilities and we will use the notation $Pr(i) = p_i$.

Proof The third property tells us that Z and $-Z$ ($= \tilde{Z}$ according to the second property) are convex cones. The second and fourth property tells us that $Z \neq \mathbb{R}^m$. Suppose that there is a point x that lies in both the interior of Z and of $-Z$. Then, the same is true for $-x$ according to the second property and for the origin according to the third property. That a ball around the origin lies in Z means that $Z = \mathbb{R}^m$ which is not true. Thus the interiors of Z and $-Z$ are disjoint open convex sets and can, therefore, according to the Hahn-Banach Theorem be separated by a hyperplane which goes through the origin since according to the first and second property the origin is both acceptable and rejectable. The first two properties tell us that $Z \cup -Z = \mathbb{R}^m$. Given a separating hyperplane between the interiors of Z and $-Z$, Z must contain everything on one side. This means that Z is a half space whose boundary is a hyperplane that goes through the origin and the closure $\bar{Z}$ of Z is a closed half space and can be written as

$$\bar{Z} = \{x \mid \sum x_i p_i \geq 0\}$$

for some vector $p = (p_i)$ such that not every p_i is 0. The fourth property tells us that $p_i \geq 0 \ \forall i$. ■

In Appendix B we extend the above results to the countable case with Banach sequence spaces as the spaces of bets. Sunehag and Hutter (2011) showed how one can derive basic probability-theoretic concepts like marginalization and conditionalization from rationality.

Rational optimism. We now present four axioms for rational optimism. They state properties that the set of accepted and the set of rejected bets must satisfy.

Definition 5 (Rational optimism, Weak rationality) *We say that the decision maker $(Z, \tilde{Z}) \subset \mathbb{R}^m \times \mathbb{R}^m$ is a rational optimist or weakly rational if it satisfies the following:*

1. **Disjoint Completeness***: $x \notin \tilde{Z} \iff x \in Z$*
2. **Optimism***: $x \in \tilde{Z} \Rightarrow -x \notin \tilde{Z}$*
3. **Convexity of rejecting***: $x, y \in \tilde{Z}$ and $\lambda, \gamma > 0 \Rightarrow \lambda x + \gamma y \in \tilde{Z}$*
4. **Rejecting sure losses***: $\forall k \; x_k < 0 \quad \Rightarrow x \in \tilde{Z} \setminus Z$*

The first axiom is again a completeness axiom where we here demand that each contract is either accepted or rejected but not both. We introduce this stronger disjoint completeness assumption since the other axioms now concern the set of rejected bets, while we want to conclude something about what is accepted. The following three axioms concern rational rejection. The second says that if x is rejected then $-x$ must not be rejected. Hence, if the agent rejects one side of a bet it must, due to the first property, accept its negation. This was also argued for in the first set of axioms in the previous setting but in the optimistic set we do not have the opposite direction. In other words, if x is accepted then $-x$ can also be accepted. The agent is strictly rational about how it rejects bets. Rational rejection also means that if the agent rejects two bets x and y, it also rejects $\lambda x + \gamma y$ if $\lambda \geq 0$ and $\gamma \geq 0$. The final axiom says that if the reward is guaranteed to be strictly negative the bet must be rejected.

The representation theorem for rational optimism differs from that of strict rationality by not having a single unique environment distribution. Instead the agent has a set of such and if the bet has positive expected utility for any of them, the bet is accepted.

Theorem 6 (Existence of a set of probabilities) *Given a rational optimist, there is a set $\mathcal{P} \subset \mathbb{R}^m$ that satisfies*

$$\{x \mid \exists p \in \mathcal{P} : \sum x_i p_i > 0\} \subseteq Z \subseteq \{x \mid \exists p \in \mathcal{P} : \sum x_i p_i \geq 0\}. \tag{4}$$

One can always replace $\mathcal{P}$ with an extreme set the size of the alphabet. Also, one can demand that all the vectors in $\mathcal{P}$ be probability vectors, i.e., $\sum p_i = 1$ and $\forall i \; p_i \geq 0$.

Proof Properties 2 and 3 tell us that the closure $\bar{\tilde{Z}}$ of $\tilde{Z}$ is a (one sided) convex cone. Let $\mathcal{P} = \{p \in \mathbb{R}^m \mid \sum p_i x_i \leq 0 \; \forall (x_i) \in \bar{\tilde{Z}}\}$. Then, it follows from convexity that $\bar{\tilde{Z}} = \{(x_i) \mid \sum x_i p_i \leq 0 \; \forall p \in \mathcal{P}\}$. Property 4 tells us that it contains all the elements of only strictly negative coefficients and this implies that for all $p \in \mathcal{P}$, $p_i \geq 0$ for all i. It follows from property 1 and the above that $\{x \mid \sum x_i p_i > 0\} \subseteq Z$ for all $p \in \mathcal{P}$. Normalizing all $p \in \mathcal{P}$ such that $\sum p_i = 1$ does not change anything. Property 1 tells us that $Z \subseteq \{x \mid \exists p \in \mathcal{P} : \sum x_i p_i \geq 0\}$. ■

Remark 7 (Pessimism) *If one wants an axiomatic system for rational pessimism, one can reverse the roles of Z and $\tilde{Z}$ in the definition of rational optimism and the theorem applies with a similar reversal: The conclusion could be rewritten by replacing $\exists$ with $\forall$ in the conclusion of Theorem 6.*

Making choices. To go from agents making decisions on accepting or rejecting bets to agents choosing between different bets $x^j,\ j = 1, 2, 3, ...$, we define preferences by saying that x is better than or equal to y if $x - y \in \bar{Z}$ (the closure of Z), while it is worse or equal if $x - y$ is rejectable. For the first form of rationality stated in Definition 3, the consequence is that the agent chooses the option with the highest expected utility. If we instead consider optimistic rationality, and if there is $p \in \mathcal{P}$ such that $\sum x_i p_i \geq \sum y_i q_i\ \forall q \in \mathcal{P}$ then $\sum p_i(x_i - y_i) \geq 0$ and, therefore, $x - y \in \bar{Z}$. Therefore, if the agent chooses the bet $x^j = (x_i^j)_i$ by

$$\arg\max_j \max_{p \in \mathcal{P}} \sum x_i^j p_i$$

it is guaranteed that this bet is preferable to all other bets. We call this the optimistic decision or the rational optimistic decision. If the environment is reactive, i.e., if the probabilities for the outcome depends on the action, then p_i is above replaced by p_i^j. We discussed this in more detail in Sunehag and Hutter (2011).

Rational sequential decisions. For the general reinforcement learning setting we consider the choice of policy to use for the next T time steps. After one chooses a policy to use for those T steps the result is a history h_T and the value/return $\sum_{t=1}^{T} r_t \gamma^t$. There are finitely many possible h_T, each of them containing a specific return. If we enumerate all the possible h_T using i and the possible policies by j then for each policy and history there is a probability p_i^j for that history to be the result when policy j is used. Further we will denote the return achieved in history i by x_i. The bet x_i does depend on j since the rewards are part of the history.

By considering the choice to be for a policy π (previously j), an extension to finitely many sequential decisions is directly achieved. The discounted value $\sum r_t \gamma^t$ achieved then plays the role of the bet x_i and the decision on what policy to follow is taken according to

$$\pi^* \in \arg\max_\pi V_\xi^\pi$$

where ξ is the probabilistic a priori belief (the p_i^j) and $V_\xi^\pi = \sum p_i^j (\sum r_t^i \gamma^t)$ where r_t^i is the reward achieved at time t in outcome sequence i in an enumeration of all the possible histories. The rational optimist chooses the next action based on a policy

$$\pi^\circ \in \arg\max_\pi \max_{\xi \in \Xi} V_\xi^\pi$$

for a finite set of environments Ξ ($\mathcal{P}$ before) and recalculates this at every time step.

3. Our Agent Framework

In this section, we introduce an agent framework that all agents we study in this paper can be fitted into by a choice of what we call a decision function and a hypothesis-generating function.

3.1 Decision Functions

The primary component of our agent framework is a decision function $f : \mathbb{M} \to \mathcal{A}$ where $\mathbb{M}$ is the class of all finite sets $\mathcal{M}$ of environments. The function value only depends on

the class of environments $\mathcal{M}$ that is the argument. The decision function is independent of the history, however, the class $\mathcal{M}$ fed to the decision function introduces an indirect dependence. For example, the environments at time $t+1$ can be the environments at time t, conditioned on the new observation. Therefore, we will in this section often write the value function without an argument: $V_{\nu_t}^{\tilde{\pi}} = V_{\nu_0}^{\pi}(h_t)$ if $\nu_t = \nu_0(\cdot|h_t)$ where the policy $\tilde{\pi}$ on the left hand side is the same as the policy π on the right, just after h_t have been seen. It starts at a later stage, meaning $\tilde{\pi}(h) = \pi(h_t h)$, where $h_t h$ is a concatenation.

Definition 8 (Rational decision function) *Given alphabets $\mathcal{A}$, $\mathcal{O}$ and $\mathcal{R}$ we say that a decision function $f : \mathbb{M} \to \mathcal{A}$ is a function $f(\mathcal{M}) = a$ that for any class of environments $\mathcal{M}$ based on those alphabets produces an action $a \in \mathcal{A}$. We say that f is strictly rational for the class $\mathcal{M}$ if there are $\omega_\nu \geq 0,\ \nu \in \mathcal{M}, \sum_{\nu\in\mathcal{M}} w_\nu = 1$ and there is a policy*

$$\pi \in \arg\max_{\tilde{\pi}} \sum_{\nu\in\mathcal{M}} \omega_\nu V_\nu^{\tilde{\pi}} \tag{5}$$

such that $a = \pi(\epsilon)$.

Agents as in Definition 8 are also called admissible if $w_\nu > 0\ \forall \nu \in \mathcal{M}$ since then they are Pareto optimal (Hutter, 2005). Being Pareto optimal means that if another agent (of this form or not) is strictly better (higher expected value) than a particular agent of this form in one environment, then it is strictly worse in another. A special case is when $|\mathcal{M}| = 1$ and (5) becomes

$$\pi \in \arg\max_{\tilde{\pi}} V_\nu^{\tilde{\pi}}$$

where ν is the environment in $\mathcal{M}$. The more general case connects to this by letting $\tilde{\nu}(\cdot) := \sum_{\nu\in\mathcal{M}} w_\nu \nu(\cdot)$ since then $V_{\tilde{\nu}}^{\pi} = \sum w_\nu V_\nu^{\pi}$ (Hutter, 2005). The next definition defines optimistic decision functions. They only coincide with strictly rational ones for the case $|\mathcal{M}| = 1$, however agents based on such decision functions satisfy the looser axioms that define a weaker form of rationality as presented in Section 2.

Definition 9 (Optimistic decision function) *We call a decision function f optimistic if $f(\mathcal{M}) = a$ implies that $a = \pi(\epsilon)$ for an optimistic policy π, i.e., for*

$$\pi \in \arg\max_{\tilde{\pi}} \max_{\nu\in\mathcal{M}} V_\nu^{\tilde{\pi}}. \tag{6}$$

3.2 Hypothesis-Generating Functions

Given a decision function, what remains to create a complete agent is a hypothesis-generating function $\mathcal{G}(h) = \mathcal{M}$ that for any history $h \in \mathcal{H}$ produces a set of environments $\mathcal{M}$. A special form of hypothesis-generating function is defined by combining the initial class $\mathcal{G}(\epsilon) = \mathcal{M}_0$ with an update function $\psi(\mathcal{M}_{t-1}, h_t) = \mathcal{M}_t$. An agent is defined from a hypothesis-generating function $\mathcal{G}$ and a decision function f by choosing action $a = f(\mathcal{G}(h))$ after seeing history h. We discuss a number of examples below to elucidate the framework and as a basis for the results we later present.

Example 10 (Bayesian agent) *Suppose that ν is a stochastic environment and $\mathcal{G}(h) = \{\nu(\cdot|h)\}$ for all h and let f be a strictly rational decision function. The agent formed by combining f and $\mathcal{G}$ is a rational agent in the stricter sense . Also, if $\mathcal{M}$ is a finite or countable class of environments and $\mathcal{G}(h) = \{\nu(\cdot|h) \mid \nu \in \mathcal{M}\}$ for all $h \in \mathcal{H}$ (same $\mathcal{M}$ for all h) and there are $\omega_\nu > 0,\ \nu \in \mathcal{M}, \sum_{\nu\in\mathcal{M}} w_\nu = 1$ such that $a = \pi(\epsilon)$ for a policy*

$$\pi \in \arg\max_{\tilde{\pi}} \sum_{\nu\in\mathcal{G}(h)} \omega_\nu V_\nu^{\tilde{\pi}}, \tag{7}$$

then we say that the agent is Bayesian and it can be represented more simply in the first way by $\mathcal{G}(h) = \{\sum w_\nu \nu(\cdot|h)\}$ due to linearity of the value function (Hutter, 2005)

Example 11 (Optimist deterministic case) *Suppose that $\mathcal{M}$ is a finite class of deterministic environments and let $\mathcal{G}(h) = \{\nu(\cdot|h) \mid \nu \in \mathcal{M}$ consistent with $h\}$. If we combine $\mathcal{G}$ with the optimistic decision function we have defined the optimistic agents for classes of deterministic environments (Algorithm 1) from Section 4. In Section 7 we extend these agents to infinite classes by letting $\mathcal{G}(h_t)$ contain new environments that were not in $\mathcal{G}(h_{t-1})$.*

Example 12 (Optimistic AIXI) *Suppose that $\mathcal{M}$ is a finite class of stochastic environments and that $\mathcal{G}(h) = \{\nu(\cdot|h) \mid \nu \in \mathcal{M}\}$. If we combine $\mathcal{G}$ with the optimistic decision function we have defined the optimistic AIXI agent (Equation 2 with $\Xi = \mathcal{M}$).*

Example 13 (MBIE) *The Model Based Interval Estimation (MBIE) (Strehl et al., 2009) method for Markov Decision Processes (MDPs) defines $\mathcal{G}(h)$ as a set of MDPs (for a given state space) with transition probabilities in confidence intervals calculated from h. This is combined with the optimistic decision function. MBIE satisfies strong sample complexity guarantees for MDPs and is, therefore, an example of what we want but in a narrower setting.*

Example 14 (Optimist stochastic case) *Suppose that $\mathcal{M}$ is a finite class of stochastic environments and that $\mathcal{G}(h) = \{\nu(\cdot|h) \mid \nu \in \mathcal{M} : \nu(h) \geq z \max_{\tilde{\nu}\in\mathcal{M}} \tilde{\nu}(h)\}$ for some $z \in (0, 1)$. If we combine $\mathcal{G}$ with the optimistic decision function we have defined the optimistic agent with stochastic environments from Section 5.*

Example 15 (MERL and BayesExp) *Agents that switch explicitly between exploration and exploitation are typically* **not** *satisfying even our weak rationality demand. An example is Lattimore et al. (2013a) where the introduced Maximum Exploration Reinforcement Learning (MERL) agent performs certain tests when the remaining candidate environments are disagreeing sufficiently. This decision function is not satisfying rationality while our Algorithm 3, which uses the exclusion criteria of MERL but with an optimistic decision function, does satisfy our notion of rationality. Another example of an explicitly exploring irrational agent is BayesExp (Lattimore, 2014).*

4. Finite Classes of Dominant A Priori Environments

In this section, we study convergence results for optimistic agents with finite classes of dominant environments. In terms of the agent framework we here use an optimistic decision

function and a hypothesis-generating function that neither adds to nor removes from the initial class but just updates the environments through conditioning. Such agents were previously described in Example 12. In the next section we consider a setting where we instead of domination assume that one of the environments in the class is the true environment. The first setting is natural for Bayesian approaches, while the second is more frequentist in flavor. If we assume that all uncertainty is epistemic, i.e., caused by the agent's lack of knowledge, and that the true environment is deterministic, then for the first (Bayesian) setting the assumption means that the environments assign strictly positive probability to the truth. In the second (frequentist) setting, the assumption says that the environment class must contain this deterministic environment. In Section 6, we also consider a stochastic version of the second setting where the true environment is potentially stochastic in itself.

We first prove that AIXI is asymptotically optimal if the a priori environment ξ both dominates the true environment μ in the sense that $\exists c > 0 : \xi(\cdot) \geq c\mu(\cdot)$ and optimistic in the sense that $\forall h_t \; V^*_\xi(h_t) \geq V^*_\mu(h_t)$ (for large t). We extend this by replacing ξ with a finite set Ξ and prove that we then only need there to be, for each h_t (for t large), some $\xi \in \Xi$ such that $V^*_\xi(h_t) \geq V^*_\mu(h_t)$. We refer to this second domination property as optimism. The first domination property, which we simply refer to as domination, is most easily satisfied for $\xi(\cdot) = \sum_{\nu \in \mathcal{M}} w_\nu \nu(\cdot)$ with $w_\nu > 0$ where $\mathcal{M}$ is a countable class of environments with $\mu \in \mathcal{M}$. We provide a simple illustrative example for the first theorem and a more interesting one after the second theorem. First, we introduce some definitions related to the purpose of domination, namely it implies absolute continuity which according to the Blackwell-Dubins Theorem (Blackwell and Dubins, 1962) implies merging in total variation.

Definition 16 (Total variation distance, Merging, Absolute continuity)
i) The total variation distance between two (non-negative) measures P and Q is defined to be

$$d(P, Q) \; = \; \sup_A |P(A) - Q(A)|$$

where A ranges over the σ-algebra of the relevant measure space.
ii) P and Q are said to merge iff $d(P(\cdot|\omega_{1:t}), Q(\cdot|\omega_{1:t})) \to 0$ P-a.s. as $t \to \infty$, i.e., almost surely if the sequence ω is generated by P. The environments ν_1 and ν_2 merge under π if $\nu_1(\cdot|h_t, \pi)$ and $\nu_2(\cdot|h_t, \pi)$ merge.
iii) P is absolutely continuous with respect to Q if $Q(A) = 0$ implies that $P(A) = 0$.

We will make ample use of the classical Blackwell-Dubins Theorem (Blackwell and Dubins, 1962) so we state it explicitly.

Theorem 17 (Blackwell-Dubins Theorem) *If P is absolutely continuous with respect to Q, then P and Q merge P-almost surely.*

Lemma 18 (Value convergence for merging environments) *Given a policy π and environments μ and ν it follows that for all h*

$$|V^\pi_\mu(h) - V^\pi_\nu(h)| \; \leq \; \frac{1}{1-\gamma} d(\mu(\cdot|h, \pi), \nu(\cdot|h, \pi)).$$

Proof The lemma follows from the general inequality

$$\left|\mathbb{E}_P(f) - \mathbb{E}_Q(f)\right| \;\leq\; \sup|f| \cdot \sup_A \left|P(A) - Q(A)\right|$$

by letting f be the return in the history and $P = \mu(\cdot|h,\pi)$ and $Q = \nu(\cdot|h,\pi)$, and using $0 \leq f \leq 1/(1-\gamma)$ that follows from the rewards being in $[0,1]$. ■

The next theorem is the first of the two convergence theorems in this section. It relates to a strictly rational agent and imposes two conditions. The domination condition is a standard assumption that a Bayesian agent satisfies if it has strictly positive prior weight for the truth. The other assumption, the optimism assumption, is restrictive but the convergence result does not hold if only domination is assumed and the known alternative (Hutter, 2005) of demanding that a Bayesian agent's hypothesis class is self-optimizing is only satisfied for environments of very particular form such as ergodic Markov Decision Processes.

Algorithm 1: Optimistic-AIXI Agent (π°)

Require: Finite class of dominant a priori environments Ξ

$t = 1, h_0 = \epsilon$
repeat
  $(\pi^*, \xi^*) \in \arg\max_{\pi\in\Pi, \xi\in\Xi} V_\xi^\pi(h_{t-1})$
  $a_{t-1} = \pi^*(h_{t-1})$
  Perceive $o_t r_t$ from environment μ
  $h_t \leftarrow h_{t-1} a_{t-1} o_t r_t$
  $t \leftarrow t+1$
until end of time

Theorem 19 (AIXI convergence) *Suppose that $\xi(\cdot) \geq c\mu(\cdot)$ for some $c > 0$ and μ is the true environment. Also suppose that there μ-almost surely is $T_1 < \infty$ such that $V_\xi^*(h_t) \geq V_\mu^*(h_t)$ $\forall t \geq T_1$. Suppose that the policy π^* acts in μ according to the AIXI agent based on ξ, i.e.,*

$$\pi^* \in \arg\max_\pi V_\xi^\pi(\epsilon)$$

or equivalently Algorithm 1 with $\Xi = \{\xi\}$. Then there is μ-almost surely, i.e., almost surely if the sequence h_t is generated by π^* acting in μ, for every $\varepsilon > 0$, a time $T < \infty$ such that $V_\mu^{\pi^*}(h_t) \geq V_\mu^*(h_t) - \varepsilon$ $\forall t \geq T$.

Proof Due to the dominance we can (using the Blackwell-Dubins merging of opinions theorem (Blackwell and Dubins, 1962)) say that μ-almost surely there is for every $\varepsilon' > 0$, a $T < \infty$ such that $\forall t \geq T$ $d(\xi(\cdot|h_t,\pi^*), \mu(\cdot|h_t,\pi^*)) < \varepsilon'$ where d is the total variation distance. This implies that $|V_\xi^{\pi^*}(h_t) - V_\mu^{\pi^*}(h_t)| < \frac{\varepsilon'}{1-\gamma} := \varepsilon$ which means that, if $t \geq T$, $V_\mu^{\pi^*}(h_t) \geq V_\xi^*(h_t) - \varepsilon \geq V_\mu^*(h_t) - \varepsilon$. ■

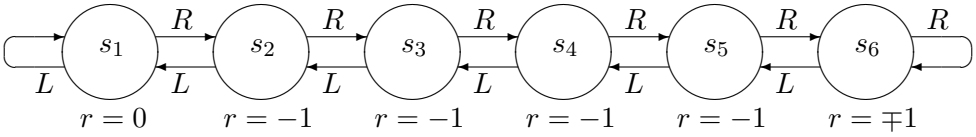


Figure 1: Line environment

Example 20 (Line environment) *We consider an agent who, when given a class of environments, will choose its prior based on simplicity in accordance with Occam's razor (Hutter, 2005). First let us look at a class $\mathcal{M}$ of two environments which both have six states (Figure 1) $s_1, ..., s_6$ and two actions L (left) and R (right). Action R changes s_k to s_{k+1}, L to s_{k-1}. Also L in s_1 or R in s_6 result in staying. We start at s_1. Being at s_1 has a reward of 0, s_2, s_3, s_4, s_5 have reward -1 while the reward in s_6 depends on the environment. In one of the environments ν_1, this reward is $+1$ while in ν_2 it is -1. Since ν_2 is not simpler than ν_1 it will not have higher weight and if γ is only modestly high the agent will not explore along the line despite that in ν_2 it would be optimal to do so. However, if we define another environment ν_3 by letting the reward at s_6 be really high, then when including ν_3 in the mixture, the agent will end up with an a priori environment that is optimistic for ν_1 and ν_2 and we can guarantee optimality for any γ.*

In the next theorem we prove that for the optimistic agent with a class of a priori environments, only one of them needs to be optimistic at a time while all are assumed to be dominant. As before, domination is achieved if the a priori environments are of the form of a mixture over a hypothesis class containing the truth. The optimism is in this case milder and is e.g., trivially satisfied if the truth is one of the a priori environments. Since the optimistic agent is guaranteed convergence under milder assumptions we believe that it would succeed in a broader range of environments than the single-prior rational agent.

Theorem 21 (Multiple-prior convergence) *Suppose that Ξ is a finite set of a priori environments such that for each $\xi \in \Xi$ there is $c_{\xi,\mu} > 0$ such that $\xi(\cdot) \geq c_{\xi,\mu}\mu(\cdot)$ where μ is the true environment. Also suppose that there μ-almost surely is $T_1 < \infty$ such that for $t \geq T_1$ there is $\xi_t \in \Xi$ such that $V^*_{\xi_t}(h_t) \geq V^*_\mu(h_t)$. Suppose that the policy π°, defined as in (2) or equivalently Algorithm 1, acts according to the rational optimistic agent based on Ξ in μ. Then there is μ-almost surely, for every $\varepsilon > 0$, a time $T < \infty$ such that $V^{\pi^\circ}_\mu(h_t) \geq V^*_\mu(h_t) - \varepsilon \ \forall t \geq T$.*

The theorem is proven by combining the proof technique from the previous theorem with the following lemma. We have made this lemma easier to formulate by formulating it for time $t = 0$ (when the history is the empty string ϵ), though when proving Theorem 21 it is used for a later time point when the environments in the class have merged sufficiently under π° in the sense of total variation diameter. The lemma simply says that if the environments are sufficiently close under π°, then π° must be nearly optimal. This follows from optimism since it means that the value function that π° maximizes is the highest among the value functions for the environments in the class and it is also close to the actual value by the

assumption. The only thing that makes the proof non-trivial is that π° might maximize for different environments at each step but since they are all close, the conclusion that would otherwise have been trivial is still true. One can simply construct a new environment that combines the dynamics of the environments that are optimistic at different times. Then, the policy maximizes value for this environment at each times step and this new environment is also close to all the environments in the class. We let ν_h^* be an environment in $\arg\max_\nu \max_\pi V_\nu^\pi(h)$ that π° uses to choose the next action after experiencing h.

Definition 22 (Environment used by π°) *Suppose that Ξ is a finite set of environments and that π° is the optimistic agent. Let ν_h^* be an environment in $\arg\max_\nu \max_\pi V_\nu^\pi(h)$ that π° uses to choose the next action after experiencing h, i.e., ν_h^* is such that $V_{\nu_h^*}^*(h) = \max_{\nu,\pi} V_\nu^\pi(h)$ and $\pi^\circ(h) = \tilde{\pi}(h)$ for some $\tilde{\pi} \in \arg\max_\pi V_{\nu_h^*}^\pi(h)$. Note, the choice might not be unique.*

The next definition introduces the concept of constructing an environment that is consistently used.

Definition 23 (Constructed environment) *Define $\hat{\nu}$ by $\hat{\nu}(o,r|h,a) = \nu_h^*(o,r|h,a)$.*

The following lemma is intuitively obvious. It says that if at each time step we define an environment by using the dynamics of the environment in the class that promises the most value, then the resulting environment will always be optimistic relative to any environment in the class. The proof is only complicated by the cumbersome notation required due to studying fully general reinforcement learning. The key tool is the Bellman equation that for general reinforcement learning is

$$V_\nu^\pi(h) = \sum_{o,r} \nu(o,r|h,\pi(h))[r + \gamma V_\nu^\pi(h')]$$

where $h' = h\pi(h)or$. Together with induction this will be used to prove the next lemma.

Lemma 24 $V_{\hat{\nu}}^{\pi^\circ} \geq \max_{\nu\in\mathcal{M},\pi} V_\nu^\pi(\epsilon)$

Proof Let V_ν^π denote $V_\nu^\pi(\epsilon)$. We reason by induction using a sequence of environments approaching $\hat{\nu}$. Let

$$\hat{\nu}_s(o_t r_t|h_{t-1},a) = \hat{\nu}(o_t r_t|h_{t-1},a)\ \forall h_{t-1}\forall a,\ t \leq s$$

and

$$\hat{\nu}_s(o_t r_t|h_{t-1},a) = \nu_{h_s}^*(o_t r_t|h_{t-1},a),\ \forall h_{t-1}\forall a, t > s.$$

$\hat{\nu}_1$ equals ν_ε^* at all time points and thus $V_{\hat{\nu}_1}^\pi = V_{\nu_\varepsilon^*}^\pi$. Let $\hat{R}_t^\nu$ be the expected accumulated (discounted) reward ($\mathbb{E}\sum_{i=1}^t \gamma^{i-1} r_i$) up to time t when following π° up until that time in the environment ν. We first do the base case $t = 1$.

$$\max_{\pi_{2:\infty}} V_{\hat{\nu}_2}^{\pi^\circ{}_{0:1}\pi_{2:\infty}} = \max_{\pi_{1:\infty}} (\hat{R}_1^{\nu_\epsilon^*} + \gamma \mathbb{E}_{h_1|\nu_\epsilon^*,\pi^\circ} V_{\nu_{h_1}^*}^{\pi_{1:\infty}}(h_1)) \geq$$

$$\max_{\pi_{1:\infty}}(\hat{R}_1^{\nu^*_\epsilon} + \gamma \mathbb{E}_{h_1|\nu^*_\epsilon,\pi^\circ} V^{\pi_{1:\infty}}_{\nu^*_\epsilon}(h_1)) = \max_\pi V^\pi_{\hat{\nu}_1}.$$

The middle inequality is due to $\max_\pi V^\pi_{\nu^*_{h_1}}(h_1) \geq \max_\pi V^\pi_\nu(h_1)\ \forall \nu \in \Xi$. The first equality is the Bellman equation together with the fact that π° makes a first action that optimize for ν^*_ϵ. The second is due to $\hat{\nu}_1 = \nu^*_\epsilon$ and the Bellman equation. In the same way,

$$\forall k \ \max_{\pi_{k:\infty}} V^{\pi^\circ{}_{0:k-1}\pi_{k:\infty}}_{\hat{\nu}_k} \geq \max_{\pi_{k-1:\infty}} V^{\pi^\circ{}_{0:k-2}\pi_{k-1:\infty}}_{\hat{\nu}_{k-1}}$$

and it follows by induction that $V^{\pi^\circ}_{\hat{\nu}} \geq \max_{\pi,\nu\in\mathcal{M}} V^\pi_\nu \geq V^*_\mu$.

■

Lemma 25 (Optimism is nearly optimal) *Suppose that the assumptions of Theorem 21 hold and that we denote the optimistic agent again by (π°). Then for each $\varepsilon > 0$ there exists $\tilde{\varepsilon} > 0$ such that $V^{\pi^\circ}_\mu(\epsilon) \geq \max_\pi V^\pi_\mu(\epsilon) - \varepsilon$ whenever*

$$\forall h, \forall \nu_1, \nu_2 \in \Xi,\ |V^{\pi^\circ}_{\nu_1}(h) - V^{\pi^\circ}_{\nu_2}(h)| < \tilde{\varepsilon}.$$

Proof We will show that if we choose $\tilde{\varepsilon}$ small enough, then

$$|V^{\pi^\circ}_{\hat{\nu}} - V^{\pi^\circ}_\mu| < \varepsilon \tag{8}$$

where μ is the true environment. Equation (8), when proven to hold when $\tilde{\varepsilon}$ is chosen small enough, concludes the proof since then $|V^*_\mu - V^{\pi^\circ}_\mu| < \varepsilon$, due to $V^{\pi^\circ}_{\hat{\nu}} \geq V^*_\mu \geq V^{\pi^\circ}_\mu$. This is easy since

$$|V^{\pi^\circ}_{\hat{\nu}_\varepsilon} - V^{\pi^\circ}_{\hat{\nu}}| < \frac{\tilde{\varepsilon}}{1-\gamma}$$

and if $\tilde{\varepsilon} + \frac{\tilde{\varepsilon}}{1-\gamma} \leq \varepsilon$ then (8) holds and the proof is complete as we concluded above since $|V^{\pi^\circ}_{\hat{\nu}_\varepsilon} - V^{\pi^\circ}_\mu| < \tilde{\varepsilon}$. ■

Proof of Theorem 21. Since Ξ is finite and by using Theorem 17 (Blackwell-Dubins), there is for every ε', a $T < \infty$ when $\forall \xi \in \Xi\ \forall t \geq T,\ d(\xi(\cdot|h_t,\pi^\circ), \mu(\cdot|h_t,\pi^\circ)) < \varepsilon'$. This implies that $\forall \xi \in \Xi\ |V^{\pi^\circ}_\xi(h_t) - V^{\pi^\circ}_\mu(h_t)| < \frac{\varepsilon'}{1-\gamma}$ by Lemma 18. Choose ε' such that $\frac{\varepsilon'}{1-\gamma} = \varepsilon$. Applying Lemma 25 with class $\tilde{\Xi} = \{\xi(\cdot|\ h_T) : \xi \in \Xi\}$ now directly proves the result. The application of Lemma 25 is viewing time T from this proof as time zero and the ϵ context. ■

Example 26 (Multiple-prior AIXI) *For any Universal Turing Machine (UTM) U the corresponding Solomonoff distribution ξ_U is defined by putting coin flips on the input tape (see Li and Vitani (2008); Hutter (2005) for details). ξ_U is dominant for any lower semi-computable semi-measure over infinite sequences. Hutter (2005) extends these constructions and introduces an environment ξ_U that is dominant for all reactive lower semi-computable reactive environments and defines the AIXI agent based on it as in Theorem 19. A difficulty*

is to choose the UTM to use. Many have without success tried to find a single "natural" Turing machine and it might in fact be impossible (Müller, 2010). Examples includes defining a machine from a programming language like C or Haskell and another possibility is to use Lambda calculus. With the approach that we introduce in this article one can pick finitely many machines that one considers to be natural. Though this does not fully resolve the issue, and the issue might not be fully resolvable, it alleviates it.

5. Finite Classes of Deterministic (Non-Dominant) A Priori Environments

In this section, we perform a different sort of analysis where it is not assumed that all the environments in Ξ dominate the true environment μ. We instead rely on the assumption that the true environment is a member of the agent's class of environments. The a priori environments are then naturally thought of as a hypothesis class rather than mixtures over some hypothesis class and we will write $\mathcal{M}$ instead of Ξ to mark this difference. We begin with the deterministic case, where one could not have introduced the domination assumption, in this section and look at stochastic non-dominant a priori environments in the next. The agent in this section can be described, as was done in Example 11 as having an optimistic decision function and a hypothesis-generating function that begins with an initial class and removes excluded environments.

5.1 Optimistic Agents for Deterministic Environments

Given a finite class of deterministic environments $\mathcal{M} = \{\nu_1, ..., \nu_m\}$, we define an algorithm that for any unknown environment from $\mathcal{M}$ eventually achieves optimal behavior in the sense that there exists T such that maximum reward is achieved from time T onwards. The algorithm chooses an optimistic hypothesis from $\mathcal{M}$ in the sense that it picks the environment in which one can achieve the highest reward and then the policy that is optimal for this environment is followed. If this hypothesis is contradicted by the feedback from the environment, a new optimistic hypothesis is picked from the environments that are still consistent with h. This technique has the important consequence that if the hypothesis is not contradicted, the agent acts optimally even when optimizing for an incorrect hypothesis.

Let $h_t^{\pi,\nu}$ be the history up to time t generated by policy π in environment ν. In particular let $h^\circ := h^{\pi^\circ,\mu}$ be the history generated by Algorithm 2 (policy π°) interacting with the actual "true" environment μ. At the end of cycle t we know $h_t^\circ = h_t$. An environment ν is called consistent with h_t if $h_t^{\pi^\circ,\nu} = h_t$. Let $\mathcal{M}_t$ be the environments consistent with h_t. The algorithm only needs to check whether $o_t^{\pi^\circ,\nu} = o_t$ and $r_t^{\pi^\circ,\nu} = r_t$ for each $\nu \in \mathcal{M}_{t-1}$, since previous cycles ensure $h_{t-1}^{\pi^\circ,\nu} = h_{t-1}$ and trivially $a_t^{\pi^\circ,\nu} = a_t$. The maximization in Algorithm 2 that defines optimism at time t is performed over $\nu \in \mathcal{M}_{t-1}$, the set of consistent hypotheses at time t, and $\pi \in \Pi = \Pi^{all}$ is the class of *all* deterministic policies. In Example 11, we described the same agent by saying that it combines an optimistic decision function with a hypothesis generating function that begins with an initial finite class of deterministic environments and excludes those that are contradicted. More precisely, we have here first narrowed down the optimistic decision function further by saying that it needs to stick to hypothesis until contradicted, while we will below further discuss not

Algorithm 2: Optimistic Agent (π°) for Deterministic Environments

Require: Finite class of deterministic environments $\mathcal{M}_0 \equiv \mathcal{M}$
$t = 1$
repeat
  $(\pi^*, \nu^*) \in \arg\max_{\pi\in\Pi,\nu\in\mathcal{M}_{t-1}} V_\nu^\pi(h_{t-1})$
  repeat
    $a_{t-1} = \pi^*(h_{t-1})$
    Perceive $o_t r_t$ from environment μ
    $h_t \leftarrow h_{t-1}a_{t-1}o_t r_t$
    Remove all inconsistent ν from $\mathcal{M}_t$ ($\mathcal{M}_t := \{\nu \in \mathcal{M}_{t-1} : h_t^{\pi^\circ,\nu} = h_t\}$)
    $t \leftarrow t + 1$
  until $\nu^* \notin \mathcal{M}_{t-1}$
until $\mathcal{M}$ is empty

making this simplifying extra specification. Its an important fact, proven below, that an optimistic hypothesis does not cease to be optimistic until contradicted. The guarantees we prove for this agent are stronger than in the previous chapter where only dominance was assumed while here we assume that the truth belongs to the given finite class of deterministic environments.

Theorem 27 (Optimality, Finite deterministic class) *Suppose $\mathcal{M}$ is a finite class of deterministic environments. If we use Algorithm 2 (π°) in an environment $\mu \in \mathcal{M}$, then there is $T < \infty$ such that*

$$V_\mu^{\pi^\circ}(h_t) \;=\; \max_\pi V_\mu^\pi(h_t) \;\forall t \geq T.$$

A key to proving Theorem 27 is time-consistency (Lattimore and Hutter, 2011b) of geometric discounting. The following lemma tells us that if the agent acts optimally with respect to a chosen optimistic hypothesis, this hypothesis remains optimistic until contradicted.

Lemma 28 (Time-consistency) *Suppose $(\pi^*, \nu^*) \in \arg\max_{\pi\in\Pi,\nu\in\mathcal{M}_{t-1}} V_\nu^\pi(h_{t-1})$ and that an agent acts according to π^* from a time point t to another time point $\tilde{t} - 1$, i.e., $a_s = \pi^*(h_{s-1})$ for $t \leq s \leq \tilde{t} - 1$. For any choice of $t < \tilde{t}$ such that ν^* is still consistent at time $\tilde{t}$, it holds that $(\pi^*, \nu^*) \in \arg\max_{\pi\in\Pi,\nu\in\mathcal{M}_{\tilde{t}}} V_\nu^\pi(h_{\tilde{t}})$.*

Proof Suppose that $V_{\nu^*}^{\pi^*}(h_{\tilde{t}}) < V_{\tilde{\nu}}^{\tilde{\pi}}(h_{\tilde{t}})$ for some $\tilde{\pi}$, $\tilde{\nu}$. It holds that $V_{\nu^*}^{\pi^*}(h_t) = C + \gamma^{\tilde{t}-t} V_{\nu^*}^{\pi^*}(h_{\tilde{t}})$ where C is the accumulated reward between t and $\tilde{t} - 1$. Let $\hat{\pi}$ be a policy that equals π^* from t to $\tilde{t} - 1$ and then equals $\tilde{\pi}$. It follows that $V_{\tilde{\nu}}^{\hat{\pi}}(h_t) = C + \gamma^{\tilde{t}-t} V_{\tilde{\nu}}^{\hat{\pi}}(h_{\tilde{t}}) > C + \gamma^{\tilde{t}-t} V_{\nu^*}^{\pi^*}(h_{\tilde{t}}) = V_{\nu^*}^{\pi^*}(h_t)$ which contradicts the assumption $(\pi^*, \nu^*) \in \arg\max_{\pi\in\Pi,\nu\in\mathcal{M}_t} V_\nu^\pi(h_t)$. Therefore, $V_{\nu^*}^{\pi^*}(h_{\tilde{t}}) \geq V_{\tilde{\nu}}^{\tilde{\pi}}(h_{\tilde{t}})$ for all $\tilde{\pi}$, $\tilde{\nu}$. ■

Proof (Theorem 27) At time t we know h_t. If some $\nu \in \mathcal{M}_{t-1}$ is inconsistent with h_t, i.e., $h_t^{\pi^\circ,\nu} \neq h_t$, it gets removed, i.e., is not in $\mathcal{M}_{t'}$ for all $t' \geq t$.

Since $\mathcal{M}_0 = \mathcal{M}$ is finite, such inconsistencies can only happen finitely often, i.e., from some T onwards we have $\mathcal{M}_t = \mathcal{M}_\infty$ for all $t \geq T$. Since $h_t^{\pi^\circ,\mu} = h_t \ \forall t$, we know that $\mu \in \mathcal{M}_t \ \forall t$.

Assume $t \geq T$ henceforth. The optimistic hypothesis will not change after this point. If the optimistic hypothesis is the true environment μ, the agent has obviously chosen a truly optimal policy.

In general, the optimistic hypothesis ν^* is such that it will never be contradicted while actions are taken according to π°, hence (π^*, ν^*) do not change anymore. This implies

$$V_\mu^{\pi^\circ}(h_t) \;=\; V_\mu^{\pi^*}(h_t) \;=\; V_{\nu^*}^{\pi^*}(h_t) \;=\; \max_{\nu\in\mathcal{M}_t}\max_{\pi\in\Pi} V_\nu^\pi(h_t) \;\geq\; \max_{\pi\in\Pi} V_\mu^\pi(h_t)$$

for all $t \geq T$. The first equality follows from π° equals π^* from $t \geq T$ onwards. The second equality follows from consistency of ν^* with $h_{1:\infty}^\circ$. The third equality follows from optimism, the constancy of π^*, ν^*, and $\mathcal{M}_t$ for $t \geq T$, and time-consistency of geometric discounting (Lemma 28). The last inequality follows from $\mu \in \mathcal{M}_t$. The reverse inequality $V_\mu^{\pi^*}(h_t) \leq \max_\pi V_\mu^\pi(h_t)$ follows from $\pi^* \in \Pi$. Therefore π° is acting optimally at all times $t \geq T$. ■

Besides the eventual optimality guarantee above, we also provide a bound on the number of time steps for which the value of following Algorithm 2 is more than a certain $\varepsilon > 0$ less than optimal. The reason this bound is true is that we only have such suboptimality for a certain number of time steps immediately before the current hypothesis becomes inconsistent and the number of such inconsistency points are bounded by the number of environments. Note that the bound tends to infinity as $\varepsilon \to 0$, hence we need Theorem 27 with its distinct proof technique for the $\varepsilon = 0$ case.

Theorem 29 (Finite error bound) *Following π° (Algorithm 2),*

$$V_\mu^{\pi^\circ}(h_t) \geq \max_{\pi\in\Pi} V_\mu^\pi(h_t) - \varepsilon,\ 0 < \varepsilon < 1/(1-\gamma)$$

for all but at most $K\frac{-\log\varepsilon(1-\gamma)}{1-\gamma} \leq |\mathcal{M}-1|\frac{-\log\varepsilon(1-\gamma)}{1-\gamma}$ time steps t where K is the number of times that some environment is contradicted.

Proof Consider the ℓ-truncated value

$$V_{\nu,\ell}^\pi(h_t) \;:=\; \sum_{i=t+1}^{t+\ell} \gamma^{i-t-1} r_i$$

where the sequence r_i are the rewards achieved by following π from time $t+1$ to $t+\ell$ in ν after seeing h_t. By letting $\ell = \frac{\log\varepsilon(1-\gamma)}{\log\gamma}$ (which is positive due to negativity of both numerator and denominator) we achieve $|V_{\nu,\ell}^\pi(h_t) - V_\nu^\pi(h_t)| \leq \frac{\gamma^l}{1-\gamma} = \epsilon$. Let (π_t^*, ν_t^*) be the policy-environment pair selected by Algorithm 2 in cycle t.

Let us first assume $h_{t+1:t+\ell}^{\pi^\circ,\mu} = h_{t+1:t+\ell}^{\pi^\circ,\nu_t^*}$, i.e., ν_t^* is consistent with $h_{t+1:t+\ell}^\circ$, and hence π_t^* and ν_t^* do not change from $t+1, ..., t+\ell$ (inner loop of Algorithm 2). Then

$$V_\mu^{\pi^\circ}(h_t) \overset{\text{drop terms,}}{\geq} V_{\mu,\ell}^{\pi^\circ}(h_t) \overset{\text{same } h_{t+1:t+\ell},}{=} V_{\nu_t^*,\ell}^{\pi^\circ}(h_t) \overset{\pi^\circ=\pi_t^* \text{ on } h_{t+1:t+\ell},}{=} V_{\nu_t^*,\ell}^{\pi_t^*}(h_t)$$

$$\underset{\text{bound extra terms}}{\overset{\geq}{\uparrow}} V_{\nu_t^*}^{\pi_t^*}(h_t) - \tfrac{\gamma^\ell}{1-\gamma} \quad \underset{\text{def. of } (\pi_t^*, \nu_t^*) \text{ and } \varepsilon := \frac{\gamma^\ell}{1-\gamma}}{=} \quad \max_{\nu \in \mathcal{M}_t} \max_{\pi \in \Pi} V_\nu^\pi(h_t) - \varepsilon \quad \underset{\mu \in \mathcal{M}_t}{\overset{\geq}{\uparrow}} \quad \max_{\pi \in \Pi} V_\mu^\pi(h_t) - \varepsilon.$$

Now let $t_1, ..., t_K$ be the times t at which the currently selected ν_t^* becomes inconsistent with h_t, i.e., $\{t_1, ..., t_K\} = \{t : \nu_t^* \notin \mathcal{M}_t\}$.

Therefore $h^\circ_{t+1:t+\ell} \neq h^{\pi^\circ, \nu_t^*}_{t+1:t+\ell}$ (only) at times $t \in \mathcal{T}_\times := \bigcup_{i=1}^K \{t_i - \ell, ..., t_i - 1\}$, which implies $V_\mu^{\pi^\circ}(h_t) \geq \max_{\pi \in \Pi} V_\mu^\pi(h_t) - \varepsilon$ except possibly for $t \in \mathcal{T}_\times$. Finally

$$|\mathcal{T}_\times| \; = \; \ell \cdot K \; = \; \frac{\log \varepsilon(1-\gamma)}{\log \gamma} K \; \leq \; K \frac{\log \varepsilon(1-\gamma)}{\gamma - 1} \; \leq \; |\mathcal{M} - 1| \frac{\log \varepsilon(1-\gamma)}{\gamma - 1}$$

■

Conservative or liberal optimistic agents. We refer to the algorithm above as the conservative agent since it keeps its hypothesis for as long as it can. We can define a more *liberal* agent that re-evaluates its optimistic hypothesis at every time step and can switch between different optimistic policies at any time. Algorithm 2 is actually a special case of this as shown by Lemma 28. The liberal agent is really a class of algorithms and this larger class of algorithms consists of exactly the algorithms that are optimistic at every time step without further restrictions. The conservative agent is the subclass of algorithms that only switch hypothesis when the previous is contradicted. The results for the conservative agent can be extended to the liberal one. We do this for Theorem 27 in Appendix A together with analyzing further subtleties about the conservative case. It is worth noting that the liberal agent can also be understood as a conservative agent but for an extended class of environments where one creates a new environment by letting it have, at each time step, the dynamics of the chosen optimistic environment. Contradiction of such an environment will then always coincide with contradiction of the chosen optimistic environment and there will be no extra contradictions due to these new environments. Hence, the finite-error bound can also be extended to the liberal case. In the stochastic case below, we have to use a liberal agent. Note that both the conservative and liberal agents are based on an optimistic decision function and the same hypothesis-generating function. There can be several optimistic decision functions due to ties.

5.2 Environments and Laws

The bounds given above have a linear dependence on the number of environments in the class and though this is the best one can do in general (Lattimore et al., 2013a), it is bad compared to what we are used to from Markov Decision Processes (Lattimore and Hutter, 2012) where the linear (up to logarithms) dependence is on the size of the state space instead. Markov Decision Processes are finitely generated in a sense that makes it possible to exclude whole parts of the environment class together, e.g., all environments for which a state s_2 is likely to follow the state s_1 if action a_1 is taken. Unfortunately, the Markov assumption is very restrictive.

In this section we will improve the bounds above by introducing the concept of laws and of an environment being generated by a set of laws. Any environment class can be described this way and the linear dependence on the size of the environment class in the bounds is

replaced by a linear dependence on the size of the smallest set of laws that can generate the class. Since any class is trivially generated by the laws that simply equal an environment from the class each, we are not making further restrictions compared to previous results. However, in the worst situations the bounds presented here equal the previous bounds, while for other environment classes the bounds in this section are exponentially better. The latter classes with good bounds are the only option for practical generic agents. Classes of such form have the property that one can exclude laws and thereby exclude whole classes of environments simultaneously like when one learns about a state transition for an MDP.

Environments defined by laws. We consider observations of the form of a feature vector $o = \vec{x} = (x_j)_{j=1}^m \in \mathcal{O} = \times_{j=1}^m \mathcal{O}_j$ including the reward as one coefficient where x_j is an element of some finite alphabet $\mathcal{O}_i$. Let $\mathcal{O}_\perp = \times_{j=1}^m (\mathcal{O}_j \cup \{\perp\})$, i.e., $\mathcal{O}_\perp$ consists of the feature vectors from $\mathcal{O}$ but where some elements are replaced by a special letter $\perp$. The meaning of $\perp$ is that there is no prediction for this feature. We first consider deterministic laws.

Definition 30 (Deterministic laws) *A law is a function* $\tau : \mathcal{H} \times \mathcal{A} \to \mathcal{O}_\perp$.

Using a feature vector representation of the observations and saying that a law predicts some of the features is a convenient special case of saying that the law predicts that the next observation will belong to a certain subset of the observation space. Each law τ predicts, given the history and a new action, some or none but not necessarily all of the features x_j at the next time point. We first consider sets of laws such that for any given history and action, and for every feature, there is at least one law that makes a prediction of this feature. Such sets are said to be complete.

Definition 31 (Complete set of laws) *A set of laws* $\tilde{\mathcal{T}}$ *is complete if*

$$\forall h, a \forall j \in \{1, ..., m\} \; \exists \tau \in \tilde{\mathcal{T}} : \tau(h, a)_j \neq \perp.$$

We will only consider combining deterministic laws that never contradict each other and we call such sets of laws coherent. The main reason for this restriction is that one can then exclude a law when it is contradicted. If one does not demand coherence, an environment might only sometimes be consistent with a certain law and the agent can then only exclude the contradicted environment, not the contradicted law which is key to achieving better bounds.

Definition 32 (Coherent set of laws) *We say that* $\tilde{\mathcal{T}}$ *is coherent if for all* $\tau \in \tilde{\mathcal{T}}, h, a$ *and* j

$$\tau(h, a)_j \neq \perp \; \Rightarrow \; \tilde{\tau}(h, a)_j \in \{\perp, \tau(h, a)_j\} \; \forall \tilde{\tau} \in \tilde{\mathcal{T}}.$$

Definition 33 (Environment from a complete and coherent set of laws) *Given a complete and coherent set of laws* $\tilde{\mathcal{T}}$, $\nu(\tilde{\mathcal{T}})$ *is the unique environment* ν *which is such that*

$$\forall h, a \forall j \in \{1, ..., m\} \exists \tau \in \tilde{\mathcal{T}} : \nu(h, a)_j = \tau(h, a)_j.$$

The existence of $\nu(\tilde{\mathcal{T}})$ *follows from completeness of* $\tilde{\mathcal{T}}$ *and uniqueness is due to coherence.*

Definition 34 (Environment class from deterministic laws) *Given a set of laws $\mathcal{T}$, let $\mathcal{C}(\mathcal{T})$ denote the complete and coherent subsets of $\mathcal{T}$. Given a set of laws $\mathcal{T}$, we define the class of environments generated by $\mathcal{T}$ through*

$$\mathcal{M}(\mathcal{T}) := \{\nu(\tilde{\mathcal{T}}) \; | \tilde{\mathcal{T}} \in \mathcal{C}(\mathcal{T})\}.$$

Example 35 (Deterministic laws for fixed vector) *Consider an environment with a constant binary feature vector of length m. There are 2^m such environments. Every such environment can be defined by combining m out of a class of $2m$ laws. Each law says what the value of one of the features is, one law for 0 and one for 1. In this example, a coherent set of laws is simply one feature for each coefficient. The generated environment is the constant vector defined by that vector and the set of all the generated environments is the full set of 2^m environments.*

Error analysis. Every contradiction of an environment is a contradiction of at least one law and there are finitely many laws. This is what is needed for the finite error result from Section 4 to hold but with $|\mathcal{M}|$ replaced by $|\mathcal{T}|$ (see Theorem 36 below) which can be exponentially smaller. Furthermore, the extension to countable $\mathcal{T}$ works the same as in Theorem 45.

Theorem 36 (Finite error bound when using laws) *Suppose that $\mathcal{T}$ is a finite class of deterministic laws and let $\mathcal{G}(h) = \{\nu(\cdot|h) \mid \nu \in \mathcal{M}(\{\tau| \; \tau \in \mathcal{T} \text{ consistent with } h\})\}$. We define $\bar{\pi}$ by combining $\mathcal{G}$ with the optimistic decision function. Following $\bar{\pi}$ for a finite class of deterministic laws $\mathcal{T}$ in an environment $\mu \in \mathcal{M}(\mathcal{T})$, we have for any $0 < \varepsilon < \frac{1}{1-\gamma}$ that*

$$V_\mu^{\bar{\pi}}(h_t) \geq \max_\pi V_\mu^\pi(h_t) - \varepsilon \tag{9}$$

for all but at most $|\mathcal{T} - l| \frac{-\log \varepsilon(1-\gamma)}{1-\gamma}$ time steps t where l is the minimum number of laws from $\mathcal{T}$ needed to define a complete environment.

Proof This theorem follows from Theorem 29 since there are at most $K = |\mathcal{T} - l|$ time steps with a contradiction.

■

6. Finite Classes of Stochastic Non-Dominant A Priori Environments

A stochastic hypothesis may never become completely inconsistent in the sense of assigning zero probability to the observed sequence while still assigning very different probabilities than the true environment. Therefore, we exclude based on a threshold for the probability assigned to the generated history proportional to the highest probability assigned by some environment in the remaining class. An obvious alternative is to instead compare to a weighted average of all the remaining environments as done by Lattimore et al. (2013b) for the BayesExp algorithm. This latter alternative means that one can interpret the criterion as excluding environments of low posterior probability where the weights define the prior. The alternatives differ only by a constant factor depending on the weights.

Unlike in the deterministic case, a hypothesis can cease to be optimistic without having been excluded. We, therefore, only consider an algorithm that re-evaluates its optimistic hypothesis at every time step. Algorithm 3 specifies the procedure and Theorem 37 states that it is asymptotically optimal. We previously introduced the agent described in Algorithm 3, in Example 14 by saying it has an optimistic decision function and by describing the hypothesis-generating function based on a criterion for excluding environments from an initial class. We also consider a different exclusion criterion, i.e., a different hypothesis-generating function, for an optimistic agent to be able to present sample complexity bounds that we believe also holds for the first agent. The criterion used to achieve near-optimal sample complexity has previously been used in the MERL algorithm (Lattimore et al., 2013a), which has a decision function that we deem irrational according to our theory. Our agent instead uses an optimistic decision function but the same hypothesis-generating function as MERL. A very similar agent and bound can also be achieved as an optimistically acting realization of the adaptive k-meteorologists' algorithm by Diuk et al. (2009) and its bound. This agent would only have a slightly different exclusion criterion compared to MERL. A further step that we do not take here would be to improve the bounds dramatically by using stochastic laws (Sunehag and Hutter, 2015) as we did with deterministic laws previously.

Algorithm 3: Optimistic Agent (π°) with Stochastic Finite Class

Require: Finite class of stochastic environments $\mathcal{M}_1 \equiv \mathcal{M}$, threshold $z \in (0,1)$

$t = 1$
repeat
  $(\pi^*, \nu^*) = \arg\max_{\pi,\nu\in\mathcal{M}_t} V_\nu^\pi(h_{t-1})$
  $a_{t-1} = \pi^*(h_{t-1})$
  Perceive $o_t r_t$ from environment μ
  $h_t \leftarrow h_{t-1} a_{t-1} o_t r_t$
  $t \leftarrow t+1$
  $\mathcal{M}_t := \{\nu \in \mathcal{M}_{t-1} : \frac{\nu(h_t|a_{1:t})}{\max_{\tilde{\nu}\in\mathcal{M}} \tilde{\nu}(h_t|a_{1:t})} > z\}$
until the end of time

Theorem 37 (Optimality, Finite stochastic class) *Define π° by using Algorithm 3 with any threshold $z \in (0,1)$ and a finite class $\mathcal{M}$ of stochastic environments containing the true environment μ, then with probability $1 - z|\mathcal{M} - 1|$ there exists, for every $\varepsilon > 0$, a number $T < \infty$ such that*

$$V_\mu^{\pi^\circ}(h_t) \; > \; \max_\pi V_\mu^\pi(h_t) - \varepsilon \;\; \forall t \geq T.$$

We borrow some techniques from Hutter (2009a) that introduced a "merging of opinions" result that generalized the classical theorem by Blackwell and Dubins (1962), restated here as Theorem 17. The classical result says that it is sufficient that the true measure (over infinite sequences) is absolutely continuous with respect to a chosen a priori distribution to guarantee that they will almost surely merge in the sense of total variation distance. The generalized version is given in Lemma 38. When we combine a policy π with an environment ν by letting the actions be taken by the policy, we have defined a measure, denoted by $\nu(\cdot|\pi)$, on the space of infinite sequences from a finite alphabet. We denote such a sample

sequence by ω and the a:th to b:th elements of ω by $\omega_{a:b}$. The σ-algebra is generated by the cylinder sets $\Gamma_{y_{1:t}} := \{\omega|\omega_{1:t} = y_{1:t}\}$ and a measure is determined by its values on those sets. To simplify notation in the next lemmas we will write $P(\cdot) = \nu(\cdot|\pi)$, meaning that $P(\omega_{1:t}) = \nu(h_t|a_{1:t})$ where $\omega_j = o_j r_j$ and $a_j = \pi(h_{j-1})$. Furthermore, $\nu(\cdot|h_t, \pi) = P(\cdot|h_t)$.

The results from Hutter (2009a) are based on the fact that $Z_t = \frac{Q(\omega_{1:t})}{P(\omega_{1:t})}$ is a martingale sequence if P is the true measure and therefore converges with P probability 1 (Doob, 1953). The crucial question is if the limit is strictly positive or not. The following lemma shows that with P probability 1 we are either in the case where the limit is 0 or in the case where $d(P(\cdot|\omega_{1:t}), Q(\cdot|\omega_{1:t})) \to 0$.

Lemma 38 (Generalized merging of opinions Hutter (2009a)) *For any measures P and Q it holds that $P(\Omega^\circ \cup \bar{\Omega}) = 1$ where*

$$\Omega^\circ := \Big\{\omega : \frac{Q(\omega_{1:t})}{P(\omega_{1:t})} \to 0\Big\} \quad \textit{and} \quad \bar{\Omega} := \Big\{\omega : d(P(\cdot|\omega_{1:t}), Q(\cdot|\omega_{1:t})) \to 0\Big\}$$

The following lemma replaces the property for deterministic environments that either they are consistent indefinitely or the probability of the generated history becomes 0.

Lemma 39 (Merging of environments) *Suppose we are given two environments μ (the true one) and ν and a policy π (defined e.g., by Algorithm 3). Let $P(\cdot) = \mu(\cdot|\pi)$ and $Q(\cdot) = \nu(\cdot|\pi)$. Then with P probability 1 we have that*

$$\lim_{t\to\infty} \frac{Q(\omega_{1:t})}{P(\omega_{1:t})} = 0 \quad \textit{or} \quad \lim_{t\to\infty} |V_\mu^\pi(h_t) - V_\nu^\pi(h_t)| = 0.$$

Proof This follows from a combination of Lemma 38 and Lemma 18. ■

Proof (Theorem 37) Given a policy π, let $P(\cdot) = \mu(\cdot|\pi)$ where $\mu \in \mathcal{M}$ is the true environment and $Q = \nu(\cdot|\pi)$ where $\nu \in \mathcal{M}$. Let the outcome sequence $(o_1r_1), (o_2r_2), ...$ be denoted by ω. It follows from Doob's martingale inequality (Doob, 1953) that for all $z \in (0, 1)$

$$P\Big(\sup_t \frac{Q(\omega_{1:t})}{P(\omega_{1:t})} \geq 1/z\Big) \leq z, \qquad \text{which implies} \quad P\Big(\inf_t \frac{P(\omega_{1:t})}{Q(\omega_{1:t})} \leq z\Big) \leq z.$$

This implies, using a union bound, that the probability of Algorithm 3 ever excluding the true environment is less than $z|\mathcal{M} - 1|$.

The limits $\frac{\nu(h_t|\pi^\circ)}{\mu(h_t|\pi^\circ)}$ converge μ-almost surely as argued before using the martingale convergence theorem. Lemma 39 tells us that any given environment (with probability one) is eventually excluded or is permanently included and merges with the true one under π°. Hence, the remaining environments do merge with the true environment, according to and in the sense of Lemma 39. Lemma 18 tells us that the difference between value functions (for the same policy) of merging environments converges to zero. Since there are finitely many environments and the ones that remain indefinitely in $\mathcal{M}_t$ merge with the true environment under π°, there is for every $\tilde{\varepsilon} > 0$ a T such that for all continuations h of h_T, it holds that

$$|V_{\nu_1}^{\pi^\circ}(h) - V_{\nu_2}^{\pi^\circ}(h)| < \tilde{\varepsilon} \ \forall \nu_1, \nu_2 \in \mathcal{M}_{\ell(h)}.$$

The proof is concluded by Lemma 25 (applied to $\Xi = \mathcal{M}_t$) in the case where the true environment remains indefinitely included which happens with probability $z|\mathcal{M}-1|$. ■

Optimal sample complexity for optimistic agent. We state the below results for $\gamma = 0$ even if some of the results referred to are more general, both for simplicity and because we can only prove that our new agent is optimal for this myopic case and only conjecture that the result extends to $0 < \gamma < 1$. For $\gamma = 0$ we can replace π by a in e.g., V^π because the value then only depends on the immediate action.

Definition 40 (ε-error) *Given $0 \leq \varepsilon < 1$, we define the number of ε-errors for $\gamma = 0$ in history h to be*

$$m(h,\varepsilon) = |\{t \leq \ell(h) \mid V_\mu^{a_t}(h_t) < V_\mu^*(h_t) - \varepsilon\}|$$

where μ is the true environment, $\ell(h)$ is the length of h, a_t is the t:th action of an agent π and $V_\mu^*(h) = \max_a V_\mu^a(h)$. Each such time point t where $V_\mu^{a_t}(h_t) < V_\mu^*(h_t) - \varepsilon$ is called an ε-error.

In Lattimore et al. (2013a), an agent (MERL) that achieves optimal sample complexity for general finite classes of stochastic environments was presented and we provided a high-level description of it in Example 15 in terms of an irrational decision function and a hypothesis-generating function. Here we point out that one can take the hypothesis-generating function of MERL and combine it with an optimistic decision function and still satisfy optimal sample complexity for the case $\gamma = 0$. We conjecture that our optimistic agent also satisfies MERL's bound for $0 < \gamma < 1$, but it is even harder to prove than the difficult analysis of MERL, which was designed to enable the proof. Our resulting optimistic agent is described in Algorithm 4. Lattimore et al. (2013a) proves the matching lower bound $O(\frac{M}{\varepsilon^2(1-\gamma)^3}\log\frac{1}{\delta})$. We conjecture that the optimistic agent just like MERL satisfies an upper bound matching the generic lower up to logarithmic factors for all $\gamma < 1$ and not just for $\gamma = 0$, which we can prove it for.

The advantage of the optimistic agent is that its exploration is not irrationally exploring an option with values in e.g., the interval $[0, 0.3]$ if there is an option with guaranteed value of 0.9. MERL does this because it looks for the maximum discrepancy in values, which is why it is called Maximum Exploration Reinforcement Learning. The agent eliminates all wrong environments regardless if this is useful or not. The exclusion criterion is based on what return is predicted by the remaining environments. If the most optimistic and the most pessimistic differ substantially one of them will turn out to be wrong and the plausibility of it being the truth decreases. When an environment becomes sufficiently implausible it is excluded. The technical difficulty is about both making sure that the truth is with high probability not excluded while also not keeping an environment unnecessarily long which would cause excess exploration. Investigating this particular technical difficulty, while important, is not among the main conceptual issues this article is focused on.

Theorem 41 (Sample complexity for optimistic agent) *Suppose we have a finite class of M (stochastic) environments $\mathcal{M}$. Letting $\alpha = 1 + (4\sqrt{M}-1)^{-1}$ and $\delta_1 = \delta(32(3 + \log_2 1/\varepsilon)M^{3/2})^{-1}$ in Algorithm 4, the number of ε-errors, i.e., time points t such that*

Algorithm 4: Optimistic agent with hypothesis-generation from Lattimore et al. (2013a)

Require: $\varepsilon, \delta_1, \alpha, \mathcal{M} = \{\nu_1, ..., \nu_M\}$
Ensure: $t = 1$, $h = \epsilon$, $\alpha_j = \lceil \alpha^j \rceil$, $n_{\nu,\kappa} := 0 \ \forall \nu \in \mathcal{M}, \kappa \in \mathbb{N}$
  while True **do**
    $(\overline{\nu}, a_t) := \arg\max_{\nu \in \mathcal{M}, a \in \mathcal{A}} V_\nu^a(h)$ # Choosing the optimistic action.
    Take action a_t, receive r_t,o_t # h is not appended until the end of the loop
    $\underline{\nu} := \arg\min_{\nu \in \mathcal{M}} V_\nu^{a_t}(h)$ # Find the pessimistic environment for a_t
    $\Delta = V_{\overline{\nu}}^{a_t}(h) - V_{\underline{\nu}}^{a_t}(h)$ # Difference between optimistic and pessimistic
    if $\Delta > \varepsilon/4$ # If large, one of them is significantly off
    # and we got an effective test
   then
      $\kappa = \max\{k \in \mathbb{N} : \Delta > \varepsilon 2^{k-2}\}$
      $n_{\overline{\nu},\kappa} = n_{\overline{\nu},\kappa} + 1, n_{\underline{\nu},\kappa} = n_{\underline{\nu},\kappa} + 1$
      $X_{\overline{\nu},\kappa}^{n_{\overline{\nu},\kappa}} = V_{\overline{\nu}}^{a_t}(h) - r_t$
      $X_{\underline{\nu},\kappa}^{n_{\underline{\nu},\kappa}} = r_t - V_{\underline{\nu}}^{a_t}(h)$
      if $\exists j, \kappa : n_{\overline{\nu},\kappa} = \alpha_j$ and $\sum_{i=1}^{n_{\overline{\nu},\kappa}} X_{\overline{\nu},\kappa}^i \geq \sqrt{2 n_{\overline{\nu},\kappa} \log \frac{n_{\overline{\nu},\kappa}}{\delta_1}}$ **then**
        $\mathcal{M} = \mathcal{M} \setminus \{\overline{\nu}\}$
      end if
      if $\exists j, \kappa : n_{\underline{\nu},\kappa} = \alpha_j$ and $\sum_{i=1}^{n_{\overline{\nu},\kappa}} X_{\underline{\nu},\kappa}^i \geq \sqrt{2 n_{\underline{\nu},\kappa} \log \frac{n_{\underline{\nu},\kappa}}{\delta_1}}$ **then**
        $\mathcal{M} = \mathcal{M} \setminus \{\underline{\nu}\}$
      end if
      $t := t + 1$, $h := h a_t o_t r_t$
    end if
  end while

$V_\mu^*(h_t) - V_\mu^\pi(h_t) > \varepsilon$ where π is Algorithm 4, resulting from running it on any environment in $\mathcal{M}$ is with probability $1 - \delta$ less than

$$\tilde{\mathcal{O}}(\frac{M}{\varepsilon^2} \log^2 \frac{1}{\delta})$$

where $\tilde{O}$ means O but up to logarithmic factors.

Proof The result follows from the analysis in Lattimore et al. (2013a) and we only provide an overview here. More precisely, the claim follows from the proofs of Lemma 2 and 4 in Lattimore et al. (2013a) which are both based on Azuma's inequality. Lemma 2 proves that the true environment will not be excluded with high probability (we need this to be at least $1 - \delta/2$). Lemma 4 shows that the number of exploration phases will not be larger than $\tilde{O}(\frac{M}{\varepsilon^2} \log^2 \frac{1}{\delta})$ with high probability, at least $1 - \delta/2$. The proof shows that before we reach that many we will with at least that probability have excluded all but the true environment. However, all environments do not have to be excluded and some environments might remain indefinitely by offering just slightly less reward for the optimal action than the true environment. For our agent, unlike MERL, an environment might also remain by differing arbitrarily much on actions that will never optimistically be taken. ■

For a reader that is familiar with MERL we explain why the bound for our agent should naturally be expected to be the same as for the MERL agent for $\gamma = 0$. To ensure that it can be guaranteed that no ε-errors are made during exploitation, MERL checks the maximum distance between environments for any policy and decides based on this if it needs to explore. Our agent, however, will still have this guarantee in the case $\gamma = 0$ and we can, from the analysis of MERL in Lattimore et al. (2013a), conclude that it makes, with probability $1 - \delta$, at most $\tilde{O}(\frac{M}{\varepsilon^2} \log^2 \frac{1}{\delta})$ ε-errors. In fact, for $\gamma = 0$ we only need to know that the maximum difference between any two environments' values under the optimistic action is less than $\varepsilon/2$, to guarantee that the agent does not make an ε-error.

Model-free vs model-based. We will here discuss our two main ways of excluding environments, namely exclusion by accuracy of return predictions (Algorithm 4 and MERL) and plausibility given observations and rewards (Algorithm 3 and BayesExp). Algorithm 4 above is essentially a model-free algorithm since what is used from each environment are two things; a recommended policy and a predicted return (its value in the given environment). Algorithm 4 evaluates the plausibility of an environment based on its predicted return. Hence, for each time step it only needs pairs of policy and return prediction and not complete environments. Such pairs are exactly what is considered in the Learning Classifier Systems (LCS) approach as mentioned in the introduction and as will be discussed in Section 5.2.

We will primarily consider a model-based situation where predictions are made also for future observations. Also, including the observations in the evaluation of one's hypotheses makes better use of available data. However, Hutter (2009b) argues that observations can be extremely complex and that focusing on reward prediction for selecting a model, may still be preferable due its more discriminative nature. We do not here take a definite position.

Lattimore et al. (2013b) studied confidence and concentration in sequence prediction and used exclusion based on a probability ratio, in that case with a weighted average instead of the max in our Algorithm 3. This alternative expression, which is closely related to the one used by Algorithm 3, differing only by a constant factor, can be interpreted as the posterior probability for the hypothesis and hypotheses with low posterior probability are excluded. Lattimore (2014) extended this work to a reinforcement learning algorithm BayesExp that like MERL above switches between phases of exploitation and pure exploration. When the remaining environments are sufficiently concentrated, one can guarantee that an agent does not make a mistake and the agent exploits this. The exploitation in BayesExp is performed by maximizing value for a weighted average, although one can also use optimism and not make a mistake. We deem both behaviors rational based on the definitions in Section 2. However, when the environments are not close enough, BayesExp explores by maximizing Bayesian information gain or by acting greedily with respect to the policy with the largest Hellinger distance to the Bayes mixture. Pure exploration is in this article not deemed rational and we suggest replacing it with acting greedily with respect to the most optimistic environment, i.e., being optimistic. This results again in an always optimistic agent with a criterion for when to exclude environments and we conjecture that this agent satisfies near optimal sample-complexity.

Compact classes. One can extend our results for finite classes to classes that are compact in a suitable topology, e.g., defined by the pseudo-metric

$$\tilde{d}(\nu_1, \nu_2) = \sup_{h,\pi} |V^{\pi}_{\nu_1}(h) - V^{\pi}_{\nu_2}(h)|$$

used by Lattimore et al. (2013a) or a variant based on total variation distance used for the same purpose in Sunehag and Hutter (2012a). If one wants accuracy of $\varepsilon > 0$ one can cover the compact space with finitely many $\tilde{d}$-balls of radius $\varepsilon/2$ and then apply an algorithm for finite classes to the finite class of ball centers to achieve accuracy $\varepsilon/2$. This adds up to accuracy ε for the actual compact class. The number of environments in the finite class is then equal to the number of balls. This number also feature prominently in the theory of supervised learning using reproducing kernel Hilbert spaces (Cucker and Smale, 2002).

Feature Markov decision processes. One can define interesting compact classes of environments using the feature Markov decision process framework (ϕMDP) (Hutter, 2009b; Sunehag and Hutter, 2010). The main idea in this framework is to reduce an environment to an MDP through applying a function ϕ to the history h_t and define a state $s_t = \phi(h_t)$. Given a class of functions of this sort, Sunehag and Hutter (2010) define a class of environments that consists of those that can be exactly represented as an MDP using a function from the class. The class of feature Markov decision processes defined from a finite set of maps is a compact continuously parameterized class. Given a map ϕ from histories to a finite state set $\mathcal{S}$, a sequence of actions, observations, rewards is transformed into a sequence of states $s_1, ..., s_n$ where $s_t = \phi(h_t)$. Defining probability distributions $Pr(or|s, a)$ leads to having defined an environment. In other words, a combination of a map from histories to states with probability parameters stating, for each state-action pair (s, a) the probability of each possible perception $or \in \mathcal{O} \times \mathcal{R}$, is a fully specified environment. Furthermore,

$$Pr(s_{t+1}, r_{t+1}|s_t, a_{t+1}) = \sum_{o_{t+1}r_{t+1}|\phi(h_t a_{t+1} o_{t+1} r_{t+1})=s_{t+1}} Pr(o_{t+1}r_{t+1}|s_t, a_{t+1})$$

and we have, therefore, also defined a Markov Decision Process based on the states defined by the map ϕ. When considering an environment's optimal policy, this means that we can restrict our study to policies that are functions from the states of the environment to actions. Finding the best such *stationary policy* becomes the goal in this setting. Considering a finite class of maps, each map gives us a compact class of environments and we can embed all of them into $\mathbb{R}^d$ for some d. Since a finite union of compact sets is compact, we have defined a compact class. Hence, one can cover the space with finite many balls regardless of how small positive radius one chooses. However, the bounds are linear in the number of balls which can be very large. This is because those bounds are worst case bounds for fully general environments. In the feature MDP case we learn simultaneously about large subsets of environments and one should be able to have bounds that are linear in the size of a maximal state space (see Section 5.2).

Example 42 (Automata) *A special form of maps are those that can be defined by a deterministic function (a table) $\tau(s, a, o, r) = s'$. Maps of this sort have been considered by Mahmud (2010) for the class of Probabilistic-Deterministic Finite Automata.*

7. Countable and Growing Classes

In this section, we extend the agents and analysis from the previous section to arbitrary countable environment classes.

Properties of hypothesis-generating functions. After seeing examples of decision functions and hypothesis generating functions above, we will discuss what properties are desirable in a hypothesis-generating function. We discussed what a decision function should be like in Section 3.1 based on decision-theoretic axioms defining rationality. In the choice of hypothesis-generating functions we focus on what kind of performance can be guaranteed in terms of how many suboptimal decisions will be taken. First, however, we want to restrict our study to hypothesis-generating functions that are following Epicurus' principle that says that one should keep all consistent hypotheses. In the case of deterministic environments it is clear what it means to have a contradiction between a hypothesis and an observation while in the stochastic case it is not. One can typically only say that the data makes a hypothesis unlikely as in Example 14. We say that a hypothesis generating function satisfies Epicurus if the update function is such that it might add new environments in any way while removing environments if a hypothesis becomes implausible (likely to be false) in light of the observations made. Aside from satisfying Epicurus' principle, we design hypothesis generating functions based mainly on wanting few mistakes to be made. For this purpose we first define the term ε-(in)confidence. We are going to formulate the rest of the definitions and results in this section for $\gamma = 0$, while explaining also how the general $0 < \gamma < 1$ works. We choose to formulate the formal results for this case ($\gamma = 0$) to clarify the reasoning and conceptual issues that apply to endless variations of the setting.

Since the true environment is unknown, an agent cannot know if it has made an ε-error or not. However, if one assumes that the true environment is in the class $\mathcal{G}(h_t)$, or more generally that the class contains an environment that is optimistic with respect to the true environment, and if the class is narrow in total variation distance in the sense (of Lemma 25) that the distance between any pair of environments in the class is small, then one can conclude that an error is not made. Since we do not know if this extra assumption holds for $\mathcal{G}(h_t)$, we will use the terms ε-confident and ε-inconfident.

If the value functions in the class $\mathcal{G}(h_t)$ differ in their predicted value by more than $\varepsilon > 0$, then we cannot be sure not to make an ε-error even if we knew that the true environment is in $\mathcal{G}(h_t)$. We call such points ε-inconfidence points.

Definition 43 (ε-(in)confidence) *Given $0 < \varepsilon < 1$, we define the number of ε-inconfidence points in the history h to be*

$$n(h, \varepsilon) := |\{t \leq \ell(h) \mid \max_{\nu_1, \nu_2 \in \mathcal{G}(h_t)} |V_{\nu_1}^{\pi^*} - V_{\nu_2}^{\pi^*}| > \varepsilon\}|$$

where $\pi^* := \arg\max_\pi \max_{\nu \in \mathcal{G}(h_t)} V_\nu^\pi$. In the $\gamma = 0$ case studied here, we can equivalently write $a^* := \arg\max_a \max_{\nu \in \mathcal{G}(h_t)} V_\nu^a$ instead of π^*. The individual time points where $\max_{\nu_1, \nu_2 \in \mathcal{G}(h_t)} |V_{\nu_1}^{\pi^*} - V_{\nu_2}^{\pi^*}| > \varepsilon$ are the points of ε-inconfidence and the other points are the points of ε-confidence.

Hypothesis-generating functions with budget. We suggest defining a hypothesis-generating function from a countable enumerated class $\mathcal{M}$ based on a *budget function* for ε-inconfidence.

The budget function $N : \mathbb{N} \to \mathbb{N}$ is always such that $N(t) \to \infty$ as $t \to \infty$. The idea is simply that when the number of ε-inconfidence points is below budget the next environment is introduced into the class. The intuition is that if the current hypotheses are frequently contradictory, then the agent should resolve these contradictions before adding more. The definition is also mathematically convenient for proving bounds on ε-errors. Besides the budget function we also require a criterion for excluding environments. An exclusion function (criterion) is here a function $\psi(\tilde{\mathcal{M}}, h) = \mathcal{M}'i$ for $\tilde{\mathcal{M}} \subset \mathcal{M}$ and $h \in \mathcal{H}$ such that $\mathcal{M}' \subset \tilde{\mathcal{M}}$. We will use the trivial $\psi(\tilde{\mathcal{M}}, h) = \tilde{\mathcal{M}}$ when the class of environments is guaranteed to asymptotically merge with the truth. The definitions below are slightly complicated by the fact that the hypothesis class $\mathcal{G}(h)$ consists of environments $\tilde{\nu}(\cdot) = \nu(\cdot|h)$ for ν in a subset of $\mathcal{M}$ that can be described as $\{\nu \in \mathcal{M} \mid \nu(\cdot|h) \in \mathcal{G}(h)\}$.

Definition 44 (Hypothesis generation with budget and exclusion function) *The hypothesis-generating function $\mathcal{G}$ with class $\mathcal{M}$, initial class $\mathcal{M}^0 \subset \mathcal{M}$, accuracy $\varepsilon \geq 0$, budget N and exclusion criterion ψ, is defined recursively: First, let $\mathcal{G}(\epsilon) := \mathcal{M}^0$. If $n(h_t, \varepsilon) \geq N(t)$, then*

$$\mathcal{G}(h_t) := \{\nu(\cdot|h_t) \mid \nu \in \psi(\{\nu \in \mathcal{M} \mid \nu(\cdot|h_{t-1}) \in \mathcal{G}(h_{t-1})\}, h_t)\}$$

while if $n(h_t, \varepsilon) < N(t)$, let $\tilde{\nu}$ be the environment in $\mathcal{M}$ with the lowest index that is not in $\bigcup_{i=1}^{t-1}\{\nu \in \mathcal{M} \mid \nu(\cdot|h_i) \in \mathcal{G}(h_i)\}$ (i.e., the next environment to introduce) and let

$$\mathcal{G}(h_t) := \{\nu(\cdot|h_t) \mid \nu \in \{\tilde{\nu} \cup \psi(\{\nu \in \mathcal{M} \mid \nu(\cdot|h_{t-1}) \in \mathcal{G}(h_{t-1})\}, h_t)\}\}.$$

7.1 Error Analysis

We now extend the agents described in Example 11 and Example 12 by removing the demand for the class $\mathcal{M}$ to be finite and analyze the effect on the number of ε-errors made. We still use the optimistic decision function and apply it to finite classes but incrementally add environments from the full class to the finite working class of environments. The resulting agent differs from agents such as the one in Example 15 by (among other things) instead of having exploration phases as part of the decision function, it has a hypothesis-generating function that sometimes adds an environment. This may cause new explorative behavior if it becomes the optimistic hypothesis and it deviates significantly from the other environments. A point to note about our results is that the agent designer chooses the asymptotic error rate but a constant term gets higher for slower rates. This trade-off is due to the fact that if new environments are included at a slower rate, then it takes longer until the right environment is introduced while the error rate afterwards is better. If the agent knew that the true environment had been found, then it could stop introducing more but this is typically impossible to know.

Deterministic environments. We first extend the agent for finite classes of deterministic environments in Example 11 to the countable case. In the finite case with a fixed class, the proof of the finite error bound builds on the fact that every ε-error must be within $\frac{-\log(\varepsilon(1-\gamma))}{1-\gamma}$ time steps before a contradiction and the bound followed immediately because there are at most $|\mathcal{M} - 1|$ contradictions. In the case where environments are being added, errors occur either before the truth is added or within that many time steps before a

contradiction or that many time steps before the addition of a new environment. The addition of a new environment can change the optimistic policy without a contradiction, because the event temporarily breaks time-consistency. Hence, every added environment after the truth has been included can add at most $2\frac{-\log(\varepsilon(1-\gamma))}{1-\gamma}$ ε-errors. In the $\gamma = 0$ case it is only at contradictions and when the truth has not been added that errors occur.

Theorem 45 (Countable deterministic class) *Suppose we have a countable class of deterministic environments $\mathcal{M}$ with a chosen enumeration and containing the true environment. Also suppose we have a hypothesis-generating function $\mathcal{G}$ with a finite initial class $\mathcal{G}(\epsilon) := \mathcal{M}^0 \subset \mathcal{M}$, budget function $N : \mathbb{N} \to \mathbb{N}$, accuracy $\varepsilon = 0$ and exclusion function $\psi(\tilde{\mathcal{M}}, h) := \{\nu \in \tilde{\mathcal{M}} \mid \nu \text{ consistent with } h\}$. π° is defined by combining $\mathcal{G}$ with an optimistic decision function. It follows that*
i) The number of 0-errors $m(h_t, 0)$ is for all t at most $n(h_t, 0) + C$ for some constant $C \geq 0$ (the time steps until the true environment is introduced) dependent on choice of budget function N but not on t.
ii) $\forall i \in \mathbb{N}$ there is $t_i \in \mathbb{N}$ such that $t_i < t_{i+1}$ and $n(h_{t_i}, 0) < N(t_i)$.
Further, if we modify the hypothesis-generating function above by delaying a new environment from being introduced if more than $N(t)$ environments (including the initial class) have been introduced at time t, then
iii) $\forall t: \ n(h_t, 0) < N(t)$
iv) $m(h_t, 0)/t \to 0$ if $N(t)/t \to 0$, i.e., π° satisfies weak asymptotic optimality

In the theorem above, ii) says that we will always see the number of errors fall within the budget $N(t)$ again (except for a constant term) even if it can be temporarily above. This means that we will always introduce more environments and exhaust the class in the limit. The final conclusion (iv) is that π° satisfies weak asymptotic optimality as defined by Lattimore and Hutter (2011a) and previously considered by Orseau (2010) who showed that AIXI does not achieve this for the class of all computable environments. An agent with explicit exploration phases that achieved such weak asymptotic optimality was presented by Lattimore and Hutter (2011a) where it was also showed that for the specific countable class of all computable environments, no agent can achieve strong asymptotic optimality, i.e., convergence to optimal performance without averaging.

Comparing to the previous results on finite deterministic environments, we then assumed that the truth was already in that initial class and, therefore, $C = 0$. Further, one will in that case have at most have $|\mathcal{M} - 1|$ inconfidence points as argued in the proof of Theorem 29. Hence, $m(h_t, 0) \leq n(h_t, 0) + C$ says that we will at most have $|\mathcal{M} - 1|$ errors as stated also by Theorem 29 with $\gamma = 0$. The second part of the conclusion of Theorem 45 does not mean anything for the finite case since it relates to an indefinitely increasing budget and environments being continually added. Therefore, the case with a finite fixed class is more cleanly studied first by itself to then reuse the techniques adapted to the setting of growing classes in this section.

Proof Suppose that at time t, the true environment μ is in $\mathcal{G}(h_t)$. Then, if we do not have a 0-inconfidence point, it follows from optimism that

$$V_\mu^{\pi^\circ}(h_t) = \max_a V_\mu^a(h_t) \tag{10}$$

since all the environments in $\mathcal{G}(h_t)$ agree on the reward for the optimistic action. Hence $m(h_t, 0) \leq n(h_t, 0) + C$ where C is the time the true environment is introduced.

However, we need to show that the truth will be introduced by proving that the class will be exhausted in the limit. If this was not the case, then there is T such that $n(0, h_t) \geq N(t)\ \forall t \geq T$. Since we have 0-inconfidence points exactly when a contradiction is guaranteed, $n(0, h_t)$ is then bounded by the number of environments that have been introduced up to time t if we include the number of environments in the initial class. Hence $n(0, h_t)$ is bounded by a finite number while (by the definition of budget function) $N(t) \to \infty$ which contradicts the assumption. iii) follows because if there are at most $N(t)$ environments, and if the truth has been introduced, then one cannot have had more than $N(t)$ contradictions. iv) follows directly from iii). ■

Stochastic environments. We continue by also performing the extension of the agent in Example 12 from finite to countable classes of stochastic environments. The absolute continuity assumption (Definition 16) is best understood in a Bayesian setting but with multiple priors. That is, the environment class can arise as different mixtures of the environments in a hypothesis class that the true environment is assumed to belong to. An alternative stochastic setting is the one in Example 14 where one does not make this assumption but instead assumes that the true environment is in the class and the agent needs to have an exclusion criterion. In this section no exclusion is necessary but we instead rely on the merging of environments guaranteed by Theorem 17. As for the deterministic setting, one can derive the corresponding finite class result, Theorem 21, from the inequality $m(h_t, \varepsilon) \leq n(h_t, \varepsilon) + C$ but it requires some of the reasoning of its proof.

Theorem 46 (Countable stochastic class) *Suppose we have a enumerated countable class of stochastic environments $\mathcal{M}$ such that the true environment μ is absolutely continuous with respect to every environment in $\mathcal{M}$, a hypothesis-generating function $\mathcal{G}$ with a finite initial class $\mathcal{G}(\epsilon) = \mathcal{M}^0 \subset \mathcal{M}$, a budget function $N : \mathbb{N} \to \mathbb{N}$ and accuracy $\varepsilon > 0$ and exclusion function $\psi(\tilde{\mathcal{M}}, h) := \tilde{\mathcal{M}}$. The agent is defined by combining $\mathcal{G}$ with an optimistic decision function. If for all h, there is $\nu_h \in \mathcal{M}$ that is optimistic in the sense that*

$$\max_a V^a_{\nu_h}(h) \geq \max_a V^a_{\mu}(h),$$

then there is
i) μ-almost surely a $C \geq 0$ such that

$$\forall t\ m(h_t, \varepsilon) \leq n(h_t, \varepsilon) + C$$

ii) μ-almost surely a sequence $t_i \to \infty$ such that $n(h_{t_i}, \varepsilon) < N(t_i)$ and, therefore, any environment in $\mathcal{M}$ is eventually included in $\mathcal{G}(h_t)$ for sufficiently large t.

Proof Suppose we have a finite class Ξ of stochastic environments such that the true environment μ is absolutely continuous with respect to all of them. Suppose that $\varepsilon > 0$ and that π is defined by letting $\mathcal{G}(h_t) = \{\nu(\cdot|h_t) \mid \nu \in \Xi\}$ for all t and letting the decision

function be optimistic. If we act according to π then we will first show that there will μ-almost surely only be finitely many ε-inconfidence points. Furthermore, if Ξ contains an environment that is optimistic relative to μ then only ε -inconfidence points can be ε-errors so there are only finitely many of those.

By Theorem 17 and the finiteness of the class, there is (μ-almost surely) for any $\varepsilon > 0$ and policy π, a $T < \infty$ such that $d(\xi(\cdot|h_t, \pi), \mu(\cdot|h_t, \pi)) < \varepsilon \ \forall \xi \in \Xi \ \forall t \geq T$. We cannot know for sure when the environments in Ξ have merged with the truth (under policy π) in this sense but we do know when the environments have merged with each other. That they will merge with each other follows from the fact that they all merge with μ under π. More precisely, for all $\varepsilon' > 0$ there is $T < \infty$ such that $d(\xi_1(\cdot|h_t, \pi), \xi_2(\cdot|h_t, \pi)) < \varepsilon' \ \forall \xi_1, \xi_2 \in \Xi \ \forall t \geq T$. It follows that then $|V^\pi_{\xi_1}(h_t) - V^\pi_{\xi_2}(h_t)| < \frac{\varepsilon'}{1-\gamma} \ \forall \xi_1, \xi_2 \in \Xi \ \forall t \geq T$ by Lemma 18. Hence, for any $\varepsilon > 0$ there are only finitely many ε-inconfidence points.

Now, let $t_i, i = 1, 2, 3, ...$ be the points where $n(\varepsilon, h_t) < N(t)$, i.e., where new environments are added. $t_i < t_{i+1}$ by definition. One of the aims is to show that $t_i \to \infty$ as $i \to \infty$. Before that we do not know if there is a t_i defined for each i. Suppose that i is such that t_i is defined and suppose that there is no t_{i+1}, i.e., that $n(h_t, \varepsilon) \geq N(t) \ \forall t > t_i$. Let $\Xi := \mathcal{G}(h_{t_i+1})$. Then the argument above shows that there are only finitely many ε-inconfidence points which contradicts the assumption that $n(h_t, \varepsilon) \geq N(t) \ \forall t > t_i$ since $N(t) \to \infty$. Hence t_i is defined for all i and since $t_i < t_{i+1}$, $t_i \to \infty$ as $i \to \infty$.

Finally, ε-errors can only occur at time points before there always is an optimistic environment for μ in $\mathcal{G}(h_t)$, before an environment in the class has merged sufficiently with μ or at points of ε-inconfidence and this proves the claims. ■

Remark 47 (Extensions: $\gamma > 0$, Separable classes) *As in the deterministic case, the difference between the $\gamma = 0$ case and the $0 < \gamma < 1$ case is that ε-errors can then also occur within $\frac{-\log(\varepsilon(1-\gamma))}{1-\gamma}$ time steps before a new environment is introduced, hence the Theorem still holds. Further, one can extend our algorithms for countable classes to separable classes since they can by definition be covered by countably many balls of arbitrarily small radius.*

Discussion and future plans. The agent studied above has the behaviour that after its current class merges it could remain confident for such a long time that its average number of points of inconfidence gets close to zero, but then when a new environments is introduced a finite but potentially very long stretch of inconfidence sets in before we are back to a stretch of confidence. Since we do not have bound on how long the inconfidence will last, we can not set the budget function such as to guarantee convergence to zero for the average number of errors.

If we want to achieve such convergence, extending the agent that excludes implausible stochastic environments is more promising. The reasoning is closer to the deterministic case. In particular if we look at the adaptive k-meteorologist algorithm, when two environments have disagreed sufficiently much m times, one of them is excluded. The number m depends on the desired confidence. In the deterministic case $m = 1$ and the confidence is complete. Having an environment excluded after m disagreements, bounds the amount of inconfidence caused by adding a new environment. If one wants asymptotic optimality in average, the agent also needs to decrease ε when a new environment is introduced. We intend in

the future to pursue the investigation into asymptotic optimality for countable classes of stochastic environments, which together with stochastic laws (Sunehag and Hutter, 2015) and practical implementation constitute important questions not addressed here.

8. Conclusions

We studied sequential decision-making in general reinforcement learning. Our starting point was decision-theoretic axiomatic systems of rational behavior and a framework to define agents within. We wanted to axiomatically exclude agents that are doing things that one clearly should not, before considering achieving good performance guarantees. This is important because if the guarantees are for a relatively short horizon they can sometimes be achieved by highly undesirable strategies. The guarantees only imply that the agent learns well from its experiences.

After introducing two sets of rationality axioms, one for agents with a full horizon and one for agents with a limited horizon that required optimism, we then introduced a framework using hypothesis-generating functions and decision functions to define rational general reinforcement learning agents. Further, we designed optimistic agents within this framework for different kinds of environment classes and proved error bounds and asymptotic properties. This was first done for finite classes and then extended to arbitrary countable classes. Along the way we introduced the concept of deterministic environments defined by combining partial laws and showed that the studied optimistic agents satisfy more desirable, potentially exponentially better, guarantees in such a setting. A further step would be to also apply that strategy in the stochastic setting.

Acknowledgments

This work was supported by ARC grant DP120100950. The first author was affiliated with the Australian National University during most of this work. The authors are also grateful to the helpful reviewers.

References

Y. Abbasi-Yadkori, P. Bartlett, V. Kanade, Y. Seldin, and C. Szepesvári. Online learning in Markov decision processes with adversarially chosen transition probability distributions. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 2508–2516, 2013.

J. Asmuth, L. Li, M. Littman, A. Nouri, and D. Wingate. A Bayesian sampling approach to exploration in reinforcement learning. In *Uncertainty in Artificial Intelligence (UAI)*, pages 19–26, 2009.

P. Auer and R. Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. In *Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems(NIPS'2006)*, pages 49–56, 2006.

E. Baum and I. Durdanovic. Evolution of cooperative problem solving in an artificial economy. *Neural Computation*, 12(12):2743–2775, 2001.

D. Blackwell and L. Dubins. Merging of opinions with increasing information. *The Annals of Mathematical Statistics*, 33(3):882–886, 1962.

R. Casadesus-Masanell, P. Klibanoff, and E. Ozdenoren. Maxmin Expected Utility over Savage Acts with a Set of Priors. *Journal of Economic Theory*, 92(1):35–65, May 2000.

F. Cucker and S. Smale. On the mathematical foundations of learning. *Bulletin of the American Mathematical Society*, 39:1–49, 2002.

B. deFinetti. La prevision: Ses lois logiques, ses sources subjectives. In *Annales de l'Institut Henri Poincare 7*, pages 1–68. Paris, 1937.

J. Diestel. *Sequences and series in Banach spaces*. Springer-Verlag, 1984.

C. Diuk, L. Li, and B. Leffler. The adaptive k-meteorologists problem and its application to structure learning and feature selection in reinforcement learning. In *Proceedings of the 26th International Conference on Machine Learning, ICML 2009*, pages 249–256, 2009.

J. Doob. *Stochastic processes*. Wiley, New York, NY, 1953.

J. Drugowitsch. *Learning Classifier Systems from First Principles: A Probabilistic Reformulation of Learning Classifier Systems from the Perspective of Machine Learning*. Technical report (University of Bath. Dept. of Computer Science). University of Bath, Department of Computer Science, 2007.

I. Gilboa and D. Schmeidler. Maxmin expected utility with non-unique prior. *Journal of Mathematical Economics*, 18(2):141–153, April 1989.

J.H. Holland. Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine learning: An artificial intelligence approach*, volume 2, chapter 20, pages 593–623. Morgan Kaufmann, Los Altos, CA, 1986.

M. Hutter. Implementierung eines Klassifizierungs-Systems. Master's thesis, Theoretische Informatik, TU München, 1991.

M. Hutter. *Universal Articial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005.

M. Hutter. Discrete MDL predicts in total variation. In *Advances in Neural Information Processing Systems 22: (NIPS'2009)*, pages 817–825, 2009a.

M. Hutter. Feature reinforcement learning: Part I: Unstructured MDPs. *Journal of Artificial General Intelligence*, 1:3–24, 2009b.

E. Kreyszig. *Introductory Functional Analysis With Applications*. Wiley, 1989.

I. Kwee, M. Hutter, and J. Schmidhuber. Market-based reinforcement learning in partially observable worlds. *Proceedings of the International Conference on Artificial Neural Networks (ICANN-2001)*, pages 865–873, 2001.

T. Lattimore. *Theory of General Reinforcement Learning*. PhD thesis, Australian National University, 2014.

T. Lattimore and M. Hutter. Asymptotically optimal agents. In *Proc. of Algorithmic Learning Theory, (ALT'2011)*, volume 6925 of *Lecture Notes in Computer Science*, pages 368–382. Springer, 2011a.

T. Lattimore and M. Hutter. Time consistent discounting. In *Proc. 22nd International Conf. on Algorithmic Learning Theory (ALT'11)*, volume 6925 of *LNAI*, pages 383–397, Espoo, Finland, 2011b. Springer, Berlin.

T. Lattimore and M. Hutter. PAC Bounds for Discounted MDPs. In Nader H. Bshouty, Gilles Stoltz, Nicolas Vayatis, and Thomas Zeugmann, editors, *ALT*, volume 7568 of *Lecture Notes in Computer Science*, pages 320–334. Springer, 2012. ISBN 978-3-642-34105-2.

T. Lattimore, M. Hutter, and P. Sunehag. The sample-complexity of general reinforcement learning. *Journal of Machine Learning Research, W&CP: ICML*, 28(3):28–36, 2013a.

T. Lattimore, M. Hutter, and P. Sunehag. Concentration and confidence for discrete Bayesian sequence predictors. In *Proc. 24th International Conf. on Algorithmic Learning Theory (ALT'13)*, volume 8139 of *LNAI*, pages 324–338, Singapore, 2013b. Springer, Berlin.

J. Leike and M. Hutter. Bad Universal Priors and Notions of Optimality. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015*, pages 1244–1259, 2015.

M. Li and P. Vitani. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2008.

M. M. Mahmud. Constructing states for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 727–734, 2010.

O.-A. Maillard, R. Munos, and D. Ryabko. Selecting the state-representation in reinforcement learning. In *Advances in Neural Information Processing Systems 24 (NIPS'2011)*, pages 2627–2635, 2011.

O.-A. Maillard, P. Nguyen, R. Ortner, and D. Ryabko. Optimal regret bounds for selecting the state representation in reinforcement learning. In *International Conference on Machine Learning (ICML'2013)*, 2013.

M. Müller. Stationary algorithmic probability. *Theor. Comput. Sci.*, 411(1):113–130, 2010.

L. Naricia and E. Beckenstein. The Hahn-Banach theorem: the life and times. *Topology and its Applications*, 77(2):193–211, 1997.

G. Neu, A. György, C. Szepesvári, and A. Antos. Online Markov decision processes under bandit feedback. In *Advances in Neural Information Processing Systems 23: 2010.*, pages 1804–1812, 2010.

J. Neumann and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

P. Nguyen, O.-A. Maillard, D. Ryabko, and Ronald Ortner. Competing with an infinite set of models in reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS'2013).*, 2013.

L. Orseau. Optimality issues of universal greedy agents with static priors. In *Proc. of Algorithmic Learning Theory, 21st International Conference, (ALT'2010)*, volume 6331 of *Lecture Notes in Computer Science*, pages 345–359. Springer, 2010.

F. Ramsey. Truth and probability. In R. B. Braithwaite, editor, *The Foundations of Mathematics and other Logical Essays*, chapter 7, pages 156–198. Brace & Co., 1931.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, NJ, 3^{nd} edition, 2010.

D. Ryabko and M. Hutter. On the possibility of learning in reactive environments with arbitrary dependence. *Theoretical Computer Science*, 405(3):274–284, 2008.

L. Savage. *The Foundations of Statistics*. Wiley, New York, 1954.

A. L. Strehl, L. Li, and M. L. Littman. Reinforcement learning in finite MDPs: PAC analysis. *J. of Machine Learning Research*, 10:2413–2444, 2009.

P. Sunehag and M. Hutter. Consistency of feature Markov processes. In *Proc. 21st International Conf. on Algorithmic Learning Theory (ALT'10)*, volume 6331 of *LNAI*, pages 360–374, Canberra, 2010. Springer, Berlin.

P. Sunehag and M. Hutter. Axioms for rational reinforcement learning. In *Algorithmic Learning Theory, (ALT'2011)*, volume 6925 of *Lecture Notes in Computer Science*, pages 338–352. Springer, 2011.

P. Sunehag and M. Hutter. Optimistic agents are asymptotically optimal. In *Proc. 25th Australasian Joint Conference on Artificial Intelligence (AusAI'12)*, volume 7691 of *LNAI*, pages 15–26, Sydney, Australia, 2012a. Springer.

P. Sunehag and M. Hutter. Optimistic AIXI. In *Proc. 5th Conf. on Artificial General Intelligence (AGI'12)*, volume 7716 of *LNAI*, pages 312–321. Springer, Heidelberg, 2012b.

P. Sunehag and M. Hutter. Learning agents with evolving hypothesis classes. In *Proc. 6th Conf. on Artificial General Intelligence (AGI'13)*, volume 7999 of *LNAI*, pages 150–159. Springer, Heidelberg, 2013.

P. Sunehag and M. Hutter. A dual process theory of optimistic cognition. In *Proc. 36th Annual Meeting of the Cognitive Science Society (CogSci'14)*, pages 2949–2954, 2014.

P. Sunehag and M. Hutter. Using Localization and Factorization to Reduce the Complexity of Reinforcement Learning In *Proc. 8th Conf. on Artificial General Intelligence (AGI'15)*, volume 9205 of *LNAI*, pages 177–186. Springer, Heidelberg, 2015.

I. Szita and A. Lörincz. The many faces of optimism: a unifying approach. In *Proceedings of the* 20^{th} *International Conference on Machine Learning*, pages 1048–1055, 2008.

J. Veness, K. S. Ng, M. Hutter, W. Uther, and D. Silver. A Monte-Carlo AIXI approximation. *Journal of Artificial Intelligence Research*, 40(1):95–142, 2011.

P. Walley. Towards a unified theory of imprecise probability. *Int. J. Approx. Reasoning*, pages 125–148, 2000.

F. Willems, Y. Shtarkov, and T. Tjalkens. The context tree weighting method: Basic properties. *IEEE Transactions on Information Theory*, 41:653–664, 1995.

Appendix A. Asymptotic Optimality of the Liberal Agent

This section contains a proof of the asymptotic optimality Theorem 27 for the liberal version of Algorithm 1 called Algorithm 1', which can (but does not have to) leave the inner loop even when $\nu^* \in \mathcal{M}_{t-1}$. We are also more explicit and provide some intuition behind the subtleties hidden in the conservative case. The notation used here is somewhat different to the main paper. The fact that environments and policies are deterministic is heavily exploited in notation and proof technique.

Policies versus action sequences. A deterministic policy $\pi : \mathcal{H} \to \mathcal{A}$ in some fixed deterministic environment $\nu : \mathcal{H} \times \mathcal{A} \to \mathcal{O} \times \mathcal{R}$ induces a unique history $h^{\pi,\nu}$, and in particular an action an sequence $a_{1:\infty}$. Conversely, an action sequence $a_{1:\infty}$ defines a policy in a *fixed* environment ν. Given ν, a policy and an action sequence are therefore equivalent. But a policy applied to multiple environments is more than just an action sequence. More on this later. For now we only consider action sequences $a_{1:\infty}$ rather than policies.

Definitions. Let

$$\begin{aligned}
\mathcal{M}_\infty &= \text{finite class of environments} \\
r_t^\nu(a_{1:t}) &= \text{reward at time } t \text{ when performing actions } a_{1:t} \text{ in environment } \nu \\
V_\nu^{a_{t:\infty}}(a_{<t}) &= \textstyle\sum_{k=t}^\infty r_k^\nu(a_{1:k})\gamma^{k-t} = \text{value of } \nu \text{ and } a_{1:\infty} \text{ from time } t \text{ on} \\
V_*^{a_{t:\infty}}(a_{<t}) &= \max_{\nu\in\mathcal{M}_\infty} V_\nu^{a_{t:\infty}}(a_{<t}) = \text{optimistic value from time } t \text{ on} \\
a^*_{1:\infty} \in \mathcal{A}^*_{1:\infty} &= \{\arg\max_{a_{1:\infty}} V_*^{a_{1:\infty}}(\epsilon)\} = \text{set of optimistic action sequences} \\
h_t^\circ &= h_t^{\pi^\circ,\mu} = \dot{a}_1\dot{o}_1\dot{r}_1 ... \dot{a}_t\dot{o}_t\dot{r}_t = \text{actually realized history} \\
&\quad \text{by Algorithm } \pi^\circ \text{ in true environment } \mu \\
&\quad \text{generated via } \mu(h_{t-1}^\circ, \dot{a}_t) = \dot{o}_t\dot{r}_t \text{ and } \pi^\circ(h_{t-1}^\circ) = \dot{a}_t
\end{aligned}$$

Consistency. There is a finite initial phase during which environments ν can become inconsistent with h_t° in the sense of $h_t^{\pi^\circ,\nu} \neq h_t^\circ$. Algorithm 1 eliminates environments as soon as they become known to be inconsistent. Since here we are interested in asymptotic optimality only, we can ignore this finite initial phase $1, ..., T-1$ and shift time T back to 1. This simplifies notation considerable. We hence assume that *all* environments in $\mathcal{M}_\infty$ are from the outset and forever consistent, i.e., $h_\infty^{\pi^\circ,\nu} = h_\infty^\circ \ \forall \nu \in \mathcal{M}_\infty$. This implies that

$$\dot{r}_t \ = \ r_t^\nu(\dot{a}_{1:t}) \ \text{ is independent of } \nu \in \mathcal{M}_\infty \text{ for all } t \quad (\infty\text{-consistency}) \tag{11}$$

It does *not* imply that all environments in $\mathcal{M}_\infty$ are the same, they only *look* the same on the one chosen action path $\dot{a}_{1:\infty}$, but different actions, e.g., $\tilde{a}_t$ =*look-left* instead of $\dot{a}_t$ =*look-right* could reveal that ν differs from μ, and $\tilde{a}_t$ =*go-left* instead of $\dot{a}_t$ =*go-right* can probe completely different futures. This is relevant and complicates analysis and actually foils many naively plausible conjectures, since an action $\dot{a}_t$ is only optimal if alternative actions are not better, and this depends on how the environment looks off the trodden path, and there the environments in $\mathcal{M}_\infty$ can differ.

Optimistic liberal algorithm $\boldsymbol{\pi^\circ}$. At time t, given $\dot{a}_{<t}$, Algorithm π° chooses action $\dot{a}_t$ optimistically, i.e., among

$$\dot{a}_t \in \{\arg\max_{a_t} \max_{a_{t+1:\infty}} V_*^{a_{t:\infty}}(\dot{a}_{<t})\} \tag{12}$$

More precisely, we define Algorithm 1' properly with using $\mathcal{M}_{t-1}$ at time t generating action sequence $\dot{a}_{1:\infty}$. After $t > T$, we can use $\mathcal{M}_\infty = \mathcal{M}_{t-1}$, i.e., (12) is equivalent to Algorithm 1' for $t > T$. Now we shift back $T \rightsquigarrow 1$, and (12), which uses $\mathcal{M}_\infty$, is a correct formalization of Algorithm 1'. Note that $\mathcal{M}_\infty$ depends on the choice of $\dot{a}_{1:\infty}$ the algorithm actually makes in case of ambiguities. From now on $\dot{a}_{1:\infty}$ will be a single fixed sequence, chosen by some particular deterministic optimistic algorithm.

Lemma 48 (Optimistic actions) $\dot{a}_{1:\infty} \in \mathcal{A}^*_{1:\infty}$ *i.e.,* $V_*^{\dot{a}_{1:\infty}}(\epsilon) = \max_{a_{1:\infty}} V_*^{a_{1:\infty}}(\epsilon)$.

Proof For $|\mathcal{M}_\infty| = 1$, this follows from the well-known fact in planning that optimal action trees lead to optimal policies and vice versa (under time-consistency (Lattimore and Hutter, 2011b)). For general $|\mathcal{M}_\infty| \geq 1$, ∞-consistency (11) is crucial. Using the value recursion

$$\begin{aligned}
V_\nu^{a_{1:\infty}}(\epsilon) &= \sum_{k=1}^{t-1} r_k^\nu(a_{1:k})\,\gamma^{k-1} + \gamma^t V_\nu^{a_{t:\infty}}(a_{<t}), \quad \text{we get:} \\
\gamma^t \max_{a_{t:\infty}} V_*^{a_{t:\infty}}(\dot{a}_{<t}) &= \max_{a_{t:\infty}} \max_{\nu\in\mathcal{M}_\infty} \Big[V_\nu^{\dot{a}_{<t}a_{t:\infty}}(\epsilon) - \underbrace{\textstyle\sum_{k=1}^{t-1} \overbrace{r_k^\nu(\dot{a}_{1:k})}^{=\dot{r}_k} \gamma^{k-1}}_{\text{independent } \nu \text{ and } a_{t:\infty}} \Big] \\
&= \max_{a_{t:\infty}} V_*^{\dot{a}_{<t}a_{t:\infty}}(\epsilon) - \text{const.}
\end{aligned}$$

Replacing $\max_{a_t}$ by $\arg\max_{a_t}$ we get

$$\arg\max_{a_t} \max_{a_{t+1:\infty}} V_*^{a_{t:\infty}}(\dot{a}_{<t}) \;=\; \arg\max_{a_t} \max_{a_{t+1:\infty}} V_*^{\dot{a}_{<t}a_{t:\infty}}(\epsilon) \tag{13}$$

We can define the set of optimistic action sequences $\mathcal{A}^*_{1:\infty} = \{\arg\max_{a_{1:\infty}} V_*^{a_{1:\infty}}(\epsilon)\}$ recursively as

$$\begin{aligned}
\mathcal{A}^*_{1:t} &:= \{\arg\max_{a_{1:t}} \max_{a_{t+1:\infty}} V_*^{a_{1:\infty}}(\epsilon))\} \\
&= \{(a^*_{<t}, \arg\max_{a_t} \max_{a_{t+1:\infty}} V_*^{a^*_{<t}a_{t:\infty}}(\epsilon)) : a^*_{<t} \in \mathcal{A}^*_{<t}\}, \\
\mathcal{A}^*_{1:\infty} &= \{a_{1:\infty} : a_{1:t} \in \mathcal{A}^*_{1:t}\, \forall t\}
\end{aligned}$$

This shows that any sequence $\tilde{a}_{1:\infty}$ that satisfies the recursion

$$\tilde{a}_t \in \{\arg\max_{a_t} \max_{a_{t+1:\infty}} V_*^{\tilde{a}_{<t}a_{t:\infty}}(\epsilon))\} \tag{14}$$

is in $\mathcal{A}^*_{1:\infty}$. Plugging (13) into (12) shows that $\tilde{a}_{1:\infty} = \dot{a}_{1:\infty}$ satisfies recursion (14), hence $\dot{a}_{1:\infty} \in \mathcal{A}^*_{1:\infty}$. ■

Lemma 49 (Optimism is optimal) $V_\mu^{\dot{a}_{1:\infty}}(\epsilon) = \max_{a_{1:\infty}} V_\mu^{a_{1:\infty}}(\epsilon)$.

Note that by construction and Lemma 48, $\dot{a}_{1:\infty}$ maximizes the (known) optimistic value $V_*^{a_{1:\infty}}$ and by Lemma 49 also the (unknown) true value $V_\mu^{a_{1:\infty}}$; a consequence of the strong asymptotic consistency condition (11). Also note that $V_\mu^{\dot{a}_{1:\infty}} = V_*^{\dot{a}_{1:\infty}}$ but $V_\mu^{a_{1:\infty}} \neq V_*^{a_{1:\infty}}$ for $a_{1:\infty} \neq \dot{a}_{1:\infty}$ is possible and common.

Proof The $\leq$ direction is trivial (since maximization is over all action sequences. For limited policy spaces $\Pi \neq \Pi^{all}$ this may no longer be true). The following chain of (in)equalities proves the $\geq$ direction

$$\max_{a_{1:\infty}} V_\mu^{a_{1:\infty}}(\epsilon) \;\leq\; \max_{a_{1:\infty}} V_*^{a_{1:\infty}}(\epsilon) \;=\; V_*^{\dot{a}_{1:\infty}}(\epsilon) \;=\; \max_{\nu\in\mathcal{M}_\infty} \sum_{k=1}^{\infty} r_k^\nu(\dot{a}_{1:k})\gamma^{k-1}$$
$$= \max_{\nu\in\mathcal{M}_\infty} \sum_{k=1}^{\infty} \underbrace{\dot{r}_k\gamma^{k-1}}_{\text{indep.}\nu} \;=\; \sum_{k=1}^{\infty} \dot{r}_k\gamma^{k-1} \;=\; \sum_{k=1}^{\infty} r_k^\mu(\dot{a}_{1:k})\gamma^{k-1} \;=\; V_\mu^{\dot{a}_{1:\infty}}(\epsilon)$$

where we used in order: definition, Lemma 48, definition, consistency of $\nu \in \mathcal{M}_\infty$, independence of ν, $\mu \in \mathcal{M}_\infty$ and consistency again, and definition. ■

Proof of Theorem 27 for liberal Algorithm 1.

As mentioned, for a fixed deterministic environment ν, policies and action sequences are interchangeable. In particular $\max_\pi V_\nu^\pi(\epsilon) = \max_{a_{1:\infty}} V_\nu^{a_{1:\infty}}(\epsilon)$. This is no longer true for V_*: There are π such that for all $a_{1:\infty}$, $V_*^\pi \neq V_*^{a_{1:\infty}}$, since π may depend on ν but $a_{1:\infty}$ not. This causes us no problems, since still $\max_\pi V_*^\pi = \max_{a_{1:\infty}} V_*^{a_{1:\infty}}$, since

$$\max_\pi \max_\nu V_\nu^\pi(\epsilon) \;=\; \max_\nu \max_\pi V_\nu^\pi(\epsilon) \;=\; \max_\nu \max_{a_{1:\infty}} V_\nu^{a_{1:\infty}}(\epsilon) \;=\; \max_{a_{1:\infty}} \max_\nu V_\nu^{a_{1:\infty}}(\epsilon)$$

Similar (non)equalities hold for $V(h_t)$. Hence Lemmas 48 and 49 imply $V_*^{\pi^\circ} = \max_\pi V_*^\pi$ and $V_\mu^{\pi^\circ} = \max_\pi V_\mu^\pi$.

Now if we undo the shift $T \rightsquigarrow 1$, actually shift $T \rightsquigarrow t$, Lemma 49 implies $V_\mu^{\pi^\circ}(h_t^\circ) = \max_\pi V_\mu^\pi(h_t^\circ)$ for all $t \geq T$. This is just Theorem 1 for the liberal algorithm. ■

Appendix B. Countable Sets of Events

Instead of a finite set of possible outcomes, we will in this section assume a countable set. We suppose that the set of bets is a vector space of sequences $x_k, k = 0, 1, 2, ...$ where we use point-wise addition and multiplication with a scalar. We will define a space by choosing a norm and let the space consist of the sequences that have finite norm as is common in Banach space theory. If the norm makes the space complete it is called a Banach sequence space (Diestel, 1984). Interesting examples are ℓ^∞ of bounded sequences with the maximum norm

$\|(\alpha_k)\|_\infty = \max |\alpha_k|$, c_0 of sequence that converges to 0 equipped with the same maximum norm and ℓ^p which for $1 \leq p < \infty$ is defined by the norm

$$\|(\alpha_k)\|_p = (\sum |\alpha_k|^p)^{1/p}.$$

For all of these spaces we can consider weighted versions ($w_k > 0$) where

$$\|(\alpha_k)\|_{p,w_k} = \|(\alpha_k w_k)\|_p.$$

This means that $\alpha \in \ell^p(w)$ iff $(\alpha_k w_k) \in \ell^p$, e.g., $\alpha \in \ell^\infty(w)$ iff $\sup_k |\alpha_k w_k| < \infty$. Given a Banach (sequence) space X we use X' to denote the dual space that consists of all continuous linear functionals $f : X \to \mathbb{R}$. It is well known that a linear functional on a Banach space is continuous if and only if it is bounded, i.e that there is $C < \infty$ such that $\frac{|f(x)|}{\|x\|} \leq C \; \forall x \in X$. Equipping X' with the norm $\|f\| = \sup \frac{|f(x)|}{\|x\|}$ makes it into a Banach space. Some examples are $(\ell^1)' = \ell^\infty$, $c_0' = \ell^1$ and for $1 < p < \infty$ we have that $(\ell^p)' = \ell^q$ where $1/p + 1/q = 1$. These identifications are all based on formulas of the form

$$f(x) = \sum x_i p_i$$

where the dual space is the space that (p_i) must lie in to make the functional both well defined and bounded. It is clear that $\ell^1 \subset (\ell^\infty)'$ but $(\ell^\infty)'$ also contains "stranger" objects.

The existence of these other objects can be deduced from the Hahn-Banach theorem (see e.g., Kreyszig (1989) or Naricia and Beckenstein (1997)) that says that if we have a linear function defined on a subspace $Y \in X$ and if it is bounded on Y then there is an extension to a bounded linear functional on X. If Y is dense in X the extension is unique but in general it is not. One can use this Theorem by first looking at the subspace of all sequences in ℓ^∞ that converge and let $f(\alpha) = \lim_{k\to\infty} \alpha_k$. The Hahn-Banach theorem guarantees the existence of extensions to bounded linear functionals that are defined on all of ℓ^∞. These are called Banach limits. The space $(\ell^\infty)'$ can be identified with the so called ba space of bounded and finitely additive measures with the variation norm $\|\nu\| = |\nu|(A)$ where A is the underlying set. Note that ℓ^1 can be identified with the smaller space of countably additive bounded measures with the same norm. The Hahn-Banach Theorem has several equivalent forms. One of these identifies the hyper-planes with the bounded linear functionals (Naricia and Beckenstein, 1997).

Definition 50 (Rationality (countable case)) *Given a Banach sequence space X of bets, we say that the decision maker (subset Z of X defining acceptable bets and $\tilde{Z}$ the rejectable bets) is rational if*

1. *Every bet $x \in X$ is either acceptable or rejectable or both*
2. *x is acceptable if and only if $-x$ is rejectable.*
3. *$x, y \in Z$, $\lambda, \gamma \geq 0$ then $\lambda x + \gamma y \in Z$*
4. *If $x_k > 0 \; \forall k$ then x is acceptable and not rejectable*

In the case of a finite dimensional space X, the above definition reduces to Definition 8.

Theorem 51 (Linear separation) *Suppose that we have a space of bets X that is a Banach sequence space. Given a rational decision maker there is a positive continuous linear functional $f : X \to \mathbb{R}$ such that*

$$\{x \mid f(x) > 0\} \subseteq Z \subseteq \{x \mid f(x) \geq 0\}. \tag{15}$$

Proof The third property tells us that Z and $-Z$ are convex cones. The second and fourth property tells us that $Z \neq X$. Suppose that there is a point x that lies in both the interior of Z and of $-Z$. Then the same is true for $-x$ according to the second property and for the origin. That a ball around the origin lies in Z means that $Z = X$ which is not true. Thus the interiors of Z and $-Z$ are disjoint open convex sets and can, therefore, be separated by a hyperplane (according to the Hahn-Banach theorem) which goes through the origin (since according to the second and fourth property the origin is both acceptable and rejectable). The first two properties tell us that $Z \cup -Z = X$. Given a separating hyperplane (between the interiors of Z and $-Z$), Z must contain everything on one side. This means that Z is a half space whose boundary is a hyperplane that goes through the origin and the closure $\bar{Z}$ of Z is a closed half space and can be written as $\{x \mid f(x) \geq 0\}$ for some $f \in X'$. The fourth property tells us that f is positive. ■

Corollary 52 (Additivity) *1. If $X = c_0$ then a rational decision maker is described by a countably additive (probability) measure.*
2. If $X = \ell^\infty$ then a rational decision maker is described by a finitely additive (probability) measure.

It seems from Corollary 52 that we pay the price of losing countable additivity for expanding the space of bets from c_0 to ℓ^∞ but we can expand the space even more by looking at $c_0(w)$ where $w_k \to 0$ which contains ℓ^∞ and X' is then $\ell^1((1/w_k))$. This means that we get countable additivity back but we instead have a restriction on how fast the probabilities p_k must tend to 0. Note that a bounded linear functional on c_0 can always be extended to a bounded linear functional on ℓ^∞ by the formula $f(x) = \sum p_i x_i$ but that is not the only extension. Note also that every bounded linear functional on ℓ^∞ can be restricted to c_0 and there be represented as $f(x) = \sum p_i x_i$. Therefore, a rational decision maker for ℓ^∞-bets has probabilistic beliefs (unless $p_i = 0\ \forall i$), though it might also take asymptotic behavior of a bet into account. For example the decision maker that makes decisions based on asymptotic averages $\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} x_i$ when they exist. This strategy can be extended to all of ℓ^∞ and is then called a Banach limit. The following proposition will help us decide which decision maker on ℓ^∞ is endowed with countably additive probabilities.

Proposition 53 *Suppose that $f \in (\ell^\infty)'$. For any $x \in \ell^\infty$, let $x_i^j = x_i$ if $i \leq j$ and $x_i^j = 0$ otherwise. If for any x,*

$$\lim_{j\to\infty} f(x^j) = f(x),$$

then f can be written as $f(x) = \sum p_i x_i$ where $p_i \geq 0$ and $\sum_{i=1}^{\infty} p_i < \infty$.

Proof The restriction of f to c_0 gives us numbers $p_i \geq 0$ such that $\sum_{i=1}^{\infty} p_i < \infty$ and $f(x) = \sum p_i x_i$ for $x \in c_0$. This means that $f(x^j) = \sum_{i=1}^{j} p_i x_i$ for any $x \in \ell^\infty$ and $j < \infty$. Thus $\lim_{j\to\infty} f(x^j) = \sum_{i=1}^{\infty} p_i x_i$. ■

Definition 54 (Monotone decisions) *We define the concept of a* monotone decision maker in the following way. Suppose that for every $x \in \ell^\infty$ there is $N < \infty$ such that the decision is the same for all (as defined above) $x^j,\ j \geq N$ as for x. Then we say that the decision maker is monotone.

Example 55 *Let $f \in \ell^\infty$ be such that if $\lim \alpha_k \to L$ then $f(\alpha) = L$ (i.e., f is a Banach limit). Furthermore define a rational decision maker by letting the set of acceptable bets be $Z = \{x \mid f(x) \geq 0\}$. Then $f(x^j) = 0$ (where we use notation from Proposition 53) for all $j < \infty$ and regardless of which x we define x^j from. Therefore, all sequences that are eventually zero are acceptable bets. This means that this decision maker is not monotone since there are bets that are not acceptable.*

Theorem 56 (Monotone rationality) *Given a monotone rational decision maker for ℓ^∞ bets, there are $p_i \geq 0$ such that $\sum p_i < \infty$ and*

$$\{x \mid \sum x_i p_i > 0\} \subseteq Z \subset \{x \mid \sum x_i p_i \geq 0\}. \tag{16}$$

Proof According to Theorem 51 there is $f \in (\ell^\infty)'$ such that (the closure of Z) $\bar{Z} = \{x|\ f(x) \geq 0\}$. Let $p_i \geq 0$ be such that $\sum p_i < \infty$ and such that $f(x) = \sum x_i p_i$ for $x \in c_0$. Remember that x^j (notation as in Proposition 53) is always in c_0. Suppose that there is x such that x is accepted but $\sum x_i p_i < 0$. This violate monotonicity since there exists $N < \infty$ such that $\sum_{i=1}^{n} x_i p_i < 0$ for all $n \geq N$ and, therefore, x^j is not accepted for $j \geq N$ but x is accepted. We conclude that if x is accepted then $\sum p_i x_i \geq 0$ and if $\sum p_i x_i > 0$ then x is accepted. ■

Appendix C. List of important notation

t	generic time point
T	special time point
$\mathcal{A}, \mathcal{O}, \mathcal{R}$	action/observation/reward sets
h_t	$= a_1 o_1 r_1 ... a_t o_t r_t$ = (action,observation,reward) history
$h_0 = \epsilon$	empty history/string
$\varepsilon \geq 0$	accuracy
δ	probability/confidence
$0 \leq \gamma < 1$	discount factor
$\mathcal{O}_j$	set for the j:th feature
$\vec{x} = (x_i) \in \mathcal{O} = \times_{j=1}^{m} \mathcal{O}_j$	feature vector in Section 5

$\perp$	not predicted feature
$\mathcal{O}_\perp = \times_{j=1}^m (\mathcal{O}_j \cup \{\perp\})$	observation set enhanced by $\perp$
$\pi : \mathcal{H} \to \mathcal{A}$	generic policy $\pi \in \Pi$
$\tilde{\pi}$	some specific policy π
π°	optimistic policy actually followed.
(π_t^*, ν_t^*)	optimistic (policy,environment) (used only) at time t
$V_\nu^\pi(h_t)$	future value of π interacting with ν given h_t
$\mathcal{M}, \tilde{\mathcal{M}}, \hat{\mathcal{M}}$	finite or countable class of environments
$\mathcal{M}^0$	initial class of environments
$m(h, \varepsilon)$	number of ε-errors during h
$n(h, \varepsilon)$	number of ε-inconfidence points
Ξ	finite class of dominant environments
$\nu \in \mathcal{M}$	generic environment
$\xi \in \Xi$	dominant environment
μ	true environment
$\mathcal{T}$	finite class of laws
$\tau \in \mathcal{T}$	generic law
$q_1(\tau, h, a)$	features not predicted by τ in context h, a
$q_2(\tau, h, a)$	features predicted by τ in context h, a
$\mathcal{M}(\mathcal{T})$	environments generated by deterministic laws
$\Xi(\mathcal{T})$	environments generated by stochastic laws
$\bar{\mathcal{M}}(P, \mathcal{T})$	semi-deterministic environments from background and laws
ω	elementary random outcome from some sample space
ω_t	$= o_t r_t =$ perception at time t
$x = (x_i)$	bet in Section 2
$y = (y_i)$	bet in Section 2
$p = (p_i)$	probability vector
f	decision function
$\mathcal{G}$	hypothesis-generating function

Journal of Machine Learning Research 16 (2015) 1391-1436 Submitted 4/13; Revised 8/14; Published 8/15

The Algebraic Combinatorial Approach for Low-Rank Matrix Completion

Franz J. Király F.KIRALY@UCL.AC.UK
Department of Statistical Science
University College London
London WC1E 6BT, United Kingdom
and
Mathematisches Forschungsinstitut Oberwolfach
77709 Oberwolfach-Walke, Germany

Louis Theran LOUIS.THERAN@AALTO.FI
*Aalto Science Institute and Department of Information and Computer Science**
Aalto University
02150 Espoo, Finland

Ryota Tomioka TOMIOKA@TTIC.EDU
Toyota Technological Institute at Chicago
6045 S. Kenwood Ave.
Chicago, Illinois 60637, USA

Editor: Gábor Lugosi

Abstract

We present a novel algebraic combinatorial view on low-rank matrix completion based on studying relations between a few entries with tools from algebraic geometry and matroid theory. The intrinsic locality of the approach allows for the treatment of single entries in a closed theoretical and practical framework. More specifically, apart from introducing an algebraic combinatorial theory of low-rank matrix completion, we present probability-one algorithms to decide whether a particular entry of the matrix can be completed. We also describe methods to complete that entry from a few others, and to estimate the error which is incurred by any method completing that entry. Furthermore, we show how known results on matrix completion and their sampling assumptions can be related to our new perspective and interpreted in terms of a completability phase transition.[1]

Keywords: Low-rank matrix completion, entry-wise completion, matrix reconstruction, algebraic combinatorics

∗. Author's current address. Research was carried out while the author was at Freie Universität Berlin and supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 247029-SDModels

1. This paper is the much condensed, final version of (Király et al., 2013). For convenience, we have included references which have appeared since.

1. Introduction

Matrix completion is the task to reconstruct (to "complete") matrices, given a subset of entries at known positions. It occurs naturally in many practically relevant problems, such as missing feature imputation, multi-task learning (Argyriou et al., 2008), transductive learning (Goldberg et al., 2010), or collaborative filtering and link prediction (Srebro et al., 2005; Acar et al., 2009; Menon and Elkan, 2011).

For example, in the "NetFlix problem", the rows of the matrix correspond to users, the columns correspond to movies, and the entries correspond to the rating of a movie by a user. Predicting how *one specific* user will rate *one specific* movie then reduces to completing a *single unobserved entry* from the observed ratings.

For arbitrarily chosen position (i, j), the primary questions are:

- *Is it possible to reconstruct the entry* (i, j)*?*
- *How many possible completions are there for the entry* (i, j)*?*
- *What is the value of the entry* (i, j)*?*
- *How accurately can one estimate the entry* (i, j)*?*

In this paper, we answer these questions *algorithmically* under the common *low-rank assumption* - that is, under the model assumption (or approximation) that there is an underlying complete matrix of some low rank r from which the partial observations arise. Our algorithms are the first in the low-rank regime that provide information about single entries. They adapt to the combinatorial structure of the observations in that, if it is possible, the reconstruction process can be carried out using much less than the full set of observations. We validate our algorithms on real data. We also identify *combinatorial* features of the low-rank completion problem. This then allows us to study low-rank matrix completion via tools from, e.g., graph theory.

1.1 Results

Here is a preview of the results and themes of this paper, including the answers to the main questions.

1.1.1 Is it possible to reconstruct the entry (i, j)?

We show that whether the entry (i, j) is completable depends, with probability one for *any continuous sampling regime*, only on the *positions* of the observations and the position (i, j) that we would like to reconstruct (Theorem 10). The proof is explicit and easily converted into an exact (probability one) algorithm for computing the set of completable positions (Algorithm 1).

1.1.2 How many possible completions are there for the entry (i, j)?

Whether the entry at position (i, j) is uniquely completable from the observations, or, more generally, how many completions there are also depends, with probability one, only on the positions of the observed entries and (i, j) (Theorem 17). We also give an efficient

(randomized probability one) algorithm (Algorithm 1) that verifies a sufficient condition for every unobserved entry to be uniquely completable.

1.1.3 What is the value of the entry (i, j)?

To reconstruct the missing entries, we introduce a general scheme based on finding polynomial relations between the observations and one unobserved one at position (i, j) (Algorithm 5). For rank one matrices (Algorithm 6), and, in any rank, observation patterns with a special structure (Algorithm 4) that allows "solving minor by minor", we instantiate the scheme completely and efficiently.

Since, for a specific (i, j), the polynomials needed can be very sparse, our approach has the property that it adapts to the combinatorial structure of the observed positions. To our knowledge, other algorithms for low-rank matrix completion do not have this property.

1.1.4 How accurately can one estimate entry (i, j)?

Our completion algorithms separate out *finding* the relevant polynomial relations from *solving* them. When there is more than one relation, we can use them as different estimates for the missing entry, allowing for estimation in the noisy setting (Algorithm 5). Because the polynomials are independent of specific observations, the same techniques yield *a priori estimates* of the variance of our estimators.

1.1.5 Combinatorics of matrix completion

Section 6 contains a detailed analysis of whether an entry (i, j) is completable in terms of a *bipartite graph* encoding the combinatorics of the observed positions. We obtain necessary (Theorem 38) and sufficient (Proposition 42) conditions for local completability, which are sharp in the sense that our local algorithms apply when they are met. We then relate the properties we find to standard graph-theoretic concepts such as edge-connectivity and cores. As an application, we determine a binomial sampling density that is sufficient for solving minor-by-minor nearly exactly via a random graph argument.

1.1.6 Experiments

Section 7 validates our algorithms on the Movie Lens data set and shows that the structural features identified by our theories predict completability and completability phase transitions in practice.

1.2 Tools and themes

Underlying our results are a new view of low-rank matrix completion based on algebraic geometry. Here are some of the key ideas.

1.2.1 Using the local-to-global principle

Our starting point is that the set of rank r, $(m \times n)$-matrices carries the additional structure of an *irreducible algebraic variety* (see Section 2.1). Additionally, the observation process is a polynomial map. The key feature of this setup is that it gives us access to fundamental

algebraic-geometric "local-to-global" results (see Appendix A) that assert the observation process will exhibit a *prototypical behavior*: the answers to the main questions will be the *same for almost all* low-rank matrices, so they are essentially properties of the rank and observation map. This lets us study the main questions in terms of observed and unobserved *positions* rather than specific *partial matrices*.

On the other hand, the same structural results show we can *certify* that properties like completability hold via *single examples*. We exploit this to replace very complex basis eliminations with fast algorithms based on numerical linear algebra.

1.2.2 Finding relations among entries using an ideal

Another fundamental aspect of algebraic sets are characterized exactly by the *vanishing ideal* of polynomials that evaluate to zero on them. For matrix completion, the meaning is: *every* polynomial relation between the observations and a specific position (i, j) is generated by a *finite* set of polynomials we can in principle identify (See Section 5).

1.2.3 Connecting geometry to combinatorics using matroids

Our last major ingredient is the use of the Jacobian of the observation map, evaluated at a "generic point". The independence/dependence relation among its rows is invariant (with matrix-sampling probability one) over the set of rank r matrices that characterizes whether a position (i, j) is completable. Considering the subsets of independent rows as simply subsets of a finite set, we obtain a *linear matroid* characterizing completability. This perspective allows access to combinatorial tools of matroid theory, enabling the analysis in Section 6.

1.3 Context and novelty

Low-rank matrix completion has received a great deal of attention from the community. Broadly speaking, two main approaches have been developed: convex relaxations of the rank constraints (e.g., Candès and Recht, 2009; Candès and Tao, 2010; Negahban and Wainwright, 2011; Salakhutdinov and Srebro, 2010; Negahban and Wainwright, 2012; Foygel and Srebro, 2011; Srebro and Shraibman, 2005); and spectral methods (e.g., Keshavan et al., 2010; Meka et al., 2009). Both of these (see Candès and Tao, 2010; Keshavan et al., 2010) yield, in the noiseless case, optimal sample complexity bounds (in terms of the number of positions uniformly sampled) for exact reconstruction of an underlying matrix meeting certain analytic assumptions. All the prior work of which we are aware concentrates on: (A) sets of observed positions sampled from some known distribution; (B) completing all the unobserved entries. The results here, by contrast, apply specifically to *fixed* sets of observations and provide information about *any* unobserved position (i, j).

To point (A), there are three notable exceptions: Singer and Cucuringu (2010) discuss a mathematical analogy to combinatorial rigidity, studying which fixed observation patterns allow unique and stable completions; their work is to a large part conjectural but exposes the connection to graph combinatorics and anticipates some of our theoretical results. Lee and Shraibman (2013) study completion guarantees for fixed observation patterns with tools inspired by and related to the nuclear norm. Bhojanapalli and Jain (2014) showed a sufficient condition for exact recovery by nuclear norm minimization when the bipartite

graph corresponding to the observed positions has a large spectral gap under a strong incoherence assumption.

Regarding point (B) more specifically, all the prior work on low-rank matrix completion from noisy observations concentrates on: (i) estimating every missing entry; (ii) denoising every observed entry; and (iii) minimizing the MSE over the whole matrix. Our approach allows, for the first time, to construct *single-entry estimators* that minimize the variance of the entry under consideration; we have recently shown how to do this efficiently in rank 1 (Kiraly and Theran, 2013).

1.4 Organization

The sequel is structured as follows: Section 2 introduces the background material we need; Sections 3 and 4 develop our algebraic-combinatorial theory and derive algorithms for determining when an entry is completable; Section 5 formulates the reconstruction process itself algebraically; Section 6 contains a combinatorial analysis of the problem; finally Section 7 validates our approach on real data. The Appendix collects some technical results required in the proofs of the main theorems.

2. Background and Setup

In this section, we introduce two essential objects, the set of low-rank matrices $\mathcal{M}(m \times n, r)$ and the set of observed positions E. We also define the concept of *genericity*.

2.1 The determinantal variety

First, we set up basic notation. A matrix is denoted by upper-case bold character like $\mathbf{A}$. We denote by $[n]$ the set of integers $\{1, 2, \ldots, n\}$. $\mathbf{A}_{I,J}$ denotes the submatrix of an $m \times n$ matrix $\mathbf{A}$ specified by the sets of indices $I \subseteq [m]$ and $J \subseteq [n]$. The (i, j) element of a matrix $\mathbf{A}$ is denoted by A_{ij}. The cardinality of a set I is denoted by $|I|$.

Now we define the set of matrices of rank at most r.

Definition 1 *The set of all complex $(m \times n)$-matrices of rank r or less will be denoted by $\mathcal{M}(m \times n, r) = \{\mathbf{A} \in \mathbb{C}^{m \times n} : \operatorname{rank}(\mathbf{A}) \leq r\}$. We will always assume that $r \leq m \leq n$; by transposing the matrices, this is no loss of generality.*

Some basic properties of $\mathcal{M}(m \times n, r)$ are summarized in the following proposition.

Proposition 2 (Properties of the determinantal variety) *The following hold:*

(i) $\mathcal{M}(m \times n, r)$ is the image of the map $\Upsilon : (\mathbf{U}, \mathbf{V}) \mapsto \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{C}^{m \times r}$ and $\mathbf{V} \in \mathbb{C}^{n \times r}$, and is therefore irreducible.

(ii) $\mathcal{M}(m \times n, r)$ has dimension

$$d_r(m, n) := \dim \mathcal{M}(m \times n, r) = \begin{cases} r(m + n - r) & \text{if } m \geq r \text{ and } n \geq r \\ mn & \text{otherwise} \end{cases}$$

(iii) *Every* $(r+1)\times(r+1)$ *minor of a matrix in* $\mathcal{M}(m\times n, r)$ *is zero, namely,*

$$\det(\mathbf{A}_{I,J}) = 0, \quad \forall I \subseteq [m], J \subseteq [n],$$

where $|I| = r+1$, $|J| = r+1$, *and* $\mathbf{A} \in \mathcal{M}(m\times n, r)$.

(iv) *The vanishing ideal of* $\mathcal{M}(m\times n, r)$ *is generated by the vanishing of the minors from part 3.*

Proof **(i)** The existence of the singular-value decomposition imply that $\mathcal{M}(m\times n, r)$ is the surjective image of $\mathbb{C}^{r(m+n)}$ under the algebraic map Υ.

(ii) This follows from **(i)** and the uniqueness of the singular value decomposition, or Bruns and Vetter (1988, section 1.C, Proposition 1.1).

(iii) The rank of a matrix equals the order of the largest non-vanishing minor.

(iv) By Bruns and Vetter (1988, Theorem 2.10, Remark 2.12, and Corollary 5.17f), the ideal generated by the $r\times r$ minors is prime. Since it vanishes on the irreducible $\mathcal{M}(m\times n, r)$, it is the vanishing ideal. ■

The set of observed positions is denoted by E and can be viewed as a bipartite graph as follows.

Definition 3 *Let* $\mathcal{E} := [m]\times[n]$. *The set containing the positions of observed entries is denoted by* $E \subseteq \mathcal{E}$. *We define the bipartite graph* $G(E) = (V, W, E)$ *with vertices* $V = [m]$ *corresponding to rows and vertices* $W = [n]$ *corresponding to columns. We call the* $m\times n$ *adjacency matrix* $\mathbf{M}(E)$ *of the bipartite graph* $G(E)$ *a* mask. *The map*

$$\Omega : \mathbf{A} \mapsto (A_{ij})_{(i,j)\in E},$$

where $\mathbf{A} \in \mathcal{M}(m\times n, r)$, *is called a* masking *(in rank* r*).*

Note that the set of observed positions E, the adjacency matrix $\mathbf{M}$, and the map Ω can be used interchangeably. For example, we denote by $\mathbf{M}(\Omega)$ the adjacency matrix corresponding to the map Ω, and by $E(\mathbf{M})$ the set of positions specified by $\mathbf{M}$, and so on. Figure 1 shows two bipartite graphs G_1 and G_2 corresponding to the following two masks:

$$\mathbf{M}_1 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \mathbf{M}_2 = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}.$$

2.2 The Jacobian of the masking operator

Informally, the question we are going to address is:

Which entries of $\mathbf{A}$ *are (uniquely) reconstructable, given the masking* $\Omega(\mathbf{A})$*?*

The answer will depend on the interaction between the algebraic structure of $\mathcal{M}(m\times n, r)$ and the combinatorial structure of E. The main tool we use to study this is the Jacobian of the map Υ, since at smooth points, we can obtain information about the dimension of the pre-image $\Omega^{-1}(\mathbf{A})$ from its rank.

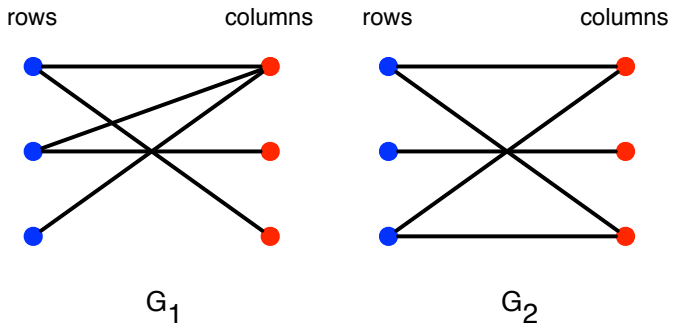


Figure 1: Two bipartite graphcs G_1 and G_2 corresponding to the masks $\mathbf{M}_1$ and $\mathbf{M}_2$, respectively. Every non-edge corresponds to an unobserved entry.

Definition 4 *We denote by* $\mathbf{J}$ *the Jacobian of the map* $\Upsilon : \mathbf{U}, \mathbf{V} \mapsto \mathbf{A} = \mathbf{U}\mathbf{V}^\top$. *More specifically, the Jacobian of the map from* $\mathbf{U}$ *and* $\mathbf{V}$ *to* A_{ij} *can be written as follows:*

$$\left(\frac{\partial A_{ij}}{\partial \mathbf{u}_1^\top}, \ldots, \frac{\partial A_{ij}}{\partial \mathbf{u}_m^\top}, \frac{\partial A_{ij}}{\partial \mathbf{v}_1^\top}, \ldots, \frac{\partial A_{ij}}{\partial \mathbf{v}_n^\top}\right) = \underset{\substack{\uparrow \\ \text{Derivative wrt } \mathbf{u}_i}}{\left(0 \quad \cdots \quad \mathbf{v}_j^\top \quad \cdots 0 \quad 0 \cdots \quad \underset{\uparrow}{\mathbf{u}_i^\top} \quad \cdots \quad 0\right)} \qquad (1)$$

$$\text{Derivative wrt } \mathbf{v}_j$$

where $\mathbf{u}_i^\top$ *is the* i*th row vector of* $\mathbf{U}$ *and* $\mathbf{v}_j^\top$ *is the* j*th row vector of* $\mathbf{V}$. *Stacking the above row vectors for* $(i, j) \in [m] \times [n]$, *we can write the Jacobian* $\mathbf{J}(\mathbf{U}, \mathbf{V})$ *as an* $mn \times r(m+n)$ *matrix as follows:*

$$\mathbf{J}(\mathbf{U}, \mathbf{V}) = \begin{pmatrix} \mathbf{I}_m \otimes \mathbf{v}_1^\top & \\ \mathbf{I}_m \otimes \mathbf{v}_2^\top & \mathbf{I}_n \otimes \mathbf{U} \\ \vdots & \\ \mathbf{I}_m \otimes \mathbf{v}_n^\top & \end{pmatrix}, \qquad (2)$$

where $\otimes$ *denotes the Kronecker product. Here the rows of* $\mathbf{J}$ *correspond to the entries of* $\mathbf{A}$ *in the column major order.*

Lemma 5 *Every matrix* $\mathbf{S} \in \mathbb{C}^{m \times n}$ *whose vectorization* $\mathrm{vec}(\mathbf{S})$ *lies in the left null space of* $\mathbf{J}(\mathbf{U}, \mathbf{V})$ *satisfies*

$$\mathbf{U}^\top \mathbf{S} = 0, \qquad \mathbf{S}\mathbf{V} = 0,$$

and any $\mathbf{S}$ *satisfying the above lies in the left null space of* $\mathbf{J}(\mathbf{U}, \mathbf{V})$. *In addition, the dimension of the null space is* $(m-r)(n-r)$ *if* $\mathbf{U}$ *and* $\mathbf{V}$ *have full column rank* r.

Proof Let $\mathbf{P}$ be the $mn \times mn$ permutation matrix defined by

$$\mathbf{P}\,\mathrm{vec}(\mathbf{X}) = \mathrm{vec}(\mathbf{X}^\top).$$

Note that $\mathbf{P}^\top \mathbf{P} = \mathbf{I}_{mn}$, and

$$\mathbf{P} \begin{pmatrix} \mathbf{I}_m \otimes \mathbf{v}_1^\top \\ \vdots \\ \mathbf{I}_m \otimes \mathbf{v}_n^\top \end{pmatrix} = \mathbf{I}_m \otimes \mathbf{V}.$$

Thus we have

$$\begin{aligned} \mathrm{vec}^\top(\mathbf{S})\mathbf{J} &= \left(\mathrm{vec}^\top(\mathbf{S})\mathbf{P}^\top (\mathbf{I}_m \otimes \mathbf{V}), \mathrm{vec}^\top(\mathbf{S}) (\mathbf{I}_n \otimes \mathbf{U})\right) \\ &= \left(\mathrm{vec}^\top(\mathbf{V}^\top \mathbf{S}^\top), \mathrm{vec}^\top(\mathbf{U}^\top \mathbf{S})\right), \end{aligned}$$

which is what we wanted. To show the last part of the lemma, let $\mathbf{U}_\perp \in \mathbb{C}^{m\times(m-r)}$ and $\mathbf{V}_\perp \in \mathbb{C}^{n\times(n-r)}$ be any basis of the orthogonal complement space of $\mathbf{U}$ and $\mathbf{V}$, respectively. Since the null space can be parametrized as $\mathbf{S} = \mathbf{U}_\perp \mathbf{S}' \mathbf{V}_\perp^\top$ by $\mathbf{S}' \in \mathbb{C}^{(m-r)\times(n-r)}$, and this parametrization is one-to-one, we see that the dimension of the null space is $(m-r)(n-r)$. ■

Now we define the Jacobian corresponding to the set of observed positions E.

Definition 6 *For a position (k,ℓ), we define $\mathbf{J}_{(k,\ell)}$ to be the single row of $\mathbf{J}$ corresponding to the position (k,ℓ). Similarly, we define $\mathbf{J}_E$ to be the submatrix of $\mathbf{J}$ consisting of rows corresponding to the set of observed positions E. Due to the chain rule, $\mathbf{J}_E$ is the Jacobian of the map $\Omega \circ \Upsilon$.*

2.3 Genericity

The pattern of zero and non-zero entries in (1) hints at a connection to purely combinatorial structure. To make the connection precise, we introduce *genericity*.

Definition 7 *We say a boolean statement $P(\mathbf{X})$ holds* for a generic $\mathbf{X}$ *in irreducible algebraic variety $\mathcal{X}$, if for any Hausdorff continuous measure μ on $\mathcal{S}$, $P(\mathbf{X})$ holds with probability* 1.

These kinds of statements are sometimes called "generic properties," and they are properties of $\mathcal{X}$, rather than any specific μ. The prototypical example of a generic property is where $\mathcal{X} = \mathbb{C}^n$, $p \neq 0$ is a polynomial, and the statement P is "$p(\mathbf{X}) \neq 0$."

Here, we are usually concerned with the case $\mathcal{X} = \mathcal{M}(m \times n, r)$. Proposition 2 tells us that m, n and r define $\mathcal{M}(m \times n, r)$ completely. Assertions of the form "For generic $\mathbf{X} \in \mathcal{M}(m \times n, r)$, $P(\mathbf{X})$ depends only on $(t_1, t_2, \ldots)$" mean $P(\mathbf{X})$ is a generic statement for *all* $\mathcal{M}(m \times n, r)$ with the parameters t_i fixed.

Although showing whether some statement P holds generically might seem hard, we are interested in P defined by polynomials. In this case, results in Appendix A imply that it is enough to show that either P holds: (a) on an open subset of $\mathcal{X}$ in the metric topology; or (b) almost surely, with respect to a Hausdorff continuous measure.

As a first step, and to illustrate the "generic philosophy" we show that the generic behavior of the Jacobian $\mathbf{J}_E(\mathbf{U}, \mathbf{V})$ is a property of E. We first start by justifying the definition via $(\mathbf{U}, \mathbf{V})$ (as opposed to $\mathbf{A}$).

Lemma 8 *For all $E \subset \mathcal{E}$ and $\mathbf{A} \in \mathcal{M}(m \times n, r)$ generic, with $\mathbf{A} = \Upsilon(\mathbf{U}, \mathbf{V})$, and $\mathbf{U}$ and $\mathbf{V}$ generic, the rank of $\mathbf{J}_E$ is independent of $\mathbf{A}$, $\mathbf{U}$, and $\mathbf{V}$.*

Proof We first consider the composed map $\Omega \circ \Upsilon$. This is a polynomial map in the entries of $\mathbf{U}$ and $\mathbf{V}$, so its critical points (at which the differential $\mathbf{J}_E$ attains less than its maximum rank) is an algebraic subset of $\mathbb{C}^{r(m+n)}$. The "Semialgebraic Sard Theorem" (Kurdyka et al., 2000, Theorems 3.1, 4.1) then implies that the set of critical points is, in fact, a *proper* algebraic subset of $\mathbb{C}^{r(m+n)}$.

So far, we have proved that the rank of $\mathbf{J}_E$ is independent of $\mathbf{U}$ and $\mathbf{V}$. However, $\mathbf{U}$ and $\mathbf{V}$ are not uniquely determined by $\mathbf{A}$. To reach the stronger conclusion, we first observe that a generic $\mathbf{A} \in \mathcal{M}(m \times n, r)$ is a regular value of Υ, again by Semialgebraic Sard. Thus, the set of $(\mathbf{U}, \mathbf{V})$ such that $\Upsilon(\mathbf{U}, \mathbf{V})$ and $\Omega \circ \Upsilon(\mathbf{U}, \mathbf{V})$ are both regular values is the intersection of two dense sets in $\mathbb{C}^{r(m+n)}$. ■

3. Finite Completability

This section is devoted to the question *"Is it possible to reconstruct the entry (i, j)?"*. We will show that under mild assumptions, the answer depends only on the *position* (i, j), the observed positions, and the rank, but not the observed entries. The main idea behind this result is relating reconstructability to the rows of the Jacobian $\mathbf{J}$, and their rank, which can be shown to be independent of the actual entries for almost all low-rank matrices. Therefore, we can later separate the question of reconstructability from the actual reconstruction process.

3.1 Finite completability as a property of the positions

We show how to predict whether the entry at a specific *position* (k, ℓ) will be reconstructable from a specific set of positions $E \subset \mathcal{E}$. For the rest of this section, we fix the parameters r, m and n, and denote by E a set of observed positions. The symbol $\mathbb{K}$ will denote either of the real numbers $\mathbb{R}$ or the complex numbers $\mathbb{C}$.

We start by precisely defining what it means for one set of entries to imply the imputability of another entry.

Definition 9 *Let $E \subset \mathcal{E}$ be a set of observed positions and $\mathbf{A}$ be a rank r true matrix. The entry $A_{k\ell}$ is* finitely completable in rank r *from the observed set of entries $\{A_{ij} : (i, j) \in E\}$ if the entry $A_{k\ell}$ can take only finitely many values when fixing $\Omega(\mathbf{A})$.*

There are two subtleties here: the first is that, even if there is an infinity of possible completions for the whole matrix $\mathbf{A}$, it is possible that some specific $A_{k\ell}$ takes on only finitely many values; the question of whether the entry $A_{k\ell}$ at position (k, ℓ) is finitely completable may have different answers for different $\mathbf{A}$. The theoretical results in this section take care of both issues.

Theorem 10 *Let $E \subset \mathcal{E}$ be a set of positions, $(k, \ell) \in \mathcal{E} \setminus E$ be arbitrary, and let $\mathbf{A} \in \mathbb{K}^{m \times n}$ be a generic, $(m \times n)$-matrix of rank r. Whether the entry $A_{k\ell}$ at position (k, ℓ) is finitely completable depends only on the position (k, ℓ), the true rank r, and the observed positions E (and not on $\mathbf{A}$, m, n, or $\mathbb{K}$).*

This lets us talk about the finite completability of positions instead of entries.

Definition 11 *Let $E \subset \mathcal{E}$ be a set of observed positions, and $(k,\ell) \in \mathcal{E} \setminus E$. We say that the* position (k,ℓ) is finitely completable from E in rank r *if, for generic $\mathbf{A}$, the entry $A_{k\ell}$ is finitely completable from $\Omega(\mathbf{A})$. The* rank r finitely completable closure $\mathrm{cl}_r(E)$ *is the set of positions generically finitely completable from E.*

The main tool we use to prove Theorem 10 is the Jacobian matrix $\mathbf{J}_E$. For it, we obtain

Theorem 12 *Let $E \subset \mathcal{E}$ and let $\mathbf{A}$ be a generic, rank r matrix. Then*

$$\mathrm{cl}_r(E) = \{(k,\ell) \in \mathcal{E} : \mathbf{J}_{\{(k,\ell)\}} \in \operatorname{rowspan} \mathbf{J}_E\}.$$

One implication of Theorem 12 is that linear independence of subsets of rows of $\mathbf{J}_E$ is also a generic property. (In fact, the proof in Section 3.3 goes in the other direction.) The combinatorial object that captures this independence is a matroid.

Definition 13 *Let $\mathbf{A}$ be a generic rank r matrix. The* rank r determinantal matroid *is the linear matroid $(\mathcal{E}, \mathrm{rank}_r)$, with rank function $\mathrm{rank}_r(E) = \operatorname{rank} \mathbf{J}_E$.*

Note that due to Lemma 8, $\operatorname{rank} \mathbf{J}_E$ is independent of $\mathbf{A}$ as long as we are concerned with generic matrices and the rank function is well defined.

In the language of matroids, Theorem 12 says that, generically, the finitely completable closure is equal to the matroid closure in the rank r determinantal matroid. This perspective will prove profitable when we consider entry-by-entry algorithms for completion in Section 5 and combinatorial conditions related to finite completability in Section 6.

3.2 Computing the finite closure

We describe, in pseudo-code, Algorithm 1 which computes the finite closure of E. An algorithm for testing whether a single entry (k,ℓ) is finitely completable is easily obtained by only testing the entry (k,ℓ) in step 4. The correctness of Algorithm 1 follows from Theorem 12 and the fact that, if we sample $\mathbf{U}$ and $\mathbf{V}$ from any continuous density, with probability one, we obtain generic $\mathbf{U}$ and $\mathbf{V}$[2].

Remark 14 *For clarity and practicality, we have presented Algorithm 1 as a numerical routine based on SVD. To analyze it in the RAM model, instead of sampling $\mathbf{U}$ and $\mathbf{V}$ from a continuous density, we sample the entries uniformly from a finite field $\mathbb{Z}_p$ of prime order $p \approx (n+m)^2$. With this modification Algorithm 1 becomes strongly polynomial time via, e.g., Gaussian elimination. Using the main results of Schwartz (1980), one can show that this finite field variant computes the generic rank with probability $1 - O(1/(n+m))$.*

3.3 Proofs

3.3.1 Proof of Theorem 12

Let $(i,j) \in \mathcal{E} \setminus E$. Factor the map $\Omega \circ \Upsilon$ into

$$\mathbb{C}^{r(m+n)} \xrightarrow{\Upsilon} \mathcal{M}(m \times n, r) \xrightarrow{f} \mathbb{C}^{|E|+1} \xrightarrow{g} \mathbb{C}^{|E|}$$

2. If we discretize the continuous density, then "probability one" becomes "with high probability".

Algorithm 1 Completable closure.
Input: A set $E \subset \mathcal{E}$ of observed positions.
Output: The rank r completable closure $\mathrm{cl}_r(E)$.

1: Sample $\mathbf{U} \in \mathbb{R}^{m\times r}, \mathbf{V} \in \mathbb{R}^{n\times r}$ from a continuous density.
2: Compute the Jacobian matrix $\mathbf{J}_E(\mathbf{U}, \mathbf{V})$.
3: Compute the singular value decomposition of $\mathbf{J}_E(\mathbf{U}, \mathbf{V})$. Let $\mathbf{V}_E$ be the right singular vectors corresponding to singular values greater than 10^{-12}.
4: For each $e \in \mathcal{E}\backslash E$, compute the projection of $\mathbf{J}_{\{e\}}(\mathbf{U}, \mathbf{V}) \in \mathbb{R}^{r(m+n)}$ on the subspace spanned by $\mathbf{V}_E$. Let the Euclidean norm of the residual of the projection be r_e; let $r_e = 0$ for $e \in E$.
5: Return $\mathrm{cl}_r(E) := \{(i,j) \in \mathcal{E} \,;\, r_e \leq 10^{-8}\}$.

so that f is the projection of $\Upsilon(\mathbf{U}, \mathbf{V})$ onto the set of entries at positions $E \cup \{(i,j)\}$ and g then projects out the coordinate corresponding to (i,j). Lemma 8 implies that, since $(\mathbf{U}, \mathbf{V})$ is generic, all the intermediate image points are smooth. The constant rank theorem then implies:

1. We can find open neighborhoods $f(\mathcal{M}(m\times n, r)) \supset M \ni f(\Upsilon(\mathbf{U}, \mathbf{V}))$ and $g\circ f(\mathcal{M}(m\times n, r)) \supset N \ni g(f(\Upsilon(\mathbf{U}, \mathbf{V})))$ such that the restriction of g to M is smooth and $g^{-1}(N) \subset M$.

2. We have
$$\dim\left(g^{-1}(N)\right) + \dim N = \dim M.$$

Since by using smoothness again

$$\dim N = \dim\left(g(f(\Upsilon(\mathbf{U}, \mathbf{V})))\right) = \operatorname{rank}\left(\mathbf{J}_E(\mathbf{U}, \mathbf{V})\right),$$

and

$$\dim M = \dim\left(f(\Upsilon(\mathbf{U}, \mathbf{V}))\right) = \operatorname{rank}\left(\mathbf{J}_{E\cup\{i,j\}}(\mathbf{U}, \mathbf{V})\right),$$

$\dim\left(g^{-1}(N)\right) = 0$, that is, the position (i,j) is finitely completable from E and $\Upsilon(\mathbf{U}, \mathbf{V})$, if and only if

$$\operatorname{rank}\left(\mathbf{J}_E(\mathbf{U}, \mathbf{V})\right) = \operatorname{rank}\left(\mathbf{J}_{E\cup\{i,j\}}(\mathbf{U}, \mathbf{V})\right). \tag{3}$$

Equation (3) is just the assertion that $\mathbf{J}_{\{(i,j)\}} \in \operatorname{rowspan} \mathbf{J}_E$.

By Lemma 8, Equation (3) is a generic statement, independent of $\mathbf{A}$, $\mathbf{U}$ and $\mathbf{V}$. Because the rows of $\mathbf{J}_E$ and $\mathbf{J}_{\{(i,j\})}$ have non-zero columns only at positions depending on E and (i,j), whether (3) holds does not depend on m and n (which are, by hypothesis, large enough).

Finally, statement that finite completability is the same for $\mathbb{K} = \mathbb{R}$ and $\mathbb{K} = \mathbb{C}$ follows from Theorem 68 in the appendix.

3.3.2 Proof of Theorem 10

The theorem follows directly from Theorem 12 and the definition of closure.

3.4 Discussion

The kernel of $\mathbf{J}_E$ spans the space of infinitesimal deformations of $(\mathbf{U}, \mathbf{V})$ that preserve $\Omega \circ \Upsilon(\mathbf{U}, \mathbf{V})$. Because generic points are smooth, Milnor (1968, Curve Selection Lemma) implies that every infinitesimal deformation can be integrated to a finite deformation. Conversely (this is the harder direction) every curve in $(\Omega \circ \Upsilon)^{-1}(\mathbf{A})$ through $(\mathbf{U}, \mathbf{V})$ has, as its tangent vector a non-zero infinitesimal deformation. At non-generic points, this equivalence does not hold, so the arguments here require genericity and smoothness in an essential way.

The finite identifiability statements in this section are instances of a more general phenomenon, which is explored in Király et al. (2013). The results there imply similar identifiability results, such as Bamber (1985); Allman et al. (2009); Hsu et al. (2012); Mahdi et al. (2014); Meshkat et al. (2014), that use criteria based on a Jacobian, and also show that our use of the "Υ" parameterization of $\mathcal{M}(m \times n, r)$ is not essential.

Another connection is that, since permuting the rows and columns of a matrix preserves its rank, we get:

Corollary 15 *The rank function* $\mathrm{rank}_r(\cdot)$ *of the determinantal matroid depends only on the graph isomorphism type of the graph associated with* E*.*

In Section 6, we consider completability as a property of *graphs*. This relies on Corollary 15.

4. Unique Completability

In this section, we will address the question *"How many possible completions are there for the entry* (i, j)*?"*. In Section 3.1, it was shown that whether the *entry* (i, j) is completable depends (under mild assumptions) only on the *position* (i, j), the observed entries, and the rank. In this section, we show an analogue result that the same holds for the number of possible completions as well. Whether there is exactly one solution is of the most practical relevance, and we give a sufficient condition for unique completability.

4.1 Unique completability as a property of the positions

We start by defining what it means for one entry to be uniquely completable:

Definition 16 *Let* $E \subset \mathcal{E}$ *be a set of observed positions and* $\mathbf{A}$ *be a rank* r *true matrix. The entry* $A_{k\ell}$ *at position* $(k, \ell) \in \mathcal{E} \setminus E$ *is called* uniquely completable *from the entries* $A_{ij}, (i, j) \in E$*, if* $A_{k\ell}$ *is uniquely determined by the* $A_{ij}, (i, j) \in E$*.*

The main theoretical statement for unique completability is an analogue to the main theorem for finite completability; again, whether an entry is uniquely completable, depends only on the positions of the observations, assuming the true matrix is generic.

Theorem 17 *Let* $\mathbf{A} \in \mathbb{C}^{m \times n}$ *be a generic* $(m \times n)$*-matrix of rank* r*, and consider a masking where the entries* A_{ij} *with* $(i, j) \in E \subseteq [m] \times [n]$ *are observed. Let* $(k, \ell) \in [m] \times [n]$ *be arbitrary. Then, whether* $A_{k\ell}$ *is uniquely completable from the* $A_{ij}, (i, j) \in E$ *depends only on the position* (k, ℓ)*, the true rank* r *and the observed positions* E *(and not on* $\mathbf{A}$*,* m *or* n*).*

The proof of Theorem 17 is a bit more technical than its finite completability analogue, Theorem 10. The main problem is that the constant rank theorem cannot be applied since the latter is a local statement only and does not make say anything about the global number of solutions. The proper tools to overcome that are found in algebraic geometry; a complete proof is deferred to Section 4.4. The proof we give also shows that there is an analog statement for the total number of possible completions, even if there is more than one. Since the number of completions over the reals can potentially change even with generic $\mathbf{A}$, the result is stated only over the complex numbers.

Theorem 17 shows that it makes sense to talk about positions instead of entries that are uniquely completable, in analogy to the finite case; moreover, it shows that there is a biggest such set:

Definition 18 *Let $E \subseteq [m] \times [n]$ be the set of observed positions, and let $(k, \ell) \in [m] \times [n]$ be a position. We will call (k, ℓ) uniquely completable if $A_{k\ell}$ is uniquely completable from $A_{ij}, (i, j) \in E$ for a generic matrix $\mathbf{A} \in \mathbb{K}^{m \times n}$ of rank r.*

Furthermore, we will denote by $\operatorname{ucl}_r(E)$ the inclusion-wise maximal set of positions such that every index $(k, \ell) \in \operatorname{ucl}_r(E)$ is uniquely completable from E. We will call the $\operatorname{ucl}_r(E)$ unique closure *of E in rank r.*

As for finite completability, we can check generic unique completability of a position by testing a random $\mathbf{A}$. However, we don't have an analogue for the Jacobian $\mathbf{J}_E$ that exactly characterizes unique completability. One could, of course, use general Gröbner basis methods, but these are computationally impractical. In the next section, we describe an easy-to-check sufficient condition for unique completability in terms of the Jacobian.

4.2 Characterization by Jacobian stresses

As for the case of finite completability, the Jacobian of the masking can be used to provide algorithmic criteria to determine whether an entry is uniquely completable. The characterizing objects will be the so-called *stresses*, dual objects to the column space of the Jacobian. Intuitively, they correspond to infinitesimal dual deformations. Singer and Cucuringu (2010, Equation 3.7) have defined a similar concept which is closely related to the equilibrium stresses of Connelly (2005, Section 1.3).

Mathematically, stresses are left kernels of the Jacobian:

Definition 19 *A rank-r* stress *of the matrix $\mathbf{A} = \mathbf{U}\mathbf{V}^\top$ is a matrix $\mathbf{S} \in \mathbb{C}^{m \times n}$ whose vectorization is in the left kernel of the Jacobian $\mathbf{J}(\mathbf{U}, \mathbf{V})$; that is,*

$$\operatorname{vec} \mathbf{S} \cdot \mathbf{J}(\mathbf{U}, \mathbf{V}) = 0.$$

*Let $E \subseteq [m] \times [n]$ be a set of observed entries. A stress $\mathbf{S}$ such that $\mathbf{S}_{ij} = 0$ for all $(ij) \notin E$ is called E-*stress *of $\mathbf{A}$.*

The $\mathbb{C}$-vector space of E-stresses of $\mathbf{A}$ will be denoted by $\Psi_{\mathbf{A}}(E)$, noting that it does not depend on the choice of $\mathbf{U}, \mathbf{V}$.

Note that E-stresses are, after vectorization and removing zeroes, in the left kernel of the partial Jacobian $\mathbf{J}_E$.

The central property of the stress which allows to test for unique completability is its rank (as a matrix in $\mathbb{C}^{m\times n}$):

Definition 20 *Let $E \subseteq [m] \times [n]$ be as set of observed entries. We define the* maximal E-stress rank *of $\mathbf{A}$ in rank r to be*

$$\rho_{\mathbf{A}}(E) = \max_{\mathbf{S}\in\Psi_{\mathbf{A}}(E)} \operatorname{rank} \mathbf{S}.$$

As for the rank of the Jacobian, the dependence on $\mathbf{A}$ can be removed for generic matrices:

Proposition 21 *Let $\mathbf{A}$ be a generic $(m \times n)$-matrix of rank r. The maximal stress rank $\rho_{\mathbf{A}}(E)$ depends only on E and r. In particular, $\rho_{\mathbf{A}}(E)$ does not depend on the entries of $\mathbf{A}$.*

Proof Let $\mathbf{A} = \Upsilon(\mathbf{U}, \mathbf{V})$. By Cramer's rule, if $\mathbf{S} \in \Psi_{\mathbf{A}}(E)$, the entries of $\mathbf{S}$ are rational functions of the entries of $\mathbf{U}$ and $\mathbf{V}$. After clearing denominators, the proof is similar to that of Lemma 8. ■

We can therefore just talk about the generic E-stress rank, omitting again the dependence on the entries $\mathbf{A}$:

Definition 22 *Let $E \subseteq [m] \times [n]$ be as set of observed entries. We define the* generic E-stress rank $\rho(E)$ *to be equal to $\rho_{\mathbf{A}}(E)$ for generic $\mathbf{A}$ or rank r.*

Our main theorem states that if the generic E-stress rank is maximal for finitely completable E, then E is also uniquely completable:

Theorem 23 *Let $E \subseteq \mathcal{E}$. If the generic E-stress rank in rank r is $\rho(E) \geq \min(m, n) - r$, then $\operatorname{cl}_r(E) = \operatorname{ucl}_r(E)$.*

We defer the somewhat technical proof to Section 4.4.

4.3 Computing the generic stress rank

Theorem 23 implies that the generic stress rank $\rho(E)$ can be used to certify unique completability of an observation pattern E. We explicitly describe the necessary computational steps in Algorithm 2.

As the algorithm for finite completion, it uses a randomized strategy which allows to compute over the real numbers instead of a field of rational functions by substituting a generic entry. Steps 1 and the beginning of step 2 are thus analogous as in Algorithm 1. In step 2, the completion matrix $\mathbf{J}_E$ is computed, evaluated at the matrices $(\mathbf{U}, \mathbf{V})$. In 3, an evaluated stress $\mathbf{S}$ is obtained in the left kernel of $\mathbf{J}_E$. Its rank, which is computed in step 5, will be the generic stress rank. Correctness (with probability one) is implied by Proposition 21. Also, similar to Algorithm 1, Algorithm 2 is a randomized algorithm for which considerations analogue to those in Remark 14 hold.

Algorithm 2 Generic stress rank.
Input: Observed positions $E \subseteq \mathcal{E}$. *Output:* The generic stress rank $\rho(E)$ of E in rank r.

1: Randomly sample $\mathbf{U} \in \mathbb{R}^{m\times r}, \mathbf{V} \in \mathbb{R}^{n\times r}$.
2: Compute $\mathbf{J}_E(\mathbf{U}, \mathbf{V})$ with rows $\mathbf{J}_{(i,j)} := (\mathbf{e}_i \otimes \mathbf{v}_j^\top, \mathbf{e}_j \otimes \mathbf{u}_i^\top)$,
3: where $\mathbf{u}_i$ is the i-th row of $\mathbf{U}$, and $\mathbf{v}_j$ the j-th row of $\mathbf{V}$.
4: Compute a random vector $\mathbf{S} \in \mathbb{R}^{|E|}$ in the left kernel of $\mathbf{J}_E$. Reformat $\mathbf{S}$ as $(m \times n)$ matrix, where entries with index not in E are zero, and the remaining indices correspond to the row positions in $\mathbf{J}_E$.
5: Output $\rho(E) = \operatorname{rank}(\mathbf{S})$.

4.4 Proofs

4.4.1 Proof of Theorem 17

Proof Consider the algebraic map

$$g : (A_{k\ell}; A_{ij}, (i,j) \in E) \mapsto (A_{ij}, (i,j) \in E)$$

By Proposition 58 in the appendix, Ω is a surjective algebraic map of irreducible varieties. Therefore, the generic fiber cardinality $\left|g^{-1} \circ g(x)\right|$ for generic $x \in \mathcal{X}$ does not depend on x by Corollary 62. In particular, whether $1 = \left|g^{-1} \circ g(x)\right|$ or not.

■

4.4.2 Proof of Theorem 23

This sections contains the proof for Theorem 23 and some related results.

Lemma 24 *Let $\mathbf{S} \in \mathbb{C}^{m\times n}$ be a stress w.r.t. $m, n, r, \mathbf{A} = \mathbf{U}\mathbf{V}^\top$. Then,*

$$\mathbf{U}^\top \cdot \mathbf{S} = 0 \quad \textit{and} \quad \mathbf{S} \cdot \mathbf{V} = 0$$

(where 0 denotes the zero matrix of the correct size).

Proof Since $\mathbf{S}$ is an stress, it holds by definition that $\operatorname{vec}\mathbf{S} \cdot \mathbf{J}(\mathbf{U}, \mathbf{V}) = 0$. The statement then follows from Lemma 5. ■

Lemma 24 immediately implies a rank inequality:

Corollary 25 *Let $E \subseteq [m] \times [n]$, assume the true matrix has full rank r. Then, it holds that $\rho(E) \leq \min(m, n) - r$*

Proof Keep the notations of Lemma 24. The statement Lemma 24 implies that for arbitrary $\mathbf{S}$, one has $\mathbf{S} \cdot \mathbf{V} = 0$. Since $\mathbf{V}$ is a matrix of full rank r, this implies that the null space dimension of $\mathbf{S}$ is at least r, which is equivalent to the statement by the rank-nullity theorem.
■

In keeping with our development of finite completability in terms of $\mathbf{J}_E$, we have defined

stresses in a way that might depend on the coordinates $(\mathbf{U}, \mathbf{V})$. In the proof of Theorem 23, we will check that this can be removed when necessary. An alternative but probably less concise approach would be to express the matrix $\mathbf{J}_E$ directly in terms of the entries $\mathbf{A}$.

4.4.3 Proof of Theorem 23

We start with a general statement that stresses are invariant over the pre-image $(\Omega \circ \Upsilon)^{-1}(\Omega(\mathbf{A}))$, loosely inspired by the work of Connelly (2005).

Lemma 26 *Let* $\mathbf{A}$ *be generic, with* $\Upsilon(\mathbf{U}, \mathbf{V}) = \mathbf{A}$*,* $E \subset \mathcal{E}$*, and* $\mathbf{S}$ *an* E*-stress. Then* $\mathbf{S}$ *is also an* E*-stress for any* $(\mathbf{U}', \mathbf{V}')$ *with* $\Omega \circ \Upsilon(\mathbf{U}', \mathbf{V}') = \Omega(\mathbf{A})$

Proof Let $(\mathbf{U}', \mathbf{V}') \in (\Omega \circ \Upsilon)^{-1}(\Omega \circ \Upsilon(\mathbf{U}, \mathbf{V}))$ be a point different from $(\mathbf{U}, \mathbf{V})$. Because $\Omega(\mathbf{A})$ is a regular value of the composed map $\Omega \circ \Upsilon$, the Inverse Function Theorem provides diffeomorphic neighborhoods $M \ni (\mathbf{U}, \mathbf{V})$ and $N \ni (\mathbf{U}', \mathbf{V}')$; let $f : M \to N$ be the diffeomorphism.

By construction, df is non-singular. The chain rule then implies that $(\mathbf{J}_E)_{(\mathbf{U}', \mathbf{V}')} = (\mathbf{J}_E)_{(\mathbf{U}, \mathbf{V})} \cdot df^{-1}$, so the left kernels of both Jacobians are the same. The definition of stress as a vector in the left kernel then proves the lemma. ■

Proof [of Theorem 23] It is clear that $\mathrm{cl}_r(E) \supseteq \mathrm{ucl}_r(E)$. Thus we show that $\mathrm{cl}_r(E) \subseteq \mathrm{ucl}_r(E)$. By Lemma 26, $\mathbf{S}$ is a stress for any $(\mathbf{U}, \mathbf{V})$ that agrees with the observed entries $\Omega(\mathbf{A})$ on the observed positions E. Then by Lemma 24, any such pair $(\mathbf{U}, \mathbf{V})$ must satisfy $\mathbf{U}^\top \cdot \mathbf{S} = 0$ and $\mathbf{S} \cdot \mathbf{V}$. Since generically the stress has rank $\min(m, n) - r$, these equations determine the row and column spans of $\mathbf{A}$. Once the row and column spans are fixed, any row or column with at least r observed positions can be uniquely determined. On the other hand, any row or column with fewer than r observed positions cannot be recovered (even if the row or column span is known). Therefore we have $\mathrm{cl}_r(E) \subseteq \mathrm{ucl}_r(E)$. ■

5. Local Completion

In this section, we connect our theoretical results to the process of reconstructing the missing entries. In a nutshell, the idea is that a completable missing entry $(i, j) \in \mathcal{E} \setminus E$ is covered by at least one so-called *circuit* in $E \cup \{(i, j)\}$, to which we can associate *circuit polynomials* which can be used to solve for A_{ij} in terms of the observations, addressing the question *"What is the value of the entry* (i, j)*?"*. Just as in theory where we could separate the reconstructability from the reconstruction, we can obtain a quantitative version of this separation by estimating the entry-wise reconstruction error without actually performing the reconstruction, allowing to give an answer to *"How accurately can one estimate the entry* (i, j)*?"*. We give general algorithms for arbitrary rank, and a closed-form solution for rank one.

5.1 Circuits as rank certificates

We start with some concepts from matroid theory.

Definition 27 *A set of observed positions $C \subseteq \mathcal{E}$ is called a* circuit *of rank r if* $\operatorname{rank}_r(C) = |C| - 1$ *and* $\operatorname{rank}_r(S) = |S|$ *for all proper subsets $S \subsetneq C$. The graph $G(C)$ is called* circuit graph *of rank r.*

A reformulation of Theorem 10, in terms of circuits is the following.

Theorem 28 *The position (i, j) is finitely completable if and only if there is a circuit $C \subset E \cup \{(i, j)\}$ with $(i, j) \in C$.*

Proof See (Oxley, 2011, Lemma 1.4.3) ■

The connection to reconstructing missing entries is that every circuit comes with a unique polynomial:

Theorem 29 *Let $C \subseteq \mathcal{E}$ be a circuit in rank r, Ω_C be the mask corresponding to C, and $\mathbf{A} \in \mathbb{C}^{m\times n}$. There is a unique, up to scalar multiplication, square-free polynomial θ_C such that: $\theta_C(\Omega_C(\mathbf{A})) = 0$ if and only if there is $\mathbf{A}' \in \mathcal{M}(m \times n, r)$ and $\Omega_C(\mathbf{A}) = \Omega_C(\mathbf{A}')$.*

Proof This follows indirectly from Theorem 1.1 in Dress and Lovász (1987), or from the discussion in Section 5.2 of Király et al. (2013) ■

In other words, circuit polynomials minimally certify for the rank r condition being fulfilled on the entries in C. The simplest example of a circuit is an $(r + 1) \times (r + 1)$ rectangle in $\mathcal{E}$. The associated polynomial is the determinant of an $(r + 1) \times (r + 1)$ minor of $\mathbf{A}$. Thus, Theorem 29 is a generalization of the linear algebra fact that a matrix is rank r if and only if all $(r + 1)$-minors vanish.

Definition 30 *We will call the polynomial θ_C from Theorem 29 a* circuit polynomial *associated with the circuit C. Understanding that there are an infinity up to multiplication with a scalar multiple, we will also talk about* the *circuit polynomial when that does not make a difference.*

Remark 31 *The circuit polynomial can be interpreted algorithmically as follows: let $C \subseteq \mathcal{E}$ be a circuit, assume all entries but one in C are observed, e.g., $(k, \ell) \in C$ is not observed and $E = C \setminus (k, \ell)$ is observed. Then, $\theta_C(\Omega_C(\mathbf{A})) = \theta_C(A_{k\ell}, \Omega_E(\mathbf{A}))$ can be interpreted as a polynomial in the one unknown $A_{k\ell}$. That is, the circuit polynomial allows to solve entry-wise for single missing entries.*

Definition 32 *Fix some set of observed entries $E \subseteq \mathcal{E}$. A circuit $C \subseteq \mathcal{E}$ is called* completing *for the observations in E, or with respect to E, if $|C \cap E| \geq |C| - 1$.*

5.2 Completion with circuit polynomials

The circuit properties inspire a general solution strategy. In general, Algorithm 3 is ineffective, in the sense that Step 4 is unlikely to have a sub-exponential time algorithm in the general case. However, there is a specific instance in which it is effective: when the circuit C is always an $(r + 1) \times (r + 1)$ rectangle. In this case, the circuit polynomial is the corresponding $(r+1)\times(r+1)$ minor. This means that enumerating all the circuits through (i, j) is not necessary, because a minor is linear in the unknown entry $A_{k\ell}$.

Algorithm 3 Completion with circuits.
Input: A set $E \subset \mathcal{E}$ of observed positions.
Output: Estimates for the entries $\mathrm{cl}_r(E) \setminus E$

repeat
  Find an unobserved entry $(k, \ell) \in \mathrm{cl}_r(E) \setminus E$,
  Find the set $\mathcal{C} = \{C_1, \ldots, C_t\}$ of all circuits (w.r.t. E) containing (k, ℓ).
  Compute the circuit polynomials θ_{C_i}.
  Substitute the entries $\{A_{ij} : (i, j) \in E\}$ into the θ_{C_i} to get a family of polynomials in the variable $A_{k\ell}$ and find a solution $A_{k\ell}$ common to all of them.
  $E \leftarrow E \cup (k, \ell)$
until $E = \mathrm{cl}_r(E)$.

A practical algorithm for computing the closure of a mask E and recovering the corresponding entries based on $(r+1) \times (r+1)$ minors is given in Algorithm 4. In Step 5, $N(j)$ and $N(i)$ denote the set of neighbors of vertices $j \in W$ and $i \in V$, respectively. In Step 10, $A^+_{I',J'}$ denotes the Moore-Penrose pseudoinverse of $A_{I',J'}$. Intuitively, the algorithm iterates over missing edges and look if there is a $(r+1) \times (r+1)$ biclique in the union of current set of edges E_k and (i, j). If such a biclique exists, then the edge (i, j) is added to E_{k+1} so that the edge is used in the next round. The iteration terminates when there is no more edge to add.

Algorithm 4 `MinorClosure`$((V, W, E), r)$
Inputs: bipartite graph (V, W, E), rank r.
Outputs: completed matrix A and minor closure of E.

1: Let $E_0 \leftarrow E$ and $k \leftarrow 0$.
2: **repeat**
3: $E_{k+1} \leftarrow E_k$
4: **for** each missing edge (i, j) in $\mathcal{E} \setminus E_k$ **do**
5: Let $I \leftarrow N(j) \subseteq V$, $J \leftarrow N(i) \subseteq W$, where the neighbors are defined with respect to graph (V, W, E_k).
6: $E'_k \leftarrow I \times J \cap E_k$.
7: $(I', J') \leftarrow$ `FindAClique`$((I, J, E'_k), r, r)$.
8: **if** $|I'| > 0$ and $|J'| > 0$ **then**
9: $E_{k+1} \leftarrow E_{k+1} \cup (i, j)$.
10: $A_{ij} \leftarrow A_{i,J'} A^+_{I',J'} A_{I',j}$.
11: **end if**
12: **end for**
13: $k \leftarrow k + 1$.
14: **until** $E_k = E_{k-1}$ or $E_k = \mathcal{E}$
15: Return (A, E_k).

Note that E_{k+1} is uniquely determined from E_k and the process is monotone and bounded, i.e., $E_k \subseteq E_{k+1} \subseteq \mathcal{E}$. The first statement is true because the order of the iteration over missing edges in line 4 is irrelevant as we look if there is a $(r+1) \times (r+1)$

biclique in $E_k \cup (i,j)$ for each missing edge (i,j). Therefore, Algorithm 4 terminates with either $E_k = \mathcal{E}$ or $E_k \subsetneq \mathcal{E}$ and the following definition is valid.

Definition 33 *A set $E \subset \mathcal{E}$ is* minor closable in rank r *if Algorithm 4 reconstructs all the entries in positions $\mathcal{E} \setminus E$. Moreover, we say E is k-step minor closable in rank r, if Algorithm 4 terminates with k steps, i.e., $E_k = \mathcal{E}$ in line 14.*

Since each entry is uniquely determined when it is reconstructed, any minor closable set is uniquely completable.

A crucial step in Algorithm 4 is `FindAClique` in line 7. The function should return the indices of rows and columns, if an $r \times r$ biclique exists in subgraph (I, J, E'). This can be achieved in various ways. Although the worst case complexity is $O(|I|^r|J|^r)$, it can be much more efficient in practice, because many vertices can be safely pruned due to the fact that any $r \times r$ biclique may not contain vertices with degree less than r. An efficient implementation that employs a row-wise recursion of this step, proposed by Takeaki Uno, is presented in Appendix B.

We would like to note that Algorithm 3, as presented above, and all related algorithms below, need the true matrix to be generic. Probabilities for this supposition to hold can be backed out of from Remark 14.

5.3 Local completion

The circuit property can also be interpreted differently: instead of using multiple circuits to complete many different entries, one can also think of concentrating on one single entry and trying to reconstruct that as accurately as possible. Algorithm 5 describes a general strategy on how to obtain estimates of single finitely or uniquely completable entries, from noisy observations via local circuit completion.

Algorithm 5 Local completion/denoising of a single entry (k, ℓ).
Input: A set $E \subset \mathcal{E}$ of observed positions, the entry.
Output: Estimate for $A_{k\ell}$

1: Find completing (w.r.t. E) circuits $C_1, \dots, C_N$ containing (k, ℓ)
2: Compute the circuit polynomials θ_{C_i}, where the observed entries are substituted and $A_{k\ell}$ is the only unknown
3: For all i, find all solutions $a^{(i,j)}$ of θ_{C_i}.
4: Return $A_{k\ell} = f(\dots, a^{(i,j)}, \dots)$, where f is an appropriate averaging function

The idea in Algorithm 5 is to obtain many candidate solutions in step 3 and then trade them off appropriately in step 4. If all circuit polynomials θ_{C_i} have degree one, there is only one solution per polynomial, and f can be taken as the mean, or a weighted average that minimizes some loss or a variance. If there are some circuit polynomial with higher degree, then one can try to decide which solution is the right one - e.g., by clustering the $a^{(i,j)}$ and rejecting all candidate solutions except the one which contains some $a^{(i,j)}$ for the highest number of i, and then proceeding as in the degree one case. Also, one can imagine f being adaptive, e.g. including Bayesian learning methods.

For rank one, an closed explicit form is possible for the variance minimizing estimate, as it was shown in Kiraly and Theran (2013). For arbitrary rank, a first-order approximation to variance minimization can be employed to yield fast and competitive single-entry estimates; see Blythe et al. (2014) for a derivation of variance minimization in higher rank, and Blythe and Király (2015) for a practical adaptation of the algorithm to the context of athletic performance prediction.

For illustration, we give a short overview of the crucial statements in the rank one case. The proofs can be found in Kiraly and Theran (2013).

Theorem 34 *The rank one circuit graphs are exactly the simple cycles (bipartite and thus of even length). The corresponding circuit polynomials are all binomials of the form*

$$\theta_C = \prod_{\nu=1}^{L} A_{i_\nu j_\nu} - \prod_{\nu=1}^{L} A_{i_\nu j_{\nu+1}},$$

where L is an arbitrary number, $i_1, \dots, i_L$ are arbitrary disjoint numbers, and $j_1, \dots, j_L$ are arbitrary disjoint numbers, with the convention that $j_1 = j_{L+1}$. The i_ν and j_ν do not need to be disjoint from each other.

In particular, Theorem 34 implies that the circuit polynomials are all linear in every occurring variable. Moreover, the specific structure of the problem allows a further simplification:

Remark 35 *Keep the notations of Theorem 34. Write $B_{ij} := \log |A_{ij}|$. Then, the equations*

$$L_C = \sum_{\nu=1}^{L} B_{i_\nu j_\nu} - \sum_{\nu=1}^{L} B_{i_\nu j_{\nu+1}}$$

vanish on all rank one matrices.

With the elementary computation in Remark 35, matrix completion becomes estimation with linear boundary constraints. That is, the function f in step 4 of Algorithm 5 could be taken as the least squares regressor of all $B_{k\ell}$ obtained from completing circuits for (k, ℓ). The algorithm in Kiraly and Theran (2013) gives a version which takes different observation variances into account, and efficient graph theoretic observations making the computation polynomial.

We paraphrase this as Algorithm 6; more details, e.g. on how to efficiently find a basis for the set of completing circuits[3] is efficiently found, or how the kernel matrix Σ is constructed, can be found in Kiraly and Theran (2013).

5.4 Variance and error estimation

The locality of circuits also allows to obtain estimates for the reconstruction error of single missing entries obtained by the strategy in Section 5.3, independent of the method which does the actual reconstruction. The simplest estimate of this kind is obtained from a

3. This is equivalent to finding a basis for first $\mathbb{Z}$-homology of the graph G, taken as a 1-complex.

Algorithm 6 Local completion/denoising of a single entry (k, ℓ) in a rank 1 matrix.
Input: A set $E \subset \mathcal{E}$ of observed positions, observation variances σ, the position (k, ℓ).
Output: Estimate for $A_{k\ell}$

1: Find a basis $C_1, \ldots, C_N$ for the set of completing circuits (w.r.t E) for (k, ℓ)
2: Find solutions a_i for the corresponding circuit polynomials, write $b_i := \log |a_i|$
3: Compute the $(N \times N)$-path kernel matrix $\Sigma = \Sigma(E, \sigma)$ corresponding to the C_i; set $\alpha := \Sigma^{-1} \cdot \mathbf{1}$
4: Compute the weighted mean $b := \left(\sum_{i=1}^{N} \alpha_i \cdot a_i\right) / \left(\sum_{i=1}^{N} \alpha_i\right)$
5: As estimate, return $\widehat{A}_{k\ell} = \pm \exp(b)$, where the sign is determined by the sign parity of the circuits.

variational approach: say θ_C is a completing circuit (w.r.t $E \subseteq \mathcal{E}$) for the missing entry (k, ℓ). In the simplest case, where θ_C is linear in the missing entry $A_{k\ell}$, we can obtain a solving equation

$$\widehat{A}_{k\ell} = \theta_C(A_e, e \in E),$$

by solving for $A_{k\ell}$ as an unknown. A first order approximation for the standard error can be obtained by the variational approach

$$\delta \widehat{A}_{k\ell} = \sum_{e \in E} \frac{\partial \theta_C}{\partial A_e}(A_e, e \in E)\ \delta A_e.$$

The right hand side can be obtained from a suitable noise model and the observations A_e, or, if the error should be estimated independently from the A_e, from a noise model plus a sampling model for the A_e. A general strategy for entry-wise error estimation is analogous to Algorithm 5 for local completion. For rank one, it has been shown in Kiraly and Theran (2013) that the variance estimate depends only on the noise model and not on the actual observation, and takes a closed logarithmic-linear form, as it is sketched in Algorithm 7.

Algorithm 7 Error prediction for a single entry (k, ℓ), rank one.
Input: A set $E \subset \mathcal{E}$ of observed positions, observation variances σ, the position (k, ℓ).
Output: Estimate for the (log-)variance error of the estimate $\widehat{A}_{k\ell}$

1: Calculate Σ and α, as in Algorithm 6.
2: As log-variance, return $\alpha^\top \Sigma \alpha$.
3: If an estimate $\widehat{A}_{k\ell}$ is available, as standard error, return $\widehat{A}_{k\ell} \cdot \left(\exp(\alpha^\top \Sigma \alpha) - 1\right)$

Note that the log-variance error is independent of the actual estimate $\widehat{A}_{k\ell}$, therefore the variance patterns can be estimated without actually reconstructing the entries.

6. Combinatorial Completability Conditions

Through Sections 3 and 4, we have shown that for a given $E \subseteq \mathcal{E}$, both finitely completable closure (Theorem 12) and uniquely completable closure (Theorem 17) are properties of the (isomorphism type of) the associated bipartite graph $G(E)$; see also Corollary 15.

In this Section, using tools from graph and matroid theories, we relate the structural properties of the bipartite graph $G(E)$ to finite completability.

For a set of observed positions, $E \subseteq \mathcal{E}$, let $G(E) = (V, W, E)$ be a bipartite graph, where the sets of vertices V and W correspond to row and column of the observed positions; we call V and W *row vertices* and *column vertices*, respectively. We assume that $G(E)$ has no isolated vertices (those corresponding to rows or columns with no observed positions.)

As usual, we will take r, n, and m to be the rank and parameters of the ground set $\mathcal{E}$, respectively. However, since our convention for graphs is that they do not have isolated vertices, we will take care to indicate the ambient ground set.

6.1 Sparsity and independence

Suppose we want to maximize the size of the completable closure $\mathrm{cl}_r(E)$, with the number of positions to observe fixed. To do this, consider the process of constructing E one position at a time. What we need is to pick each successive entry in a way that causes $\mathrm{cl}_r(E)$ to grow. Theorem 12 implies that a position (k, l) is finitely completable from E, if and only if $\mathbf{J}_{\{(k,l)\}}$ lies in the span of $\mathbf{J}_E$. In particular, this tells us that adding such a (k, ℓ) to E will not affect the finite completability of other unobserved positions; in matroid terminology, we say (k, l) is dependent on E. We see, then, that it is wasteful to choose positions that are dependent on the already chosen positions. Therefore intuitively we need to choose the positions so that they are well spread out, which we call *rank-r sparse*; see Section 6.1.1. Rank-r sparsity implies a more classical combinatorial property, namely r-connectivity; see Section 6.1.2. Finally, in Section 6.1.3, we show by a counterexample that rank-r sparsity, though necessary, is not a sufficient condition for finite completability.

We recall some basic terminologies from matroid theory. The rank function $\mathrm{rank}_r(E)$ of the rank r determinantal matroid is defined in Definition 13. Note that $\mathrm{rank}_r(E) \leq d_r(m, n)$, where $d_r(m, n) = r(m + n - r)$ if $m \geq r$ and $n \geq r$, $d_r(m, n) = mn$, otherwise. A set of positions $E \subseteq \mathcal{E}$ is called *independent* if $|E| = \mathrm{rank}_r(E)$. On the other hand, it is called *dependent* if $|E| > \mathrm{rank}_r(E)$. A basis B of $E \subseteq \mathcal{E}$ is a maximally independent subset of E. In addition, a basis of $\mathcal{E}$ is called a basis of the rank r determinantal matroid. A basis B of E consists of $\mathrm{rank}_r(E)$ edges. In particular, a basis B of the rank r determinantal matroid consists of $\mathrm{rank}_r(\mathcal{E}) = d_r(m, n)$ edges. A basis of E is not unique unless E is independent. A circuit $C \subseteq \mathcal{E}$ of of the rank r determinantal matroid is a minimally dependent set in the sense that for any $(i, j) \in C$, $C - \{(i, j)\}$ is an independent set; see also Definition 27.

We have the following two properties from matroid theory.

Proposition 36 *1. Let $E \subseteq \mathcal{E}$ be a set of observed positions and $B \subseteq E$ be any basis of E. Then, $\mathrm{cl}_r(B) = \mathrm{cl}_r(E)$.*

2. Let $E \subseteq \mathcal{E}$ be an independent set in the rank r determinantal matroid. Then, any $E' \subseteq E$ is independent.

In other words, (i) the finitely completable closures of E and any basis B of E are the same (ii) and an independent graph $G(E)$ cannot contain a dependent subgraph $G(E')$. Both statements arise from the fact that the rank-r determinantal matroid is a linear matroid defined by the linear independence of the rows of the Jacobian $\mathbf{J}_E$ and that the matroid closure coincides with the finitely completable closure.

6.1.1 Rank-r-sparsity

Let $G' = (V', W', E')$ be a subgraph of $G = (V, W, E)$. Since E' being independent implies a bound on the cardinality $|E'| \leq d_r(|V'|, |W'|)$, we consider the notion of *rank-r-sparsity* defined as follows.

Definition 37 *A graph $G = (V, W, E)$ is* rank-r-sparse *if, for all subgraphs $G' = (V', W', E')$ of G, it holds that $|E'| \leq d_r(|V'|, |W'|)$.*

Theorem 38 *Let $E \subseteq \mathcal{E}$ be an independent set in the rank r determinantal matroid on $[m] \times [n]$. Then $G(E)$ is rank-r-sparse.*

Proof Suppose that there is a subgraph $G' = (V', W', E')$ with $|E'| > d_r(|V'|, |W'|) \geq \mathrm{rank}_r(E')$, then this subgraph must be dependent, which contradicts Proposition 36, part 2. ■

6.1.2 Connectivity and vertex degrees

Rank r sparsity implies some other, more classical, graph theoretic properties in a straightforward way, since rank-r-sparsity is hereditary.

Corollary 39 *Let $m, n > r$, and $E \subseteq \mathcal{E}$ be the set of observed positions. If $G(E)$ contains a rank-r sparse subgraph $G(E')$ with $|E'| = d_r(m, n)$ edges, then:*

1. *$G(E)$ has minimum vertex degree at least r.*
2. *$G(E)$ is r-edge-connected.*

In particular, if E is finitely completable, it contains a basis E' (Proposition 36, part 1) with $|E'| = d_r(m, n)$ edges and $G(E')$ is rank-r sparse. Thus, E is r-edge connected.

The proof of the above corollary relies on the following lemma:

Lemma 40 *Let $E \subseteq \mathcal{E}$ be rank-r sparse with $|E| = d_r(m, n)$ edges, and $E = \cup_{i=1}^{N} E_i$ be an edge disjoint partition of E. For any set $E' \subseteq \mathcal{E}$ of edges incident to m' row and n' column vertices, we define $d_r(E') := d_r(m', n')$. Then we have*

$$d_r(m, n) \leq \sum_{i=1}^{N} d_r(E_i).$$

Proof By the assumption,

$$d_r(m, n) = |E| = \sum_{i=1}^{N} |E_i| \leq \sum_{i=1}^{N} d_r(E_i),$$

where the first equality holds because E is independent and the last inequality follows from Theorem 38. ■

Proof [Proof of Corollary 39] Since Statement 2 implies statement 1, we prove Statement 2. First, we can assume without loss of generality that E is rank-r sparse and $E' = E$ without loss of generality, because if E' is r-edge-connected, so is E.

Consider any partition $V = V_1 \cup V_2$ and $W = W_1 \cup W_2$. V_1 or W_1 can be empty (but not at the same time). This induces an edge disjoint partition $E = E_1 \cup E_2 \cup_{(i,j)\in E-E_1-E_2} \{(i,j)\}$, where E_1 and E_2 are sets of edges induced by (V_1, W_1) and (V_2, W_2), respectively. Treating each edge in $E - E_1 - E_2$ as a subgraph, we have $d_r((i,j)) = 1$. By applying Lemma 40, we have

$$|E - E_1 - E_2| \geq d_r(m,n) - d_r(E_1) - d_r(E_2). \tag{4}$$

Let $m_1 := |V_1|$, $m_2 := |V_2|$, $n_1 := |W_1|$, and $n_2 := |W_2|$. Due to symmetry, there are three situations that we need to consider. First, if $m_1, m_2, n_1, n_2 \geq r$, RHS of (4) $= r^2$. Next, if $m_1 \leq r$ and $n_2 \leq r$, RHS of (4) $= r(m+n-r) - m_1 n_1 - m_2 n_2 \geq r^2$, which is true considering maximizing the inner product between (m_1, m_2) and (n_1, n_2) subject to $m_1 + m_2 = m$ and $n_1 + n_2 = m$. Finally, if $m_1, n_1 \leq r$, RHS of (4) $= r(m_1 + n_1) - m_1 n_1 \geq r$. The minimum is obtained for $m_1 = 1$ and $n_1 = 0$, or vice versa. Therefore E is r-edge connected. ■

6.1.3 Sparsity is not sufficient

On the other hand, rank r sparsity is *not* a sufficient condition for independence in determinantal matroids. The bipartite graph defined by the following mask in rank 2 have $d_2(5,5) = 16$ edges and rank-2 sparse but not independent:

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

This example amounts, graph theoretically, to gluing the graphs of two bases of the determinantal matroid together along r vertices in a way that preserves rank-r-sparsity but not independence. One can make the construction rigorous to show that, for any $r \geq 2$, there are infinitely many rank-r-sparse dependent sets in the determinantal matroid.

6.2 Circuit and stress supports

We have discussed stresses in Section 4 and circuits in Section 5. Here we show that for each circuit C, there is a corresponding stress $\mathbf{S}$ that is supported on every position of C. Here the support $S \in \mathcal{E}$ of stress $\mathbf{S}$ is defined as $S = \{(i,j) \in \mathcal{E} : S_{ij} \neq 0\}$. Moreover, using the structure of the Jacobian matrix (see Definition 4), we show that every vertex of circuit C has degree at least $r+1$. These results further imply that any finitely completable position spans vertices in the r-core (see Section 6.2.1). Furthermore, combining the above degree lower bound with the rank-r sparsity shown in the previous subsection, we show a bound on the number of circuits in the rank r determinantal matroid in Section 6.2.2. The proof of the key Theorem 41 is presented in Section 6.2.3.

Theorem 41 *For a generic $\mathbf{A} \in \mathcal{M}(m \times n, r)$, and a circuit C, the stress space $\Psi_{\mathbf{A}}(C)$ is one dimensional; thus a stress $\mathbf{S}$ of a circuit C is unique up to scalar multiplication. Moreover, the support of $\mathbf{S}$ is all of C.*

The power of Theorem 41 can be seen in the following proposition, which lower bounds the degree of a vertex in a circuit.

Proposition 42 *Let $C \subseteq E$ be a circuit in the rank r determinantal matroid. Then every vertex in the graph $G(C)$ has degree at least $r+1$ edges.*

Proof By Theorem 41, for generic $(\mathbf{U}, \mathbf{V})$, the rows of $\mathbf{J}_C$ are dependent, with the associated stress $\mathbf{S}$ supported on all the rows.

From (2), we see that any vertex j is associated with exactly r columns in $\mathbf{J}_C$. Let J be the indices of these columns. The number of non-zero rows in $\mathbf{J}_C[\cdot, J]$ is exactly the degree d of j. If we suppose $d \leq r$, the stress $\mathbf{S}$ cannot generically cancel these d columns. Therefore, it holds that $d \geq r+1$. ■

6.2.1 Where are the completable positions?

The concept of k-core is useful for narrowing down where the completable positions can be and where the circuits can lie.

We recall a concept from graph theory:

Definition 43 *Let G be a graph, and let $k \in \mathbb{N}$. The k-*core *of G, denoted $\operatorname{core}_k(G)$, is the maximal subgraph of G with minimum vertex degree k.*

In rank r, the non-trivial aspects of matrix completion occur inside the r-core.

Theorem 44 *Let $E \subseteq \mathcal{E}$,*

(i) If $(i,j) \in \mathcal{E} \setminus E$ and $(i,j) \in \operatorname{cl}_r(E)$, then the vertices i and j are in $\operatorname{core}_r(G(E))$.

(ii) Any circuit $C \subseteq E$ is contained in $\operatorname{core}_{r+1}(G(E))$.

Proof (i) We have $(i,j) \in \operatorname{cl}_r(E)$ if and only if there is a circuit $C \subseteq E \cup \{(i,j)\}$ with $(i,j) \in C$. Then (i) will follow from (ii) because for $G(C) \subseteq \operatorname{core}_{r+1}(G(E))$, we need $i \in \operatorname{core}_r(G(E))$ and $j \in \operatorname{core}_r(G(E))$.
(ii) This follows from the fact that the $(r+1)$-core is the union of all induced subgraphs with minimum degree at least $r+1$ and by Proposition 42, every C lies inside such an induced subgraph. ■

Note here that $\operatorname{ucl}_r(E) \subseteq \operatorname{cl}_r(E)$, so the same things are true for the uniquely completable closure.

6.2.2 Circuit size and counting

Combining the results in this section, we obtain bounds on the number of circuits in the rank r determinantal matroid.

Theorem 45 *Let C be a circuit in the rank r determinantal matroid with graph $G(C) = (V, W, C)$. Then $|W| \leq r(|V| - r) + 1$*

Proof Let $m' = |V|$ and $n' = |W|$. Using Proposition 42 for the lower bound and Theorem 38 for the upper bound, we have $n'(r+1) \leq |C| \leq r(m' + n' - r) + 1$. Subtracting $n'r$ from both sides, we get $|W| = n' \leq r(m' - r) + 1$. ■

Corollary 46 *The number of circuits in the rank r determinantal matroid on $[m] \times [n]$ is at most $2^{mr(m-r)+m}$.*

6.2.3 Proof of Theorem 41

First, by Definition 19, $\operatorname{rank}_r(C) = \operatorname{rank} \mathbf{J}_C = |C| - 1$. Thus the left null space of $\mathbf{J}_C$ is one dimensional.

Next, we explicitly construct a stress $\mathbf{S}$. By Theorem 29, there is a unique polynomial θ_C for each circuit C. Then taking the derivative of θ_C, we have

$$\sum_{(i,j)\in C} \left.\frac{\partial \theta_C}{\partial A_{ij}}\right|_{\Omega_C(\mathbf{A})} dA_{ij} = 0,$$

for any tangent vector $(dA_{ij})_{(i,j)\in C}$ of $\mathcal{M}(m \times n, r)$ at $\mathbf{A}$. The vector $(\partial\theta_C/\partial A_{ij})_{(i,j)\in C}|_{\Omega_C(\mathbf{A})}$ is, then, a stress for C. In addition, the coefficient of the stress is uniquely determined by the entries $\Omega_C(\mathbf{A})$. If any of the coefficients of $(\partial\theta_C/\partial A_{ij})$ were identically zero, we could remove the associated row ij of $\mathbf{J}_C$ and the left-kernel of $\mathbf{J}_{C\setminus(i,j)}$ would still be one-dimensional. Since this is a contradiction to C being a circuit, we conclude that none of the coefficients are identically zero. Since the coefficients are, in addition, rational functions in $\Omega_C(\mathbf{A})$, each of them is non-vanishing on a Zariski open subset of $\mathcal{M}(m \times n, r)$. The (finite) intersection of these sets is again open, proving that the generic support of the stress is all of C.

6.3 Completability of random masks

Up to this point we have considered the completability of a fixed mask, which we have shown to be equivalent to questions about the associated bipartite graph. We now turn to the case where the masking is sampled at random, which, by Corollary 15, implies that, generically, this is a question about *random bipartite graphs.*

6.3.1 Random graph models

A *random graph* is a graph valued random variable. We are specifically interested in two such models for bipartite random graphs:

Definition 47 *The* Erdős-Rényi *random bipartite graph $G(m, n, p)$ is a bipartite graph on m row and n column vertices vertices with each edge present with probability p, independently.*

Definition 48 *The* (d, d')-biregular *random bipartite graph* $G(m, n, d, d')$ *is the uniform distribution on graphs with* m *row vertices,* n *column ones, and each row vertex with degree* d *and each column vertex with degree* d'*.*

Clearly, we need $md = nd'$, and if $m = n$, the (d, d')-regular random bipartite graph is, in fact d-regular.

We will call a mask corresponding to a random graph a *random mask*. We now quote some standard properties of random graphs we need.

Proposition 49 (i) *Connectivity threshold. The threshold for* $G(m, n, p)$ *to become connected, w.h.p., is* $p = \Theta((m + n)^{-1} \log n)$ *(Bollobás, 2001, Theorem 7.1).*

(Minimum degree threshold) *The threshold for the minimum degree in* $G(n, n, p)$ *to reach* d *is* $p = \Theta((m + n)^{-1}(\log n + d \log \log n + \omega(1)))$*. When* $p = cn$*, w.h.p., there are isolated vertices (Bollobás, 2001, Exercise 3.2).*

(ii) *Connectivity threshold. With high probability,* $G(m, n, d, d')$ *is* d*-connected (Bollobás, 2001, Theorem 7.3.2). (Recall that we assume* $m \leq n$*) .*

(iii) *Density principle. Suppose that the expected number of edges in either of our random graph models is at most* Cn*, for constant* C*. Then for every* $\epsilon > 0$*, there is a constant* c*, depending on only* C *and* ϵ *such that, w.h.p., every subgraph of* n' *vertices spanning at least* $(1 + \epsilon)n'$ *edges has* $n' \geq cn$ *(Janson and Luczak, 2007, Lemma 5.1).*

(iv) *Emergence of the* k*-core. Define the* k*-core of a graph to be the maximal induced subgraph with minimum* k*. For each* k*, there is a constant* c_k *such that* $p = c_k/n$ *is the first-order threshold for the* k*-core to emerge. When the* k*-core emerges, it is giant and afterwards its size and number of edges spanned grows smoothly with* p *(Pittel et al., 1996).*

6.3.2 Sparser sampling and the completable closure

The lower bounds on sample size for completion of rank r *incoherent* matrices do *not* carry over verbatim to the generic setting of this paper. This is because genericity and incoherence are related, but incomparable concepts: there are generic matrices that are not incoherent (consider a very small perturbation of the identity matrix); and, importantly, the block diagonal examples showing the lower bound for incoherent completability are not generic, since many of the entries are zero.

Thus, in the generic setting, we expect sparse sampling to be more powerful. This is demonstrated experimentally in Section 7.2. In the rest of this section, we derive some heuristics for the expected generic completability behavior of sparse random masks. We are particularly interested in the question of: *when are* $\Omega(mn)$ *of the entries completable from a sparse random mask*? We call this the *completability transition*. We will conjecture that there is a sharp threshold for the completability transition, and that the threshold occurs well below the threshold for $G(n, m, p)$ to be completable.

Let c be a constant. We first consider the emergence of a circuit in $G(n, n, c/n)$. Theorem 44 implies that any circuit is a subgraph of the $(r + 1)$-core. By Theorem 12 and

Proposition 36, having a circuit is a monotone property, which occurs with probability one for graphs with more than $2rn$ edges, and thus the value

$$t_r := \sup\{t : G(n, n, t/n) \text{ is } r\text{-independent, w.h.p.}\}$$

is a constant. If we define C_r as

$$C_r := \sup\{c : \text{the } (r+1)\text{-core of } G(n, n, c/n) \text{ has average degree at most } 2r, \text{ w.h.p.}\}$$

smoothness of the growth of the $(r+1)$-core implies that we have

$$c_{r+1} \leq t_r \leq C_{r+1}$$

where we recall that c_{r+1} is the threshold degree for the $(r+1)$-core to emerge. Putting things together we get:

Proposition 50 *There is a constant t_r such that, if $c < t_r$ then w.h.p., $G(n, n, c/n)$ is r-independent, and, if $c > t_r$ then w.h.p. $G(n, n, c/n)$ contains a giant r-circuit inside the $(r+1)$-core. Moreover, t_r is at most the threshold for the $(r+1)$-core to reach average degree $2r$.*

Proposition 50 gives us some structural information about where to look for rank r circuits in $G(n, n, c/n)$: they emerge suddenly inside of the $(r+1)$-core and are all giant when they do. If rank r circuits were themselves completable, this would then yield a threshold for the completability transition. Unfortunately, the discussion in Section 6.1.3 tell us that this is not always true. Nonetheless, we conjecture:

Conjecture 51 *The constant t_r is the threshold for the completability transition in $G(n, n, c/n)$. Moreover, we conjecture that almost all of the $(r+1)$-core is completable above the threshold.*

We want to stress that the conjecture includes a conjecture about the *existence* of the threshold for the completabilty transition, which hasn't been established here, unlike the existence for the emergence of a circuit. The subtlety is that we haven't ruled out examples of r-independent graphs with no rank-r-spanning subgraph for which, nonetheless, the closure in the rank r completion matroid is giant. Conjecture 51 is explored experimentally in Sections 7.1 and 7.2. The conjectured behavior is analogous to what has been proved for distance matrices (also known as *bar-joint frameworks*) in dimension 2 in (Kasiviswanathan et al., 2011).

Our second conjecture is about $2r$-regular masks.

Conjecture 52 *With high probability $G(n, n, 2r, 2r)$ is completable. Moreover, we conjecture that it remains so, w.h.p., after removing r^2 edges uniformly at random.*

We provide evidence in Section 7.2. This behavior is strikingly different than the incoherent case, and consistent with proven results about 2-dimensional distance matrices (Jackson et al., 2007, Theorem 4.1).

6.3.3 Denser sampling and the r-closure

The conjectures above, even if true, provide only information about matrix *completability* and not matrix *completion*. In fact, the convex relaxation of Candès and Recht (2009) does not seem to do very well on $2r$-regular masks in our experiments, and the density principle for sparse random graphs implies that, w.h.p., a $2r$-regular mask has no dense enough subgraphs for our closability algorithm in Section B.1 to even get started. Thus it seems possible that these instances are quite "hard" to complete even if they are known to be completable.

If we consider denser random masks, then the closability algorithm becomes more practical. A particularly favorable case for it is when every missing entry is part of some $K^-_{r+1,r+1}$. In this case, the error propagation will be minimal and, heuristically, finding a $K^-_{r+1,r+1}$ is not too hard, even though the problem is NP-complete in general.

Define the 1-*step r-closure* of a bipartite graph G as the graph G' obtained by adding the missing edge to each $K^-_{r+1,r+1}$ in G. If the 1-step closure of G is $K_{n,n}$, we define G to be 1-*step r-closable*. We conjecture an upper bound on the threshold for 1-step r-closability.

Conjecture 53 *There is a constant $C > 0$ such that, if $p = Cn^{-2/(r+2)} \log n$ then, w.h.p., $G(n, n, p)$ is 1-step r-closable.*

7. Experiments

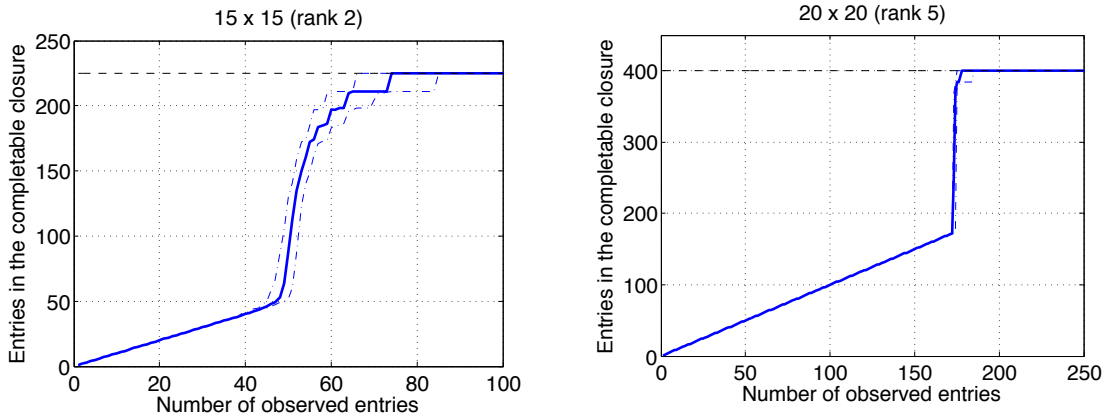
In this section we will investigate the set of entries that are finitely completable from a set of given entries. In Section 3 we have seen that the finitely completable closure $\mathrm{cl}_r(E)$ does not depend on the values of the observed entries but only on their positions E. First, we check the set of completable entries for synthetic random positions and empirically investigate the completability phase transitions in terms of the number of known entries, as described in Section 6.3. We also check the number of completable entries for MovieLens data set in terms of the putative rank. Then, we present experiments on actual reconstruction and algorithm-independent error estimation in the case of rank one matrices.

7.1 Randomized algorithms for completability

For a quantitative analysis, we perform experiments to investigate how the expected number of completable entries is influenced by the number of known entries. In particular, Section 6.3 suggests that a phase transition between the state where only very few additional entries can be completed and the state where a large set of entries can be completed should take place at some point. Figure 2 shows that this is indeed the case when slowly increasing the number of known entries: first, the set of completable entries is roughly equal to the set of known entries, but then, a sudden phase transition occurs and the set of completable entries quickly reaches the set of all entries.

7.2 Phase transitions

Figure 3 shows phase transition curves of various conditions for 100×100 matrices at rank 3. We consider uniform sampling model here. More specifically, we generated random 100×100 masks with various number of edges by first randomly sampling the order of



(a) Results for $m = 15, n = 15, r = 2$

(b) Results for $m = 20, n = 20, r = 5$

Figure 2: Expected number of completable entries (in rank r) versus the number of known entries where the positions of the known entries are uniformly randomly sampled in an $(m \times n)$-matrix. The expected number of completable entries was estimated for each data points from repeated calculations of the completable closure (200 for $r = 2$, and 20 for $r = 5$). The blue solid line is the median, the blue dotted lines are the 1st and 3rd quartiles. The black dotted line is the total number of entries, $m \cdot n$.

edges (using MATLAB `randperm` function) and adding 100 entries at a time from 100 to 6000 sequentially. In this way, we made sure to preserve the monotonicity of the properties considered here. This experiment was repeated 100 times and averaged to obtain estimates of success probabilities. The conditions plotted are (a) minimum degree at least r, (b) r-connected, (c) completable at rank r, (d) minor closable in rank r (e) nuclear norm successful, and (f) one-step minor closable. For nuclear norm minimization (e), we used the implementation of the algorithm in (Tomioka et al., 2010) which solves the minimization problem

$$\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} \|\mathbf{X}\|_* \quad \text{subject to} \quad X_{ij} = A_{ij} \quad \forall (i,j) \in E,$$

where $\|\mathbf{X}\|_* = \sum_{j=1}^{r} \sigma_j(\mathbf{X})$ is the nuclear norm of X. The success of nuclear norm minimization is defined as the relative error $\|\hat{\mathbf{X}} - \mathbf{A}\|_F / \|\mathbf{A}\|_F$ less than 0.01.

The success probabilities of the (a) minimum degree, (b) r-connected, and (c) completable are almost on top of each other, and exceeds chance (probability 0.5) around $|E| \simeq 1,000$. The success probability of the (d) minor closable curve passes through 0.5 around $|E| \simeq 1,300$. Therefore the r-closure method is nearly optimal. On the other hand, the nuclear norm minimization required about $2,200$ entries to succeed with probability larger than 0.5.

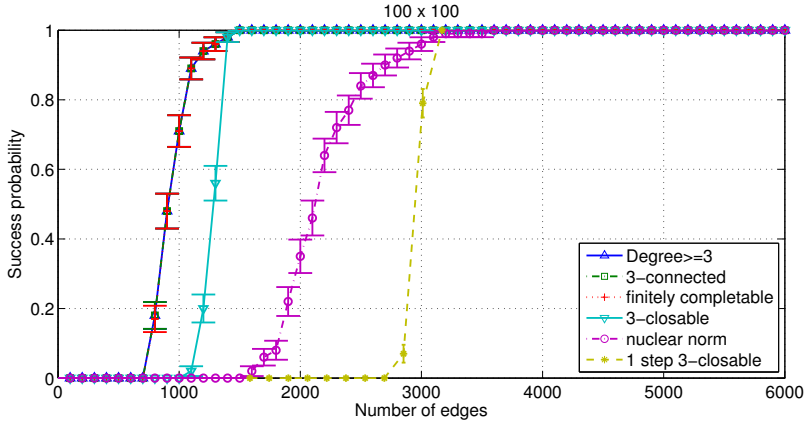


Figure 3: Phase transition curves of various conditions for 100×100 matrices at rank 3.

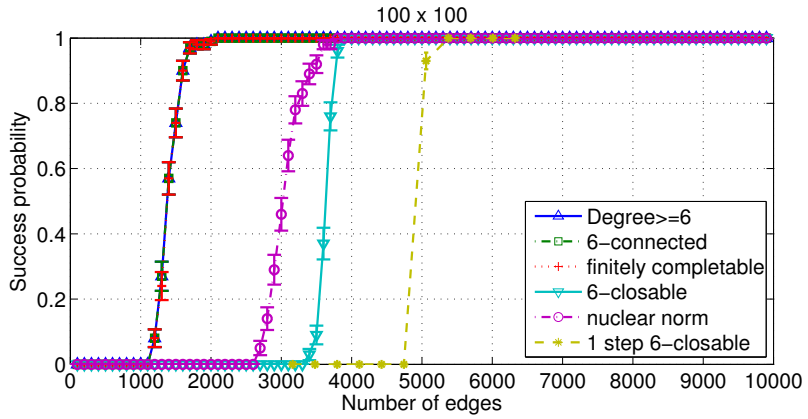


Figure 4: Phase transition curves of various conditions for 100×100 matrices at rank 6.

Figure 4 shows the same plot as above for 100×100 matrices at rank 6. The success probabilities of the (a) minimum degree, (b) r-connected, (c) completable are again almost the same, and exceeds chance probability 0.5 around $|E| \simeq 1,400$. On the other hand, the number of entries required for minor closability is at least $3,700$. This is because the masks that we need to handle around the optimal sampling density is so large and sparse that we cannot hope to find a 6×6 biclique required by the minor clusre algorithm to even get started. The nuclear norm minimization required about $3,100$ samples.

Figure 5 shows the phase transition from a non-completable mask to a completable mask for almost $2r$-regular random masks. Here we first randomly sampled $2r$-regular $(n \times n)$- masks using Steger & Wormald algorithm (Steger and Wormald, 1999). Next we randomly permuted the edges included in the mask and the edges not included in the mask independently and concatenated them into a single list of edges. In this way, we obtained a length mn ordered list of edges that become $2r$-regular exactly at the $2rn$th edge. For

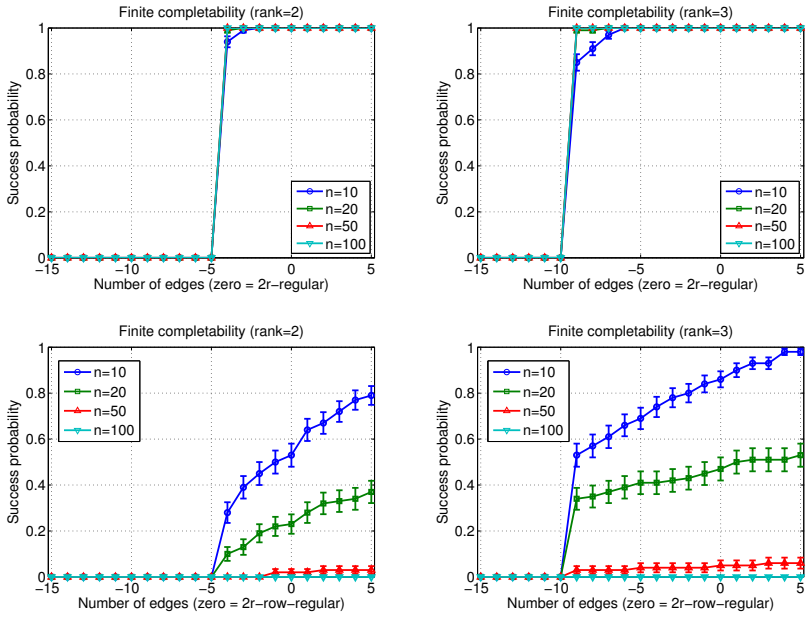


Figure 5: Phase transition in an almost regular mask.

each ordered list sampled this way, we took the first $2rn + i$ edges and checked whether the mask corresponding to these edges was completable for $i = -15, -14, \ldots, 5$. This procedure was repeated 100 times and averaged to obtain a probability estimate. In order to make sure that the phase transition is indeed caused by the regularity of the mask, we conducted the same experiment with row-wise $2r$-regular masks, i.e., each row of the mask contained exactly $2r$ entries while the number of non-zero entries varied from a column to another.

In Figure 5, the phase transition curves for different n at rank 2 and 3 are shown. The two plots in the top part show the results for the $2r$-regular masks, and the two plots in the bottom show the same results for the $2r$-row-wise regular masks. For the $2r$-regular masks, the success probability of completability sharply rises when the number of edges exceeds $2rn - r^2$ ($i = -4$ for $r = 2$ and $i = -9$ for $r = 3$); the phase transition is already rather sharp for $n = 10$ and for $n \geq 20$ it becomes almost zero or one. On the other hand, the success probabilities for the $2r$-row-wise regular masks grow rather slowly and approach zero for large n. This is natural, since it is likely for large n that there is some column with non-zero entries less than r, which violates the necessary conditions in Corollary 39.

7.3 Completability of the MovieLens data set

This section is devoted to studying a well-known data set - the MovieLens data published by GroupLens - with the methods developed in this paper. We demonstrated how the algorithms given above can be used to make statements about the sets of entries which are (a) completable, (b) uniquely completable, and (c) not completable with any algorithm.

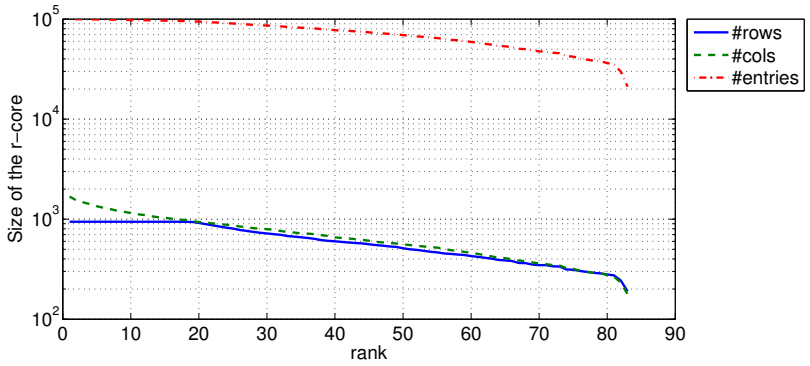


Figure 6: Size of the r-core of the MovieLens 100k data set for varying r. For each rank r, the figure shows the number of rows (solid blue), the number of columns (dashed green), and the number of entries (dash-dotted red) in the r-core of the mask corresponding to the observed entries of the MovieLens 100k data set. The biggest rank with non-empty r-core is $r = 83$.

The underlying data set for the following analyses is the MovieLens 100k data set. By convention, columns will correspond to the 1682 movies, while the rows will correspond to the 943 users in the data set.

For growing rank r, the r-core of the MovieLens data set was computed by the algorithm which is standard in graph theory - by Theorem 44 only the missing entries in the r-core can be completed, and any entry not contained in the r-core is not completable by any algorithm. Figure 6 shows the size (columns, rows, entries) of the r-core of the MovieLens data for growing r.

Under rank 18, the vast majority of the entries are in the r-core, and so is the majority of the rows, while some columns with very few entries are removed with increasing r. At rank $r = 18$, the number of columns in the r-core attains the number of rows in the r-core; above rank 18, the number of rows and columns in the r-core diminish exponentially with the same speed. Above rank 79, the r-core rapidly starts to shrink, with $r = 83$ being the biggest rank with non-empty r-core.

For growing rank r, the finitely completable closure $\mathrm{cl}_r(E)$ in the MovieLens data set was identified in the following way: First, it was checked with Algorithm 1 whether the 83-core was r-completable. If not, the completable entries in the 83-core were computed by an implementation of Algorithm 1. Then, the minor closure of the completed 83-cores was computed by Algorithm 4; by Theorem 44, it was sufficient to check for completable entries in the r-core. Note that the positions of the completable entries were also computed in the process.

Figure 7 shows the number of completable entries in the MovieLens data set for growing r determined in this way.

An interesting thing to note is the inflection point at rank $r = 18$. It corresponds to the phase transition in Figure 6 where the r-core starts to shrink exponentially and

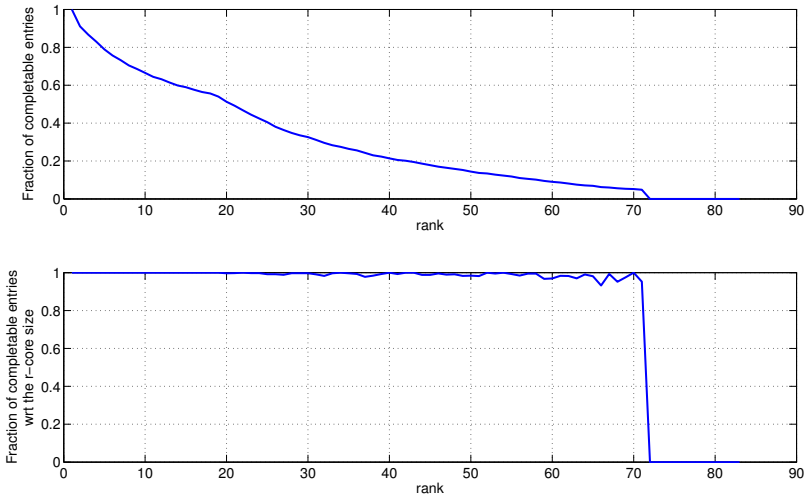


Figure 7: Number of completable entries in the MovieLens 100k data set for varying r; observed entries are not counted as completable, only completable entries which are not observed. For each rank r, the upper figure shows the number of completable entries, as a fraction of all missing entries. The lower figure shows the number of completable entries, as a fraction of the missing entries in the r-core. For $r \geq 84$, the r-core is empty, thus no missing entries can be completed, see Figure 6.

simultaneously in rows and columns. At rank $r = 72$ and above, no missing entry in the 83-core can be completed.

7.4 Entry-wise completion and error prediction

In the rest of the experiments, we recapitulate some results from Kiraly and Theran (2013) on entry-wise reconstruction and error prediction for rank one matrices.

To test reconstruction, we generated 10 random masks of size 50×50 with 200 entries sampled uniformly and a random (50×50) matrix of rank one. The multiplicative noise was chosen entry-wise independent, with variance $\sigma_i = (i-1)/10$ for each entry. Figure 9(a) compares the Mean Squared Error (MSE) for three algorithms: Nuclear Norm (using the implementation Tomioka et al. (2010)), OptSpace (Keshavan et al., 2010), and Algorithm 6. It can be seen that on these masks, Algorithm 6 is competitive with the other methods and even outperforms them for low noise.

Figure 9(b) compares the error of each of the methods with the variance predicted by Algorithm 7 each time the noise level changed. The figure shows that for any of the algorithms, the mean of the actual error increases with the predicted error, showing that the error estimate is useful for a-priori prediction of the actual error - independently of the particular algorithm. Note that by construction of the data this statement holds in particular for entry-wise predictions. Furthermore, in quantitative comparison Algorithm 7 also outperforms the other two in each of the bins. The qualitative reversal between the

algorithms in Figures 9(b) (a) and (b) comes from the different error measure and the conditioning on the bins.

7.5 Universal error estimates

For three different masks, we calculated the predicted minimum variance for each entry of the mask. The mask sizes are all 140×140. The noise was assumed to be i.i.d. Gaussian multiplicative with $\sigma_e = 1$ for each entry. Figure 8 shows the predicted a-priori minimum variances for each of the masks. The structure of the mask affects the expected error. Known entries generally have least variance, and it is less than the initial variance of 1, which implies that the (independent) estimates coming from other paths can be used to successfully denoise observed data. For unknown entries, the structure of the mask is mirrored in the pattern of the predicted errors; a diffuse mask gives a similar error on each missing entry, while the more structured masks have structured error which is determined by combinatorial properties of the completion graph.

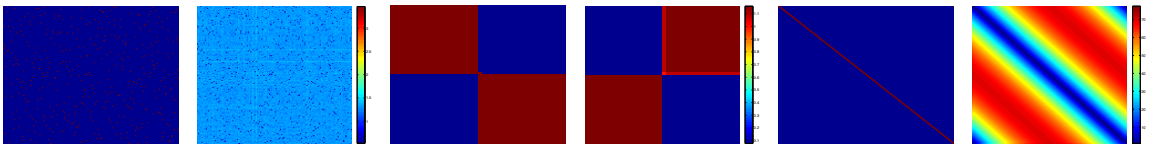


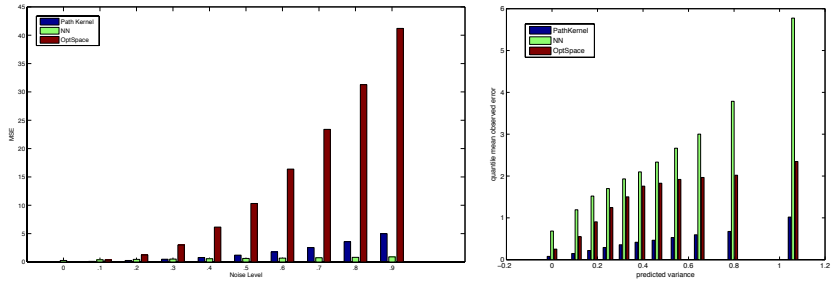
Figure 8: The figure shows three pairs of masks and predicted variances. A pair consists of two adjacent squares. The left half is the mask which is depicted by red/blue heatmap with red entries known and blue unknown. The right half is a multicolor heatmap with color scale, showing the predicted variance of the completion. Variances were calculated by our implementation of Algorithm 7.

8. Discussion and Outlook

In this paper we have demonstrated the usefulness and practicability of the algebraic combinatorial approach for matrix completion, by deriving reconstructability statements, and actual reconstruction algorithms for single missing entries. Our theory allows to treat the positions of the observations separately from the entries themselves. As a prominent model feature, we are able to separate the sampling scheme from algebraic and combinatorial conditions for reconstruction and explain existing reconstruction bounds by the combinatorial phase transition for the uniform random sampling scheme.

The discussed framework provides the foundation for a number of **novel matrix completion strategies for the practitioner**:

- The presented algorithms allow for **entry-wise error estimates** which are independent of the method. More precisely, as it has been studied by Kiraly and Theran (2013) for rank 1, the algorithm of actual reconstruction can be separated from the question whether the entry is reconstructible, and with which error, allowing the com-



(a) mean squared errors (b) error vs. predicted variance

Figure 9: For 10 randomly chosen masks and 50×50 true matrix, matrix completions were performed with Nuclear Norm (green), OptSpace (red), and Algorithm 6 (blue) under multiplicative noise with variance increasing in increments of 0.1. For each completed entry, minimum variances were predicted by Algorithm 7. 9(a) shows the mean squared error of the three algorithms for each noise level, coded by the algorithms' respective colors. 9(b) shows a bin-plot of errors (y-axis) versus predicted variances (x-axis) for each of the three algorithms: for each completed entry, a pair (predicted error, true error) was calculated, predicted error being the predicted variance, and the actual prediction error being the squared logarithmic error (i.e., $(\log|a_{true}| - \log|a_{predicted}|)^2$ for an entry a). Then, the points were binned into 11 bins with equal numbers of points. The figure shows the mean of the errors (second coordinate) of the value pairs with predicted variance (first coordinate) in each of the bins, the color corresponds to the particular algorithm; each group of bars is centered on the minimum value of the associated bin.

bination of any reconstruction algorithm with reconstruction bounds obtained from our framework.

- The presented ideas allow **completion/denoising of single entries** in the practically relevant case where only one entry or a subset of all entries should be reconstructed or denoised. A rank one method has been presented by Kiraly and Theran (2013), the case of rank 2 and higher is studied by Blythe et al. (2014).
- The use of circuits for reconstruction pave the way for **local completion/denoising**, that is, a good reconstruction can be obtained from a small combinatorial neighborhood of entries which can be determined from the theory (and which is not necessarily a submatrix), allowing to avoid processing of the whole matrix - which is especially desirable if the matrix is huge.

In our new setting, we are also left with a number of major **open questions**:

- Characterize all **circuits and circuit polynomials** in rank 2 or higher.
- Give a sufficient and necessary combinatorial **criterion for unique completability**.

- Give an **efficient**[4] **algorithm certifying for unique completability** when given the positions of the observed entries (or, more generally, one which computes the number of solutions).
- Prove the **phase transition bound for the completable core** (the phase transition bound for completability has been shown in Király and Theran, 2013).
- Explain the existing **guarantees for whole matrix reconstruction MSE in terms of single entry expected error**, for the various sampling models in literature (an explanation for rank one can be inferred from Kiraly and Theran, 2013).

Finally, our presented results suggest a number of **future directions**:

- Problems such as **matrix completion under further constraints** such as for **symmetric matrices**, **distance matrices** or **kernel matrices**, are closely related to the ones we consider here, and can be treated by similar techniques. Under a phase transition aspect, these models were studied by Király and Theran (2013); for general matroids, the theory in (Király et al., 2013) yields a starting point.
- **Completion of tensors** is a natural generalization of matrix completion and accessible to the techniques presented here or in (Király and Theran, 2013; Király et al., 2013).
- We have essentially shown matrix completion to be an **algebraic manifold learning problem**. This makes it accessible to the kernel/ideal learning techniques presented in (Király et al., 2014).
- The algebraic theory used to infer genericity and identifiability is largely independent of the matrix completion setting and can be applied in a very general context of **compressed sensing, identifiability and inverse problems that are algebraic**. For a more detailed discussion and some related problems, see Section 3.4.

Summarizing, we argue that recognizing and exploiting algebra and combinatorics in machine learning problems is beneficial from the practical and theoretical perspectives. When it is present, methods using underlying algebraic and combinatorial structures yield sounder statements and more practical algorithms than can be obtained when ignoring it, conversely algebra and combinatorics can profit from the various interesting structure surfacing in machine learning problems. Therefore all involved fields can only profit from a more widespread interdisciplinary collaboration with and between each other.

Acknowledgments

We thank Andriy Bondarenko, Winfried Bruns, Eyke Hüllermeyer, Mihyun Kang, Yusuke Kobayashi, Martin Kreuzer, Cris Moore, Klaus-Robert Müller, Kazuo Murota, Kiyohito Nagano, Zvi Rosen, Raman Sanyal, Bernd Sturmfels, Sumio Watanabe, Volkmar Welker, and Günter Ziegler for valuable discussions. RT is partially supported by MEXT KAKENHI

4. say polynomial time, success with high probability

22700138, the Global COE program "The Research and Training Center for New Development in Mathematics", FK by Mathematisches Forschungsinstitut Oberwolfach (MFO), and LT by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement no 247029-SDModels. This research was partially carried out at MFO, supported by FK's Oberwolfach Leibniz Fellowship.

Appendix A. Algebraic Geometry Fundamentals

This section collects some algebraic geometric tools used in the main corpus.

A.1 Algebraic Genericity

We will briefly review the concept of genericity for our purposes. Intuitively, algebraic genericity describes that some statements holds for almost all objects, with the exceptions having an algebraic structure. The following results will be stated for algebraic varieties over the real or complex numbers, that is, over the field $\mathbb{K}$, where $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$.

Definition 54 *Let $\mathcal{Y} \subseteq \mathbb{K}^n$ be an algebraic variety. Let P be some property of points $y \in \mathcal{Y}$. Write $P(\mathcal{Y}) = \{y \in \mathcal{Y} \,:\, y \text{ has property } P\}$, and $\neg P(\mathcal{Y}) = \mathcal{Y} \setminus P(\mathcal{Y})$.*

(i) *We call P an* open condition *if $P(\mathcal{Y})$ is a Zariski open subset of $\mathcal{Y}$.*

(ii) *We call P a* Zariski-generic condition *if there is an open dense subset $U \subseteq \mathcal{Y}$ such that $U \subseteq P(\mathcal{Y})$.*

(iii) *We call P a* Hausdorff-generic condition *if $\neg P(\mathcal{Y})$ is a $\mathcal{Y}$-Hausdorff zero set.*

The different types of conditions above can be put in relation to each other:

Proposition 55 *Keep the notation of Definition 54.*

(i) *If P is a Zariski-generic condition, then P is a Hausdorff-generic condition as well.*

(ii) *Assume $\mathcal{Y}$ is irreducible, and $P(\mathcal{Y})$ is non-empty. If P is an open condition, then it is a Zariski-generic condition.*

(iii) *Assume $\mathcal{Y}$ is irreducible, and $P(\mathcal{Y})$ is constructible in the Zariski topology, i.e., can be written as finite union and intersection of open and closed sets. If P is a Hausdorff-generic condition, then it is a Zariski-generic condition as well.*

Proof (i) follows from the fact that Zariski closed sets of smaller Krull dimension are Hausdorff zero sets.
(ii) follows from the fact that non-empty Zariski open sets are dense in an irreducible algebraic set.
(iii) as $P(\mathcal{Y})$ is Zariski-constructible, it will have positive Hausdorff measure if and only if it contains a non-empty (relatively Zariski) open set. The statement then follows from (ii). ■

Furthermore, Hausdorff-genericity is essentially states that the condition holds, universally with probability one:

Proposition 56 *Keep the notation of Definition 54. The following are equivalent:*

(i) *P is a Hausdorff-generic condition.*

(ii) *For all Hausdorff-continuous random variables X taking values in $\mathcal{Y}$, the statement $P(X)$ holds with probability one.*

Proof (i)$\Leftrightarrow$ (ii) follows from taking Radon-Nikodym derivatives. ■

All relevant properties and conditions which are referenced from the main corpus describe (a) irreducible varieties - in this case, the determinantal variety, and (b) are Zariski-constructible. Therefore, by Proposition 55, all three definitions agree for the purpose of this paper. The terminology used in the paper can be given as follows in the above definitions:

Definition 57 *Let $\mathcal{Y} \subseteq \mathbb{K}^n$ be an algebraic variety. Let P be some property of points $y \in \mathcal{Y}$. We say "a generic $y \in \mathcal{Y}$ has property P" if P is a Hausdorff-generic condition for points in $\mathcal{Y}$.*

A.2 Open Conditions and Generic Properties of Morphisms

In this section, we will summarize some algebraic geometry results used in the main corpus. The following results will always be stated for algebraic varieties over $\mathbb{C}$.

Proposition 58 *Let $f : \mathcal{X} \to \mathcal{Y}$ be a morphism of algebraic varieties (over any field). Then, if $\mathcal{X}$ is irreducible, so is $f(\mathcal{X})$. In particular, if f is surjective, and $\mathcal{X}$ is irreducible, then $\mathcal{Y}$ also is.*

Proof This is classical and follows directly from the fact that morphisms of algebraic varieties are continuous in the Zariski topology. ■

Theorem 59 *Let $f : \mathcal{X} \to \mathcal{Y}$ be a morphism of algebraic varieties. The function*

$$\mathcal{Y} \to \mathbb{N}, \quad y \mapsto \dim f^{-1}(y)$$

is upper semicontinuous in the Zariski topology.

Proof This follows from (Grothendieck and Dieudonné, 1966, Théorème 13.1.3). ■

Proposition 60 *Let $f : \mathcal{X} \to \mathcal{Y}$ be a morphism of algebraic varieties, with $\mathcal{Y}$ be irreducible. Then, there is an open dense subset $V \subseteq \mathcal{Y}$ such that $f : U \to V$, where $U = f^{-1}(V)$, is a flat morphism.*

Proof This follows from (Grothendieck and Dieudonné, 1965, Théorème 6.9.1). ■

Theorem 61 *Let $f : \mathcal{X} \to \mathcal{Y}$ be a morphism of algebraic varieties. Let $d, \nu \in \mathbb{N}$. Then, the following are open conditions for $y \in \mathcal{Y}$:*

(i) $\dim f^{-1}(y) \leq d$.

(ii) *f is unramified over y.*

(iii) *f is unramified over y, and the number of irreducible components of $f^{-1}(y)$ equals ν.*

In particular, if f is surjective, then the following is an open property as well:

(iv) *f is unramified over y, and $\left|f^{-1}(y)\right| = \nu$, for some $\nu \in \mathbb{N}$.*

Proof (i) follows from (Grothendieck and Dieudonné, 1965, Corollaire 6.1.2).
(ii) follows from (Grothendieck and Dieudonné, 1966, Théorème 12.2.4(v)).
(iii) follows from (Grothendieck and Dieudonné, 1966, Théorème 12.2.4(vi)).
(iv) follows from (i), applied in the case $\dim f^{-1}(y) \leq 0$ which is equivalent to $\dim f^{-1}(y) = 0$ due to surjectivity of f, and (iii). ■

Corollary 62 *Let $f : \mathcal{X} \to \mathcal{Y}$ be a generically unramified and surjective morphism of algebraic varieties, with $\mathcal{Y}$ be irreducible. Then, there are unique $d, \nu \in \mathbb{N}$ such that the following sets are Zariski closed, proper subsets of $\mathcal{Y}$ (and therefore Hausdorff zero sets):*

(i) $\{y \,:\, \dim f^{-1}(y) \neq d\}$

(ii) $\{y \,:\, f \text{ is ramified at } y\}$

(iii) $\{y \,:\, f \text{ is ramified at } y\} \cup \{y \,:\, \left|f^{-1}(y)\right| \neq \nu\}$

Proof This is implied by Theorem 61 (i), (ii) and (iii), using that a non-zero open subset of the irreducible variety $\mathcal{Y}$ must be open dense, therefore its complement in $\mathcal{Y}$ a closed and a proper subset of $\mathcal{Y}$. ■

Proposition 63 *Let $f : \mathcal{X} \to \mathcal{Y}$ be a morphism of algebraic varieties, with $\mathcal{Y}$ irreducible. Then, the following are equivalent:*

(i) *f is unramified over y and $\left|f^{-1}(y)\right| = \nu$.*

(ii) *There is a Borel open neighborhood $U \subseteq \mathcal{Y}$ of $y \in U$, such that f is unramified over U and $\left|f^{-1}(z)\right| = \nu$ for all $z \in U$.*

(iii) *There is a Zariski open neighborhood $U \subseteq \mathcal{Y}$ of $y \in U$, dense in $\mathcal{Y}$, such that f is unramified over U and $\left|f^{-1}(z)\right| = \nu$ for all $z \in U$.*

Proof The equivalence is implied by Corollary 62 and the fact that $\mathcal{Y}$ is irreducible. Note that either condition implies that f is generically unramified due to Theorem 61 (ii) and irreducibility of $\mathcal{Y}$. ■

A.3 Real versus Complex Genericity

We derive some elementary results how generic properties over the complex and real numbers relate. While some could be taken for known results, they appear not to be folklore - except maybe Lemma 65. In any case, they seem not to be written up properly in literature known to the authors. A first version of the statements below has also appeared as part of Király and Ehler (2014).

Definition 64 *Let $\mathcal{X} \subseteq \mathbb{C}^n$ be a variety. We define the* real part *of $\mathcal{X}$ to be $\mathcal{X}_\mathbb{R} := \mathcal{X} \cap \mathbb{R}^n$.*

Lemma 65 *Let $\mathcal{X} \subseteq \mathbb{C}^n$ be a variety. Then, $\dim \mathcal{X}_\mathbb{R} \leq \dim \mathcal{X}$, where $\dim \mathcal{X}_\mathbb{R}$ denotes the Krull dimension of $\mathcal{X}_\mathbb{R}$, regarded as a (real) subvariety of $\mathbb{R}^n$, and $\dim \mathcal{X}$ the Krull dimension of $\mathcal{X}$, regarded as subvariety of $\mathbb{C}^n$.*

Proof Let $k = n - \dim \mathcal{X}$. By (Mumford, 1999, Section 1.1), $\mathcal{X}$ is contained in some complete intersection variety $\mathcal{X}' = \mathrm{V}(f_1, \ldots, f_k)$. That is $(f_1, \ldots, f_k)$ is a complete intersection, with $f_i \in \mathbb{C}[X_1, \ldots, X_n]$ and $\dim \mathcal{X}' = \dim \mathcal{X}$, such that f_i is a non-zero divisor modulo $f_1, \ldots, f_{i-1}$. Define $g_i := f_i \cdot f_i^*$, one checks that $g_i \in \mathbb{R}[X_1, \ldots, X_n]$, and define $\mathcal{Y} := \mathrm{V}(g_1, \ldots, g_k)$ and $\mathcal{Y}_\mathbb{R} := \mathcal{Y} \cap \mathbb{R}^n$. The fact that f_i is a non-zero divisor modulo $f_1, \ldots, f_{i-1}$ implies that g_i is a non-zero divisor modulo $g_1, \ldots, g_{i-1}$; since $g_i \cdot h \cong 0$ modulo $g_1, \ldots, g_{i-1}$ implies $f_i \cdot (h \cdot f_i^*) \cong 0$ modulo $f_1, \ldots, f_{i-1}$. Therefore, $\dim \mathcal{Y}_\mathbb{R} \leq \dim \mathcal{X}$; by construction, $\mathcal{X}' \subseteq \mathcal{Y}$, and $\mathcal{X} \subseteq \mathcal{X}'$, therefore $\mathcal{X}_\mathbb{R} \subseteq \mathcal{Y}_\mathbb{R}$, and thus $\dim \mathcal{X}_\mathbb{R} \leq \dim \mathcal{Y}_\mathbb{R}$. Combining it with the above inequality yields the claim. ■

Definition 66 *Let $\mathcal{X} \subseteq \mathbb{C}^n$ be a variety. If $\dim \mathcal{X} = \dim \mathcal{X}_\mathbb{R}$, we call $\mathcal{X}$* observable over the reals*. If $\mathcal{X}$ equals the (complex) Zariski-closure of $\mathcal{X}_\mathbb{R}$, we call $\mathcal{X}$* defined over the reals*.*

Proposition 67 *Let $\mathcal{X} \subseteq \mathbb{C}^n$ be a variety.*

(i) *If $\mathcal{X}$ is defined over the reals, then $\mathcal{X}$ is also observable over the reals.*

(ii) *The converse of (i) is false.*

(iii) *If $\mathcal{X}$ irreducible and observable over the reals, then $\mathcal{X}$ is defined over the reals.*

Proof (i) Let $k = n - \dim \mathcal{X}_\mathbb{R}$. By (Mumford, 1999, Section 1.1), $\mathcal{X}_\mathbb{R}$ is contained in some complete intersection variety $\mathcal{X}' = \mathrm{V}(f_1, \ldots, f_k)$, with $f_i \in \mathbb{R}[X_1, \ldots, X_n]$ a complete intersection. By an argument, analogous to the proof of Lemma 65, one sees that the f_i are a complete intersection in $\mathbb{C}[X_1, \ldots, X_n]$ as well. Since the Zariski-closure of $\mathcal{X}_\mathbb{R}$ and $\mathcal{X}$ are equal, it holds that $f_i \in \mathrm{I}(\mathcal{X})$. Therefore, $\mathcal{X} \subseteq \mathrm{V}(f_1, \ldots, f_k)$, which imples $\dim \mathcal{X} \leq n - k$, and by definition of k, as well $\dim \mathcal{X} \leq \dim \mathcal{X}_\mathbb{R}$. With Lemma 65, we obtain $\dim \mathcal{X}_\mathbb{R} = \dim \mathcal{X}$, which was the statement to prove.
(ii) It suffices to give a counterexample: $\mathcal{X} = \{1, i\} \subseteq \mathbb{C}$. Alternatively (in a context where $\varnothing$ is not a variety) $\mathcal{X} = \{(1, x) \,:\, x \in \mathbb{C}\} \cup \{(i, x) \,:\, x \in \mathbb{C}\} \subseteq \mathbb{C}^2$.
(iii) By definition of dimension, Zariski-closure preserves dimension. Therefore, the closure $\overline{\mathcal{X}_\mathbb{R}}$ is a sub-variety of $\mathcal{X}$, with $\dim \overline{\mathcal{X}_\mathbb{R}} = \dim \mathcal{X}$. Since $\mathcal{X}$ is irreducible, equality $\overline{\mathcal{X}_\mathbb{R}} = \mathcal{X}$ must hold. ■

Theorem 68 *Let $\mathcal{X} \subseteq \mathbb{C}^n$ be an irreducible variety which is observable over the reals, let $\mathcal{X}_\mathbb{R}$ be its real part. Let P be an algebraic property. Assume that a generic $x \in \mathcal{X}$ is P. Then, a generic $x \in \mathcal{X}_\mathbb{R}$ has property P as well.*

Proof Since P is an algebraic property, the P points of $\mathcal{X}$ are contained in a proper sub-variety $\mathcal{Z} \subseteq \mathcal{X}$, with $\dim \mathcal{Z} \lneq \dim \mathcal{X}$. Since $\mathcal{X}$ is observable over the reals, it holds $\dim \mathcal{X} = \dim \mathcal{X}_\mathbb{R}$. By Lemma 65, $\dim \mathcal{Z}_\mathbb{R} \leq \dim \mathcal{Z}$. Putting all (in-)equalities together, one obtains $\dim \mathcal{Z}_\mathbb{R} \lneq \dim \mathcal{X}_\mathbb{R}$. Therefore, the $\mathcal{Z}_\mathbb{R}$ is a proper sub-variety of $\mathcal{X}_\mathbb{R}$; and the P points of $\mathcal{X}_\mathbb{R}$ are contained in it - this proves the statement. ■

A.4 Algebraic Properties of the Masking

We conclude with checking the conditions previously discussed in the specific case of the masking:

Proposition 69 *For $E \subseteq \mathcal{E}$, consider the determinantal variety $\mathcal{M}(m \times n, r)$ (over $\mathbb{C}$), and the masking*

$$\Omega : \mathcal{M}(m \times n, r) \to \mathbb{C}^{|E|}, \quad \mathbf{A} \mapsto \{A_e, e \in E\}.$$

(i) *The determinantal variety $\mathcal{M}(m \times n, r)$ is irreducible.*

(ii) *The determinantal variety $\mathcal{M}(m \times n, r)$ is observable over the reals.*

(iii) *The determinantal variety $\mathcal{M}(m \times n, r)$ is defined over the reals.*

(iv) *The variety $\Omega(\mathcal{M}(m \times n, r))$ is irreducible.*

(v) *The map Ω is generically unramified.*

Proof (i) follows from Proposition 58, applied to the surjective map

$$\Upsilon : \mathbb{C}^{m \times r} \times \mathbb{C}^{n \times r} \to \mathcal{M}(m \times n, r),\ (U, V) \mapsto UV^\top,$$

and irreducibility of affine space $\mathbb{C}^{m \times r} \times \mathbb{C}^{n \times r}$.
(ii) follows from considering the map Υ over the reals, observing that the rank its Jacobian is not affected by this.
(iii) follows from (i), (ii) and Proposition 67 (iii).
(iv) follows from (i) and Proposition 58, applied to Ω.
(v) follows from the fact that Ω is a coordinate projection, therefore linear. ■

Appendix B. Advanced Algorithm for Minor Closure

Takeaki Uno
uno@nii.jp
National Institute of Informatics
Tokyo 101-8430, Japan

B.1 Closability and the r-Closure

A crucial step in minor closure algorithm 4 is to find a $r \times r$ biclique in a (sub)graph(V, W, E).

Algorithm 8 can find an $d_1 \times d_2$ biclique efficiently based on two ideas: (i) iterate over row vertices and (ii) work only on the (d_2, d_1)-core; here (d_2, d_1)-core is the maximal subgraph of (V, W, E) that the degrees of the row and column vertices are at least d_2 and d_1, respectively.

A naive approach for finding an $r \times r$ biclique might be to iterate over edges in E and for each edge $(v, w) \in E$ whose nodes have at least $r - 1$ neighbors, check whether the subgraph induced by $(N(w), N(v))$ contains an $r-1 \times r-1$ biclique. Instead our algorithm iterates over nodes in V and for each node $v \in V$, check whether the subgraph induced by $(V', N(v))$ contains an $r - 1 \times r$ biclique, where V' is defined by removing all previously attempted nodes and the current v from V. Iterating over row vertices results in smaller number of iterations because $|V| \leq |E|$, and allows us to avoid double checking, because previously attempted nodes can be removed from V'. Concentrating on the (d_2, d_1)-core is natural, because no $d_1 \times d_2$ biclique contains row or column vertex with degree less than d_2 or d_1, respectively. We present the pruning step for finding the (d_2, d_1)-core in Algorithm 9.

Algorithm 8 `FindAClique`$((V, W, E), d_1, d_2)$

Inputs: bipartite graph (V, W, E), size of the bipartite clique to be found $d_1 \times d_2$.
Output: vertex sets of a clique (I, J).
$(V, W, E) \leftarrow$ `FindCore`$((V, W, E), d_2, d_1)$.
if $|V| < d_1$ or $|W| < d_2$ **then**
    Return $(\emptyset, \emptyset)$.
end if
$V' \leftarrow V$.
for each $v \in V$ **do**
    if $d_1 = 1$ and $|N(v)| \geq d_2$ **then**
        Return $(\{v\}, N(v))$.
    end if
    $V' \leftarrow V' \backslash \{v\}$, $W' \leftarrow N(v)$, $E' \leftarrow (V' \times W') \cap E$.
    $(I', J') \leftarrow$ `FindAClique`$((V', W', E'), d_1 - 1, d_2)$.
    if $|I'| > 0$ and $|J'| > 0$ **then**
        Return $(I' \cup \{v\}, J')$.
    end if
end for
Return $(\emptyset, \emptyset)$.

References

Evrim Acar, Daniel M. Dunlavy, and Tamara G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*, pages 262–269. IEEE, 2009.

Elizabeth S. Allman, Catherine Matias, and John A. Rhodes. Identifiability of parameters in latent structure models with many observed variables. *Annals of Statistics*, 37(6A):3099–3132, 2009.

Algorithm 9 `FindCore`$((V, W, E), d_1, d_2)$

Inputs: bipartite graph (V, W, E), minimum degree d_1 for the row vertices and d_2 for the column vertices.
Output: pruned bipartite graph (V, W, E).
while true **do**
  $V' \leftarrow \{v \in V : |N(v)| < d_1\}$.
  $W' \leftarrow \{w \in W : |N(w)| < d_2\}$.
  if $V' = \emptyset$ and $W' = \emptyset$ **then**
    Return $(V, W, (V \times W) \cap E)$.
  end if
  $V \leftarrow V \backslash V'$.
  $W \leftarrow W \backslash W'$.
end while

Andreas Argyriou, Craig A. Micchelli, Massimiliano Pontil, and Yi Ying. A spectral regularization framework for multi-task structure learning. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in NIPS 20*, pages 25–32. MIT Press, Cambridge, MA, 2008.

Donald Bamber. How many parameters can a model have and still be testable? *Journal of Mathematical Psychology*, 29(4):443 – 473, 1985.

Srinadh Bhojanapalli and Prateek Jain. Universal matrix completion. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*. MIT Press, 2014.

Duncan Blythe and Franz J. Király. Prediction and quantification of individual athletic performance. *arXiv e-prints*, 2015. arXiv 1505.01147.

Duncan Blythe, Franz J. Király, and Louis Theran. Algebraic combinatorial methods for low-rank matrix completion with application to athletic performance prediction. *arXiv e-prints*, 2014. arXiv 1406.2864.

Béla Bollobás. *Random graphs*, volume 73 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, second edition, 2001.

Winfried Bruns and Udo Vetter. *Determinantal Rings*. Springer-Verlag New York, Inc., 1988.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics. The Journal of the Society for the Foundations of Computational Mathematics*, 9(6):717–772, 2009. ISSN 1615-3375.

Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 56(5):2053–2080, 2010.

Robert Connelly. Generic global rigidity. *Discrete Comput. Geom.*, 33(4):549–563, 2005. ISSN 0179-5376. doi: 10.1007/s00454-004-1124-4. URL `http://dx.doi.org/10.1007/s00454-004-1124-4`.

Andreas Dress and László Lovász. On some combinatorial properties of algebraic matroids. *Combinatorica*, 7(1):39–48, 1987.

Rina Foygel and Nathan Srebro. Concentration-based guarantees for low-rank matrix reconstruction. *arXiv e-prints*, 2011. arXiv:1102.3923.

Andrew Goldberg, Xiaojin Zhu, Benjamin Recht, Jun-Ming Xu, and Robert Nowak. Transduction with matrix completion: Three birds with one stone. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS) 23*, pages 757–765. 2010.

Alexander Grothendieck and Jean Dieudonné. Éléments de géométrie algébrique IV, deuxième partie. *Publications mathématiques de l'IHÉS*, 24, 1965.

Alexander Grothendieck and Jean Dieudonné. Éléments de géométrie algébrique IV, troisième partie. *Publications mathématiques de l'IHÉS*, 28, 1966.

GroupLens. Movielens 100k data set. Available online at `http://grouplens.org/datasets/movielens/`; as downloaded on November 27th 2012.

Daniel Hsu, Sham Kakade, and Percy Liang. Identifiability and unmixing of latent parse trees. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems (NIPS) 25*, pages 1520–1528. 2012.

Bill Jackson, Brigitte Servatius, and Herman Servatius. The 2-dimensional rigidity of certain families of graphs. *Journal of Graph Theory*, 54(2):154–166, 2007. ISSN 0364-9024.

Svante Janson and Malwina J. Luczak. A simple solution to the k-core problem. *Random Structures Algorithms*, 30(1-2):50–62, 2007.

Shiva Prasad Kasiviswanathan, Cristopher Moore, and Louis Theran. The rigidity transition in random graphs. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1237–1252, Philadelphia, PA, 2011. SIAM.

Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *Institute of Electrical and Electronics Engineers. Transactions on Information Theory*, 56 (6):2980–2998, 2010.

Franz J. Király and Martin Ehler. The algebraic approach to phase retrieval and explicit inversion at the identifiability threshold. *arXiv e-prints*, 2014. arXiv:1402.4053.

Franz J. Király and Louis Theran. Coherence and sufficient sampling densities for reconstruction in compressed sensing. *arXiv e-prints*, 2013. arXiv 1302.2767.

Franz J. Kiraly and Louis Theran. Error-minimizing estimates and universal entry-wise error bounds for low-rank matrix completion. *Advances in Neural Information Processing Systems (NIPS) 26*, pages 2364–2372, 2013.

Franz J. Király, Louis Theran, Ryota Tomioka, and Takeaki UNo. The algebraic combinatorial approach for low-rank matrix completion. *arXiv e-prints*, 2013. arXiv 1211.4116v3.

Franz J. Király, Martin Kreuzer, and Louis Theran. Learning with cross-kernel matrices and ideal PCA. *arXiv e-prints*, 2014. arXiv:1406.2646.

Franz J. Király, Zvi Rosen, and Louis Theran. Algebraic matroids with graph symmetry. *arXiv e-prints*, 2013. arXiv:1312.3777.

Krysztof Kurdyka, Patrice Orro, and S. Simon. Semialgebraic Sard theorem for generalized critical values. *Journal of Differential Geometry*, 56(1):67–92, 2000.

Troy Lee and Adi Shraibman. Matrix completion from any given set of observations. In *Advances in Neural Information Processing Systems (NIPS) 26*, pages 1781–1787, 2013.

Adam Mahdi, Nicolette Meshkat, and Seth Sullivant. Structural identifiability of viscoelastic mechanical systems. *PLOS ONE*, 9(2):e86411, 2014.

Raghu Meka, Prateek Jain, and Inderjit S. Dhillon. Guaranteed rank minimization via singular value projection. *arXiv e-prints*, 2009. arXiv:0909.5457.

Aditya K. Menon and Charles Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.

Nicolette Meshkat, Seth Sullivant, and Marisa Eisenberg. Identifiability results for several classes of linear compartment models. *arXiv e-prints*, 2014. arXiv:1410.8587.

John Milnor. *Singular points of complex hypersurfaces*. Annals of Mathematics Studies, No. 61. Princeton University Press, Princeton, N.J., 1968.

David Mumford. *The Red Book of Varieties and Schemes*. Lecture Notes in Mathematics. Springer-Verlag Berlin Heidelberg, 1999.

Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *Annals of Statistics*, 39(2), 2011.

Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697, 2012.

James Oxley. *Matroid theory*, volume 21 of *Oxford Graduate Texts in Mathematics*. Oxford University Press, Oxford, second edition, 2011.

Boris Pittel, Joel Spencer, and Nicholas Wormald. Sudden emergence of a giant k-core in a random graph. *Journal of Combinatorial Theory. Series B*, 67(1):111–151, 1996.

Ruslan Salakhutdinov and Nathan Srebro. Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS) 23*, pages 2056–2064. 2010.

Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the Association for Computing Machinery*, 27(4):701–717, October 1980.

Amit Singer and Mihai Cucuringu. Uniqueness of low-rank matrix completion by rigidity theory. *SIAM Journal on Matrix Analysis and Applications*, 31(4):1621–1641, 2010.

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Proceedings of the 18th Annual Conference on Learning Theory (COLT)*, pages 545–560. Springer, 2005.

Nathan Srebro, Jason D. M. Rennie, and Tommi S. Jaakkola. Maximum-margin matrix factorization. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems (NIPS) 17*, pages 1329–1336. MIT Press, Cambridge, MA, 2005.

Angelika Steger and Nicholas Wormald. Generating random regular graphs quickly. *Combinatorics Probability and Computing*, 8(4):377–396, 1999.

Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. On the extension of trace norm to tensors. In *NIPS Workshop on Tensors, Kernels, and Machine Learning*, 2010.

Journal of Machine Learning Research 16 (2015) 1437-1480 Submitted 12/13; Revised 11/14; Published 8/15

A Comprehensive Survey on Safe Reinforcement Learning

Javier García FJGPOLO@INF.UC3M.ES
Fernando Fernández FFERNAND@INF.UC3M.ES
Universidad Carlos III de Madrid,
Avenida de la Universidad 30,
28911 Leganes, Madrid, Spain

Editor: Joelle Pineau

Abstract

Safe Reinforcement Learning can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes. We categorize and analyze two approaches of Safe Reinforcement Learning. The first is based on the modification of the optimality criterion, the classic discounted finite/infinite horizon, with a safety factor. The second is based on the modification of the exploration process through the incorporation of external knowledge or the guidance of a risk metric. We use the proposed classification to survey the existing literature, as well as suggesting future directions for Safe Reinforcement Learning.

Keywords: reinforcement learning, risk sensitivity, safe exploration, teacher advice

1. Introduction

In reinforcement learning (RL) tasks, the agent perceives the state of the environment, and it acts in order to maximize the long-term return which is based on a real valued reward signal (Sutton and Barto, 1998). However, in some situations in which the safety of the agent is particularly important, for example in expensive robotic platforms, researchers are paying increasing attention not only to the long-term reward maximization, but also to damage avoidance (Mihatsch and Neuneier, 2002; Hans et al., 2008; Martín H. and Lope, 2009; Koppejan and Whiteson, 2011; García and Fernández, 2012).

The safety concept, or its opposite, risk, have taken many forms in the RL literature, and it does not necessarily refer to physical issues. In many works, risk is related to the stochasticity of the environment and with the fact that, in those environments, even an optimal policy (with respect the return) may perform poorly in some cases (Coraluppi and Marcus, 1999; Heger, 1994b). In these approaches, the risk concept is related to the *inherent uncertainty* of the environment (i.e., with its stochastic nature). Since maximizing the long-term reward does not necessarily avoid the rare occurrences of large negative outcomes, we need other criteria to evaluate risk. In this case, the long-term reward maximization is transformed to include some notion of risk related to the variance of the return (Howard and Matheson, 1972; Sato et al., 2002) or its worst-outcome (Heger, 1994b; Borkar, 2002; Gaskett, 2003). In other works, the optimization criterion is transformed to include the probability of visiting *error states* (Geibel and Wysotzki, 2005), or transforming the tem-

poral differences to more heavily weighted events that are unexpectedly bad (Mihatsch and Neuneier, 2002).

Other works do not change the optimization criterion, but the exploration process directly. During the learning process, the agent makes decisions about which action to choose, either to find out more about the environment or to take one step closer towards the goal. In RL, techniques for selecting actions during the learning phase are called exploration/exploitation strategies. Most exploration methods are based on heuristics, rely on statistics collected from sampling the environment, or have a random exploratory component (e.g., $\epsilon - greedy$). Their goal is to explore the state space efficiently. However, most of those exploration methods are blind to the risk of actions. To avoid risky situations, the exploration process is often modified by including prior knowledge of the task. This prior knowledge can be used to provide initial information to the RL algorithm biasing the subsequent exploratory process (Driessens and Džeroski, 2004; Martín H. and Lope, 2009; Koppejan and Whiteson, 2011), to provide a finite set of demonstrations on the task (Abbeel and Ng, 2005; Abbeel et al., 2010), or to provide guidance (Clouse, 1997; García and Fernández, 2012). Approaches based on prior knowledge were not all originally built to handle risky domains but, by the way they were designed, they have been demonstrated to be particularly suitable for this kind of problem. For example, initial knowledge was used to bootstrap an evolutionary approach by the winner of the helicopter control task of the 2009 RL competition (Martín H. and Lope, 2009). In this approach, several neural networks that clone error-free teacher policies are added to the initial population (facilitating the rapid convergence of the algorithm to a near-optimal policy and, indirectly, reducing agent damage or injury). Indeed, as the winner of the helicopter domain is the agent with the highest cumulative reward, the winner must also indirectly reduce helicopter crashes insofar as these incur large catastrophic negative rewards. Although the competition is based on the performance after the learning phase, these methods demonstrate that reducing the number of catastrophic situations, also during the learning phase, can be particularly interesting in real robots where the learning phase is performed in an on-line manner, and not through simulators. Instead, Abbeel and Ng (2005); Abbeel et al. (2010) use a finite set of demonstrations from a teacher to derive a safety policy for the helicopter control task, while minimizing the helicopter crashes. Finally, the guidance provided by a teacher during the exploratory process has also been demonstrated to be an effective method to avoid dangerous or catastrophic states (García and Fernández, 2012). In another line of research, the exploration process is conducted using some form of risk metric based on the *temporal differences* (Gehring and Precup, 2013) or in the weighted sum of an entropy measure and the expected return (Law, 2005).

In this manuscript, we present a comprehensive survey of work which considers the concepts of safety and/or risk within the RL community. We call this subfield within RL, Safe Reinforcement Learning. Safe RL can be defined as the process of learning policies that maximize the expectation of the return in problems in which it is important to ensure reasonable system performance and/or respect safety constraints during the learning and/or deployment processes. Safe RL algorithms suffer from the lack of an established taxonomy in which to organize existing approaches. In this survey, we have contributed such a structure, through a categorization of Safe RL algorithms. We segment Safe RL algorithms into two fundamental tendencies. The first consists of transforming the optimization criterion.

The second consists of modifying the exploration process in two ways: (i) through the incorporation of external knowledge, and, (ii) through the use of a risk metric. In this category, we focus on those RL approaches tested in risky domains that reduce or prevent undesirable situations through the modification of the exploration process. The objective of this is to become a starting point for researchers who are initiating their endeavors in Safe RL. It is important to note that the second category includes the first since modifying the optimization criterion will also modify the exploration process. However, in the first category we consider those approaches that transform the optimization criterion in some way to include a form of risk. On the other hand, the optimization criterion in the second category remains, while the exploration process is modified to consider some form of risk.

Resulting from these considerations, the remainder of the paper is organized as follows. Section 2 presents an overview and a categorization of Safe RL algorithms existing in the literature. The methods based on the transformation of the optimization criterion are examined in Section 3. The methods that modify the exploration process by the use of prior knowledge or a risk metric are considered in Section 4. In Section 5 we discuss the surveyed methods and identify open areas of research for future work. Finally, we conclude with Section 6.

2. Overview of Safe Reinforcement Learning

We consider learning in **M**arkov **D**ecision **P**rocesses (MDP) described formally by a tuple $< S, A, T, R >$, where S is the state space, A is the action space, $T : S \times A \to S$ is the transition function and $R : S \times A \to \mathbb{R}$ is the reward function (Putterman, 1994). In this survey, we consider two main trends for Safe RL (Table 1) to learn in MDPs. The first one is based on the modification of the optimality criterion to introduce the concept of risk (Section 3). The second is based on the modification of the exploration process to avoid the exploratory actions that can lead the learning system to undesirable or catastrophic situations (Section 4).

Optimization Criterion. As regards the first, the objective of traditional RL algorithms is to find an optimal control policy; that is, to find a function which specifies an action or a strategy for some state of the system to optimize a criterion. This optimization criterion may be to minimize time or any other cost metric, or to maximize rewards, etc. The optimization criterion in RL is described by a variety of terms within the published literature, including the expected return, expected sum of rewards, cumulative reward, cumulative discounted reward or return. Within this article, to avoid terminology misunderstandings, we use the term return.

Definition 1 ***Return***. *The term return is used to refer to the expected cumulative future discounted reward* $R = \sum_{t=0}^{\infty} \gamma^t r_t$*, where* r_t *represents a single real value used to evaluate the selection of an action in a particular state (i.e., the reward), and* $\gamma \in [0, 1]$ *is the discount factor that allows the influence of future rewards to be controlled.*

This optimization criterion is not always the most suitable one in dangerous or risky tasks (Heger, 1994b; Mihatsch and Neuneier, 2002; Geibel and Wysotzki, 2005). There are several alternatives to this optimization criterion in order to consider *risk*. In this survey,

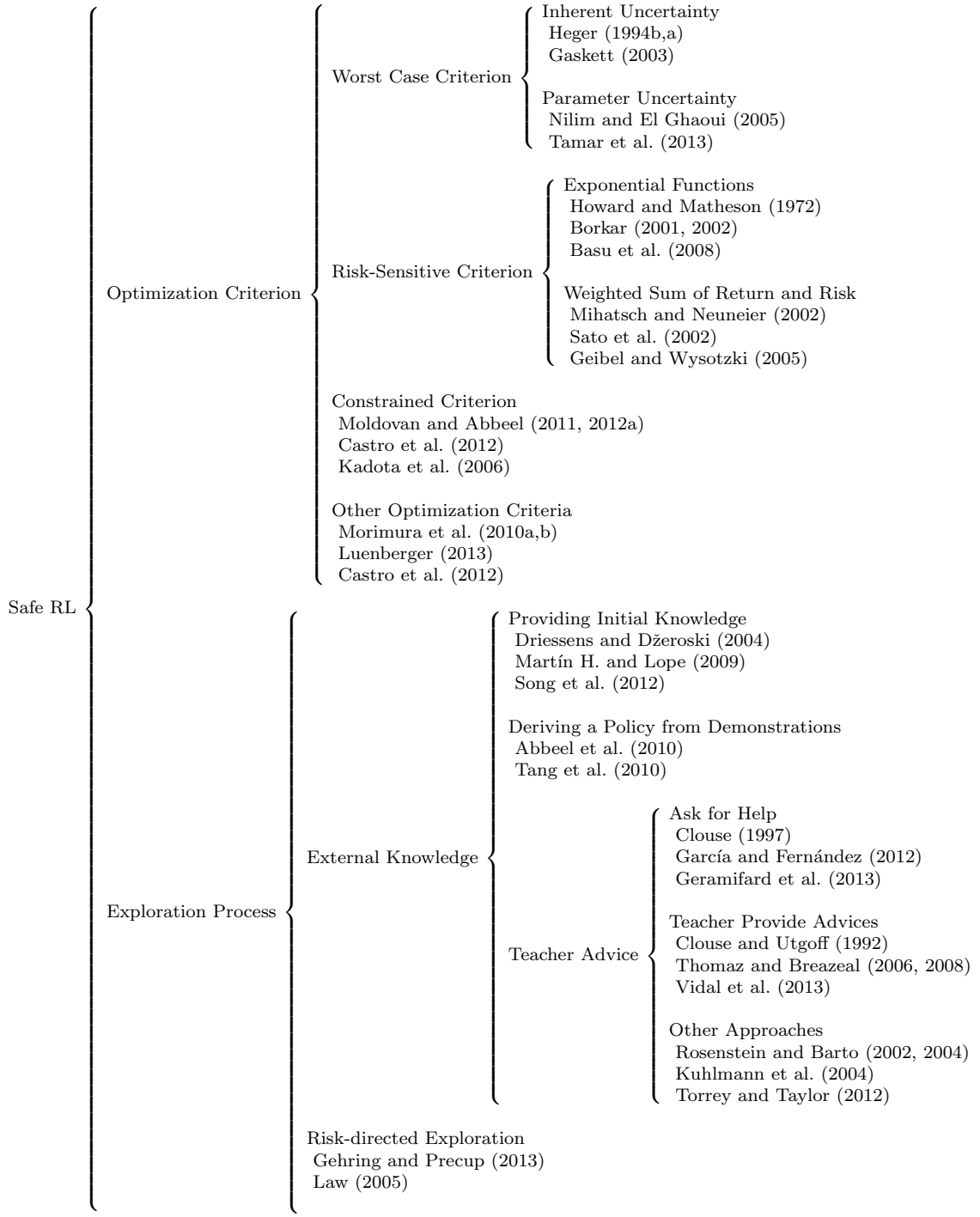


Table 1: Overview of the approaches for Safe Reinforcement Learning considered in this survey.

we categorize these optimization criteria in four groups: (i) the worst-case criterion, (ii) the risk-sensitive criterion, (iii) the constrained criterion, and (iv) other optimization criteria.

- *Worst Case Criterion.* The first criterion is based on the *Worst Case Criterion* where a policy is considered to be optimal if it has the maximum worst-case return (Section 3.1). This criterion is used to mitigate the effects of the variability induced by a given policy, since this variability can lead to risk or undesirable situations. This variability can be due to two types of uncertainties: the *inherent uncertainty* related to the stochastic nature of the system (Heger, 1994b,a; Gaskett, 2003), and the *parameter uncertainty* related to some of the parameters of the MDP are not known exactly (Nilim and El Ghaoui, 2005; Tamar et al., 2013).

- *Risk-Sensitive Criterion.* In other approaches, the optimization criterion is transformed so as to reflect a subjective measure balancing the return and the *risk*. These approaches are known as *risk-sensitive approaches* and are characterized by the presence of a parameter that allows the *sensitivity* to the *risk* to be controlled (Section 3.2). In these cases, the optimization criterion is transformed into an exponential utility function (Howard and Matheson, 1972), or a linear combination of return and *risk*, where *risk* can be defined as the *variance of the return* (Markowitz, 1952; Sato et al., 2002), or as the probability of entering into an *error state* (Geibel and Wysotzki, 2005).

- *Constrained Criterion.* The purpose of this objective is to maximize the return subject to one or more constraints resulting in the *constrained optimization criterion* (Section 3.3). In such a case, we want to maximize the return while keeping other types of expected measures higher (or lower) than some given bounds (Kadota et al., 2006; Moldovan and Abbeel, 2012a).

- *Other Optimization Criteria.* Finally, other approaches are based on the use of optimization criteria falling into the area of financial engineering, such as the *r-squared*, *value-at-risk* (*VaR*) (Mausser and Rosen, 1998; Kashima, 2007; Luenberger, 2013), or the density of the return (Morimura et al., 2010a,b) (Section 3.4).

Exploration Process. As regards the modification of the exploration process, there are also several approaches to overcoming the problems where the exploratory actions may have serious consequences. Most RL algorithms begin learning with no external knowledge of the task. In such cases, exploration strategies such as $\epsilon-greedy$ are used. The application of this strategy results in the random exploration of the state and action spaces to gather knowledge on the task. Only when enough information is discovered from the environment, does the algorithm's behavior improve. The randomized exploration strategies, however, waste a significant amount of time exploring irrelevant regions of the state and action spaces, or lead the agent to undesirable states which may result in damage or injury to the agent, the learning system or external entities. In this survey we consider two ways of modifying the exploration process to avoid risk situations: (i) through the incorporation of external knowledge and, (ii) through the use of a risk-directed exploration.

- *External Knowledge.* We distinguish three ways of incorporating prior knowledge into the exploration process (Section 4.1) by: (i) providing initial knowledge, (ii) deriving a policy from a finite set of demonstrations and, (iii) providing teach advice.

  - *Providing Initial Knowledge.* To mitigate the aforementioned exploration difficulties, examples gathered from a teacher or previous information on the task can be used to *provide initial knowledge* for the learning algorithm (Section 4.1.1). This knowledge can be used to bootstrap the learning algorithm (i.e., a type of initialization procedure). Following this initialization, the system can switch to a Boltzmann or fully greedy exploration based on the values predicted in the initial training phase (Driessens and Džeroski, 2004). In this way, the learning algorithm is exposed to the most relevant regions of the state and action spaces from the earliest steps of the learning process, thereby eliminating the time needed in random exploration for the discovery of these regions.

  - *Deriving a policy from a finite set of demonstrations.* In a similar way, a set of examples provided by a teacher can be used to *derive a policy from demonstrations* (Section 4.1.2). In this case, the examples provided by the random exploration policy are replaced by the examples provided by the teacher. In contrast to the previous category, this external knowledge is not used to bootstrap the learning algorithm, but is used to learn a model from which to derive a policy in an off-line and, hence, safe manner (Abbeel et al., 2010; Tang et al., 2010).

  - *Providing Teach Advice.* Other approaches based on *teacher advice* assist the exploration during the learning process (Section 4.1.3). They assume the availability of a teacher for the learning agent. The teacher may be a human or a simple controller, but in both cases it does not need to be an expert in the task. At every step, the agent observes the state, chooses an action, and receives the reward with the objective of maximizing the return or other optimization criterion. The teacher shares this goal, and provides actions or information to the learner agent. Both the agent and the teacher can initiate this interaction during the learning process. In the *ask for help* approaches (Section 4.1.3.1), the learner agent requests advice from the teacher when it considers it necessary (Clouse, 1997; García and Fernández, 2012). In other words, the teacher only provides advice to the learner agent when it is explicitly asked to. In other approaches (Section 4.1.3.2), it is the teacher who provides actions whenever it feels it is necessary (Thomaz and Breazeal, 2008; Vidal et al., 2013). In another group of approaches (Section 4.1.3.3), the main role in this interaction is not so clear (Rosenstein and Barto, 2004; Torrey and Taylor, 2012).

- *Risk-directed Exploration.* In these approaches a risk measure is used to determine the probability of selecting different actions during the exploration process (Section 4.2) while the classic optimization criterion remains (Gehring and Precup, 2013; Law, 2005).

3. Modifying the Optimization Criterion

This section describes the methods of the first category of the proposed taxonomy based on the transformation of the optimization criterion. The approaches using the return as the objective function are referred to as risk-neutral control (Putterman, 1994), because the variance and higher order moments in the probability distribution of the rewards are neglected.

Definition 2 ***Risk-Neutral Criterion****. In risk-neutral control, the objective is to compute (or learn) a control policy that maximizes the expectation of the return,*

$$\max_{\pi \in \Pi} E_\pi(R) = \max_{\pi \in \Pi} E_\pi(\sum_{t=0}^{\infty} \gamma^t r_t), \tag{1}$$

where $E_\pi(\cdot)$ stands for the expectation with respect to the policy π.

The decision maker may be also interested in other objective functions, different from the expectation of the return, to consider the notion of risk. In this case, risk is related to the fact that even an optimal policy may perform poorly in some cases due to the variability of the problem, and the fact that the process behavior is partially known. Because of the latter, the objective function is transformed, resulting in various risk-aware approaches. In this survey, we focus on three optimization criterion: the worst case criterion, the risk-sensitive criterion, and the constrained criterion. These approaches are discussed in detail in the following sections.

3.1 Worst-Case Criterion

In many applications, we would like to use an optimization criterion that incorporates a penalty for the variability induced by a given policy, since this variability can lead to risk or undesirable situations. This variability can be due to two types of uncertainties: a) the *inherent uncertainty* related to the stochastic nature of the system, and b) the *parameter uncertainty*, related to some of the parameters of the MDP are not known exactly. To mitigate this problem, the agent maximizes the return associated to the worst-case scenario, even though the case may be highly unlikely.

3.1.1 Worst-Case Criterion under Inherent Uncertainty

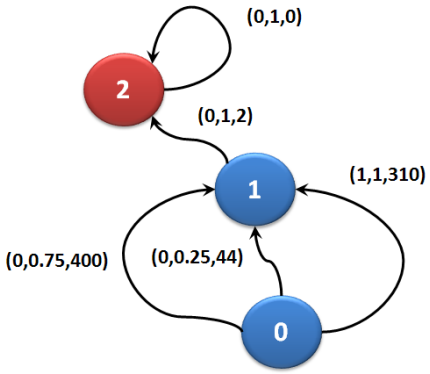
This approach is discussed at length in the literature (Heger, 1994b; Coraluppi, 1997; Coraluppi and Marcus, 1999, 2000).

Definition 3 ***Worst-Case or Minimax Criterion under inherent uncertainty****. In worst-case or minimax control the objective is to compute (or learn) a control policy that maximizes the expectation of the return with respect to the worst case scenario (i.e., the worst outcome) incurred in the learning process using,*

$$\max_{\pi \in \Pi} \min_{w \in \Omega^\pi} E_{\pi,w}(R) = \max_{\pi \in \Pi} \min_{w \in \Omega^\pi} E_{\pi,w}(\sum_{t=0}^{\infty} \gamma^t r_t), \tag{2}$$

where Ω^{π} *is a set of trajectories of the form* $(s_0, a_0, s_1, a_1, \dots)$ *that occurs under policy* π*, and where* $E_{\pi,w}(\cdot)$ *stands for the expectation with respect to the policy* π *and the trajectory* w*. That is, we are interested in the policy* $\pi \in \Pi$ *with the max-min outcome.*

We briefly review the difference between the risk-neutral and worst-case criterion using the example provided by Hedger, replicated in Figure 1 (see Heger, 1994b).



s	π(s)	μ(s)
0	0	1
1	0	0
2	0	0

Figure 1: Difference between risk-neutral and worst-case criterion. Example provided by Heger (1994b). Each transition is labeled as a triple. The first number a in the triple is an admissible action for the state s. The second number stands for the probability that the state transition will occur if action a is selected in the corresponding starting state s. The third number represents the immediate reward for the transition.

In Figure 1, there are three states and transitions between them. Each transition is represented by three components: the first is the action that performs the transition from one state to the other, the second is the probability of the transition, and the last is the reward obtained when performing this transition. Additionally, there are two policies labeled π and μ. In Figure 1, $E_{\pi}(R) = 311 + 2\gamma$ and $E_{\mu}(R) = 310 + 2\gamma$. Therefore, by applying the risk-neutral criterion, the policy π is optimal. However, max inf $(E_{\pi}(R)) = 44 + 2\gamma$ and max inf $(E_{\mu}(R)) = 310 + 2\gamma$. Therefore μ is optimal when applying the worst-case criterion. In worst case control strategies, the optimality criterion is exclusively focused on risk-avoidance or risk-averse policies. A policy is considered to be optimal if its worst-case return is superior.

Heger (1994b) introduces the $\hat{Q} - Learning$ which can be regarded as the counterpart to Q-Learning (Watkins, 1989) related to the minimax criterion,

$$\hat{Q}(s_t, a_t) = \min(\hat{Q}(s_t, a_t), r_{t+1} + \gamma \max_{a_{t+1} \in A} \hat{Q}(s_{t+1}, a_{t+1})) \tag{3}$$

The $\hat{Q}$ value is essentially a lower bound on value. $\hat{Q} - learning$ and the minimax criterion are useful when avoiding risk is imperative. Jiang et al. (1998) combine the simple function approximation state aggregation with the minimax criterion and present the convergence theory for $\hat{Q} - learning$. However, Gaskett (2003) tested $\hat{Q} - learning$ in a

stochastic cliff world environment, under the condition that actions are picked greedily, and found that $\hat{Q}-learning$ demonstrated extreme pessimism which can be more injurious than beneficial (Gaskett, 2003). Chakravorty and Hyland (2003) apply the minimax criterion to the actor-critic architecture and presents error bounds when using state aggregation as a function approximation. In general, the minimax criterion is too restrictive as it takes into account severe but extremely rare events which may never occur (Mihatsch and Neuneier, 2002). The $\alpha - value$ of the return $\hat{m_\alpha}$ introduced by Heger (1994a) can be seen as an extension of the worst case control of MDPs. This concept establishes that the returns $R < \hat{m_\alpha}$ of a policy that occur with a probability of less than α are ignored. The algorithm is less pessimistic than the pure worst case control, given that extremely rare scenarios have no effect on the policy.

Gaskett (2003) proposes a new extension to Q-learning, β-pessimistic $Q - learning$, which compromises between the extreme optimism of standard $Q-learning$ and the extreme pessimism of minimax approaches,

$$Q_\beta(s_t, a_t) = Q_\beta(s_t, a_t) + \alpha(r_{t+1} + \gamma((1-\beta) \max_{a_{t+1} \in A} Q_\beta(s_{t+1}, a_{t+1}) + \beta \min_{a_{t+1} \in A} Q_\beta(s_{t+1}, a_{t+1}))) \tag{4}$$

In the β-pessimistic $Q - learning$ algorithm the value of $\beta \in [0, 1]$ renders the equation into the standard $Q - learning$ or the minimax algorithm respectively (Gaskett, 2003). Experimental results show that when $\beta = 0.5$, the algorithm reaches the same level of pessimism as $\hat{Q} - learning$, although the agent manages to reach the goal state in some cases, unlike in $\hat{Q} - learning$.

3.1.2 Worst-Case Criterion under Parameter Uncertainty

Some RL approaches are focused to learning the model first which is assumed to be correct, and then applying a dynamic programming to it to learn and optimal policy. However, in practice, the model learned is typically estimated from noisy data or insufficient training examples, or even worse, they may change during the execution of a policy. These modeling errors may have fatal consequences in real, physical systems, where there are often states that are really catastrophic and must be avoided even during learning. This problem is faced by the robust control community (Zhou et al., 1996), whose one goal is to build policies with satisfactory online performance and robustness to model errors. Specifically, a robust MDP deals with uncertainties in parameters; that is, some of the parameters, namely, transition probabilities, of the MDP are not known exactly (Bagnell et al., 2001; Iyengar, 2004; Nilim and El Ghaoui, 2005).

Definition 4 ***Worst-Case or Minimax Criterion under parameter uncertainty.*** *Typically this criterion is described in terms of a set (uncertainty set), P, of possible transition matrices, and the objective is to maximize the expectation of the return for the worst case policy over all possible models $p \in P$,*

$$\max_{\pi \in \Pi} \min_{p \in P} E_{\pi,p}(R) = \max_{\pi \in \Pi} \min_{p \in P} E_{\pi,p}(\sum_{t=0}^{\infty} \gamma^t r_t), \tag{5}$$

where $E_{\pi,p}(\cdot)$ stands for the expectation with respect to the policy π and the transition model p.

The problem of parameter uncertainty has been recognized in the reinforcement learning community as well, and algorithms have been suggested to deal with it (Tamar et al., 2013). Typical model-based reinforcement algorithms ignore this type of uncertainty. However, minimizing the risk is one of several imports in model-based reinforcement learning solutions, particularly ones in which failure has important consequences (Bagnell and Schneider, 2008), as a learned model invariably has certain inaccuracies, due to insufficient or noise training data (Bagnell, 2004).

3.2 Risk-Sensitive Criterion

In risk-sensitive RL, the agent has to strike a balance between getting large reinforcements and avoiding catastrophic situations even if they occur with very small probability. For example, a profit-maximizing firm may want to be conservative in making business decisions to avoid bankruptcy even if its conservation will probably lower the expected profits.

Definition 5 ***Risk-Sensitive Criterion****. In risk-sensitive RL, the objective function includes a scalar parameter β that allows the desired level of risk to be controlled. The parameter β is known as the risk sensitivity parameter, and is generally either positive or negative: $\beta > 0$ implies risk aversion, $\beta < 0$ implies a risk-seeking preference, and (through a limiting argument) $\beta = 0$ implies risk neutrality.*

Depending on the form of the objective function, it is possible to consider various risk-sensitive RL algorithms.

3.2.1 Risk-Sensitive Based on Exponential Functions

In risk-sensitive control based on the use of *exponential utility functions*, the return R is transformed to reflect a subjective measure of utility (Howard and Matheson, 1972; Chung and Sobel, 1987). Instead of maximizing the expected value of R, the objective here is to maximize

$$\max_{\pi \in \Pi} \ \beta^{-1} log E_{\pi}(\exp^{\beta R}) = \max_{\pi \in \Pi} \ \beta^{-1} log E_{\pi}(\exp^{\beta \sum_{t=0}^{\infty} \gamma^t r_t}), \tag{6}$$

where β is a parameter and R is the return. A straightforward Taylor expansion of the *exp* and *log* terms of Equation 6 yields in Equation (by using the big $\mathcal{O}$ notation)

$$\max_{\pi \in \Pi} \ \beta^{-1} log E_{\pi}(\exp^{\beta R}) = \max_{\pi \in \Pi} \ E_{\pi}(R) + \frac{\beta}{2} Var(R) + \mathcal{O}(\beta^2), \tag{7}$$

where $Var(R)$ denotes the variance of the return. Variability is penalized for $\beta < 0$ and enforced for $\beta > 0$. Therefore, the objective is risk-averse for $\beta < 0$, risk-seeking for $\beta > 0$ and risk-neutral for $\beta = 0$.

Most of the work of this trend is within the MDP framework where the transition probabilities and rewards are explicitly available. As an example, Patek (2001) analyzed a class of terminating MDPs with a risk-averse, expected-exponential criterion, with compact constraint sets. By restricting attention to risk-averse problems ($\beta > 0$) with all transition

costs strictly positive, and by assuming the existence of a stationary policy, the authors established the existence of stationary optimal policies. More recently, Osogami (2012) and Moldovan and Abbeel (2012b) demonstrate that a risk-sensitive MDP for maximizing the expected exponential utility is equivalent to a robust MDP for maximizing the worst-case criterion. Although the exponential utility approach constitutes the most popular and best analyzed risk-sensitive control framework in the literature, there remain serious drawbacks which prevent the formulation of corresponding RL algorithms (Mihatsch and Neuneier, 2002): time-dependent optimal policies, and no model-free RL algorithms for both deterministic and stochastic reward structures. As a result, the use of this criterion does not lend itself easily to model-free reinforcement learning methods such as TD(0) or Q-learning (Heger, 1994b). Therefore, much less work has been done within the RL framework using this exponential utility function as an optimization criterion, with a notable exception of Borkar (2001, 2002) who relaxes the assumption of a system model by deriving a variant of the Q-learning algorithm for finite MDPs with an exponential utility. Basu et al. (2008) present an approach, extending the works by Borkar (2001, 2002), for Markov decision processes with an infinite horizon risk-sensitive cost based on an exponential function. Its convergence is proved using the **o**rdinary **d**ifferential **e**quation (o.d.e) method for stochastic approximation, and it is also extended to continuous state space processes.

In a different line of work, Chang-Ming et al. (2007) demonstrated that the max operator in Equation 6 can be replaced with a generalized averaged operator in order to improve the robustness of RL algorithms. From a more practical point of view, Liu et al. (2003) use an exponential function in the context of auction agents. Since companies are often risk-averse, the authors derive a closed form of the optimal bidding function for auction agents that maximize the expected utility of the profit for concave exponential utility functions.

However, all the approaches considered in this trend share the same idea: associate the risk with the variance of the return. Higher variance implies more instability and, hence, more risk. Therefore, it should be noted that the aforementioned approaches are not suited for problems where a policy with a small variance can produce a large risk (Geibel and Wysotzki, 2005).

3.2.2 Risk-Sensitive RL Based on the Weighted Sum of Return and Risk

In this trend, the objective function is expressed as the weighted sum of return and risk given by

$$\max_{\pi \in \Pi} \left(E_\pi(R) - \beta \, \omega \right) \tag{8}$$

In Equation 8, $E_\pi(R)$ refers to the expectation of the return with respect the policy π, β is the risk-sensitive parameter, and ω refers to the consideration of the risk concept which can take various forms. A general objective function is the well-known Markowitz criterion (Markowitz, 1952) where the ω in Equation 8 is replaced by the variance of the return, $Var(R)$. This criterion is also known in the literature as *variance-penalized criterion* (Gosavi, 2009), *expected value-variance criterion* (Taha, 1992; Heger, 1994b) and *expected-value-minus-variance-criterion* (Geibel and Wysotzki, 2005). Within the RL framework, Sato et al. (2002) propose an approach that directly optimizes an objective function defined as a linear combination of the mean and the variance of the return. However, this is based on the assumption of the mean-variance model where the distribution

of the return follows a Gaussian distribution, which does not hold in most situations. There are several limitations when using the return variance as a measure of risk. First, the fat tails of the distribution are not accounted for. Consequently, risk can be underestimated due to the ignorance of low probability, but highly severe events. Second, variance penalizes both positive and negative risk equally and does not distinguish between the two. Third, this criterion is incorrectly applied to many cases in which risk cannot be described by the variance of the return (Szegö, 2005). Additionally, mean minus variance optimization within the MDP framework has been shown to be NP-hard in general, and optimizing this criterion can directly lead to counterintuitive policies (Mannor and Tsitsiklis, 2011).

Mihatsch and Neuneier (2002) replace the ω in Equation 8 with the temporal difference errors that occur during learning. Their learning algorithm has a parameter $\beta \in (-1.0, 1.0)$ that allows for switching between *risk-averse* behavior ($\beta = 1$), *risk-neutral* behavior ($\beta = 0$) and risk-seeking behavior ($\beta = -1$). Loosely speaking, the authors overweigh transitions to successor states where the immediate return happen to be smaller than in the average, and they underweigh transitions to states that promise a higher return than the average. In the study, the authors demonstrate that the learning algorithm has the same limiting behavior as exponential utility functions. This method is extended by Campos to deal with large dimensional state/action spaces (Campos and Langlois, 2003).

Geibel and Wysotzki (2005) replace the ω in Equation 8 with the probability, $\rho^\pi(s)$, in which a state sequence $(s_i)_{i\geq 0}$ with $s_0 = s$, generated by the execution of policy π, terminates in an error state,

$$\rho^\pi(s) = E(\sum_{i=0}^{\infty} \gamma^i \bar{r}) \tag{9}$$

In Equation 9, $\bar{r}$ is a cost function in which $\bar{r} = 1$ if an error state occurs and $\bar{r} = 0$ if not. In this case and, as demonstrated by García and Fernández (2012), $\rho^\pi(s)$ is learned by TD methods which require error states (i.e., helicopter crashes or company bankruptcies) to be visited repeatedly in order to approximate the risk function and, subsequently, to avoid dangerous situations.

Common to the works of Mihastch and Geibel is the fact that risk-sensitive behavior is induced by transforming the action values, $Q(s, a)$, or the state values, $V(s)$. There are several reasons why this may not be desirable: (i) if these values are updated based on a conservative criterion, the policy may be overly pessimistic; (ii) the worst thing that can happen to an agent in an environment may have high utility in the long term, but fatal consequences in the short term; and (iii) the distortion of these values means that the true long term utility of the actions are lost.

3.3 Constrained Criterion

The constrained criterion is applied in the literature to *constrained* Markov processes in which we want to maximize the expectation of the return while keeping other types of expected utilities lower than some given bounds (Altman, 1992). This approach might be considered within the second category of the taxonomy described here, since the optimization criterion remains. However, the addition of constraints to this optimization criterion is sufficient to consider a transformation and so we consider that it must be included within

this category. The *constrained* MDP is an extension of the MDP framework described as the tuple $< S, A, R, T, C >$, where S, A, R, T are defined as in standard MDP, and C is a set of constraints applied to the policy.

Definition 6 ***Constrained Criterion***. *In the constrained criterion, the expectation of the return is maximized subject to one or more constraints, $c_i \in C$. The general form of this criterion is shown in the following*

$$\max_{\pi \in \Pi} E_\pi(R) \text{ subject to } c_i \in C, c_i = \{h_i \leq \alpha_i\}, \tag{10}$$

where c_i represents the ith constraint in C that the policy π must fulfill, with $c_i = \{h_i \leq \alpha_i\}$ where h_i is a function related with the return and α_i is the threshold restricting the values of this function. Depending of the problem the symbol $\leq$ in the constraints $c_i \in C$ may be replaced by $\geq$.

We can see the proposed constraints in Equation 10 as restrictions on the space of allowable policies. Figure 2 shows the entire policy space, Π, and the set of allowable policies, $\Gamma \subset \Pi$, where each policy $\pi \in \Gamma$ satisfies the constraints $c_i \in C$.

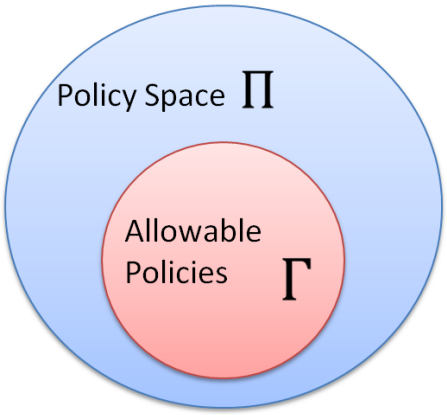


Figure 2: Policy space, Π, and the set of allowable policies, $\Gamma \subset \Pi$, where each policy $\pi \in \Gamma$ satisfies the constraints $c_i \in C$.

Therefore, Equation 10 can be transformed into

$$\max_{\pi \in \Gamma} E_\pi(R) \tag{11}$$

From a safety point of view, this optimization criterion is particularly suitable for risky domains. In these domains, the objective may be seen as finding the best policy π in the space of considered safe policies, Γ. This space, Γ, may be restricted using various types of constraints: constraints to ensure that the expectation of the return exceeds some specific minimum threshold, to ensure that the variance of the return does not exceed specific maximum threshold, to enforce *ergodicity*, to ensure specific restrictions of the problem.

A typical constraint is referred to ensure that the expectation of the return exceeds some specific minimum threshold, $E(R) \geq \alpha$ (Geibel, 2006). In this case, the space of considered safe policies, Γ, is made up of the policies for which the expectation of return exceeds the specific threshold, α. This is suitable in situations in which we already know a reasonably good policy, and we want to improve it through exploration, but the expectation of the return may not fall below a given safety margin. In these kinds of problem, one can derive the LP problem by using a Lagrangian approach which allows us to transform the constrained problem into a equivalent non-constrained one. As an example, Kadota et al. (2006) transform the constrained criterion into a Lagrangian expression. In this way, the method reduces the constrained problem with n variables to one with $n + k$ unrestricted variables, where k is equal to the number of restrictions. Thus, the resulting expression can be solved more easily. The previous constraint is a hard constraint that cannot be violated, but other approaches allow a certain admissible chance of constraint violation. This chance-constraint metric, $P(E(R) \geq \alpha) \geq (1 - \epsilon)$, is interpreted as guaranteeing that the expectation of the return (considered a random variable) will be at least as good as α with a probability greater than or equal to $(1 - \epsilon)$ (Delage and Mannor, 2010; Ponda et al., 2013).

Instead, other approaches use a different constrained criterion in which the variance of the return must not exceed a given threshold, $Var(R) \leq \alpha$ (Castro et al., 2012). In this case, the space of safe policies, Γ, is made up of policies for which the variance does not exceed a threshold, α. This constrained problem is also transformed into an equivalent unconstrained problem by using *penalty methods* (Smith et al., 1997). Then, the problem is solved using standard unconstrained optimization techniques.

Other approaches rely on *ergodic* MDPs (Hutter, 2002) which guarantee that any state is reachable from any other state by following a suitable policy. Unfortunately, many risky domains are not *ergodic*. For example, our robot helicopter learning to fly cannot recover on its own after crashing. The space of safe policies, Γ, is restricted to those policies that preserve ergodicity with some user-specified probability, α, called the safety level. That is, only visiting states s so that one can always get back from s to the initial state (Moldovan and Abbeel, 2011, 2012a). In this case, the authors use plain linear programming formulation after removing the non-linear dependences to solve the constrained MDP efficiently. It is important to note that this constrained criterion is closely related to the *recoverable* and *value-state* concepts described by Ryabko and Hutter. An environment is *recoverable* if it is able to forgive initial *wrong* actions, i.e., after any arbitrary finite sequence of actions, the optimal policy is still achievable. Additionally, an environment is *value-stable* if from any sequence of k actions, it is possible to return to the optimal level of reward in $o(k)$ steps; that is, it is not just possible to recover after any sequence of (wrong) actions, but it is possible to recover fast.

Finally, Abe et al. (2010) proposed a constrained RL algorithm and reported their experience in an actual deployment of a tax collection optimization system based on their approach, at New York State Department of Taxation and Finance. In this case, the set of constraints, C, is made up of legal, business and resource constraints, which are specific to the problem under consideration. In contrast to the previous general formulations in which the constraints are defined as a function of the entire state trajectory, the authors formulate the constraints as being fixed and known at each learning iteration.

However these approaches have three main drawbacks. First, the correct selection of the threshold α. Higher values mean that they are too permissive, or conversely, too restrictive. Second, they do not prevent the fatal consequences in the short term. Finally, these methods associate the risk to policies in which the return or its variance is greater than a specified threshold, which is not suitable for most risk domains.

3.4 Other Optimization Criteria

In the area of financial engineering, various risk metrics such as *r-squared*, *beta*, *Sharpe ratio* or *value-at-risk* (*VaR*) have been studied for decision making with a low risk of huge costs (Mausser and Rosen, 1998; Kashima, 2007; Luenberger, 2013). Castro et al. (2012) also use the *Sharpe ratio* criterion, $\max_{\pi\in\Pi} E_\pi(R)\sqrt{Var(R)}$. The performance of this criterion is compared with the constrained criterion, $Var(R) \leq \alpha$, and the classic optimization criterion (Equation 1) in a portfolio management problem where the available investment options include both liquid and non-liquid assets. The non-liquid asset has some risk of not being paid (i.e., a default) with a given probability. The policy for the classic criterion is risky, and yields a higher gain than the policy for the constrained criterion. Interestingly, $\max_{\pi\in\Pi} E_\pi(R)\sqrt{Var(R)}$ resulted in a very risk-averse policy, that almost never invested in the non-liquid asset. This interesting phenomenon discourages the use of this optimization criterion. Even the authors suggest that it might be more prudent to consider other risk measures instead of the $\max_{\pi\in\Pi} E_\pi(R)\sqrt{Var(R)}$. Morimura et al. (2010a,b) focus their risk-sensitive approach on estimating the density of the returns, which allows them to handle various risk-sensitive criteria. However, the resulting distributional-SARSA-with-CVaR (or d-SARSA with CVaR) algorithm, has proved effectiveness only in a very simple and discrete MDP with 14 states.

4. Modifying the Exploration Process

This section describes the methods of the second category of the proposed taxonomy. In this category, in contrast with the previous one, the optimization criterion remains, but the exploration process is modified to consider some form of risk. Classic exploration/exploitation strategies in RL assume that the agent must explore and learn everything from scratch. In this framework, the agent is blind to the risk of actions during learning, potentially ending up in catastrophic states (Geibel and Wysotzki, 2005; García and Fernández, 2012). The helicopter hovering control task is one such case involving high risk, since some policies can crash the helicopter, incurring catastrophic negative reward. Exploration/exploitation strategies such as $\epsilon - greedy$ may even result in constant helicopter crashes (especially where there is a high probability of random action selection). In addition, random exploration policies waste a significant amount of time exploring irrelevant regions of the state and action spaces in which the optimal policy will never be encountered. This problem is more pronounced in environments with extremely large and continuous state and action spaces. Finally, it is impossible to completely avoid undesirable situations in high-risk environments without a certain amount of external knowledge (that is, not coming from interaction between the agent and the system): the use of random exploration would require an undesirable state to be visited before it can be labeled as undesirable. However, such visits to undesirable states usually lead to unrecoverable situations or *traps* (Ryabko and

Hutter) (i.e., the agent is not able to achieve the optimal policy after a sequence of *wrong* actions) and may result in damage or injury to the agent, the learning system or external entities. Consequently, visits to these states should be avoided from the earliest steps of the learning process. In this paper, we focus on two ways of modifying the exploration process in order to avoid visits to undesirable states: through the incorporation of external knowledge or through a directed exploration based on a risk measure. Both approaches are discussed in detail in the following sections.

4.1 Incorporating External Knowledge

Mitigating the difficulties described above, external knowledge (e.g., finite sets of teacher-provided examples or demonstrations) can be used in three general ways, either (i) to provide initial knowledge (i.e., a type of initialization procedure) or (ii) to derive a policy from a finite set of examples or (iii) to guide the exploration process through teacher advice. In the first case, the knowledge is used to bootstrap the value function approximation and lead the agent through the more relevant regions of the space. In the second way, finite sets of teacher-provided examples or demonstrations can be used to derive a policy. In these ways, the learning algorithm is exposed to the most relevant regions of the state and action spaces from the earliest steps of the learning process, thereby eliminating the time needed in random exploration for the discovery of these regions.

However, while furnishing the agent with initial knowledge helps to mitigate the problems associated with random exploration, this initialization alone is not sufficient to prevent the undesirable situations that arise in the subsequent explorations undertaken to improve learner ability. An additional mechanism is necessary to guide this subsequent exploration process in such a way that the agent may be kept far away from catastrophic states. So, in the third case, a teacher is used to provide information when it is considered necessary. These three ways of incorporating external knowledge are widely discussed in the following sections. Some of the approaches described here were not created originally as specific Safe RL methods, but they have some properties that make them particularly suitable for these kinds of problem.

4.1.1 Providing Initial Knowledge

The most elementary method for biasing learning is to choose some initialization based on prior knowledge of the problem. In Driessens and Džeroski (2004), a bootstrapping procedure is used for relational RL in which a finite set of demonstrations are recorded from a human teacher and later presented to a regression algorithm (Driessens and Džeroski, 2004). This allows the regression algorithm to build a partial Q-function which can later be used to guide further exploration of the state space using a Boltzmann exploration strategy. Smart and Kaelbling also use examples, training runs to bootstrap the Q-learning approach for their HEDGER algorithm (Smart and Kaelbling, 2000). The initial knowledge bootstrapped into the Q-learning approach allows the agent to learn more effectively and helps to reduce the time spent with random actions. Teacher behaviors are also used as a form of *population seeding* in neuroevolution approaches (Siebel and Sommer, 2007). Evolutionary methods are used to optimize the weights of neural networks, but starting from a prototype network whose weights correspond to a teacher (or baseline policy). Using this

technique, RL Competition helicopter hovering task winners Martín H. and Lope (2009) developed an evolutionary RL algorithm in which several teachers are provided in the initial population. The algorithm restricts crossover and mutation operators, allowing only slight changes to the policies given by the teachers. Consequently, it facilitates a rapid convergence of the algorithm to a near-optimal policy, as is the indirect minimization of damage to the agent. In Koppejan and Whiteson (2009, 2011), neural networks are also evolved, beginning with one whose weights correspond to the behavior of the teacher. While this approach has been proven advantageous in numerous applications of evolutionary methods (Hernández-Díaz et al., 2008; Koppejan and Whiteson, 2009), Koppejan's algorithm nevertheless seems somewhat ad-hoc and designed for a specialized set of environments.

Maire (2005) propose an approach for deriving high quality initial value functions from existing demonstrations by a teacher. The resulting value function constitutes a starting point for any value function-based RL method. As the initial value function is substantially more informative than a random value function initialization frequently used with RL methods, the remaining on-line learning process is conducted safer and faster. Song et al. (2012) also improve the performance of the Q-learning algorithm initializing the Q-values appropriately. These approaches are used in the Grid-World domain and are able to reduce drastically the times the agent moves into an obstacle.

Some *Transfer Learning* (TL) algorithms are also used to initialize a learner in a target task (Taylor and Stone, 2009). The core idea of transfer is that experience gained in learning to perform one task can help to improve learning performance in a related, but different, task. Taylor and Stone (2007) train an agent in a source task recording the agent's trajectories (i.e., state-action pairs). Then, the agent uses this experience to train in the target task off-line before the on-line training begins. These authors also learn an action-value function in a source task, translate the function into a target task via a hand-coded inter-task mapping, and then use the transferred function to initialize the target task agent (Taylor et al., 2007). Despite TL approaches having been shown effective in speeding up the learning processes, they present two main difficulties in their applicability to risky domains: (i) the knowledge to be reused in the target task requires it to be previously learned in a source task(s) (which is not always possible to do in a safe manner), and (ii) it is not always trivial to transfer this knowledge from the source task(s) to the target task since they could be of a different nature.

There is extensive literature on initialization in RL algorithms (Burkov and Chaib-draa, 2007), but their intensive analysis falls outside the scope of this paper since not all of them are focused to preserving the agent's safety or avoiding risky or undesirable situations. But the bias introduced in the learning process and the rapid convergence produced by most of these algorithms, can ensure their applicability to risky domains. However, this approach is problematic for two main reasons. First, if the initialization does not provide information for all important states the agent may end up with a suboptimal policy. Second, the exploration process following the initial training phase can result in visiting new states for which the agent has no information on how to act. As a result, the probability of incurring damage or injury is greatly increased. In addition, the relevance of these methods is highly dependent on the internal representations used by the agent. If the agent simply maintains a table, initialization is easy, but if the agent uses a more complex representation, it maybe very difficult or impossible to initialize the learning algorithm.

4.1.2 Deriving a Policy from a Finite Set of Demonstrations

All approaches falling under this category are framed according to the field of Learning from Demonstration (LfD) (Argall et al., 2009). Highlighting the study by Abbeel and Ng (2005); Abbeel et al. (2010) based on apprenticeship learning, the approach is made up of three distinct steps. In the first, a teacher demonstrates the task to be learned and the state-action trajectories of the teacher's demonstration are recorded. In the second step, all state-action trajectories seen so far are used to learn a model from the system's dynamics. For this model, a (near-)optimal policy is to be found using any reinforcement learning (RL) algorithm. Finally, the policy obtained should be tested by running it on the real system. In Tang et al. (2010), an algorithm based on apprenticeship learning is also presented for automatically-generating trajectories for difficult control tasks. The proposal is based on the learning of parameterized versions of desired maneuvers from multiple expert demonstrations. In these approaches, the learner is able to exceed the performance of the teacher. Despite each approach's potential strengths and general interest, all are inherently linked to the information provided in the demonstration data set. As a result, learner performance is heavily limited by the quality of the teacher's demonstrations. While one way to circumvent the difficulty and improve performance is by exploring beyond what is provided in the teacher demonstrations, this again raises the question of how the agent should act when it encounters a state for which no demonstration exists. One possible answer to this question is based on the use of teacher advice techniques, as defined below.

4.1.3 Using Teacher Advice

Exploring the environment while avoiding fatal states is critical for learning in domains where a bad decision can lead the agent to a dangerous situation. In such domains, different ways of teacher advice in reinforcement learning has been proposed as a form of safe exploration (Clouse, 1997; Hans et al., 2008; Geramifard et al., 2013; García and Fernández, 2012). The guidance provided by a teacher supports the safe exploration in two ways. First, the teacher can guide the learner in promising parts of the state space where suggested by the teacher's policy. This guidance reduces the sample complexity of learning techniques which is important when dealing with dangerous or high-risk domains. Secondly, the teacher is able to provide advice (e.g., safe actions) to the learner when either the learner or the teacher considers it is necessary so as to prevent catastrophic situations.

The idea of a program learning from external advice was first proposed in 1959 by John McCarthy (Mccarthy, 1959). Teacher advice is based on the use of two fundamental sources of training information: future payoff achieved by taking actions according to a given policy from a given state (derived from classic exploration in RL), and the advice from a teacher as regards which action to take next (Utgoff and Clouse, 1991). The objective of the approaches considered here is to combine these two sources of training information. In these approaches, a learner agent improves its policy based on the information (i.e., the advice) provided by a teacher.

Definition 7 ***Teacher Advising (VN and Ravindran, 2011).*** *Any external entity which is able to provide an input to the control algorithm that could be used by the agent to take decisions and modify the progress of its exploration.*

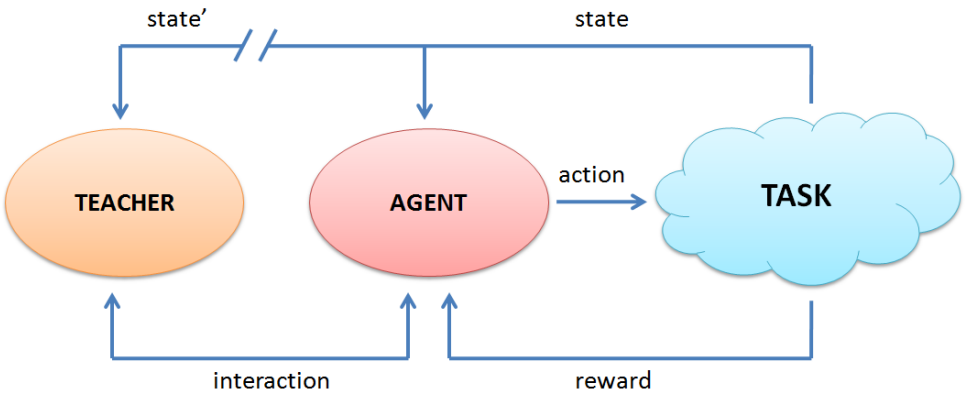


Figure 3: General Teacher-Learner Agent Interaction Scheme.

Figure 3 details such an interaction scheme between the teacher and the learner agent. In every time step, the learner agent perceives the state, chooses the action to perform, and receives a reward as in the classic RL interaction. In this framework, the teacher generally observes the same state as the learner and either the learner or the teacher determines when it is appropriate for the teacher to give an advice. However, the state observed by the agent, *state*, and the teacher, *state'*, could be different if they have different sensing mechanisms (e.g., a robot learner's camera will not detect state changes in the same way as a human teacher's eyes) (Argall et al., 2009). Additionally, the nature of the advice can have various forms: a single action that the learner carries out at that time (Clouse and Utgoff, 1992; Clouse, 1997; García and Fernández, 2012); a complete sequence of actions that the learner agent replays internally (Lin, 1992; Driessens and Džeroski, 2004); reward used to judge the agent's behavior interactively (Thomaz and Breazeal, 2006; Knox and Stone, 2009, 2010; Knox et al., 2011); a set of actions from which the agent has to select one randomly or greedily (Thomaz and Breazeal, 2006; Cetina, 2008).

The general framework of teacher's advice includes five main steps (Philip Klahr Hayes-Roth and Mostow., 1981): (i) requesting or receiving the advise; (ii) converting advice into a usable form; (iii) integrating the reformulated advice into the agent's knowledge base; and (iv) judging the value of advice. In this survey, we focus on step one to classify the different approaches of this trend. Thus, there are two main categories of algorithms: the learner agent asks for advice from the teacher when it needs to, the teacher provides advice to the learner agent when it is necessary.

4.1.3.1 The Learner Agent Asks for Advice

In this approach, the learner agent poses a *confidence parameter* and when this confidence in a state is low, the learner agent asks for advice from the teacher. Typically, this advice corresponds to the action that the teacher would carry out if it were in the place of the learner agent in the current state. In case of advice, the learner agent assimilates the teacher's action by first performing the action (as the learner itself selected it), and later receiving the corresponding reward. This reward is used to update the policy using any RL

algorithm. These teacher-advising algorithms are called *Ask for help* algorithms (Clouse, 1997). In the original *Ask for help* approach, Clouse (1997) uses the *confidence parameter* in two different strategies: uniform asking strategy and uncertainty asking strategy. In the first, the learner's request is spread uniformly throughout the learning process. In this case, this parameter establishes the percentage of time steps in which the learner requests help. The second is based on the learner agent's uncertainty about its current action selection. In this case, Clouse establishes that the agent is unsure about the action to choose when all the actions in the current learning step have similar *Q-values*; i.e., if the minimum and maximum *Q-values* are very similar (which is specified by the *confidence parameter*), the learner agent asks for advice. However, this interval-estimation measure between the highest and lowest Q-values produces counterintuitive results in some domains, such the maze domain. In this domain, the true Q-values of actions for each state are very similar since the maze states are highly connected. Interval estimation is therefore not a stable measure of confidence for maze-like domains.

Hans et al. (2008) and García and Fernández (2011, 2012); García et al. (2013) use this *confidence parameter* to detect *risky* situations. In this case, the concept of risk is based on the definition of *fatal transitions* or *unknown states*. Hans et al. (2008) consider a transition to be fatal if the corresponding reward is less than a given threshold τ, and an action a to be unsafe in a given state s if it leads to a fatal transition. In this work, the authors also build the teacher's policy with an altered Bellman optimality equation that does not maximize the return, but the minimal reward to come. The learner agent tries to explore all actions considered safe for all states using the teacher policy or previously identified safe actions in a level-based exploration strategy, which requires storing large amounts of tuples.

García and Fernández (2011, 2012); García et al. (2013) present a new definition of risk based on *unknown* and *known* space, and that reflects the author's intuition as to when human learners require advice. Certainly, humans benefit from help when they are in novel or unknown situations. The authors use a case-based approach to detect such situations. Traditionally, case-based approaches use a *density threshold* θ in order to determine when a new case should be added to the memory. When the distance of the nearest neighbor to the query state is greater than θ, a new case is added. García and Fernández (2011, 2012) propose the PI-SRL algorithm in which a risk function, $\varrho^B(s)$, measures the risk of a state in terms of its similarity to previously visited (and secure) states in a case base, $B = \{c_1 \ldots, c_n\}$. Every case c_i consists of a state-action pair (s_i, a_i) the agent has experienced in the past and with an associated value $V(s_i)$. When the distance to the closest state in the case base is larger than a parameter θ, the risk is maximum, while the risk is minimal if this distance is less than θ. Therefore, in that work, the risk function is defined as a step function. However, to define the risk function in such a way demonstrates that it may still produce damage in the learning agent. The reason is that to follow the teacher's advice only when the distance to the closest known state is larger than θ may be too late. On the other hand, one would expect that the risk function is progressive. Therefore, while the limit of θ is approaching, the risk should start to grow, and the learning agent could start to use the teacher's advice. In a further work, García et al. (2013) propose the use of a progressive risk function that determines the probability of following the teacher advice. The integration of this advice together with the π-reuse exploration strategy (Fernández and Veloso, 2006; Fernández et al., 2010) results in the PR-SRL algorithm (García et al., 2013). The π-reuse

exploration strategy allows the agent to use a past policy, Π_{past}, with probability ϕ, explores with probability ϵ, and exploits the current policy, Π_{new}, with probability $1-\phi-\epsilon$. In the PR-SRL algorithm the past policy Π_{past} is replaced by the teacher policy, the new policy to be learned Π_{new} is replaced by the case base policy in B, and the parameter ψ is replaced by a sigmoid risk function, $\varrho^B(s)$, which computes the probability of teacher's advice.

Hailu and Sommer (1998) also associates the concept of risk to the concept of distance to distinguish novel situations. In this case, the learner agent consists of a feedforward neural network made up of RBF neurons in the input layer and a stochastic neuron in the output layer. Each neuron represents a localized receptive field of width $\sum$ that covers a hyper-sphere of the input space. The learner agent has no neurons at the beginning of the learning process. At this point, the robot perceives a new state s, and it cannot generalize the situation. Therefore, it invokes the teacher which sends its action to the learner. The learner receives the action and adds a neuron. When a new state is perceived, the learner identifies the first winning neuron closest to the state perceived. If the distance of the winning neuron is larger than $\sum$, the state is regarded as novel and the learner invokes the teacher and adds a neuron that generalized the new situation perceived. In this way, the learner grows gradually, thus increasing its competence.

Instead, Geramifard et al. (2011); Geramifard (2012); Geramifard et al. (2013) assume the presence of a function named safe: $S \times A \rightarrow \{0,1\}$ that returns *true* if the carrying out of action a at state s will result in a catastrophic outcome and *false* otherwise. At every time step, if the learner agent considers the action to be safe, it will be carried out during the next step, otherwise the learner invokes the teacher's action which is assumed to be safe. The safe function is based on the existence of a *constrained* function: $S \rightarrow \{0,1\}$, which indicates whether being in a particular state is allowed or not. Risk is defined as the probability of visiting any of the constrained states. However, this approach presents two main drawbacks: (i) modeling the *constrained* function correctly, and (ii) it assumes the system model is known or partially known although it is only used for risk analysis. Jessica Vleugel and Gelens (2011) proposed an approach where the unsafe states are previously labeled. In this method, the learner agent asks for advice when it reaches a previously labeled unsafe state.

Finally, although more related to learning from demonstration, Chernova and Veloso (2009) also use the confidence parameter to select between agent autonomy and a request for a demonstration based on the measure of action-selection confidence returned by a classifier. Confidence below a given threshold indicates that the agent is uncertain about which action to take, so it seeks help from the teacher in the form of a demonstration, improving the policy and increasing the confidences for future similar states.

4.1.3.2 The Teacher Provides Advice

In the approaches grouped in this trend, the teacher provides actions or information whenever the teacher feels its help is necessary. Therefore in all these approaches, an explicit mechanism in the learner agent to recognize (and express) its need for advice is not necessary. Therefore, a new open question arises namely what is the best time for a teacher to provide information. Clouse and Utgoff (1992) add a simple interface to a RL algorithm to allow a human teacher to interact with the learner agent during the learning process. In this work, the teacher monitors the learner's behavior and provides an action when it considers

it necessary. This action is supposed to be the correct choice to be made in that state. Otherwise, the learner agent takes its own action based on its developing policy. Maclin and Shavlik (1996) use a similar approach in their RATLE algorithm, where a teacher at any point can interrupt the agent learning execution and types its advice using simple IF-THEN rules and more complex rules involving multiple steps. Thomaz and Breazeal (2006, 2008) also introduce an interface between the human teacher and the learner agent. The human teacher advises the agent in two ways: using an interactive reward interface and sending human advice messages. Through the first, the teacher introduces a reward signal $r \in [-1, 1]$ for each step of the learning process. The human teacher receives visual feedback enabling him/her to tune the reward signal before sending it to the agent. Through the second, the agent begins each iteration of the learning loop by pausing to allow the teacher time to introduce advice messages (1.5 seconds). If advice messages are received, the agent will choose randomly between the set of actions derived from these messages. Otherwise, the agent chooses randomly between the set of actions with the highest *Q-values*. In a similar way, Suay and Chernova (2011) also use a teacher to provide rewards and guidance to an Aldebaran Nao humanoid robot. Instead, Vidal et al. (2013) presents a learning algorithm in which the reinforcement comes from a human teacher that is seeing what the robot does. This teacher is able to punish the robot by simply pressing a button on a wireless joystick. When the teacher presses the button to give the robot negative reinforcement, the robot learns from it and transfers the control to the teacher, so that the teacher will be able to move the robot and place it in a suitable position to go on learning. Once this manual control is over, the teacher will press a second button to continue the learning process. In all these approaches, the teacher decides when to provide information based on him/her own feelings (i.e., when the teacher deems it necessary), but there is no metric or rule as to the best time for to do it. Additionally, using the constant monitoring of the learner agent by the teacher, it might not be desirable in practice due to the time or cost implications.

Maclin et al. (2005b) implement the advice as a set of rules provided by a teacher. When a rule applies (i.e., the LHS is satisfied) it is used to say the *Q-value* for some action should be *high* or *low*. The experiments are conducted using the keep away domain, and an example of the rule suggests the keeper with the ball should hold it when the nearest taker is at least 8 metres away (i.e., $Q(hold) \geq high$). In a later work, Maclin et al. (2005a) extends the previous approach to recommend that some action is preferred over another in a specified set of states. Therefore, the teacher is giving advice on policies rather than *Q-values*, which is a more natural way for humans to present advice (e.g., when the nearest taker is at least 8 meters, holding the ball is preferred to passing it). Similarly, Torrey et al. (2005) also used a set of rules, but these rules were learned in a previous related task using inductive logic programming following a transfer learning approach (Taylor and Stone, 2009). The user can also add supplementary teacher advice on the learned rules before the learning process begins. During the learning process, the learner agent receives the teacher's advice and the agent can follow it, refine it, or ignore it according to its value.

Walsh et al. (2011) use a teacher that analyzes the return of the learner agent for each episode. This return provides enough information for the teacher to decide whether or not to provide a demonstration. For each episode, if the return of the agent is lower than a certain measurement, the teacher decides to show a demonstration of that episode starting

at the same initial state. In this way, the agent learns concepts that it cannot efficiently learn on its own.

4.1.3.3 Other Approaches

In other approaches, the control of the teacher and the learner agent in the advice-taking interaction is not pre-defined. Rosenstein and Barto (2002, 2004) present a supervised RL algorithm that computes a composite action that is a weighted average of the action suggested by the teacher (a_T) and the exploratory action suggested by the evaluation function (a_E) using

$$a = ka_E + (1 - k)a_T \tag{12}$$

In Equation 12, $a_E = a_A + N(0, \sigma)$, where a_A is the action derived from the policy of the agent, $\pi_A(s)$, for a given state s; and $N(0, \sigma)$ is a normal distribution. The parameter k can be used to interpolate between an *ask for help* approach and an approach in which the teacher has the main role. Therefore, k determines the level of control, or autonomy, on the parts of the learner agent and the teacher. On the one hand, the learner agent can set $k = 1$ if it is fully confidence about the action to be taken. Instead, it can set the value of k close to 0 whenever it needs help from its teacher obtaining an *ask for help* approach (Section 4.1.3.1). On the other hand, the teacher can set $k = 0$ whenever it loses confidence in the autonomous behavior of the learner agent similarly to the approaches in Section 4.1.3.2. It is important to note that the proposed way of combining the action suggested by the teacher and the exploratory action is originally conceived for continuous action spaces but it could be extended to a discrete action space as well, by considering distributions over actions.

Kuhlmann et al. (2004) also computes an action using the suggestions of the teacher and the agent. In this work, the teacher generates values for the possible actions in the current world state. It is implemented as a set of rules. If a rule applies, the corresponding action value is increased or decreased by a constant amount. The values generated by the teacher are added to those generated by the learning agent. The final action selected is the action with the greatest final composite value. On the other hand, Judah et al. (2010) use a teacher that is allowed to observe the execution of the agent's current policy and then scroll back and forth along the trajectory and mark any of the available actions in any state as good or bad. The learner agent uses these suggestions and the trajectories generated by itself to compose the agent policy that maximizes the return in the environment. Moreno et al. (2004), by extending the approach proposed by Iglesias et al. (1998b,a), computes an action as the combination of the actions suggested by different teachers. At every time step, each teacher produces a vector of utilities u which contains a value $u(s, a_i) \in [0, 1]$ for each action a_i that it is possible to carry out in the current state s. Then the teacher's vectors are amalgamated into a one single vector, $w(a)$. This vector and an exploratory vector, $e(a)$, computed as $e(a) = 1 - w(a)$, are used to draw up a final *decision vector* which indicates which actions are the most suitable for the current state.

Torrey and Taylor (2012) present an algorithm in which the advice probability depends on the relationship between the learner agent's confidence and the teacher's confidence. In states where the teacher has much greater confidence than that of the student, it gives advice with a greater probability. As the agent's confidence in a state grows, the advice

probability decreases. Other approaches are based on interleaving episodes carried out by a teacher with normal exploration episodes. This mixture of teacher and normal exploration makes it easier for the RL algorithms to distinguish between beneficial and poor or unsafe actions. Lin (1991, 1992) use two different interleaving strategies. In the first, after each complete episode, the learner agent replays n demonstrations chosen randomly from the most recent 100 experienced demonstrations, with recent lessons exponentially more likely to be chosen. In the second, after each complete episode, the agent also stochastically chooses already taught demonstrations for replay. Driessens and Džeroski (2004) also use an interleaving strategy and compares its influence when it is supplied at different frequencies.

In other works the advice takes the form of a reward. In this case, the teacher judges the quality of the agent's behavior sending a feedback signal that can be mapped onto a scalar value (e.g. by pressing a button or verbal feedback of "good" and "bad") (Thomaz and Breazeal, 2006; Knox and Stone, 2009). In contrast to RL, a learner agent seeks to directly maximize the short-term reinforcement given by the teacher. Other works combine the reward function of the MDP and that provided by the teacher (Knox and Stone, 2010; Knox et al., 2011).

4.2 Risk-directed Exploration

In this trend the exploration process is carried out by taking into account a defined risk metric. Gehring and Precup (2013) defines a risk metric based on the notion of *controllability*. If a particular state (or state-action pair) yields a considerable variability in the temporal-difference error signal, it is less controllable. The authors compute the controllability of a state-action pair as defined in

$$C(s_t, a_t) \leftarrow C(s_t, a_t) - \alpha'(|\delta_t| + C(s_t, a_t)) \tag{13}$$

The exploration algorithm uses controllability as an exploration bonus, picking actions greedily according to $Q(s_t, a_t) + wC(s_t, a_t)$. In this way, the agent is encouraged to seek controllable regions of the environment. This approach is successfully applied to the helicopter hovering control domain (García and Fernández, 2012) used in the RL Competition. Law (2005) uses a risk metric to guide the exploration process. In this case, the measurement of risk for a particular action in a given state is the weighted sum of the entropy (i.e., the stochasticity of the outcomes of a given action in a given state) and normalized expected return of that action. The risk measure of an action, $U(s, a)$, is combined with the action value to form the *risk-adjusted utility* of an action, i.e., $p(1 - U(s_t, a_t)) + (1 - p)Q(s_t, a_t)$ where $p \in [0, 1]$. The first term measures the safety value of an action, while the second term measures the long term utility of that action. The *risk-adjusted utility* is replaced in the Boltzmann function instead of the Q-values in order to safely guide the exploration process. However, the main drawback of these approaches is that the mechanism of risk avoidance is achieved by learning the risk values of actions during learning, i.e., when the functions $C(s_t, a_t)$ and $U(s_t, a_t)$ are correctly approximated. But it would be desirable to prevent risk situations from the early steps in the learning process.

Finally, it is important to note that the approaches considered here are similar in spirit to those in section 3.3. As an example, if the controllability was added as a constraint, then it becomes a constrained optimization criterion and, hence, this approach should be

included in Section 3.3. However, we would obtain a different algorithm with different results. The approaches considered here only introduce a bias in the exploration process without satisfying hard constraints or constraints with a fixed probability of violation. Instead, the approaches in Section 3.3 must fulfill all the given constraints (although it is considered a fixed probability of constraint violation).

5. Discussion and Open Issues

In this Section, we complete the study of the techniques surveyed in this paper by classifying them across different dimensions. Additionally, we summarize the main advantages and drawbacks for each group of techniques in order to define future work directions.

5.1 Characterization of Safe RL Algorithms

As highlighted in the previous sections, current approaches to Safe RL have been designed to address a wide variety of problems where the risk considered and its detection have a large variety of forms. Table 2 analyzes most of the surveyed approaches across four dimensions.

5.1.1 Allowed Learner

We distinguish various RL approaches used in Safe RL. The *model free* methods such as Q-Learning (Sutton and Barto, 1998) which learn by backing up experienced rewards over time. The *model-based* methods which attempt to estimate the true model of the environment by interacting with it. The *policy search* methods which directly modify a policy over time to increase the expected long-term reward by using search or other optimization techniques. Finally, the *relational* RL methods which use a different state/action representation (relational or first-order language).

Table 2 shows that most of the approaches correspond to *model-free* RL algorithms. *Model-based* approaches are also used but few such methods handle continuous or large state and action spaces (Abbeel and Ng, 2005) and they generally have trouble scaling to tasks with many state and action variables due to the curse of dimensionality. *Model-based* approaches demand run exploration policies until they have an accurate model of the entire MDP (or at least the *reachable* parts of it). This makes many *model-based* approaches require exhaustive exploration that can take an undesirably long time for complex systems. Additionally, an aggressive exploration policy in order to build an accurate model can lead to catastrophic consequences. Therefore, *model-based* approaches suffer a similar exploration problem as *model-free* approaches, but in this case the question is: how can we safely explore the relevant parts of the state/action spaces to build up a sufficiently accurate dynamics model from which derive a good policy? Abbeel (2008) offers a solution to these problems by learning a dynamics model from teacher demonstrations. *Policy search* and *Relational* RL methods are also identified as techniques that can be applied to risky domains, but usually refer to the use of bootstrapping approaches.

5.1.2 Space Complexity

The column entitled by *Space* in Table 2 describes the complexity of the state and action spaces of the domains where the method has been used. The S refers to continuous or large

Citation	Allowed Learner	Spaces	Risk	Exploration
Modifying the Optimization Criterion: Section 3				
Worst Case Criterion: Section 3.1				
Heger (1994b)	model free	s/a	Var	$greedy$
Gaskett (2003)	model free	s/a	Var	$greedy$
Risk-Sensitive Criterion: Section 3.2				
Borkar (2002)	model free	s/a	Var	$greedy$
Mihatsch and Neuneier (2002)	model free	S/a	TD-error	$\epsilon - greedy$
Campos and Langlois (2003)	model free	S/a	TD-error	$\epsilon - greedy$
Geibel and Wysotzki (2005)	model free	S/a	error states	$greedy$
Constrained Criterion: Section 3.3				
Geibel (2006)	model free	s/a	$E(R) \geq \alpha$	greedy
Castro et al. (2012)	model free	S/A	$Var(R) \leq \alpha$	softmax
Moldovan and Abbeel (2011, 2012a)	model based	s/a	ergodicity	bonuses
Abe et al. (2010)	model free	s/a	ad-hoc constraints	greedy
Modifying the Exploration Process: Section 4				
Providing Initial Knowledge: Section 4.1.1				
Driessens and Džeroski (2004)	relational	S/a	initial exploration	softmax
Smart and Kaelbling (2000)	model free	S/A	initial exploration	gaussian
Martín H. and Lope (2009)	policy search	S/A	initial exploration	evolving NN
Koppejan and Whiteson (2011)	policy search	S/A	initial exploration	evolving NN
Maire (2005)	model free	s/a	initial exploration	$greedy$
Deriving a Policy from a Finite Set of Demonstrations: Section 4.1.2				
Abbeel and Ng (2005)	model based	S/A	accurate model	$greedy$
Tang et al. (2010)	model based	S/A	accurate model	$greedy$
Using Teacher Advice: Section 4.1.3				
Clouse (1997)	model free	S/a	similar $Q - values$	softmax
Hans et al. (2008)	model free	s/a	fatal transitions	level-based
García et al. (2013)	model free	S/A	unknown states	gaussian
Geramifard (2012)	model based	S/A	constrained states	softmax
Clouse and Utgoff (1992)	model free	s/a	human	$\epsilon - greedy$
Maclin and Shavlik (1996)	model free	s/a	human	softmax
Thomaz and Breazeal (2006, 2008)	model free	S/a	human	softmax
Walsh et al. (2011)	model based	s/a	$E(R) \leq \alpha$	R_{max}
Rosenstein and Barto (2002, 2004)	model free	S/A	agent/teacher confidence	gaussian
Kuhlmann et al. (2004)	model free	S/a	human	$\epsilon - greedy$
Torrey and Taylor (2012)	model free	S/a	agent/teacher confidence	$\epsilon - greedy$
Risk-directed Exploration: Section 4.2				
Gehring and Precup (2013)	model free	S/a	TD-error	risk directed
Law (2005)	model based	s/a	entropy and E(R)	risk directed

Table 2: This table lists most of the Safe RL methods discussed in this survey and classifies each in terms of four dimensions.

state space, and s to discrete and small state space. The same interpretation can be applied analogously to A and a in the case of the action space. In this way, S/a means that the method has been applied to domains with continuous or large state space and discrete and small action space.

Most of the research on RL has studied solutions to finite MDPs. On the other hand, learning in real-world environments requires handling with continuous state and action spaces. While several studies have focused on problems with continuous states, little attention has been paid to tasks involving continuous actions. These conclusions can also be obtained for Safe RL from Table 2 where most of the approaches address finite MDPs (Heger, 1994a; Gaskett, 2003; Moldovan and Abbeel, 2012a) and problems with continuous or large state spaces (Mihatsch and Neuneier, 2002; Geibel and Wysotzki, 2005; Thomaz and Breazeal, 2008), and much fewer approaches address problems with continuous or large state and action spaces (Abbeel et al., 2009; García and Fernández, 2012).

5.1.3 Risk

The forms of risk considered in this survey are also listed in Table 2. These forms are related to the variance of the return or its worst possible outcome (entitled *Var* in Table 2), to the temporal differences (entitled *TD-error*), to *error states*, to constraints related to the expected return (entitled by $E(R) \geq \alpha$ or $E(R) \leq \alpha$) or the variance of the return (entitled $Var(R) \leq \alpha$), to the *ergodicity* concept, to the effects of initial exploration in early stages in unknown environments (entitled *initial exploration*), to the obtaining of accurate models used later to derive a policy (entitled *accurate model*), to *similar Q-values*, to *fatal transitions*, to *unknown states*, to human decisions which determine what is considered a risk situation and when to provide help (entitled *human*), and to the degree of confidence both the teacher and the agent (denoted by *agent/teacher confidence*).

Table 2 shows the wide variety of forms of risk considered in the literature. This makes the drawing up of a benchmark problem difficult, or the identification of an environment to test different notions of risk. That is, in most cases, the approaches have different safety objectives and, hence, they result in different safe policies. For instance, the $\hat{Q} - Learning$ algorithm by Heger (1994b) leads to a safe policy completely different from that obtained by the method proposed by García and Fernández (2012). The applicability of one or another depends on the particular domain we are considering, and the type of risk it involves. Additionally, it is important to note that, in some cases, the risk metric selected places restrictions on which RL algorithm is used. For instance, the risk related to the *ergodicity* requires the model of the MDP to be known or learned.

5.1.4 Exploration

Table 2 also describes the exploration/exploitation strategy used for action selection. The *greedy* strategy is referred to the $\epsilon - greedy$ strategy where the ϵ is fixed at 0. For instance, Gaskett (2003) applies this exploration strategy and uses the inherent stochasticity of the environment to explore efficiently. In the classic $\epsilon - greedy$ action selection strategy the agent selects a random action with chance ϵ and the current best action with probability $1 - \epsilon$. In *softmax* action selection, the action probabilities are ranked according to their value estimates. The *gaussian* exploration is related to continuous action spaces and

at every moment the action is selected by sampling from a Gaussian distribution with the mean at the current best action. The evolution of Neural Networks (NN) are used in policy search methods to explore around the policy space. In the *level-based* exploration proposed by Hans et al. (2008), the agent tries to explore all actions considered safe (i.e., it does not lead to a fatal transition) for each state gradually. The *exploration bonuses* adds a bonus to states with higher potential of learning (Baranes and Oudeyer, 2009), or with higher uncertainty (Brafman and Tennenholtz, 2003). Regarding the latter, the R_{max} exploration is related to model-based approaches and it divides states into two groups, known and unknown states, and focuses on reaching unknown states by assigning them maximum possible values (Brafman and Tennenholtz, 2003). Finally, the *risk directed* exploration uses a risk metric to guide the exploration process.

As the most widely used methods are *model free* in discrete and small action space, the exploration strategies most commonly used are $\epsilon - greedy$ and *softmax*. However, due to their random component of action selection, there is a certain chance of exploring dangerous or undesirable states. This chance affects the approaches differently using these strategies in Section 3 and Section 4. Most of the approaches in Section 3 are not interested in obtaining a safe exploration during the learning process; they are more interested in obtaining a safe policy at the end (Heger, 1994b; Gaskett, 2003; Mihatsch and Neuneier, 2002; Campos and Langlois, 2003). Therefore, the random component of these strategies is not so relevant for these approaches. In the approaches in Section 3.3 the use of these exploration strategies is limited to the space considered safe (i.e., that fulfills the constraints) (Geibel, 2006; Castro et al., 2012; Abe et al., 2010), which limits visiting undesirable regions despite this random component. Instead, most of the approaches in Section 4 address the problem of safe exploration using these exploration strategies. The approaches in Section 4.1.1 introduce an initial bias into the exploration space which mitigate (but do not prevent) the number of visits to undesirable states that produce these exploration strategies (Driessens and Džeroski, 2004; Maire, 2005). The approaches in Section 4.1.2 derive a model from a finite set of demonstrations, and then use this model to derive a policy greedily in an off-line and, hence, safe manner (Abbeel et al., 2009; Tang et al., 2010). The approaches in Section 4.1.3 combine the advice provided by the teacher with these exploration strategies to produce a safe exploration process (Clouse, 1997; Maclin et al., 2005a; Thomaz and Breazeal, 2006, 2008; Torrey and Taylor, 2012). For instance, the *softmax* exploration is used by Thomaz and Breazeal (2008) to select an action if no advice is introduced by the human. Otherwise, it selects a random action from among that derived from the advice introduced.

Other exploration strategies based on exploration bonuses such as R_{max} are related to the use of model-based algorithms (Walsh et al., 2011; Moldovan and Abbeel, 2012a). This exploration technique was first presented for finite MDPs (Brafman and Tennenholtz, 2003), but also there are versions for continuous state space where the number of samples required to learn an accurate model increases as the number of dimensions of the space grows (Nouri, 2011). Moldovan and Abbeel (2012a) use an adapted version of R_{max} where the exploration bonus of moving between two states is proportional to the number of neighboring unknown states that would be uncovered as a result of the move. It is important to note that R_{max} exploration by itself may be considered unsafe since it is encouraged to explore areas of *unknown* space, and other authors establish a direct relationship between the risk and the

unknown concept (García and Fernández, 2012). However, Moldovan and Abbeel (2012a) use this exploration method to explore from among the policies whose preserve the *ergodicity* and are considered safe.

The gaussian exploration also introduces a random component in the algorithms proposed by García and Fernández (2012) and Smart and Kaelbling (2000). As regards the former, when an *unknown* state is found, the action is carried out by the teacher, otherwise, small amounts of Gaussian noise are randomly added to the greedy actions of the current policy. This ensures a safe exploration. As regards the latter, at the beginning of the learning process, the gaussian noise is added to greedy actions derived from a policy previously bootstrapped by teacher demonstrations. Other approaches uses a risk metric to direct the safe exploration based on the *temporal differences* (Gehring and Precup, 2013) or the weighted sum of an entropy measurement and the expected return (Law, 2005). Hans et al. (2008) uses a level-based exploration approach where the safe exploration is carried out gradually by exploring all the considered safe actions for each state. This exploration approach seems suitable for finite MDPs, but in MDPs with large state and action spaces, this exhaustive exploration is computationally intractable. Finally, the safe exploration conducted by Martín H. and Lope (2009) and Koppejan and Whiteson (2011) is due to the *population seeding* of the initial population which biases the subsequent exploratory process.

5.2 Discussion

Table 3 summarizes the main advantages and drawbacks of the approaches surveyed in this paper. Attending to the main advantages and drawbacks identified for each approach, we believe that there are several criteria that must be analyzed when developing Safe RL algorithms and risk metrics.

5.2.1 Selection of the Risk Metric

The algorithms based on the variance of the return (Sato et al., 2002; Borkar, 2002; Osogami, 2012) or its worst possible outcome (Heger, 1994b; Coraluppi, 1997) are not generalizable to problems in which a policy with a small variance can produce a large risk. To clarify this statement, we have reproduced the example set out by Geibel and Wysotzki (Geibel and Wysotzki, 2005). The example is a grid-world problem in which there are error states (i.e., undesirable or dangerous situations), and two goal states (one of them is placed next to the error state, and the other in a safer part of the state space). This grid-world is detailed in Figure 4.

The agent is able to move North, South, East, or West. With a probability of 0.21, the agent is not transported to the desired direction but to one of the three remaining directions. The agent receives a reward of 1 if it enters a goal state and a reward of 0 in every other case. There is no explicit negative reward for entering an error state, but when the agent enters it, the learning episode ends. In this domain, we found that a policy leading to the error states as fast as possible does not have a higher variance than one that reaches the goal states as fast as possible. Therefore, a policy with a small variance can therefore have a large risk, because this policy can lead the agent to error states. Additionally, we can see that all policies have the same worst case outcome and, hence, this optimization

Advantages	Drawbacks
Modifying the Optimization Criterion: Section 3	
Worst Case Criterion: Section 3.1	
• Useful when avoiding rare occurrences of large negative return is imperative	• Overly pessimistic • Variance of the return not generalizable to arbitrary domains • The true long term utility of the actions are lost • Not detect risky situations from the early steps
Risk-Sensitive Criterion: Section 3.2	
• Easy switch between risk-averse and risk-seeking behavior • Detect long-term risk situations	• If a conservative criterion is used, the policy may be overly pessimistic • The true long term utility of the actions are lost • Not detect risky situations from the early steps
Constrained Criterion: Section 3.3	
• Intuitively it seems a natural solution to the problem of safe exploration: the exploration is carried out only in the region of space considered safe (i.e., that fulfills the constraints)	• Many of these problems are computationally intractable, which difficult the formulation of RL algorithms • Correct selection of the parameter constraints • Constraints related to the return or its variance are not generalizable to arbitrary domains
Modifying the Exploration Process: Section 4	
Providing Initial Knowledge: Section 4.1.1	
• Bootstrap the value function approximation and lead the agent through the more relevant regions of the space from the earliest steps of the learning process	• Bias introduced may produce suboptimal policies • Exploration process following the initial training phase can result in visiting catastrophic states • Difficult initialization in complex structures
Deriving a Policy from a Finite set of Demonstrations: Section 4.1.2	
• The learning algorithm derives a policy from a finite set of demonstrations in an off-line and, hence, safe manner	• Learner performance is heavily limited by the quality of the teacher's demonstrations • How the agent should act when it encounters a state for which no demonstration exists?
Using Teacher Advice: Section 4.1.3	
• Guide the exploration process keeping the agent far away from catastrophic states from the earliest steps of the learning process • In ask for help approaches – Automatic detection of risk and request for advice when needed – Generalizable mechanisms of risk detection	• In ask for help approaches – Detect short term risk situations but not long term • In teacher provides advices approaches: – Teacher decides when to provide information based on its own feelings – Constant monitoring of the learner agent by the teacher might not be desirable in practice
Risk-directed Exploration: Section 4.2	
• Detect long-term risk situations	• Not detect risky situations from the early steps

Table 3: This table lists the main advantages and drawbacks of the Safe RL methods discussed in this survey.

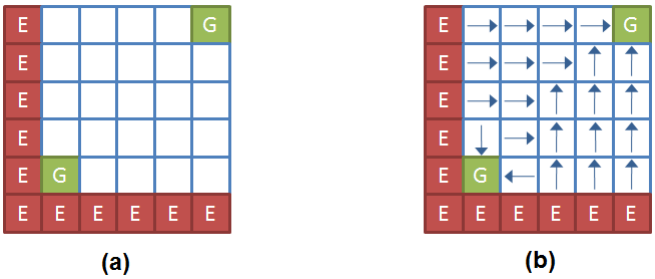


Figure 4: (a) The grid-world domain. (b) The minimum risk policy computed by Geibel and Wysotzki (Geibel and Wysotzki, 2005).

criterion is also unsuitable for this kind of risky domain. Accordingly, the variance or the worst-outcome criterion may not be generalizable to any risky domain.

The risk metric considered should be easily generalizable to any risky domain and be independent of the nature of the task. The risk based on the level of knowledge of a particular state is an example of generalizable risk metric (Hailu and Sommer, 1998; García and Fernández, 2012; Chernova and Veloso, 2009; Torrey and Taylor, 2012). This level of knowledge about a state can be based on the distance between the known and the unknown space (Hailu and Sommer, 1998; García and Fernández, 2012), on the difference between the highest and lowest Q-values (Clouse, 1997), or on the number of times an agent has made a non-trivial Q-value update in a state (Torrey and Taylor, 2012). In this sense, other knowledge level metrics in a state proposed in the literature can be used in Safe RL to identify potentially catastrophic situations (e.g., those based on the number of times an agent visits a state (Kearns and Singh, 2002), or on the *knownness criterion* (Nouri and Littman, 2008). The study of risk metrics easily generalizable to arbitrary domains is still an open issue in Safe RL.

5.2.2 Selection of the Optimization Criterion

We distinguish five possible situations based on two criteria: (i) the kind of optimization criterion (long term optimization of risk or risk-neutral), and (ii) the kind of risk detection (immediate and/or long term risk detection).

- *Long term optimization of risk.* In this case, we are interested in maximizing a long term measurement which considers some form of risk. This is common to most of the works reviewed in Section 3 where the risk-averse behavior is induced by transforming the classic optimization criterion of RL by introducing a risk metric. However, it seems difficult to find an optimization objective which correctly models our intuition of risk awareness. In Section 3.1, most of the approaches are updated based on a conservative criterion and the resulting policy tends to be overly pessimistic. Something similar happens with the approaches in Section 3.2 based on the variance of the return. In addition, the transformation of the optimization criterion produces a distortion in the action values and the true long term utility of the actions are lost.

Finally, most of these approaches repeatedly visit risk situations until the optimization criterion is correctly approximated and, subsequently, avoid dangerous situations. As an example of the latter, the optimization criterion used by Geibel and Wysotzki (2005) helps to reduce the number of visits to error states once the risk function is approximated (Geibel and Wysotzki, 2005; García and Fernández, 2012). It could be interesting to avoid future risk situations since it can provide a margin of reaction before reaching a point where it is unavoidable to reach an error state.

- *Detection of immediate risk.* We are interested in detecting and reacting to immediate risk situations from the early steps of the learning process while the classic optimization criterion remains. This second one is related to the approaches in Section 4. The worst thing that can happen to an agent in an environment may have a high return in the long term, but fatal consequences in the short term. The Safe RL algorithm should incorporate a mechanism to detect and react to immediate risk by manifesting different risk attitudes, while leaving the optimization criterion untouched. The ability to assess the amount of immediate risk in any action allows one to make step-by-step tradeoffs between attaining and abandoning the goal (i.e., the maximization of the return) in order to ensure the safety of the agent, the learning system and any external entity. The teacher advice approaches presented in Section 4.1.3.1 and Section 4.1.3.2 are good examples of this property. In these approaches, when a risk situation is detected by the agent or the teacher, the teacher provides safe information to the agent to prevent fatal situations. The main drawback of most of these approaches is that the risk is detected on the basis of the current state (Clouse, 1997; Geramifard, 2012; García and Fernández, 2012), and it may be too late to react.

- *Long term optimization of risk and immediate risk.* We are interested in maximizing a long term measurement which considers some form of risk and, at the same time, detects and reacts to immediate risk situations from the early steps of the learning process. The two previous approaches can be integrated into a same Safe RL algorithm. As an example, Geibel's approach (Geibel and Wysotzki, 2005) can be combined with an approach based on the level of knowledge of the current state from Section 4.1.3.1. The learner agent can ask for help in little known (Torrey and Taylor, 2012) or unknown states (Hailu and Sommer, 1998; García and Fernández, 2012) mitigating the effects of risk situations from early steps of the learning process. At the same time, the exploration directed by Geibel's optimization criterion, which include the risk function, $\rho^{\pi}(s)$, ensures the selection of *safe* actions preventing long-term risk situations once the risk function is correctly approximated. The development of these Safe RL algorithms is an area open for future research.

- *Detection of long term risk.* We are interested in detecting long-term risk situations when the risk function is correctly approximated, but not in detect and react to immediate risk situations from the early steps in the learning process. In addition, the classic optimization criterion remains. This is related to the approaches in Section 4.2. In this case, a risk metric is used to guide the exploration of the state and action spaces in a risk-directed exploration process based on the controllability (Gehring and Precup, 2013) or on the entropy (Law, 2005). In these approaches, the value function

is learned separately, so optimal values and policies can be recovered if desired at a later time. As regards the latter, it would be interesting to decouple Geibel's risk function based on error states, $\rho^{\pi}(s)$, from the value function. In this way, it would be possible to analyze the effect of considering risk as part of the objective function with respect to considering risk only for risk-directed exploration while the objective function remains.

- *Detection of immediate and long term risk.* In this case, we are interested in detecting long-term risk situations when the risk function is correctly approximated, and in detect and react to immediate risk situations from the early steps in the learning process. The combination of immediate and long term risk detection mechanisms is still an open issue in Safe RL. As an example, the approaches in Section 4.1.3.1 could be combined with the approaches in Section 4.2. The learner agent can ask for help in little known (Torrey and Taylor, 2012) or unknown states (Hailu and Sommer, 1998; García and Fernández, 2012), and when the risk metric is greater than a certain threshold (e.g., $C(s_t, a_t) \geq w$ or $\rho^{\pi}(s) \geq w$). The first helps to mitigate the effects of immediate risk, the second immediate risk situations to be prevented in the long term through the delegation of the action taking to an external teacher instead of the ongoing exploratory process. At the same time, the risk-directed exploration mitigates the selection of actions which bring the risk situations closer.

5.2.3 Selection of the Mechanism for Risk Detection

The mechanism for risk detection should be automatic and not based on the intuition of a human teacher. In most of the approaches in Section 4.1.3.2, the teacher decides when to provide information to the learner agent based on its own feelings, but there is no metric as to the best time to do it. It is important to be aware of the fact that this way of providing the information is highly non-deterministic, i.e., the same human teacher can give the agent information in certain situations but remain impassive in other scenarios that are very similar. Moreover, the teacher observer can change his/her mind as to what is risky or not while the agent is still learning.

5.2.4 Selection of the Learning Schema

Although *policy search* methods have been demonstrated to be good techniques for avoiding risky situations, their safe exploration was related to the incorporation of *teachers* in the initial population (Martín H. and Lope, 2009; Koppejan and Whiteson, 2011). The problem of extracting knowledge (e.g., on the *known space*) from the networks or their weights makes it almost impossible to incorporate mechanisms for safe exploration during the learning process. As regards *model-free* vs. *model-based* approaches, there is still an open debate within the RL community as to whether *model-based* or *model-free* could be shown to be clearly superior to the other. This debate can also be taken in Safe RL. *Model-based* methods have relative higher space and computational complexities and lower sample complexity than *model-free* methods (Strehl et al., 2006). In general, this prevents the use of *model-based* methods in large space and stochastic problems in which the approximation of an accurate model from which derive a good policy is not possible. However, recent *model-based* approaches have demonstrated successful handling with continuous state domains (Nouri,

2011; Hester and Stone, 2013). Having such a model is a useful entity for Safe RL: it allows the agent to predict the consequences of actions before they are taken, allowing the agent to generate virtual experience. Therefore, we consider that building models is an open issue and a key ingredient of research and progress in Safe RL. So far, as shown in Table 2, most of the Safe RL approaches are *model-free*.

5.2.5 Selection of the Exploration Strategy

The exploratory process is responsible for visits to undesirable states or risky situations but also for progressively improve the policies learned. Techniques such as that proposed in Section 4.1.2 do not require exploration, but only the exploitation of the learning model derived from the teacher demonstrations. However, without additional exploration, the policies learned are heavily limited by the teacher demonstrations. The methods in Section 3.3 carry out safe exploration from among the policies in the constrained space. This may be the more advisable intuitively, but many of these problems are computationally intractable, which makes the formulation of RL algorithms difficult for large space and stochastic tasks.

As regards the other strategies used, all of them lead to a risky behavior. In exploration methods such as R_{max} the algorithm still tends to end up generating (and using) exploration policies in its initial stage. Additionally, R_{max} follows the *optimism in the face of uncertainty* principle, which consists of assuming a higher return on the most unknown states. This *optimism* for reaching unknown states can produce an unsafe exploration, since other authors establish a direct relationship between unknown and dangerous situations (García and Fernández, 2012). Exploration methods such as $\epsilon - greedy$, *softmax*, or gaussian incorporates a random component which give rise to a certain chance of exploring dangerous or undesirable states. The risk-directed exploration conducted by the use of risk metrics requires the function to be correctly approximated beforehand to avoid risk situations. Therefore, we consider that if the exploration is carried out in the entire state and action spaces (i.e., without constraints restricting the explorable/safe space), whatever the exploration used, this should be carried out in combination with an automatic risk detection mechanism (able to detect immediate and/or long term risk situations from the early steps in the learning process) and abandoning the goal in order to ensure the safety of the agent (e.g., asking for help from a teacher). Finally, if the goal is to obtain a safe policy at the end, without worrying about the number of dangerous or undesirable situations that occur during the learning process, then the exploratory strategy used to learn this policy is not so relevant from a safety point of view.

6. Conclusions

In this paper we have presented a comprehensive survey on Safe Reinforcement Learning techniques used to address control problems in which it is important to respect safety constraints. In this survey, we have contributed with a categorization of Safe RL techniques. We first segment the Safe RL techniques into two fundamental trends: the approaches based on the modification of the optimization criterion, and those based on the modification of the exploration process. We use this structure to survey the existing literature highlighting the major advantages and drawbacks of the techniques presented. We present techniques created specifically to address domains with a diverse nature of risk (e.g., those based on

the variance of the return (Sato et al., 2002), on error states (Geibel and Wysotzki, 2005), or on the *controllability* concept (Gehring and Precup, 2013), and others that have not been created for this purpose, but have shown that their application to these domains can be effective in reducing the number of undesirable situations. Most of these techniques are described in Section 4.1 where external knowledge is used. As regards the latter, different forms of initialization have been shown to reduce the number of helicopter crashes successfully (Martín H. and Lope, 2009; Koppejan and Whiteson, 2011), or the number of times the agent moves into an obstacle in a Grid-World domain (Song et al., 2012; Maire, 2005); deriving a policy from a finite set of safe demonstrations provided by a teacher have also been shown to be a safe way of learning policies in risky domains (Abbeel et al., 2010); finally, the effectiveness of using teacher advice to provide actions in situations identified as dangerous has recently been demonstrated (García and Fernández, 2012; Geramifard, 2012).

The current proliferation of robots requires that the techniques used for learning tasks are safe. It has been shown that parameters learned in simulation often do not translate directly to reality, especially as heavy optimization on simulation has been observed to exploit the inevitable simplification of the simulator, thus creating a gap between simulation and application that reduces the usefulness of learning in simulation. In addition, autonomous robotic controllers must deal with a large number of factors such as the mechanical system and electrical characteristics of the robot, as well as the environmental complexity. Therefore, it is important to develop learning algorithms directly applicable to robots such as Safe RL algorithms since it could reduce the amount of damage incurred and, consequently, allow the lifespan of the robots to be extended.

Although Safe RL has proven to be a successful tool for learning policies which consider some form of risk, there are still many areas open for research, several of which we have identified in Section 5. As an example, the techniques based on the use of a risk function (Geibel and Wysotzki, 2005; Law, 2005; Gehring and Precup, 2013) have demonstrated their effectiveness in preventing risky situations once the risk function is correctly approximated. However, it would be desirable to prevent the risk situations from the early steps in the learning process. In this way, teacher advice techniques can be used to incorporate prior knowledge, thus mitigating the effects of immediate risk situations until the risk function is correctly approximated.

Acknowledgments

This paper has been partially supported by the Spanish Ministerio de Economía y Competitividad TIN2012-TIN2012-38079 and FEDER funds, and by the Innterconecta Programme 2011 project ITC-20111030 ADAPTA.

References

Pieter Abbeel. *Apprenticeship Learning and Reinforcement Learning with Application to Robotic Control*. PhD thesis, Stanford, CA, USA, 2008. AAI3332983.

Pieter Abbeel and Andrew Y. Ng. Exploration and apprenticeship learning in reinforcement learning. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

Pieter Abbeel, Adam Coates, Timothy Hunter, and Andrew Y. Ng. Autonomous autorotation of an rc helicopter. In *Experimental Robotics*, volume 54 of *Springer Tracts in Advanced Robotics*, pages 385–394. Springer Berlin Heidelberg, 2009.

Pieter Abbeel, Adam Coates, and Andrew Y. Ng. Autonomous helicopter aerobatics through apprenticeship learning. *International Journal of Robotic Research*, 29(13):1608–1639, 2010.

Naoki Abe, Prem Melville, Cezar Pendus, Chandan K. Reddy, David L. Jensen, Vince P. Thomas, James J. Bennett, Gary F. Anderson, Brent R. Cooley, Melissa Kowalczyk, Mark Domick, and Timothy Gardinier. Optimizing debt collections using constrained reinforcement learning. In *Proceedings of the 16th international conference on Knowledge discovery and data mining*, pages 75–84, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0055-1.

Eitan Altman. Asymptotic properties of constrained markov decision processes. Rapport de recherche RR-1598, INRIA, 1992.

Brenna Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, May 2009. ISSN 09218890.

Drew Bagnell. *Learning Decisions: Robustness, Uncertainty, and Approximation.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2004.

Drew Bagnell and Jeff Schneider. Robustness and exploration in policy-search based reinforcement learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 544–551, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-205-4.

Drew Bagnell, Andrew Ng, and Jeff Schneider. Solving uncertain markov decision problems. Technical report, Robotics Institute Carnegie Mellon, 2001.

A. Baranes and P. Y. Oudeyer. R-IAC: Robust intrinsically motivated exploration and active learning. *Autonomous Mental Development, IEEE Transactions on*, 1(3):155–169, October 2009. ISSN 1943-0604.

Arnab Basu, Tirthankar Bhattacharyya, and Vivek S. Borkar. A learning algorithm for risk-sensitive cost. *Mathematics of Operational Research*, 33(4):880–898, 2008.

Vivek S. Borkar. A sensitivity formula for risk-sensitive cost and the actor-critic algorithm. *Systems & Control Letters*, 44:339–346, 2001.

Vivek S. Borkar. Q-learning for risk-sensitive control. *Mathematics of Operations Research*, 27(2):294–311, May 2002. ISSN 0364-765X.

Ronen I. Brafman and Moshe Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, March 2003. ISSN 1532-4435.

Andriy Burkov and Brahim Chaib-draa. Reducing the complexity of multiagent reinforcement learning. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, page 44, 2007.

Pedro Campos and Thibault Langlois. Abalearn: Efficient self-play learning of the game abalone. In *INESC-ID, Neural Networks and Signal Processing Group*, 2003.

Dotan Di Castro, Aviv Tamar, and Shie Mannor. Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012.

Victor Uc Cetina. Autonomous agent learning using an actor-critic algorithm and behavior models. In *Proceedings of the 7th International Conference on Autonomous Agents and Multi-Agent Systems, Estoril, Portugal*, pages 1353–1356, 2008.

Suman Chakravorty and David C. Hyland. Minimax reinforcement learning. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, USA, 2003.

Yin Chang-Ming, Han-Xing Wang, and Fei Zhao. Risk-sensitive reinforcement learning algorithms with generalized average criterion. *Applied Mathematics and Mechanics*, 28 (3):405–416, March 2007. ISSN 0253-4827.

Sonia Chernova and Manuela M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34:1–25, 2009.

Kun-Jen Chung and Matthew J. Sobel. Discounted mdps: distribution functions and exponential utility maximization. *SIAM Journal on Control Optimization*, 25(1):49–62, January 1987. ISSN 0363-0129.

Jeffery A. Clouse. On integrating apprentice learning and reinforcement learning. Technical report, Amherst, MA, USA, 1997.

Jeffery A. Clouse and Paul E. Utgoff. A teaching method for reinforcement learning. In *ML*, pages 92–110. Morgan Kaufmann, 1992. ISBN 1-55860-247-X.

Stefano P. Coraluppi. *Optimal control of markov decision processes for performance and robustness*. University of Maryland, College Park, Md., 1997.

Stefano P. Coraluppi and Steven I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state markov decision processes. *Automatica*, 35:301–309, 1999.

Stefano P. Coraluppi and Steven I. Marcus. Mixed risk-neutral/minimax control of markov decision processes. *IEEE Transactions on Automatic Control*, 45(3):528–532, 2000.

Erick Delage and Shie Mannor. Percentile optimization for markov decision processes with parameter uncertainty. *Operations Research*, 58(1):203–213, January 2010.

Kurt Driessens and Sašo Džeroski. Integrating guidance into relational reinforcement learning. *Machine Learning*, 57(3):271–304, December 2004. ISSN 0885-6125.

Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multi-Agent Systems*, Hakodate, Japan, May 2006.

Fernando Fernández, Javier García, and Manuela M. Veloso. Probabilistic policy reuse for inter-task transfer learning. *Robotics and Autonomous Systems*, 58(7):866–871, 2010.

Javier García and Fernando Fernández. Safe reinforcement learning in high-risk tasks through policy improvement. In *Proceedings of the IEEE Symposium on Adaptive Dynamic Programming And Reinforcement Learning*, pages 76–83. IEEE, 2011.

Javier García and Fernando Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, December 2012.

Javier García, Daniel Acera, and Fernando Fernández. Safe reinforcement learning through probabilistic policy reuse. In *Proceedings of the 1st Multidisciplinary Conference on Reinforcement Learning and Decision Making*, October 2013.

Chris Gaskett. Reinforcement learning under circumstances beyond its control. In *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation*, 2003.

Clement Gehring and Doina Precup. Smart exploration in reinforcement learning using absolute temporal difference errors. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems, Saint Paul, MN, USA*, pages 1037–1044, 2013.

Peter Geibel. Reinforcement learning for mdps with constraints. In *Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany,*, volume 4212 of *Lecture Notes in Computer Science*, pages 646–653. Springer, 2006. ISBN 3-540-45375-X.

Peter Geibel and Fritz Wysotzki. Risk-sensitive reinforcement learning applied to control under constraints. *Journal of Artificial Intelligence Research*, 24:81–108, 2005.

Alborz Geramifard. *Practical Reinforcement Learning using Representation Learning and Safe Exploration for Large Scale Markov Decision Processes*. PhD thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, February 2012.

Alborz Geramifard, Joshua Redding, Nicholas Roy, and Jonathan P. How. UAV cooperative control with stochastic risk models. In *Proceedings of the American Control Conference*, pages 3393 – 3398, June 2011.

Alborz Geramifard, Joshua Redding, and JonathanP. How. Intelligent cooperative control architecture: A framework for performance improvement using safe learning. *Journal of Intelligent & Robotic Systems*, 72(1):83–103, 2013. ISSN 0921-0296.

Abhijit Gosavi. Reinforcement learning for model building and variance-penalized control. In *Proceedings of the Winter Simulation Conference*, pages 373–379. WSC, 2009.

Getachew Hailu and Gerald Sommer. Learning by biasing. In *Proceedings of the International Conference on Robotics and Automation*, pages 2168–2173. IEEE Computer Society, 1998. ISBN 0-7803-4301-8.

Alexander Hans, Daniel Schneegass, Anton M. Schäfer, and Steffen Udluft. Safe Exploration for Reinforcement Learning. In *Proceedings of the European Symposium on Artificial Neural Network*, pages 143–148, 2008.

Matthias Heger. *Risk and reinforcement learning: concepts and dynamic programming.* ZKW-Bericht. ZKW, 1994a.

Matthias Heger. Consideration of risk in reinforcement learning. In *Proceedings of the 11th International Conference on Machine Learning*, pages 105–111, 1994b.

Alfredo García Hernández-Díaz, Carlos A. Coello Coello, Fatima Perez, Rafael Caballero, Julián Molina Luque, and Luis V. Santana-Quintero. Seeding the initial population of a multi-objective evolutionary algorithm using gradient-based information. In *Proceedings of the IEEE Congress on Evolutionary Computation, Hong Kong, China*, pages 1617–1624, 2008.

Todd Hester and Peter Stone. TEXPLORE: Real-time sample-efficient reinforcement learning for robots. *Machine Learning*, 90(3), 2013.

Ronald A. Howard and James E. Matheson. Risk-sensitive markov decision processes. *Management Science*, 18(7):356–369, 1972.

Marcus Hutter. Self-optimizing and pareto-optimal policies in general environments based on bayes-mixtures. In *Proceedings of the 15th Annual Conference on Computational Learning Theory*, Sydney, Australia, 2002.

Roberto Iglesias, Carlos V. Regueiro, J. Correa, E. Sanchez, and Senen Barro. Improving wall following behaviour in a mobile robot using reinforcement learning. In *Proceedings of the International symposium on engineering of intelligent systems*, Tenerife (España), February 1998a. ISBN 3-906454-12-6.

Roberto Iglesias, Carlos V. Regueiro, J.Correa, and Senen Barro. Supervised reinforcement learning: Application to a wall following behaviour in a mobile robot. In *Methodology and tools in knowledge-based systems*, pages 300–309, Castellon (España), June 1998b. Lecture notes in artificial intelligence 1415. ISBN 3-540-64574-8.

Garud N. Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30:257–280, 2004.

Koen Hermans Jessica Vleugel, Michelle Hoogwout and Imre Gelens. Reinforcement learning with avoidance of unsafe regions. *BSc Project*, 2011.

Guofei Jiang, Cang-Pu Wu, and George Cybenko. Minimax-based reinforcement learning with state aggregation. In *Proceedings of the 37th IEEE Conference on Decision & Control*, Tampa, Florida, USA, 1998.

Kshitij Judah, Saikat Roy, Alan Fern, and Thomas G. Dietterich. Reinforcement learning via practice and critique advice. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence, Atlanta, Georgia, USA*, 2010.

Yoshinobu Kadota, Masami Kurano, and Masami Yasuda. Discounted markov decision processes with utility constraints. *Computers & Mathematics with Applications*, 51(2): 279–284, 2006.

Hisashi Kashima. Risk-sensitive learning via minimization of empirical conditional value-at-risk. *IEICE Transactions*, 90-D(12):2043–2052, 2007.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine Learning*, 49(2-3):209–232, 2002. ISSN 0885-6125.

W. Bradley Knox and Peter Stone. Interactively shaping agents via human reinforcement: the tamer framework. In *Proceedings of the 5th International Conference on Knowledge Capture*, September 2009.

W. Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *Proceedings of 9th International Conference on Autonomous Agents and Multiagent Systems*, May 2010.

W. Bradley Knox, Matthew E. Taylor, and Peter Stone. Understanding human teaching modalities in reinforcement learning environments: A preliminary report. In *Proceedings of the Agents Learning Interactively from Human Teachers Workshop*, July 2011.

Rogier Koppejan and Shimon Whiteson. Neuroevolutionary reinforcement learning for generalized helicopter control. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 145–152, July 2009.

Rogier Koppejan and Shimon Whiteson. Neuroevolutionary reinforcement learning for generalized control of simulated helicopters. *Evolutionary Intelligence*, 4:219–241, 2011.

Gregory Kuhlmann, Peter Stone, Raymond J. Mooney, and Jude W. Shavlik. Guiding a reinforcement learner with natural language advice: Initial results in robocup soccer. In *Proceedings of the AAAI-2004 Workshop on Supervisory Control of Learning and Adaptive Systems*, July 2004.

Edith L.M. Law. *Risk-directed exploration in reinforcement learning.* McGill University, 2005.

Long Ji Lin. Programming robots using reinforcement learning and teaching. In *Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14-19, 1991, Volume 2*, pages 781–786, 1991.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3–4):293–321, 1992.

Yaxin Liu, Richard Goodwin, and Sven Koenig. Risk-averse auction agents. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems*, pages 353–360. ACM, 2003. ISBN 1-58113-683-8.

David G. Luenberger. *Investment science*. Oxford University Press, Incorporated, 2013.

Richard Maclin and Jude W. Shavlik. Creating advice-taking reinforcement learners. *Machine Learning*, 22(1-3):251–281, 1996. doi: 10.1023/A:1018020625251.

Richard Maclin, Jude Shavlik, Lisa Torrey, Trevor Walker, and Edward Wild. Giving advice about preferred actions to reinforcement learners via knowledge-based kernel regression. In *Proceedings of the 20th National Conference on Artificial Intelligence*, 2005a.

Richard Maclin, Jude Shavlik, Trevor Walker, and Lisa Torrey. Knowledge-based support-vector regression for reinforcement learning. In *Proceedings of the IJCAI'05 Workshop on Reasoning, Representation, and Learning in Computer Games*, 2005b.

Frederic Maire. Apprenticeship learning for initial value functions in reinforcement learning. In *Proceedings of the IJCAI'05 Workshop on Planning and Learning in A Priori Unknown or Dynamic Domains*, pages 23–28, 2005.

Shie Mannor and John N. Tsitsiklis. Mean-variance optimization in markov decision processes. In *Proceedings of the 28th International Conference on Machine Learning, Bellevue, Washington, USA*, pages 177–184, 2011.

Harry Markowitz. Portfolio selection. In *Journal of Finance*, volume 7, pages 77–91, 1952.

José Antonio Martín H. and Javier Lope. Learning autonomous helicopter flight with evolutionary reinforcement learning. In *Proceedings of the 12th International Conference on Computer Aided Systems Theory*, pages 75–82, 2009. ISBN 978-3-642-04771-8.

Helmut Mausser and Dan Rosen. Beyond var: From measuring risk to managing risk. *ALGO Research Quarterly*, 1(2):5–20, 1998.

John Mccarthy. Programs with common sense. In *Semantic Information Processing*, pages 403–418. MIT Press, 1959.

Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine Learning*, 49(2-3):267–290, 2002. ISSN 0885-6125.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *Proceedings of NIPS Workshop on Bayesian Optimization, Experimental Design and Bandits*, 2011.

Teodor Mihai Moldovan and Pieter Abbeel. Safe exploration in markov decision processes. In *Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, UK*, 2012a.

Teodor Mihai Moldovan and Pieter Abbeel. Risk aversion in markov decision processes via near optimal chernoff bounds. In *Advances in Neural Information Processing Systems 25, Lake Tahoe, Nevada, United States.*, pages 3140–3148, 2012b.

David L. Moreno, Carlos V. Regueiro, Roberto Iglesias, and Senen Barro. Using prior knowledge to improve reinforcement learning in mobile robotics. In *Proceedings fo the Conference Towards Autonomous Robotics Systems*, Bath (Reino Unido), September 2004.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Nonparametric return distribution approximation for reinforcement learning. In *Proceedings of the 27th International Conference on Machine Learning*, pages 799–806, 2010a.

Tetsuro Morimura, Masashi Sugiyama, Hisashi Kashima, Hirotaka Hachiya, and Toshiyuki Tanaka. Parametric return density estimation for reinforcement learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 368–375, Catalina Island, California, USA, Jul. 8–11 2010b.

Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operational Research*, 53(5):780–798, September 2005. ISSN 0030-364X.

Ali Nouri. *Efficient Model-Based Exploration in Continuous State-Space Environments.* PhD thesis, New Brunswick, NJ, USA, 2011. AAI3444957.

Ali Nouri and Michael L. Littman. Multi-resolution exploration in continuous spaces. In *Advances in Neural Information Processing Systems 21*, pages 1209–1216, 2008.

Takayuki Osogami. Robustness and risk-sensitivity in markov decision processes. In *Advances in Neural Information Processing Systems 25, Lake Tahoe, Nevada, United States*, pages 233–241, 2012.

Stephen D. Patek. On terminating markov decision processes with a risk-averse objective function. *Automatica*, 37(9):1379–1386, 2001.

Frederick Philip Klahr Hayes-Roth and David J. Mostow. Advice-taking and knowledge refinement: An iterative view of skill acquisition. *Cognitive Skills and Their Acquisition*, 1981.

Sameera S. Ponda, Luke B. Johnson, and Jonathan P. How. Risk allocation strategies for distributed chance-constrained task allocation. In *American Control Conference*, June 2013.

Martin L. Putterman. Markov decision processes: Discrete stochastic dynamic programming. Jhon Wiley & Sons, Inc, 1994.

Michael T. Rosenstein and Andrew G. Barto. Supervised learning combined with an actor-critic architecture. Technical report, Amherst, MA, USA, 2002.

Michael T. Rosenstein and Andrew G. Barto. *Supervised actor-critic reinforcement learning.* Wiley-IEEE Press, 2004.

Daniil Ryabko and Marcus Hutter. *Theorical Computer Science*, (3):274–284.

Makoto Sato, Hajime Kimura, and Shigenobu Kobayashi. TD algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence*, 16:353–362, 2002.

Nils T. Siebel and Gerald Sommer. Evolutionary reinforcement learning of artificial neural networks. *International Journal of Hybrid Intelligent Systems*, 4:171–183, August 2007. ISSN 1448-5869.

William D. Smart and Leslie Pack Kaelbling. Practical reinforcement learning in continuous spaces. In *Artificial Intelligence*, pages 903–910. Morgan Kaufmann, 2000.

Alice Smith, Alice E. Smith, David W. Coit, Thomas Baeck, David Fogel, and Zbigniew Michalewicz. *Penalty functions*. Oxford University Press and Institute of Physics Publishing, 1997.

Yong Song, Yi bin Li, Cai hong Li, and Gui fang Zhang. An efficient initialization approach of q-learning for mobile robots. *International Journal of Control, Automation and Systems*, 10(1):166–172, 2012.

Alexander L. Strehl, Lihong Li, Eric Wiewiora, John Langford, and Michael L. Littman. PAC model-free reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 881–888, New York, NY, USA, 2006. ACM. ISBN 1-59593-383-2.

Halit B. Suay and Sonia Chernova. Effect of human guidance and state space size on interactive reinforcement learning. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication*, pages 1–6. IEEE, July 2011. ISBN 978-1-4577-1571-6.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. The MIT Press, March 1998. ISBN 0262193981.

Giorgio Szegö. Measures of risk. *European Journal of Operational Research*, 163(1):5–19, 2005.

Hamdy A. Taha. *Operations research: an introduction*. Number 1. Macmillan Publishing Company, 1992. ISBN 9780024189752.

Aviv Tamar, Huan Xu, and Shie Mannor. Scaling Up Robust MDPs by Reinforcement Learning. *Computing Research Repository*, abs/1306.6189, 2013.

Jie Tang, Arjun Singh, Nimbus Goehausen, and Pieter Abbeel. Parameterized maneuver learning for autonomous helicopter flight. In *International Conference on Robotics and Automation*, 2010.

Matthew E. Taylor and Peter Stone. Representation transfer for reinforcement learning. In *Fall Symposium on Computational Approaches to Representation Change during Learning and Development*, November 2007.

Matthew E. Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(1):1633–1685, 2009.

Matthew E. Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.

Andrea L. Thomaz and Cynthia Breazeal. Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance. In *Proceedings of the 21st National Conference on Artificial Intelligence*, AAAI'06, pages 1000–1005. AAAI Press, 2006. ISBN 978-1-57735-281-5.

Andrea Lockerd Thomaz and Cynthia Breazeal. Teachable robots: Understanding human teaching behavior to build more effective robot learners. *Artificial Intelligence*, 172(6-7): 716–737, 2008.

Lisa Torrey and Matthew E. Taylor. Help an agent out: Student/teacher learning in sequential decision tasks. In *Proceedings of the AAMAS Workshop Adaptive and Learning Agents*, June 2012.

Lisa Torrey, Trevor Walker, Jude Shavlik, and Richard Maclin. Using advice to transfer knowledge acquired in one reinforcement learning task to another. *Machine Learning: ECML 2005*, pages 412–424, 2005.

Paul E. Utgoff and Jeffrey A. Clouse. Two kinds of training information for evaluation function learning. In *Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, July 14-19, 1991, Volume 2*, pages 596–600, 1991.

Pablo Quintía Vidal, Roberto Iglesias Rodríguez, Miguel Rodríguez González, and Carlos Vázquez Regueiro. Learning on real robots from experience and simple user feedback. *Journal of Physical Agents*, 7(1), 2013. ISSN 1888-0258.

Pradyot Korupolu VN and Balaraman Ravindran. Beyond rewards: Learning from richer supervision. In *Proceedings of the 9th European Workshop on Reinforcement Learning*, Athens Greece, September 2011.

Thomas J. Walsh, Daniel Hewlett, and Clayton T. Morrison. Blending autonomous exploration and apprenticeship learning. In *Proceedings of the Conference Advances in Neural Information Processing Systems 24, Granada, Spain*, pages 2258–2266, 2011.

Christopher Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, Cambridge, UK, May 1989.

Kemin Zhou, John C. Doyle, and Keith Glover. *Robust and Optimal Control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1996. ISBN 0-13-456567-3.

Journal of Machine Learning Research 16 (2015) 1481-1517 Submitted 5/14; Revised 11/14; Published 8/15

Second-Order Non-Stationary Online Learning for Regression

Edward Moroshko EDWARD.MOROSHKO@GMAIL.COM
Nina Vaits NINAVAITS@GMAIL.COM
Koby Crammer KOBY@EE.TECHNION.AC.IL
Department of Electrical Engineering
The Technion - Israel Institute of Technology
Haifa 32000, Israel

Editor: Manfred Warmuth

Abstract

The goal of a learner in standard online learning, is to have the cumulative loss not much larger compared with the best-performing function from some fixed class. Numerous algorithms were shown to have this gap arbitrarily close to zero, compared with the best function that is chosen off-line. Nevertheless, many real-world applications, such as adaptive filtering, are non-stationary in nature, and the best prediction function may drift over time. We introduce two novel algorithms for online regression, designed to work well in non-stationary environment. Our first algorithm performs adaptive resets to forget the history, while the second is last-step min-max optimal in context of a drift. We analyze both algorithms in the worst-case regret framework and show that they maintain an average loss close to that of the best slowly changing sequence of linear functions, as long as the cumulative drift is sublinear. In addition, in the stationary case, when no drift occurs, our algorithms suffer logarithmic regret, as for previous algorithms. Our bounds improve over existing ones, and simulations demonstrate the usefulness of these algorithms compared with other state-of-the-art approaches.

Keywords: online learning, regret bounds, non-stationary input

1. Introduction

We consider the classical problem of online learning for regression. On each iteration, an algorithm receives a new instance (for example, input from an array of antennas) and outputs a prediction of a real value (for example distance to the source). The correct value is then revealed, and the algorithm suffers a loss based on both its prediction and the correct output value.

In the past half a century many algorithms were proposed (see e.g. a comprehensive book of Cesa-Bianchi and Lugosi 2006) for this problem, some of which are able to achieve an average loss arbitrarily close to that of the best function in retrospect. Furthermore, such guarantees hold even if the input and output pairs are chosen in a fully adversarial manner with no distributional assumptions. Many of these algorithms exploit first-order information (e.g. gradients).

Recently, there is an increased amount of interest in algorithms that exploit second-order information. For example the second-order perceptron algorithm (Cesa-Bianchi et al.,

2005), confidence-weighted learning (Dredze et al., 2008; Crammer et al., 2008), adaptive regularization of weights (AROW) (Crammer et al., 2009), all designed for classification; and AdaGrad (Duchi et al., 2010) and FTPRL (McMahan and Streeter, 2010) for general loss functions.

Despite the extensive and impressive guarantees that can be made for algorithms in such settings, competing with the best *fixed* function is not always good enough. In many real-world applications, the true target function is not *fixed*, but is *slowly* changing over time. Consider a filter designed to cancel echoes in a hall. Over time, people enter and leave the hall, furniture are being moved, microphones are replaced and so on. When this drift occurs, the predictor itself must also change in order to remain relevant.

With such properties in mind, we develop new learning algorithms, based on second-order quantities, designed to work with target drift. The goal of an algorithm is to maintain an average loss close to that of the best slowly changing sequence of functions, rather than compete well with a single function. We focus on problems for which this sequence consists only of linear functions. Most previous algorithms (e.g. Littlestone and Warmuth 1994; Auer and Warmuth 2000; Herbster and Warmuth 2001; Kivinen et al. 2001) designed for this problem are based on first-order information, such as gradient descent, with additional control on the norm of the weight-vector used for prediction (Kivinen et al., 2001) or the number of inputs used to define it (Cavallanti et al., 2007).

In Section 2 we review three second-order learning algorithms: the recursive least squares (RLS) (Hayes, 1996) algorithm, the Aggregating Algorithm for regression (AAR) (Vovk, 1997, 2001), which can be shown to be derived based on a last-step min-max approach (Forster, 1999), and the AROWR algorithm (Vaits and Crammer, 2011) which is a modification of the AROW algorithm (Crammer et al., 2009) for regression. All three algorithms obtain logarithmic regret in the stationary setting, although derived using different approaches, and they are not equivalent in general.

In Section 3 we formally present the non-stationary setting both in terms of algorithms and in terms of theoretical analysis. For the RLS algorithm, a variant called CR-RLS (Salgado et al., 1988; Goodwin et al., 83; Chen and Yen, 1999) for the non-stationary setting was described, yet not analyzed, before. In Section 4 we present two new algorithms for the non-stationary setting, that build on the other two algorithms (AROWR and AAR). Specifically, in Section 4.1 we extend the AROWR algorithm to the non-stationary setting, yielding an algorithm called ARCOR for adaptive regularization with covariance reset. Similar to CR-RLS, ARCOR performs a step called covariance-reset, which resets the second-order information from time-to-time, yet it is done based on the properties of this covariance-like matrix, and not based on the number of examples observed, as in CR-RLS.

In Section 4.2 we derive a different algorithm based on the last-step min-max approach proposed by Forster (1999) and later used (Takimoto and Warmuth, 2000) for online density estimation. On each iteration the algorithm makes the optimal min-max prediction with respect to the regret, assuming it is the last iteration. Yet, unlike previous work (Forster, 1999), it is optimal when a drift is allowed. As opposed to the derivation of the last-step min-max predictor for a fixed vector, the resulting optimization problem is not straightforward to solve. We develop a dynamic program (a recursion) to solve this problem, which allows to compute the optimal last-step min-max predictor. We call this algorithm LASER for last step adaptive regressor algorithm. We conclude the algorithmic part in Section 4.3

in which we compare all non-stationary algorithms head-to-head highlighting their similarities and differences. Additionally, after describing the details of our algorithms, we provide in Section 5 a comprehensive review of previous work, that puts our contribution in perspective. Both algorithms reduce to their stationary counterparts when no drift occurs.

We then move to Section 6 which summarizes our next contribution stating and proving regret bounds for both algorithms. We analyze both algorithms in the worst-case regret-setting and show that as long as the amount of average-drift is sublinear, the average-loss of both algorithms will converge to the average-loss of the best sequence of functions. Specifically, we show in Section 6.1 that the cumulative loss of ARCOR after observing T examples, denoted by $L_T(\text{ARCOR})$, is upper bounded by the cumulative loss of any *sequence* of weight-vectors $\{\boldsymbol{u}_t\}$, denoted by $L_T(\{\boldsymbol{u}_t\})$, plus an additional term $\mathcal{O}\left(T^{1/2}\left(V(\{\boldsymbol{u}_t\})\right)^{1/2}\log T\right)$ where $V(\{\boldsymbol{u}_t\})$ measures the differences (or variance) between consecutive weight-vectors of the sequence $\{\boldsymbol{u}_t\}$. Later, we show in Section 6.2 a similar bound for the loss of LASER, denoted by $L_T(\text{LASER})$, for which the second term is $\mathcal{O}\left(T^{2/3}\left(V(\{\boldsymbol{u}_t\})\right)^{1/3}\right)$. We emphasize that in both bounds the measure $V(\{\boldsymbol{u}_t\})$ of differences between consecutive weight-vectors is not defined in the same way, and thus, the bounds are not comparable in general.

In Section 7 we report results of simulations designed to highlight the properties of both algorithms, as well as the commonalities and differences between them. We conclude in Section 8 and most of the technical proofs appear in the appendix.

The ARCOR algorithm was presented in a shorter publication (Vaits and Crammer, 2011), as well with its analysis and some of its details. The LASER algorithm and its analysis was also presented in a shorter version (Moroshko and Crammer, 2013). The contribution of this submission is three-fold. First, we provide head-to-head comparison of three second-order algorithms for the stationary case. Second, we fill the gap of second-order algorithms for the non-stationary case. Specifically, we add to the CR-RLS (which extends RLS) and design second-order algorithms for the non-stationary case and analyze them, building both on AROWR and AAR. Our algorithms are derived from different principles, which is reflected in our analysis. Finally, we provide empirical evidence showing that under various conditions different algorithm performs the best.

Some notation we use throughout the paper: For a symmetric matrix Σ we denote its $\mathrm{j}th$ eigenvalue by $\lambda_j(\Sigma)$. Similarly we denote its smallest eigenvalue by $\lambda_{min}(\Sigma) = \min_j \lambda_j(\Sigma)$, and its largest eigenvalue by $\lambda_{max}(\Sigma) = \max_j \lambda_j(\Sigma)$. For a vector $\boldsymbol{u} \in \mathbb{R}^d$, we denote by $\|\boldsymbol{u}\|$ the ℓ_2-norm of the vector. Finally, for $y > 0$ we define $clip(x, y) = \text{sign}(x)\min\{|x|, y\}$.

2. Stationary Online Learning

We focus on the online regression task evaluated with the squared loss, where algorithms work in iterations (or rounds). On each round an online algorithm receives an input-vector $\boldsymbol{x}_t \in \mathbb{R}^d$ and predicts a real value $\hat{y}_t \in \mathbb{R}$. Then the algorithm receives a target label $y_t \in \mathbb{R}$ associated with $\boldsymbol{x}_t$, uses it to update its prediction rule, and proceeds to the next round.

On each iteration, the performance of the algorithm is evaluated using the squared loss, $\ell_t(\text{alg}) = \ell\left(y_t, \hat{y}_t\right) = \left(\hat{y}_t - y_t\right)^2$. The cumulative loss suffered by the algorithm over T iterations is, $L_T(\text{alg}) = \sum_{t=1}^{T} \ell_t(\text{alg})$.

The goal of the algorithm is to have low cumulative loss compared to predictors from some class. A large body of work, which we adopt as well, is focused on linear prediction functions of the form $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{u}$ where $\boldsymbol{u} \in \mathbb{R}^d$ is some weight-vector. We denote by $\ell_t(\boldsymbol{u}) = \left(\boldsymbol{x}_t^\top \boldsymbol{u} - y_t\right)^2$ the instantaneous loss of a weight-vector $\boldsymbol{u}$. The cumulative loss suffered by a fixed weight-vector $\boldsymbol{u}$ is, $L_T(\boldsymbol{u}) = \sum_{t=1}^T \ell_t(\boldsymbol{u})$.

The goal of the learning algorithm is to suffer low loss compared with the best linear function. Formally we define the regret of an algorithm to be

$$R(T) = L_T(\text{alg}) - \inf_{\boldsymbol{u}} L_T(\boldsymbol{u}) \ .$$

The goal of an algorithm is to have $R(T) = o(T)$, such that the average loss of the algorithm will converge to the average loss of the best linear function $\boldsymbol{u}$.

Numerous algorithms were developed for this problem, see a comprehensive review in the book of Cesa-Bianchi and Lugosi (2006). Among these, a few second-order online algorithms for regression were proposed in recent years, which we summarize in Table 1. One approach for online learning is to reduce the problem into consecutive batch problems, and specifically use all previous examples to generate a regressor, which is used to predict the current example. The least squares approach, for example, sets a weight-vector to be the solution of the following optimization problem

$$\boldsymbol{w}_t = \arg\min_{\boldsymbol{w}} \left(\sum_{i=1}^t r^{t-i} \left(y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i\right)^2 \right) \ ,$$

for $0 < r \leq 1$. Since the last problem grows with time, the well known recursive least squares (RLS) (Hayes, 1996) algorithm was developed to generate a solution recursively. The RLS algorithm maintains both a vector $\boldsymbol{w}_t$ and a positive semi-definite (PSD) matrix Σ_t. On each iteration, after making a prediction $\hat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}$, the algorithm receives the true label y_t and updates

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})\Sigma_{t-1}\boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t} \tag{1}$$

$$\Sigma_t^{-1} = r\Sigma_{t-1}^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top \ . \tag{2}$$

The update of the prediction vector $\boldsymbol{w}_t$ is additive, with vector $\Sigma_{t-1}\boldsymbol{x}_t$ scaled by the error $(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})$ over the norm of the input measured using the norm defined by the matrix $\boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t$. The algorithm is summarized in the right column of Table 1.

The Aggregating Algorithm for regression (AAR) (Vovk, 1997; Azoury and Warmuth, 2001), summarized in the middle column of Table 1, was introduced by Vovk and it is similar to the RLS algorithm, except it shrinks its predictions. The AAR algorithm was shown to be last-step min-max optimal by Forster (1999). Given a new input $\boldsymbol{x}_T$ the algorithm predicts $\hat{y}_T$ which is the minimizer of the following problem

$$\arg\min_{\hat{y}_T} \max_{y_T} \left[\sum_{t=1}^T (y_t - \hat{y}_t)^2 - \inf_{\boldsymbol{u}} \left(b \left\|\boldsymbol{u}\right\|^2 + L_T(\boldsymbol{u}) \right) \right] . \tag{3}$$

Forster proposed also a simpler analysis with the same regret bound as of Vovk (1997).

The AROWR algorithm (Vaits and Crammer, 2011) is a modification of the AROW algorithm (Crammer et al., 2009) for regression. In a nutshell, the AROW algorithm maintains a Gaussian distribution parameterized by a mean $\boldsymbol{w}_t \in \mathbb{R}^d$ and a full covariance matrix $\Sigma_t \in \mathbb{R}^{d\times d}$. Intuitively, the mean $\boldsymbol{w}_t$ represents a current linear function, while the covariance matrix Σ_t captures the uncertainty in the linear function $\boldsymbol{w}_t$. Given a new input $\boldsymbol{x}_t$ the algorithm uses its current mean to make a prediction $\hat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}$. Then, given the true label y_t, AROWR sets the new distribution to be the solution of the following optimization problem

$$\arg\min_{\boldsymbol{w},\Sigma} \left[\mathrm{D}_{\mathrm{KL}}[\mathcal{N}(\boldsymbol{w},\Sigma) \,\|\, \mathcal{N}(\boldsymbol{w}_{t-1},\Sigma_{t-1})] + \frac{1}{2r}\left(y_t - \boldsymbol{w}^\top \boldsymbol{x}_t\right)^2 + \frac{1}{2r}\left(\boldsymbol{x}_t^\top \Sigma \boldsymbol{x}_t\right) \right] , \qquad (4)$$

for $r > 0$. This optimization problem is similar to the one of AROW (Crammer et al., 2009) for classification, except we use the square loss rather than squared-hinge loss used in AROW. Intuitively, the optimization problem trades off between three requirements. The first term forces the parameters not to change much per example, as the entire learning history is encapsulated within them. The second term requires that the new vector $\boldsymbol{w}_t$ should perform well on the current instance, and finally, the third term reflects the fact that the uncertainty about the parameters reduces as we observe the current example $\boldsymbol{x}_t$.

The weight vector solving this optimization problem (details given by Vaits and Crammer 2011) is given by

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \left(\frac{y_t - \boldsymbol{w}_{t-1}\cdot \boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1}\boldsymbol{x}_t}\right)\Sigma_{t-1}\boldsymbol{x}_t \ , \qquad (5)$$

and the optimal covariance matrix is

$$\Sigma_t^{-1} = \Sigma_{t-1}^{-1} + \frac{1}{r}\boldsymbol{x}_t \boldsymbol{x}_t^\top \ . \qquad (6)$$

The algorithm is summarized in the left column of Table 1. Comparing AROWR to RLS we observe that while the update of the weights of (5) is equivalent to the update of RLS in (1), the update of the matrix (2) for RLS is not equivalent to (6), as in the former case the matrix goes via a multiplicative update as well as additive, while in (6) the update is only additive. The two updates are equivalent only by setting $r = 1$. Moving to AAR, we note that the update rules for $\boldsymbol{w}_t$ and Σ_t in AROWR and AAR are the same if we define $\Sigma_t^{AAR} = \Sigma_t^{AROWR}/r$, but AROWR does not shrink its predictions as AAR. Thus all three algorithms are not equivalent, although very similar.

3. Non-Stationary Online Learning

The analysis of all algorithms discussed above compares their performance to that of a single fixed weight vector $\boldsymbol{u}$, and all suffer regret that is logarithmic in T. However, in many real-world applications, the true target function is not fixed, but is slowly changing over time.

We use an extended notion of evaluation, comparing our algorithms to a sequence of functions. We define the loss suffered by such a sequence to be

$$L_T(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_T) = L_T(\{\boldsymbol{u}_t\}) = \sum_{t=1}^{T} \ell_t(\boldsymbol{u}_t) \ ,$$

and the tracking regret is then defined to be

$$R(T) = L_T(\text{alg}) - \inf_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_T} L_T(\{\boldsymbol{u}_t\}) \ .$$

We focus on algorithms that are able to compete against sequences of weight-vectors, $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T) \in \mathbb{R}^d \times \cdots \times \mathbb{R}^d$, where $\boldsymbol{u}_t$ is used to make a prediction for the tth example $(\boldsymbol{x}_t, y_t)$. Note the difference between tracking regret (where the algorithm is compared to a good sequence of experts, as we do) and adaptive regret (Adamskiy et al., 2012), which measures how well the algorithm approximates the best expert locally on some time interval.

Clearly, with no restriction over the set $\{\boldsymbol{u}_t\}$ the right term of the regret can easily be zero by setting, $\boldsymbol{u}_t = \boldsymbol{x}_t(y_t / \|\boldsymbol{x}_t\|^2)$, which implies $\ell_t(\boldsymbol{u}_t) = 0$ for all t. Thus, in the analysis below we will make use of the total drift of the weight-vectors defined to be

$$V^{(P)} = V_T^{(P)}(\{\boldsymbol{u}_t\}) = \sum_{t=1}^{T-1} \|\boldsymbol{u}_t - \boldsymbol{u}_{t+1}\|^P \ ,$$

where $P \in \{1, 2\}$, and the total loss of the algorithm is allowed to scale with the total drift.

For the three algorithms in Table 1 the matrix Σ can be interpreted as adaptive learning rate, as was also observed previously in the context of CW (Dredze et al., 2008), AROW (Crammer et al., 2009), AdaGrad (Duchi et al., 2010) and FTPRL (McMahan and Streeter, 2010). As these algorithms process more examples, that is larger values of t, the eigenvalues of the matrix Σ_t^{-1} increase, the eigenvalues of the matrix Σ_t decrease, and we get that the rate of updates is getting smaller, since the additive term $\Sigma_{t-1}\boldsymbol{x}_t$ is getting smaller. As a consequence the algorithms will gradually stop updating using current instances which lie in the subspace of examples that were previously observed numerous times. This property leads to a very fast convergence in the stationary case. However, when we allow these algorithms to be compared with a sequence of weight-vectors, each applied to a different input example, or equivalently, there is a drift or shift of a good prediction vector, these algorithms will perform poorly, as they will converge and will not be able to adapt to the non-stationarity nature of the data.

This phenomena motivated the proposal of the CR-RLS algorithm (Salgado et al., 1988; Goodwin et al., 83; Chen and Yen, 1999), which re-sets the covariance matrix every fixed number of input examples, causing the algorithm not to converge or get stuck. The pseudo-code of CR-RLS algorithm is given in the right column of Table 2. The only difference of CR-RLS from RLS is that after updating the matrix Σ_t, the algorithm checks whether T_0 (a predefined natural number) examples were observed since the last restart, and if this is the case, it sets the matrix to be the identity matrix. Clearly, if $T_0 = \infty$ the CR-RLS algorithm is reduced to the RLS algorithm.

4. Algorithms for Non-Stationary Regression

In this work we fill the gap and propose extension to non-stationary setting for the two other algorithms in Table 1. Similar to CR-RLS, both algorithms modify the matrix Σ_t to prevent its eigenvalues to shrink to zero. The first algorithm, described in Section 4.1,

		AROWR	**AAR**	**RLS**
Parameters		$0 < r$	$0 < b$	$0 < r \leq 1$
Initialize		$\boldsymbol{w}_0 = 0\ ,\ \Sigma_0 = I$	$\boldsymbol{w}_0 = 0\ ,\ \Sigma_0 = b^{-1}I$	$\boldsymbol{w}_0 = 0\ ,\ \Sigma_0 = I$
for $t = 1...T$		Receive an instance $\boldsymbol{x}_t$		
	Output prediction	$\hat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}$	$\hat{y}_t = \frac{\boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}}{1 + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t}$	$\hat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}$
		Receive a correct label y_t		
	Update Σ_t	$\Sigma_t^{-1} = \Sigma_{t-1}^{-1} + \frac{1}{r}\boldsymbol{x}_t \boldsymbol{x}_t^\top$	$\Sigma_t^{-1} = \Sigma_{t-1}^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top$	$\Sigma_t^{-1} = r\Sigma_{t-1}^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top$
	Update $\boldsymbol{w}_t$	$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})\Sigma_{t-1}\boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t}$	$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})\Sigma_{t-1}\boldsymbol{x}_t}{1 + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t}$	$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})\Sigma_{t-1}\boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t}$
Output		$\boldsymbol{w}_T\ ,\ \Sigma_T$	$\boldsymbol{w}_T\ ,\ \Sigma_T$	$\boldsymbol{w}_T\ ,\ \Sigma_T$
Extension to non-stationary setting		**ARCOR** Section 4.1 below	**LASER** Section 4.2 below	**CR-RLS** (Goodwin et al., 83)
Analysis		yes, Section 6.1 below	yes, Section 6.2 below	No

Table 1: Algorithms for stationary setting and their extension to non-stationary case

extends AROWR to the non-stationary setting and is similar in spirit to CR-RLS, yet the restart operations it performs depend on the spectral properties of the covariance matrix, rather than the time index t. Additionally, this algorithm performs a projection of the weight vector into a predefined ball. Similar technique was used in first order algorithms by Herbster and Warmuth (2001), and Kivinen and Warmuth (1997). Both steps are motivated from the design and analysis of AROWR. Its design is composed of solving small optimization problems defined in (4), one per input example. The non-stationary version performs explicit corrections to its update, in order to prevent from the covariance matrix to shrink to zero, and the weight-vector to grow too fast.

The second algorithm, described in Section 4.2, is based on a last-step min-max prediction principle and objective, where we replace $L_T(\boldsymbol{u})$ in (3) with $L_T(\{\boldsymbol{u}_t\})$ and some additional modifications preventing the solution being degenerate. Here the algorithmic modifications from the original AAR algorithm are implicit and are due to the modifications of the objective. The resulting algorithm smoothly interpolates the covariance matrix with the identity matrix.

4.1 ARCOR: Adaptive Regularization of Weights for Regression with Covariance Reset

Our first algorithm is based on the AROWR. We propose two modifications to (5) and (6), which in combination overcome the problem that the algorithm's learning rate gradually goes to zero. The modified algorithm operates on segments of the input sequence. In each segment indexed by i, the algorithm checks whether the lowest eigenvalue of Σ_t is greater than a given lower bound Λ_i. Once the lowest eigenvalue of Σ_t is smaller than Λ_i the algorithm resets $\Sigma_t = I$ and updates the value of the lower bound Λ_{i+1}. Formally, the

algorithm uses the update (6) to compute an intermediate candidate for Σ_t, denoted by

$$\tilde{\Sigma}_t = \left(\Sigma_{t-1}^{-1} + \frac{1}{r}\boldsymbol{x}_t\boldsymbol{x}_t^\top\right)^{-1} . \tag{7}$$

If indeed $\tilde{\Sigma}_t \succeq \Lambda_i I$ then it sets $\Sigma_t = \tilde{\Sigma}_t$, otherwise it sets $\Sigma_t = I$ and the segment index is increased by 1.

Additionally, before our modification, the norm of the weight vector $\boldsymbol{w}_t$ did not increase much as the effective learning rate (the matrix Σ_t) went to zero. After our update, as the learning rate is effectively bounded from below, the norm of $\boldsymbol{w}_t$ may increase too fast, which in turn will cause a low update-rate in non-stationary inputs.

We thus employ additional modification which is exploited by the analysis. After updating the mean $\boldsymbol{w}_t$ as in (5),

$$\tilde{\boldsymbol{w}}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})\Sigma_{t-1}\boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1}\boldsymbol{x}_t} , \tag{8}$$

we project it into a ball B around the origin of radius R_B using a Mahalanobis distance. Formally, we define the function $\text{proj}(\tilde{\boldsymbol{w}}, \Sigma, R_B)$ to be the solution of the following optimization problem

$$\arg \min_{\|\boldsymbol{w}\| \le R_B} \frac{1}{2} (\boldsymbol{w} - \tilde{\boldsymbol{w}})^\top \Sigma^{-1} (\boldsymbol{w} - \tilde{\boldsymbol{w}}) .$$

We write the Lagrangian,

$$\mathcal{L} = \frac{1}{2} (\boldsymbol{w} - \tilde{\boldsymbol{w}})^\top \Sigma^{-1} (\boldsymbol{w} - \tilde{\boldsymbol{w}}) + \alpha \left(\frac{1}{2}\|\boldsymbol{w}\|^2 - \frac{1}{2}R_B^2\right) .$$

Setting the gradient with respect to $\boldsymbol{w}$ to zero we get, $\Sigma^{-1} (\boldsymbol{w} - \tilde{\boldsymbol{w}}) + \alpha\boldsymbol{w} = 0$. Solving for $\boldsymbol{w}$ we get

$$\boldsymbol{w} = \left(\alpha I + \Sigma^{-1}\right)^{-1} \Sigma^{-1}\tilde{\boldsymbol{w}} = (I + \alpha\Sigma)^{-1} \tilde{\boldsymbol{w}} .$$

From KKT conditions we get that if $\|\tilde{\boldsymbol{w}}\| \le R_B$ then $\alpha = 0$ and $\boldsymbol{w} = \tilde{\boldsymbol{w}}$. Otherwise, α is the unique positive scalar that satisfies $\| (I + \alpha\Sigma)^{-1} \tilde{\boldsymbol{w}}\| = R_B$. The value of α can be found using binary search and eigen-decomposition of the matrix Σ. We write explicitly $\Sigma = V\Lambda V^\top$ for a diagonal matrix Λ. By denoting $\boldsymbol{u} = V^\top \tilde{\boldsymbol{w}}$ we rewrite the last equation, $\| (I + \alpha\Lambda)^{-1} \boldsymbol{u}\| = R_B$. We thus wish to find α such that $\sum_j^d \frac{u_j^2}{(1+\alpha\Lambda_{j,j})^2} = R_B^2$. It can be done using a binary search for $\alpha \in [0, a]$ where $a = (\|\boldsymbol{u}\|/R_B - 1)/\lambda_{\min}(\Lambda)$. To summarize, the projection step can be performed in time cubic in d and logarithmic in R_B and Λ_i.

We call the algorithm ARCOR for adaptive regularization with covariance reset. A pseudo-code of the algorithm is summarized in the left column of Table 2. We defer a comparison of ARCOR and CR-RLS after the presentation of our second algorithm now.

		ARCOR	**LASER**	**CR-RLS**
Parameters		$0 < r, R_B$, a sequence $1 > \Lambda_1 \geq \Lambda_2 ...$	$0 < b < c$	$0 < r \leq 1, T_0 \in \mathbb{N}$
Initialize		$\boldsymbol{w}_0 = 0$, $\Sigma_0 = I$, $i = 1$	$\boldsymbol{w}_0 = 0$, $\Sigma_0 = \frac{c-b}{bc} I$	$\boldsymbol{w}_0 = 0$, $\Sigma_0 = I$
for $t = 1...T$		Receive an instance $\boldsymbol{x}_t$		
	Output prediction	$\hat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}$	$\hat{y}_t = \frac{\boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}}{1 + \boldsymbol{x}_t^\top \left(\Sigma_{t-1} + c^{-1} I\right) \boldsymbol{x}_t}$	$\hat{y}_t = \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}$
		Receive a correct label y_t		
	Update Σ_t	$\tilde{\Sigma}_t^{-1} = \Sigma_{t-1}^{-1} + \frac{1}{r} \boldsymbol{x}_t \boldsymbol{x}_t^\top$ If $\tilde{\Sigma}_t \succeq \Lambda_i I$ set $\Sigma_t = \tilde{\Sigma}_t$ else set $\Sigma_t = I$, $i = i + 1$	$\Sigma_t^{-1} = \left(\Sigma_{t-1} + c^{-1} I\right)^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top$	$\tilde{\Sigma}_t^{-1} = r\Sigma_{t-1}^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top$ If $\mod(t, T_0) > 0$ set $\Sigma_t = \tilde{\Sigma}_t$ else set $\Sigma_t = I$
	Update $\boldsymbol{w}_t$	$\tilde{\boldsymbol{w}}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}) \Sigma_{t-1} \boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t}$ $\boldsymbol{w}_t = \text{proj}\left(\tilde{\boldsymbol{w}}_t, \Sigma_t, R_B\right)$	$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}) \left(\Sigma_{t-1} + c^{-1} I\right) \boldsymbol{x}_t}{1 + \boldsymbol{x}_t^\top \left(\Sigma_{t-1} + c^{-1} I\right) \boldsymbol{x}_t}$	$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} + \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}) \Sigma_{t-1} \boldsymbol{x}_t}{r + \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t}$
Output		$\boldsymbol{w}_T$, Σ_T	$\boldsymbol{w}_T$, Σ_T	$\boldsymbol{w}_T$, Σ_T

Table 2: ARCOR, LASER and CR-RLS algorithms

4.2 Last-Step Min-Max Algorithm for Non-Stationary Setting

Our second algorithm is based on a last-step min-max predictor proposed by Forster (1999) and later modified by Moroshko and Crammer (2012) to obtain sub-logarithmic regret in the stationary case. On each round, the algorithm predicts as in the last round, and assumes a worst case choice of y_t given the algorithm's prediction.

We extend the rule given in (3) to the non-stationary setting, and re-define the last-step minmax predictor $\hat{y}_T$ to be[1]

$$\arg\min_{\hat{y}_T} \max_{y_T} \left[\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 - \min_{\boldsymbol{u}_1, .., \boldsymbol{u}_T} Q_T\left(\boldsymbol{u}_1, ..., \boldsymbol{u}_T\right) \right], \tag{9}$$

where

$$Q_t\left(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t\right) = b \left\|\boldsymbol{u}_1\right\|^2 + c \sum_{s=1}^{t-1} \left\|\boldsymbol{u}_{s+1} - \boldsymbol{u}_s\right\|^2 + \sum_{s=1}^{t} \left(y_s - \boldsymbol{u}_s^\top \boldsymbol{x}_s\right)^2 \ , \tag{10}$$

for some positive constants b, c. The first term of (9) is the loss suffered by the algorithm while $Q_t\left(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t\right)$ defined in (10) is a sum of the loss suffered by some sequence of linear functions $(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t)$, and a penalty for consecutive pairs that are far from each other, and for the norm of the first to be far from zero.

1. y_T and $\hat{y}_T$ serve both as quantifiers (over the max and min operators, respectively), and as the optimal arguments of this optimization problem.

We develop the algorithm by solving the three optimization problems in (9), first, minimizing the inner term, $\min_{\boldsymbol{u}_1,...,\boldsymbol{u}_T} Q_T(\boldsymbol{u}_1,...,\boldsymbol{u}_T)$, maximizing over $\boldsymbol{y}_T$, and finally, minimizing over $\hat{y}_T$. We start with the inner term for which we define an auxiliary function

$$P_t(\boldsymbol{u}_t) = \min_{\boldsymbol{u}_1,...,\boldsymbol{u}_{t-1}} Q_t(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_t) \ ,$$

which clearly satisfies

$$\min_{\boldsymbol{u}_1,...,\boldsymbol{u}_t} Q_t(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_t) = \min_{\boldsymbol{u}_t} P_t(\boldsymbol{u}_t) \ .$$

The following lemma states a recursive form of the function-sequence $P_t(\boldsymbol{u}_t)$.

Lemma 1 *For $t = 2, 3, \ldots$*

$$\begin{aligned} P_1(\boldsymbol{u}_1) &= Q_1(\boldsymbol{u}_1) \\ P_t(\boldsymbol{u}_t) &= \min_{\boldsymbol{u}_{t-1}} \left(P_{t-1}(\boldsymbol{u}_{t-1}) + c\left\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\right\|^2 + \left(y_t - \boldsymbol{u}_t^\top \boldsymbol{x}_t\right)^2 \right) . \end{aligned}$$

The proof appears in Appendix A. Using Lemma 1 we write explicitly the function $P_t(\boldsymbol{u}_t)$.

Lemma 2 *The following equality holds*

$$P_t(\boldsymbol{u}_t) = \boldsymbol{u}_t^\top D_t \boldsymbol{u}_t - 2\boldsymbol{u}_t^\top \boldsymbol{e}_t + f_t \ ,$$

where

$$D_1 = bI + \boldsymbol{x}_1\boldsymbol{x}_1^\top \qquad D_t = \left(D_{t-1}^{-1} + c^{-1}I\right)^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top \tag{11}$$

$$\boldsymbol{e}_1 = y_1\boldsymbol{x}_1 \qquad \boldsymbol{e}_t = \left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t\boldsymbol{x}_t \tag{12}$$

$$f_1 = y_1^2 \qquad f_t = f_{t-1} - \boldsymbol{e}_{t-1}^\top \left(cI + D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t^2 \ . \tag{13}$$

Note that $D_t \in \mathbb{R}^{d\times d}$ is a positive definite matrix, $\boldsymbol{e}_t \in \mathbb{R}^{d\times 1}$ and $f_t \in \mathbb{R}$. The proof appears in Appendix B. From Lemma 2 we conclude that

$$\min_{\boldsymbol{u}_1,...,\boldsymbol{u}_t} Q_t(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_t) = \min_{\boldsymbol{u}_t} P_t(\boldsymbol{u}_t) = \min_{\boldsymbol{u}_t} \left(\boldsymbol{u}_t^\top D_t \boldsymbol{u}_t - 2\boldsymbol{u}_t^\top \boldsymbol{e}_t + f_t\right) = -\boldsymbol{e}_t^\top D_t^{-1}\boldsymbol{e}_t + f_t \ . \tag{14}$$

Substituting (14) back in (9) we get that the last-step minmax predictor is given by

$$\hat{y}_T = \arg\min_{\hat{y}_T} \max_{y_T} \left[\sum_{t=1}^{T} (y_t - \hat{y}_t)^2 + \boldsymbol{e}_T^\top D_T^{-1}\boldsymbol{e}_T - f_T \right] \ . \tag{15}$$

Since $\boldsymbol{e}_T$ depends on y_T we substitute (12) in the second term of (15),

$$\boldsymbol{e}_T^\top D_T^{-1}\boldsymbol{e}_T = \left(\left(I + c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1} + y_T\boldsymbol{x}_T\right)^\top D_T^{-1}\left(\left(I + c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1} + y_T\boldsymbol{x}_T\right) \ . \tag{16}$$

Substituting (16) and (13) in (15) and omitting terms not depending explicitly on y_T and $\hat{y}_T$ we get

$$\begin{aligned}\hat{y}_T &= \arg\min_{\hat{y}_T}\max_{y_T}\left[(y_T-\hat{y}_T)^2 + y_T^2\boldsymbol{x}_T^\top D_T^{-1}\boldsymbol{x}_T + 2y_T\boldsymbol{x}_T^\top D_T^{-1}\left(I+c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1} - y_T^2\right] \\ &= \arg\min_{\hat{y}_T}\max_{y_T}\left[\left(\boldsymbol{x}_T^\top D_T^{-1}\boldsymbol{x}_T\right)y_T^2 + 2y_T\left(\boldsymbol{x}_T^\top D_T^{-1}\left(I+c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1} - \hat{y}_T\right) + \hat{y}_T^2\right] .\end{aligned} \tag{17}$$

The last equation is strictly convex in y_T and thus the optimal solution is not bounded. To solve it, we follow an approach used by Forster (1999) in a different context. In order to make the optimal value bounded, we assume that the adversary can only choose labels from a bounded set $y_T \in [-Y, Y]$. Thus, the optimal solution of (17) over y_T is given by the following equation, since the optimal value is $y_T \in \{+Y, -Y\}$,

$$\hat{y}_T = \arg\min_{\hat{y}_T}\left[\left(\boldsymbol{x}_T^\top D_T^{-1}\boldsymbol{x}_T\right)Y^2 + 2Y\left|\boldsymbol{x}_T^\top D_T^{-1}\left(I+c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1} - \hat{y}_T\right| + \hat{y}_T^2\right] .$$

This problem is of a similar form to the one discussed by Forster (1999), from which we get the optimal solution, $\hat{y}_T = clip\left(\boldsymbol{x}_T^\top D_T^{-1}\left(I+c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1}, Y\right)$.

The optimal solution depends explicitly on the bound Y, and as its value is not known, we thus ignore it, and define the output of the algorithm to be

$$\hat{y}_T = \boldsymbol{x}_T^\top D_T^{-1}\left(I+c^{-1}D_{T-1}\right)^{-1}\boldsymbol{e}_{T-1} = \boldsymbol{x}_T^\top D_T^{-1}D'_{T-1}\boldsymbol{e}_{T-1} , \tag{18}$$

where we define

$$D'_{t-1} = \left(I+c^{-1}D_{t-1}\right)^{-1} . \tag{19}$$

We call the algorithm LASER for last step adaptive regressor algorithm. Clearly, for $c=\infty$ the LASER algorithm reduces to the AAR algorithm. Similar to CR-RLS and ARCOR, this algorithm can be also expressed in terms of weight-vector $\boldsymbol{w}_t$ and a PSD matrix Σ_t, by denoting $\boldsymbol{w}_t = D_t^{-1}\boldsymbol{e}_t$ and $\Sigma_t = D_t^{-1}$. The algorithm is summarized in the middle column of Table 2.

4.3 Discussion

Table 2 enables us to compare the three algorithms head-to-head. All algorithms perform predictions, and then update the prediction vector $\boldsymbol{w}_t$ and the matrix Σ_t. CR-RLS and ARCOR are more similar to each other, both stem from a stationary algorithm, and perform resets from time-to-time. For CR-RLS it is performed every fixed time steps, while for ARCOR it is performed when the eigenvalues of the matrix (or effective learning rate) are too small. ARCOR also performs a projection step, which is motivated to ensure that the weight-vector will not grow to much, and is used explicitly in the analysis below. Note that CR-RLS (as well as RLS) also uses a forgetting factor (if $r<1$).

Our second algorithm, LASER, controls the covariance matrix in a smoother way. On each iteration it interpolates it with the identity matrix before adding $\boldsymbol{x}_t\boldsymbol{x}_t^\top$. Note that if λ is an eigenvalue of Σ_{t-1}^{-1} then $\lambda\times(c/(\lambda+c)) < \lambda$ is an eigenvalue of $\left(\Sigma_{t-1}+c^{-1}I\right)^{-1}$. Thus

the algorithm implicitly reduces the eigenvalues of the inverse covariance (and increases the eigenvalues of the covariance).

Finally, all three algorithms can be combined with Mercer kernels as they employ only sums of inner- and outer-products of its inputs. This allows them to perform non-linear predictions, similar to SVM.

5. Related Work

There is a large body of research in online learning for regression problems. Almost half a century ago, Widrow and Hoff (1960) developed a variant of the least mean squares (LMS) algorithm for adaptive filtering and noise reduction. The algorithm was further developed and analyzed extensively, for example by Feuer and Weinstein (1985). The normalized least mean squares filter (NLMS) (Bershad, 1986; Bitmead and Anderson, 1980) is similar to LMS but it is insensitive to scaling of the input. The recursive least squares (RLS) (Hayes, 1996) is the closest to our algorithms in the signal processing literature and also maintains a weight-vector and a covariance-like matrix, which is positive semi-definite (PSD), that is used to re-weight inputs.

In the machine learning literature the problem of online regression was studied extensively, and clearly we cannot cover all the relevant work. Cesa-Bianchi et al. (1993) studied gradient descent based algorithms for regression with the squared loss. Kivinen and Warmuth (1997) proposed various generalizations for general regularization functions. We refer the reader to a comprehensive book in the subject (Cesa-Bianchi and Lugosi, 2006).

Foster (1991) studied an online version of the ridge regression algorithm in the worst-case setting. Vovk (1990) proposed a related algorithm called the Aggregating Algorithm (AA), which was later applied to the problem of linear regression with square loss (Vovk, 1997, 2001). Forster (1999) simplified the regret analysis for this problem. Both algorithms employ second-order information. ARCOR for the separable case is very similar to these algorithms, although has alternative derivation. Recently, few algorithms were proposed either for classification (Cesa-Bianchi et al., 2005; Dredze et al., 2008; Crammer et al., 2008, 2009) or for general loss functions (Duchi et al., 2010; McMahan and Streeter, 2010) in the online convex programming framework. AROWR (Vaits and Crammer, 2011) shares the same design principles of AROW (Crammer et al., 2009) yet it is aimed for regression. The ARCOR algorithm takes AROWR one step further and it has two important modifications which makes it work in the drifting or shifting settings. These modifications make the analysis more complex than of AROW.

Two of the approaches used in previous algorithms for non-stationary setting are bounding the weight vector and covariance reset. Bounding the weight vector was performed either by projecting it into a bounded set (Herbster and Warmuth, 2001), shrinking it by multiplication (Kivinen et al., 2001), or subtraction of previously seen examples (Cavallanti et al., 2007). These three methods (or at least most of their variants) can be combined with kernel operators, and in fact, the last two approaches were designed and motivated by kernels.

The Covariance Reset RLS algorithm (CR-RLS) (Salgado et al., 1988; Goodwin et al., 83; Chen and Yen, 1999) was designed for adaptive filtering. CR-RLS makes covariance reset every fixed amount of data points, while ARCOR performs restarts based on the actual properties of the data - the eigenspectrum of the covariance matrix. Furthermore,

as far as we know, there is no analysis in the mistake bound model for CR-RLS. Both ARCOR and CR-RLS are motivated from the property that the covariance matrix goes to zero and becomes rank deficient. In both algorithms the information encapsulated in the covariance matrix is lost after restarts. In a rapidly varying environments, like a wireless channel, this loss of memory can be beneficial, as previous contributions to the covariance matrix may have little correlation with the current structure. Recent versions of CR-RLS (Goodhart et al., 1991; Song et al., 2002) employ covariance reset to have numerically stable computations.

ARCOR algorithm combines two techniques with second-order algorithm for regression. In this aspect it has the best of all worlds, fast convergence rate due to the usage of second-order information, and the ability to adapt in non-stationary environments due to projection and resets.

LASER is simpler than all these algorithms as it controls the increase of the eigenvalues of the covariance matrix, implicitly rather than explicitly, by "averaging" it with a fixed diagonal matrix (see 11), and it does not involve projection steps. The Kalman filter (Kalman, 1960) and the H_∞ algorithm (e.g. the work of Simon 2006) designed for filtering take a similar approach, yet the exact algebraic form is different.

The derivation of the LASER algorithm in this work shares similarities with the work of Forster (1999) and the work of Moroshko and Crammer (2012). These algorithms are motivated from the last-step min-max predictor. Yet, the algorithms of Forster and Moroshko and Crammer are designed for the stationary setting, while LASER is primarily designed for the non-stationary setting. Moroshko and Crammer (2012) also discussed a weak variant of the non-stationary setting, where the complexity is measured by the total distance from a reference vector $\bar{\mathbf{u}}$, rather than the total distance of consecutive vectors (as in this paper), which is more relevant to non-stationary problems.

6. Regret Bounds

We now analyze our algorithms in the non-stationary case, upper bounding the regret using more than a single comparison vector. Specifically, our goal is to prove bounds that would hold uniformly for all inputs, and are of the form

$$L_T(\text{alg}) \leq L_T(\{\boldsymbol{u}_t\}) + \alpha(T) \left(V^{(P)}\right)^{\gamma} \ ,$$

for either $P = 1$ or $P = 2$, a constant γ and a function $\alpha(T)$ that may depend implicitly on other quantities of the problem.

Specifically, in the next section we show (Corollary 6) that under a particular choice of $\Lambda_i = \Lambda_i(V^{(1)})$ for the ARCOR algorithm, its regret is bounded by

$$L_T(\text{ARCOR}) \leq L_T(\{\boldsymbol{u}_t\}) + \mathcal{O}\left(T^{\frac{1}{2}} \left(V^{(1)}\right)^{\frac{1}{2}} \log T\right) \ .$$

Additionally, in Section 6.2, we show (Corollary 12) that under proper choice of the constant $c = c\left(V^{(2)}\right)$, the regret of LASER is bounded by

$$L_T(\text{LASER}) \leq L_T(\{\boldsymbol{u}_t\}) + \mathcal{O}\left(T^{\frac{2}{3}} \left(V^{(2)}\right)^{\frac{1}{3}}\right) \ .$$

The two bounds are not comparable in general. For example, assume a constant instantaneous drift $\|\boldsymbol{u}_{t+1}-\boldsymbol{u}_t\| = \nu$ for some constant value ν. In this case the variance and squared variance are, $V^{(1)} = T\nu$ and $V^{(2)} = T\nu^2$. The bound of ARCOR becomes asymptotically $\nu^{\frac{1}{2}} T \log T$, while the bound of LASER becomes asymptotically $\nu^{\frac{2}{3}} T$. Hence the bound of LASER is better in this case.

Another example is polynomial decay of the drift, $\|\boldsymbol{u}_{t+1} - \boldsymbol{u}_t\| \leq t^{-\kappa}$ for some $\kappa > 0$. In this case, for $\kappa \neq 1$ we get[2] $V^{(1)} \leq \sum_{t=1}^{T-1} t^{-\kappa} \leq \int_1^{T-1} t^{-\kappa} dt + 1 = \frac{(T-1)^{1-\kappa}-\kappa}{1-\kappa}$. For $\kappa = 1$ we get $V^{(1)} \leq \log(T-1) + 1$. For LASER we have, for $\kappa \neq 0.5$, $V^{(2)} \leq \sum_{t=1}^{T-1} t^{-2\kappa} \leq \int_1^{T-1} t^{-2\kappa} dt + 1 = \frac{(T-1)^{1-2\kappa}-2\kappa}{1-2\kappa}$. For $\kappa = 0.5$ we get $V^{(2)} \leq \log(T-1) + 1$. Asymptotically, ARCOR outperforms LASER about when $\kappa \geq 0.7$.

Herbster and Warmuth (2001) developed shifting bounds for general gradient descent algorithms with projection of the weight-vector using the Bregman divergence. In their bounds, there is a factor greater than 1 multiplying the term $L_T(\{\boldsymbol{u}_t\})$ (see also theorem 11.4 in Cesa-Bianchi and Lugosi 2006). However, it is possible to get regret bound similar to ARCOR bound above, as they have an implicit parameter that can be tuned with the prior knowledge of $L_T(\{\boldsymbol{u}_t\})$ and $V^{(1)}$, leading to regret of $\mathcal{O}\left(\sqrt{L_T(\{\boldsymbol{u}_t\})V^{(1)}}\right)$, or just $\mathcal{O}\left(\sqrt{TV^{(1)}}\right)$, assuming only the knowledge of $V^{(1)}$.

Busuttil and Kalnishkan (2007) developed a variant of the Aggregating Algorithm (Vovk, 1990) for the non-stationary setting. However, to have sublinear regret they require a strong assumption on the drift $V^{(2)} = o(1)$, while we require only $V^{(2)} = o(T)$ (for LASER) or $V^{(1)} = o(T)$ (for ARCOR).

6.1 Analysis of the ARCOR Algorithm

Let us define additional notation that we will use in our bounds. We denote by t_i the example index for which a restart was performed for the ith time, that is $\Sigma_{t_i} = I$ for all i. We define by n the total number of restarts, or intervals. We denote by $T_i = t_i - t_{i-1}$ the number of examples between two consecutive restarts. Clearly $T = \sum_{i=1}^{n} T_i$. Finally, we denote by $\Sigma^{i-1} = \Sigma_{t_i-1}$ just before the ith restart, and we note that it depends on exactly T_i examples (since the last restart).

In what follows we compare the performance of the ARCOR algorithm to the performance of a sequence of weight vectors $\boldsymbol{u}_t \in \mathbb{R}^d$, which are of norm bounded by R_B. In other words, all the vectors $\boldsymbol{u}_t$ belong to B. We break the proof into four steps. In the first step (Theorem 3) we bound the regret when the algorithm is executed with some value of parameters $\{\Lambda_i\}$ and the resulting covariance matrices. In the second step, summarized in Corollary 4, we remove the dependencies in the covariance matrices by taking a worst case bound. In the third step, summarized in Lemma 5, we upper bound the total number of switches n given the parameters $\{\Lambda_i\}$. Finally, in Corollary 6 we provide the regret bound for a specific choice of the parameters. We now move to state the first theorem.

Theorem 3 *Assume that the ARCOR algorithm is run with an input sequence* $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_T, y_T)$. *Assume that all the inputs are upper bounded by unit norm* $\|\boldsymbol{x}_t\| \leq 1$ *and that the outputs*

2. This is correct because $f(t) = t^{-\kappa}$ is a monotonically decreasing function for $\kappa > 0$ and thus we can lower bound the integral with the right Riemann sum. In addition $f(1) = 1$.

are bounded by $Y = \max_t |y_t|$. *Let* $\boldsymbol{u}_t$ *be any sequence of bounded weight vectors* $\|\boldsymbol{u}_t\| \leq R_B$. *Then, the cumulative loss is bounded by*

$$\begin{aligned} L_T(ARCOR) \leq \; & L_T(\{\boldsymbol{u}_t\}) + 2R_B r \sum_t \frac{1}{\Lambda_{i(t)}} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \\ & + r\boldsymbol{u}_T^\top \Sigma_T^{-1} \boldsymbol{u}_T + 2\left(R_B^2 + Y^2\right) \sum_i^n \log\det\left(\left(\Sigma^i\right)^{-1}\right) \;, \end{aligned}$$

where n *is the number of covariance restarts and* Σ^{i-1} *is the value of the covariance matrix just before the* i*th restart.*

The proof appears in Appendix C. Note that the number of restarts n is not fixed but depends both on the total number of examples T and the scheme used to set the values of the lower bound of the eigenvalues Λ_i. In general, the lower the values of Λ_i are, the smaller number of covariance-restarts occur, yet the larger the value of the last term of the bound is, which scales inversely proportional to Λ_i. A more precise statement is given in the next corollary.

Corollary 4 *Assume that the ARCOR algorithm made* n *restarts and* $\{\Lambda_i\}$ *are monotonically decreasing with* i *(which is satisfied by our choice later). Under the conditions of Theorem 3 we have*

$$\begin{aligned} L_T(ARCOR) \leq \; & L_T(\{\boldsymbol{u}_t\}) + 2R_B r \Lambda_n^{-1} \sum_t \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \\ & + 2\left(R_B^2 + Y^2\right) dn \log\left(1 + \frac{T}{nrd}\right) + r\boldsymbol{u}_T^\top \Sigma_T^{-1} \boldsymbol{u}_T \;. \end{aligned}$$

Proof By definition we have

$$\left(\Sigma^i\right)^{-1} = I + \frac{1}{r} \sum_{t=t_i}^{T_i+t_i} \boldsymbol{x}_t \boldsymbol{x}_t^\top \;.$$

Denote the eigenvalues of $\sum_{t=t_i}^{T_i+t_i} \boldsymbol{x}_t \boldsymbol{x}_t^\top$ by $\lambda_1, \ldots, \lambda_d$. Since $\|\boldsymbol{x}_t\| \leq 1$ their sum is $\mathrm{Tr}\left(\sum_{t=t_i}^{T_i+t_i} \boldsymbol{x}_t \boldsymbol{x}_t^\top\right) \leq T_i$. We use the concavity of the log function to bound $\log\det\left(\left(\Sigma^i\right)^{-1}\right) = \sum_j^d \log\left(1 + \frac{\lambda_j}{r}\right) \leq d\log\left(1 + \frac{T_i}{rd}\right)$. We use concavity again to bound the sum

$$\sum_i^n \log\det\left(\left(\Sigma^i\right)^{-1}\right) \leq \sum_i^n d\log\left(1 + \frac{T_i}{rd}\right) \leq dn\log\left(1 + \frac{T}{nrd}\right) \;,$$

where we used the fact that $\sum_i^n T_i = T$. Substituting the last inequality in Theorem 3, as well as using the monotonicity of the coefficients, $\Lambda_i \geq \Lambda_n$ for all $i \leq n$, yields the desired bound. ■

Implicitly, the second and third terms of the bound have opposite dependence on n. The

second term is decreasing with n. If n is small it means that the lower bound Λ_n is very low (otherwise we would make many restarts) and thus Λ_n^{-1} is large. The third term is increasing with $n \ll T$. We now make this implicit dependence explicit.

Our goal is to bound the number of restarts n as a function of the number of examples T. This depends on the exact sequence of values Λ_i used. The following lemma provides a bound on n given a specific sequence of Λ_i.

Lemma 5 *Assume that the ARCOR algorithm is run with some sequence of Λ_i. Then, the number of restarts is upper bounded by*

$$n \leq \max_N \left\{ N \ : \ T \geq r \sum_i^N \left(\Lambda_i^{-1} - 1 \right) \right\} \ .$$

Proof Since $\sum_{i=1}^n T_i = T$, then the number of restarts is maximized when the number of examples between restarts T_i is minimized. We prove now a lower bound on T_i for all $i = 1 \dots n$. A restart occurs for the ith time when the smallest eigenvalue of Σ_t is smaller (for the first time) than Λ_i.

As before, by definition, $\left(\Sigma^i\right)^{-1} = I + \frac{1}{r} \sum_{t=t_i}^{T_i+t_i} \boldsymbol{x}_t \boldsymbol{x}_t^\top$. By a result in matrix analysis (Golub and Van Loan, 1996, Theorem 8.1.8) we have that there exists a matrix $A \in \mathbb{R}^{d \times T_i}$ with each column belongs to a bounded convex body that satisfy $a_{k,l} \geq 0$ and $\sum_k a_{k,l} \leq 1$ for $l = 1, \dots, T_i$, such that the kth eigenvalue λ_k^i of $\left(\Sigma^i\right)^{-1}$ equals to $\lambda_k^i = 1 + \frac{1}{r} \sum_{l=1}^{T_i} a_{k,l}$. The value of T_i is defined when the largest eigenvalue of $\left(\Sigma^i\right)^{-1}$ hits Λ_i^{-1}. Formally, we get the following lower bound on T_i,

$$\begin{aligned}
\arg \min_{\{a_{k,l}\}} \ & s \\
& \text{s.t.} \ \max_k \left(1 + \frac{1}{r} \sum_{l=1}^{s} a_{k,l} \right) \geq \Lambda_i^{-1} \\
& \quad a_{k,l} \geq 0 \quad \text{for } k = 1, \dots, d, l = 1, \dots, s \\
& \quad \sum_k a_{k,l} \leq 1 \quad \text{for } l = 1, \dots, s \ .
\end{aligned}$$

For a fixed value of s, a maximal value $\max_k \left(1 + \frac{1}{r} \sum_{l=1}^s a_{k,l}\right)$ is obtained when each column of A concentrates the "mass" in one value $k = k_0$ and equal to its maximal value $a_{k_0,l} = 1$ for $l = 1, \dots, s$. That is, we have $a_{k,l} = 1$ for $k = k_0$ and $a_{k,l} = 0$ otherwise. In this case $\max_k \left(1 + \frac{1}{r} \sum_{l=1}^s a_{k,l}\right) = 1 + \frac{1}{r} s$ and the lower bound is obtained when $1 + \frac{1}{r} s = \Lambda_i^{-1}$. Solving for s we get that the shortest possible length of the ith interval is bounded by, $T_i \geq r \left(\Lambda_i^{-1} - 1\right)$. Summing over the last equation we get, $T = \sum_i^n T_i \geq r \sum_i^n \left(\Lambda_i^{-1} - 1\right)$. Thus, the number of restarts is upper bounded by the maximal value n that satisfies the last inequality. ■

We now prove a bound for a specific choice of the parameters $\{\Lambda_i\}$, namely polynomial decay, $\Lambda_i^{-1} = i^{q-1} + 1$ for $q > 1$ (note that the thresholds $\{\Lambda_i\}$ are monotonically decreasing

with i). This scheme of setting $\{\Lambda_i\}$ balances between the amount of drift (need for many restarts) and the property that using the covariance matrix for updates achieves fast convergence. We note that an exponential scheme $\Lambda_i = 2^{-i}$ will lead to very few restarts, and very small eigenvalues of the covariance matrix. Intuitively, this is because the last segment will be about half the length of the entire sequence. Combining Lemma 5 with Corollary 4 we get,

Corollary 6 *Assume that the ARCOR algorithm is run with a polynomial scheme, that is* $\Lambda_i^{-1} = i^{q-1} + 1$ *for some* $q > 1$. *Under the conditions of Theorem 3 we have*

$$\begin{aligned} L_T(ARCOR) \leq \; & L_T(\{\boldsymbol{u}_t\}) + r\boldsymbol{u}_T^\top \Sigma_T^{-1} \boldsymbol{u}_T \\ & + 2\left(R_B^2 + Y^2\right) d\,(qT+1)^{\frac{1}{q}} \log\left(1 + \frac{T}{nrd}\right) \qquad (20) \\ & + 2R_B r\left((qT+1)^{\frac{q-1}{q}} + 1\right) \sum_t \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \; . \qquad (21) \end{aligned}$$

Proof Substituting $\Lambda_i^{-1} = i^{q-1} + 1$ in Lemma 5 we get

$$T \geq r \sum_i^n \left(\Lambda_i^{-1} - 1\right) = r \sum_{i=1}^n i^{q-1} \geq r \int_1^n x^{q-1} dx = \frac{r}{q}\left(n^q - 1\right) \; ,$$

where the middle inequality is correct because $f(x) = x^{q-1}$ for $q > 1$ is a monotonically increasing function and thus we can upper bound the integral with the right Riemann sum. This yields an upper bound on n,

$$n \leq (qT+1)^{\frac{1}{q}} \quad \Rightarrow \quad \Lambda_n^{-1} \leq (qT+1)^{\frac{q-1}{q}} + 1 \; .$$

■

Comparing the last two terms of the bound of Corollary 6 we observe a natural tradeoff in the value of q. The third term of (20) is decreasing with large values of q, while the fourth term of (21) is increasing with q.

Assuming a bound on the deviation $\sum_t \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| = V_T^{(1)} \leq \mathcal{O}\left(T^{1/p}\right)$, or in other words $p = (\log T) / \left(\log V^{(1)}\right)$. We set a drift dependent parameter $q = (2p)/(p+1) = (2\log T) / \left(\log T + \log V^{(1)}\right)$ and get that the sum of (20) and (21) is of order $\mathcal{O}\left(T^{\frac{p+1}{2p}} \log(T)\right) = \mathcal{O}\left(\sqrt{V^{(1)} T} \log T\right)$.

Few comments are in order. First, as long as $p > 1$ the sum of (20) and (21) is $o(T)$ and thus vanishing. Second, when the drift is very small, that is $p \approx -(1+\epsilon)$, the algorithm sets $q \approx 2 + (2/\epsilon)$, and thus it will not make any restarts, and the bound of $\mathcal{O}(\log T)$ for the stationary case is retrieved. In other words, for this choice of q the algorithm will have only one interval, and there will be no restarts.

To conclude, we showed that if the algorithm is given an upper bound on the amount of drift, which is sub-linear in T, it can achieve sub-linear regret. Furthermore, if it is known that there is no non-stationarity in the reference vectors, then running the algorithm with large enough q will have a regret logarithmic in T.

6.2 Analysis of the LASER Algorithm

We now analyze the performance of the LASER algorithm in the worst-case setting in six steps. First, state a technical lemma that is used in the second step (Theorem 8), in which we bound the regret with a quantity proportional to $\sum_{t=1}^{T} \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t$. Third, in Lemma 9 we bound each of the summands with two terms, one logarithmic and one linear in the eigenvalues of the matrices D_t. In the fourth (Lemma 10) and fifth (Lemma 11) steps we bound the eigenvalues of D_t first for scalars and then extend the results to matrices. Finally, in Corollary 12 we put all these results together and get the desired bound.

Lemma 7 *For all t the following statement holds*

$$D'_{t-1} D_t^{-1} \boldsymbol{x}_t \boldsymbol{x}_t^\top D_t^{-1} D'_{t-1} + D'_{t-1} \left(D_t^{-1} D'_{t-1} + c^{-1} I \right) - D_{t-1}^{-1} \preceq 0 \ ,$$

where as defined in (19) *we have* $D'_{t-1} = \left(I + c^{-1} D_{t-1} \right)^{-1}$.

The proof appears in Appendix D. We next bound the cumulative loss of the algorithm.

Theorem 8 *Assume that the labels are bounded* $\sup_t |y_t| \leq Y$ *for some* $Y \in \mathbb{R}$. *Then the following bound holds*

$$L_T(LASER) \leq \min_{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_T} \left[L_T(\{\boldsymbol{u}_t\}) + c V_T^{(2)}(\{\boldsymbol{u}_t\}) + b \left\| \boldsymbol{u}_1 \right\|^2 \right] + Y^2 \sum_{t=1}^{T} \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \ . \tag{22}$$

Proof Fix t. A long algebraic manipulation, given in Appendix E, yields

$$\begin{aligned}
&(y_t - \hat{y}_t)^2 + \min_{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{t-1}} Q_{t-1}(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{t-1}) - \min_{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t} Q_t(\boldsymbol{u}_1, \ldots, \boldsymbol{u}_t) \\
&= (y_t - \hat{y}_t)^2 + 2 y_t \boldsymbol{x}_t^\top D_t^{-1} D'_{t-1} \boldsymbol{e}_{t-1} + \boldsymbol{e}_{t-1}^\top \left[- D_{t-1}^{-1} + D'_{t-1} \left(D_t^{-1} D'_{t-1} + c^{-1} I \right) \right] \boldsymbol{e}_{t-1} \\
&\quad + y_t^2 \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t - y_t^2 \ .
\end{aligned} \tag{23}$$

Substituting the specific value of the predictor $\hat{y}_t = \boldsymbol{x}_t^\top D_t^{-1} D'_{t-1} \boldsymbol{e}_{t-1}$ from (18), we get that (23) equals to

$$\begin{aligned}
& \hat{y}_t^2 + y_t^2 \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t + \boldsymbol{e}_{t-1}^\top \left[- D_{t-1}^{-1} + D'_{t-1} \left(D_t^{-1} D'_{t-1} + c^{-1} I \right) \right] \boldsymbol{e}_{t-1} \\
= \ & \boldsymbol{e}_{t-1}^\top D'_{t-1} D_t^{-1} \boldsymbol{x}_t \boldsymbol{x}_t^\top D_t^{-1} D'_{t-1} \boldsymbol{e}_{t-1} + y_t^2 \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \\
& + \boldsymbol{e}_{t-1}^\top \left[- D_{t-1}^{-1} + D'_{t-1} \left(D_t^{-1} D'_{t-1} + c^{-1} I \right) \right] \boldsymbol{e}_{t-1} \\
= \ & \boldsymbol{e}_{t-1}^\top \tilde{D}_t \boldsymbol{e}_{t-1} + y_t^2 \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \ ,
\end{aligned} \tag{24}$$

where $\tilde{D}_t = D'_{t-1} D_t^{-1} \boldsymbol{x}_t \boldsymbol{x}_t^\top D_t^{-1} D'_{t-1} - D_{t-1}^{-1} + D'_{t-1} \left(D_t^{-1} D'_{t-1} + c^{-1} I \right)$. Using Lemma 7 we upper bound $\tilde{D}_t \preceq 0$ and thus (24) is bounded,

$$y_t^2 \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \leq Y^2 \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \ .$$

Finally, summing over $t \in \{1, \ldots, T\}$ gives the desired bound,

$$L_T(\text{LASER}) - \min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_T} \left[b \left\| \boldsymbol{u}_1 \right\|^2 + cV_T^{(2)}(\{\boldsymbol{u}_t\}) + L_T(\{\boldsymbol{u}_t\}) \right] \leq Y^2 \sum_{t=1}^{T} \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \ .$$

■

In the next lemma we further bound the right term of (22). This type of bound is based on the usage of the covariance-like matrix D.

Lemma 9

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \leq \ln \left| \frac{1}{b} D_T \right| + c^{-1} \sum_{t=1}^{T} \text{Tr}\left(D_{t-1} \right) \ . \tag{25}$$

Proof Let $B_t \doteq D_t - \boldsymbol{x}_t \boldsymbol{x}_t^\top = \left(D_{t-1}^{-1} + c^{-1} I \right)^{-1} \succ 0$.

$$\begin{aligned}
\boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t &= \text{Tr}\left(\boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \right) = \text{Tr}\left(D_t^{-1} \boldsymbol{x}_t \boldsymbol{x}_t^\top \right) \\
&= \text{Tr}\left(D_t^{-1} \left(D_t - B_t \right) \right) \\
&= \text{Tr}\left(D_t^{-1/2} \left(D_t - B_t \right) D_t^{-1/2} \right) \\
&= \text{Tr}\left(I - D_t^{-1/2} B_t D_t^{-1/2} \right) \\
&= \sum_{j=1}^{d} \left[1 - \lambda_j \left(D_t^{-1/2} B_t D_t^{-1/2} \right) \right] \ .
\end{aligned}$$

We continue using $1 - x \leq -\ln(x)$ and get

$$\begin{aligned}
\boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t &\leq -\sum_{j=1}^{d} \ln \left[\lambda_j \left(D_t^{-1/2} B_t D_t^{-1/2} \right) \right] \\
&= -\ln \left[\prod_{j=1}^{d} \lambda_j \left(D_t^{-1/2} B_t D_t^{-1/2} \right) \right] \\
&= -\ln \left| D_t^{-1/2} B_t D_t^{-1/2} \right| \\
&= \ln \frac{|D_t|}{|B_t|} = \ln \frac{|D_t|}{\left| D_t - \boldsymbol{x}_t \boldsymbol{x}_t^\top \right|} \ .
\end{aligned}$$

It follows that

$$\begin{aligned}
\boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t &\leq \ln \frac{|D_t|}{\left| \left(D_{t-1}^{-1} + c^{-1} I \right)^{-1} \right|} \\
&= \ln \frac{|D_t|}{|D_{t-1}|} \left| \left(I + c^{-1} D_{t-1} \right) \right| \\
&= \ln \frac{|D_t|}{|D_{t-1}|} + \ln \left| \left(I + c^{-1} D_{t-1} \right) \right| \ .
\end{aligned}$$

and because $\ln\left|\frac{1}{b}D_0\right| \geq 0$ we get

$$\sum_{t=1}^{T} \boldsymbol{x}_t^\top D_t^{-1} \boldsymbol{x}_t \leq \ln\left|\frac{1}{b}D_T\right| + \sum_{t=1}^{T} \ln\left|\left(I + c^{-1}D_{t-1}\right)\right| \leq \ln\left|\frac{1}{b}D_T\right| + c^{-1}\sum_{t=1}^{T} \mathrm{Tr}\left(D_{t-1}\right) \ .$$

■

At first sight it seems that the right term of (25) may grow super-linearly with T, as each of the matrices D_t grows with t. The next two lemmas show that this is not the case, and in fact, the right term of (25) is not growing too fast, which will allow us to obtain a sub-linear regret bound. Lemma 10 analyzes the properties of the recursion of D defined in (11) for scalars, that is $d = 1$. In Lemma 11 we extend this analysis to matrices.

Lemma 10 *Define $f(\lambda) = \lambda\beta/\left(\lambda + \beta\right) + x^2$ for $\beta, \lambda \geq 0$ and some $x^2 \leq \gamma^2$. Then:*

1. $f(\lambda) \leq \beta + \gamma^2$
2. $f(\lambda) \leq \lambda + \gamma^2$
3. $f(\lambda) \leq \max\left\{\lambda, \frac{3\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2\beta}}{2}\right\}$

Proof For the first property we have $f(\lambda) = \lambda\beta/\left(\lambda + \beta\right) + x^2 \leq \beta \times 1 + x^2$. The second property follows from the symmetry between β and λ. To prove the third property we decompose the function as, $f(\lambda) = \lambda - \frac{\lambda^2}{\lambda+\beta} + x^2$. Therefore, the function is bounded by its argument $f(\lambda) \leq \lambda$ if, and only if, $-\frac{\lambda^2}{\lambda+\beta} + x^2 \leq 0$. Since we assume $x^2 \leq \gamma^2$, the last inequality holds if, $-\lambda^2 + \gamma^2\lambda + \gamma^2\beta \leq 0$, which holds for $\lambda \geq \frac{\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2\beta}}{2}$.

To conclude. If $\lambda \geq \frac{\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2\beta}}{2}$, then $f(\lambda) \leq \lambda$. Otherwise, by the second property, we have

$$f(\lambda) \leq \lambda + \gamma^2 \leq \frac{\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2\beta}}{2} + \gamma^2 = \frac{3\gamma^2 + \sqrt{\gamma^4 + 4\gamma^2\beta}}{2},$$

as required. ■

We build on Lemma 10 to bound the maximal eigenvalue of the matrices D_t.

Lemma 11 *Assume $\|\boldsymbol{x}_t\|^2 \leq X^2$ for some X. Then, the eigenvalues of D_t (for $t \geq 1$), denoted by $\lambda_i\left(D_t\right)$, are upper bounded by*

$$\max_i \lambda_i\left(D_t\right) \leq \max\left\{\frac{3X^2 + \sqrt{X^4 + 4X^2c}}{2}, b + X^2\right\} \ .$$

Proof By induction. From (11) we have that $\lambda_i(D_1) \leq b + X^2$ for $i = 1, \ldots, d$. We proceed with a proof for some t. For simplicity, denote by $\lambda_i = \lambda_i(D_{t-1})$ the ith eigenvalue of D_{t-1}

with a corresponding eigenvector $\boldsymbol{v}_i$. From (11) we have

$$\begin{aligned} D_t &= \left(D_{t-1}^{-1} + c^{-1} I\right)^{-1} + \boldsymbol{x}_t \boldsymbol{x}_t^\top \\ &\preceq \left(D_{t-1}^{-1} + c^{-1} I\right)^{-1} + I \left\|\boldsymbol{x}_t\right\|^2 \\ &= \sum_i^d \boldsymbol{v}_i \boldsymbol{v}_i^\top \left(\left(\lambda_i^{-1} + c^{-1}\right)^{-1} + \left\|\boldsymbol{x}_t\right\|^2\right) \\ &= \sum_i^d \boldsymbol{v}_i \boldsymbol{v}_i^\top \left(\frac{\lambda_i c}{\lambda_i + c} + \left\|\boldsymbol{x}_t\right\|^2\right) \ . \end{aligned} \tag{26}$$

Plugging Lemma 10 in (26) we get

$$\begin{aligned} D_t &\preceq \sum_i^d \boldsymbol{v}_i \boldsymbol{v}_i^\top \max\left\{\frac{3X^2 + \sqrt{X^4 + 4X^2 c}}{2}, b + X^2\right\} \\ &= \max\left\{\frac{3X^2 + \sqrt{X^4 + 4X^2 c}}{2}, b + X^2\right\} I \ . \end{aligned}$$

■

Finally, equipped with the above lemmas we are able to prove the main result of this section.

Corollary 12 *Assume* $\left\|\boldsymbol{x}_t\right\|^2 \leq X^2$, $|y_t| \leq Y$. *Then*

$$\begin{aligned} L_T(LASER) \leq \ & b \left\|\mathbf{u}_1\right\|^2 + L_T(\{\boldsymbol{u}_t\}) + Y^2 \ln\left|\frac{1}{b} D_T\right| + c^{-1} Y^2 \mathrm{Tr}\left(D_0\right) + c V^{(2)} \\ & + c^{-1} Y^2 T d \max\left\{\frac{3X^2 + \sqrt{X^4 + 4X^2 c}}{2}, b + X^2\right\} \ . \end{aligned} \tag{27}$$

Furthermore, set $b = \varepsilon c$ *for some* $0 < \varepsilon < 1$. *Denote by* $\mu = \max\left\{9/8X^2, \frac{\left(b+X^2\right)^2}{8X^2}\right\}$ *and* $M = \max\left\{3X^2, b + X^2\right\}$. *If* $V^{(2)} \leq T\frac{\sqrt{2}Y^2 dX}{\mu^{3/2}}$ *(low drift) then by setting*

$$c = \frac{\sqrt{2} T Y^2 d X}{\left(V^{(2)}\right)^{2/3}} \tag{28}$$

we have

$$\begin{aligned} L_T(LASER) \leq \ & b \left\|\mathbf{u}_1\right\|^2 + 3\left(\sqrt{2} Y^2 dX\right)^{2/3} T^{2/3} \left(V^{(2)}\right)^{1/3} + \frac{\varepsilon}{1-\varepsilon} Y^2 d + L_T(\{\boldsymbol{u}_t\}) \\ & + Y^2 \ln\left|\frac{1}{b} D_T\right| \ . \end{aligned} \tag{29}$$

The proof appears in Appendix F. Note that if $V^{(2)} \geq T\frac{Y^2 dM}{\mu^2}$ then by setting $c = \sqrt{Y^2 dMT/V^{(2)}}$ we have

$$L_T(\text{LASER}) \leq b\,\|\mathbf{u}_1\|^2 + 2\sqrt{Y^2 dTMV^{(2)}} + \frac{\varepsilon}{1-\varepsilon}Y^2 d + L_T(\{\boldsymbol{u}_t\}) + Y^2 \ln\left|\frac{1}{b}D_T\right| \ . \quad (30)$$

(see Appendix G for details). The last bound is linear in T and can be obtained also by a naive algorithm that outputs $\hat{y}_t = 0$ for all t.

A few remarks are in order. When the variance $V^{(2)} = 0$ goes to zero, we set $c = \infty$ and thus we have $D_t = bI + \sum_{s=1}^{t} \boldsymbol{x}_s \boldsymbol{x}_s^\top$ used in recent algorithms (Vovk, 2001; Forster, 1999; Hayes, 1996; Cesa-Bianchi et al., 2005). In this case the algorithm reduces to the algorithm by Forster (1999) (which is also the AAR algorithm of Vovk 2001), with the same logarithmic regret bound (note that the term $\ln\left|\frac{1}{b}D_T\right|$ in the bounds is logarithmic in T, see the proof of Forster 1999). See also the work of Azoury and Warmuth (2001).

7. Simulations

We evaluated our algorithms on four data sets, one synthetic and three real-world. The synthetic data set contains $2,000$ points $\boldsymbol{x}_t \in \mathbb{R}^{20}$, where the first ten coordinates were grouped into five groups of size two. Each such pair was drawn from a 45° rotated Gaussian distribution with standard deviations 10 and 1. The remaining 10 coordinates of $\boldsymbol{x}_t$ were drawn from independent Gaussian distributions $\mathcal{N}(0, 2)$. The data set was generated using a sequence of vectors $\boldsymbol{u}_t \in \mathbb{R}^{20}$ for which the only non-zero coordinates are the first two, where their values are the coordinates of a unit vector that is rotating with a constant rate. Specifically, we have $\|\boldsymbol{u}_t\| = 1$ and the instantaneous drift $\|\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\|$ is constant. The labels were set according to $y_t = \boldsymbol{x}_t^\top \boldsymbol{u}_t$.

The first two real-world data sets were generated from echoed speech signal. The first speech echoed signal was generated using FIR filter with k delays and varying attenuated amplitude. This effect imitates acoustic echo reflections from large, distant and dynamic obstacles. The difference equation $y(n) = x(n) + \sum_{D=1}^{k} A(n)x(n-D) + v(n)$ was used, where D is a delay in samples, the coefficient $A(n)$ describes the changing attenuation related to object reflection and $v(n) \sim \mathcal{N}\left(0, 10^{-3}\right)$ is a white noise. The second speech echoed signal was generated using a flange IIR filter, where the delay is not constant, but changing with time. This effect imitates time stretching of audio signal caused by moving and changing objects in the room. The difference equation $y(n) = x(n) + Ay\left(n - D(n)\right) + v(n)$ was used.

The last real-world data set was taken from the Kaggle competition "Global Energy Forecasting Competition 2012 - Load Forecasting".[3] This data set includes hourly demand for four and a half years from 20 different geographic regions, and similar hourly temperature readings from 11 zones, which we used as features $\boldsymbol{x}_t \in \mathbb{R}^{11}$. Based on this data set, we generated drifting and shifting data as follows: we predict the load 3 times a day (thus there is a drift between day and night), and every half a year there is a switch in the region where the load is predicted.

Five algorithms were evaluated: NLMS (normalized least mean square) (Bershad, 1986; Bitmead and Anderson, 1980) which is a state-of-the-art first-order algorithm, AROWR

3. The data set was taken from
`http://www.kaggle.com/c/global-energy-forecasting-competition-2012-load-forecasting` .

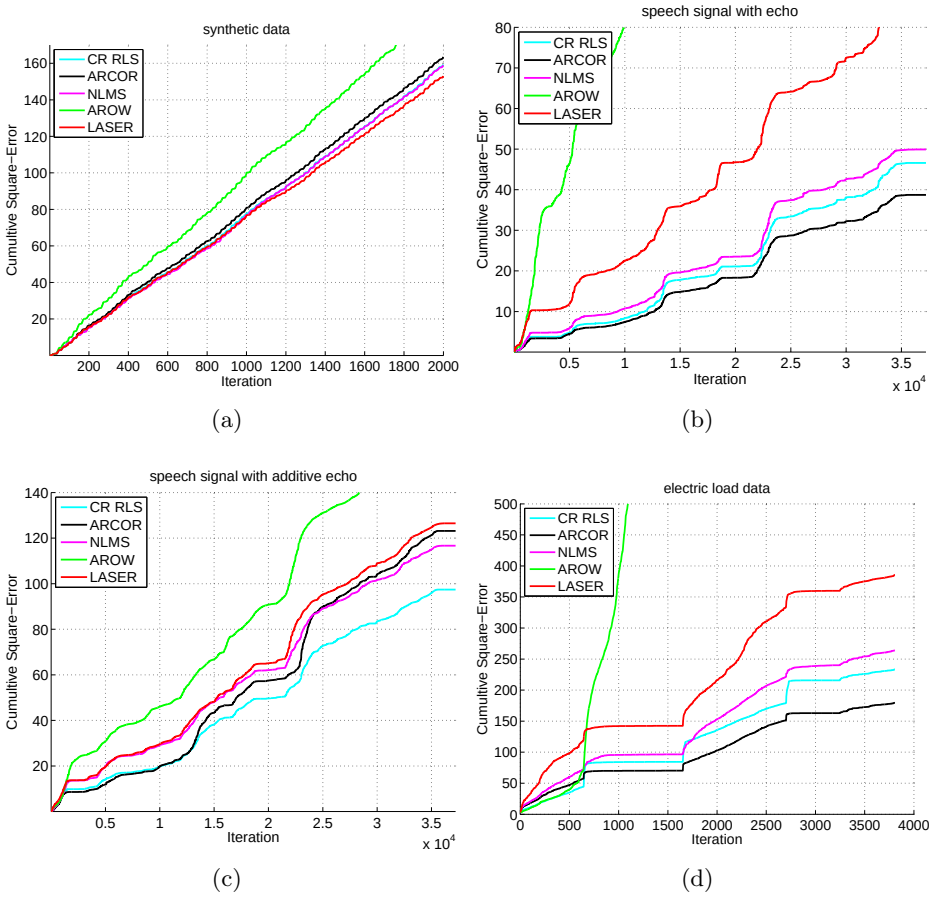


Figure 1: Cumulative squared loss for AROWR, ARCOR, LASER, NLMS and CR-RLS vs iteration. (a) Results for synthetic data set with drift. (b) Results for a problem of acoustic echo cancellation on speech signal generated using FIR filter and (c) IIR filter. (d) Results for a problem of electric load prediction (best shown in color).

(AROW for Regression) with no restarts nor projection, ARCOR, LASER and CR-RLS. We note that AAR (Vovk, 2001) is a special case of LASER and RLS is a special case of CR-RLS, for a specific choice of their respective parameters ($c = \infty$ for LASER and $T_0 = \infty$ for CR-RLS) . Additionally, the performance of AROWR, AAR and RLS is similar, and thus only the performance of AROWR is shown. For the synthetic data set the algorithms' parameters were tuned using a single random sequence. For the speech signal the algorithms' parameters were tuned on 10% of the signal, then the best parameter choices for each algorithm were used to evaluate the performance on the remaining signal. Similarly, for the load data set the algorithms' parameters were tuned on 20% of the signal.

The results are summarized in Figure 1. AROWR performs the worst on all data sets as it converges very fast and thus not able to track the changes in the data. Focusing on Figure 1(a), showing the results for the synthetic signal, we observe that ARCOR performs relatively bad as suggested by our analysis for constant, yet not too large, drift. Both CR-RLS and NLMS perform better, where CR-RLS is slightly better as it is a second-order algorithm, and allows to converge faster between switches. On the other hand, NLMS is not converging and is able to adapt to the drift. Finally, LASER performs the best, as hinted by its analysis, for which the bound is lower where there is a constant drift.

Moving to Figure 1(b), showing the results for first echoed speech signal with varying amplitude, we observe that LASER is the worst among all algorithms except AROWR. Indeed, it prevents the convergence by keeping the learning rate far from zero, yet it is a min-max algorithm designed for the worst-case, which is not the case for real-world speech data. However, speech data is highly regular and the instantaneous drift vary. NLMS performs better as it does not converge, yet both CR-RLS and ARCOR perform even better, as they both do not converge due to covariance resets on the one hand, and second-order updates on the other hand. ARCOR outperforms CR-RLS as the former adapts the resets to actual data, and does not use pre-defined scheduling as the later.

Figure 1(c) summarizes the results for evaluations on the second echoed speech signal. Note that the amount of drift grows since the data is generated using flange filter. Both LASER and ARCOR are outperformed as both assume drift that is sublinear or at most linear, which is not the case. CR-RLS outperforms NLMS. The later is first order, so is able to adapt to changes, yet has slower convergence rate. The former is able to cope with drift due to resets.

Finally, Figure 1(d) summarizes the results for the electric load data set. ARCOR outperforms other algorithms, as the drift is sublinear and it has the ability to adapt resets to the data. Again, LASER is a min-max algorithm designed for the worst case, which is usually not the case for real-world data.

Interestingly, in all experiments, NLMS was not performing the best nor the worst. There is no clear winner among the three algorithms that are both second-order (ARCOR, LASER, CR-RLS), and designed to adapt to drifts. Intuitively, if the drift suits the assumptions of an algorithm, that algorithm would perform the best, and otherwise, its performance may even be worse than of NLMS.

We have seen above that ARCOR performs a projection step, which partially was motivated from the analysis. We now evaluate its need and affect in practice on two speech problems. We test two modifications of ARCOR, resulting in four variants altogether. First, we replace the polynomial thresholds scheme to the constant thresholds scheme, that is, all thresholds are equal. Second, we omit the projection step. The results are summarized in Figure 2. The line corresponding to the original algorithm, is called "proj, poly" as it performs a projection step and uses polynomial scheme for the lower-bound on eigenvalues. The version that omits projection and uses constant scheme, called "no proj, const", is most similar to CR-RLS. Both resets the covariance matrix, CR-RLS after fixed amount of iterations, while "ARCOR-no proj, const" when the eigenvalues meets a specified fixed lower bound. The difference between the two plots is the amount of drift used: the left plot shows results for sublinear drift, and the right plot shows results with increasing per-instance drift. The original version, as hinted by the analysis, is designed to work with sub-linear drift,

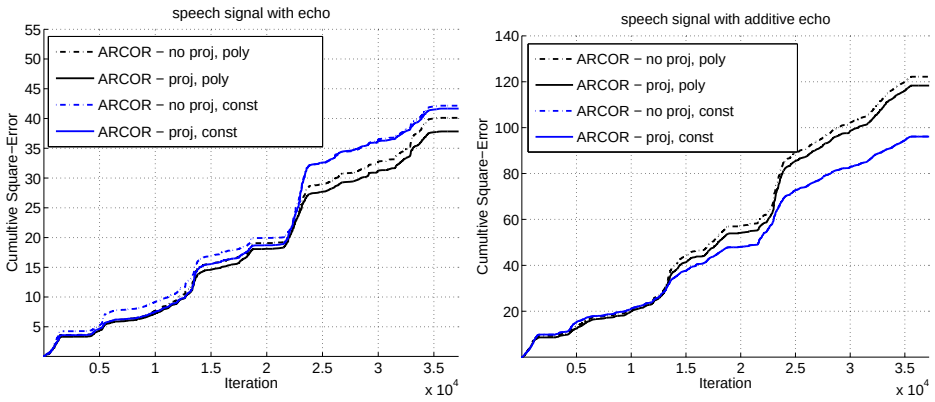


Figure 2: Cumulative squared loss of four variants of ARCOR vs iteration.

and performs the best in this case. However, when this assumption over the amount of drift breaks, this version is not optimal anymore, and constant scheme performs better, as it allows the algorithm to adapt to non-vanishing drift. Finally, in both data sets, the algorithm that performs the best performs a projection step after each iteration, providing some empirical evidence for its need.

8. Summary and Conclusions

We proposed and analyzed two novel algorithms for non-stationary online regression designed and analyzed with the squared loss in the worst-case regret framework. The ARCOR algorithm was built on AROWR. It employs second-order information, yet performs data-dependent covariance resets, which provides it the ability to track drifts. The LASER algorithm was built on the last-step minmax predictor with the proper modifications for non-stationary problems. Our algorithms require some prior knowledge of the drift to get optimal performance, and each algorithm works best in other drift level. The optimal setting depends on the actual drift in the data and the optimality of our bounds is an open issue.

Few open directions are possible. First, extension of these algorithms to other loss functions rather than the squared loss. Second, currently, direct implementation of both algorithms requires either matrix inversion or eigenvector decomposition. A possible direction is to design a more efficient version of these algorithms. Third, an interesting direction is to design algorithms that automatically detect the level of drift, or do not need this information before run-time.

Acknowledgments: This research was funded in part by the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI) and in part by an Israeli Science Foundation grant ISF- 1567/10.

Appendix A. Proof of Lemma 1

Proof We calculate

$$
\begin{aligned}
P_t\left(\boldsymbol{u}_t\right) &= \min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{t-1}} \left(b\left\|\boldsymbol{u}_1\right\|^2 + c\sum_{s=1}^{t-1}\left\|\boldsymbol{u}_{s+1}-\boldsymbol{u}_s\right\|^2 + \sum_{s=1}^{t}\left(y_s - \boldsymbol{u}_s^\top \boldsymbol{x}_s\right)^2 \right) \\
&= \min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{t-1}} \left(b\left\|\boldsymbol{u}_1\right\|^2 + c\sum_{s=1}^{t-2}\left\|\boldsymbol{u}_{s+1}-\boldsymbol{u}_s\right\|^2 + \sum_{s=1}^{t-1}\left(y_s - \boldsymbol{u}_s^\top \boldsymbol{x}_s\right)^2 + c\left\|\boldsymbol{u}_t-\boldsymbol{u}_{t-1}\right\|^2 \right. \\
&\qquad \left. + \left(y_t - \boldsymbol{u}_t^\top \boldsymbol{x}_t\right)^2 \right) \\
&= \min_{\boldsymbol{u}_{t-1}} \min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{t-2}} \left(b\left\|\boldsymbol{u}_1\right\|^2 + c\sum_{s=1}^{t-2}\left\|\boldsymbol{u}_{s+1}-\boldsymbol{u}_s\right\|^2 + \sum_{s=1}^{t-1}\left(y_s - \boldsymbol{u}_s^\top \boldsymbol{x}_s\right)^2 + c\left\|\boldsymbol{u}_t-\boldsymbol{u}_{t-1}\right\|^2 \right. \\
&\qquad \left. + \left(y_t - \boldsymbol{u}_t^\top \boldsymbol{x}_t\right)^2 \right) \\
&= \min_{\boldsymbol{u}_{t-1}} \left[\min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{t-2}} \left(b\left\|\boldsymbol{u}_1\right\|^2 + c\sum_{s=1}^{t-2}\left\|\boldsymbol{u}_{s+1}-\boldsymbol{u}_s\right\|^2 + \sum_{s=1}^{t-1}\left(y_s - \boldsymbol{u}_s^\top \boldsymbol{x}_s\right)^2 \right) \right. \\
&\qquad \left. + c\left\|\boldsymbol{u}_t-\boldsymbol{u}_{t-1}\right\|^2 + \left(y_t - \boldsymbol{u}_t^\top \boldsymbol{x}_t\right)^2 \right] \\
&= \min_{\boldsymbol{u}_{t-1}} \left(P_{t-1}\left(\boldsymbol{u}_{t-1}\right) + c\left\|\boldsymbol{u}_t-\boldsymbol{u}_{t-1}\right\|^2 + \left(y_t - \boldsymbol{u}_t^\top \boldsymbol{x}_t\right)^2 \right) .
\end{aligned}
$$

■

Appendix B. Proof of Lemma 2

Proof By definition

$$
P_1\left(\boldsymbol{u}_1\right) = Q_1\left(\boldsymbol{u}_1\right) = b\left\|\boldsymbol{u}_1\right\|^2 + \left(y_1 - \boldsymbol{u}_1^\top \boldsymbol{x}_1\right)^2 = \boldsymbol{u}_1^\top \left(bI + \boldsymbol{x}_1\boldsymbol{x}_1^\top\right)\boldsymbol{u}_1 - 2y_1\boldsymbol{u}_1^\top \boldsymbol{x}_1 + y_1^2 ,
$$

and indeed $D_1 = bI + \boldsymbol{x}_1\boldsymbol{x}_1^\top$, $\boldsymbol{e}_1 = y_1\boldsymbol{x}_1$, and $f_1 = y_1^2$.

We proceed by induction, assume that, $P_{t-1}\left(\boldsymbol{u}_{t-1}\right)=\boldsymbol{u}_{t-1}^{\top}D_{t-1}\boldsymbol{u}_{t-1}-2\boldsymbol{u}_{t-1}^{\top}\boldsymbol{e}_{t-1}+f_{t-1}$. Applying Lemma 1 we get

$$\begin{aligned}
P_t\left(\boldsymbol{u}_t\right) &= \min_{\boldsymbol{u}_{t-1}}\left(\boldsymbol{u}_{t-1}^{\top}D_{t-1}\boldsymbol{u}_{t-1}-2\boldsymbol{u}_{t-1}^{\top}\boldsymbol{e}_{t-1}+f_{t-1}+c\left\|\boldsymbol{u}_t-\boldsymbol{u}_{t-1}\right\|^2+\left(y_t-\boldsymbol{u}_t^{\top}\boldsymbol{x}_t\right)^2\right)\\
&= \min_{\boldsymbol{u}_{t-1}}\left(\boldsymbol{u}_{t-1}^{\top}\left(cI+D_{t-1}\right)\boldsymbol{u}_{t-1}-2\boldsymbol{u}_{t-1}^{\top}\left(c\boldsymbol{u}_t+\boldsymbol{e}_{t-1}\right)+f_{t-1}+c\left\|\boldsymbol{u}_t\right\|^2\right.\\
&\quad\left.+\left(y_t-\boldsymbol{u}_t^{\top}\boldsymbol{x}_t\right)^2\right)\\
&= -\left(c\boldsymbol{u}_t+\boldsymbol{e}_{t-1}\right)^{\top}\left(cI+D_{t-1}\right)^{-1}\left(c\boldsymbol{u}_t+\boldsymbol{e}_{t-1}\right)+f_{t-1}+c\left\|\boldsymbol{u}_t\right\|^2+\left(y_t-\boldsymbol{u}_t^{\top}\boldsymbol{x}_t\right)^2\\
&= \boldsymbol{u}_t^{\top}\left(cI+\boldsymbol{x}_t\boldsymbol{x}_t^{\top}-c^2\left(cI+D_{t-1}\right)^{-1}\right)\boldsymbol{u}_t-2\boldsymbol{u}_t^{\top}\left[c\left(cI+D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+y_t\boldsymbol{x}_t\right]\\
&\quad-\boldsymbol{e}_{t-1}^{\top}\left(cI+D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+f_{t-1}+y_t^2\ .
\end{aligned}$$

Using the Woodbury identity we continue to develop the last equation,

$$\begin{aligned}
&= \boldsymbol{u}_t^{\top}\left(cI+\boldsymbol{x}_t\boldsymbol{x}_t^{\top}-c^2\left[c^{-1}I-c^{-2}\left(D_{t-1}^{-1}+c^{-1}I\right)^{-1}\right]\right)\boldsymbol{u}_t\\
&\quad-2\boldsymbol{u}_t^{\top}\left[\left(I+c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+y_t\boldsymbol{x}_t\right]\\
&\quad-\boldsymbol{e}_{t-1}^{\top}\left(cI+D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+f_{t-1}+y_t^2\\
&= \boldsymbol{u}_t^{\top}\left(\left(D_{t-1}^{-1}+c^{-1}I\right)^{-1}+\boldsymbol{x}_t\boldsymbol{x}_t^{\top}\right)\boldsymbol{u}_t-2\boldsymbol{u}_t^{\top}\left[\left(I+c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+y_t\boldsymbol{x}_t\right]\\
&\quad-\boldsymbol{e}_{t-1}^{\top}\left(cI+D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+f_{t-1}+y_t^2\ ,
\end{aligned}$$

and indeed $D_t=\left(D_{t-1}^{-1}+c^{-1}I\right)^{-1}+\boldsymbol{x}_t\boldsymbol{x}_t^{\top}$, $\boldsymbol{e}_t=\left(I+c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+y_t\boldsymbol{x}_t$ and, $f_t=f_{t-1}-\boldsymbol{e}_{t-1}^{\top}\left(cI+D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1}+y_t^2$, as desired. ■

Appendix C. Proof of Theorem 3

We prove the theorem in four steps. First, we state a technical lemma, for which we define the following notation

$$\begin{aligned}
d_t\left(\boldsymbol{z},\boldsymbol{v}\right)&=\left(\boldsymbol{z}-\boldsymbol{v}\right)^{\top}\Sigma_t^{-1}\left(\boldsymbol{z}-\boldsymbol{v}\right)\ ,\\
d_{\tilde{t}}\left(\boldsymbol{z},\boldsymbol{v}\right)&=\left(\boldsymbol{z}-\boldsymbol{v}\right)^{\top}\tilde{\Sigma}_t^{-1}\left(\boldsymbol{z}-\boldsymbol{v}\right)\ ,\\
\chi_t&=\boldsymbol{x}_t^{\top}\Sigma_{t-1}\boldsymbol{x}_t\ .
\end{aligned}$$

Second, we define a telescopic sum and in Lemma 14 prove a lower bound for each element. Third, in Lemma 15 we upper bound one term of the telescopic sum, and finally, in the fourth step we combine all these parts to conclude the proof. Let us start with the technical lemma.

Lemma 13 *Let $\tilde{\boldsymbol{w}}_t$ and $\tilde{\Sigma}_t$ be defined in* (7) *and* (8)*, then*

$$d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_{\tilde{t}}\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) = \frac{1}{r}\ell_t - \frac{1}{r}g_t - \frac{\ell_t \chi_t}{r\left(r+\chi_t\right)} \ ,$$

where $\ell_t = \left(y_t - \boldsymbol{w}_{t-1}^\top \boldsymbol{x}_t\right)^2$ and $g_t = \left(y_t - \boldsymbol{u}_{t-1}^\top \boldsymbol{x}_t\right)^2$.

Proof We start by writing the distances explicitly,

$$\begin{aligned} & d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_{\tilde{t}}\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) \\ = & -\left(\boldsymbol{u}_{t-1} - \tilde{\boldsymbol{w}}_t\right)^\top \tilde{\Sigma}_t^{-1}\left(\boldsymbol{u}_{t-1} - \tilde{\boldsymbol{w}}_t\right) + \left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right)^\top \Sigma_{t-1}^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right) \ . \end{aligned}$$

Substituting $\tilde{\boldsymbol{w}}_t$ as appears in (8) the last equation becomes

$$\begin{aligned} & -\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right)^\top \tilde{\Sigma}_t^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right) + 2(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1})\tilde{\Sigma}_t^{-1}\Sigma_{t-1}\boldsymbol{x}_t \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})}{r + \boldsymbol{x}_t^\top \Sigma_{t-1}\boldsymbol{x}_t} \\ & -\left(\frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})}{r + \boldsymbol{x}_t^\top \Sigma_{t-1}\boldsymbol{x}_t}\right)^2 \boldsymbol{x}_t^\top \Sigma_{t-1}\tilde{\Sigma}_t^{-1}\Sigma_{t-1}\boldsymbol{x}_t + \left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right)^\top \Sigma_{t-1}^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right) \ . \end{aligned}$$

Plugging $\tilde{\Sigma}_t$ as appears in (7) we get

$$\begin{aligned} & d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_{\tilde{t}}\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) \\ = & -\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right)^\top \left(\Sigma_{t-1}^{-1} + \frac{1}{r}\boldsymbol{x}_t\boldsymbol{x}_t^\top\right)\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right) \\ & + 2(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1})^\top \left(\Sigma_{t-1}^{-1} + \frac{1}{r}\boldsymbol{x}_t\boldsymbol{x}_t^\top\right)\Sigma_{t-1}\boldsymbol{x}_t \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})}{r + \boldsymbol{x}_t^\top \Sigma_{t-1}\boldsymbol{x}_t} \\ & - \frac{(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1})^2}{\left(r + \boldsymbol{x}_t^\top \Sigma_{t-1}\boldsymbol{x}_t\right)^2} \boldsymbol{x}_t^\top \Sigma_{t-1}\left(\Sigma_{t-1}^{-1} + \frac{1}{r}\boldsymbol{x}_t\boldsymbol{x}_t^\top\right)\Sigma_{t-1}\boldsymbol{x}_t \\ & + \left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right)^\top \Sigma_{t-1}^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{w}_{t-1}\right) \ . \end{aligned}$$

Finally, we substitute $\ell_t = \left(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}\right)^2$, $g_t = \left(y_t - \boldsymbol{x}_t^\top \boldsymbol{u}_{t-1}\right)^2$ and $\chi_t = \boldsymbol{x}_t^\top \Sigma_{t-1} \boldsymbol{x}_t$. Rearranging the terms,

$$\begin{aligned}
&d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_{\tilde{t}}\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) \\
=& -\frac{1}{r}\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1} - \left(y_t - \boldsymbol{x}_t^\top \boldsymbol{u}_{t-1}\right)\right)^2 \\
&- \frac{2\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{u}_{t-1} - \left(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}\right)\right)\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}\right)}{r + \chi_t}\left(1 + \frac{\chi_t}{r}\right) \\
&- \frac{\ell_t \chi_t}{\left(r + \chi_t\right)^2}\left(1 + \frac{\chi_t}{r}\right) \\
=& -\frac{1}{r}\ell_t + 2\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}\right)\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{u}_{t-1}\right)\frac{1}{r} - \frac{1}{r}g_t \\
&+ \frac{2\ell_t}{r + \chi_t}\left(1 + \frac{\chi_t}{r}\right) - \frac{\ell_t \chi_t}{r\left(r + \chi_t\right)} \\
&- 2\frac{\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{w}_{t-1}\right)\left(y_t - \boldsymbol{x}_t^\top \boldsymbol{u}_{t-1}\right)}{r + \chi_t}\left(1 + \frac{\chi_t}{r}\right) \\
=& \frac{1}{r}\ell_t - \frac{1}{r}g_t - \frac{\ell_t \chi_t}{r\left(r + \chi_t\right)} \ ,
\end{aligned}$$

which completes the proof. ■

We now define one element of the telescopic sum and lower bound it.

Lemma 14 *Denote*

$$\Delta_t = d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_t\left(\boldsymbol{w}_t, \boldsymbol{u}_t\right)$$

then

$$\Delta_t \geq \frac{1}{r}\left(\ell_t - g_t\right) - \ell_t \frac{\chi_t}{r(r + \chi_t)} + \boldsymbol{u}_{t-1}^\top \Sigma_{t-1}^{-1} \boldsymbol{u}_{t-1} - \boldsymbol{u}_t^\top \Sigma_t^{-1} \boldsymbol{u}_t - 2R_B \Lambda_i^{-1} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \ ,$$

where $i - 1$ is the number of restarts occurring before example t.

Proof We write Δ_t as a telescopic sum of four terms as follows

$$\begin{aligned}
\Delta_{t,1} &= d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_{\tilde{t}}\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) \\
\Delta_{t,2} &= d_{\tilde{t}}\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) - d_t\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) \\
\Delta_{t,3} &= d_t\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) - d_t\left(\boldsymbol{w}_t, \boldsymbol{u}_{t-1}\right) \\
\Delta_{t,4} &= d_t\left(\boldsymbol{w}_t, \boldsymbol{u}_{t-1}\right) - d_t\left(\boldsymbol{w}_t, \boldsymbol{u}_t\right) \ .
\end{aligned}$$

We lower bound each of the four terms. Since the value of $\Delta_{t,1}$ was computed in Lemma 13, we start with the second term. If no reset occurs then $\Sigma_t = \tilde{\Sigma}_t$ and $\Delta_{t,2} = 0$. Otherwise, we use the facts that $0 \preceq \tilde{\Sigma}_t \preceq I$ and $\Sigma_t = I$, and get

$$\begin{aligned}
\Delta_{t,2} &= \left(\tilde{\boldsymbol{w}}_t - \boldsymbol{u}_{t-1}\right)^\top \tilde{\Sigma}_t^{-1}\left(\tilde{\boldsymbol{w}}_t - \boldsymbol{u}_{t-1}\right) - \left(\tilde{\boldsymbol{w}}_t - \boldsymbol{u}_{t-1}\right)^\top \Sigma_t^{-1}\left(\tilde{\boldsymbol{w}}_t - \boldsymbol{u}_{t-1}\right) \\
&\geq \mathrm{Tr}\left(\left(\tilde{\boldsymbol{w}}_t - \boldsymbol{u}_{t-1}\right)\left(\tilde{\boldsymbol{w}}_t - \boldsymbol{u}_{t-1}\right)^\top \left(I - I\right)\right) = 0 \ .
\end{aligned}$$

To summarize, $\Delta_{t,2} \geq 0$. We can lower bound $\Delta_{t,3} \geq 0$ by using the fact that $\boldsymbol{w}_t$ is a projection of $\tilde{\boldsymbol{w}}_t$ onto a closed set (a ball of radius R_B around the origin), which by our assumption contains $\boldsymbol{u}_t$. Employing Corollary 3 of Herbster and Warmuth (2001) we get, $d_t\left(\tilde{\boldsymbol{w}}_t, \boldsymbol{u}_{t-1}\right) \geq d_t\left(\boldsymbol{w}_t, \boldsymbol{u}_{t-1}\right)$ and thus $\Delta_{t,3} \geq 0$.

Finally, we lower bound the fourth term $\Delta_{t,4}$,

$$\begin{aligned}\Delta_{t,4} &= \left(\boldsymbol{w}_t - \boldsymbol{u}_{t-1}\right)^\top \Sigma_t^{-1}\left(\boldsymbol{w}_t - \boldsymbol{u}_{t-1}\right) - \left(\boldsymbol{w}_t - \boldsymbol{u}_t\right)^\top \Sigma_t^{-1}\left(\boldsymbol{w}_t - \boldsymbol{u}_t\right) \\ &= \boldsymbol{u}_{t-1}^\top \Sigma_t^{-1} \boldsymbol{u}_{t-1} - \boldsymbol{u}_t^\top \Sigma_t^{-1} \boldsymbol{u}_t - 2\boldsymbol{w}_t^\top \Sigma_t^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\right) \ . \end{aligned} \tag{31}$$

We use the Hölder inequality and then the Cauchy-Schwartz inequality to get the following lower bound

$$\begin{aligned}-2\boldsymbol{w}_t^\top \Sigma_t^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\right) &= -2\mathrm{Tr}\left(\Sigma_t^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\right)\boldsymbol{w}_t^\top\right) \\ &\geq -2\lambda_{max}\left(\Sigma_t^{-1}\right)\boldsymbol{w}_t^\top\left(\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\right) \\ &\geq -2\lambda_{max}\left(\Sigma_t^{-1}\right)\|\boldsymbol{w}_t\|\|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \ .\end{aligned}$$

Using the facts that $\|\boldsymbol{w}_t\| \leq R_B$ and that $\lambda_{max}\left(\Sigma_t^{-1}\right) = 1/\lambda_{min}\left(\Sigma_t\right) \leq \Lambda_i^{-1}$, where i is the current segment index, we get

$$-2\boldsymbol{w}_t^\top \Sigma_t^{-1}\left(\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\right) \geq -2\Lambda_i^{-1} R_B \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \ . \tag{32}$$

Substituting (32) in (31) and using $\Sigma_t \preceq \Sigma_{t-1}$ a lower bound is obtained,

$$\begin{aligned}\Delta_{t,4} &\geq \boldsymbol{u}_{t-1}^\top \Sigma_t^{-1} \boldsymbol{u}_{t-1} - \boldsymbol{u}_t^\top \Sigma_t^{-1} \boldsymbol{u}_t - 2R_B\Lambda_i^{-1}\|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \\ &\geq \boldsymbol{u}_{t-1}^\top \Sigma_{t-1}^{-1} \boldsymbol{u}_{t-1} - \boldsymbol{u}_t^\top \Sigma_t^{-1} \boldsymbol{u}_t - 2R_B\Lambda_i^{-1}\|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \ . \end{aligned} \tag{33}$$

Combining (33) with Lemma 13 concludes the proof. ■

Next we state an upper bound that will appear in one of the summands of the telescopic sum.

Lemma 15 *During the runtime of the ARCOR algorithm we have*

$$\sum_{t=t_i}^{t_i+T_i} \frac{\chi_t}{\left(\chi_t + r\right)} \leq \log\left(\det\left(\Sigma_{t_{i+1}-1}^{-1}\right)\right) = \log\left(\det\left(\left(\Sigma^i\right)^{-1}\right)\right) \ .$$

We remind the reader that t_i is the first example index after the ith restart, and T_i is the number of examples observed before the next restart. We also remind the reader the notation $\Sigma^i = \Sigma_{t_{i+1}-1}$ is the covariance matrix just before the next restart.

The proof of the lemma is similar to the proof of Lemma 4 by Crammer et al. (2009) and thus omitted. We now put all the pieces together and prove Theorem 3.

Proof We bound the sum $\sum_t \Delta_t$ from above and below, and start with an upper bound using the property of telescopic sum,

$$\sum_t \Delta_t = \sum_t \left[d_{t-1}\left(\boldsymbol{w}_{t-1}, \boldsymbol{u}_{t-1}\right) - d_t\left(\boldsymbol{w}_t, \boldsymbol{u}_t\right)\right] = d_0\left(\boldsymbol{w}_0, \boldsymbol{u}_0\right) - d_T\left(\boldsymbol{w}_T, \boldsymbol{u}_T\right) \leq d_0\left(\boldsymbol{w}_0, \boldsymbol{u}_0\right) \ . \tag{34}$$

We compute a lower bound by applying Lemma 14,

$$\sum_t \Delta_t \geq \sum_t \left(\frac{1}{r}\left(\ell_t - g_t\right) - \ell_t \frac{\chi_t}{r(r+\chi_t)} \right.$$
$$\left. + \boldsymbol{u}_{t-1}^\top \Sigma_{t-1}^{-1} \boldsymbol{u}_{t-1} - \boldsymbol{u}_t^\top \Sigma_t^{-1} \boldsymbol{u}_t - 2R_B \Lambda_{i(t)}^{-1} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \right) ,$$

where $i(t)$ is the number of restarts occurred before observing the tth example. Continuing to develop the last equation we obtain

$$\begin{aligned}
\sum_t \Delta_t &\geq \frac{1}{r}\sum_t \ell_t - \frac{1}{r}\sum_t g_t - \sum_t \ell_t \frac{\chi_t}{r(r+\chi_t)} + \sum_t \left(\boldsymbol{u}_{t-1}^\top \Sigma_{t-1}^{-1} \boldsymbol{u}_{t-1} - \boldsymbol{u}_t^\top \Sigma_t^{-1} \boldsymbol{u}_t \right) \\
&\quad - \sum_t 2R_B \Lambda_{i(t)}^{-1} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \\
&= \frac{1}{r}\sum_t \ell_t - \frac{1}{r}\sum_t g_t - \sum_t \ell_t \frac{\chi_t}{r(r+\chi_t)} + \boldsymbol{u}_0^\top \Sigma_0^{-1} \boldsymbol{u}_0 - \boldsymbol{u}_T^\top \Sigma_T^{-1} \boldsymbol{u}_T \\
&\quad - 2R_B \sum_t \Lambda_{i(t)}^{-1} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \ . \qquad (35)
\end{aligned}$$

Combining (34) with (35) and using $d_0\left(\boldsymbol{w}_0, \boldsymbol{u}_0\right) = \boldsymbol{u}_0^\top \Sigma_0^{-1} \boldsymbol{u}_0$ (as $\boldsymbol{w}_0 = \boldsymbol{0}$),

$$\frac{1}{r}\sum_t \ell_t - \frac{1}{r}\sum_t g_t - \sum_t \ell_t \frac{\chi_t}{r(r+\chi_t)} - \boldsymbol{u}_T^\top \Sigma_T^{-1} \boldsymbol{u}_T - 2R_B \sum_t \Lambda_{i(t)}^{-1} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \leq 0 \ .$$

Rearranging the terms of the last inequality,

$$\sum_t \ell_t \leq \sum_t g_t + \sum_t \ell_t \frac{\chi_t}{r+\chi_t} + r \boldsymbol{u}_T^\top \Sigma_T^{-1} \boldsymbol{u}_T + 2R_B r \sum_t \frac{1}{\Lambda_{i(t)}} \|\boldsymbol{u}_{t-1} - \boldsymbol{u}_t\| \ . \qquad (36)$$

Since $\|\boldsymbol{w}_t\| \leq R_B$ and we assume that $\|\boldsymbol{x}_t\| = 1$ and $\sup_t |y_t| = Y$, we get that $\sup_t \ell_t \leq 2(R_B^2 + Y^2)$. Substituting the last inequality in Lemma 15, we bound the second term in the right-hand-side of (36),

$$\begin{aligned}
\sum_t \ell_t \frac{\chi_t}{r+\chi_t} &= \sum_i^n \sum_{t=t_i}^{t_i+T_i} \ell_t \frac{\chi_t}{r+\chi_t} \\
&\leq \sum_i^n \left(\sup_t \ell_t \right) \log\det\left(\left(\Sigma^i\right)^{-1} \right) \\
&\leq 2\left(R_B^2 + Y^2\right) \sum_i^n \log\det\left(\left(\Sigma^i\right)^{-1} \right) \ ,
\end{aligned}$$

which completes the proof. ■

Appendix D. Proof of Lemma 7

Proof We first use the Woodbury identity to get the following two identities

$$\begin{aligned} D_t^{-1} &= \left[\left(D_{t-1}^{-1} + c^{-1}I\right)^{-1} + \boldsymbol{x}_t\boldsymbol{x}_t^\top\right]^{-1} \\ &= D_{t-1}^{-1} + c^{-1}I - \frac{\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t} \\ \left(I + c^{-1}D_{t-1}\right)^{-1} &= I - c^{-1}\left(D_{t-1}^{-1} + c^{-1}I\right)^{-1} \ . \end{aligned}$$

Multiplying both identities with each other we get

$$\begin{aligned} & D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1} \\ = \ & \left[D_{t-1}^{-1} + c^{-1}I - \frac{\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t}\right]\left[I - c^{-1}\left(D_{t-1}^{-1} + c^{-1}I\right)^{-1}\right] \\ = \ & D_{t-1}^{-1} - \frac{\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\boldsymbol{x}_t^\top D_{t-1}^{-1}}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t} \ , \end{aligned} \tag{37}$$

and, similarly, we multiply the identities in the other order and get

$$\begin{aligned} & \left(I + c^{-1}D_{t-1}\right)^{-1} D_t^{-1} \\ = \ & \left[I - c^{-1}\left(D_{t-1}^{-1} + c^{-1}I\right)^{-1}\right]\left[D_{t-1}^{-1} + c^{-1}I - \frac{\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t}\right] \\ = \ & D_{t-1}^{-1} - \frac{D_{t-1}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t} \ . \end{aligned} \tag{38}$$

Finally, from (37) we get

$$\begin{aligned} & \left(I + c^{-1}D_{t-1}\right)^{-1} D_t^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1} - D_{t-1}^{-1} \\ & \quad + \left(I + c^{-1}D_{t-1}\right)^{-1}\left[D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1} + c^{-1}I\right] \\ = \ & \left(I + c^{-1}D_{t-1}\right)^{-1} D_t^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1} \\ & \quad - D_{t-1}^{-1} + \left[I - c^{-1}\left(D_{t-1}^{-1} + c^{-1}I\right)^{-1}\right]\left[D_{t-1}^{-1} + c^{-1}I - \frac{\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\boldsymbol{x}_t^\top D_{t-1}^{-1}}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t}\right] . \end{aligned}$$

We further develop the last equality and use (37) and (38) in the second equality below,

$$
\begin{aligned}
&= \left(I + c^{-1}D_{t-1}\right)^{-1} D_t^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1} - D_{t-1}^{-1} \\
&\quad + D_{t-1}^{-1} - \frac{D_{t-1}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top D_{t-1}^{-1}}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t} \\
&= \left[D_{t-1}^{-1} - \frac{D_{t-1}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t}\right]\boldsymbol{x}_t\boldsymbol{x}_t^\top\left[D_{t-1}^{-1} - \frac{\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\boldsymbol{x}_t^\top D_{t-1}^{-1}}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t}\right] \\
&\quad - \frac{D_{t-1}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top D_{t-1}^{-1}}{1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t} \\
&= -\frac{\boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t D_{t-1}^{-1}\boldsymbol{x}_t\boldsymbol{x}_t^\top D_{t-1}^{-1}}{\left(1 + \boldsymbol{x}_t^\top\left(D_{t-1}^{-1} + c^{-1}I\right)\boldsymbol{x}_t\right)^2} \quad \preceq \quad 0\ .
\end{aligned}
$$

■

Appendix E. Derivations for Theorem 8

$$
\begin{aligned}
&\left(y_t - \hat{y}_t\right)^2 + \min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{t-1}} Q_{t-1}\left(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_{t-1}\right) - \min_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_t} Q_t\left(\boldsymbol{u}_1,\ldots,\boldsymbol{u}_t\right) \\
&= \left(y_t - \hat{y}_t\right)^2 - \boldsymbol{e}_{t-1}^\top D_{t-1}^{-1}\boldsymbol{e}_{t-1} + f_{t-1} + \boldsymbol{e}_t^\top D_t^{-1}\boldsymbol{e}_t - f_t \\
&= \left(y_t - \hat{y}_t\right)^2 - \boldsymbol{e}_{t-1}^\top D_{t-1}^{-1}\boldsymbol{e}_{t-1} \\
&\quad + \left(\left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t\boldsymbol{x}_t\right)^\top D_t^{-1}\left(\left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t\boldsymbol{x}_t\right) \\
&\quad + \boldsymbol{e}_{t-1}^\top\left(cI + D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} - y_t^2\ ,
\end{aligned}
$$

where the last equality follows from (12) and (13). We proceed to develop the last equality,

$$
\begin{aligned}
&= \left(y_t - \hat{y}_t\right)^2 - \boldsymbol{e}_{t-1}^\top D_{t-1}^{-1}\boldsymbol{e}_{t-1} + \boldsymbol{e}_{t-1}^\top\left(I + c^{-1}D_{t-1}\right)^{-1} D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} \\
&\quad + 2y_t\boldsymbol{x}_t^\top D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t^2\boldsymbol{x}_t^\top D_t^{-1}\boldsymbol{x}_t + \boldsymbol{e}_{t-1}^\top\left(cI + D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} - y_t^2 \\
&= \left(y_t - \hat{y}_t\right)^2 + \boldsymbol{e}_{t-1}^\top\Bigg(-D_{t-1}^{-1} + \left(I + c^{-1}D_{t-1}\right)^{-1} D_t^{-1}\Big(I + c^{-1}D_{t-1}\Big)^{-1} \\
&\quad + c^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1}\Bigg)\boldsymbol{e}_{t-1} \\
&\quad + 2y_t\boldsymbol{x}_t^\top D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t^2\boldsymbol{x}_t^\top D_t^{-1}\boldsymbol{x}_t - y_t^2 \\
&= \left(y_t - \hat{y}_t\right)^2 + \boldsymbol{e}_{t-1}^\top\left(-D_{t-1}^{-1} + \Big(I + c^{-1}D_{t-1}\Big)^{-1}\left[D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1} + c^{-1}I\right]\right)\boldsymbol{e}_{t-1} \\
&\quad + 2y_t\boldsymbol{x}_t^\top D_t^{-1}\left(I + c^{-1}D_{t-1}\right)^{-1}\boldsymbol{e}_{t-1} + y_t^2\boldsymbol{x}_t^\top D_t^{-1}\boldsymbol{x}_t - y_t^2\ .
\end{aligned}
$$

Appendix F. Proof of Corollary 12

Proof Plugging Lemma 9 in Theorem 8 we have for all $(\boldsymbol{u}_1 \dots \boldsymbol{u}_T)$,

$$\begin{aligned} L_T(\text{LASER}) \leq\ & b\left\|\boldsymbol{u}_1\right\|^2 + cV^{(2)} + L_T(\{\boldsymbol{u}_t\}) + Y^2 \ln\left|\frac{1}{b}D_T\right| + c^{-1}Y^2\sum_{t=1}^{T}\mathrm{Tr}\left(D_{t-1}\right) \\ \leq\ & b\left\|\boldsymbol{u}_1\right\|^2 + L_T(\{\boldsymbol{u}_t\}) + Y^2 \ln\left|\frac{1}{b}D_T\right| + c^{-1}Y^2\mathrm{Tr}\left(D_0\right) + cV^{(2)} \\ & + c^{-1}Y^2 T d \max\left\{\frac{3X^2+\sqrt{X^4+4X^2c}}{2}, b+X^2\right\}\ , \end{aligned}$$

where the last inequality follows from Lemma 11. The term $c^{-1}Y^2\mathrm{Tr}\left(D_0\right)$ does not depend on T, because

$$c^{-1}Y^2\mathrm{Tr}\left(D_0\right) = c^{-1}Y^2 d\frac{bc}{c-b} = \frac{\varepsilon}{1-\varepsilon}Y^2 d\ .$$

To show (29), note that

$$V^{(2)} \leq T\frac{\sqrt{2}Y^2 dX}{\mu^{3/2}} \Leftrightarrow \mu \leq \left(\frac{\sqrt{2}Y^2 dXT}{V^{(2)}}\right)^{2/3} = c\ .$$

We thus have that the right term of (27) is upper bounded,

$$\begin{aligned} & \max\left\{\frac{3X^2+\sqrt{X^4+4X^2c}}{2}, b+X^2\right\} \leq\ \max\left\{\frac{3X^2+\sqrt{8X^2c}}{2}, b+X^2\right\} \\ \leq\ & \max\left\{\sqrt{8X^2c}, b+X^2\right\} \leq 2X\sqrt{2c}\ . \end{aligned}$$

Using this bound and plugging the value of c from (28) we bound (27),

$$\begin{aligned} & \left(\frac{\sqrt{2}TY^2 dX}{V^{(2)}}\right)^{2/3} V^{(2)} + Y^2 T d 2X\sqrt{2\left(\frac{\sqrt{2}TY^2 dX}{V^{(2)}}\right)^{-2/3}} \\ & = 3\left(\sqrt{2}TY^2 dX\right)^{2/3}\left(V^{(2)}\right)^{1/3}\ , \end{aligned}$$

which concludes the proof. ■

Appendix G. Details for the bound (30)

To show the bound (30), note that

$$V^{(2)} \geq T\frac{Y^2 dM}{\mu^2} \Leftrightarrow \mu \geq \sqrt{\frac{TY^2 dM}{V^{(2)}}} = c\ .$$

We thus have that the right term of (27) is upper bounded as follows

$$\max\left\{\frac{3X^2+\sqrt{X^4+4X^2c}}{2}, b+X^2\right\} \leq \max\left\{3X^2, \sqrt{X^4+4X^2c}, b+X^2\right\}$$
$$\leq \max\left\{3X^2, \sqrt{2}X^2, \sqrt{8X^2c}, b+X^2\right\} = \sqrt{8X^2}\max\left\{\frac{3X^2}{\sqrt{8X^2}}, \sqrt{c}, \frac{b+X^2}{\sqrt{8X^2}}\right\}$$
$$= \sqrt{8X^2}\sqrt{\max\left\{\frac{(3X^2)^2}{8X^2}, c, \frac{(b+X^2)^2}{8X^2}\right\}} = \sqrt{8X^2}\sqrt{\max\{\mu, c\}} \leq \sqrt{8X^2}\sqrt{\mu} = M \ .$$

Using this bound and plugging $c = \sqrt{Y^2 dMT/V^{(2)}}$ we bound (27),

$$\sqrt{\frac{Y^2 dMT}{V^{(2)}}}V^{(2)} + \frac{1}{\sqrt{\frac{Y^2 dMT}{V^{(2)}}}}TdY^2M = 2\sqrt{Y^2 dMTV^{(2)}} \ .$$

References

Dmitry Adamskiy, Wouter M. Koolen, Alexey V. Chernov, and Vladimir Vovk. A closer look at adaptive regret. In *The 23rd International Conference on Algorithmic Learning Theory*, pages 290–304, 2012.

Peter Auer and Manfred K. Warmuth. Tracking the best disjunction. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(70), 2000.

K.S. Azoury and M.W. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

N. J. Bershad. Analysis of the normalized lms algorithm with gaussian inputs. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):793–806, 1986.

R. R. Bitmead and B. D. O. Anderson. Performance of adaptive estimation algorithms in dependent random environments. *IEEE Transactions on Automatic Control*, 25:788–794, 1980.

Steven Busuttil and Yuri Kalnishkan. Online regression competitive with changing predictors. In *The 18th International Conference on Algorithmic Learning Theory*, pages 181–195, 2007.

Giovanni Cavallanti, Nicolò Cesa-Bianchi, and Claudio Gentile. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 69(2-3):143–167, 2007.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

Nicolo Cesa-Bianchi, Philip M. Long, and Manfred K. Warmuth. Worst case quadratic loss bounds for on-line prediction of linear functions by gradient descent. Technical Report IR-418, University of California, Santa Cruz, CA, USA, 1993.

Nicoló Cesa-Bianchi, Alex Conconi, and Claudio Gentile. A second-order perceptron algorithm. *Siam Journal of Commutation*, 34(3):640–668, 2005.

Min-Shin Chen and Jia-Yush Yen. Application of the least squares algorithm to the observer design for linear time-varying systems. *Automatic Control, IEEE Transactions on*, 44(9): 1742 –1745, sep 1999.

K. Crammer, M. Dredze, and F. Pereira. Exact confidence-weighted learning. In *Advances in Neural Information Processing Systems 22*, 2008.

K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weighted vectors. In *Advances in Neural Information Processing Systems 23*, 2009.

M. Dredze, K. Crammer, and F. Pereira. Confidence-weighted linear classification. In *Proceedings of the Twenty-Five International Conference on Machine Learning*, 2008.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. In *Proceedings of the 23rd Annual Conference on Computational Learning Theory*, pages 257–269, 2010.

A. Feuer and E. Weinstein. Convergence analysis of lms filters with uncorrelated gaussian data. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 33(1):222–230, 1985.

Jurgen Forster. On relative loss bounds in generalized linear regression. In *Fundamentals of Computation Theory (FCT)*, 1999. ISBN 3-540-66412-2.

Dean P. Foster. Prediction in the worst case. *The Annals of Statistics*, 19(2):1084–1090, 1991.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996. ISBN 0-8018-5414-8.

S.G. Goodhart, K.J. Burnham, and D.J.G. James. Logical covariance matrix reset in self-tuning control. *Mechatronics*, 1(3):339 – 351, 1991.

G.C. Goodwin, E.K. Teoh, and H. Elliott. Deterministic convergence of a self-tuning regulator with covariance resetting. *Control Theory and Applications, IEE Proceedings D*, 130(1):6 –8, 83.

Monson H. Hayes. 9.4: Recursive least squares. In *Statistical Digital Signal Processing and Modeling*, page 541, 1996. ISBN 0-471-59431-8.

Mark Herbster and Manfred K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

Jyrki Kivinen and Manfred K. Warmuth. Exponential gradient versus gradient descent for linear predictors. *Information and Computation*, 132:132–163, 1997.

Jyrki Kivinen, Alex J. Smola, and Robert C. Williamson. Online learning with kernels. In *Advances in Neural Information Processing Systems 14*, pages 785–792, 2001.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

H. Brendan McMahan and Matthew J. Streeter. Adaptive bound optimization for online convex optimization. In *Proceedings of the 23rd Annual Conference on Computational Learning Theory*, pages 244–256, 2010.

Edward Moroshko and Koby Crammer. Weighted last-step min-max algorithm with improved sub-logarithmic regret. In *The 23rd International Conference on Algorithmic Learning Theory*, 2012.

Edward Moroshko and Koby Crammer. A last-step regression algorithm for non-stationary online learning. In *The Sixteenth International Conference on Artificial Intelligence and Statistics*, 2013.

Mario E. Salgado, Graham C. Goodwin, and Richard H. Middleton. Modified least squares algorithm incorporating exponential resetting and forgetting. *International Journal of Control*, 47(2):477 –491, 1988.

Dan Simon. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches.* Wiley-Interscience, 2006. ISBN 0471708585.

Hong-Seok Song, Kwanghee Nam, and P. Mutschler. Very fast phase angle estimation algorithm for a single-phase system having sudden phase angle jumps. In *Industry Applications Conference. 37th IAS Annual Meeting*, volume 2, pages 925 – 931, 2002.

Eiji Takimoto and Manfred K. Warmuth. The last-step minimax algorithm. In *The 11th International Conference on Algorithmic Learning Theory*, pages 279–290, 2000.

Nina Vaits and Koby Crammer. Re-adapting the regularization of weights for non-stationary regression. In *The 22nd International Conference on Algorithmic Learning Theory*, 2011.

Volodimir Vovk. Aggregating strategies. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 371–383. Morgan Kaufmann, 1990.

Volodya Vovk. Competitive on-line linear regression. In *Advances in Neural Information Processing Systems 10*, 1997.

Volodya Vovk. Competitive on-line statistics. *International Statistical Review*, 69, 2001.

Bernard Widrow and Marcian E. Hoff. Adaptive switching circuits. In *IRE WESCON Convention Record, Part 4*, pages 96–104, 1960.

Journal of Machine Learning Research 16 (2015) 1519-1545 Submitted 8/14; Revised 11/14; Published 8/15

A Finite Sample Analysis of the Naive Bayes Classifier*

Daniel Berend BEREND@CS.BGU.AC.IL
Department of Computer Science and Department of Mathematics
Ben-Gurion University
Beer Sheva, Israel

Aryeh Kontorovich KARYEH@CS.BGU.AC.IL
Department of Computer Science
Ben-Gurion University
Beer Sheva, Israel

Editor: Gabor Lugosi

Abstract

We revisit, from a statistical learning perspective, the classical decision-theoretic problem of weighted expert voting. In particular, we examine the consistency (both asymptotic and finitary) of the optimal Naive Bayes weighted majority and related rules. In the case of known expert competence levels, we give sharp error estimates for the optimal rule. We derive optimality results for our estimates and also establish some structural characterizations. When the competence levels are unknown, they must be empirically estimated. We provide frequentist and Bayesian analyses for this situation. Some of our proof techniques are non-standard and may be of independent interest. Several challenging open problems are posed, and experimental results are provided to illustrate the theory.

Keywords: experts, hypothesis testing, Chernoff-Stein lemma, Neyman-Pearson lemma, naive Bayes, measure concentration

1. Introduction

Imagine independently consulting a small set of medical experts for the purpose of reaching a binary decision (e.g., whether to perform some operation). Each doctor has some "reputation", which can be modeled as his probability of giving the right advice. The problem of weighting the input of several experts arises in many situations and is of considerable theoretical and practical importance. The rigorous study of majority vote has its roots in the work of Condorcet (1785). By the 70s, the field of decision theory was actively exploring various voting rules (see Nitzan and Paroush (1982) and the references therein). A typical setting is as follows. An agent is tasked with predicting some random variable $Y \in \{\pm 1\}$ based on input $X_i \in \{\pm 1\}$ from each of n experts. Each expert X_i has a *competence* level $p_i \in (0,1)$, which is his probability of making a correct prediction: $\mathbb{P}(X_i = Y) = p_i$. Two simplifying assumptions are commonly made:

*. An extended abstract of this paper appeared in NIPS 2014 under the title "Consistency of weighted majority votes," which was also the former title of this paper. A. Kontorovich was partially supported by the Israel Science Foundation (grant No. 1141/12) and a Yahoo Faculty award.

(i) *Independence*: The random variables $\{X_i : i \in [n]\}$ are mutually independent conditioned on the truth Y.

(ii) *Unbiased truth*: $\mathbb{P}(Y = +1) = \mathbb{P}(Y = -1) = 1/2$.

We will discuss these assumptions below in greater detail; for now, let us just take them as given. (Since the bias of Y can be easily estimated from data, and the generalization to the asymmetric case is straightforward, only the independence assumption is truly restrictive.) A *decision rule* is a mapping $f : \{\pm 1\}^n \to \{\pm 1\}$ from the n expert inputs to the agent's final decision. Our quantity of interest throughout the paper will be the agent's probability of error,

$$\mathbb{P}(f(\mathbf{X}) \neq Y). \tag{1}$$

A decision rule f is *optimal* if it minimizes the quantity in (1) over all possible decision rules. It follows from the work of Neyman and Pearson (1933) that, when Assumptions (i)–(ii) hold and the true competences p_i are known, the optimal decision rule is obtained by an appropriately weighted majority vote:

$$f^{\mathrm{OPT}}(\mathbf{x}) = \operatorname{sign}\left(\sum_{i=1}^{n} w_i x_i\right), \tag{2}$$

where the weights w_i are given by

$$w_i = \log \frac{p_i}{1 - p_i}, \qquad i \in [n]. \tag{3}$$

Thus, w_i is the log-odds of expert i being correct, and the voting rule in (2) is also known as *naive Bayes* (Hastie et al., 2009).

Main results. Formula (2) raises immediate questions, which apparently have not previously been addressed. The first one has to do with the *consistency* of the naive Bayes decision rule: under what conditions does the probability of error decay to zero and at what rate? In Section 3, we show that the probability of error is controlled by the *committee potential* Φ, defined by

$$\Phi = \sum_{i=1}^{n} (p_i - \tfrac{1}{2}) w_i = \sum_{i=1}^{n} (p_i - \tfrac{1}{2}) \log \frac{p_i}{1 - p_i}. \tag{4}$$

More precisely, we prove in Theorem 1 that

$$-\log \mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y) \quad \asymp \quad \Phi,$$

where $\asymp$ denotes equivalence up to universal multiplicative constants. As we show in Section 3.3, both the upper estimate of $O(e^{-\Phi/2})$ and the lower one of $\Omega(e^{-2\Phi})$ are tight in various regimes of Φ. The structural characterization in terms of "antipodes" (Lemma 2) and the additional bounds provided in Section 3.4 may also be of interest.

Another issue not addressed by the Neyman-Pearson lemma is how to handle the case where the competences p_i are not known exactly but rather estimated empirically by $\hat{p}_i$. We

present two solutions to this problem: a frequentist and a Bayesian one. As we show in Section 4, the frequentist approach does not admit an optimal empirical decision rule. Instead, we analyze empirical decision rules in various settings: high-confidence (i.e., $|\hat{p}_i - p_i| \ll 1$) vs. low-confidence, adaptive vs. nonadaptive. The low-confidence regime requires no additional assumptions, but gives weaker guarantees (Theorem 7). In the high-confidence regime, the adaptive approach produces error estimates in terms of the empirical $\hat{p}_i$s (Theorem 13), while the nonadaptive approach yields a bound in terms of the unknown p_is, which still leads to useful asymptotics (Theorem 11). The Bayesian solution sidesteps the various cases above, as it admits a simple, provably optimal empirical decision rule (Section 5). Unfortunately, we are unable to compute (or even nontrivially estimate) the probability of error induced by this rule; this is posed as a challenging open problem.

Notation. We use standard set-theoretic notation, and in particular $[n] = \{1, \ldots, n\}$.

2. Related Work

The Naive Bayes weighted majority voting rule was stated by Nitzan and Paroush (1982) in the context of decision theory, but its roots trace much earlier to the problem of hypothesis testing (Neyman and Pearson, 1933). Machine learning theory typically clusters *weighted majority* (Littlestone and Warmuth, 1989, 1994) within the framework of online algorithms; see Cesa-Bianchi and Lugosi (2006) for a modern treatment. Since the online setting is considerably more adversarial than ours, we obtain very different weighted majority rules and consistency guarantees. The weights w_i in (2) bear a striking similarity to the AdaBoost update rule (Freund and Schapire, 1997; Schapire and Freund, 2012). However, the latter assumes weak learners with access to labeled examples, while in our setting the experts are "static". Still, we do not rule out a possible deeper connection between the Naive Bayes decision rule and boosting.

In what began as the influential Dawid-Skene model (Dawid and Skene, 1979) and is now known as *crowdsourcing*, one attempts to extract accurate predictions by pooling a large number of experts, typically without the benefit of being able to test any given expert's competence level. Still, under mild assumptions it is possible to efficiently recover the expert competences to a high accuracy and to aggregate them effectively (Parisi et al., 2014+). Error bounds for the oracle MAP rule were obtained in this model by Li et al. (2013) and minimax rates were given in Gao and Zhou (2014).

In a recent line of work, Lacasse et al. (2006); Laviolette and Marchand (2007); Roy et al. (2011) have developed a PAC-Bayesian theory for the majority vote of simple classifiers. This approach facilitates data-dependent bounds and is even flexible enough to capture some simple dependencies among the classifiers — though, again, the latter are *learners* as opposed to our *experts*. Even more recently, experts with adversarial noise have been considered (Mansour et al., 2013), and efficient algorithms for computing optimal expert weights (without error analysis) were given (Eban et al., 2014). More directly related to the present work are the papers of Berend and Paroush (1998), which characterizes the conditions for the consistency of the simple majority rule, and Boland et al. (1989); Berend and Sapir (2007); Helmbold and Long (2012) which analyze various models of dependence among the experts.

3. Known Competences

In this section we assume that the expert competences p_i are known and analyze the consistency of the naive Bayes decision rule (2). Our main result here is that the probability of error $\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y)$ is small if and only if the committee potential Φ is large.

Theorem 1 *Suppose that the experts $\mathbf{X} = (X_1, \ldots, X_n)$ satisfy Assumptions (i)-(ii) and $f^{\text{OPT}} : \{\pm 1\}^n \to \{\pm 1\}$ is the naive Bayes decision rule in (2). Then*

(i) $\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) \leq \exp\left(-\frac{1}{2}\Phi\right)$.

(ii) $\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) \geq \dfrac{3}{4[1+\exp(2\Phi + 4\sqrt{\Phi})]}$.

The next two sections are devoted to proving Theorem 1. These are followed by an optimality result and some additional upper and lower bounds.

3.1 Proof of Theorem 1(i)

Define the $\{0,1\}$-indicator variables

$$\xi_i = \mathbb{1}_{\{X_i = Y\}}, \tag{5}$$

corresponding to the event that the i^{th} expert is correct. A mistake $f^{\text{OPT}}(\mathbf{X}) \neq Y$ occurs precisely when[1] the sum of the correct experts' weights fails to exceed half the total mass:

$$\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) = \mathbb{P}\left(\sum_{i=1}^n w_i \xi_i \leq \frac{1}{2}\sum_{i=1}^n w_i\right). \tag{6}$$

Since $\mathbb{E}\xi_i = p_i$, we may rewrite the probability in (6) as

$$\mathbb{P}\left(\sum_i w_i \xi_i \leq \mathbb{E}\left[\sum_i w_i \xi_i\right] - \sum_i (p_i - \tfrac{1}{2}) w_i\right). \tag{7}$$

A standard tool for estimating such sum deviation probabilities is Hoeffding's inequality (Hoeffding, 1963). Applied to (7), it yields the bound

$$\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) \leq \exp\left(-\frac{2\left[\sum_i (p_i - \frac{1}{2}) w_i\right]^2}{\sum_i w_i^2}\right), \tag{8}$$

which is far too crude for our purposes. Indeed, consider a finite committee of highly competent experts with p_i's arbitrarily close to 1 and X_1 the most competent of all. Raising X_1's competence sufficiently far above his peers will cause both the numerator and the denominator in the exponent to be dominated by w_1^2, making the right-hand-side of (8) bounded away from zero. In the limiting case of this regime, the probability of error approaches zero while the right-hand side of (8) approaches $e^{-1/2} \approx 0.6$. The inability of Hoeffding's inequality to guarantee consistency even in such a felicitous setting is an instance

1. Without loss of generality, ties are considered to be errors.

of its generally poor applicability to highly heterogeneous sums, a phenomenon explored in some depth in McAllester and Ortiz (2003). Bernstein's and Bennett's inequalities suffer from a similar weakness (see ibid.). Fortunately, an inequality of Kearns and Saul (1998) is sufficiently sharp[2] to yield the desired estimate: For all $p \in [0,1]$ and all $t \in \mathbb{R}$,

$$(1-p)e^{-tp} + pe^{t(1-p)} \leq \exp\left(\frac{1-2p}{4\log((1-p)/p)}t^2\right). \tag{9}$$

Put $\theta_i = \xi_i - p_i$, substitute into (6), and apply Markov's inequality:

$$\begin{aligned} \mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y) &= \mathbb{P}\left(-\sum_i w_i\theta_i \geq \Phi\right) \\ &\leq e^{-t\Phi}\mathbb{E}\exp\left(-t\sum_i w_i\theta_i\right). \end{aligned} \tag{10}$$

Now

$$\begin{aligned} \mathbb{E}e^{-tw_i\theta_i} &= p_i e^{-(1-p_i)w_i t} + (1-p_i)e^{p_i w_i t} \\ &\leq \exp\left(\frac{-1+2p_i}{4\log(p_i/(1-p_i))}w_i^2 t^2\right) \\ &= \exp\left[\tfrac{1}{2}(p_i - \tfrac{1}{2})w_i t^2\right], \end{aligned} \tag{11}$$

where the inequality follows from (9). By independence,

$$\begin{aligned} \mathbb{E}\exp\left(-t\sum_i w_i\theta_i\right) &= \prod_i \mathbb{E}e^{-tw_i\theta_i} \\ &\leq \exp\left(\tfrac{1}{2}\sum_i (p_i - \tfrac{1}{2})w_i t^2\right) \\ &= \exp\left(\tfrac{1}{2}\Phi t^2\right) \end{aligned}$$

and hence

$$\mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y) \leq \exp\left(\tfrac{1}{2}\Phi t^2 - \Phi t\right).$$

Choosing $t = 1$, we obtain the bound in Theorem 1(i).

3.2 Proof of Theorem 1(ii)

Define the $\{\pm 1\}$-indicator variables

$$\eta_i = 2 \cdot \mathbb{1}_{\{X_i = Y\}} - 1, \tag{12}$$

corresponding to the event that the i^{th} expert is correct, and put $q_i = 1 - p_i$. The shorthand $\mathbf{w} \cdot \boldsymbol{\eta} = \sum_{i=1}^n w_i\eta_i$ will be convenient. We will need some simple lemmata:

2. The Kearns-Saul inequality (9) may be seen as a distribution-dependent refinement of Hoeffding's for a two-valued distribution (which bounds the left-hand-side of (9) by $e^{t^2/8}$), and is not nearly as straightforward to prove; see Appendix A.

Lemma 2

$$\begin{aligned}\mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) = Y) &= \tfrac{1}{2} \sum_{\boldsymbol{\eta} \in \{\pm 1\}^n} \max\left\{P(\boldsymbol{\eta}), P(-\boldsymbol{\eta})\right\} \\ &= \sum_{\boldsymbol{\eta} \in \{+1\} \times \{\pm 1\}^{n-1}} \max\left\{P(\boldsymbol{\eta}), P(-\boldsymbol{\eta})\right\}\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y) &= \tfrac{1}{2} \sum_{\boldsymbol{\eta} \in \{\pm 1\}^n} \min\left\{P(\boldsymbol{\eta}), P(-\boldsymbol{\eta})\right\} \\ &= \sum_{\boldsymbol{\eta} \in \{+1\} \times \{\pm 1\}^{n-1}} \min\left\{P(\boldsymbol{\eta}), P(-\boldsymbol{\eta})\right\},\end{aligned}$$

where

$$P(\boldsymbol{\eta}) = \prod_{i:\eta_i=1} p_i \prod_{i:\eta_i=-1} q_i. \tag{13}$$

Proof By (5), (6) and (12), that a mistake occurs precisely when

$$\sum_{i=1}^n w_i \frac{\eta_i + 1}{2} \leq \frac{1}{2} \sum_{i=1}^n w_i,$$

which is equivalent to

$$\mathbf{w} \cdot \boldsymbol{\eta} \leq 0. \tag{14}$$

Exponentiating both sides,

$$\begin{aligned}\exp\left(\mathbf{w} \cdot \boldsymbol{\eta}\right) &= \prod_{i=1}^n e^{w_i \eta_i} \\ &= \prod_{i:\eta_i=1} \frac{p_i}{q_i} \cdot \prod_{i:\eta_i=-1} \frac{q_i}{p_i} \\ &= \frac{P(\boldsymbol{\eta})}{P(-\boldsymbol{\eta})} \;\leq\; 1. \end{aligned} \tag{15}$$

We conclude from (15) that among two "antipodal" atoms $\pm\boldsymbol{\eta} \in \{\pm 1\}^n$, the one with the greater mass contributes to the probability of being correct and the one with the smaller mass contributes to the probability of error, which proves the claim. ■

Lemma 3 *Suppose that* $\mathbf{s}, \mathbf{s}' \in (0, \infty)^m$ *satisfy*

$$\sum_{i=1}^m (s_i + s_i') \geq a$$

and

$$\frac{1}{R} \leq \frac{s_i}{s_i'} \leq R, \qquad i \in [m]$$

for some $1 \leq R < \infty$. *Then*

$$\sum_{i=1}^{m} \min\left\{s_i, s_i'\right\} \geq \frac{a}{1+R}.$$

Proof Immediate from

$$s_i + s_i' \leq \min\left\{s_i, s_i'\right\}(1+R).$$

■

Lemma 4 *Define the function* $F : (0,1) \to \mathbb{R}$ *by*

$$F(x) = \frac{x(1-x)\log(x/(1-x))}{2x-1}.$$

Then $\sup_{0<x<1} F(x) = \frac{1}{2}$.

Proof Since F is symmetric about $x = \frac{1}{2}$, it suffices to prove the claim for $\frac{1}{2} \leq x < 1$. We will show that F is concave by examining its second derivative:

$$F''(x) = -\frac{2x - 1 - 2x(1-x)\log(x/(1-x))}{x(1-x)(2x-1)^3}.$$

The denominator is obviously nonnegative on $[\frac{1}{2}, 1]$, while the numerator has the Taylor expansion

$$\sum_{n=1}^{\infty} \frac{2^{2(n+1)}(x-\frac{1}{2})^{2n+1}}{4n^2-1} \geq 0, \qquad \frac{1}{2} \leq x < 1$$

(verified through tedious but straightforward calculus). Since F is concave and symmetric about $\frac{1}{2}$, its maximum occurs at $F(\frac{1}{2}) = \frac{1}{2}$. ■

Continuing with the main proof, observe that

$$\mathbb{E}\left[\mathbf{w} \cdot \boldsymbol{\eta}\right] = \sum_{i=1}^{n} (p_i - q_i) w_i = 2\Phi \tag{16}$$

and

$$\mathrm{Var}\left[\mathbf{w} \cdot \boldsymbol{\eta}\right] = 4 \sum_{i=1}^{n} p_i q_i w_i^2.$$

By Lemma 4,

$$p_i q_i w_i^2 \leq \tfrac{1}{2}(p_i - q_i) w_i,$$

and hence

$$\mathrm{Var}\left[\mathbf{w} \cdot \boldsymbol{\eta}\right] \;\; \leq \;\; 4\Phi. \tag{17}$$

Define the segments $I, J \subset \mathbb{R}$ by

$$I = \left[2\Phi - 4\sqrt{\Phi}, 2\Phi + 4\sqrt{\Phi}\right] \subset \left[-2\Phi - 4\sqrt{\Phi}, 2\Phi + 4\sqrt{\Phi}\right] = J. \tag{18}$$

Chebyshev's inequality together with (16, 17, 18) implies that

$$\mathbb{P}\left(\mathbf{w} \cdot \boldsymbol{\eta} \in J\right) \;\geq\; \mathbb{P}\left(\mathbf{w} \cdot \boldsymbol{\eta} \in I\right) \;\geq\; \frac{3}{4}. \tag{19}$$

Consider an atom $\boldsymbol{\eta} \in \{\pm 1\}^n$ for which $\mathbf{w} \cdot \boldsymbol{\eta} \in J$. It follows from (15) and (18) that

$$\frac{P(\boldsymbol{\eta})}{P(-\boldsymbol{\eta})} = \exp\left(\mathbf{w} \cdot \boldsymbol{\eta}\right) \leq \exp(2\Phi + 4\sqrt{\Phi}). \tag{20}$$

Finally, we have

$$\begin{aligned}
\mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y) \;\; &\overset{(a)}{=} \sum_{\boldsymbol{\eta} \in \{+1\} \times \{\pm 1\}^{n-1}} \min\left\{P(\boldsymbol{\eta}), P(-\boldsymbol{\eta})\right\} \\
&\geq \sum_{\boldsymbol{\eta} \in \{+1\} \times \{\pm 1\}^{n-1} : \mathbf{w} \cdot \boldsymbol{\eta} \in J} \min\left\{P(\boldsymbol{\eta}), P(-\boldsymbol{\eta})\right\} \\
&\overset{(b)}{\geq} \frac{1}{1 + \exp(2\Phi + 4\sqrt{\Phi})} \sum_{\boldsymbol{\eta} \in \{+1\} \times \{\pm 1\}^{n-1} : \mathbf{w} \cdot \boldsymbol{\eta} \in J} \left(P(\boldsymbol{\eta}) + P(-\boldsymbol{\eta})\right) \\
&\overset{(c)}{=} \frac{1}{1 + \exp(2\Phi + 4\sqrt{\Phi})} \sum_{\boldsymbol{\eta} \in \{\pm 1\}^{n} : \mathbf{w} \cdot \boldsymbol{\eta} \in J} P(\boldsymbol{\eta}) \\
&\overset{(d)}{\geq} \frac{3/4}{1 + \exp(2\Phi + 4\sqrt{\Phi})},
\end{aligned}$$

where: (a) follows from Lemma 2, (b) from Lemma 3 and (20), (c) from the fact that $\mathbf{w} \cdot \boldsymbol{\eta} \in J \iff -\mathbf{w} \cdot \boldsymbol{\eta} \in J$, and (d) from (19). This completes the proof.

Remark 5 *The constant $\frac{3}{4}$ can be made arbitrarily close to 1 at the expense of an increased coefficient in front of the $\sqrt{\Phi}$ term. More precisely, the $4\sqrt{\Phi}$ term in (18) corresponds to taking two standard deviations about the mean. Taking instead k standard deviations would cause $4\sqrt{\Phi}$ to be replaced by $2k\sqrt{\Phi}$ and the $\frac{3}{4}$ constant to be replaced by $1 - 1/k^2$. This leads to (mild) improvements for large Φ.*

3.3 Asymptotic tightness

Although there is a 4^{th} power gap between the upper bound $U = \exp\left(-\frac{1}{2}\Phi\right)$ and lower bound $L \asymp \exp(-2\Phi)$ in Theorem 1, we will show that each estimate is tight in a certain regime of Φ.

Upper bound. To establish the tightness of the upper bound $U = e^{-\Phi/2}$, consider n identical experts with competences $p_1 = \ldots = p_n = p > \frac{1}{2}$. Then

$$\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) = \mathbb{P}(B < \tfrac{1}{2}n) = \mathbb{P}(B < n(p-\varepsilon)), \tag{21}$$

where $B \sim \text{Bin}(n,p)$ and $\varepsilon = p - \frac{1}{2}$. By Sanov's theorem (den Hollander, 2000),

$$\lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}(B < n(p-\varepsilon)) = H(p-\varepsilon||p) = H(\tfrac{1}{2}||p), \tag{22}$$

where

$$H(x||y) = x\ln\frac{x}{y} + (1-x)\ln\frac{1-x}{1-y}, \qquad 0 < x, y < 1.$$

Hence,

$$\begin{aligned} \frac{1}{n}\log\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) &\overset{(a)}{=} \frac{1}{n}\log\mathbb{P}(B < \tfrac{1}{2}n) \\ &\xrightarrow[n\to\infty]{(b)} -H(\tfrac{1}{2}||p) \\ &= \tfrac{1}{2}\ln 2p + \tfrac{1}{2}\ln 2(1-p), \end{aligned}$$

(where (a) and (b) follow from (21) and (22), respectively) whence

$$\begin{aligned} \lim_{n\to\infty} \sqrt[n]{\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y)} &= \exp\left(\tfrac{1}{2}\ln(2p) + \tfrac{1}{2}\ln(2(1-p))\right) \\ &= 2\sqrt{p(1-p)}. \end{aligned} \tag{23}$$

On the other hand,

$$\Phi = \sum_{i=1}^{n}(p_i - \tfrac{1}{2})\log\frac{p_i}{1-p_i} = n(p-\tfrac{1}{2})\log\frac{p}{1-p},$$

and hence

$$\sqrt[n]{U} = [(1-p)/p]^{(p-\frac{1}{2})/2}.$$

The tightness of the upper bound follows from

$$F(p) := \frac{2\sqrt{p(1-p)}}{[(1-p)/p]^{(p-\frac{1}{2})/2}} \xrightarrow[p\to 1/2]{} 1,$$

which is easily verified since $F(\frac{1}{2}) = 1$.

Lower bound. For the lower bound, consider a single expert with competence $p_1 = p > \frac{1}{2}$. Thus, $\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) = 1 - p$ and $L \asymp \exp(-2\Phi) = [(1-p)/p]^{2p-1}$. Again, it is easily verified that

$$\frac{[(1-p)/p]^{2p-1}}{1-p} \quad \underset{p\to 1}{\longrightarrow} \quad 1,$$

and so the lower bound is also tight.

We conclude that the committee profile Φ is not sufficiently sensitive an indicator to close the gap between the two bounds entirely.

Remark 6 *In the special case of identical experts, with $p_1 = \ldots = p_n = p$, the Chernoff-Stein lemma (Cover and Thomas, 2006) gives the best asymptotic exponent for one-sided (i.e., type I or type II) errors, while* Chernoff information *corresponds to the optimal exponent for the overall probability of error. As seen from (23), the latter is given by $\frac{1}{2}\ln(2p) + \frac{1}{2}\ln(2(1-p))$ in this case.*

In contradistinction, our bounds in Theorem 1 hold for non-identical experts and are dimension-free.

3.4 Additional bounds

An anonymous referee has pointed out that

$$\Phi = \tfrac{1}{2}D(P||Q) = \tfrac{1}{2}D(Q||P), \tag{24}$$

where P is the distribution of $\boldsymbol{\eta} \in \{\pm 1\}^n$ defined in (13), Q is the "antipodal" distribution of $-\boldsymbol{\eta}$, and $D(P||Q)$ is the Kullback-Leibler divergence, defined by

$$D(P||Q) = \sum_{\mathbf{x}\in\{\pm 1\}^n} P(\mathbf{x}) \ln \frac{P(\mathbf{x})}{Q(\mathbf{x})}.$$

This leads to an improved lower bound for $\Phi \lesssim 0.992$, as follows. By Lemma 2, we have

$$\begin{aligned} \mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) &= \tfrac{1}{2} \sum_{\boldsymbol{\eta}\in\{\pm 1\}^n} \min\{P(\boldsymbol{\eta}), Q(\boldsymbol{\eta})\} \\ &= \tfrac{1}{2}\left(1 - \tfrac{1}{2}\left\|P - Q\right\|_1\right), \end{aligned} \tag{25}$$

where the second identity follows from a well-known minorization characterization of the total variation distance (see, e.g., Kontorovich (2007, Lemma 2.2.2)). A bound relating the total variation distance and Kullback-Leibler divergence is known as Pinsker's inequality, and states that

$$\left\|P - Q\right\|_1 \leq \sqrt{2D(P||Q)} \tag{26}$$

holds for all distributions P, Q (see Berend et al. (2014) for historical background and a "reversed" direction of (26)). Combining (24), (25), and (26), we obtain

$$\mathbb{P}(f^{\text{OPT}}(\mathbf{X}) \neq Y) \geq \tfrac{1}{2}\left(1 - \sqrt{\Phi}\right),$$

which, for small Φ, is far superior to Theorem 1(ii) (but is vacuous for $\Phi \geq 1$).

The identity in (25) may also be used to sharpen the upper bound in Theorem 1(i) for small Φ. Invoking Even-Dar et al. (2007, Lemma 3.10), we have

$$D(P||Q) \leq \|P - Q\|_1 \log \left(\min_{\mathbf{x} \in \{\pm 1\}^n} P(\mathbf{x}) \right)^{-1}. \tag{27}$$

Let us suppose for concreteness that all of the experts are identical with $p_i = \frac{1}{2} + \gamma$ for $\gamma \in (0, \frac{1}{2})$, $i \in [n]$. Then

$$\Phi = n\gamma \log \frac{1/2 + \gamma}{1/2 - \gamma}$$

and

$$\log \left(\min_{\mathbf{x} \in \{\pm 1\}^n} P(\mathbf{x}) \right)^{-1} = n \log \frac{1}{1/2 - \gamma} =: \Gamma,$$

which, combined with (24, 25, 27) yields

$$\begin{aligned} \mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y) &\leq \tfrac{1}{2} \left(1 - \frac{\Phi}{\Gamma} \right) \\ &= \tfrac{1}{2} \left(1 - \gamma + \gamma \frac{\log(1/2 + \gamma)}{\log(1/2 - \gamma)} \right). \end{aligned} \tag{28}$$

Thus, for $0 < \gamma < \frac{1}{2}$ and

$$n < \frac{2}{\gamma} \left(\log \frac{1/2 - \gamma}{1/2 + \gamma} \right) \log \left(\frac{1 - \gamma}{2} + \frac{\gamma}{2} \cdot \frac{\log(1/2 + \gamma)}{\log(1/2 - \gamma)} \right),$$

(28) is sharper than Theorem 1(i).

4. Unknown Competences: Frequentist Approach

Our goal in this section is to obtain, insofar as possible, analogues of Theorem 1 for unknown expert competences. When the p_is are unknown, they must be estimated empirically before any useful weighted majority vote can be applied. There are various ways to model partial knowledge of expert competences (Baharad et al., 2011, 2012). Perhaps the simplest scenario for estimating the p_is is to assume that the i^{th} expert has been queried independently m_i times, out of which he gave the correct prediction k_i times. Taking the $\{m_i\}$ to be fixed, define the *committee profile* by $\mathbf{k} = (k_1, \ldots, k_n)$; this is the aggregate of the agent's empirical knowledge of the experts' performance. An *empirical decision rule* $\hat{f} : (\mathbf{x}, \mathbf{k}) \mapsto \{\pm 1\}$ makes a final decision based on the expert inputs $\mathbf{x}$ together with the committee profile. Analogously to (1), the probability of a mistake is

$$\mathbb{P}(\hat{f}(\mathbf{X}, \mathbf{K}) \neq Y). \tag{29}$$

Note that now the committee profile is an additional source of randomness. Here we run into our first difficulty: unlike the probability in (1), which is minimized by the naive Bayes

decision rule, the agent cannot formulate an optimal decision rule $\hat{f}$ in advance without knowing the p_is. This is because no decision rule is optimal uniformly over the range of possible p_is. Our approach will be to consider weighted majority decision rules of the form

$$\hat{f}(\mathbf{x}, \mathbf{k}) = \operatorname{sign}\left(\sum_{i=1}^{n} \hat{w}(k_i) x_i\right) \tag{30}$$

and to analyze their consistency properties under two different regimes: low-confidence and high-confidence. These refer to the confidence intervals of the frequentist estimate of p_i, given by

$$\hat{p}_i = \frac{k_i}{m_i}. \tag{31}$$

4.1 Low-confidence regime

In the low-confidence regime, the sample sizes m_i may be as small as 1, and we define[3]

$$\hat{w}(k_i) = \hat{w}_i^{\mathrm{LC}} := \hat{p}_i - \tfrac{1}{2}, \qquad i \in [n], \tag{32}$$

which induces the empirical decision rule $\hat{f}^{\mathrm{LC}}$. It remains to analyze $\hat{f}^{\mathrm{LC}}$'s probability of error. Recall the definition of ξ_i from (5) and observe that

$$\mathbb{E}\left[\hat{w}_i^{\mathrm{LC}} \xi_i\right] = \mathbb{E}[(\hat{p}_i - \tfrac{1}{2})\xi_i] = (p_i - \tfrac{1}{2}) p_i, \tag{33}$$

since $\hat{p}_i$ and ξ_i are independent. As in (6), the probability of error (29) is

$$\mathbb{P}\left(\sum_{i=1}^{n} \hat{w}_i^{\mathrm{LC}} \xi_i \le \frac{1}{2} \sum_{i=1}^{n} \hat{w}_i^{\mathrm{LC}}\right) = \mathbb{P}\left(\sum_{i=1}^{n} Z_i \le 0\right), \tag{34}$$

where $Z_i = \hat{w}_i^{\mathrm{LC}}(\xi_i - \frac{1}{2})$. Now the $\{Z_i\}$ are independent random variables, $\mathbb{E}Z_i = (p_i - \frac{1}{2})^2$ (by (33)), and each Z_i takes values in an interval of length $\frac{1}{2}$. Hence, the standard Hoeffding bound applies:

$$\mathbb{P}(\hat{f}^{\mathrm{LC}}(\mathbf{X}, \mathbf{K}) \neq Y) \le \exp\left[-\frac{8}{n}\left(\sum_{i=1}^{n} (p_i - \tfrac{1}{2})^2\right)^2\right]. \tag{35}$$

We summarize these calculations in

Theorem 7 *A sufficient condition*[4] *for* $\mathbb{P}(\hat{f}^{\mathrm{LC}}(\mathbf{X}, \mathbf{K}) \neq Y) \to 0$ *is*

$$\frac{1}{\sqrt{n}} \sum_{i=1}^{n} (p_i - \tfrac{1}{2})^2 \to \infty.$$

3. For $m_i \min\{p_i, q_i\} \ll 1$, the estimated competences $\hat{p}_i$ may well take values in $\{0, 1\}$, in which case $\log(\hat{p}_i/\hat{q}_i) = \pm\infty$. The rule in (32) is essentially a first-order Taylor approximation to $w(\cdot)$ about $p = \frac{1}{2}$.

4. Formally, we have an infinite sequence of experts with competences $\{p_i : i \in \mathbb{N}\}$, with a corresponding sequence of trials with sizes $\{m_i\}$ and outcomes $K_i \sim \mathrm{Bin}(m_i, p_i)$, in addition to the expert votes $X_i \sim Y\,[2 \cdot \mathrm{Bernoulli}(p_i) - 1]$. An empirical decision rule f_n (more precisely, a sequence of rules) is said to be *consistent* if

$$\lim_{n \to \infty} \mathbb{P}(f_n(\mathbf{X}, \mathbf{K}) \neq Y) = 0.$$

Several remarks are in order. First, notice that the error bound in (35) is stated in terms of the unknown $\{p_i\}$, providing the agent with large-committee asymptotics but giving no finitary information; this limitation is inherent in the low-confidence regime. Secondly, the condition in Theorem 7 is considerably more restrictive than the consistency condition $\Phi \to \infty$ implicit in Theorem 1. Indeed, the empirical decision rule $\hat{f}^{\mathrm{LC}}$ is incapable of exploiting a single highly competent expert in the way that f^{OPT} from (2) does. Our analysis could be sharpened somewhat for moderate sample sizes $\{m_i\}$ by using Bernstein's inequality to take advantage of the low variance of the $\hat{p}_i$s. For sufficiently large sample sizes, however, the high-confidence regime (discussed below) begins to take over. Finally, there is one sense in which this case is "easier" to analyze than that of known $\{p_i\}$: since the summands in (34) are bounded, Hoeffding's inequality gives nontrivial results and there is no need for more advanced tools such as the Kearns-Saul inequality (9) (which is actually inapplicable in this case).

4.2 High-confidence regime

In the high-confidence regime, each estimated competence $\hat{p}_i$ is close to the true value p_i with high probability. To formalize this, fix some $0 < \delta < 1$, $0 < \varepsilon \leq 5$, and put

$$q_i = 1 - p_i, \;\; \hat{q}_i = 1 - \hat{p}_i.$$

We will set the empirical weights according to the "plug-in" naive Bayes rule

$$\hat{w}_i^{\mathrm{HC}} := \log \frac{\hat{p}_i}{\hat{q}_i}, \qquad i \in [n], \tag{36}$$

which induces the empirical decision rule $\hat{f}^{\mathrm{HC}}$ and raises immediate concerns about $\hat{w}_i^{\mathrm{HC}} = \pm\infty$. We give two kinds of bounds on $\mathbb{P}(\hat{f}^{\mathrm{HC}} \neq Y)$: nonadaptive and adaptive. In the nonadaptive analysis, we show that for $m_i \min\{p_i, q_i\} \gg 1$, with high probability $|w_i - \hat{w}_i^{\mathrm{HC}}| \ll 1$, and thus a "perturbed" version of Theorem 1(i) holds (and in particular, w_i^{HC} will be finite with high probability). In the adaptive analysis, we allow $\hat{w}_i^{\mathrm{HC}}$ to take on infinite values[5] and show (perhaps surprisingly) that this decision rule still admits reasonable error estimates.

Nonadaptive analysis. In this section, $\varepsilon, \tilde{\varepsilon} > 0$ are related by $\varepsilon = 2\tilde{\varepsilon} + 4\tilde{\varepsilon}^2$ or, equivalently,

$$\tilde{\varepsilon} = \frac{\sqrt{4\varepsilon + 1} - 1}{4}. \tag{37}$$

Lemma 8 *If* $0 < \tilde{\varepsilon} < 1$ *and*

$$\tilde{\varepsilon}^2 m_i p_i \;\; \geq \;\; 3\log(2n/\delta), \qquad i \in [n], \tag{38}$$

then

$$\mathbb{P}\left(\exists i \in [n] : \frac{\hat{p}_i}{p_i} \notin (1 - \tilde{\varepsilon}, 1 + \tilde{\varepsilon})\right) \leq \delta.$$

5. When the decision rule is faced with evaluating sums involving $\infty - \infty$, we automatically count this as an error.

Proof The multiplicative Chernoff bound yields

$$\mathbb{P}\left(\hat{p}_i < (1-\tilde{\varepsilon})p_i\right) \leq e^{-\tilde{\varepsilon}^2 m_i p_i/2}$$

and

$$\mathbb{P}\left(\hat{p}_i > (1+\tilde{\varepsilon})p_i\right) \leq e^{-\tilde{\varepsilon}^2 m_i p_i/3}.$$

Hence,

$$\mathbb{P}\left(\frac{\hat{p}_i}{p_i} \notin (1-\tilde{\varepsilon}, 1+\tilde{\varepsilon})\right) \quad \leq \quad 2e^{-\tilde{\varepsilon}^2 m_i p_i/3}.$$

The claim follows from (38) and the union bound. ■

Lemma 9 *Let $\delta \in (0,1)$, $\varepsilon \in (0,5)$, and w_i be the naive Bayes weight (3). If*

$$1-\tilde{\varepsilon} \leq \frac{\hat{p}_i}{p_i}, \frac{\hat{q}_i}{q_i} \leq 1+\tilde{\varepsilon}$$

then

$$|w_i - \hat{w}_i^{\mathrm{HC}}| \leq \varepsilon.$$

Proof We have

$$\begin{aligned} |w_i - \hat{w}_i^{\mathrm{HC}}| &= \left|\log\frac{p_i}{q_i} - \log\frac{\hat{p}_i}{\hat{q}_i}\right| \\ &= \left|\log\frac{p_i}{\hat{p}_i} + \log\frac{\hat{q}_i}{q_i}\right| \\ &= \left|\log\frac{p_i}{\hat{p}_i}\right| + \left|\log\frac{\hat{q}_i}{q_i}\right|. \end{aligned}$$

Now[6]

$$\begin{aligned} [\log(1-\tilde{\varepsilon}), \log(1+\tilde{\varepsilon})] &\subseteq [-\tilde{\varepsilon} - 2\tilde{\varepsilon}^2, \tilde{\varepsilon}] \\ &\subseteq [-\tfrac{1}{2}\varepsilon, \tfrac{1}{2}\varepsilon], \end{aligned}$$

whence

$$\left|\log\frac{p_i}{\hat{p}_i}\right| + \left|\log\frac{\hat{q}_i}{q_i}\right| \leq \varepsilon.$$

■

6. The first containment requires $\log(1-x) \geq -x - 2x^2$, which holds (not exclusively) on $(0, 0.9)$. The restriction $\varepsilon \leq 5$ ensures that $\tilde{\varepsilon}$ is in this range.

Corollary 10 *If*

$$\tilde{\varepsilon}^2 m_i \min\{p_i, q_i\} \;\geq\; 3\log(4n/\delta), \qquad i \in [n],$$

then

$$\mathbb{P}\left(\max_{i\in[n]} |w_i - \hat{w}_i^{\mathrm{HC}}| > \varepsilon\right) \;\leq\; \delta.$$

Proof An immediate consequence of applying Lemma 8 to p_i and q_i with the union bound. ■

To state the next result, let us arrange the plug-in weights (36) as a vector $\hat{\mathbf{w}}^{\mathrm{HC}} \in \mathbb{R}^n$, as was done with $\mathbf{w}$ and $\boldsymbol{\eta}$ from Section 3.1. The corresponding weighted majority rule $\hat{f}^{\mathrm{HC}}$ yields an error precisely when

$$\hat{\mathbf{w}}^{\mathrm{HC}} \cdot \boldsymbol{\eta} \leq 0$$

(cf. (14)). Our nonadaptive approach culminates in the following result.

Theorem 11 *Let* $0 < \delta < 1$ *and* $0 < \varepsilon < \min\{5, 2\Phi/n\}$. *If*

$$m_i \min\{p_i, q_i\} \geq 3\left(\frac{\sqrt{4\varepsilon+1}-1}{4}\right)^{-2} \log\frac{4n}{\delta}, \qquad i \in [n], \tag{39}$$

then

$$\mathbb{P}\left(\hat{f}^{\mathrm{HC}}(\mathbf{X}, \mathbf{K}) \neq Y\right) \;\leq\; \delta + \exp\left[-\frac{(2\Phi - \varepsilon n)^2}{8\Phi}\right]. \tag{40}$$

Remark 12 *For fixed* $\{p_i\}$ *different from* 0 *or* 1 *and* $\min_{i\in[n]} m_i \to \infty$, *we may take* δ *and* ε *arbitrarily small — and in this limiting case, the bound of Theorem 1(i) is recovered.*

Proof Suppose that Z, $\hat{Z}$, and U are real numbers satisfying

$$\left|Z - \hat{Z}\right| \leq U.$$

Then

$$\forall t > 0, \quad (\hat{Z} \leq 0) \implies (U > t) \vee (Z \leq t). \tag{41}$$

Indeed, if both $U \leq t$ and $Z > t$, then $\hat{Z}$ and Z are within a distance t of each other, but $Z > t$ and so $\hat{Z}$ must be greater than 0.

Observe also that $\|\boldsymbol{\eta}\|_\infty = 1$, and thus a simple application of Hölder's inequality yields

$$\begin{aligned} |\mathbf{w}\cdot\boldsymbol{\eta} - \hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}| &= |(\mathbf{w} - \hat{\mathbf{w}}^{\mathrm{HC}})\cdot\boldsymbol{\eta}| \\ &\leq \sum_{i=1}^{n} |w_i - w_i^{\mathrm{HC}}| = \|\mathbf{w} - \hat{\mathbf{w}}^{\mathrm{HC}}\|_1 . \end{aligned}$$

Invoking (41) with $Z = \mathbf{w}$, $\hat{Z} = \hat{\mathbf{w}}^{\mathrm{HC}}$, and $t = \varepsilon n$, we obtain

$$\begin{aligned}\mathbb{P}\left(\hat{\mathbf{w}}^{\mathrm{HC}} \cdot \boldsymbol{\eta} \leq 0\right) &\leq \mathbb{P}\left(\{\|\mathbf{w} - \hat{\mathbf{w}}^{\mathrm{HC}}\|_1 > \varepsilon n\} \cup \{\mathbf{w} \cdot \boldsymbol{\eta} \leq \varepsilon n\}\right) \\ &\leq \mathbb{P}(\|\mathbf{w} - \hat{\mathbf{w}}^{\mathrm{HC}}\|_1 > \varepsilon n) + \mathbb{P}(\mathbf{w} \cdot \boldsymbol{\eta} \leq \varepsilon n).\end{aligned}$$

Corollary 10 upper-bounds the first term on the right-hand side by δ. The second term is estimated by replacing Φ by $\Phi - \varepsilon n$ in (10) and repeating the argument following that formula. ■

Adaptive analysis. Theorem 11 has the drawback of being *nonadaptive*, in that its assumptions (39) and conclusions (40) depend on the unknown $\{p_i\}$ and hence cannot be evaluated by the agent (the bound in Display 35 is also nonadaptive). In the *adaptive* approach, all results are stated in terms of empirically observed quantities:

Theorem 13 *Choose any*

$$\delta \geq \sum_{i=1}^{n} \frac{1}{\sqrt{m_i}}$$

and let R be the event

$$\exp\left(-\frac{1}{2}\sum_{i=1}^{n}(\hat{p}_i - \tfrac{1}{2})\hat{w}_i^{\mathrm{HC}}\right) \leq \frac{\delta}{2}. \tag{42}$$

Then

$$\mathbb{P}\left(R \cap \left\{\hat{f}^{\mathrm{HC}}(\mathbf{X}, \mathbf{K}) \neq Y\right\}\right) \leq \delta.$$

Remark 14 *Our interpretation for Theorem 13 is as follows. The agent observes the committee profile $\mathbf{K}$, which determines the $\{\hat{p}_i, \hat{w}_i^{\mathrm{HC}}\}$, and then checks whether the event R has occurred. If not, the adaptive agent refrains from making a decision (and may choose to fall back on the low-confidence approach described previously). If R* does *hold, however, the agent predicts Y according to $\hat{f}^{\mathrm{HC}}$. The event R will tend to occur when the estimated $\hat{p}_i$s are "favorable" in the sense of inducing a large empirical committee profile. When this fails to happen (i.e., many of the $\hat{p}_i$ are close to $\frac{1}{2}$), R will be a rare event. However, in this case little is lost by refraining from a high-confidence decision and defaulting to a low-confidence one, since near $\frac{1}{2}$, the two decision procedures are very similar.*

As explained above, there does not exist a nontrivial a priori upper bound on $\mathbb{P}(\hat{f}^{\mathrm{HC}}(\mathbf{X}, \mathbf{K}) \neq Y)$ independent of any knowledge of the p_is. Instead, Theorem 13 bounds the probability of the agent being "fooled" by an unrepresentative committee profile.[7] *Note that we have done nothing to prevent $\hat{w}_i^{\mathrm{HC}} = \pm\infty$, and this may indeed happen. Intuitively, there are two reasons for infinite $\hat{w}_i^{\mathrm{HC}}$: (a) noisy $\hat{p}_i$ due to m_i being too small, or (b) the i^{th} expert is actually highly (in)competent, which causes $\hat{p}_i \in \{0, 1\}$ to be likely even for large m_i. The $1/\sqrt{m_i}$ term in the bound insures against case (a), while in case (b), choosing infinite $\hat{w}_i^{\mathrm{HC}}$ causes no harm (as we show in the proof).*

7. These adaptive bounds are similar in spirit to *empirical Bernstein* methods, (Audibert et al., 2007; Mnih et al., 2008; Maurer and Pontil, 2009), where the agent's confidence depends on the empirical variance.

Proof We will write the probability and expectation operators with subscripts (such as $\mathbf{K}$) to indicate the random variable(s) being summed over. Thus,

$$\begin{aligned}\mathbb{P}_{\mathbf{K},\mathbf{X},Y}\left(R\cap\left\{\hat{f}^{\mathrm{HC}}(\mathbf{X},\mathbf{K})\neq Y\right\}\right) &= \mathbb{P}_{\mathbf{K},\boldsymbol{\eta}}\left(R\cap\{\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}\leq 0\}\right)\\ &= \mathbb{E}_{\mathbf{K}}\left[\mathbb{1}_R\cdot\mathbb{P}_{\boldsymbol{\eta}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}\leq 0\,|\,\mathbf{K}\right)\right].\end{aligned} \tag{43}$$

Recall that the random variable $\boldsymbol{\eta}\in\{\pm 1\}^n$, with probability mass function

$$P(\boldsymbol{\eta})=\prod_{i:\eta_i=1}p_i\prod_{i:\eta_i=-1}q_i,$$

is independent of $\mathbf{K}$, and hence

$$\mathbb{P}_{\boldsymbol{\eta}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}\leq 0\,|\,\mathbf{K}\right)=\mathbb{P}_{\boldsymbol{\eta}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}\leq 0\right). \tag{44}$$

Define the random variable $\hat{\boldsymbol{\eta}}\in\{\pm 1\}^n$ (conditioned on $\mathbf{K}$) by the probability mass function

$$P(\hat{\boldsymbol{\eta}})=\prod_{i:\eta_i=1}\hat{p}_i\prod_{i:\eta_i=-1}\hat{q}_i,$$

and the set $A\subseteq\{\pm 1\}^n$ by $A=\{\mathbf{x}:\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\mathbf{x}\leq 0\}$. Now

$$\begin{aligned}\left|\mathbb{P}_{\boldsymbol{\eta}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}\leq 0\right)-\mathbb{P}_{\hat{\boldsymbol{\eta}}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\hat{\boldsymbol{\eta}}\leq 0\right)\right| &= \left|\mathbb{P}_{\boldsymbol{\eta}}(A)-\mathbb{P}_{\hat{\boldsymbol{\eta}}}(A)\right|\\ &\leq \max_{A\subseteq\{\pm 1\}^n}\left|\mathbb{P}_{\boldsymbol{\eta}}(A)-\mathbb{P}_{\hat{\boldsymbol{\eta}}}(A)\right|\\ &= \left\|\mathbb{P}_{\boldsymbol{\eta}}-\mathbb{P}_{\hat{\boldsymbol{\eta}}}\right\|_{\mathrm{TV}}\\ &\leq \sum_{i=1}^n|p_i-\hat{p}_i|=:M,\end{aligned}$$

where the last inequality follows from a standard tensorization property of the total variation norm $\|\cdot\|_{\mathrm{TV}}$, see e.g. (Kontorovich, 2012, Lemma 2.2). By Theorem 1(i), we have

$$\mathbb{P}_{\hat{\boldsymbol{\eta}}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\hat{\boldsymbol{\eta}}\leq 0\right)\leq\exp\left(-\tfrac{1}{2}\sum_{i=1}^n(\hat{p}_i-\tfrac{1}{2})\hat{w}_i^{\mathrm{HC}}\right),$$

and hence

$$\mathbb{P}_{\boldsymbol{\eta}}\left(\hat{\mathbf{w}}^{\mathrm{HC}}\cdot\boldsymbol{\eta}\leq 0\right)\leq M+\exp\left(-\tfrac{1}{2}\sum_{i=1}^n(\hat{p}_i-\tfrac{1}{2})\hat{w}_i^{\mathrm{HC}}\right).$$

Invoking (44), we substitute the right-hand side above into (43) to obtain

$$\begin{aligned}\mathbb{P}_{\mathbf{K},\mathbf{X},Y}\left(R\cap\left\{\hat{f}^{\mathrm{HC}}(\mathbf{X},\mathbf{K})\neq Y\right\}\right) &\leq \mathbb{E}_{\mathbf{K}}\left[\mathbb{1}_R\cdot\left(M+\exp\left(-\tfrac{1}{2}\sum_{i=1}^n(\hat{p}_i-\tfrac{1}{2})\hat{w}_i^{\mathrm{HC}}\right)\right)\right]\\ &\leq \mathbb{E}_{\mathbf{K}}[M]+\mathbb{E}_{\mathbf{K}}\left[\mathbb{1}_R\exp\left(-\tfrac{1}{2}\sum_{i=1}^n(\hat{p}_i-\tfrac{1}{2})\hat{w}_i^{\mathrm{HC}}\right)\right].\end{aligned}$$

By the definition of R, the second term on the last right-hand side is upper-bounded by $\delta/2$. To bound M, we invoke a simple mean absolute deviation estimate (cf. Berend and Kontorovich, 2013a):

$$\mathbb{E}_{\mathbf{K}} |p_i - \hat{p}_i| \;\leq\; \sqrt{\frac{p_i(1-p_i)}{m_i}} \;\leq\; \frac{1}{2\sqrt{m_i}},$$

which finishes the proof. ■

Remark 15 *Actually, the proof shows that we may take a smaller δ, but with a more complex dependence on $\{m_i\}$, which simplifies to $2[1-(1-(2\sqrt{m})^{-1})^n]$ for $m_i \equiv m$. This improvement is achieved via a refinement of the bound $\left\|\mathbb{P}_{\boldsymbol{\eta}} - \mathbb{P}_{\hat{\boldsymbol{\eta}}}\right\|_{\mathrm{TV}} \leq \sum_{i=1}^n |p_i - \hat{p}_i|$ to $\left\|\mathbb{P}_{\boldsymbol{\eta}} - \mathbb{P}_{\hat{\boldsymbol{\eta}}}\right\|_{\mathrm{TV}} \leq \alpha\left(\{|p_i - \hat{p}_i| : i \in [n]\}\right)$, where $\alpha(\cdot)$ is the function defined in Kontorovich (2012, Lemma 4.2).*

Open problem. As argued in Remark 12, the nonadaptive agent achieves the asymptotically optimal rate of Theorem 1(i) in the large-sample limit. Does an analogous claim hold true for the adaptive agent? Can the dependence on $\{m_i\}$ in Theorem 13 be improved, perhaps through a better choice of $\hat{\mathbf{w}}^{\mathrm{HC}}$?

5. Unknown Competences: Bayesian Approach

A shortcoming of Theorem 13 is that, when condition R fails, the agent is left with no estimate of the error probability. An alternative (and in some sense cleaner) approach to handling unknown expert competences p_i is to assume a known prior distribution over the competence levels p_i. The natural choice of prior for a Bernoulli parameter is the Beta distribution, namely

$$p_i \sim \mathrm{Beta}(\alpha_i, \beta_i)$$

with density

$$\frac{p_i^{\alpha_i - 1} q_i^{\beta_i - 1}}{B(\alpha_i, \beta_i)}, \qquad \alpha_i, \beta_i > 0,$$

where $q_i = 1 - p_i$ and $B(x,y) = \Gamma(x)\Gamma(y)/\Gamma(x+y)$. Our full probabilistic model is as follows. First, "nature" chooses the true state of the world Y according to $Y \sim \mathrm{Bernoulli}(\frac{1}{2})$, and each of the n expert competences p_i is drawn independently from $\mathrm{Beta}(\alpha_i, \beta_i)$ with known parameters α_i, β_i. Then the i^{th} expert, $i \in [n]$, is queried (on independent instances) m_i times, with $K_i \sim \mathrm{Bin}(m_i, p_i)$ correct predictions and $m_i - K_i$ incorrect ones. As before, $\mathbf{K} = (K_1, \ldots, K_n)$ is the (random) committee profile. Additionally, $\mathbf{X} = (X_1, \ldots, X_n)$ is the random voting profile, where $X_i \sim Y\,[2 \cdot \mathrm{Bernoulli}(p_i) - 1]$, independent of the other random variables. Absent direct knowledge of the p_is, the agent relies on an empirical decision rule $\hat{f} : (\mathbf{x}, \mathbf{k}) \mapsto \{\pm 1\}$ to produce a final decision based on the expert inputs $\mathbf{x}$ together with the committee profile $\mathbf{k}$. A decision rule $\hat{f}^{\mathrm{Ba}}$ is *Bayes-optimal* if it minimizes

$$\mathbb{P}(\hat{f}(\mathbf{X}, \mathbf{K}) \neq Y), \tag{45}$$

which is formally identical to (29) but semantically there is a difference: the probability in (45) is over the p_i in addition to $(\mathbf{X}, Y, \mathbf{K})$. Unlike the frequentist approach, where no optimal empirical decision rule was possible, the Bayesian approach readily admits one:

Theorem 16 *The decision rule*

$$\hat{f}^{\mathrm{Ba}}(\mathbf{x}, \mathbf{k}) = \operatorname{sign}\left(\sum_{i=1}^{n} \hat{w}_i^{\mathrm{Ba}} x_i\right), \tag{46}$$

where

$$\hat{w}_i^{\mathrm{Ba}} = \log \frac{\alpha_i + k_i}{\beta_i + m_i - k_i}, \tag{47}$$

minimizes the probability in (45) over all empirical decision rules.

Remark 17 *For $0 < p_i < 1$, we have*

$$\hat{w}_i^{\mathrm{Ba}} \underset{m_i \to \infty}{\longrightarrow} w_i, \qquad i \in [n],$$

almost surely, both in the frequentist and the Bayesian interpretations.

Proof Denote

$$M_n \;=\; \{0, \ldots, m_1\} \times \{0, \ldots, m_2\} \times \ldots \times \{0, \ldots, m_n\}$$

and let $f : \{\pm 1\}^n \times M_n \to \{\pm 1\}$ be an arbitrary empirical decision rule. Then

$$\mathbb{P}(f(\mathbf{X}, \mathbf{K}) \neq Y) \;=\; \sum_{\mathbf{x} \in \{\pm 1\}^n,\ \mathbf{k} \in M_n} \mathbb{P}(\mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}) \cdot \mathbb{P}(f(\mathbf{X}, \mathbf{K}) \neq Y \,|\, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}).$$

Observe that the quantity $\mathbb{P}(Y = y \,|\, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k})$ is completely determined by y, $\mathbf{x}$, $\mathbf{k}$, and the parameters $\boldsymbol{\alpha}, \boldsymbol{\beta} \in \mathbb{R}^n$, and denote this functional dependence by

$$\mathbb{P}(Y = y \,|\, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}) \;=:\; G_{\boldsymbol{\alpha},\boldsymbol{\beta}}(y, \mathbf{x}, \mathbf{k}).$$

Then clearly, the optimal empirical decision rule is

$$f^*_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{x}, \mathbf{k}) \;=\; \begin{cases} +1, & G_{\boldsymbol{\alpha},\boldsymbol{\beta}}(+1, \mathbf{x}, \mathbf{k}) \geq G_{\boldsymbol{\alpha},\boldsymbol{\beta}}(-1, \mathbf{x}, \mathbf{k}), \\ -1, & G_{\boldsymbol{\alpha},\boldsymbol{\beta}}(+1, \mathbf{x}, \mathbf{k}) < G_{\boldsymbol{\alpha},\boldsymbol{\beta}}(-1, \mathbf{x}, \mathbf{k}), \end{cases}$$

and a decision rule $f_{\boldsymbol{\alpha},\boldsymbol{\beta}}$ is optimal if and only if

$$\mathbb{P}(f_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{X}, \mathbf{K}) = Y \,|\, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}) \;\geq\; \mathbb{P}(f_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{X}, \mathbf{K}) \neq Y \,|\, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}) \tag{48}$$

for all $\mathbf{x}, \mathbf{k}, \boldsymbol{\alpha}, \boldsymbol{\beta}$. Invoking Bayes' formula, we may rewrite the optimality criterion in (48) in the form

$$\mathbb{P}(f_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{X}, \mathbf{K}) = Y, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}) \;\geq\; \mathbb{P}(f_{\boldsymbol{\alpha},\boldsymbol{\beta}}(\mathbf{X}, \mathbf{K}) \neq Y, \mathbf{X} = \mathbf{x}, \mathbf{K} = \mathbf{k}). \tag{49}$$

For given $\mathbf{x} \in \{\pm 1\}^n$ and $\mathbf{k} \in M_n$, let $I_+(\mathbf{x})$ be the set of YES votes

$$I_+(\mathbf{x}) = \{i \in [n] : x_i = +1\}$$

and $I_-(\mathbf{x}) = [n] \setminus I_+(\mathbf{x})$ the set of NO votes. Let us fix some $A \subseteq [n]$, $B = [n] \setminus A$ and compute

$$\begin{aligned}
&\mathbb{P}(Y = +1, I_+(\mathbf{X}) = A, I_-(\mathbf{X}) = B, \mathbf{k} = \mathbf{K}) \\
&= \prod_{i=1}^n \int_0^1 \frac{p_i^{\alpha_i - 1} q_i^{\beta_i - 1}}{B(\alpha_i, \beta_i)} \binom{m_i}{k_i} p_i^{k_i} q_i^{m_i - k_i} p_i^{\mathbb{1}_{\{i \in A\}}} q_i^{\mathbb{1}_{\{i \in B\}}} dp_i \\
&= \prod_{i=1}^n \frac{\binom{m_i}{k_i}}{B(\alpha_i, \beta_i)} \int_0^1 p_i^{\alpha_i + k_i - 1 + \mathbb{1}_{\{i \in A\}}} q_i^{\beta_i + m_i - k_i - 1 + \mathbb{1}_{\{i \in B\}}} dp_i \\
&= \prod_{i=1}^n \frac{\binom{m_i}{k_i} B(\alpha_i + k_i + \mathbb{1}_{\{i \in A\}}, \beta_i + m_i - k_i + \mathbb{1}_{\{i \in B\}})}{B(\alpha_i, \beta_i)}. \qquad (50)
\end{aligned}$$

Analogously,

$$\begin{aligned}
&\mathbb{P}(Y = -1, I_+(\mathbf{X}) = A, I_-(\mathbf{X}) = B, \mathbf{k} = \mathbf{K}) \\
&= \prod_{i=1}^n \frac{\binom{m_i}{k_i} B(\alpha_i + k_i + \mathbb{1}_{\{i \in B\}}, \beta_i + m_i - k_i + \mathbb{1}_{\{i \in A\}})}{B(\alpha_i, \beta_i)}. \qquad (51)
\end{aligned}$$

Let us use the shorthand $P(+1, A, B, \mathbf{k})$ and $P(-1, A, B, \mathbf{k})$ for the joint probabilities in the last two displays, along with their corresponding conditionals $P(\pm 1 \,|\, A, B, \mathbf{k})$. Obviously,

$$P(1|A, B, \mathbf{k}) > P(-1|A, B, \mathbf{k}) \iff P(1, A, B, \mathbf{k}) > P(-1, A, B, \mathbf{k}),$$

which occurs precisely if

$$\prod_{i=1}^n B(\alpha_i + k_i + \mathbb{1}_{\{i \in A\}}, \beta_i + m_i - k_i + \mathbb{1}_{\{i \in B\}}) > \prod_{i=1}^n B(\alpha_i + k_i + \mathbb{1}_{\{i \in B\}}, \beta_i + m_i - k_i + \mathbb{1}_{\{i \in A\}}), \qquad (52)$$

as the other factors in (50) and (51) cancel out. Now $B(x, y) = \Gamma(x)\Gamma(y)/\Gamma(x + y)$ and

$$\begin{aligned}
\Gamma(\alpha_i + k_i + \mathbb{1}_{\{i \in A\}} + \beta_i + m_i - k_i + \mathbb{1}_{\{i \in B\}}) &= \Gamma(\alpha_i + k_i + \mathbb{1}_{\{i \in B\}} + \beta_i + m_i - k_i + \mathbb{1}_{\{i \in A\}}) \\
&= \Gamma(\alpha_i + \beta_i + m_i + 1),
\end{aligned}$$

and thus both sides of (52) share a common factor of

$$\left(\prod_{i=1}^n \Gamma(\alpha_i + \beta_i + m_i + 1) \right)^{-1}.$$

Furthermore, the identity $\Gamma(x + 1) = x\Gamma(x)$ implies

$$\begin{aligned}
\Gamma(\alpha_i + k_i + \mathbb{1}_{\{i \in A\}}) &= (\alpha_i + k_i)^{\mathbb{1}_{\{i \in A\}}} \Gamma(\alpha_i + k_i), \\
\Gamma(\beta_i + m_i - k_i + \mathbb{1}_{\{i \in B\}}) &= (\beta_i + m_i - k_i)^{\mathbb{1}_{\{i \in B\}}} \Gamma(\beta_i + m_i - k_i),
\end{aligned}$$

and thus both sides of (52) share a common factor of

$$\prod_{i=1}^{n} \Gamma(\alpha_i + k_i)\Gamma(\beta_i + m_i - k_i).$$

After cancelling out the common factors, (52) becomes equivalent to

$$\prod_{i\in A}(\alpha_i + k_i)\prod_{i\in B}(\beta_i + m_i - k_i) \quad > \quad \prod_{i\in B}(\alpha_i + k_i)\prod_{i\in A}(\beta_i + m_i - k_i),$$

which further simplifies to

$$\prod_{i\in A}\frac{\alpha_i + k_i}{\beta_i + m_i - k_i} > \prod_{i\in B}\frac{\alpha_i + k_i}{\beta_i + m_i - k_i}.$$

Hence, the choice (47) of $\hat{w}_i^{\mathrm{Ba}}$ guarantees that the decision rule in (46) is indeed optimal. ■

Remark 18 *Unfortunately, although*

$$\mathbb{P}(\hat{f}^{\mathrm{Ba}}(\mathbf{X}, \mathbf{K}) \neq Y) = \mathbb{P}(\hat{\mathbf{w}}^{\mathrm{Ba}} \cdot \boldsymbol{\eta} \leq 0)$$

is a deterministic function of $\{\alpha_i, \beta_i, m_i\}$, we are unable to compute it at this point, or even give a non-trivial bound. The main source of difficulty is the coupling between $\hat{\mathbf{w}}^{\mathrm{Ba}}$ and $\boldsymbol{\eta}$.

Open problem. Give a non-trivial estimate for $\mathbb{P}(\hat{f}^{\mathrm{Ba}}(\mathbf{X}, \mathbf{K}) \neq Y)$.

6. Experiments

It is most instructive to take the committee size n to be small when comparing the different voting rules. Indeed, for a large committee of "marginally competent" experts with $p_i = \frac{1}{2} + \gamma$ for some $\gamma > 0$, even the simple majority rule $f^{\mathrm{MAJ}}(\mathbf{x}) = \mathrm{sign}(\sum_{i=1}^{n} x_i)$ has a probability of error decaying as $\exp(-4n\gamma^2)$, as can be easily seen from Hoeffding's bounds. The more sophisticated voting rules discussed in this paper perform even better in this setting; see Helmbold and Long (2012) for an in-depth study of the utility gained from weak experts. Hence, small committees provide the natural test-bed for gauging a voting rule's ability to exploit highly competent experts. In our experiments, we set $n = 5$ and the sample sizes m_i were identical for all experts. The results were averaged over 10^5 trials. Two of our experiments are described below.

Low vs. high confidence. The goal of this experiment was to contrast the extremal behavior of $\hat{f}^{\mathrm{LC}}$ vs. $\hat{f}^{\mathrm{HC}}$. To this end, we numerically optimized the $\mathbf{p} \in [0,1]^n$ so as to maximize the *absolute gap*

$$\Delta_n(\mathbf{p}) := \mathbb{P}(f^{\mathrm{LC}}(\mathbf{X}) \neq Y) - \mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y),$$

where $f^{\mathrm{LC}}(\mathbf{x}) = \mathrm{sign}\left(\sum_{i=1}^{n}(p_i - \frac{1}{2})x_i\right)$. We were surprised to discover that, though the ratio $\mathbb{P}(f^{\mathrm{LC}}(\mathbf{X}) \neq Y)/\mathbb{P}(f^{\mathrm{OPT}}(\mathbf{X}) \neq Y)$ can be made arbitrarily large by setting $p_1 \approx 1$ and the remaining $p_i < 1 - \varepsilon$, the absolute gap appears to be rather small: we conjecture (with some heuristic justification[8]) that $\sup_{n\geq 1}\sup_{\mathbf{p}\in[0,1]^n}\Delta_n(\mathbf{p}) = 1/16$. For $\hat{f}^{\mathrm{Ba}}$, we used $\alpha_i = \beta_i = 1$ for all i. The results are reported in Figure 1.

8. The intuition is that we want one of the experts to be perfect (i.e., $p = 1$) and two others to be "moderately strong," whereby under the low confidence rule, the two can collude to overwhelm the perfect

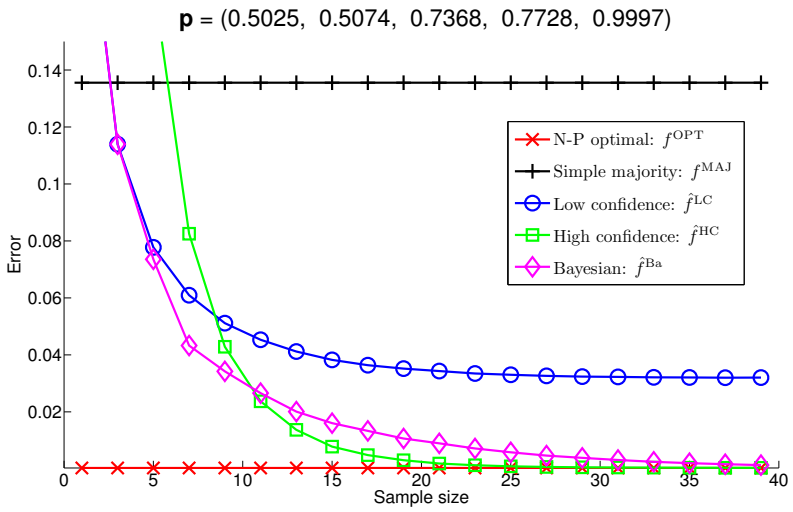


Figure 1: For very small sample sizes, $\hat{f}^{\mathrm{LC}}$ outperforms $\hat{f}^{\mathrm{HC}}$ but is outperformed by $\hat{f}^{\mathrm{Ba}}$. Starting from sample size ≈ 13, $\hat{f}^{\mathrm{HC}}$ dominates the other empirical rules. The empirical rules are (essentially) sandwiched between f^{OPT} and f^{MAJ}.

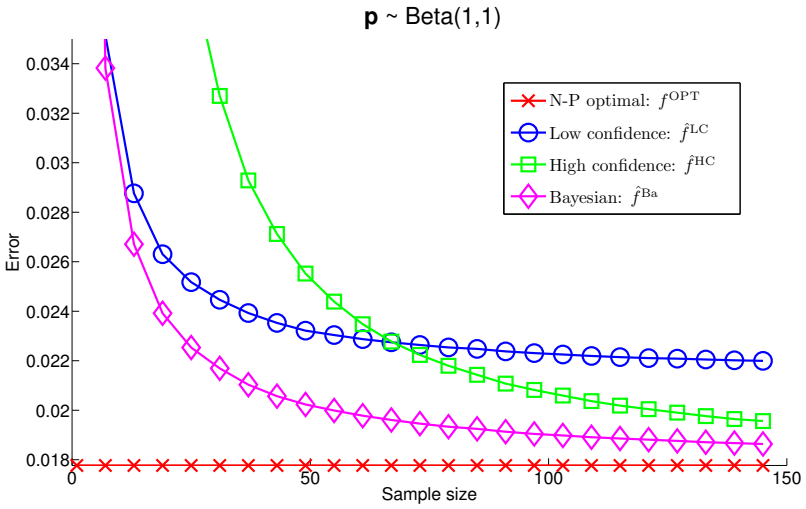


Figure 2: Unsurprisingly, $\hat{f}^{\mathrm{Ba}}$ uniformly outperforms the other two empirical rules. We found it somewhat surprising that $\hat{f}^{\mathrm{HC}}$ required so many samples (about 60 on average) to overtake $\hat{f}^{\mathrm{LC}}$. The simple majority rule f^{MAJ} (off the chart) performed at an average accuracy of 50%, as expected.

expert, but neither of them alone can. For $n = 3$, the choice $\mathbf{p} = (1, 3/4 + \varepsilon, 3/4 + \varepsilon)$ asymptotically achieves the gap $\Delta_3(\mathbf{p}) = 1/16$.

Bayesian setting. In each trial, a vector of expert competences $\mathbf{p} \in [0,1]^n$ was drawn independently componentwise, with $p_i \sim \mathrm{Beta}(1,1)$. These values (i.e., $\alpha_i = \beta_i \equiv 1$) were used for $\hat{f}^{\mathrm{Ba}}$. The results are reported in Figure 2.

7. Discussion

The classic and seemingly well-understood problem of the consistency of weighted majority votes continues to reveal untapped depth and suggest challenging unresolved questions. We hope that the results and open problems presented here will stimulate future research.

Acknowledgements

We thank Tony Jebara, Phil Long, Elchanan Mossel, and Boaz Nadler for enlightening discussions and for providing useful references. This paper greatly benefited from a careful reading by two diligent referees, who corrected inaccuracies and even supplied some new results. A special thanks to Lawrence Saul for writing up the new proof of the Kearns-Saul inequality and allowing us to print it here.

Appendix A. Bibliographical Notes on the Kearns-Saul Inequality

Given the recent interest surrounding the Kearns-Saul inequality (9), we find it instructive to provide some historical notes on this and related results. Most of the material in this section is taken from Saul (2014), to whom we are indebted for writing the note and for his kind permission to include it in this paper.

Lemma 19 *Let $f(x) = \log\cosh(\frac{1}{2}\sqrt{x})$. Then $f(x)$ is concave on $x \geq 0$.*

Proof The second derivative is given by

$$f''(x) \;=\; \frac{\mathrm{sech}^2(\frac{1}{2}\sqrt{x})}{16x^{3/2}}\left[\sqrt{x} - \sinh(\sqrt{x})\right].$$

For $x > 0$, the first of these factors is positive, and the second is negative. To show the latter, recall the Taylor series expansion

$$\sinh(t) = t + \frac{t^3}{3!} + \frac{t^5}{5!} + \frac{t^7}{7!} + \ldots,$$

from which we observe that $\sqrt{x} \leq \sinh(\sqrt{x})$. It also follows from the Taylor series that $f''(0) = -\frac{1}{96}$. It follows that f'' is negative on the positive half-line, and hence f is concave on this domain. ■

Corollary 20 *For $x, x_0 > 0$, we have*

$$\log\cosh(\tfrac{1}{2}\sqrt{x}) \;\leq\; \log\cosh(\tfrac{1}{2}\sqrt{x_0}) + \left[\frac{\tanh(\frac{1}{2}\sqrt{x_0})}{4\sqrt{x_0}}\right](x - x_0). \tag{53}$$

Proof A concave function $f(x)$ is upper-bounded by its first-order Taylor approximation: $f(x) \leq f(x_0) + f'(x_0)(x - x_0)$. The claim follows from Lemma 19. ■

The results in Lemma 19 and Corollary 20 were first stated by Jaakkola and Jordan (1997); see Jebara (2011); Jebara and Choromanska (2012) for extensions, including a multivariate version. As pointed out by a referee, Theorem 1 in Hoeffding (1963) contains some bounds that bear a resemblance to the Kearns-Saul inequality. However, we were unable to derive the latter from the former — which, in particular, requires all of the summands to be bounded between 0 and 1.

Suppose that in Equation (53), we make the substitutions

$$\sqrt{x} = \left| t + \log \frac{p}{1-p} \right|, \tag{54}$$

$$\sqrt{x_0} = \left| \log \frac{p}{1-p} \right|, \tag{55}$$

where $t \in \mathbb{R}$ and $p \in (0,1)$. Then we obtain a particular form of the bound that will be especially useful in what follows.

Corollary 21 *For all $t \in \mathbb{R}$ and $p \in (0,1)$,*

$$\log \cosh \left(\tfrac{1}{2} \left[t + \log \frac{p}{1-p} \right] \right) \leq -\log \left[2\sqrt{p(1-p)} \right] + (p - \tfrac{1}{2})t + \left(\frac{2p-1}{4 \log \frac{p}{1-p}} \right) t^2.$$

Proof Make the substitutions suggested in (54, 55) and apply Corollary 20. The result follows from tedious but elementary algebra. ■

The above result yields perhaps the most natural and direct proof of the Kearns-Saul inequality to date:

Theorem 22 *For all $t \in \mathbb{R}$ and $p \in (0,1)$,*

$$\log \left[(1-p)e^{-pt} + pe^{(1-p)t} \right] \leq \left(\frac{2p-1}{4 \log \frac{p}{1-p}} \right) t^2.$$

Proof Rewrite the left-hand side by symmetrizing the argument inside the logarithm,

$$\log \left[(1-p)e^{-pt} + pe^{(1-p)t} \right] = \log \cosh \left(\tfrac{1}{2} \left[t + \log \frac{p}{1-p} \right] \right) - (p - \tfrac{1}{2})t + \log \left[2\sqrt{p(1-p)} \right],$$

and invoke Corollary 21. ■

The inequality in Theorem 22 was first stated by Kearns and Saul (1998) and first rigorously proved by Berend and Kontorovich (2013b). Shortly thereafter, Raginsky (2012) provided a very elegant proof based on transportation and information-theoretic techniques, which currently appears as Theorem 37 in Raginsky and Sason (2013). A third proof, found by Schlemm (2014), fleshes out the original strategy suggested by Kearns and Saul (1998). The fourth proof, given here, is due to Saul (2014).

References

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Tuning bandit algorithms in stochastic environments. In *Algorithmic Learning Theory (ALT)*, 2007.

Eyal Baharad, Jacob Goldberger, Moshe Koppel, and Shmuel Nitzan. Distilling the wisdom of crowds: weighted aggregation of decisions on multiple issues. *Autonomous Agents and Multi-Agent Systems*, 22(1):31–42, 2011.

Eyal Baharad, Jacob Goldberger, Moshe Koppel, and Shmuel Nitzan. Beyond Condorcet: Optimal aggregation rules using voting records. *Theory and Decision*, 72(1):113–130, 2012.

Daniel Berend and Aryeh Kontorovich. A sharp estimate of the binomial mean absolute deviation with applications. *Statistics & Probability Letters*, 83(4):1254–1259, 2013a.

Daniel Berend and Aryeh Kontorovich. On the concentration of the missing mass. *Electron. Commun. Probab.*, 18:no. 3, 1–7, 2013b.

Daniel Berend and Aryeh Kontorovich. Consistency of weighted majority votes. In *Neural Information Processing Systems (NIPS)*, 2014.

Daniel Berend and Jacob Paroush. When is Condorcet's jury theorem valid? *Soc. Choice Welfare*, 15(4):481–488, 1998.

Daniel Berend and Luba Sapir. Monotonicity in Condorcet's jury theorem with dependent voters. *Social Choice and Welfare*, 28(3):507–528, 2007.

Daniel Berend, Peter Harremoës, and Aryeh Kontorovich. Minimum KL-divergence on complements of L_1 balls. *IEEE Transactions on Information Theory*, 60(6):3172–3177, 2014.

Philip J. Boland, Frank Proschan, and Y. L. Tong. Modelling dependence in simple and indirect majority systems. *J. Appl. Probab.*, 26(1):81–88, 1989. ISSN 0021-9002.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, 2006.

Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. Wiley-Interscience, Hoboken, NJ, second edition, 2006.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28(1):20–28, 1979.

J.A.N. de Caritat marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. AMS Chelsea Publishing Series. Chelsea Publishing Company, 1785.

Frank den Hollander. *Large deviations*, volume 14 of *Fields Institute Monographs*. American Mathematical Society, Providence, RI, 2000.

Elad Eban, Elad Mezuman, and Amir Globerson. Discrete chebyshev classifiers. In *International Conference on Machine Learning (ICML) (2)*, 2014.

Eyal Even-Dar, Sham M. Kakade, and Yishay Mansour. The value of observation for monitoring dynamic systems. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2007.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997.

Chao Gao and Dengyong Zhou. Minimax optimal convergence rates for estimating ground truth from crowdsourced labels (arxiv:1310.5764). 2014.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, 2009.

David P. Helmbold and Philip M. Long. On the necessity of irrelevant variables. *Journal of Machine Learning Research*, 13:2145–2170, 2012.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *American Statistical Association Journal*, 58:13–30, 1963.

Tommi S. Jaakkola and Michael I. Jordan. A variational approach to Bayesian logistic regression models and their extensions. In *Artificial Intelligence and Statistics, AISTATS*, 1997.

Tony Jebara. Multitask sparsity via maximum entropy discrimination. *Journal of Machine Learning Research*, 12:75–110, 2011.

Tony Jebara and Anna Choromanska. Majorization for CRFs and latent likelihoods. In *Neural Information Processing Systems (NIPS)*, 2012.

Michael J. Kearns and Lawrence K. Saul. Large deviation methods for approximate probabilistic inference. In *Uncertainty in Artificial Intelligence (UAI)*, 1998.

Aryeh Kontorovich. Obtaining measure concentration from Markov contraction. *Markov Processes and Related Fields*, 4:613–638, 2012.

Aryeh (Leonid) Kontorovich. *Measure Concentration of Strongly Mixing Processes with Applications*. PhD thesis, Carnegie Mellon University, 2007.

Alexandre Lacasse, François Laviolette, Mario Marchand, Pascal Germain, and Nicolas Usunier. PAC-Bayes bounds for the risk of the majority vote and the variance of the Gibbs classifier. In *Neural Information Processing Systems (NIPS)*, 2006.

François Laviolette and Mario Marchand. PAC-Bayes risk bounds for stochastic averages and majority votes of sample-compressed classifiers. *Journal of Machine Learning Research*, 8:1461–1487, 2007.

Hongwei Li, Bin Yu, and Dengyong Zhou. Error rate bounds in crowdsourcing models. *CoRR*, abs/1307.2674, 2013.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. In *Foundations of Computer Science (FOCS)*, 1989.

Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

Yishay Mansour, Aviad Rubinstein, and Moshe Tennenholtz. Robust aggregation of experts signals. 2013.

Andreas Maurer and Massimiliano Pontil. Empirical Bernstein bounds and sample-variance penalization. In *Conference on Learning Theory (COLT)*, 2009.

David A. McAllester and Luis E. Ortiz. Concentration inequalities for the missing mass and for histogram rule error. *Journal of Machine Learning Research*, 4:895–911, 2003.

Volodymyr Mnih, Csaba Szepesvári, and Jean-Yves Audibert. Empirical Bernstein stopping. In *International Conference on Machine Learning (ICML)*, 2008.

Jerzy Neyman and Egon S. Pearson. On the problem of the most efficient tests of statistical hypotheses. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 231(694-706):289–337, 1933.

Shmuel Nitzan and Jacob Paroush. Optimal decision rules in uncertain dichotomous choice situations. *International Economic Review*, 23(2):289–297, 1982.

Fabio Parisi, Francesco Strino, Boaz Nadler, and Yuval Kluger. Ranking and combining multiple predictors without labeled data. *Proceedings of the National Academy of Sciences*, 111(4):1253–1258, 2014.

Maxim Raginsky. Derivation of the Kearns-Saul inequality by optimal transportation (private communication), 2012.

Maxim Raginsky and Igal Sason. Concentration of measure inequalities in information theory, communications and coding. *Foundations and Trends in Communications and Information Theory*, 10(1-2):1–247, 2013.

Jean-Francis Roy, François Laviolette, and Mario Marchand. From PAC-Bayes bounds to quadratic programs for majority votes. In *International Conference on Machine Learning (ICML)*, 2011.

Lawrence K. Saul. Yet another proof of an obscure inequality (private communication), 2014.

Robert E. Schapire and Yoav Freund. *Boosting. Foundations and algorithms.* Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2012.

Eckhard Schlemm. The Kearns–Saul inequality for Bernoulli and Poisson-binomial distributions. *Journal of Theoretical Probability*, pages 1–15, 2014.

Journal of Machine Learning Research 16 (2015) 1547-1572 Submitted 1/14; Revised 9/14; Published 8/15

Flexible High-Dimensional Classification Machines and Their Asymptotic Properties

Xingye Qiao QIAO@MATH.BINGHAMTON.EDU
Department of Mathematical Sciences
Binghamton University
State University of New York
Binghamton, NY 13902-6000, USA

Lingsong Zhang LINGSONG@PURDUE.EDU
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Editor: Massimiliano Pontil

Abstract

Classification is an important topic in statistics and machine learning with great potential in many real applications. In this paper, we investigate two popular large-margin classification methods, Support Vector Machine (SVM) and Distance Weighted Discrimination (DWD), under two contexts: the high-dimensional, low-sample size data and the imbalanced data. A unified family of classification machines, the FLexible Assortment MachinE (FLAME) is proposed, within which DWD and SVM are special cases. The FLAME family helps to identify the similarities and differences between SVM and DWD. It is well known that many classifiers overfit the data in the high-dimensional setting; and others are sensitive to the imbalanced data, that is, the class with a larger sample size overly influences the classifier and pushes the decision boundary towards the minority class. SVM is resistant to the imbalanced data issue, but it overfits high-dimensional data sets by showing the undesired data-piling phenomenon. The DWD method was proposed to improve SVM in the high-dimensional setting, but its decision boundary is sensitive to the imbalanced ratio of sample sizes. Our FLAME family helps to understand an intrinsic connection between SVM and DWD, and provides a trade-off between sensitivity to the imbalanced data and overfitting the high-dimensional data. Several asymptotic properties of the FLAME classifiers are studied. Simulations and real data applications are investigated to illustrate theoretical findings.

Keywords: classification, Fisher consistency, high-dimensional low-sample size asymptotics, imbalanced data, support vector machine

1. Introduction

Classification refers to predicting the class label, $y \in \mathcal{C}$, of a data object based on its covariates, $\boldsymbol{x} \in \mathcal{X}$. Here $\mathcal{C}$ is the space of class labels, and $\mathcal{X}$ is the space of the covariates. Usually we consider $\mathcal{X} \equiv \mathbb{R}^d$, where d is the number of variables or the dimension. See Duda et al. (2001) and Hastie et al. (2009) for a comprehensive introduction to many popular classification methods. When $\mathcal{C} = \{+1, -1\}$, this is an important class of classification

problems, called binary classification. The classification rule for a binary classifier usually has the form $\phi(\boldsymbol{x}) = \text{sign}\{f(\boldsymbol{x})\}$, where $f(\boldsymbol{x})$ is called the discriminant function. Linear classifiers are the most important and the most commonly used classifiers, as they are often easy to interpret in addition to reasonable classification performance. We focus on linear classifier in this article. In the above formula, linear classifiers correspond to $f(\boldsymbol{x}; \boldsymbol{\omega}, \beta) = \boldsymbol{x}^T\boldsymbol{\omega}+\beta$. The sample space is divided into halves by the *separating hyperplane*, also known as the *classification boundary*, defined by $\{\boldsymbol{x} : f(\boldsymbol{x}) \equiv \boldsymbol{x}^T\boldsymbol{\omega} + \beta = 0\}$. Note that the coefficient vector $\boldsymbol{\omega} \in \mathbb{R}^d$ defines the normal vector, and hence the orientation, of the classification boundary; and the intercept term $\beta \in \mathbb{R}$ defines the location of the classification boundary.

In this paper, two popular classification methods, Support Vector Machine (SVM; Cortes and Vapnik, 1995; Vapnik, 1998; Cristianini and Shawe-Taylor, 2000) and Distance Weighted Discrimination (DWD; Marron et al., 2007) are investigated under two important contexts: the High-Dimensional, Low-Sample Size (HDLSS) data and the imbalanced data. Both methods are large-margin classifiers (Smola et al., 2000), that seek separating hyperplanes which maximize certain notions of *gap* (that is, distances) between the two classes. The investigation of the performance of SVM and DWD motivates the notion of a unified family of classifiers, the FLexible Assortment MachinE (FLAME), which connects the two classifiers, and helps to understand their connections and differences.

There is a large literature in statistics and machine learning on large-margin classifiers. For example, Wahba (1999) studied kernel SVM in Reproducing Kernel Hilbert Spaces. Lin (2004) introduced and proved Fisher consistency for SVM. Bartlett et al. (2006) quantified the excess risk of a loss function in a learning problem including the case of large-margin classification. On the methodology level, Shen et al. (2003) invented ψ-learning; Wu and Liu (2007) introduced robust SVM. Recently, Liu, Zhang, and Wu (2011) studies a unified class of classifiers which connected hard classification and soft classification (probability estimation).

It is worth mentioning that the FLAME family is not proposed as a better classification method to replace SVM or DWD. Instead, it is proposed as a unified machine, which is very helpful to investigate the trade-off between generalization errors and overfitting. A single parameter will be used to control the trade-off, of which DWD and SVM sit on the two ends.

1.1 Motivation: Pros and Cons of SVM and DWD

SVM is a very popular classifier in statistics and machine learning. It has been shown to have Fisher consistency, that is, when sample size goes to infinity, its decision rule converges to the Bayes rule (Lin, 2004). SVM has several nice properties: 1) Its dual formulation is relatively easy to implement (through Quadratic Programming). 2) SVM is robust to the model specification, which makes it very popular in various real applications. However, when being applied to HDLSS data, it has been observed that a large portion of the data (usually the support vectors, to be properly defined later) lie on two hyperplanes parallel to the SVM classification boundary. This is known as the *data-piling* phenomenon (Marron et al., 2007; Ahn and Marron, 2010). Data-piling of SVM indicates a type of overfitting. Other overfitting phenomenon of SVM under the HDLSS context include:

1. The angle between the SVM direction and the Bayes rule direction is usually large.

2. The variability of the sampling distribution of the SVM direction $\boldsymbol{\omega}$ is very large (Zhang and Lin, 2013). Moreover, because the separating hyperplane is decided only by the support vectors, the SVM direction tends to be unstable, in the sense that small turbulence or measurement error to the support vectors can lead to a big change of the estimated direction.
3. In some cases, the out-of-sample classification performance may not be optimal due to the suboptimal direction of the estimated SVM discrimination direction.

DWD is a recently developed classifier to improve SVM in the HDLSS setting. It uses a different notion of gap from SVM. While SVM is to maximize the smallest distance between classes, DWD is to maximize a special average distance (harmonic mean) between classes. It has been shown in many earlier simulations that DWD largely overcomes the overfitting (data-piling) issue and it usually gives a better discrimination direction.

On the other hand, the intercept term β of the DWD method is sensitive to the sample size ratio between the two classes, that is, to the imbalanced data (Qiao et al., 2010). Note that, even though a good discriminant direction $\boldsymbol{\omega}$ is more important in revealing the structure of the data, the classification/prediction performance heavily depends on the intercept β, more than on the direction $\boldsymbol{\omega}$. As shown in Qiao et al. (2010), usually the β term of the SVM classifier is not sensitive to the sample size ratio, while the β term of the DWD method will become too large (or too small) if the sample size of the positive class (or negative class) is very large.

In summary, both methods have pros and cons. SVM has a greater stochastic variability and usually overfits the data by showing data-piling phenomena, but is less sensitive to the imbalanced data issue. DWD usually overcomes the overfitting/data-piling issue, and has a smaller sampling variability, but is very sensitive to the imbalanced data. Driven by their similarity, we propose a unified class of classifiers, FLAME, in which the above two classifiers are special cases. FLAME provides a framework to study the connections and differences between SVM and DWD. Each FLAME classifier has a parameter θ which is used to control the performance balance between overfitting the HDLSS data and the sensitivity to the imbalanced data. It turns out that the DWD method is FLAME with $\theta = 0$; and that the SVM method corresponds to FLAME with $\theta = 1$. The optimal θ depends on the trade-off among several factors: stochastic variability, overfitting and resistance against the imbalanced data. In this paper, we also propose an approach to select θ, where the resulting FLAME have the potential to achieve a balanced performance between the SVM and DWD methods.

1.2 Outline

The rest of the paper is organized as follows. Section 2 provides toy examples and highlights the strengths and drawbacks of SVM and DWD on classifying the HDLSS and imbalanced data. We develop the FLAME method in Section 3, which is motivated by the investigation of the loss functions of SVM and DWD. Section 4 provides suggestions on choosing the parameters. Three types of asymptotic results for the FLAME classifier are studied in Section 5. Section 6 demonstrates its properties using simulation experiments. A real application study is conducted in Section 7. Some concluding remarks and discussions are

made in Section 8. Technical proofs of theorems and propositions are included in Online Appendix 1.

2. Comparison of SVM and DWD

In this section, we use several toy examples to illustrate the strengths and drawbacks of SVM and DWD under two contexts: HDLSS data and imbalanced data.

2.1 Overfitting HDLSS Data

We use several simulated examples to compare SVM and DWD. The results show that the stochastic variability of the SVM direction is usually larger than that of the DWD method, and SVM directions are deviated farther away from Bayes rule directions. In addition, the new proposed FLAME machine (see details in Section 3) is also included in the comparison, and it turns out that FLAME with a mediocre θ is between the above two methods.

Figure 1 shows the comparison results between SVM, DWD and FLAME (with tuning parameter $\theta = 1/2$). We simulate 10 samples with the same underlying distribution. Each simulated data set contains 12 variables and two classes, with 120 observations in each class. The two classes have mean difference on only the first three dimensions and the within-class covariances are diagonal, that is, the variables are independent. For each simulated data set, we plot the first three components of the resulting discriminant directions from SVM, DWD and FLAME (after normalizing the 3D vectors to have unit L_2 norms), as shown in Figure 1. It clearly shows that the DWD directions (the blue down-pointing triangles) are the closest ones to the *true* Bayes rule direction (the cyan diamond marker) among the three approaches. In addition, the DWD directions have the smallest variation (that is, more stable) over different samples. The SVM directions (the red up-pointing triangles) are farthest from the *true* Bayes rule direction and have a larger variation than the other two methods. To highlight the direction variabilities of the three methods, we introduce a novel measure for the variation (instability) of the discriminant directions: the trace of the sample covariance of the resulting direction vectors over the 10 replications, which we name as *dispersion*. The dispersion for the DWD method (0.0031) is much smaller than that of the SVM method (0.0453), as highlighted in the figure as well. The new FLAME classifiers usually have a performance between DWD and SVM. Figure 1 shows the results of a specific FLAME ($\theta = 0.5$, the magenta squares), which are better than SVM but worse than DWD.

Besides the advantage in terms of the stochastic variability and the deviation from the true direction, DWD outperforms SVM in terms of stability in the presence of small perturbations. In Figure 2, we use a two-dimensional example to illustrate this phenomenon. We simulate a perfectly separable 2-dimensional data set. The theoretical Bayes rule decision boundary is shown as the thick black line. The dashed red line and the dashed dotted blue line are the SVM and the DWD classification boundaries respectively before the perturbation. We then move one observation in the positive group slightly (from the solid triangle to the solid diamond as shown in the figure). This perturbation leads to a visible change of the SVM direction (shown as the dotted red line), but a smaller change for DWD (shown as the solid blue line). Note that all four hyperplanes are capable of classifying this training

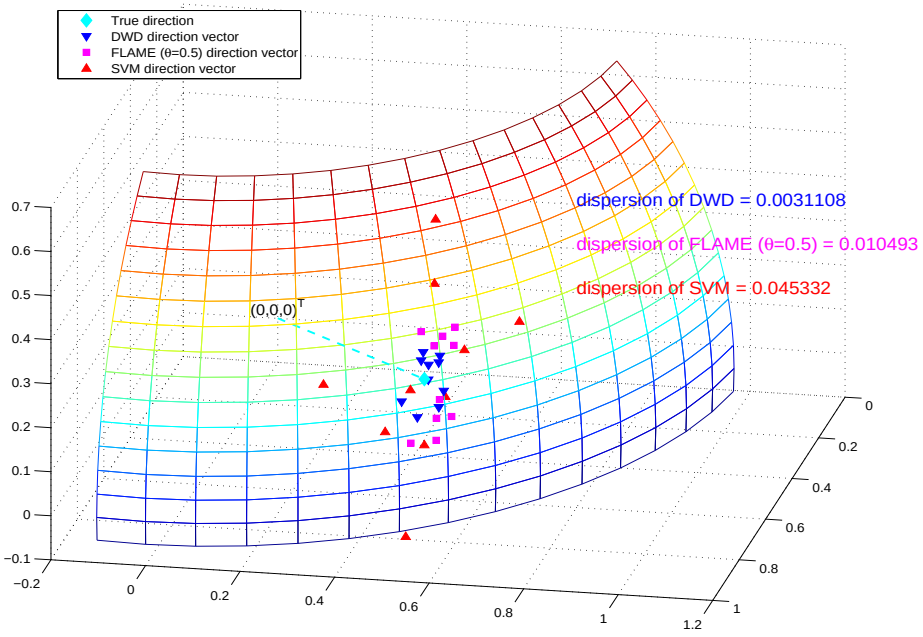


Figure 1: The true population mean difference direction vector (the cyan dashed line and diamond marker; equivalent to the Bayes rule direction), the DWD directions (blue down-pointing triangles), the FLAME directions with $\theta = 0.5$ (magenta squares), and the SVM directions (red up-pointing triangles) for 10 realizations of simulated data. Each direction vector has norm 1 and thus is depicted as a point on the 3D unit sphere. On average, all machines have their discriminant direction vectors scattering around the true direction. The DWD directions are the closest to the true direction and have the smallest variation. The SVM directions have the largest variation and are farthest from the true direction. The variation of the intermediate FLAME direction vectors is between the two machines above. The variation (dispersion) of a machine is also measured by the trace of the sample covariance calculated from the 10 resulting direction vectors for the 10 simulations.

data set perfectly. But it may not be true for an out-of-sample test set. This example shows the unstableness of SVM.

2.2 Sensitivity to Imbalanced Data

In the last subsection, we have shown that DWD outperforms SVM in estimating the discrimination direction, that is, DWD directions are closer to the Bayes rule discrimination directions and usually have a smaller variability. However, it was found that the location of DWD classification boundary, which is characterized by the intercept β, is sensitive to the sample size ratio between the two classes (Qiao et al., 2010).

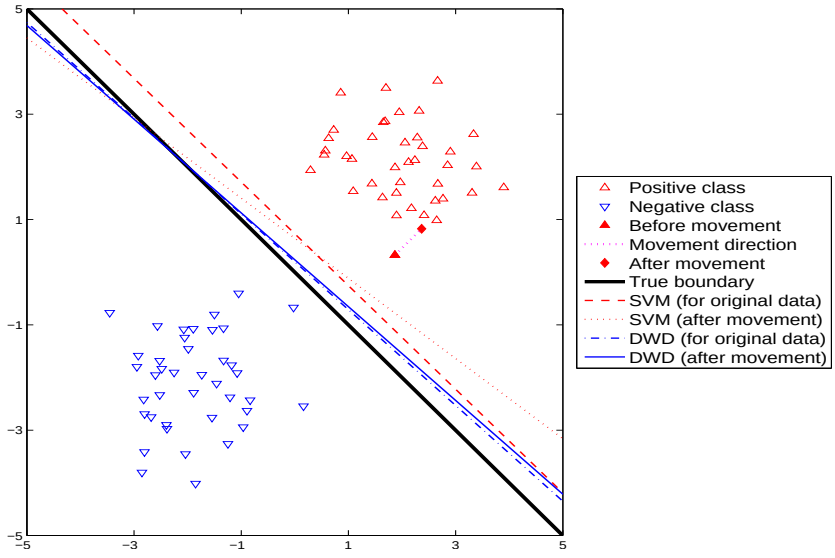


Figure 2: A 2D example shows that the unstable SVM boundary has changed due to a small turbulence of a support vector (the solid red triangle and diamond) while the DWD boundary remains almost still.

Usually, a good discriminant direction $\boldsymbol{\omega}$ helps to reveal the profiling difference between two classes of populations. But the classification/prediction performance heavily depends on the location coefficient β. We define the *imbalance factor* $m \geq 1$ as the sample size ratio between the majority class and the minority class. It turns out that β in the SVM classifier is not sensitive to m. However, the β term for the DWD method is very sensitive to m. We also notice that, as a consequence, the DWD separating hyperplane will be pushed toward the minority class, when the ratio m is close to infinity, that is, DWD classifiers intend to ignore the minority class. In this section, we use another toy example to better illustrate the impact of the imbalanced data on both the estimated β and the classification performance.

Figure 3 uses a one-dimensional example, so that estimating $\boldsymbol{\omega}$ is not needed. This also corresponds to a multivariate data set, where $\boldsymbol{\omega}$ is estimated correctly first, after which the data set is projected to $\boldsymbol{\omega}$ to form the one-dimensional data. In this plot, the x-coordinates of the red dots and the blue dots are the values of the data while the y-coordinates are random jitters for better visualization. The red and blue curves are the kernel density estimations for both classes. In the top subplot of Figure 3, where $m = 1$ (that is, the balanced data), both the DWD (blue lines) and SVM (red lines) boundaries are close to the Bayes rule boundary (black solid line), which sits at 0. In the bottom subplot, the sample size of the red class is tripled, which corresponds to $m = 3$. Note that the SVM boundary moves a little towards the minority (blue) class, but still fairly close to the true boundary. The DWD boundary, however, is pushed towards the minority. Although this does not impose immediate problems for the training data set, the DWD classifier will suffer from a great loss of classification performance when it is applied to an out-of-sample data set. It can be shown that when m goes to infinity, the DWD classification boundary will tends to

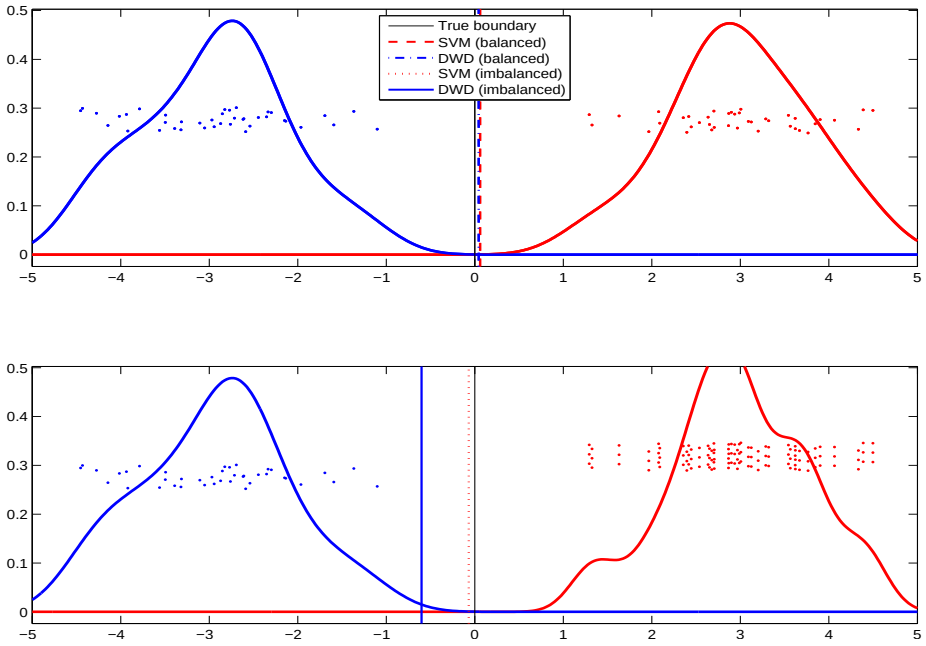


Figure 3: A 1D example shows that the DWD boundary is pushed towards the minority class (blue) when the majority class (red) has tripled its sample size.

negative infinity, which totally ignores the minority group (see our Theorem 4). However, SVM will not suffer from the imbalanced data issue. One reason is that SVM only needs a small fraction of data (called support vectors) to estimate both $\boldsymbol{\omega}$ and β, which mitigate the imbalanced data issue naturally.

Imbalanced data issues have been investigated in both statistics and machine learning. See an extensive survey in Chawla et al. (2004). Recently, Owen (2007) studied the asymptotic behavior of infinitely imbalanced binary logistic regression. In addition, Qiao and Liu (2009) and Qiao et al. (2010) proposed to use adaptive weighting approaches to overcome the imbalanced data issue.

In summary, the performance of DWD and SVM is different in the following ways: 1) The SVM direction usually has a larger variation and deviates farther from the Bayes rule direction than the DWD direction, which are indicators of overfitting HDLSS data. 2) The SVM intercept is not sensitive to the imbalanced data, but the DWD intercept is. These observations have motivated us to investigate their similarity and differences. In the next section, a new family of classifier will be proposed, which unifies the above two classifiers.

3. FLAME Family

In this section, we introduce FLAME, a family of classifiers, through a thorough investigation of the loss functions of SVM and DWD in Section 3.1. The formulation and implementation of the FLAME classifiers are given in Section 3.2.

3.1 SVM and DWD Loss Functions

The key factors that drive the very distinct performances of the SVM and the DWD methods are their associated loss functions (see Figure 4.)

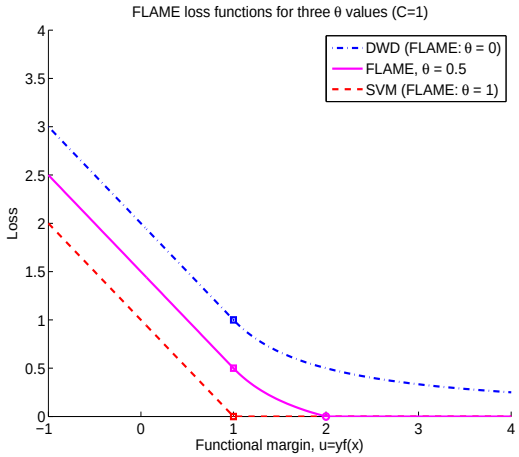


Figure 4: FLAME loss functions for three θ values: $\theta = 0$ (equivalent to SVM/Hinge loss), $\theta = 0.5$, $\theta = 1$ (equivalent to DWD). The parameter C is set to be 1.

Figure 4 displays the loss functions of SVM, DWD and FLAME with some specific tuning parameters. SVM uses the Hinge loss function, $H(u) = (1-u)_+$ (the red dashed curve in Figure 4), where u corresponds to the functional margin $u \equiv yf(\boldsymbol{x})$. Note that the functional margin u can be viewed as the distance of vector $\boldsymbol{x}$ from the separating hyperplane (defined by $\{\boldsymbol{x} : f(\boldsymbol{x}) = 0\}$). When $u > 0$ and is large, the data vector is correctly classified and is far from the separating hyperplane; when $u < 0$, the data vector is wrongly classified. Note that when $u > 1$, the corresponding Hinge loss equals zero. Thus, only those observations with $u \leq 1$ contribute to the estimation of $\boldsymbol{\omega}$ and β. These observations are called *support vectors*. Hence, SVM is insensitive to the observations that are far away from the decision boundary, which is the reason that it is less sensitive to the imbalanced data issue. However, the influence by only the support vectors makes the SVM solution subject to overfitting (data-piling). This can be explained by the following: in the optimization process of SVM, the functional margins for the vectors are pushed towards a region with small loss, that is, functional margins u are encouraged to be large. But once a vector is pushed to the point where $u = 1$, the optimization mechanism lacks further incentive to continue pushing it towards a larger function margin as the Hinge loss cannot be further reduced for this vector. Therefore many data vectors are piling along the hyperplane corresponding to $u = 1$. Data-piling is bad for generalization because a small turbulence to the support vectors could lead to a big difference of the estimated discriminant direction vector (recall the examples in Section 2.1).

The DWD method corresponds to a different DWD loss function,

$$V(u) = \begin{cases} 2\sqrt{C} - Cu & \text{if } u \le \frac{1}{\sqrt{C}}, \\ 1/u & \text{otherwise.} \end{cases} \tag{1}$$

Here C is a pre-defined constant. Figure 4 shows the DWD loss function with $C = 1$. It is clear that the DWD loss function is very similar to the SVM loss function when u is small (both are linearly decreasing with respect to u). The major difference is that the DWD loss is always positive. This property will make the DWD method behave in a very different way than SVM. As there is always an incentive to make the function margin to be larger (and the loss to be smaller), the DWD loss function kills data-piling, and mitigates the overfitting issue for HDLSS data.

On the other hand, the DWD loss function makes the DWD method very sensitive to the imbalanced data issue. This is because now that each observation will have some influence, the larger class will have a larger influence. The decision boundary of the DWD method tends to ignore the smaller class, because sacrificing the smaller class (boundary being closer to the smaller class and farther from the larger class) can lead to a dramatic reduction of the loss from the larger class, which ultimately lead to a minimized overall loss.

3.2 FLAME

We propose to borrow strengths from both methods to simultaneously deal with both the imbalanced data and the overfitting (data-piling) issues. We first highlight the connections between the DWD loss and an modified version of the Hinge loss (of SVM). Then we modify the DWD loss so that samples far from the classification boundary will have zero loss.

Let $f(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{\omega} + \beta$. The formulation of SVM can be rewritten (see details in Appendix A) in the form of $\underset{\boldsymbol{\omega},\beta}{\operatorname{argmin}} \sum_i H^*(y_i f(\boldsymbol{x}_i))$, s.t. $\|\boldsymbol{\omega}\|^2 \le 1$ where the modified Hinge loss function H^* is defined as

$$H^*(u) = \begin{cases} \sqrt{C} - Cu & \text{if } u \le \frac{1}{\sqrt{C}}, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

Comparing the DWD loss (1) and this modified Hinge loss (2), one can easily see their connections: for $u \le \frac{1}{\sqrt{C}}$, the DWD loss is greater than the Hinge loss of SVM by an exact constant $\sqrt{C}$, and for $u > \frac{1}{\sqrt{C}}$, the DWD loss is $1/u$ while the SVM Hinge loss equals 0. Clearly the modified Hinge loss (2) is the result of soft-thresholding the DWD loss at $\sqrt{C}$. In other words, SVM can be seen as a special case of DWD where the losses of those vectors with $u = y_i f(\boldsymbol{x}_i) > 1/\sqrt{C}$ are shrunken to zero. To allow different levels of soft-thresholding, we propose to use a new loss function which (soft-)thresholds the DWD loss function by constant $\theta\sqrt{C}$ where $0 \le \theta \le 1$, that is, a fraction of $\sqrt{C}$. The new loss function is

$$L(u) = \left[V(u) - \theta\sqrt{C}\right]_+ = \begin{cases} (2-\theta)\sqrt{C} - Cu & \text{if } u \le \frac{1}{\sqrt{C}}, \\ 1/u - \theta\sqrt{C} & \text{if } \frac{1}{\sqrt{C}} \le u < \frac{1}{\theta\sqrt{C}}, \\ 0 & \text{if } u \ge \frac{1}{\theta\sqrt{C}}, \end{cases} \tag{3}$$

that is, to reduce the DWD loss by a constant, and truncate it at 0. The magenta solid curve in Figure 4 is the FLAME loss when $C = 1$ and $\theta = 0.5$. This simple but useful modification unifies the DWD and SVM methods. When $\theta = 1$, the new loss function (when $C = 1$) reduces to the SVM Hinge loss function; while when $\theta = 0$, it remains as the DWD loss.

Note that $L(u) = 0$ for $u > 1/(\theta\sqrt{C})$. Thus, those data vectors with very large functional margins will still have zero loss. For DWD loss (corresponding to $\theta = 0$), note that $1/(\theta\sqrt{C}) = \infty$. Thus no data vector can have zero loss. For SVM loss, all the data vector with $u > 1/(\theta\sqrt{C}) = 1/\sqrt{C}$ will have zero loss. Training a FLAME classifier with $0 < \theta < 1$ can be interpreted as sampling a portion of data which are farther from the boundary than $1/\theta\sqrt{C}$ and assign zero loss to them. Alternatively, it can be viewed as sampling data that are closer to the boundary than $1/\theta\sqrt{C}$ and assign positive loss to them. Note that the larger θ is, the fewer data are sampled to have positive loss. As one can flexibly choose θ, the new classification method with this new loss function is called the FLexible Assortment MachinE (FLAME).

FLAME can be implemented by a Second-Order Cone Programming algorithm (Toh et al., 1999; Tütüncü et al., 2003). Let $\theta \in [0, 1]$ be the FLAME parameter. The proposed method minimizes $\min_{\boldsymbol{\omega},b,\boldsymbol{\xi}} \sum_{i=1}^{n} \left(\frac{1}{r_i} + C\xi_i - \theta\sqrt{C} \right)_+$. A slack variable $\varphi_i \geq 0$ can be introduced to absorb the $(\cdot)_+$ function. The optimization of the FLAME can be written as

$$\begin{aligned} &\min_{\boldsymbol{\omega},b,\boldsymbol{\xi}} \sum_i \varphi_i, \\ \text{s.t.} \quad &\left(\frac{1}{r_i} + C\xi_i - \theta\sqrt{C} \right) - \varphi_i \leq 0, \ \varphi_i \geq 0, \\ &r_i = y_i(\boldsymbol{x}_i^T \boldsymbol{\omega} + \beta) + \xi_i, \ r_i \geq 0 \text{ and } \xi_i \geq 0, \\ &\|\boldsymbol{\omega}\|^2 \leq 1. \end{aligned}$$

A MATLAB routine has been implemented and is available at the authors' personal websites. See Online Appendix 1 for more details on the implementation.

4. Choice of Parameters

There are two tuning parameters in the FLAME model: one is the C, inherited from the DWD loss, which controls the amount of allowance for misclassification; the other is the FLAME parameter θ, which controls the level of soft-thresholding. Similar to the discussion in DWD (Marron et al., 2007), the classification performance of FLAME is insensitive to different values of C. In addition, it can be shown for any C, FLAME is Fisher consistent, by applying the general results in Lin (2004). Thus, the default value for C as proposed in Marron et al. (2007) will be used in FLAME. As the property and the performance of FLAME depends on the choice of the θ parameter, it is important to select the right amount of thresholding. In this section, we introduce a way of choosing the second parameter θ, which is motivated by a theoretical consideration and is heuristically meaningful as well.

Having observed that the DWD discrimination direction is usually closer to the Bayes rule direction, but its location term β is sensitive to the imbalanced data issue, we propose

the following data-driven approach to select an appropriate θ. Without loss of generality, we assume that the negative class is the majority class with sample size n_- and the positive class is the minority class with sample size n_+. We point out that the main reason that DWD is sensitive to the imbalanced data issue is that it uses all vectors in the *majority* class to build up a classifier. A heuristic strategy to correct this would be to force the optimization to use the same number of vectors from both classes (so as to mimic a balanced data set) to build up a classifier: we first apply DWD to the data set, and calculate the distances of all data in the majority (negative) class to the current DWD classification boundary; we then train FLAME with a carefully chosen parameter θ which assigns positive loss to the closet $n_+(< n_-)$ data vectors in the majority (negative) class to the classification boundary. As a consequence, each class will have exactly n_+ vectors which have positive loss. In other words, while keeping the least imbalance (because we have the same numbers of vectors from both classes that have influence over the optimization), we obtain a model with the least possible overfitting (because $2n_+$ vectors have influence, instead of only the limited support vectors as in SVM.)

In practice, since the new FLAME classification boundary using the θ value chosen above may be different from the initial DWD classification boundary, the n_+ closest points to the FLAME classification boundary may not be the same n_+ closest points to the DWD boundary. This means that it is not guaranteed that exactly n_+ points from the majority class will have positive loss. However, one can expect that reasonable approximation can be achieved. Moreover, an iterative scheme for finding θ is introduced as follows in order to minimize such discrepancy.

For simplicity, we let $(\boldsymbol{x}_i, y_i)$ with index i be an observation from the positive/minority class and $(\boldsymbol{x}_j, y_j)$ with index j be an observation from the negative/majority class.

Algorithm 1 ***(Adaptive parameter)***

1. *Initiate* $\theta_0 = 0$.
2. *For* $k = 0, 1, \cdots,$
   (a) *Solve FLAME solutions* $\boldsymbol{\omega}(\theta_k)$ *and* $\beta(\theta_k)$ *given parameter* θ_k.
   (b) *Let* $\theta_{k+1} = \max\left(\theta_k,\ \left\{g_{(n_+)}(\theta_k)\sqrt{C}\right\}^{-1}\right)$, *where* $g_j(\theta_k)$ *is the functional margin* $u_j \equiv y_j(\boldsymbol{x}_j^T\boldsymbol{\omega}(\theta_k) + \beta(\theta_k))$ *of the* j*th vector in the negative/majority class and* $g_{(l)}(\theta_k)$ *is the* l*th order statistic of these functional margins.*
3. *When* $\theta_k = \theta_{k-1}$, *the iteration stops.*

The goal of this algorithm is to make $g_{(n_+)}(\theta_k)$ to be the greatest functional margin among all the data vectors that have positive loss in the negative/majority class. To achieve this, we calibrate θ by aligning $g_{(n_+)}(\theta_k)$ to the turning point $u = 1/(\theta\sqrt{C})$ in the definition of the FLAME loss (3), that is $g_{(n_+)}(\theta_k) = 1/(\theta\sqrt{C}) \Rightarrow \theta = \left(g_{(n_+)}(\theta_k)\sqrt{C}\right)^{-1}$.

We define the equivalent sample objective function of FLAME for the iterative algorithm above, $s(\boldsymbol{\omega}, \beta, \theta) = \dfrac{1}{n_+ + n_-}\left[\sum_{i=1}^{n_+} L((\boldsymbol{x}_i^T\boldsymbol{\omega} + \beta), \theta) + \sum_{j=1}^{n_-} L(-(\boldsymbol{x}_j'\boldsymbol{\omega} + \beta), \theta)\right] + \dfrac{\lambda}{2}\|\boldsymbol{\omega}\|^2$. Then the convergence of this algorithm is shown in Theorem 1. The proofs of all the theorems and propositions in this article are included in Online Appendix 1.

Theorem 1 *In Algorithm 1, $s(\boldsymbol{\omega}_k, \beta_k, \theta_k)$ is non-increasing in k. As a consequence, Algorithm 1 converges to a stationary point $s(\boldsymbol{\omega}_\infty, \beta_\infty, \theta_\infty)$ where $s(\boldsymbol{\omega}_k, \beta_k, \theta_k) \geq s(\boldsymbol{\omega}_\infty, \beta_\infty, \theta_\infty)$. Moreover, Algorithm 1 terminates finitely.*

Ideally, one would hope to get an optimal parameter θ^* which satisfies $\theta^* = \left(g_{(n_+)}(\theta^*)\sqrt{C}\right)^{-1}$. In practice, θ_∞ will approximate θ^* very well. In addition, we notice that one-step iteration usually gives decent results for simulation examples and some real examples.

5. Theoretical Properties

In this section, several important theoretical properties of the FLAME classifier are investigated. We first prove the Fisher consistency (Lin, 2004) of the FLAME in Section 5.1. As one focus of this paper is imbalanced data classification, the asymptotic properties for FLAME under extremely imbalanced data setting is studied in Section 5.2. Lastly, a novel HDLSS asymptotics where n is fixed and $d \to \infty$, the other focus of this article, is studied in Section 5.3.

5.1 Fisher Consistency and Large Sample Asymptotics

Fisher consistency is a very basic property for a classifier. A classifier is Fisher consistent implies that the minimizer of the conditional risk of the classifier given observation $\boldsymbol{x}$ has the same sign as the Bayes rule, $\underset{k\in\{+1,-1\}}{\operatorname{argmax}} \mathrm{P}(Y = k|\boldsymbol{X} = \boldsymbol{x})$. It has been shown that both SVM and DWD are Fisher consistent (Lin, 2004; Qiao et al., 2010). The following proposition states that the FLAME classifier is Fisher consistent too.

Proposition 2 *Let f^* be the global minimizer of the expected loss $\mathbb{E}[L(Yf(\boldsymbol{X}), \theta)]$, where $L(\cdot)$ is the loss function for the FLAME classifier, given parameters C and θ. Then* $\operatorname{sign}(f^*(\boldsymbol{x})) = \operatorname{sign}(\mathrm{P}(Y = +1|\boldsymbol{X} = \boldsymbol{x}) - 1/2)$.

Fisher consistency is also known as classification-calibrated, notably by Bartlett et al. (2006). With this weakest possible condition on the loss function, they extended the results of Zhang (2004) and showed that there was a nontrivial upper bound on the excess risk. Moreover, they were able to derive faster rates of convergence in some low noise settings. In particular, for a classification-calibrated loss function $L(\cdot)$, there exists a function $\psi : [-1, 1] \mapsto [0, \infty)$ so that $\psi(R_{0-1}(f) - R^*_{0-1}) \leq R_L(f) - R^*_L$ or $c(R_{0-1}(f) - R^*_{0-1})^\alpha \psi\left(\frac{(R_{0-1}(f)-R^*_{0-1})^\alpha}{2c}\right) \leq R_L(f) - R^*_L$ for some constant $c > 0$ with certain low noise parameter α, where $R_{0-1}(f)$ and $R_L(f)$ are the risk of the prediction function f with respect to the 0-1 loss and the loss function L respectively, and R^*_{0-1} and R^*_L are the corresponding Bayes risk and "optimal L-risk" respectively. The techniques in Zhang (2004) and Bartlett et al. (2006) can be directly applied to the FLAME classifier. The form of the ψ transform above which establishes the relations between the two excess risks, being applied to the current article, is given by Proposition 3.

Proposition 3 *The ψ-transform of the FLAME loss function with parameters C and θ is*

$$\psi(\gamma) = (2 - \theta)\sqrt{C} - H((1 + \gamma)/2),$$

where

$$H(\eta) = \begin{cases} \sqrt{C}\min(\eta, 1-\eta)(2+\frac{1}{\theta}-\theta), \ \textit{if } \eta < \frac{\theta^2}{1+\theta^2} \ \textit{ or } \eta > \frac{1}{1+\theta^2}, \\ \sqrt{C}[2\min(\eta, 1-\eta) - \theta + 2\sqrt{\eta(1-\eta)}], \ \textit{otherwise.} \end{cases} \tag{4}$$

These results provide bounds for the excess risk $R_{0-1}(f) - R_{0-1}^*$ in terms of the excess L-risk $R_L(f) - R_L^*$. Combined with a bound on the excess L-risk, they can give us a bound on the excess risk. Recent works for SVM have focused on fast rates of convergence. Vito et al. (2005) studied classification problems as inverse problems; Steinwart and Scovel (2007) studied the convergence properties of the standard SVM with Gaussian kernels; Blanchard et al. (2008) used a method called "localization". See also Chen et al. (2004) for another relevant work for the q-soft margin SVM.

5.2 Asymptotics under Imbalanced Setting

In this subsection, we investigate the asymptotic performance of SVM, DWD and FLAME. The asymptotic setting we focus on is when the minority sample size n_+ is fixed and the majority sample size $n_- \to \infty$, which is similar to the setting in Owen (2007). We will show that DWD is sensitive to the imbalanced data, while FLAME with proper choices of parameter θ and SVM are not.

Let $\overline{\boldsymbol{x}}_+$ be the sample mean of the positive/minority class. Theorem 4 shows that in the imbalanced data setting, when the size of the negative/majority class grows while that of the positive/minority class is fixed, the intercept term for DWD tends to negative infinity, in the order of $\sqrt{m}$. Therefore, DWD will classify all the observations to the negative/majority class, that is, the minority class will be 100% misclassified.

Theorem 4 *Let n_+ be fixed. Assume that the conditional distribution of the negative majority class $F_-(\boldsymbol{x})$ surrounds $\overline{\boldsymbol{x}}_+$ by the definition given in Owen (2007), and that γ is a constant satisfying $\inf_{\|\boldsymbol{\omega}\|=1} \int_{(\boldsymbol{x}-\overline{\boldsymbol{x}}_+)'\boldsymbol{\omega}>0} dF_-(\boldsymbol{x}) > \gamma \geq 0$, then the DWD intercept $\widehat{\boldsymbol{\beta}}$ satisfies*

$$\widehat{\boldsymbol{\beta}} < -\sqrt{\frac{\gamma}{C}m} - \overline{\boldsymbol{x}}_+^T\boldsymbol{\omega} = -\sqrt{\frac{n_-\gamma}{n_+C}} - \overline{\boldsymbol{x}}_+^T\boldsymbol{\omega}.$$

In Section 4, we have introduced an iterative approach to select the parameter θ. Theorem 5 shows that with the optimal parameter θ^* found by Algorithm 1, the discriminant direction of FLAME is in the same direction of the vector that joins the sample mean of the positive class and the *tilted* population mean of the negative class. Moreover, in contrast to DWD, the intercept term of FLAME in this case is finite.

Theorem 5 *Suppose that $n_- \gg n_+$ and $\boldsymbol{\omega}^*$ and β^* are the FLAME solutions trained with the parameter θ^* that satisfies $\theta^* = \left(g_{(n_+)}(\theta^*)\sqrt{C}\right)^{-1}$. Then $\boldsymbol{\omega}^*$ and β^* satisfy that*

$$\boldsymbol{\omega}^* = \frac{C}{(1+m)\lambda}\left[\overline{\boldsymbol{x}}_+ - \frac{\int(\boldsymbol{x}^T\boldsymbol{\omega}^* + \beta^*)^{-2}\boldsymbol{x}dF_-(\boldsymbol{x} \mid E)}{\int(\boldsymbol{x}^T\boldsymbol{\omega}^* + \beta^*)^{-2}dF_-(\boldsymbol{x} \mid E)}\right], \tag{5}$$

where E is the event that $[Y(\boldsymbol{X}^T\boldsymbol{\omega}^* + \beta^*)]^{-1} \geq \theta^*\sqrt{C}$ where $(\boldsymbol{X}, Y)$ is a random sample from the negative/majority class, and that

$$\int (\boldsymbol{x}^T\boldsymbol{\omega}^* + \beta^*)^{-2} dF_-(\boldsymbol{x} \mid E) = \frac{n_+}{n^o} C, \text{where } 0 < n^o \leq n_+.$$

Note that event E is $[Y(\boldsymbol{X}^T\boldsymbol{\omega}^* + \beta^*)]^{-1} \geq \theta^*\sqrt{C}$, which implies that the second term in (5) focuses on data vectors in the negative class with positive loss since their functional margins are less than $1/(\theta\sqrt{C})$. Recall that this is precisely the interpretation of FLAME (see Section 3.2), namely, to sample a subset of the majority class to have positive loss, so as to make the problem less imbalanced.

Remark: As a consequence of Theorem 5, when $m = n_-/n_+ \to \infty$, we have $\|\boldsymbol{\omega}^*\| \to 0$. Since the right-hand-side of the last equation above is positive and finite, β^* does not diverge. In addition, since $\mathrm{P}(\overline{E}) \to 1$ with probability converging to 1, $\beta^* < -1/(\theta\sqrt{C})$.

The following theorem shows the performance of SVM under the imbalanced data context, which completes our comparisons between SVM, DWD and FLAME.

Theorem 6 *Suppose that $n_- \gg n_+$. The solutions $\widehat{\boldsymbol{\omega}}$ and $\widehat{\beta}$ to SVM satisfy that*

$$\widehat{\boldsymbol{\omega}} = \frac{1}{(1+m)\lambda} \left\{ \overline{\boldsymbol{x}}_+ - \int \boldsymbol{x} dF_-(\boldsymbol{x} \mid G) \right\},$$

where G is the event that $1 - Y(\boldsymbol{X}^T\widehat{\boldsymbol{\omega}} + \widehat{\beta}) > 0$ where $(\boldsymbol{X}, Y)$ is a random sample from the negative/majority class, and that

$$\mathrm{P}(\overline{G}) = \mathrm{P}(1 + \boldsymbol{X}^T\widehat{\boldsymbol{\omega}} + \widehat{\beta} \leq 0) = 1 - 1/m.$$

Remark: The last statement in Theorem 6 means that with probability converging to 1, $\widehat{\beta} \leq -1$. However, note this is the only restriction that SVM solution has for the intercept term (recall that the counterpart in DWD is $\widehat{\boldsymbol{\beta}} < -\sqrt{\frac{\gamma}{C}m} - \overline{\boldsymbol{x}}_+^T\boldsymbol{\omega}$).

5.3 High-Dimensional, Low-Sample Size Asymptotics

HDLSS data are emerging in many areas of scientific research. The HDLSS asymptotics is a recently developed theoretical framework. Hall et al. (2005) gave a geometric representation for the HDLSS data, which can be used to study these new "n fixed, $d \to \infty$" asymptotic properties of binary classifiers such as SVM and DWD. Ahn et al. (2007) weakened the conditions under which the representation holds. Qiao et al. (2010) improved the conditions and applied this representation to investigate the performance of the weighted DWD classifier. Bolivar-Cime and Marron (2013) compared several binary classification methods in the HDLSS setting under the same theoretical framework. The same geometric representation can be used to analyze FLAME. See summary of some previous HDLSS results in Online Appendix 1. We develop the HDLSS asymptotic properties of the FLAME family by providing conditions in Theorem 7 under which the FLAME classifiers always correctly classify HDLSS data.

We first introduce the notations and give some regularity assumptions, then state the main theorem. Let $k \in \{+1, -1\}$ be the class index. For the kth class and given a fixed n_k,

consider a sequence of random data matrices $\boldsymbol{X}_1^k, \boldsymbol{X}_2^k, \cdots \boldsymbol{X}_d^k, \cdots$, indexed by the number of rows d, where each column of $\boldsymbol{X}_d^k$ is a random observation vector from $\mathbb{R}^d$ and each row represents a variable. Assume that each column of $\boldsymbol{X}_d^k$ comes from a multivariate distribution with dimension d and with covariance matrix $\boldsymbol{\Sigma}_d^k$ independently. Let $\lambda_{1,d}^k \geq \cdots \geq \lambda_{d,d}^k$ be the eigenvalues of the covariance, and $\left(\sigma_d^k\right)^2 = d^{-1}\sum_{i=1}^d \lambda_{i,d}^k$ the average eigenvalue. The eigenvalue decomposition of $\boldsymbol{\Sigma}_d^k$ is $\boldsymbol{\Sigma}_d^k = \boldsymbol{V}_d^k \boldsymbol{\Lambda}_d^k \left(\boldsymbol{V}_d^k\right)^T$. We may define the square root of $\boldsymbol{\Sigma}_d^k$ as $\left(\boldsymbol{\Sigma}_d^k\right)^{1/2} = \boldsymbol{V}_d^k \left(\boldsymbol{\Lambda}_d^k\right)^{1/2}$, and the inverse square root $\left(\boldsymbol{\Sigma}_d^k\right)^{-1/2} = \left(\boldsymbol{\Lambda}_d^k\right)^{-1/2}\left(\boldsymbol{V}_d^k\right)^T$. With minimal abuse of notation, let $\mathbb{E}(\boldsymbol{X}_d^k)$ denote the expectation of columns of $\boldsymbol{X}_d^k$. Lastly, the $n^k \times n^k$ *dual* sample covariance matrix is denoted by $\boldsymbol{S}_{D,d}^k = d^{-1}\{\boldsymbol{X}_d^k - \mathbb{E}(\boldsymbol{X}_d^k)\}^T\{\boldsymbol{X}_d^k - \mathbb{E}(\boldsymbol{X}_d^k)\}$.

Assumption 1 *There are five components:*

(i) Each column of $\boldsymbol{X}_d^k$ has mean $\mathbb{E}(\boldsymbol{X}_d^k)$ and the covariance matrix $\boldsymbol{\Sigma}_d^k$ of its distribution is positive definite.

(ii) The entries of $\boldsymbol{Z}_d^k \equiv \left(\boldsymbol{\Sigma}_d^k\right)^{-\frac{1}{2}}\{\boldsymbol{X}_d^k - \mathbb{E}(\boldsymbol{X}_d^k)\} = \left(\boldsymbol{\Lambda}_d^k\right)^{-\frac{1}{2}}\left(\boldsymbol{V}_d^k\right)^T\{\boldsymbol{X}_d^k - \mathbb{E}(\boldsymbol{X}_d^k)\}$ are independent.

(iii) The fourth moment of each entry of each column is uniformly bounded by $M > 0$ and the Wishart representation holds for each dual sample covariance matrix $\boldsymbol{S}_{D,d}^k$ associated with $\boldsymbol{X}_d^k$, that is,

$$d\boldsymbol{S}_{D,d}^k = \left\{\left(\boldsymbol{Z}_d^k\right)^T\left(\boldsymbol{\Lambda}_d^k\right)^{1/2}\left(\boldsymbol{V}_d^k\right)^T\right\}\left\{\boldsymbol{V}_d^k\left(\boldsymbol{\Lambda}_d^k\right)^{1/2}\boldsymbol{Z}_d^k\right\} = \sum_{i=1}^d \lambda_{i,d}^k \boldsymbol{W}_{i,d}^k,$$

where $\boldsymbol{W}_{i,d}^k \equiv \left(Z_{i,d}^k\right)^T Z_{i,d}^k$ and $Z_{i,d}$ is the ith row of $\boldsymbol{Z}_d^k$ defined above. It is called Wishart representation because if $\boldsymbol{X}_d^k$ is Gaussian, then each $\boldsymbol{W}_{i,d}^k$ follows the Wishart distribution $\mathcal{W}_{n^k}(1, \mathbf{I}_{n^k})$ independently.

(iv) The eigenvalues of $\boldsymbol{\Sigma}_d^k$ are sufficiently diffused, in the sense that

$$\epsilon_d^k = \frac{\sum_{i=1}^d (\lambda_{i,d}^k)^2}{(\sum_{i=1}^d \lambda_{i,d}^k)^2} \to 0 \quad as \quad d \to \infty. \tag{6}$$

(v) The sum of the eigenvalues of $\boldsymbol{\Sigma}_d^k$ is the same order as d, in the sense that $\left(\sigma_d^k\right)^2 = O(1)$ and $1/\left(\sigma_d^k\right)^2 = O(1)$.

Assumption 2 *The distance between the two population expectations satisfies,*

$$d^{-1}\left\|\mathbb{E}(\boldsymbol{X}_d^{(+1)}) - \mathbb{E}(\boldsymbol{X}_d^{(-1)})\right\|^2 \to \mu^2, \text{ as } d \to \infty.$$

Moreover, there exist constants σ^2 and τ^2, such that

$$\left(\sigma_d^{(+1)}\right)^2 \to \sigma^2, \text{ and } \left(\sigma_d^{(-1)}\right)^2 \to \tau^2.$$

Let $\nu^2 \equiv \mu^2 + \sigma^2/n_+ + \tau^2/n_-$. The following theorem gives the sure classification condition for FLAME, which includes SVM and DWD as special cases.

Theorem 7 *Without loss of generality, assume that $n_+ \le n_-$. The situation of $n+ > n_-$ is similar and omitted.*

- *If either one of the following three conditions is satisfied,*
  1. *for $\theta \in \left[0, (1+\sqrt{m^{-1}})/(\nu\sqrt{dC})\right)$, $\mu^2 > (n_-/n_+)^{\frac{1}{2}}\sigma^2/n_+ - \tau^2/n_- > 0$;*
  2. *for $\theta \in \left[(1+\sqrt{m^{-1}})/(\nu\sqrt{dC}), 2/(\nu\sqrt{dC})\right)$, $\mu^2 > T - \tau^2/n_- > 0$ where $T := \left(1/(2\theta\sqrt{dC}) + \sqrt{1/(4\theta^2 dC) + \sigma^2/n_+}\right)^2 - \sigma^2/n_+$;*
  3. *for $\theta \in \left[2/(\nu\sqrt{dC}), 1\right]$, $\mu^2 > \sigma^2/n_+ - \tau^2/n_- > 0$,*

  then for a new data point $\boldsymbol{x}_0^+$ from the positive class ($+1$),
  $\mathrm{P}(\boldsymbol{x}_0^+$ is correctly classified by FLAME$) \to 1$, as $d \to \infty$.
  Otherwise, the probability above $\to 0$.
- *If either one of the following three conditions is satisfied,*
  1. *for $\theta \in \left[0, (1+\sqrt{m^{-1}})/(\nu\sqrt{dC})\right)$, $(n_-/n_+)^{\frac{1}{2}}\sigma^2/n_+ - \tau^2/n_- > 0$;*
  2. *for $\theta \in \left[(1+\sqrt{m^{-1}})/(\nu\sqrt{dC}), 2/(\nu\sqrt{dC})\right)$, $T - \tau^2/n_- > 0$;*
  3. *for $\theta \in \left[2/(\nu\sqrt{dC}), 1\right]$, $\sigma^2/n_+ - \tau^2/n_- > 0$,*

  then for any $\mu > 0$, for a new data point $\boldsymbol{x}_0^-$ from the negative class (-1),
  $\mathrm{P}(\boldsymbol{x}_0^-$ is correctly classified by FLAME$) \to 1$, as $d \to \infty$.

Remark: Theorem 7 has two parts. The first part gives the conditions under which FLAME correctly classifies a new data point from the positive class, and the second part is for the negative class. Each part lists three conditions based on three disjoint intervals of parameter θ. Note the first and third intervals of each part generalize results which were shown to hold only for DWD and SVM before (*c.f.* Theorem 1 and Theorem 2 in Hall et al., 2005). In particular, it shows that all the FLAME classifiers with θ falling into the first interval behave like DWD asymptotically. Similarly, all the FLAME classifiers with θ falling into the third interval behave like SVM asymptotically. This partially explains the shape of the within-group error curve that we will show in Figure 6 (see also Figures A.2 and A.3 in Online Appendix 1), which we will discuss in the next section.

In the first part, the condition for other FLAMEs (with θ in the second interval) is weaker than the DWD-like FLAMEs (in the first interval), but stronger than the SVM-like FLAMEs (in the third interval). This means that it is easier to classify a new data point from the positive/minority class by SVM, than by an intermediate FLAME, which is easier than by DWD. Note that when $n_+ \le n_-$, the hyperplane for FLAME is in general closer to the positive class.

In terms of classifying data points from the negative class, the order of the difficulties among DWD, FLAME and SVM reverses.

6. Simulations

FLAME is not only a unified representation of DWD and SVM, but also introduces a new family of classifiers which has the potential of avoiding the overfitting HDLSS data issue and the sensitivity to imbalanced data issue. In this section, we use simulations to show the performance of FLAME at various parameter levels.

6.1 Measures of Performance

Before we introduce our simulation examples, we first introduce the performance measures in this paper. Note that the Bayes rule classifier can be viewed as the "gold standard" classifier. In our simulation settings, we assume that data are generated from two multivariate normal distributions with different mean vectors $\boldsymbol{\mu}_+$ and $\boldsymbol{\mu}_-$ and same covariance matrices $\boldsymbol{\Sigma}$. This setting leads to the following Bayes rule,

$$\text{sign}(\boldsymbol{x}^T\boldsymbol{\omega}_B + \beta_B) \text{ where } \boldsymbol{\omega}_B = \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) \text{ and } \beta_B = -\frac{1}{2}(\boldsymbol{\mu}_+ + \boldsymbol{\mu}_-)'\boldsymbol{\omega}_B. \quad (7)$$

Five performance measures are evaluated in this paper:

1. The *mean within-class error* (MWE) for an out-of-sample test set, which is defined as

$$MWE = \frac{1}{2n_+}\sum_{i=1}^{n_+}\mathbb{1}(\widehat{Y}_i^+ \neq Y_i^+) + \frac{1}{2n_-}\sum_{j=1}^{n_-}\mathbb{1}(\widehat{Y}_j^- \neq Y_j^-)$$

2. The *deviation* of the estimated intercept β from the Bayes rule intercept β_B: $|\beta - \beta_B|$.
3. *Dispersion*: a measure of the stochastic variability of the estimated discrimination direction vector $\boldsymbol{\omega}$. The dispersion measure was introduced in Section 1, as the trace of the sample covariance of the resulting discriminant direction vectors: disperson = $\text{Var}([\boldsymbol{\omega}_r]_{r=1:R})$ where R is the number of repeated runs.
4. *Angle* between the estimated discrimination direction $\boldsymbol{\omega}$ and the Bayes rule direction $\boldsymbol{\omega}_B$: $\angle(\boldsymbol{\omega}, \boldsymbol{\omega}_B)$.
5. *RankComp*$(\boldsymbol{\omega}, \boldsymbol{\omega}_B)$: In general, for two direction vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^*$, RankComp is defined as the proportion of the pairs of variables, among all $d(d-1)/2$ pairs, whose relative importances (in terms of their absolute values) given by the two directions are different, that is,

$$\text{RankComp}(\boldsymbol{\omega}, \boldsymbol{\omega}^*) \equiv \frac{1}{d(d-1)/2}\sum_{1\le i<j\le d}\mathbb{1}\left\{(|\omega_i| - |\omega_j|) \times (|\omega_i^*| - |\omega_j^*|) < 0\right\},$$

   where ω_i and ω_i^* are the ith components of the vectors $\boldsymbol{\omega}$ and $\boldsymbol{\omega}^*$ respectively. The RankComp measure can be viewed as a discretized analog to the angle between two vectors, and it provides more insights in the ranking of variables that a direction vector may suggest. We report the RankComp between the estimated direction $\boldsymbol{\omega}$ and the Bayes rule direction $\boldsymbol{\omega}_B$ to measure their closeness.

We will investigate these measures based on different dimensions d and different imbalance factors m.

6.2 Effects of Dimensions and Imbalanced Data

In Section 1, a specific FLAME ($\theta = 0.5$) has been compared with SVM ($\theta = 1$) and DWD ($\theta = 0$) in Figure 1, and on average, its discriminant directions are closer to the Bayes rule direction $\boldsymbol{\omega}_B$ compared to the SVM directions, but are less close than the DWD directions. In this subsection, we will further investigate the performance of FLAME with several

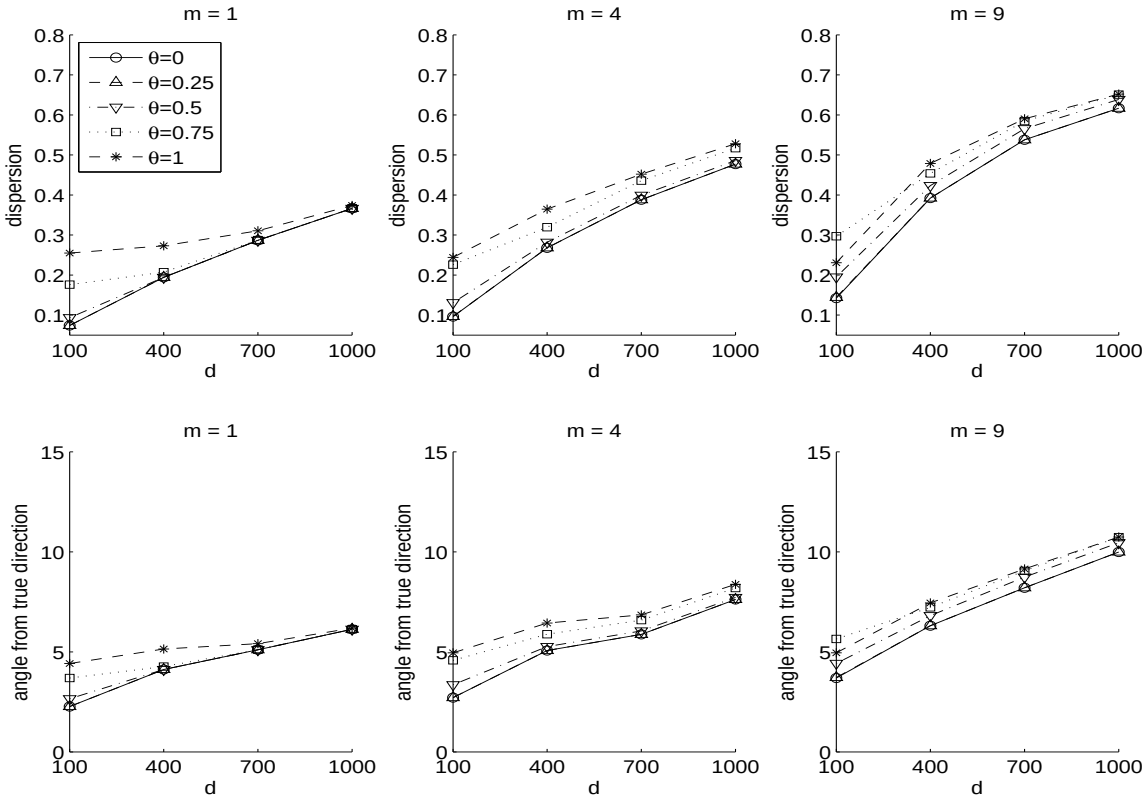


Figure 5: The dispersions (top row) and the angles between the FLAME direction and the Bayes direction (bottom row) for 50 runs of simulations, where the imbalance factors m are 1, 4 and 9 (the left, center and right panels), in the increasing dimension setting ($d = 100, 400, 700, 1000$; shown on the x-axes). The FLAME machines have $\theta = 0, 0.25, 0.5, 0.75, 1$ which are depicted using different curve styles (the first and the last cases correspond to DWD and SVM, respectively.) Note that with θ and the dimension d increase, both the dispersion and the deviation from the Bayes direction increase. The emergence of the imbalanced data (the increase of m) does not much deteriorate the FLAME directions except for large d.

different values of θ, and compare them with DWD and SVM under various simulation settings.

Figure 5 shows the comparison results under the same simulation setting with various combinations of (d, m)'s. In this simulation setting, data are from multivariate normal distributions with identity covariance matrix $MVN_d(\boldsymbol{\mu}_\pm, \mathbf{I}_d)$, where $d = 100, 400, 700$ and 1000. We let $\boldsymbol{\mu}_0 = c(d, d-1, d-2, \cdots, 1)^T$ where $c > 0$ is a constant which scales $\boldsymbol{\mu}_0$ to have norm 2.7. Then we let $\boldsymbol{\mu}_+ = \boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_- = -\boldsymbol{\mu}_0$. The imbalance factor varies among 1, 4 and 9 while the total sample size is 240. For each experiment, we repeat the simulation 50

times, and plot the average performance measure in Figure 5. The Bayes rule is calculated according to (7). It is obvious that when the dimension increases, both the dispersion and the angle increase. They are indicators of overfitting HDLSS data. When the imbalance factor m increases, the two measures increase as well, although not as much as when the dimension increases. More importantly, it shows that when θ decreases (from 1 to 0, or equivalently FLAME changes from SVM to DWD), the dispersion and the angle both decrease, which is promising because it shows that FLAME improves SVM on the overfitting issue.

6.3 Effects of Tuning Parameters with Covariance

We also investigate the effect of different covariance structures, since independence structure among variables as in the last subsection is less common in real applications. We investigate three covariance structures: independent, interchangeable and block-interchangeable covariance. Data are generated from two multivariate normal distributions $MVN_{300}(\boldsymbol{\mu}_{\pm}, \boldsymbol{\Sigma})$ with $d = 300$. We fist let $\boldsymbol{\mu}_1 = (75, 74, 73, \cdots, 1, 0, 0, \cdots, 0)'$, then scale it by multiply a constant c such that the Mahalanobis distance between $\boldsymbol{\mu}_+ = c\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_- = -c\boldsymbol{\mu}_1$ equals 5.4, that is, $(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-)'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-) = 5.4$. Note that this represents a reasonable signal-to-noise ratio.

We consider the FLAME machines with different parameter θ from a grid of 11 values $(0, 0.1, 0.2, \cdots, 1)$, and apply them to nine simulated examples (three different imbalance factors $(m = 2, 3, 4) \times$ three covariance structures). For the independent structure example, $\boldsymbol{\Sigma} = \mathbf{I}_{300}$; For the interchangeable structure example, $\boldsymbol{\Sigma}_{ii} = 1$ and $\boldsymbol{\Sigma}_{ij} = 0.8$ for $i \neq j$; For the block-interchangeable structure example, we let $\boldsymbol{\Sigma}$ be a block diagonal matrix with five diagonal blocks, the sizes of which are 150, 100, 25, 15, 10, and each block is an interchangeable covariance matrix with diagonal entries 1 and off-diagonal entries 0.8.

Figure 6 shows the results of the interchangeable structure example. Since the results under different covariance structures are similar, those for the other two covariance structures are reported in Online Appendix 1 to save space (Figure A.2 for the independent structure, and Figure A.3 for the block-interchangeable covariance).

In each plot, we include the within-group error (top-left), the absolute value of the difference between the estimated intercept and the Bayes intercept $|\beta - \beta_B|$ (top-middle), the angle between the estimated direction and the Bayes direction $\angle(\boldsymbol{\omega}, \boldsymbol{\omega}_B)$ (bottom-left), the RankComp between the estimated direction and the Bayes direction (bottom-middle) and the dispersion of the estimated directions (bottom-right).

We can see that in Figure 6 (and Figures A.2 and A.3 in Online Appendix 1), when we increase θ from 0 to 1, that is, when the FLAME moves from the DWD end to the SVM end, the within-group error decreases. This is mostly due to the fact that the intercept term β comes closer to the Bayes rule intercept β_B. On the other hand, the estimated direction is deviating from the true direction (larger angle), is giving the wrong rank of the variables (larger RankComp), and is more unstable (larger dispersion). Similar phenomena hold for the other two covariance structures, with one exception in the block interchangeable setting (Figure A.3 in Online Appendix 1) where the RankComp first decreases then increases.

In the entire FLAME family, DWD represents one extreme which provides better estimation of the direction, is closer to the Bayes direction, provides the right order for all

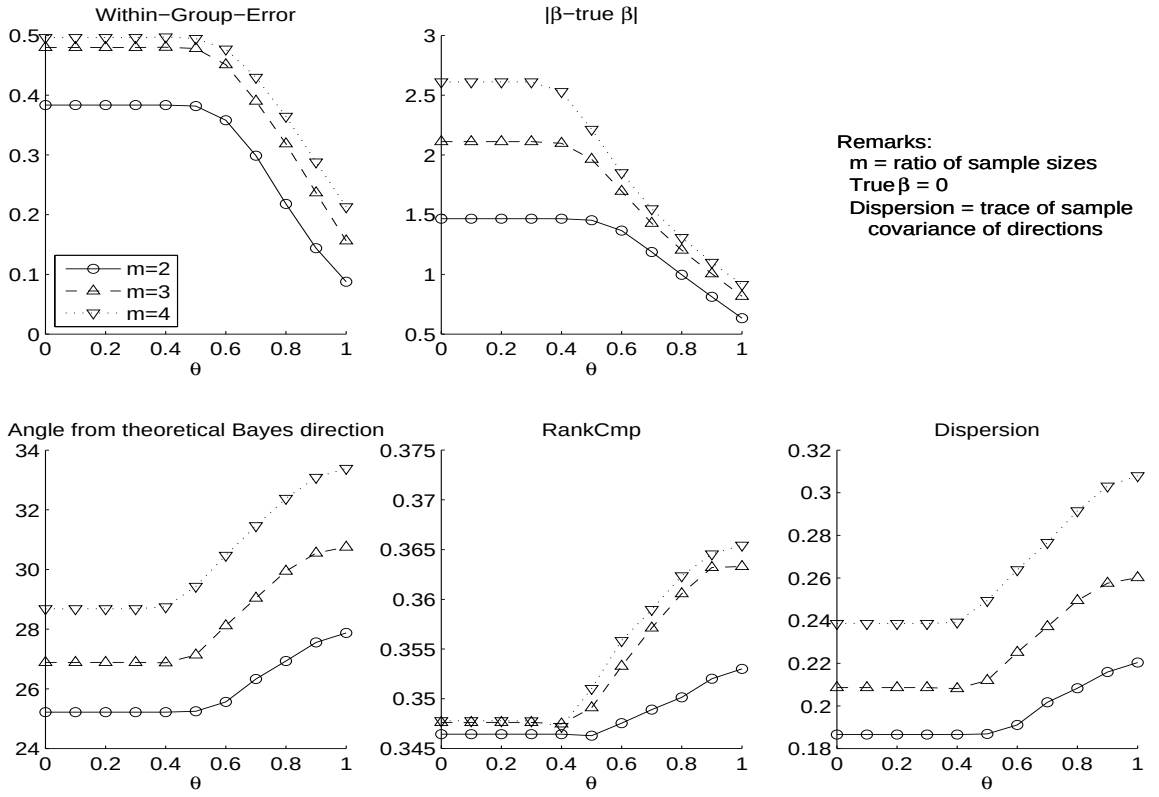


Figure 6: Interchangeable example. It can be seen that with FLAME turns from DWD to SVM (θ from 0 to 1), the within-class error decreases (top-left), thanks to the more accurate estimate of the intercept term (top-middle). On the other hand, this comes at the cost of larger deviation from the Bayes direction (bottom-left), incorrect rank of the importance of the variables (bottom-middle) and larger stochastic variability of the estimation directions (bottom-right).

variables, and is more stable. But it suffers from the inaccurate estimation of β in the presence of imbalanced data; SVM represents the other extreme, which is not sensible to imbalanced data and usually provides a good estimation of β, but is in general outperformed by DWD in terms of closeness to the Bayes optimal direction. In most situations, within the FLAME family, there is no single machine that is better than the both ends from the two aspects at the same time.

7. Real Data Application

In this section we demonstrate the performance of FLAME on a real example: the Human Lung Carcinomas Microarray Data set, which has been analyzed earlier in Bhattacharjee et al. (2001).

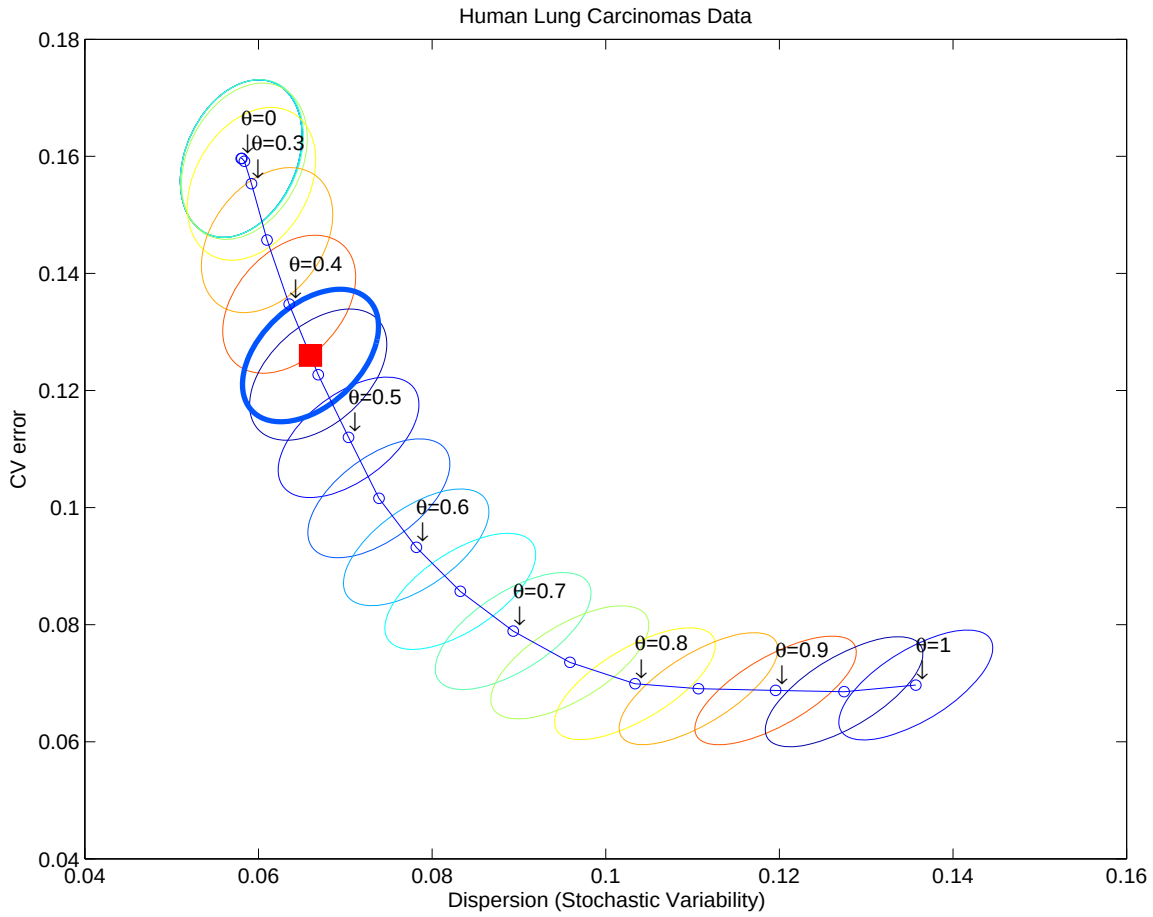


Figure 7: The dispersion and cross-validation error for the Human Lung Carcinomas Data set over 100 random splittings for different choices of θ values. The mean and standard error of the two measurements are depicted by ellipses as detour plots. The red square shows the performance for the adaptive parameter recommendation after one step. This plot shows clear trade-off between generalization error and stochastic variability.

The Human Lung Carcinomas Data set contains six classes: adenocarcinoma, squamous, pulmonary carcinoid, colon, normal and small cell carcinoma, with sample sizes of 128, 21, 20, 13, 17 and 6 respectively. Liu et al. (2008) used this data as a test set to demonstrate their proposed significance analysis of clustering approach. We combine the first two subclasses and the last four subclasses to form the positive and negative classes respectively. The sample sizes are 149 and 56 with imbalance factor $m = 2.66$. The original data contain 12,625 genes. We first filter genes using the ratio of the sample standard deviation and sample mean of each gene and keep 2,530 of them with large ratios (Dudoit et al., 2002; Liu et al., 2008).

We conduct five-fold cross-validations (CV) to evaluate the within-group error for the two classes over 100 random splits. In each split, we apply FLAME with 21 different θ values, ranging from $0, 0.05, 0.1, \ldots$ to 1. Because the true Bayes rule is unknown, we cannot evaluate the RankComp measure or the angle measure. Instead, we calculate the dispersion of the resulting direction vectors when conducting five-fold cross-validation. The adaptive value for θ (after one step) is calculated based on the DWD direction using all the samples in the data set, and the performance of the resulting FLAME is evaluated as well. We report the cross-validation error and the dispersion in a scatter plot.

Figure 7 shows the dispersion and cross-validation error. The mean and standard error of both measurements are depicted by ellipses as detour plots. This plot clears illustrates the existence of the trade-off between generalization error and stochastic variability.

This experiment shows that FLAME opens a new dimension to improve both the classification performance and the interpretative ability of the classifier. In particular, compared to SVM (FLAME with $\theta = 1$), we can probably choose $\theta = 0.7$ or 0.75 so that the stability of the classifier can be much improved at a very small cost of the generalization error. Compared to DWD (FLAME with $\theta = 0$), any increase in θ after 0.3 can lead to dramatic improvement of the cross-validation error, again with very minimal compromise of the stability. The optimal choice of θ seem to be ad-hoc, and depends on the preference of the user. Our adaptive parameter recommendation gives θ at around 0.44.

8. Conclusion and Discussion

In this paper, we thoroughly investigate SVM and DWD on their performance when applied to the HDLSS and imbalanced data. A novel family of binary classifiers called FLAME is proposed, where SVM and DWD are the two ends of the spectrum. On the DWD end, the estimation of the intercept term is deteriorated while it provides better estimation of the direction vector, and thus better handles the HDLSS data. On the hand, SVM is good at estimating the intercept term but not the direction and is subject to overfitting, and thus is more suitable for imbalanced data but not HDLSS data.

We conduct extensive study of the asymptotic properties of the FLAME family in three different flavors, the "d fixed, $n \to \infty$" asymptotics (Fisher consistency), the "d and n_+ fixed, $n_- \to \infty$" asymptotics (extremely imbalanced data), and the "n fixed, $d \to \infty$" asymptotics (the HDLSS asymptotics). These results explain the performance we have seen in the simulations and suggest that with a smart choice of θ, FLAME can properly handle both the HDLSS data and the imbalanced data, by improving the estimations of the direction and the intercept term.

The FLAME family can be immediately extended to multi-class classification, as was done for SVM and DWD such as in Weston and Watkins (1999); Crammer and Singer (2002); Lee et al. (2004) or Huang et al. (2013). Another natural extension is variable selection for FLAME.

The FLAME machines generalize the concepts of support vectors. In SVM, support vectors are referred to vectors that sit on or fall into the two hyperplanes corresponding to $u \leq 1$ (or $u \leq 1/\sqrt{C}$ for the modified version of Hinge loss (2)). In SVM, only support vectors have impacts on the final solution. DWD is the other extreme case where all the data vectors have some impacts. In the presence of imbalanced sample size, the fact that all the

data vectors influence the solution cause the optimization to ignore the minority class. The FLAME with $0 < \theta < 1$ is somewhere in the middle. For FLAME, part of the data vectors, more than the support vectors, but fewer than all the vectors, have impacts. Smart choice of θ means that one needs to include as many vectors, and as balanced influential samples, as possible. More vectors usually lead to mitigated overfitting, and balanced sample size of the influential vectors from two classes means that the sensitivity issue of the intercept term can be alleviated.

The authors are aware that it is possible to implement a two-step procedure to conduct binary linear classification. In the first step, a good direction is found, probably in the fashion of DWD; in the second step, a fine intercept is chosen by borrowing idea of SVM. This idea is elaborated in Qiao and Zhang (2015).

The choice of θ usually depends on the nature of the data and the scientific context. If the users prefer better classification performance over reasonable discrimination direction for interpretation of the data, θ may be chosen to be closer to 1. If the right direction is the first priority, then θ should be chosen to be closer to 0. Note that, under some circumstances, the primary goal is to obtain a direction vector which can provide a score $\boldsymbol{x}^T\boldsymbol{\omega}$ for each observation for further use, and the intercept parameter β is of no use at all. For example, some users may use a receiver operating characteristic (ROC) curve as a graphical tool to evaluate classification performance over different β value instead of using a single β value given by the classifier. In this case, a FLAME machine close to the DWD method may be ideal.

Qiao et al. (2010) considered the sample weighted versions of DWD and SVM. One could in theory extend the FLAME directly to the so-called weighted FLAME family. Such extension is quite straightforward. It is easy to see that all the classifiers in such a family are Fisher consistent with respect to the weighted 0-1 loss function. The intercept term from weighted FLAME does not diverge. Similar HDLSS asymptotic results to what are presented in the current article can be expected as well.

Acknowledgments

The first author's work was partially supported by Binghamton University Harpur College Dean's New Faculty Start-up Funds and Dean's Research Semester Awards for Junior Faculty, and a collaboration grant from the Simons Foundation (#246649). Both authors thank the Statistical and Applied Mathematical Sciences Institute (SAMSI) for the support and the hospitality during the 2012-13 Program on Statistical and Computational Methodology for Massive Data sets and the 2013-14 Program on Low-dimensional Structure in High-dimensional Systems.

Appendix A. Derivation of Modified Hinge Loss

Note that the original SVM formulation is $\underset{\tilde{\boldsymbol{\omega}},\tilde{\beta}}{\operatorname{argmin}} \sum \left(1 - y_i \tilde{f}(\boldsymbol{x}_i)\right)_+$, s.t. $\|\tilde{\boldsymbol{\omega}}\|^2 \leq C$, where $\tilde{f}(\boldsymbol{x}) = \boldsymbol{x}^T\tilde{\boldsymbol{\omega}} + \tilde{\beta}$. Here the coefficient vector $\tilde{\boldsymbol{\omega}}$ does not have unit norm. We let $\boldsymbol{\omega} = \tilde{\boldsymbol{\omega}}/\sqrt{C}$,

$\beta = \tilde{\beta}/\sqrt{C}$ and $f = \tilde{f}/\sqrt{C}$. Thus SVM solution is given by $\underset{\boldsymbol{\omega},\beta}{\operatorname{argmin}} \sum \left(1 - \sqrt{C} y_i f(\boldsymbol{x}_i)\right)_+$, s.t. $\|\boldsymbol{\omega}\|^2 \leq 1$, or equivalently, $\underset{\boldsymbol{\omega},\beta}{\operatorname{argmin}} \sum \left(\sqrt{C} - C y_i f(\boldsymbol{x}_i)\right)_+$, s.t. $\|\boldsymbol{\omega}\|^2 \leq 1$.

References

J. Ahn and J. S. Marron. The maximal data piling direction for discrimination. *Biometrika*, 97(1):254–259, 2010.

J. Ahn, J. S. Marron, K.M. Muller, and Y.Y. Chi. The high-dimension, low-sample-size geometric representation holds under mild conditions. *Biometrika*, 94(3):760–766, 2007. doi: 10.1093/biomet/asm050. URL `http://biomet.oxfordjournals.org/content/94/3/760.short`.

P.L. Bartlett, M.I. Jordan, and J.D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006. ISSN 0162-1459. doi: 10.1198/016214505000000907. URL `http://pubs.amstat.org/doi/abs/10.1198/016214505000000907`.

Arindam Bhattacharjee, William G Richards, Jane Staunton, Cheng Li, Stefano Monti, Priya Vasa, Christine Ladd, Javad Beheshti, Raphael Bueno, Michael Gillette, et al. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses. *Proceedings of the National Academy of Sciences*, 98(24): 13790–13795, 2001. URL `http://www.pnas.org/content/98/24/13790.short`.

Gilles Blanchard, Olivier Bousquet, and Pascal Massart. Statistical performance of support vector machines. *The Annals of Statistics*, pages 489–531, 2008.

A Bolivar-Cime and J. S. Marron. Comparison of binary discrimination methods for high dimension low sample size data. *Journal of Multivariate Analysis*, 115:108–121, 2013.

N.V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1):1–6, 2004. doi: 10.1145/1007730.1007733.

Di-Rong Chen, Qiang Wu, Yiming Ying, and Ding-Xuan Zhou. Support vector machine soft margin classifiers: Error analysis. *Journal of Machine Learning Research*, 5:1143–1175, 2004.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2):201–233, 2002.

N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, 2000.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, 2001.

S. Dudoit, J. Fridlyand, and T.P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002. doi: 10.1198/016214502753479248. URL http://amstat.tandfonline.com/doi/abs/10.1198/016214502753479248.

Peter Hall, J. S. Marron, and Amnon Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):427–444, 2005. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2005.00510.x. URL http://dx.doi.org/10.1111/j.1467-9868.2005.00510.x.

T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

Hanwen Huang, Yufeng Liu, Ying Du, Charles M Perou, D Neil Hayes, Michael J Todd, and J. S. Marron. Multiclass distance weighted discrimination. *Journal of Computational and Graphical Statistics*, 22(just-accepted):953–969, 2013.

Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines. *Journal of the American Statistical Association*, 99(465):67–81, 2004. ISSN 0162-1459. doi: 10.1198/016214504000000098. URL http://pubs.amstat.org/doi/abs/10.1198/016214504000000098.

Y. Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73–82, 2004. ISSN 0167-7152. doi: 10.1016/j.spl.2004.03.002. URL http://www.sciencedirect.com/science/article/pii/S0167715204000707.

Y. Liu, D.N. Hayes, A. Nobel, and J. S. Marron. Statistical significance of clustering for high-dimension, low-sample size data. *Journal of the American Statistical Association*, 103(483):1281–1293, 2008. doi: 10.1198/016214508000000454. URL http://amstat.tandfonline.com/doi/abs/10.1198/016214508000000454.

Y. Liu, H.H. Zhang, and Y. Wu. Hard or soft classification? large-margin unified machines. *Journal of the American Statistical Association*, 106(493):166–177, 2011. doi: 10.1198/jasa.2011.tm10319. URL http://pubs.amstat.org/doi/abs/10.1198/jasa.2011.tm10319.

J. S. Marron, M.J. Todd, and J. Ahn. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271, 2007. doi: 10.1198/016214507000001120. URL http://pubs.amstat.org/doi/abs/10.1198/016214507000001120.

A.B. Owen. Infinitely imbalanced logistic regression. *Journal of Machine Learning Research*, 8:761–773, 2007. URL http://jmlr.csail.mit.edu/papers/v8/owen07a.html.

X. Qiao and Y. Liu. Adaptive weighted learning for unbalanced multicategory classification. *Biometrics*, 65(1):159–168, 2009. doi: 10.1111/j.1541-0420.2008.01017.x. URL http://onlinelibrary.wiley.com/doi/10.1111/j.1541-0420.2008.01017.x/abstract.

X. Qiao, H.H. Zhang, Y. Liu, M.J. Todd, and J. S. Marron. Weighted distance weighted discrimination and its asymptotic properties. *Journal of the American Statistical Association*, 105(489):401–414, 2010. doi: 10.1198/jasa.2010.tm08487. URL `http://pubs.amstat.org/doi/abs/10.1198/jasa.2010.tm08487`.

Xingye Qiao and Lingsong Zhang. Distance-weighted support vector machine. *Statistics and Its Interface*, 8:3, 2015.

Xiaotong Shen, George C Tseng, Xuegong Zhang, and Wing Hung Wong. On ψ-learning. *Journal of the American Statistical Association*, 98(463):724–734, 2003.

Alexander J Smola, Peter L Bartlett, Bernhard Schölkopf, and Dale Schuurmans. *Advances in Large Margin Classifiers*, volume 1. MIT Press Cambridge, MA, 2000.

Ingo Steinwart and Clint Scovel. Fast rates for support vector machines using gaussian kernels. *The Annals of Statistics*, pages 575–607, 2007.

K.C. Toh, M.J. Todd, and R.H. Tütüncü. Sdpt3–a matlab software package for semidefinite programming, version 1.3. *Optimization Methods and Software*, 11(1):545–581, 1999. URL `http://www.math.nus.edu.sg/~mattohkc/papers/guide.ps.Z`.

R.H. Tütüncü, K.C. Toh, and M.J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95(2):189–217, 2003. URL `http://dx.doi.org/10.1007/s10107-002-0347-5`.

V.N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Ernesto D Vito, Lorenzo Rosasco, Andrea Caponnetto, Umberto D Giovannini, and Francesca Odone. Learning from examples as an inverse problem. *Journal of Machine Learning Research*, 6:883–904, 2005.

Grace Wahba. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. *Advances in Kernel Methods-Support Vector Learning*, 6:69–87, 1999.

J. Weston and C. Watkins. Support vector machines for multi-class pattern recognition. In *European Symposium on Artificial Neural Networks*, pages 219–224, 1999.

Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007. URL `http://amstat.tandfonline.com/doi/full/10.1198/016214507000000617`.

Lingsong Zhang and Xihong Lin. Some considerations of classification for high dimension low-sample size data. *Statistical Methods in Medical Research*, 22:537–550, 2013.

Tong Zhang. Statistical behavior and consistency of classification methods based on convex risk minimization. *Annals of Statistics*, pages 56–85, 2004.

Journal of Machine Learning Research 16 (2015) 1573-1578 Submitted 11/13; Revised 11/14; Published 8/15

RLPy: A Value-Function-Based Reinforcement Learning Framework for Education and Research

Alborz Geramifard[12] AGF@CSAIL.MIT.EDU
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology,
77 Massachusetts Ave., Cambridge, MA 02139 – USA
Christoph Dann[1] CDANN@CMU.EDU
Machine Learning Department, Carnegie Mellon University,
5000 Forbes Ave., Pittsburgh, PA 15213 – USA
Robert H. Klein[1] BOBKLEIN2@ALUM.MIT.EDU
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology,
77 Massachusetts Ave., Cambridge, MA 02139 – USA
William Dabney[2] WDDABNEY@AMAZON.COM
Amazon.com,
440 Terry Ave. N, Seattle, WA 98109 – USA
Jonathan P. How JHOW@MIT.EDU
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology,
77 Massachusetts Ave., Cambridge, MA 02139 – USA

Editor: Geoff Holmes

Abstract

RLPy is an object-oriented reinforcement learning software package with a focus on value-function-based methods using linear function approximation and discrete actions. The framework was designed for both educational and research purposes. It provides a rich library of fine-grained, easily exchangeable components for learning agents (e.g., policies or representations of value functions), facilitating recently increased specialization in reinforcement learning. RLPy is written in Python to allow fast prototyping, but is also suitable for large-scale experiments through its built-in support for optimized numerical libraries and parallelization. Code profiling, domain visualizations, and data analysis are integrated in a self-contained package available under the Modified BSD License at `http://github.com/rlpy/rlpy`. All of these properties allow users to compare various reinforcement learning algorithms with little effort.

Keywords: reinforcement learning, value-function, empirical evaluation, open source

1. Introduction

An integral part of most artificial intelligence courses are value-function-based methods using linear function approximation for solving Markov decision processes (such as linear Q-learning or SARSA). In addition, many researchers build upon this well-understood and

1. The first three authors contributed equally to this work.
2. The majority of this work was done prior to Amazon involvement of the authors. This paper does not reflect the views of the Amazon company.

powerful framework and aim at improving existing methods by, for example, feature learning (Keller et al., 2006; Parr et al., 2007; Geramifard et al., 2011), or policies with better exploration-exploitation trade-off (Nouri and Littman, 2009; Jaksch et al., 2010; Li, 2012).

The need to unify and increase reusability of software packages for reinforcement learning research has been widely discussed (Tanner and White, 2009; Schaul et al., 2010), and many successful tools have been created (see Section 2). However, it is desirable to have a software framework that is 1) easily accessible by novices so that they may compare and understand existing algorithms, and 2) efficient for researchers who perform large scale experiments and advance the state-of-the-art.

By focusing on the prominent class of value-function-based methods with linear function approximation using discrete actions, RLPy aims at being such a software framework that provides simple and convenient tools for conducting sequential decision making experiments. In the following, we present the main features of RLPy and highlight those that distinguish it from existing frameworks.

2. Existing Frameworks

The following existing software packages have some overlap with RLPy:

1. *RL-Toolbox: (Neumann, 2005)* C++ RL toolbox focusing on continuous state-spaces
2. *CLSquare: (Riedmiller et al., 2012)* C++ RL framework focused on interfaces with several robotics
3. *libPG: (Aberdeen, 2007)* RL library focused on high-performance policy-gradient algorithm implementations
4. *rllib (Frezza-Buet and Geist, 2013):* Template-based C++ RL library for value-function methods
5. *rl-texplore-ros-pkg:(Hester, 2013)* ROS package for RL algorithms
6. *JRLF: (Kochenderfer, 2006)* Small-scale Java-Framework for RL experiments
7. *PIQLE: (de Comite, 2006)* Java-Framework for RL experiments
8. *RLPark:(Degris, 2013)* Java reinforcement learning library
9. *RLLib: (Abeyruwan, 2013)* Port of RLPark into C++
10. *RL-Glue, RL-Library: (Tanner and White, 2009)* Protocol for RL experiments and reference implementations
11. *ApproxRL: (Busoniu, 2010)* Matlab Toolbox with RL and dynamic programming algorithms
12. *MMLF: (Metzen and Edgington, 2011)* Python-based framework for reinforcement learning
13. *PyBrain (Schaul et al., 2010):* Machine learning library focused on neural networks with RL support

For the sake of brevity, we do not compare RLPy against each of the existing frameworks in detail but highlight key differences in the following section by referencing the list above.

3. Why RLPy?

Improved Granularity of Agents with Linear Value Functions. RL has advanced significantly over the past decade, leading researchers to narrow their focus towards specialized, independent aspects of RL agents, such as approximate function representations, exploration schemes, and learning rates. The structure of numerous existing frameworks (2, 6, 7, 10, 12) does not properly account for this increased specialization and makes it cumbersome to exchange, for example, the way the value function is represented in a learning agent. RLPy addresses this issue by separating these components into exchangeable classes (shown as green boxes in Figure 1) and other minor components such as learning rates into separate functions. This division reduces implementation effort, promotes reusability, and facilitates automated testing. Code for an example experiment that exploits this modularity is shown in Figure 2. In addition, the assumption of linearly parameterizing the value function allows RLPy to provide many tools and helpers for designing state features. For example, in large

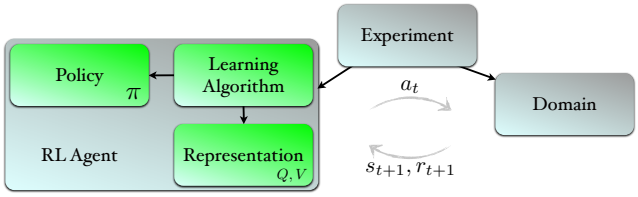


Figure 1: RLPy framework - Green components constitute an RL agent which did not exist as separate components in previous RL frameworks. The experiment module handles the interaction between the agent and the domain; gray arrows depict the information flow in a conventional RL framework (Sutton and Barto, 1998).

```
import rlpy
#### Domain ####
domain = rlpy.Domains.InfCartPoleBalance()
### Agent ####
representation  = rlpy.Representations.Tabular(domain, discretization=20)
policy = rlpy.Policies.eGreedy(representation, epsilon=0.1)
agent = rlpy.Agents.SARSA(policy, representation, domain.discount_factor)
### Experiment ####
experiment = rlpy.Experiments.Experiment(agent, domain, max_steps=100000)
experiment.run()
experiment.save()
```

Figure 2: RLPy code for setting up and running an experiment: SARSA learning for $100,000$ steps how to balance an inverted pole on a cart while following an ϵ-greedy policy and using discretized tabular features.

MDPs where using a tabular representation is infeasible, the `IndependentDiscretization` representation creates features by ignoring dependency among dimensions of states.

Rapid Prototyping with Python. RLPy is fully object-oriented and based primarily on the Python language (van Rossum and de Boer, 1991). Low-level, computationally-intensive tools are implemented in Cython (a compiled and typed version of Python) or C++. In contrast to other packages (1 – 9) written solely in C++ or Java, this approach leverages the user-friendliness, conciseness, and portability of Python while supplying computational efficiency where needed. This combination allows researchers to prototype new ideas quickly and comfortably without sacrificing the computing speed necessary to conduct large-scale experiments. In addition, the Python-based approach of RLPy is particularly suited for education as it does not require any proprietary software (in contrast to 11).

"Batteries Included" – Many Existing Components and Benchmarks. RLPy includes an ever-growing repository of components which may be combined to form new RL agents. While many frameworks (1, 3, 4, 6) only include classic benchmark domains such as *Puddle-World* or an *Inverted Pendulum on Cart*, RLPy supplies a large number of more challenging domains such as *HIV-Treatment*, *Hovering a Helicopter*, and *Pac-Man*. In addition to implementations of most value-function-based RL algorithms, RLPy includes experimental support for dynamic programming methods that require full domain knowledge but yield optimal policies. This is especially useful as a baseline for comparison with (often sub-optimal) policies generated by RL agents.

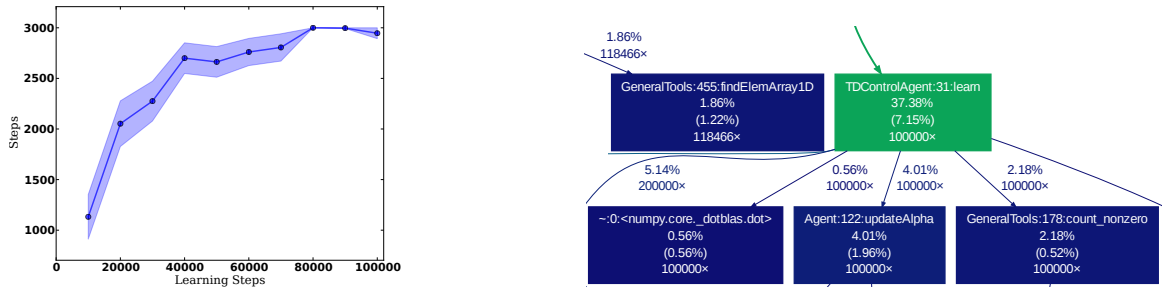


Figure 3: RLPy sample outputs of RLPy plotting (left) and profiling (right) tools: A portion of the profiling graph of the example code (Figure 2) in which the green box shows the statistics of executing the `learn` function 10^5 times. It required 37.38% of the CPU-time for completion, out of which its main body was responsible only for 7.15% of the computation while the rest was spent in other called functions.

Ease of Use and Development. Numerous tools are shipped with RLPy that facilitate ease of use and efficiency. One example is the code profiler, which produces a visual runtime graph of the source code (c.f. Figure 3 right) and identifies slow routines. This information allows the researcher to reduce the runtime of an algorithm with minimal effort and discourages premature runtime optimization. Additionally, every RLPy domain has a visualization, an important feature lacking in other frameworks (3, 4, 7). These visuals help user quickly assess and gain intuition about the algorithm and domain behavior.

Automation of Experiments. RLPy aims to promote reproducible research. To this end, it provides a suite of tools to automate the entire experiment pipeline. For example, RLPy allows concise specification of experiment settings (see Figure 2) and automated and efficient hyperparameter optimization with the `hyperopt` package (Yamins et al., 2013). Researchers can share their experimental setups by publishing short settings files, and colleagues can reproduce the results when running the scripts independent of their hardware or operating system. Additionally, RLPy experiments are natively parallelizable. Once parameters are selected, the user simply specifies the number of CPU cores RLPy can utilize for multiple experiments to test statistical significance. RLPy enables further scaling by switching seamlessly from a single machine to a job-based cluster (e.g. HTCondor) while ensuring results remain identical across varying hardware. RLPy also provides automated tools for generation of final publication-ready plots of results (see Figure 3 left); researchers need only specify the quantities that should appear on the plot. To the best of our knowledge this degree of automation of the entire experimentation pipeline is unique to RLPy.

4. Conclusion

RLPy is a new reinforcement learning framework focused on value-function-based reinforcement learning using linear function approximation with discrete actions. It simplifies the construction of learning agents and makes it easier for novices and experts alike to evaluate and compare algorithms, representations, environments, and other RL components. RLPy also provides many tools for conducting reproducible experiments from initial prototyping to final plotting. The framework is entirely open-source and all contributions are welcome.

References

Douglas Aberdeen. LibPGRL: A high performance reinforcement learning library in C++, 2007. URL `https://code.google.com/p/libpgrl`.

Saminda Abeyruwan. RLLib reinforcement learning c++ template library, 2013. URL `http://web.cs.miami.edu/home/saminda/rllib.html`.

Lucian Busoniu. Approxrl: A matlab toolbox for approximate reinforcement learning and dynamic programming, 2010. URL `http://busoniu.net/files/repository/readme_approxrl.html`.

Francesco de Comite. PIQLE: A platform for implementation of Q-learning experiments, 2006. URL `http://piqle.sourceforge.net`.

Thomas Degris. RLPark, 2013. URL `http://rlpark.github.io`.

Herve Frezza-Buet and Matthieu Geist. A C++ template-based reinforcement learning library: Fitting the code to the mathematics. *Journal of Machine Learning Research (JMLR)*, 14:625–628, 2013.

Alborz Geramifard, Finale Doshi, Joshua Redding, Nicholas Roy, and Jonathan How. Online discovery of feature dependencies. In *International Conference on Machine Learning (ICML)*, 2011.

Todd Hester. rl-texplore-ros-pkg: Reinforcement learning framework, agents, and environments with ROS interface, 2013. URL `https://code.google.com/p/rl-texplore-ros-pkg`.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 11:1563–1600, 2010.

Philipp W. Keller, Shie Mannor, and Doina Precup. Automatic basis function construction for approximate dynamic programming and reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2006.

Mykel Kochenderfer. JRLF: Java reinforcement learning framework, 2006. URL `http://mykel.kochenderfer.com/jrlf`.

Lihong Li. Sample complexity bounds of exploration. In Marco Wiering and Martijn van Otterlo, editors, *Reinforcement Learning: State of the Art*. Springer Verlag, 2012.

Jan Hendrik Metzen and Mark Edgington. Maja machine learning framework, 2011. URL `http://mmlf.sourceforge.net`.

Gerhard Neumann. The reinforcement learning toolbox, reinforcement learning for optimal control tasks, 2005.

Ali Nouri and Michael L. Littman. Multi-resolution exploration in continuous spaces. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

Ronald Parr, Christopher Painter-Wakefield, Lihong Li, and Michael Littman. Analyzing Feature Generation for Value-Function Approximation. In *International Conference on Machine Learning (ICML)*, 2007.

Martin Riedmiller, Manuel Blum, and Thomas Lampe. CLS^2: Closed loop simulation system, 2012. URL `http://ml.informatik.uni-freiburg.de/research/clsquare`.

Tom Schaul, Justin Bayer, Daan Wierstra, Yi Shun, Martin Felder, Frank Sehnke, Thomas Rückstieß, and Jürgen Schmidhuber. PyBrain. *Journal of Machine Learning Research (JMLR)*, 11:743–746, 2010.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Brian Tanner and Adam White. RL-Glue : Language-independent software for reinforcement-learning experiments. *Journal of Machine Learning Research (JMLR)*, 10:2133–2136, 2009.

Guido van Rossum and Jelke de Boer. Interactively testing remote servers using the python programming language. *CWI Quarterly*, 4(4):283–303, 1991.

Daniel Yamins, David Tax, and James S. Bergstra. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International Conference on Machine Learning (ICML)*, 2013.

Journal of Machine Learning Research 16 (2015) 1579-1606 Submitted 3/14; Revised 10/14; Published 8/15

Calibrated Multivariate Regression with Application to Neural Semantic Basis Discovery*

Han Liu HANLIU@PRINCETON.EDU
Department of Operations Research and Financial Engineering, Princeton University, NJ 08544, USA

Lie Wang LIEWANG@MATH.MIT.EDU
Department of Mathematics, Massachusetts Institute of Technology, Cambridge MA 02139, USA

Tuo Zhao† TOUR@CS.JHU.EDU
Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA

Editor: Arthur Gretton

Abstract

We propose a calibrated multivariate regression method named CMR for fitting high dimensional multivariate regression models. Compared with existing methods, CMR calibrates regularization for each regression task with respect to its noise level so that it simultaneously attains improved finite-sample performance and tuning insensitiveness. Theoretically, we provide sufficient conditions under which CMR achieves the optimal rate of convergence in parameter estimation. Computationally, we propose an efficient smoothed proximal gradient algorithm with a worst-case numerical rate of convergence $\mathcal{O}(1/\epsilon)$, where ϵ is a pre-specified accuracy of the objective function value. We conduct thorough numerical simulations to illustrate that CMR consistently outperforms other high dimensional multivariate regression methods. We also apply CMR to solve a brain activity prediction problem and find that it is as competitive as a handcrafted model created by human experts. The R package `camel` implementing the proposed method is available on the Comprehensive R Archive Network `http://cran.r-project.org/web/packages/camel/`.

Keywords: calibration, multivariate regression, high dimension, sparsity, low Rank, brain activity prediction

1. Introduction

This paper studies the multivariate regression problem. Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the design matrix and $\mathbf{Y} \in \mathbb{R}^{n\times m}$ be the response matrix, we consider a linear model

$$\mathbf{Y} = \mathbf{X}\mathbf{B}^0 + \mathbf{Z}, \tag{1}$$

where $\mathbf{B}^0 \in \mathbb{R}^{d\times m}$ is an unknown regression coefficient matrix and $\mathbf{Z} \in \mathbb{R}^{n\times m}$ is a noise matrix (Anderson, 1958; Breiman and Friedman, 2002). For a matrix $\mathbf{A} = [\mathbf{A}_{jk}] \in \mathbb{R}^{d\times m}$, we denote its j^{th} row and k^{th} column by $\mathbf{A}_{j*} = (\mathbf{A}_{j1}, ..., \mathbf{A}_{jm}) \in \mathbb{R}^m$ and $\mathbf{A}_{*k} = (\mathbf{A}_{1k}, ..., \mathbf{A}_{dk})^T \in \mathbb{R}^d$ respectively. We assume that all $\mathbf{Z}_{i*}$'s are independently sampled from an m-dimensional distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{\Sigma} \in \mathbb{R}^{m\times m}$.

*. Some preliminaries results in this paper were presented at the 28-th Annual Conference on Neural Information Processing Systems, Montreal, Quebec, Canada, 2014 (Liu et al., 2014a). This work is partially supported by grants NIH R01MH102339, NSF IIS1408910, NSF IIS1332109, NSF CAREER DMS1454377, NIH R01GM083084, NIH R01HG06841, and NSF Grant DMS-1005539.

†. Tuo Zhao is also affiliated with Department of Operations Research and Financial Engineering at Princeton University.

We can represent (1) as an ensemble of univariate linear regression models:

$$\mathbf{Y}_{*k} = \mathbf{X}\mathbf{B}^0_{*k} + \mathbf{Z}_{*k},\ k = 1, ..., m,$$

which results in a multi-task learning problem (Baxter, 2000; Caruana, 1997; Caruana et al., 1996; Thrun, 1996; Ando and Zhang, 2005; Johnson and Zhang, 2008; Zhang et al., 2006; Zhang, 2006). Multi-task learning exploits shared common structure across tasks to obtain improved estimation performance. In the past decade, significant progress has been made on designing various modeling assumptions for multivariate regression.

One popular approach is to assume that the regression coefficients across different tasks are coupled by some shared common factors so that $\mathbf{B}^0$ has a low rank structure, i.e., $\text{rank}(\mathbf{B}^0) \ll \min(d, m)$. Under this assumption, a consistent estimator of $\mathbf{B}^0$ can be obtained by adopting either a non-convex rank constraint (Anderson, 1958; Izenman, 1975; Reinsel and Velu, 1998; Anderson, 1999; Reinsel and Velu, 1998; Izenman, 2008) or a convex relaxation using the nuclear norm regularization (Yuan et al., 2007; Amit et al., 2007; Argyriou et al., 2008; Negahban and Wainwright, 2011; Rohde and Tsybakov, 2011; Bunea et al., 2011, 2012; Bunea and Barbu, 2009; Mukherjee et al., 2012; Giraud, 2011; Argyriou et al., 2010; Foygel and Srebro, 2011; Johnson and Zhang, 2008; Salakhutdinov and Srebro, 2010; Evgeniou et al., 2006; Heskes, 2000; Teh et al., 2005; Yu et al., 2005). Such a low rank multivariate regression method is often applied to scenarios where m is large.

Another approach is to assume that all the regression tasks share a common sparsity pattern, i.e., many $\mathbf{B}^0_{j*}$'s are zero vectors. Such a joint sparsity assumption for multivariate regressions is a natural extension from sparse univariate linear regressions. Similar to using the L_1-regularization in Lasso (Tibshirani, 1996; Chen et al., 1998), group regularization can be used to obtain a consistent estimator of $\mathbf{B}^0$ (Yuan and Lin, 2005; Turlach et al., 2005; Meier et al., 2008; Lounici et al., 2011; Kolar et al., 2011). Such a sparse multivariate regression method is often applied to scenarios where the dimension d is large.

In this paper, we consider an uncorrelated structure for the noise matrix $\mathbf{Z}$, i.e.,

$$\mathbf{\Sigma} = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_{m-1}^2, \sigma_m^2). \tag{2}$$

Such an assumption allows us to efficiently solve the resulting estimation problem with a convex program and prove that the obtained estimator achieves the minimax optimal rates of convergence in parameter estimation.[1] For example, many existing work propose to solve the convex program

$$\widehat{\mathbf{B}} = \underset{\mathbf{B}}{\text{argmin}} \frac{1}{\sqrt{n}} ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{\text{F}}^2 + \lambda \mathcal{R}(\mathbf{B}), \tag{3}$$

where $\lambda > 0$ is a tuning parameter, $\mathcal{R}(\mathbf{B})$ is a regularization function of $\mathbf{B}$, and $||\mathbf{A}||_{\text{F}} = \sqrt{\sum_{j,k} \mathbf{A}_{jk}^2}$ is the Frobenius norm of a matrix $\mathbf{A}$. Popular choices of $\mathcal{R}(\mathbf{B})$ include

$$\text{Nuclear Norm}: ||\mathbf{B}||_* = \sum_{j=1}^{r} \psi_j(\mathbf{B}), \tag{4}$$

$$L_{1,p}\ \text{Norm}: ||\mathbf{B}||_{1,p} = \sum_{j=1}^{d} \left(\sum_{k=1}^{m} |\mathbf{B}_{jk}|^p \right)^{1/p} \quad \text{for } 2 \le p < \infty, \tag{5}$$

$$L_{1,\infty}\ \text{Norm}: ||\mathbf{B}||_{1,\infty} = \sum_{j=1}^{d} \max_{1 \le k \le m} |\mathbf{B}_{jk}|, \tag{6}$$

1. See more details on exploiting the covariance structure of the noise matrix $\mathbf{Z}$ for multivariate regression in Breiman and Friedman (2002); Reinsel (2003); Rothman et al. (2010).

where r in (4) is the rank of $\mathbf{B}$ and $\psi_j(\mathbf{B})$ represents the j^{th} largest singular value of $\mathbf{B}$. The optimization problem (3) can be efficiently solved by the block coordinate descent algorithm (Liu et al., 2009a,b; Liu and Ye, 2010; Zhao et al., 2014a,c), fast proximal gradient algorithm (Toh and Yun, 2010; Beck and Teboulle, 2009a,b), and alternating direction method of multipliers(Boyd et al., 2011; Liu et al., 2014b). Scalable software packages such as MALSAR have been developed (Zhou et al., 2012).

The problem in (2) is amenable to statistical analysis. Under suitable conditions on the noise and design matrices, let $\sigma_{\max} = \max_k \sigma_k$ and $||\mathbf{X}||_2 = \psi_1(\mathbf{X})$ denote the largest singular value of $\mathbf{X}$, if we choose

$$\text{Low Rank}: \ \lambda = 2c \cdot \frac{||\mathbf{X}||_2}{n} \cdot \sigma_{\max}\left(\sqrt{d} + \sqrt{m}\right), \tag{7}$$

$$\text{Joint Sparsity}: \ \lambda = 2c \cdot \sigma_{\max}\left(\sqrt{\log d} + m^{1-1/p}\right), \tag{8}$$

for some $c > 1$, then the estimator $\widehat{\mathbf{B}}$ in (3) achieves the optimal rates of convergence[2] (Lounici et al., 2011; Rohde and Tsybakov, 2011). More specifically, there exists some universal constant C such that, with high probability,

$$\text{Low Rank}: \ \frac{1}{\sqrt{m}}||\widehat{\mathbf{B}} - \mathbf{B}^0||_{\text{F}} \leq C \cdot \frac{||\mathbf{X}||_2}{\sqrt{n}} \cdot \sigma_{\max}\left(\sqrt{\frac{r}{n}} + \sqrt{\frac{rd}{nm}}\right),$$

$$\text{Joint Sparsity}: \ \frac{1}{\sqrt{m}}||\widehat{\mathbf{B}} - \mathbf{B}^0||_{\text{F}} \leq C \cdot \sigma_{\max}\left(\sqrt{\frac{s\log d}{nm}} + \sqrt{\frac{sm^{1-2/p}}{n}}\right),$$

where r is the rank of $\mathbf{B}^0$ for the low rank setting and s is the number of rows with non-zero entries in $\mathbf{B}^0$ for the setting of joint sparsity.

The estimator in (3) has two drawbacks: (i) All the tasks are regularized by the same tuning parameter λ, even though different tasks may have different σ_k's. Thus more estimation bias is introduced to the tasks with smaller σ_k's since they have to compensate the tasks with larger σ_k's. In another word, these tasks are not calibrated (Zhao and Liu, 2014). (ii) The tuning parameter selection, as shown in (7) and (8), involves the unknown parameter $\sigma_{\max}$. This requires the regularization parameter to be carefully tuned over a wide range of potential values in order to get a good finite-sample performance.

To overcome the above two drawbacks, we propose a new method named calibrated multivariate regression (CMR) based on the convex program

$$\widehat{\mathbf{B}} = \underset{\mathbf{B}}{\text{argmin}} \, ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{2,1} + \lambda\mathcal{R}(\mathbf{B}) \tag{9}$$

where $||\mathbf{A}||_{2,1} = \sum_k \sqrt{\sum_j \mathbf{A}_{jk}^2}$ is the $L_{2,1}$ norm of a matrix $\mathbf{A} = [\mathbf{A}_{jk}] \in \mathbb{R}^{d\times m}$. This is a multivariate extension of the square-root Lasso estimator (Belloni et al., 2011; Sun and Zhang, 2012). Similar to the square-root Lasso, the tuning parameter selection of CMR does not involve $\sigma_{\max}$. Thus the resulting procedure adapts to different σ_k's and achieves an improved finite-sample performance comparing with the ordinary multivariate regression estimator (OMR) defined in (3). Since both the loss and regularization functions in (9) are nonsmooth, CMR is computationally more challenging than OMR. To efficiently solve CMR, we develop a smoothed proximal gradient algorithm with a worst-case iteration complexity of $\mathcal{O}(1/\epsilon)$, where ϵ is a pre-specified accuracy of the objective value (Nesterov, 2005; Chen et al., 2012; Zhao and Liu, 2012; Zhao et al., 2014b). Theoretically, we show that under suitable conditions, CMR achieves the optimal rates of convergence in parameter

2. For the joint sparsity setting, the rate of convergence is optimal when $\mathbf{R}(\mathbf{B}) = ||\mathbf{B}||_{1,2}$. See more details in Lounici et al. (2011)

estimation. Numerical experiments on both synthetic and real data show that CMR universally outperforms existing multivariate regression methods. For a brain activity prediction task, prediction based on the features selected by CMR significantly outperforms that based on the features selected by OMR, and is even competitive with that based on the handcrafted features selected by human experts.

This paper is organized as follows: In §2, we describe the CMR method. In §3, we investigate the statistical properties of CMR; In §4, we derive a smoothed proximal gradient algorithm for solving CMR optimization. In §5, we conduct numerical experiments to illustrate the usefulness of the proposed method. In §6, we discuss the relationships between our results and other related work.

Notation: Given a vector $\boldsymbol{v} = (v_1, \ldots, v_d)^T \in \mathbb{R}^d$, for $1 \leq p \leq \infty$, we define the vector norms: $||\boldsymbol{v}||_p = \left(\sum_{j=1}^d |v_j|^p\right)^{1/p}$ for $1 \leq p < \infty$ and $||\boldsymbol{v}||_\infty = \max_{1\leq j\leq d} |v_j|$. Given two matrices $\mathbf{A} = [\mathbf{A}_{jk}]$ and $\mathbf{C} = [\mathbf{C}_{jk}] \in \mathbb{R}^{d\times m}$, we define the inner product of $\mathbf{A}$ and $\mathbf{C}$ as $\langle \mathbf{A}, \mathbf{C}\rangle = \sum_{j=1}^d \sum_{k=1}^m \mathbf{A}_{jk}\mathbf{C}_{jk} = \mathrm{tr}(\mathbf{A}^T\mathbf{C})$, where $\mathrm{tr}(\mathbf{A})$ is the trace of a matrix $\mathbf{A}$. We use $\mathbf{A}_{*k} = (\mathbf{A}_{1k}, ..., \mathbf{A}_{dk})^T$ and $\mathbf{A}_{j*} = (\mathbf{A}_{j1}, ..., \mathbf{A}_{jm})$ to denote the k^{th} column and j^{th} row of $\mathbf{A}$. Let $\mathcal{S}$ be some subspace of $\mathbb{R}^{d\times m}$, we use $\mathbf{A}_{\mathcal{S}}$ to denote the projection of $\mathbf{A}$ onto $\mathcal{S}$, i.e., $\mathbf{A}_{\mathcal{S}} = \mathrm{argmin}_{\mathbf{C}\in\mathcal{S}} ||\mathbf{C}-\mathbf{A}||_{\mathrm{F}}^2$. Given a subspace $\mathcal{U} \subset \mathbb{R}^d$, we define its orthogonal complement as $\mathcal{U}_\perp = \left\{\boldsymbol{u} \in \mathbb{R}^d \mid \boldsymbol{u}^T\boldsymbol{v} = 0, \text{ for all } \boldsymbol{v} \in \mathcal{U}\right\}$. Moreover, we define the Frobenius, spectral, and nuclear norms of $\mathbf{A}$ as $||\mathbf{A}||_{\mathrm{F}} = \sqrt{\langle \mathbf{A}, \mathbf{A}\rangle}$, $||\mathbf{A}||_2 = \psi_1(\mathbf{A})$, and $||\mathbf{A}||_* = \sum_{j=1}^r \psi_j(\mathbf{A})$, where r is the rank of $\mathbf{A}$, and $\psi_j(\mathbf{A})$ is the j^{th} largest singular value of $\mathbf{A}$. In addition, we define the matrix block norms as $||\mathbf{A}||_{2,1} = \sum_{k=1}^m ||\mathbf{A}_{*k}||_2$, $||\mathbf{A}||_{2,\infty} = \max_{1\leq k\leq m} ||\mathbf{A}_{*k}||_2$, $||\mathbf{A}||_{1,p} = \sum_{j=1}^d ||\mathbf{A}_{j*}||_p$, and $||\mathbf{A}||_{\infty,q} = \max_{1\leq j\leq d} ||\mathbf{A}_{j*}||_q$, where $1 \leq p \leq \infty$ and $1 \leq q \leq \infty$. It is easy to verify that $||\mathbf{A}||_{2,1}$ and $||\mathbf{A}||_*$ are dual norms of $||\mathbf{A}||_{2,\infty}$ and $||\mathbf{A}||_2$ respectively. Let $1/\infty = 0$, then if $1/p + 1/q = 1$, $||\mathbf{A}||_{\infty,q}$ and $||\mathbf{A}||_{1,p}$ are also dual norms of each other.

2. Method

We solve the multivariate regression problem in (1) by the convex program

$$\widehat{\mathbf{B}} = \underset{\mathbf{B}}{\mathrm{argmin}} ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{2,1} + \lambda\mathcal{R}(\mathbf{B}), \tag{10}$$

where $\mathcal{R}(\mathbf{B})$ is a regularization function and can take the forms in (4), (5), and (6).

To understand the intuition of (10), we show that the $L_{2,1}$-loss can be viewed as a special case of the weighted square loss function. More specifically, we consider the optimization problem

$$\widehat{\mathbf{B}}^* = \underset{\mathbf{B}}{\mathrm{argmin}} \sum_{k=1}^m \frac{1}{\sigma_k\sqrt{n}} ||\mathbf{Y}_{*k} - \mathbf{X}\mathbf{B}_{*k}||_2^2 + \lambda\mathcal{R}(\mathbf{B}), \tag{11}$$

where $\frac{1}{\sigma_k\sqrt{n}}$ is the weight to calibrate the k^{th} regression task. $\widehat{\mathbf{B}}^*$ is an "oracle" estimator (not practically calculable) since it assumes that all σ_k's are given. Without any prior knowledge of σ_k's, we can use the following replacement of σ_k's,

$$\widetilde{\sigma}_k = \frac{1}{\sqrt{n}} ||\mathbf{Y}_{*k} - \mathbf{X}\mathbf{B}_{*k}||_2, \ k = 1, ..., m. \tag{12}$$

We then recover (10) by replacing σ_k in (12) by $\widetilde{\sigma}_k$. In another word, CMR calibrates different tasks by solving a regularized weighted least square problem with weights defined in (12).

3. Statistical Properties

For notational simplicity, we define a rescaled noise matrix $\mathbf{W} = [\mathbf{W}_{ik}] \in \mathbb{R}^{n\times m}$ with $\mathbf{W}_{ik} = \mathbf{Z}_{ik}/\sigma_k$, where $\mathbb{E}\mathbf{Z}_{ik}^2 = \sigma_k^2$ is defined in (2). Thus $\mathbf{W}$ is a random matrix with all entries having mean 0 and

variance 1. We define $\mathbf{G}^0$ as the gradient of $||\mathbf{Y}-\mathbf{XB}||_{2,1}$ at $\mathbf{B}=\mathbf{B}^0$. We see that $\mathbf{G}^0$ does not depend on the unknown quantities σ_k's since

$$\mathbf{G}^0_{*k}=\frac{\mathbf{X}^T\mathbf{Z}_{*k}}{||\mathbf{Z}_{*k}||_2}=\frac{\mathbf{X}^T\mathbf{W}_{*k}\sigma_k}{||\mathbf{W}_{*k}\sigma_k||_2}=\frac{\mathbf{X}^T\mathbf{W}_{*k}}{||\mathbf{W}_{*k}||_2}.$$

Thus it serves as an important pivotal in our analysis. Moreover, our analysis exploits the decomposability of $\mathcal{R}(\mathbf{B})$, which is satisfied by the nuclear and $L_{1,p}$ norms (Negahban et al., 2012).

Definition 1 *Let $\mathcal{S}$ and $\mathcal{N}$ be two subspaces of $\mathbb{R}^{d\times m}$, which are orthogonal to each other and also satisfy $\mathcal{S}\subseteq\mathcal{N}_\perp$. A regularization function $\mathcal{R}(\cdot)$ is decomposable with respect to the pair $(\mathcal{S},\mathcal{N})$ if for any $\mathbf{A}\in\mathbb{R}^{d\times m}$, we have*

$$\mathcal{R}(\mathbf{A}+\mathbf{C})=\mathcal{R}(\mathbf{A})+\mathcal{R}(\mathbf{C}) \quad \textit{for} \quad \mathbf{A}\in\mathcal{S} \textit{ and } \mathbf{C}\in\mathcal{N}.$$

The decomposability of $\mathcal{R}(\mathbf{B})$ is important in analyzing the statistical properties of the estimator in (10). The next lemma shows that if we choose $\mathcal{S}$ to be some subspace of $\mathbb{R}^{d\times m}$ containing the true parameter $\mathbf{B}^0$, given a decomposable regularizer and a suitably chosen λ, the optimum to (10) lies in a restricted set.

Lemma 2 *Let $\mathbf{B}^0\in\mathcal{S}$ and $\widehat{\mathbf{B}}$ be an arbitrary[3] optimum to (10). We denote the estimation error as $\widehat{\mathbf{\Delta}}=\widehat{\mathbf{B}}-\mathbf{B}^0$ and the dual norm of $\mathcal{R}(\cdot)$ as $\mathcal{R}^*(\cdot)$. If $\lambda\geq c\mathcal{R}^*(\mathbf{G}^0)$ for some $c>1$, we have*

$$\widehat{\mathbf{\Delta}}\in\mathcal{M}_c=\left\{\mathbf{\Delta}\in\mathbb{R}^{d\times m}\mid \mathcal{R}(\mathbf{\Delta}_{\mathcal{N}})\leq\frac{c+1}{c-1}\mathcal{R}(\mathbf{\Delta}_{\mathcal{N}_\perp})\right\}. \tag{13}$$

The proof of Lemma 2 is provided in Appendix A. To prove the main result, we assume that the design matrix $\mathbf{X}$ satisfies a generalized restricted eigenvalue condition as below.

Assumption 1 *Let $\mathbf{B}^0\in\mathcal{S}$, then there exist positive constants κ and $c>1$ such that*

$$\kappa=\min_{\mathbf{\Delta}\in\mathcal{M}_c\setminus\{\mathbf{0}\}}\frac{||\mathbf{X\Delta}||_{\mathrm{F}}}{\sqrt{n}||\mathbf{\Delta}||_{\mathrm{F}}}.$$

Assumption 1 is the generalization of the restricted eigenvalue conditions for analyzing univariate sparse linear models (Negahban et al., 2012; Bickel et al., 2009). Many design matrices satisfy this assumption with high probability (Lounici et al., 2011; Negahban and Wainwright, 2011; Rohde and Tsybakov, 2011; Raskutti et al., 2010).

3.1 Main Result

We first present a deterministic result for a general norm-based regularization function $\mathcal{R}(\cdot)$, which satisfies the decomposability in Definition 1.

Theorem 3 *Suppose that the design matrix $\mathbf{X}$ satisfies Assumption 1. Let $\widehat{\mathbf{B}}$ be an arbitrary optimum to (10), and $\mathbf{G}^0$ be the gradient of $||\mathbf{Y}-\mathbf{XB}||_{2,1}$ at $\mathbf{B}=\mathbf{B}^0$. We denote*

$$\Theta(\mathcal{N}_\perp,\mathcal{R})=\max_{\mathbf{A}\in\mathbb{R}^{d\times m}\setminus\{\mathbf{0}\}}\frac{\mathcal{R}(\mathbf{A}_{\mathcal{N}_\perp})}{||\mathbf{A}_{\mathcal{N}_\perp}||_{\mathrm{F}}}.$$

Let λ satisfy

$$2\lambda\Theta(\mathcal{N}_\perp,\mathcal{R})\leq\delta(c-1)\sqrt{n}\kappa \textit{ for some } \delta<1, \textit{ and } \lambda\geq c\mathcal{R}^*(\mathbf{G}^0).$$

3. Since (10) is not a strictly convex program, the optimum to (10) is not necessarily unique.

Then we have

$$\frac{1}{\sqrt{nm}}||\mathbf{X}\widehat{\mathbf{B}}-\mathbf{X}\mathbf{B}^0||_{\mathrm{F}} \leq \frac{4\lambda\Theta(\mathcal{N}_\perp,\mathcal{R})\sigma_{\max}}{\sqrt{m}n\kappa(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty},$$
$$\frac{1}{\sqrt{m}}||\widehat{\mathbf{B}}-\mathbf{B}^0||_{\mathrm{F}} \leq \frac{4\lambda\Theta(\mathcal{N}_\perp,\mathcal{R})\sigma_{\max}}{\sqrt{m}n\kappa^2(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty},$$

where $\sigma_{\max}=\max_{1\leq k\leq m}\sigma_k$. *Moreover, if we estimate* σ_k*'s by*

$$\widehat{\sigma}_k=\frac{1}{\sqrt{n}}||\mathbf{Y}_{*k}-\mathbf{X}\widehat{\mathbf{B}}_{*k}||_2 \text{ for all } k=1,...,m, \tag{14}$$

then we have

$$\frac{1}{m}\left|\sum_{k=1}^m\widehat{\sigma}_k-\sum_{k=1}^m\sigma_k\right| \leq \max\left\{1,\frac{2}{c-1}\right\}\frac{4\lambda^2\Theta^2(\mathcal{N}_\perp,\mathcal{R})\sigma_{\max}}{\sqrt{n}mn\kappa(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty}.$$

The proof of Theorem 3 is provided in Appendix B. Note that Theorem 3 is a deterministic bound of the CMR estimator for a fixed λ. Since $\mathbf{W}$ is a random matrix, we need to bound $||\mathbf{W}||_{2,\infty}$ and show that $\lambda \geq c\mathcal{R}^*(\mathbf{G}^0)$ holds with high probability. For simplicity, we assume that each entry of $\mathbf{W}$ follows a Gaussian distribution as follows.

Assumption 2 *All* $\mathbf{W}_{ik}$*'s are independently generated from N(0,1).*

We then refine error bounds of the CMR estimator under Assumption 2 for calibrated sparse multivariate regression and calibrated low rank multivariate regression respectively .

3.2 Calibrated Low Rank Multivariate Regression

We assume that the rank of $\mathbf{B}^0$ is $r \ll \min\{d,m\}$, and $\mathbf{B}^0$ has a singular value decomposition $\mathbf{B}^0=\sum_{j=1}^r\psi_j(\mathbf{B}^0)\boldsymbol{u}_j\boldsymbol{v}_j^T$ where $\psi_j(\mathbf{B}^0)$ is the j^{th} largest singular value with $\boldsymbol{u}_j$'s and $\boldsymbol{v}_j$'s as the corresponding left and right singular vectors. We define

$$\mathcal{U}=\mathrm{span}(\{\boldsymbol{u}_1,...,\boldsymbol{u}_r\})\subset\mathbb{R}^d \text{ and } \mathcal{V}=\mathrm{span}(\{\boldsymbol{v}_1,...,\boldsymbol{v}_r\})\subset\mathbb{R}^m.$$

We then define $\mathcal{S}$ and $\mathcal{N}$ as follows,

$$\mathcal{S}=\Big\{\mathbf{C}\in\mathbb{R}^{d\times m}\ \Big|\ \mathbf{C}_{*k}\in\mathcal{U},\ \mathbf{C}_{j*}\in\mathcal{V} \text{ for all } j,k\Big\}, \tag{15}$$
$$\mathcal{N}=\Big\{\mathbf{C}\in\mathbb{R}^{d\times m}\ \Big|\ \mathbf{C}_{*k}\in\mathcal{U}_\perp,\ \mathbf{C}_{j*}\in\mathcal{V}_\perp \text{ for all } j,k\Big\}. \tag{16}$$

We can easily verify that $\mathbf{B}^0\in\mathcal{S}$ and the nuclear norm is decomposable with respect to the pair $(\mathcal{S},\mathcal{N})$, i.e.,

$$||\mathbf{A}+\mathbf{C}||_*=||\mathbf{A}||_*+||\mathbf{C}||_* \quad \text{for} \quad \mathbf{A}\in\mathcal{S} \text{ and } \mathbf{C}\in\mathcal{N}.$$

The next corollary provides the concrete rates of convergence for the calibrated low rank multivariate regression estimator.

Corollary 4 *We assume that the design matrix* $\mathbf{X}$ *satisfies Assumption 1 with* $\mathcal{S}$ *and* $\mathcal{N}$ *chosen as in* (15) *and* (16)*, and each column of* $\mathbf{X}$ *is normalized so that*

$$\frac{\|\mathbf{X}_{*j}\|_2}{\sqrt{n}}=1 \text{ for all } j=1,...,d. \tag{17}$$

We also assume that the rescaled noise matrix $\mathbf{W}$ *satisfies Assumption 2. By Theorem 3, for some universal constants* $c_0 \in (0,1)$, $c_1 > 0$, *and large enough* n, *we take*

$$\lambda = \frac{2c||\mathbf{X}||_2(\sqrt{d}+\sqrt{m})}{\sqrt{n(1-c_0)}}, \tag{18}$$

then for some $\delta < 1$, *we have*

$$\begin{aligned}
\frac{1}{\sqrt{nm}}||\mathbf{X}\widehat{\mathbf{B}} - \mathbf{X}\mathbf{B}^0||_{\mathrm{F}} &\leq \frac{8c\sqrt{2}||\mathbf{X}||_2\sigma_{\max}}{\sqrt{n}\kappa(c-1)(1-\delta)}\sqrt{\frac{1+c_0}{1-c_0}}\left(\sqrt{\frac{r}{n}}+\sqrt{\frac{rd}{nm}}\right), \\
\frac{1}{\sqrt{m}}||\widehat{\mathbf{B}} - \mathbf{B}^0||_{\mathrm{F}} &\leq \frac{8c\sqrt{2}||\mathbf{X}||_2\sigma_{\max}}{\sqrt{n}\kappa^2(c-1)(1-\delta)}\sqrt{\frac{1+c_0}{1-c_0}}\left(\sqrt{\frac{r}{n}}+\sqrt{\frac{rd}{nm}}\right), \\
\frac{1}{m}\left|\sum_{k=1}^{m}\widehat{\sigma}_k - \sum_{k=1}^{m}\sigma_k\right| &\leq \max\left\{1, \frac{2}{c-1}\right\}\frac{64c^2||\mathbf{X}||_2^2\sigma_{\max}}{n\kappa(c-1)(1-\delta)}\frac{\sqrt{1+c_0}}{1-c_0}\left(\frac{rd}{nm}+\frac{r}{n}\right)
\end{aligned}$$

with probability at least $1 - 2\exp(-c_1 d - c_1 m) - 2\exp\left(-nc_0^2/8 + \log m\right)$.

The proof of Corollary 4 is provided in Appendix C. The rate of convergence obtained in Corollary 4 matches the minimax lower bound[4] presented in Rohde and Tsybakov (2011). See more details in Theorems 5 and 6 of Rohde and Tsybakov (2011).

3.3 Calibrated Sparse Multivariate Regression

We now assume that the multivariate regression model in (1) is jointly sparse. More specifically, we assume that $\mathbf{B}^0$ has s rows with nonzero entries and define

$$\mathcal{S} = \left\{\mathbf{C} \in \mathbb{R}^{d\times m} \mid \mathbf{C}_{j*} = \mathbf{0} \text{ for all } j \text{ such that } \mathbf{B}^0_{j*} = \mathbf{0}\right\}, \tag{19}$$

$$\mathcal{N} = \left\{\mathbf{C} \in \mathbb{R}^{d\times m} \mid \mathbf{C}_{j*} = \mathbf{0} \text{ for all } j \text{ such that } \mathbf{B}^0_{j*} \neq \mathbf{0}\right\}. \tag{20}$$

We can easily verify that we have $\mathbf{B}^0 \in \mathcal{S}$ and the $L_{1,p}$ norm is decomposable with respect to the pair $(\mathcal{S}, \mathcal{N})$, i.e.,

$$||\mathbf{A}+\mathbf{C}||_{1,p} = ||\mathbf{A}||_{1,p} + ||\mathbf{C}||_{1,p} \quad \text{for} \quad \mathbf{A} \in \mathcal{S} \text{ and } \mathbf{C} \in \mathcal{N}.$$

The next corollary provides the concrete rates of convergence for the calibrated sparse multivariate regression estimator.

Corollary 5 *We assume that the design matrix* $\mathbf{X}$ *satisfies Assumption 1 with* $\mathcal{S}$ *and* $\mathcal{N}$ *chosen as in* (19) *and* (20), *and each column of* $\mathbf{X}$ *is normalized so that*

$$\frac{m^{1/2-1/p}\|\mathbf{X}_{*j}\|_2}{\sqrt{n}} = 1 \text{ for all } j = 1, ..., d. \tag{21}$$

We also assume that the rescaled noise matrix $\mathbf{W}$ *satisfies Assumption 2. By Theorem 3, for some universal constant* $c_0 \in (0,1)$ *and large enough* n, *let*

$$\lambda = \frac{2c(m^{1-1/p} + \sqrt{\log d})}{\sqrt{1-c_0}}, \tag{22}$$

4. In the fixed design setting for the low rank regression, $||\mathbf{X}||_2$ is supposed to increase as an order of $\sqrt{n}$. Thus $||\mathbf{X}||_2/\sqrt{n}$ in (18) should be viewed as a constant.

then for some $\delta < 1$, we have

$$\frac{1}{\sqrt{nm}}||\mathbf{X}\widehat{\mathbf{B}} - \mathbf{X}\mathbf{B}^0||_{\mathrm{F}} \leq \frac{8c\sigma_{\max}}{\kappa(c-1)(1-\delta)}\sqrt{\frac{1+c_0}{1-c_0}}\left(\sqrt{\frac{sm^{1-2/p}}{n}} + \sqrt{\frac{s\log d}{nm}}\right),$$

$$\frac{1}{\sqrt{m}}||\widehat{\mathbf{B}} - \mathbf{B}^0||_{\mathrm{F}} \leq \frac{8c\sigma_{\max}}{\kappa^2(c-1)(1-\delta)}\sqrt{\frac{1+c_0}{1-c_0}}\left(\sqrt{\frac{sm^{1-2/p}}{n}} + \sqrt{\frac{s\log d}{nm}}\right),$$

$$\frac{1}{m}\left|\sum_{k=1}^{m}\widehat{\sigma}_k - \sum_{k=1}^{m}\sigma_k\right| \leq \max\left\{1, \frac{2}{c-1}\right\}\frac{32c^2\sigma_{\max}}{\kappa(c-1)(1-\delta)}\frac{\sqrt{1+c_0}}{1-c_0}\left(\frac{sm^{1-2/p}}{n} + \frac{s\log d}{mn}\right)$$

with probability at least $1 - 2\exp(-2\log d) - 2\exp\left(-nc_0^2/8 + \log m\right)$.

The proof of Corollary 5 is provided in Appendix D. Note that when we choose $p = 2$, the column normalization condition (21) becomes

$$\frac{\|\mathbf{X}_{*j}\|_2}{\sqrt{n}} = 1 \text{ for all } j = 1, ..., d,$$

which is the same as (17). Then Corollary 5 implies that with high probability, we have

$$\frac{1}{\sqrt{m}}||\widehat{\mathbf{B}} - \mathbf{B}^0||_{\mathrm{F}} \leq \frac{8c\sigma_{\max}}{\kappa^2(c-1)(1-\delta)}\sqrt{\frac{1+c_0}{1-c_0}}\left(\sqrt{\frac{s}{n}} + \sqrt{\frac{s\log d}{nm}}\right). \tag{23}$$

The rate of convergence obtained in (23) matches the minimax lower bound presented in Lounici et al. (2011). See more details in Theorem 6.1 of Lounici et al. (2011).

Remark 6 *From Corollaries 4 and 5, we see that CMR achieves the same rates of convergence as the noncalibrated counterpart in parameter estimation. Moreover, the selected regularization parameter λ in* (18) *and* (22) *does not involve σ_k's. Therefore CMR makes the regularization parameter selection insensitive to $\sigma_{\max}$.*

4. Computational Algorithm

Though the $L_{2,1}$ norm is nonsmooth, it is nondifferentiable only when a task achieves exact zero residual, which is unlikely to happen in practice. This motivates us to apply the smoothing approach proposed by Nesterov (2005) to obtain a smooth approximation so that we can avoid directly evaluating the subgradient of the $L_{2,1}$ loss function. Thus we gain computational efficiency like other smooth loss functions.

4.1 Smooth Approximation

We consider the Fenchel's dual representation of the $L_{2,1}$ loss:

$$||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{2,1} = \max_{||\mathbf{U}||_{2,\infty}\leq 1}\langle \mathbf{U}, \mathbf{Y} - \mathbf{X}\mathbf{B}\rangle.$$

Let $\mu > 0$ be a smoothing parameter. The smooth approximation of the $L_{2,1}$ loss can be obtained by solving the optimization problem

$$||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{\mu} = \max_{||\mathbf{U}||_{2,\infty}\leq 1}\langle \mathbf{U}, \mathbf{Y} - \mathbf{X}\mathbf{B}\rangle - \frac{\mu}{2}||\mathbf{U}||_{\mathrm{F}}^2. \tag{24}$$

Note that the equality in (24) is attained with $\mathbf{U} = \widehat{\mathbf{U}}^{\mathbf{B}}$:

$$\widehat{\mathbf{U}}^{\mathbf{B}}_{*k} = \frac{\mathbf{Y}_{*k} - \mathbf{X}\mathbf{B}_{*k}}{\max\{||\mathbf{Y}_{*k} - \mathbf{X}\mathbf{B}_{*k}||_2, \mu\}}.$$

Nesterov (2005) has shown that $||\mathbf{Y} - \mathbf{X}\mathbf{B}||_\mu$ have good computational structures: (1) It is convex and differentiable with respect to $\mathbf{B}$; (2) Its gradient takes a simple form as

$$\mathbf{G}^\mu(\mathbf{B}) = \frac{\partial ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_\mu}{\partial \mathbf{B}} = \frac{\partial \left(\langle \widehat{\mathbf{U}}^{\mathbf{B}}, \mathbf{Y} - \mathbf{X}\mathbf{B} \rangle - \mu ||\widehat{\mathbf{U}}^{\mathbf{B}}||_{\mathrm{F}}^2 / 2 \right)}{\partial \mathbf{B}} = -\mathbf{X}^T \widehat{\mathbf{U}}^{\mathbf{B}};$$

(3) Let $\gamma = ||\mathbf{X}^T\mathbf{X}||_2$, we have that $\mathbf{G}^\mu(\mathbf{B})$ is Lipschitz continuous in $\mathbf{B}$ with the Lipschitz constant γ/μ, i.e., for any $\mathbf{B}'$, $\mathbf{B}'' \in \mathbb{R}^{d \times m}$,

$$||\mathbf{G}^\mu(\mathbf{B}') - \mathbf{G}^\mu(\mathbf{B}'')||_{\mathrm{F}} \leq \frac{\gamma}{\mu} ||\mathbf{B}' - \mathbf{B}''||_{\mathrm{F}}.$$

Therefore we consider a smoothed replacement of the optimization problem in (10):

$$\widetilde{\mathbf{B}} = \operatorname*{argmin}_{\mathbf{B}} ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_\mu + \lambda \mathcal{R}(\mathbf{B}). \tag{25}$$

4.2 Smoothed Proximal Gradient Algorithm

We then present a brief derivation of the smoothed proximal gradient algorithm for solving (25). We first define three sequences of auxiliary variables $\{\mathbf{A}^{(t)}\}$, $\{\mathbf{V}^{(t)}\}$, and $\{\mathbf{H}^{(t)}\}$ with $\mathbf{A}^{(0)} = \mathbf{H}^{(0)} = \mathbf{V}^{(0)} = \mathbf{B}^{(0)}$, a sequence of weights $\{\theta_t = 2/(t+1)\}$, and a nonincreasing sequence of step sizes $\{\eta_t\}_{t=0}^\infty$.

At the t^{th} iteration, we take $\mathbf{V}^{(t)} = (1 - \theta_t)\mathbf{B}^{(t-1)} + \theta_t \mathbf{A}^{(t-1)}$. Let $\widetilde{\mathbf{H}}^{(t)} = \mathbf{V}^{(t)} - \eta_t \mathbf{G}^\mu(\mathbf{V}^{(t)})$. When $\mathcal{R}(\mathbf{H}) = ||\mathbf{H}||_*$, we take

$$\mathbf{H}^{(t)} = \sum_{j=1}^{\min\{d,m\}} \max\left\{ \psi_j(\widetilde{\mathbf{H}}^{(t)}) - \eta_t \lambda, 0 \right\} \boldsymbol{u}_j \boldsymbol{v}_j^T,$$

where $\boldsymbol{u}_j$ and $\boldsymbol{v}_j$ are the left and right singular vectors of $\widetilde{\mathbf{H}}^{(t)}$ corresponding to the j^{th} largest singular value $\psi_j(\widetilde{\mathbf{H}}^{(t)})$. When $\mathcal{R}(\mathbf{H}) = ||\mathbf{H}||_{1,2}$, we take

$$\mathbf{H}^{(t)}_{j*} = \widetilde{\mathbf{H}}_{j*} \cdot \max\left\{ 1 - \eta_t \lambda / ||\widetilde{\mathbf{H}}_{j*}||_2, 0 \right\}.$$

See more details about other choices of p in the $L_{1,p}$ norm in Liu et al. (2009a); Liu and Ye (2010). To ensure that the objective function value is nonincreasing, we choose

$$\mathbf{B}^{(t)} = \operatorname*{argmin}_{\mathbf{B} \in \{\mathbf{H}^{(t)},\ \mathbf{B}^{(t-1)}\}} ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_\mu + \lambda \mathcal{R}(\mathbf{B}).$$

For simplicity, we can set $\{\eta_t\}$ as a constant sequence, e.g., $\eta_t = \mu/\gamma$ for $t = 1, 2, ...$ In practice, we cam use the backtracking line search to adjust η_t and boost the performance. At last, we take $\mathbf{A}^{(t)} = \mathbf{B}^{(t-1)} + \frac{1}{\theta_t}(\mathbf{H}^{(t)} - \mathbf{B}^{(t-1)})$. Given a stopping precision ε, the algorithm stops when $\max\left\{ ||\mathbf{B}^{(t)} - \mathbf{B}^{(t-1)}||_{\mathrm{F}},\ ||\mathbf{H}^{(t)} - \mathbf{H}^{(t-1)}||_{\mathrm{F}} \right\} \leq \varepsilon$.

Remark 7 *The smoothed proximal gradient algorithm has a worst-case iteration complexity of $\mathcal{O}(1/\epsilon)$, where ϵ is a pre-specified accuracy of the objective value*[5]*. See more details in Nesterov (2005); Beck and Teboulle (2009a).*

5. During this paper was under review, a dual proximal gradient algorithm was proposed for solving (10). See more details in Gong et al. (2014).

5. Numerical Experiments

To compare the finite-sample performance between the calibrated multivariate regression (CMR) and ordinary multivariate regression (OMR), we conduct numerical experiments on both simulated and real data sets.

5.1 Simulated Data

We generate training data sets of 400 samples for the low rank setting and 200 samples for joint sparsity setting. In details, for the low rank setting, we use the following data generation scheme:

(1) Generate each row of the design matrix $\mathbf{X}_{i*}$, $i = 1, ..., 400$, independently from a 200-dimensional normal distribution $N(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma}_{jj} = 1$ and $\mathbf{\Sigma}_{j\ell} = 0.5$ for all $\ell \neq j$.

(2) Generate the regression coefficient matrix $\mathbf{B}^0 = \mathbf{L}\mathbf{R}^T$, where $\mathbf{L} \in \mathbb{R}^{200\times 3}$, $\mathbf{R} \in \mathbb{R}^{3\times 101}$, and all entries of $\mathbf{L}$ and $\mathbf{R}$ are independently generated from $N(0, 0.05)$.

(3) Generate the random noise matrix $\mathbf{Z} = \mathbf{W}\mathbf{D}$ where $\mathbf{W} \in \mathbb{R}^{400\times 101}$ with all entries of $\mathbf{W}$ independently generated from $N(0, 1)$ and $\mathbf{D}$ is either of the following matrices

$$\mathbf{D} = \sigma_{\max} \cdot \operatorname{diag}\left(2^{0/100}, 2^{-3/100}, \cdots, 2^{-297/100}, 2^{-300/100}\right) \in \mathbb{R}^{101\times 101}, , \tag{26}$$

$$\mathbf{D} = \sigma_{\max} \cdot \operatorname{diag}\left(1, 1, \cdots, 1, 1\right) \in \mathbb{R}^{101\times 101}. \tag{27}$$

For the joint sparsity setting, we use the following data generation scheme:

(1) Generate each row of the design matrix $\mathbf{X}_{i*}$, $i = 1, ..., 200$, independently from a 800-dimensional normal distribution $N(\mathbf{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma}_{jj} = 1$ and $\mathbf{\Sigma}_{j\ell} = 0.5$ for all $\ell \neq j$.

(2) Let $k = 1, \ldots, 13$, set the regression coefficient matrix $\mathbf{B}^0 \in \mathbb{R}^{800\times 13}$ as $\mathbf{B}^0_{1k} = 3$, $\mathbf{B}^0_{2k} = 2$, $\mathbf{B}^0_{4k} = 1.5$, and $\mathbf{B}^0_{jk} = 0$ for all $j \neq 1, 2, 4$.

(3) Generate the random noise matrix $\mathbf{Z} = \mathbf{W}\mathbf{D}$, where $\mathbf{W} \in \mathbb{R}^{200\times 13}$ with all entries of $\mathbf{W}$ independently generated from $N(0, 1)$ and $\mathbf{D}$ is is either of the following matrices

$$\mathbf{D} = \sigma_{\max} \cdot \operatorname{diag}\left(2^{0/4}, 2^{-1/4}, \cdots, 2^{-11/4}, 2^{-12/4}\right) \in \mathbb{R}^{13\times 13}, \tag{28}$$

$$\mathbf{D} = \sigma_{\max} \cdot \operatorname{diag}\left(1, 1, \cdots, 1, 1\right) \in \mathbb{R}^{13\times 13}. \tag{29}$$

In addition, we generate validation sets (400 samples for the low rank setting and 200 samples for the joint sparsity setting) for the regularization parameter selection, and testing sets (10,000 samples for both settings) to evaluate the prediction accuracy.

Remark 8 *The scale matrices in* (26) *and* (28) *consider the scenario, where the regression tasks have different variances. The scale matrices in* (27) *and* (29) *consider the scenario, where all regression tasks have the equal variance.*

In numerical experiments, we set $\sigma_{\max} = 1,\ 2$, and 4 to illustrate the tuning insensitivity of CMR. The regularization parameter λ of both CMR and OMR is chosen over a grid

$$\mathbf{\Lambda} = \left\{2^{40/4}\lambda_0, 2^{39/4}\lambda_0, \cdots, 2^{-17/4}\lambda_0, 2^{-18/4}\lambda_0\right\}.$$

We choose

$$\lambda_0 = \frac{||\mathbf{X}||_2}{n}(\sqrt{d} + \sqrt{m}) \ \text{ and } \ \lambda_0 = \sqrt{\log d} + \sqrt{m}$$

for the low rank and joint sparsity settings. The optimal regularization parameter $\widehat{\lambda}$ is determined by the prediction error as

$$\widehat{\lambda} = \underset{\lambda \in \mathbf{\Lambda}}{\operatorname{argmin}} ||\widetilde{\mathbf{Y}} - \widetilde{\mathbf{X}}\widehat{\mathbf{B}}^{\lambda}||_{\mathrm{F}}^2,$$

where $\widehat{\mathbf{B}}^{\lambda}$ denotes the obtained estimate using the regularization parameter λ, and $\widetilde{\mathbf{X}}$ and $\widetilde{\mathbf{Y}}$ denote the design and response matrices of the validation set.

Since the noise level σ_k's may vary across different regression tasks, we adopt the following three criteria to evaluate the empirical performance:

$$\text{P.E.} = \frac{1}{10000}||\overline{\mathbf{Y}} - \overline{\mathbf{X}}\widehat{\mathbf{B}}||_{\mathrm{F}}^2, \ \text{A.P.E.} = \frac{1}{10000m}||(\overline{\mathbf{Y}} - \overline{\mathbf{X}}\widehat{\mathbf{B}})\mathbf{D}^{-1}||_{\mathrm{F}}^2, \ \text{E.E.} = \frac{1}{m}||\widehat{\mathbf{B}} - \mathbf{B}^0||_{\mathrm{F}}^2,$$

where $\overline{\mathbf{X}}$ and $\overline{\mathbf{Y}}$ denote the design and response matrices of the testing set.

All simulations are implemented by MATLAB using a PC with Intel Core i5 3.3GHz CPU and 16GB memory. We set $p = 2$ for the joint sparsity setting, but it is straightforward to extend to arbitrary $p > 2$. OMR is solved by the monotone fast proximal gradient algorithm, where we set the stopping precision $\varepsilon = 10^{-4}$. CMR is solved by the proposed smoothed proximal gradient algorithm, where we set the stopping precision $\varepsilon = 10^{-4}$, and the smoothing parameter $\mu = 10^{-4}$.

We compare the statistical performance between CMR and OMR. Tables 1-4 summarize the results averaged over 200 simulations for both settings. In addition, since we know the true values of σ_k's, we also present the results of the oracle estimator $\widehat{\mathbf{B}}^*$ defined in (11). The oracle estimator is only for comparison purpose, and it is not a practical estimator.

Tables 1 and 3 present the empirical results when we adopt the scale matrix $\mathbf{D}$ defined in (26) and (28) to generate the random noise. Though our theoretical analysis in §3 only shows CMR attains the same rates of convergence as OMR, our empirical results show that CMR universally outperforms OMR, and achieves almost the same performance as the oracle estimator. These results corroborate the effectiveness of the calibration for each task.

$\sigma_{\max}$	Method	P.E.	A.P.E.	E.E.
1	Oracle	48.394(0.7421)	1.1659(0.0241)	0.1106(0.0245)
	CMR	48.411(0.7431)	1.1668(0.0214)	0.1109(0.0133)
	OMR	53.337(0.7063)	1.2880(0.0231)	0.2077(0.0137)
2	Oracle	183.38(0.9786)	1.0917(0.0068)	0.2425(0.0187)
	CMR	183.40(1.2212)	1.0924(0.0063)	0.2430(0.0238)
	OMR	194.66(1.4109)	1.1641(0.0112)	0.4637(0.0277)
4	Oracle	713.13(3.3923)	1.0554(0.0062)	0.5696(0.0669)
	CMR	713.24(2.7685)	1.0565(0.0047)	0.5737(0.0533)
	OMR	728.55(2.6500)	1.0793(0.0051)	0.8722(0.0526)

Table 1: Quantitative comparison of the statistical performance between CMR and OMR for the low rank setting with $\mathbf{D}$ defined in (26). The results are averaged over 200 simulations with the standard errors in parentheses. CMR universally outperforms OMR, and achieves almost the same performance as the oracle estimator.

Tables 2 and 4 present the empirical results when we adopt the scale matrix $\mathbf{D}$ defined in (27) and (29) with all σ_k's being equal. We can see that CMR attains similar performance to OMR. This indicates that CMR is a safe replacement of OMR for multivariate regressions.

$\sigma_{\max}$	Method	P.E.	A.P.E.	E.E.
1	CMR	112.13(1.0051)	1.1103(0.0097)	0.2128(0.0190)
	OMR	113.69(1.0163)	1.1951(0.0100)	0.2217(0.0193)
2	CMR	428.42(1.8061)	1.0605(0.0043)	0.4576(0.0344)
	OMR	430.73(1.9636)	1.0758(0.0053)	0.4752(0.0413)
4	CMR	1669.2(5.0401)	1.0335(0.0028)	0.9621(0.0991)
	OMR	1673.5(5.7879)	1.0378(0.0035)	1.0353(0.1104)

Table 2: Quantitative comparison of the statistical performance between CMR and OMR for the low rank setting with $\mathbf{D}$ defined in (27). The results are averaged over 200 simulations with the standard errors in parentheses. CMR and OMR achieve similar statistical performance.

$\sigma_{\max}$	Method	P.E.	A.P.E.	E.E.
1	Oracle	5.8759(0.0834)	1.0454(0.0149)	0.0245(0.0086)
	CMR	5.8761(0.0669)	1.0459(0.0122)	0.0249(0.0078)
	OMR	5.9012(0.0701)	1.0581(0.0162)	0.0290(0.0091)
2	Oracle	23.464(0.3237)	1.0441(0.0148)	0.0926(0.0342)
	CMR	23.465(0.2600)	1.0446(0.0131)	0.0928(0.0268)
	OMR	23.580(0.2832)	1.0573(0.0170)	0.1115(0.0365)
4	Oracle	93.532(0.8843)	1.0418(0.0962)	0.3342(0.1255)
	CMR	93.542(0.9788)	1.0421(0.0113)	0.3346(0.1002)
	OMR	94.094(1.0978)	1.0550(0.0166)	0.4125(0.1417)

Table 3: Quantitative comparison of the statistical performance between CMR and OMR for the joint sparsity setting with $\mathbf{D}$ defined in (28). The results are averaged over 200 simulations with the standard errors in parentheses. CMR universally outperforms OMR, and achieves almost the same performance as the oracle estimator.

$\sigma_{\max}$	Method	P.E.	A.P.E.	E.E.
1	CMR	13.565(0.1411)	1.0435(0.0156)	0.0599(0.0199)
	OMR	13.697(0.1554)	1.0486(0.0142)	0.0607(0.0128)
2	CMR	54.171(0.5791)	1.0418(0.0101)	0.2252(0.0644)
	OMR	54.221(0.6173)	1.0427(0.0118)	0.2359(0.0821)
4	CMR	215.98(1.994)	1.0384(0.0099)	0.80821(0.2417)
	OMR	216.19(2.391)	1.0394(0.0114)	0.81957(0.3180)

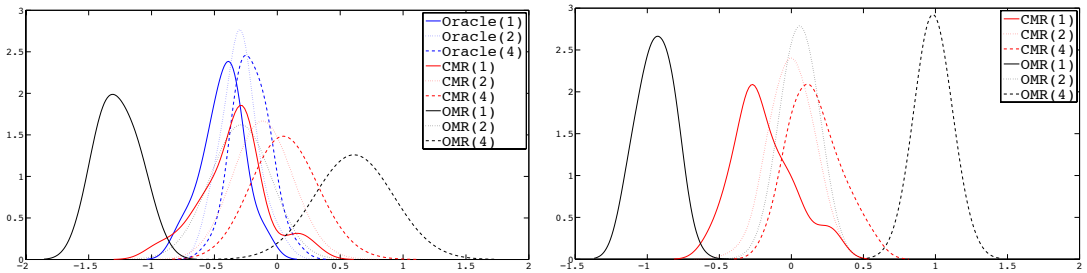
Table 4: Quantitative comparison of the statistical performance between CMR and OMR for the joint sparsity setting with $\mathbf{D}$ defined in (29). The results are averaged over 200 simulations with the standard errors in parentheses. CMR and OMR achieve similar statistical performance.

In addition, we also examine the optimal regularization parameters for CMR and OMR over all replicates. We visualize the distribution of all 200 selected $\widehat{\lambda}$'s using the kernel density estimator. In particular, we adopt the Gaussian kernel, and select the kernel bandwidth based on the 10-fold cross validation. Figure 1 illustrates the estimated density functions. The horizontal axis corresponds to the rescaled regularization parameter as follows:

$$\text{Low Rank} \quad : \quad \log\left(\frac{\widehat{\lambda}}{(\sqrt{d}+\sqrt{m})||\mathbf{X}||_2/n}\right),$$

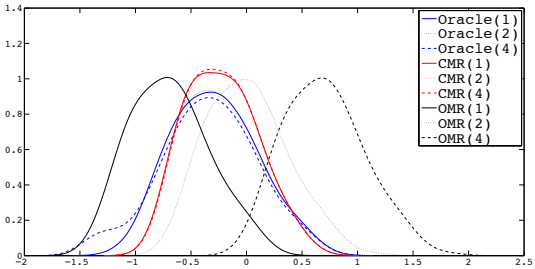
$$\text{Joint Sparsity} \quad : \quad \log\left(\frac{\widehat{\lambda}}{\sqrt{\log d}+\sqrt{m}}\right).$$

We see that the optimal regularization parameters of OMR significantly vary with different $\sigma_{\max}$. In contrast, the optimal regularization parameters of CMR are more concentrated. This is consistent with our claimed tuning insensitivity.

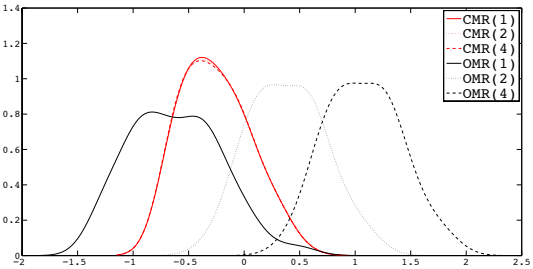


(a) The low rank setting with $\mathbf{D}$ defined in (26)

(b) The low rank setting with $\mathbf{D}$ defined in (27)



(c) The joint sparsity setting with $\mathbf{D}$ defined in (28)



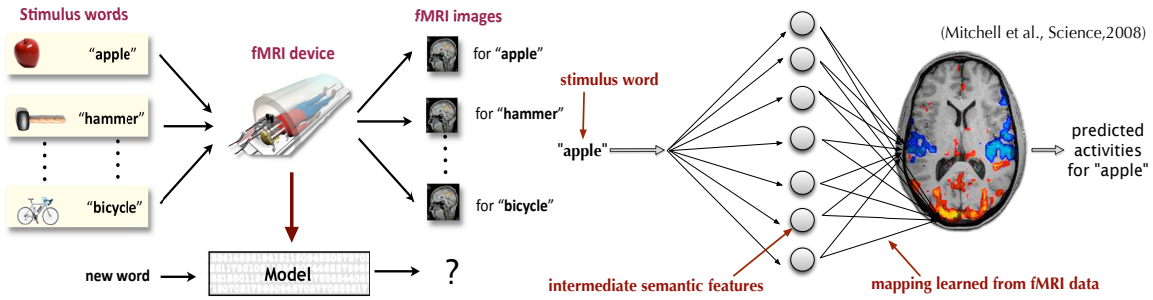
(d) The joint sparsity setting with $\mathbf{D}$ defined in (29)

Figure 1: The distributions of the selected regularization parameters using the kernel density estimator. The numbers in the parentheses are $\sigma_{\max}$'s. The optimal regularization parameters of OMR are more spread with different $\sigma_{\max}$ than those of CMR and the oracle estimator.

5.2 Real Data

We apply CMR on a brain activity prediction problem which aims to build a parsimonious model to predict a person's neural activity when seeing a stimulus word. As is illustrated in Figure 2, for a given stimulus word, we first encode it into an intermediate semantic feature vector using some corpus statistics. We then model the brain's neural activity pattern using CMR. Creating such a predictive model not only enables us to explore new analytical tools for the fMRI data, but also helps us to gain deeper understanding on how human brain represents knowledge (Mitchell et al., 2008). As will be shown in the section, prediction based on the features selected by CMR significantly

outperforms that based on the features selected by OMR, and is even better than that based on the handcrafted features selected by human experts.



(a) illustration of the data collection procedure (b) model for predicting fMRI brain activity pattern

Figure 2: An illustration of the fMRI brain activity prediction problem (Mitchell et al., 2008). (a) To collect the data, a human participant sees a sequence of English words and their images. The corresponding fMRI images are recorded to represent the brain activity patterns; (b) To build a predictive model, each stimulus word is encoded into intermediate semantic features (e.g. the co-occurrence statistics of this stimulus word in a large text corpus). These intermediate features can then be used to predict the brain activity pattern.

5.2.1 Data

The data are obtained from Mitchell et al. (2008) and contain a fMRI image data set and a text data set. The fMRI data are collected from an experiment with 9 participants.60 nouns are selected as stimulus words from 12 different categories (See Table 5). When a participant sees a stimulus word, the fMRI device records an image[6]. Each image contains 20,601 voxels that represent the neural activities of the participant's brain. Therefore the total number of images is $9 \times 60 = 540$. Since many of the 20,601 voxels are noisy, Mitchell et al. (2008) exploit a "stability score" approach to extract 500 most stable voxels. See more details in Mitchell et al. (2008).

The text data set is collected from the Google Trillion Word corpus[7]. It contains the co-occurrence frequencies of the 60 stimulus words with 5,000 most frequent English words in the corpus with 100 stop words removed. In Mitchell et al. (2008), 25 sensory-action verbs (See Table 6) are handcrafted by human experts based on the domain knowledge of cognitive neuroscience. These 25 words are closely related to the 60 stimulus words in their semantics meanings. For example, "eat" is related to vegetables such as "lettuce" or "tomato", and "wear" is related to clothing such as "shirt" and "dress".

When building multivariate linear models, Mitchell et al. (2008) use the co-occurrence frequencies of each stimulus word with 25 sensory verbs as covariates and use the corresponding fMRI image as response. They estimate a 25-dimensional multivariate linear model by the ridge regression. They show that the obtained predictive model significantly outperforms random guess. Thus, they treat these 25 words as a semantic basis.

In our experiment below, we apply CMR to automatically select a semantic basis from all 5,000 most frequent English words. Compared with the protocol used in Mitchell et al. (2008), our approach is completely data-driven and outperforms the handcraft method in the brain activity prediction accuracy for 5 out of 9 participants.

6. Each image is actually the average of 6 consecutive recordings of each word.
7. `http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html`

Category	Exemplar 1	Exemplar 2	Exemplar 3	Exemplar 4	Exemplar 5
animals	bear	cat	cow	dog	horse
body parts	arm	eye	foot	hand	leg
buildings	apartment	barn	church	house	igloo
building parts	arch	chimney	closet	door	window
clothing	coat	dress	pants	shirt	skirt
furniture	bed	chair	desk	dresser	table
insects	ant	bee	beetle	butterfly	fly
kitchen utensils	bottle	cup	glass	knife	spoon
man made objects	bell	key	refrigerator	telephone	watch
tools	chisel	hammer	pliers	saw	screwdriver
vegetables	carrot	celery	corn	lettuce	tomato
vehicles	airplane	bicycle	car	train	truck

Table 5: The 60 stimulus words used in Mitchell et al. (2008) from 12 categories (5 per category).

See	Eat	Run	Say	Enter
Hear	Touch	Push	Fear	Drive
Listen	Rub	Fill	Open	Wear
Taste	Approach	Move	Lift	Break
Smell	Manipulate	Ride	Near	Clean

Table 6: The 25 verbs used in Mitchell et al. (2008). They are handcrafted based on the domain knowledge of cognitive science, and are independent on the data set.

5.2.2 Experimental Protocol in Mitchell et al. (2008)

The evaluation procedure of Mitchell et al. (2008) is based on the leave-two-out cross validation over all $\binom{60}{2} = 1,770$ possible partitions. In each partition, we select 58 stimulus words out of 60 as the training set. Recall that each stimulus word is represented by 5,000 features and each feature is the co-occurrence frequency of a potential basis word with the stimulus word, we obtain a $58 \times 5,000$ design matrix. Similarly, we can format the fMRI images corresponding to the 58 training stimulus words into a 58×500 response matrix. In the training stage, we apply CMR and OMR to select 25 basis words by adjusting the regularization parameters. We then use the remaining two stimulus words as a validation set and apply the estimated models to predict the neural activity of these two stimulus words. We evaluate the prediction performance based on the combined cosine similarity measure defined as follow.

Definition 9 (Combined Similarity Measure, Mitchell et al. (2008)) *Let $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^m$ denote the observed fMRI images of two stimulus words in the validation set, and $\widehat{\boldsymbol{u}} \in \mathbb{R}^m$ and $\widehat{\boldsymbol{v}} \in \mathbb{R}^m$ denote the corresponding predicted fMRI images. We say that the predicted images $\widehat{\boldsymbol{u}}$ and $\widehat{\boldsymbol{v}}$ correctly label two validation stimulus words, if*

$$\cos(\boldsymbol{u}, \widehat{\boldsymbol{u}}) + \cos(\boldsymbol{v}, \widehat{\boldsymbol{v}}) > \cos(\boldsymbol{u}, \widehat{\boldsymbol{v}}) + \cos(\boldsymbol{v}, \widehat{\boldsymbol{u}}), \tag{30}$$

where $\cos(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{u}^T \boldsymbol{v})/(||\boldsymbol{u}||_2 ||\boldsymbol{v}||_2)$.

We then summarize the overall prediction accuracy for each participant by the percentage of the correct labelings over all 1,770 partitions. Table 7 presents the prediction accuracies for the 9

participants. We see that CMR universally outperforms OMR across all 9 participants by 4.42% on average. Note that the statistically significant accuracy at 5% level is 0.61, CMR achieves statistically significant advantages for 8 out of 9 participants.

Method	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9
CMR	0.783	0.724	0.748	0.528	0.772	0.713	0.728	0.739	0.763
OMR	0.749	0.685	0.732	0.485	0.724	0.661	0.688	0.682	0.693

Table 7: Prediction accuracies evaluated using the experimental protocol in Mitchell et al. (2008). CMR universally outperforms OMR across all participants.

5.2.3 An Improved Experimental Protocol

There are two drawbacks of the previous protocol: (1) The selected basis words vary a lot across different partitions of the cross validation and participants. Such high variability makes the obtained results difficult to interpret; (2) The automatic semantic basis selection method of CMR and OMR is sensitive to data outliers, which are common in fMRI studies. In this section, we improve this protocol to address these two problems in a more data-driven manner.

Our main idea is to simultaneously exploit the training data of multiple participants and use the stability criterion to select more stable semantic basis words (Meinshausen and Bühlmann, 2010). In detail, for each participant to be evaluated, we choose three other representatives out of the remaining eight according to who achieve the best three leave-two-out cross validation prediction accuracies in Table 7. Taking Participant 2 and CMR as an example, the three selected representatives are Participants 1, 3, and 9 with the three highest accuracies of 0.783, 0.772, and 0.763. In this way, we could eliminate the effects of possible data outliers. We then combine the fMRI images of three representatives and formulate a multivariate regression problem with 1,500 dimensional response. We conduct the leave-two-out cross validation as in the previous protocol using the combined data set, and count the frequency of each potential basis word that appears in all 1,770 partitions. We then choose the 25 most frequent words as the semantic basis. Finally, we apply the same procedure as in the previous protocol on the current candidate participant and evaluate the prediction accuracy using the combined cosine score.

Table 8 summarizes the prediction performance based on this improved protocol. We also report the results obtained by the 25 handcrafted basis. Compared with the results in Table 7, we see that the performance of CMR is greatly improved. For Participants 1, 2, 3, 5, and 8, the prediction performance of CMR significantly outperforms the handcraft method. Moreover, since the candidate participant is not involved in the semantic basis word selection, our results imply that the selected semantic basis have good generalization capability across participants.

Method	P. 1	P. 2	P. 3	P. 4	P. 5	P. 6	P. 7	P. 8	P. 9
CMR	0.840	0.794	0.861	0.651	0.823	0.722	0.738	0.720	0.780
OMR	0.803	0.789	0.801	0.602	0.766	0.623	0.726	0.749	0.765
Handcraft	0.822	0.776	0.773	0.727	0.782	0.865	0.734	0.685	0.819

Table 8: Prediction accuracies evaluated used a more heuristic protocol. CMR significantly outperforms the handcrafted basis words for 5 out of 9 participants.

Table 9 lists 35 basis words obtained by CMR using the improved protocol. The words in the bold font are common ones shared by all 9 participants. We see that our list contains nouns, adjectives, and verbs. These words are closely related to the 60 stimulus words. For example, lodge, hotel, and floor are closely related to "building" and "building parts"; green and fruit clearly refer to words in "vegetable"; built and using are related to "tools" and "man made objects".

av	balls	booking	**built**	cartoon	cream
cut	**country**	**discounts**	floor	fruit	green
hold	holidays	**hotel**	interior	kill	liquid
located	lodge	**log**	measure	**mesh**	**near**
offers	put	**reg**	room	sale	**separate**
shipping	**soft**	usd	**using**	went	

Table 9: The 35 basis words selected by CMR using the improved protocol. The words in the bold font are shared by predictive models for all 9 participants.

6. Discussion and Conclusion

Two other related methods are the square-root low rank multivariate regression (Klopp, 2011) and the square-root sparse multivariate regression (Bunea et al., 2013). They solve the convex program

$$\widehat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \, ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{\mathrm{F}} + \lambda \mathcal{R}(\mathbf{B}). \tag{31}$$

The Frobenius loss in (31) makes the regularization parameter selection independent of $\sigma_{\max}$, but it does not calibrate different regression tasks. We can rewrite (31) as

$$(\widehat{\mathbf{B}}, \widehat{\sigma}) = \underset{\mathbf{B},\sigma}{\operatorname{argmin}} \; \frac{1}{\sqrt{nm}\sigma} ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{\mathrm{F}}^2 + \lambda \mathcal{R}(\mathbf{B}) \;\; \text{subject to } \sigma = \frac{1}{\sqrt{nm}} ||\mathbf{Y} - \mathbf{X}\mathbf{B}||_{\mathrm{F}}. \tag{32}$$

Since σ in (32) is not specific to any individual task, it cannot calibrate the regularization. Thus it is fundamentally different from CMR.

The calibration technique proposed in this paper is quite general, and can be extended to more sophisticated scenarios, e.g. the regularization function is weakly decomposable or geometrically decomposable (Geer, 2014; Lee et al., 2013), or the regression coefficient matrix can be decomposed into multiple structured matrices (Agarwal et al., 2012; Chen et al., 2011; Gong et al., 2012; Jalali et al., 2010; Obozinski et al., 2010). Accordingly, the extensions of our proposed theory are also straightforward. We only need to replace their squared Frobenius loss-based analysis with the $L_{2,1}$ loss based analysis in this paper.

Appendix A. Proof of Lemma 2

Note that the following two relations are frequently used in our analysis,

$$\mathbf{Y} - \mathbf{X}\mathbf{B}^0 = \mathbf{X}\mathbf{B}^0 + \mathbf{Z} - \mathbf{X}\mathbf{B}^0 = \mathbf{Z} \quad \text{and} \quad \mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}} = \mathbf{X}\mathbf{B}^0 + \mathbf{Z} - \mathbf{X}\widehat{\mathbf{B}} = \mathbf{Z} - \mathbf{X}\widehat{\mathbf{\Delta}}.$$

Proof Since $\mathbf{B}^0 \in \mathcal{S}$, we have $\mathbf{B}^0_{\mathcal{S}_\perp} = \mathbf{0}$. Then we have

$$\mathcal{R}(\widehat{\mathbf{B}}) = \mathcal{R}(\mathbf{B}^0 + \widehat{\mathbf{\Delta}}) = \mathcal{R}(\mathbf{B}^0_{\mathcal{S}} + \widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp} + \widehat{\mathbf{\Delta}}_{\mathcal{N}}) \geq \mathcal{R}(\mathbf{B}^0_{\mathcal{S}} + \widehat{\mathbf{\Delta}}_{\mathcal{N}}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}). \tag{33}$$

Since $\mathcal{R}(\cdot)$ is decomposable with respect to $(\mathcal{S},\mathcal{N})$, (33) further implies

$$\mathcal{R}(\widehat{\mathbf{B}}) \geq \mathcal{R}(\mathbf{B}^0_{\mathcal{S}}) + \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}). \tag{34}$$

Since $\mathbf{B}^0 \in \mathcal{S}$, we have $\mathcal{R}(\mathbf{B}^0) = \mathcal{R}(\mathbf{B}^0_{\mathcal{S}})$. Then by rearranging (34), we obtain

$$\mathcal{R}(\mathbf{B}^0) - \mathcal{R}(\widehat{\mathbf{B}}) \leq \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}}). \tag{35}$$

Since $\widehat{\mathbf{B}}$ is the optimum to (10), by (34), we further have

$$||\mathbf{X}\widehat{\mathbf{\Delta}} - \mathbf{Z}||_{2,1} - ||\mathbf{Z}||_{2,1} \leq \lambda(\mathcal{R}(\mathbf{B}^0) - \mathcal{R}(\mathbf{B}^0 + \widehat{\mathbf{\Delta}})) \leq \lambda(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})). \tag{36}$$

Due to the convexity of $||\cdot||_{2,1}$, we know

$$||\mathbf{X}\widehat{\mathbf{\Delta}} - \mathbf{Z}||_{2,1} - ||\mathbf{Z}||_{2,1} \geq \langle \mathbf{G}^0, \widehat{\mathbf{\Delta}} \rangle \geq -|\langle \mathbf{G}^0, \widehat{\mathbf{\Delta}} \rangle|. \tag{37}$$

By the Cauchy-Schwarz inequality, we obtain

$$|\langle \mathbf{G}^0, \widehat{\mathbf{\Delta}} \rangle| \leq \mathcal{R}^*(\mathbf{G}^0)\mathcal{R}(\widehat{\mathbf{\Delta}}) \leq \frac{\lambda}{c}(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) + \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})), \tag{38}$$

where the last inequality comes from the assumption $\lambda \geq c\mathcal{R}^*(\mathbf{G}^0)$ and the triangle inequality $\mathcal{R}(\widehat{\mathbf{\Delta}}) \leq \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) + \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})$. By combining (36), (37), and (38), we obtain

$$-\frac{\lambda}{c}(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) + \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})) \leq \lambda(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})). \tag{39}$$

By rearranging (39), we obtain $(c-1)\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}}) \leq (c+1)\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp})$, which completes the proof. ■

Appendix B. Proof of Theorem 3

Proof We have

$$\begin{aligned} &||\mathbf{X}\widehat{\mathbf{\Delta}} - \mathbf{Z}||_{2,1} - ||\mathbf{Z}||_{2,1} = \sum_{k=1}^{m}(||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k} - \mathbf{Z}_{*k}||_2 - ||\mathbf{Z}_{*k}||_2) \\ &= \sum_{k=1}^{m} \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2^2 - 2(\mathbf{X}\widehat{\mathbf{\Delta}}_{*k})^T\mathbf{Z}_{*k}}{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k} - \mathbf{Z}_{*k}||_2 + ||\mathbf{Z}_{*k}||_2} \geq \sum_{k=1}^{m} \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2^2}{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2 + 2||\mathbf{Z}_{*k}||_2} - 2\sum_{k=1}^{m} \frac{|\widehat{\mathbf{\Delta}}_{*k}^T\mathbf{X}^T\mathbf{Z}_{*k}|}{||\mathbf{Z}_{*k}||_2}. \end{aligned} \tag{40}$$

Since $\mathbf{G}^0_{*k} = \mathbf{X}^T\mathbf{Z}_{*k}/||\mathbf{Z}_{*k}||_2$, we have

$$\sum_{k=1}^{m} \frac{|\widehat{\mathbf{\Delta}}_{*k}^T\mathbf{X}^T\mathbf{Z}_{*k}|}{||\mathbf{Z}_{*k}||_2} = \sum_{k=1}^{m} |\widehat{\mathbf{\Delta}}_{*k}^T\mathbf{G}^0_{*k}| \leq \sum_{k=1}^{m}\sum_{j=1}^{d} |\widehat{\mathbf{\Delta}}_{jk}\mathbf{G}^0_{jk}| \leq \mathcal{R}^*(\mathbf{G}^0)\mathcal{R}(\widehat{\mathbf{\Delta}}), \tag{41}$$

where the last inequality follows from the Cauchy-Schwarz inequality. Recall that in the proof of Lemma 2, we already have (36) as follows,

$$||\mathbf{X}\widehat{\mathbf{\Delta}} - \mathbf{Z}||_{2,1} - ||\mathbf{Z}||_{2,1} \leq \lambda(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})). \tag{42}$$

Therefore by combining (42) with (40) and (41), we obtain

$$\begin{aligned} \sum_{k=1}^{m} \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2^2}{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2 + 2||\mathbf{Z}_{*k}||_2} &\leq \lambda\big(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) - \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})\big) + 2\mathcal{R}^*(\mathbf{G}^0)\mathcal{R}(\widehat{\mathbf{\Delta}}) \\ &\leq \lambda\,(1 + 2/c)\,\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) + \lambda\,(2/c - 1)\,\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}}) \leq \frac{2\lambda}{c-1}\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}), \end{aligned} \tag{43}$$

where the second inequality comes from the assumption $\lambda \geq c\mathcal{R}^*(\mathbf{G}^0)$ and the triangle inequality $\mathcal{R}(\widehat{\mathbf{\Delta}}) \leq \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) + \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})$, and the last inequality comes from (13) in Lemma 2. Meanwhile, by the triangle inequality, we also have

$$\sum_{k=1}^{m} \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2^2}{||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2 + 2||\mathbf{Z}_{*k}||_2} \geq \frac{\sum_{k=1}^{m} ||\mathbf{X}\widehat{\mathbf{\Delta}}_{*k}||_2^2}{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{2,\infty} + 2||\mathbf{Z}||_{2,\infty}} \geq \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}^2}{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}} + 2||\mathbf{Z}||_{2,\infty}}, \tag{44}$$

where the last inequality comes from the fact $||\mathbf{X}\widehat{\mathbf{\Delta}}||_{2,\infty} \leq ||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}$. Combining (43) and (44), we obtain

$$\frac{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}^2}{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}} + 2||\mathbf{Z}||_{2,\infty}} \leq \frac{2\lambda}{c-1}\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) \leq \frac{2\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})||\widehat{\mathbf{\Delta}}||_{\mathrm{F}}}{c-1}, \tag{45}$$

where the last inequality comes from the definition of $\Theta(\mathcal{N}_\perp, \mathcal{R})$. By Assumption 1, we can rewrite (45) as

$$||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}^2 \leq \frac{2\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})}{(c-1)\sqrt{n}\kappa}||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}^2 + \frac{4\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})}{\sqrt{n}\kappa(c-1)}||\mathbf{Z}||_{2,\infty}||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}.$$

Given $2\lambda\Theta(\mathcal{N}_\perp, \mathcal{R}) \leq \delta(c-1)\sqrt{n}\kappa$ for some $\delta < 1$, we have

$$||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}} \leq \frac{4\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})}{\sqrt{n}\kappa(c-1)(1-\delta)}||\mathbf{Z}||_{2,\infty} \leq \frac{4\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})\sigma_{\max}}{\sqrt{n}\kappa(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty}. \tag{46}$$

By Assumption 1 again, we obtain

$$||\widehat{\mathbf{\Delta}}||_{\mathrm{F}} \leq \frac{4\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})\sigma_{\max}}{n\kappa^2(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty}. \tag{47}$$

We proceed with the standard deviation estimation. By (36), we have

$$||\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}||_{2,1} - ||\mathbf{Y} - \mathbf{X}\mathbf{B}^0||_{2,1} \leq \lambda\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) - \lambda\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}}) \leq \lambda\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}). \tag{48}$$

Combining (48) with a simple variant of Assumption 1

$$\kappa \leq \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}}{\sqrt{n}||\widehat{\mathbf{\Delta}}||_{\mathrm{F}}} \leq \frac{||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}}{\sqrt{n}||\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}||_{\mathrm{F}}} \leq \frac{\Theta(\mathcal{N}_\perp, \mathcal{R})||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}}{\sqrt{n}\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp})}, \tag{49}$$

we have

$$\sqrt{n}\left(\sum_{k=1}^{m}\widehat{\sigma}_k - \sum_{k=1}^{m}\sigma_k\right) \leq \frac{\lambda\Theta(\mathcal{N}_\perp, \mathcal{R})||\mathbf{X}\widehat{\mathbf{\Delta}}||_{\mathrm{F}}}{\sqrt{n}\kappa} \leq \frac{4\lambda^2\Theta^2(\mathcal{N}_\perp, \mathcal{R})\sigma_{\max}}{n\kappa(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty}, \tag{50}$$

where the last inequality comes from (46). By (37), (38), and Lemma 2, we have

$$||\mathbf{Y} - \mathbf{X}\widehat{\mathbf{B}}||_{2,1} - ||\mathbf{Y} - \mathbf{X}\mathbf{B}^0||_{2,1} \geq -\frac{\lambda}{c}(\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}) + \mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}})) \geq -\frac{2\lambda}{c-1}\mathcal{R}(\widehat{\mathbf{\Delta}}_{\mathcal{N}_\perp}). \tag{51}$$

By (49) again, we have

$$\sqrt{n}\left(\sum_{k=1}^{m}\widehat{\sigma}_k - \sum_{k=1}^{m}\sigma_k\right) \geq -\frac{8\lambda^2\Theta^2(\mathcal{N}_\perp, \mathcal{R})\sigma_{\max}}{n\kappa(c-1)^2(1-\delta)}||\mathbf{W}||_{2,\infty}. \tag{52}$$

Thus combining (50) and (52), we have

$$\frac{1}{m}\left|\sum_{k=1}^{m}\widehat{\sigma}_k - \sum_{k=1}^{m}\sigma_k\right| \leq \max\left\{1, \frac{2}{c-1}\right\}\frac{4\lambda^2\Theta^2(\mathcal{N}_\perp, \mathcal{R})\sigma_{\max}}{\sqrt{n}mn\kappa(c-1)(1-\delta)}||\mathbf{W}||_{2,\infty}. \tag{53}$$

■

Appendix C. Proof of Corollary 4

We need to introduce the following lemmas for our proof.

Lemma 10 *Suppose that we have all entries of a random vector* $\boldsymbol{v} = (v_1, ..., v_n)^T \in \mathbb{R}^n$ *independently generated from the standard Gaussian distribution with mean* 0 *and variance* 1. *For any* $c_0 \in (0, 1)$, *we have*

$$\mathbb{P}\left(\left|||\boldsymbol{v}||_2^2 - n\right| \geq c_0 n\right) \leq 2\exp\left(-\frac{nc_0^2}{8}\right).$$

The proof of Lemma 10 is provided in Johnstone (2001), therefore omitted.

Lemma 11 *Suppose that we have all entries of* $\mathbf{W}$ *independently generated from the standard Gaussian distribution with mean* 0 *and variance* 1, *then there exists some universal constant* c_1 *such that*

$$\mathbb{P}\left(\frac{||\mathbf{X}^T\mathbf{W}||_2}{\sqrt{n}} \leq \frac{2||\mathbf{X}||_2}{\sqrt{n}}(\sqrt{m} + \sqrt{d})\right) \geq 1 - 2\exp(-c_1(d+m)). \tag{54}$$

The proof of Lemma 11 is provided in Appendix E. Now we proceed to derive the refined error bound for the calibrated low rank regression estimator.

Proof Since we have all entries of $\mathbf{W}$ independently generated from $N(0, 1)$, then by Lemma 10, for any $c_0 \in (0, 1)$, we have

$$\mathbb{P}\left(\sqrt{(1-c_0)n} \leq ||\mathbf{W}_{*k}||_2 \leq \sqrt{(1+c_0)n}\right) \geq 1 - 2\exp\left(-\frac{nc_0^2}{8}\right).$$

By taking the union bound over all $k = 1, ..., m$, we have

$$\mathbb{P}\Big(\sqrt{(1-c_0)n} \leq \min_{1\leq k\leq m} ||\mathbf{W}_{*k}||_2 \leq \max_{1\leq k\leq m} ||\mathbf{W}_{*k}||_2 \leq \sqrt{(1+c_0)n}\Big) \geq 1 - 2m\exp\left(-\frac{nc_0^2}{8}\right). \tag{55}$$

Now conditioning on the event $\sqrt{(1-c_0)n} \leq \min_{1\leq k\leq m} ||\mathbf{W}_{*k}||_2$, we have

$$\mathcal{R}^*(\mathbf{G}^0) = ||\mathbf{G}^0||_2 = \max_{||\boldsymbol{v}||_2\leq 1} \sqrt{\sum_{k=1}^m \frac{(\boldsymbol{v}^T\mathbf{X}^T\mathbf{W}_{*k})^2}{||\mathbf{W}_{*k}||_2^2}} \leq \max_{||\boldsymbol{v}||_2\leq 1} \sqrt{\frac{\sum_{k=1}^m (\boldsymbol{v}^T\mathbf{X}^T\mathbf{W}_{*k})^2}{(1-c_0)n}} = \frac{||\mathbf{X}^T\mathbf{W}||_2}{\sqrt{(1-c_0)n}}. \tag{56}$$

By Lemma 11, there exists some universal positive constant c_1 such that we have

$$\mathbb{P}\left(\frac{||\mathbf{X}^T\mathbf{W}||_2}{\sqrt{(1-c_0)n}} \leq \frac{2||\mathbf{X}||_2(\sqrt{d}+\sqrt{m})}{\sqrt{n(1-c_0)}}\right) \geq 1 - 2\exp\left(-c_1(d+m)\right). \tag{57}$$

Given any matrix $\mathbf{A}$ in $\mathcal{N}_\perp$, $\mathbf{A}$ has at most rank $2r$ (See more details in Appendix B of Negahban and Wainwright (2011)). Then we have

$$||\mathbf{A}||_* = \sum_{j=1}^{2r} \psi_j(\mathbf{A}) \leq \sqrt{2r}\sqrt{\sum_{j=1}^{2r} \psi_j(\mathbf{A})^2} = \sqrt{2r}||\mathbf{A}||_{\mathrm{F}}.$$

Therefore we have $\Theta(\mathcal{N}_\perp, ||\cdot||_*) = \sqrt{2r}$. Theorem 3 requires

$$2\lambda\Theta(\mathcal{N}_\perp, \mathcal{R}) \leq \delta\kappa(c-1)\sqrt{n} \text{ for some } \delta < 1. \tag{58}$$

Thus if we take

$$\lambda = \frac{2c||\mathbf{X}||_2(\sqrt{m}+\sqrt{d})}{\sqrt{n(1-c_0)}},$$

then we need n to be large enough

$$n \geq \frac{4\sqrt{2}c||\mathbf{X}||_2(\sqrt{rm}+\sqrt{rd})}{\delta\kappa(c-1)\sqrt{1-c_0}},$$

such that (58) can be secured. Then by combining (55), (56), (57), (47), and (53), we complete the proof. ■

Appendix D. Proof of Corollary 5

We need to introduce the following lemma for our proof.

Lemma 12 *Suppose that we have all entries of* $\mathbf{W}$ *independently generated from the standard Gaussian distribution with mean* 0 *and variance* 1*, then we have*

$$\mathbb{P}\left(\max_{1\leq j\leq d}\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q \leq 2\left(m^{1-1/p}+\sqrt{\log d}\right)\right) \geq 1-\frac{2}{d},$$

where $1/p + 1/q = 1$.

The proof of Lemma 12 is provided in Appendix F. Now we proceed to derive the refined error bound for the joint sparsity setting.

Proof Recall that we already have (55),

$$\mathbb{P}\Big(\sqrt{(1-c_0)n} \leq \min_{1\leq k\leq m}||\mathbf{W}_{*k}||_2 \leq \max_{1\leq k\leq m}||\mathbf{W}_{*k}||_2 \leq \sqrt{(1+c_0)n}\Big) \geq 1-2m\exp\left(-\frac{nc_0^2}{8}\right). \tag{59}$$

Now conditioning on the event $\sqrt{(1-c_0)n} \leq \min_{1\leq k\leq m}||\mathbf{W}_{*k}||_2$, we have

$$\mathcal{R}^*(\mathbf{G}^0) = ||\mathbf{G}^0||_{\infty,q} = \max_{1\leq j\leq d}\left(\sum_{k=1}^n \frac{(\mathbf{W}_{*k}^T\mathbf{X}_{*j})^q}{||\mathbf{W}_{*k}||_2^q}\right)^{1/q} \leq \frac{\max_{1\leq j\leq d}||\mathbf{W}^T\mathbf{X}_{*j}||_q}{\min_{1\leq k\leq m}||\mathbf{W}_{*k}||_2} \leq \frac{||\mathbf{X}^T\mathbf{W}||_{\infty,q}}{\sqrt{(1-c_0)n}}. \tag{60}$$

By Lemma 12, we have

$$\mathbb{P}\left(\frac{||\mathbf{X}^T\mathbf{W}||_{\infty,q}}{\sqrt{(1-c_0)n}} \leq \frac{2m^{1-1/p}}{\sqrt{(1-c_0)}} + \frac{2\sqrt{\log d}}{\sqrt{(1-c_0)}}\right) \geq 1-\frac{2}{d}. \tag{61}$$

Given any matrix $\mathbf{A}$ in $\mathcal{N}_\perp$, $\mathbf{A}$ has at most s nonzero rows. Then we have

$$||\mathbf{A}||_{1,p} = \sum_{\mathbf{A}_{j*}\neq\mathbf{0}}||\mathbf{A}_{j*}||_p \leq \sum_{\mathbf{A}_{j*}\neq\mathbf{0}}||\mathbf{A}_{j*}||_2 \leq \sqrt{s}\sqrt{\sum_{\mathbf{A}_{j*}\neq\mathbf{0}}||\mathbf{A}_{j*}||_2^2} = \sqrt{s}||\mathbf{A}||_{\mathrm{F}}.$$

Therefore we have $\Theta(\mathcal{N}_\perp, ||\cdot||_{1,p}) = \sqrt{s}$ for any $2 \leq p \leq \infty$. Theorem 3 requires

$$2\lambda\Theta(\mathcal{N}_\perp, \mathcal{R}) \leq \delta\kappa(c-1)\sqrt{n} \text{ for some } \delta < 1. \tag{62}$$

Thus if we take

$$\lambda = \frac{2c(m^{1-1/p} + \sqrt{\log d})}{\sqrt{1-c_0}},$$

then we need n to be large enough

$$\sqrt{n} \geq \frac{4c\sqrt{s}(m^{1-1/p} + \sqrt{\log d})}{\delta\kappa(c-1)\sqrt{1-c_0}},$$

such that (62) can be secured. Then by combining (59), (60), (61), (47), and (53), we complete the proof.

■

Appendix E. Proof of Lemma 11

Proof Since $\mathbf{W}$ has all its entries independently generated from the standard Gaussian distribution with mean 0 and variance 1, then all $\mathbf{X}^T\mathbf{W}_{*k}/\sqrt{n}$'s are essentially independently generated from a multivariate Gaussian distribution with mean $\mathbf{0}$ and covariance matrix $\mathbf{X}^T\mathbf{X}/n$.

Thus by Corollary 5.50 in Vershynin (2010) on the singular values of Gaussian random matrices (Davidson and Szarek, 2001), we know that there exists a universal positive constant c_1 such that

$$\mathbb{P}\left(\frac{||\mathbf{X}^T\mathbf{W}||_2}{\sqrt{n}} \leq \frac{2||\mathbf{X}||_2}{\sqrt{n}}(\sqrt{m} + \sqrt{d})\right) \geq 1 - 2\exp(-c_1(d+m)), \tag{63}$$

which completes the proof. ■

Appendix F. Proof of Lemma 12

Proof We adopt the similar proof strategy in Negahban et al. (2012), and begin our proof by establishing the tail bound of $||\mathbf{W}^T\mathbf{X}_{*j}||_q/\sqrt{n}$.

Deviation above the mean: Given any pair of $\mathbf{W}$, $\widetilde{\mathbf{W}} \in \mathbb{R}^{n\times m}$ and $1/q + 1/p = 1$, we have

$$\begin{aligned}\left|\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q - \frac{1}{\sqrt{n}}||\widetilde{\mathbf{W}}^T\mathbf{X}_{*j}||_q\right| &\leq \frac{1}{\sqrt{n}}||(\mathbf{W}-\widetilde{\mathbf{W}})^T\mathbf{X}_{*j}||_q \\ &= \frac{1}{\sqrt{n}}\max_{||\boldsymbol{\theta}||_p\leq 1}\langle\boldsymbol{\theta}, (\mathbf{W}-\widetilde{\mathbf{W}})^T\mathbf{X}_{*j}\rangle. \end{aligned}\tag{64}$$

By the Cauchy-Schwartz inequality, we have

$$\frac{1}{\sqrt{n}}\max_{||\boldsymbol{\theta}||_p\leq 1}\langle\boldsymbol{\theta}\mathbf{X}_{*j}^T, \mathbf{W}-\widetilde{\mathbf{W}}\rangle \leq \frac{||\mathbf{W}-\widetilde{\mathbf{W}}||_{\mathrm{F}}}{\sqrt{n}}\max_{||\boldsymbol{\theta}||_p\leq 1}||\boldsymbol{\theta}\mathbf{X}_{*j}^T||_{\mathrm{F}}. \tag{65}$$

Since $\boldsymbol{\theta}\mathbf{X}_{*j}^T$ is a rank one matrix, its singular value decomposition is

$$\boldsymbol{\theta}\mathbf{X}_{*j}^T = ||\boldsymbol{\theta}||_2||\mathbf{X}_{*j}||\cdot\frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||_2}\cdot\frac{\mathbf{X}_{*j}^T}{||\mathbf{X}_{*j}||_2}.$$

Consequently, we have

$$\frac{1}{\sqrt{n}}\max_{||\boldsymbol{\theta}||_p\leq 1}||\boldsymbol{\theta}\mathbf{X}_{*j}^T||_{\mathrm{F}}=\frac{||\mathbf{X}_{*j}||_2}{\sqrt{n}}\max_{||\boldsymbol{\theta}||_p\leq 1}||\boldsymbol{\theta}||_2\overset{\text{(i)}}{\leq}\frac{m^{1/2-1/p}||\mathbf{X}_{*j}||_2}{\sqrt{n}}\overset{\text{(ii)}}{\leq}1. \tag{66}$$

where (i) comes from $||\boldsymbol{\theta}||_2\leq m^{1/2-1/p}||\boldsymbol{\theta}||_p$, and (ii) comes from the column normalization condition (21). Combining (64), (65), and (66), we obtain

$$\left|\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q-\frac{1}{\sqrt{n}}||\widetilde{\mathbf{W}}^T\mathbf{X}_{*j}||_q\right|\leq||\mathbf{W}-\widetilde{\mathbf{W}}||_{\mathrm{F}}. \tag{67}$$

which implies that $||\mathbf{W}^T\mathbf{X}_{*j}||_q/\sqrt{n}$ is a Lipschitz continuous function of $\mathbf{W}$ with a Lipschitz constant as 1. By the Gaussian concentration of measure for Lipschitz functions (Ledoux and Talagrand, 2011), we have

$$\mathbb{P}\left(\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q\geq\mathbb{E}\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q+\xi\right)\leq 2\exp\left(-\frac{\xi^2}{2}\right). \tag{68}$$

Upper bound of the mean: Given any $\boldsymbol{\beta}\in\mathbb{R}^m$, we define a zero mean Gaussian random variable $J_{\boldsymbol{\beta}}=\boldsymbol{\beta}^T\mathbf{W}^T\mathbf{X}_{*j}/\sqrt{n}$, and note that we have $\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q=\max_{||\boldsymbol{\beta}||_p=1}J_{\boldsymbol{\beta}}$. Thus given any two vectors $||\boldsymbol{\beta}||_p\leq 1$ and $||\boldsymbol{\beta}'||_p\leq 1$, we have

$$\mathbb{E}(J_{\boldsymbol{\beta}}-J_{\boldsymbol{\beta}'})^2=\frac{1}{n}||\mathbf{X}_{*j}||_2^2||\boldsymbol{\beta}-\boldsymbol{\beta}'||_2^2\leq||\boldsymbol{\beta}-\boldsymbol{\beta}'||_2^2,$$

where the last inequality comes from (21) and $m^{1-1/p}\geq 1$.

Then we define another Gaussian random variable $K_{\boldsymbol{\beta}}=\boldsymbol{\beta}^T\boldsymbol{\omega}$, where $\boldsymbol{\omega}=(\omega_1,...,\omega_m)^T\sim N(\mathbf{0},\mathbf{I}_m)$ is standard Gaussian. By construction, for any pair $\boldsymbol{\beta},\boldsymbol{\beta}'\in\mathbb{R}^m$, we have

$$\mathbb{E}[(K_{\boldsymbol{\beta}}-K_{\boldsymbol{\beta}'})^2]=\|\boldsymbol{\beta}-\boldsymbol{\beta}'\|_2^2\geq\mathbb{E}(J_{\boldsymbol{\beta}}-J_{\boldsymbol{\beta}'})^2.$$

Thus by the Sudakov-Fernique comparison principle (Ledoux and Talagrand, 2011), we have

$$\mathbb{E}\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q=\mathbb{E}\max_{||\boldsymbol{\beta}||_p=1}J_{\boldsymbol{\beta}}\leq\mathbb{E}\max_{||\boldsymbol{\beta}||_p=1}K_{\boldsymbol{\beta}}.$$

By definition of $K_{\boldsymbol{\beta}}$, we have

$$\mathbb{E}\max_{||\boldsymbol{\beta}||_p=1}K_{\boldsymbol{\beta}}=\mathbb{E}||\boldsymbol{\omega}||_q\leq m^{1/q}(\mathbb{E}|\boldsymbol{\omega}_1|^q)^{1/q}, \tag{69}$$

where the last inequality comes from Jensen's inequality and the fact that $|\boldsymbol{\omega}_1|^{1/q}$ is a concave function of $\boldsymbol{\omega}_1$ for $q\in[1,2]$. Eventually, by Hölder inequality, we obtain

$$(\mathbb{E}|\boldsymbol{\omega}_1|^q)^{1/q}\leq\sqrt{\mathbb{E}\omega_1^2}=1. \tag{70}$$

Combing (69) and (70), we obtain

$$\mathbb{E}\max_{||\boldsymbol{\beta}||_p=1}K_{\boldsymbol{\beta}}\leq m^{1-1/p}\leq 2m^{1-1/p}. \tag{71}$$

Then combing (68) and (71), we have

$$\mathbb{P}\left(\frac{1}{\sqrt{n}}||\mathbf{W}^T\mathbf{X}_{*j}||_q\geq 2m^{1-1/p}+\xi\right)\leq 2\exp\left(-\frac{\xi^2}{2}\right).$$

Taking the union bound over $j=1,...,d$ and let $\xi=2\sqrt{\log d}$, we have

$$\mathbb{P}\left(\frac{1}{\sqrt{n}}||\mathbf{X}^T\mathbf{W}||_{\infty,q}\geq 2m^{1-1/p}+2\sqrt{\log d}\right)\leq\frac{2}{d}.$$

This finishes the proof. ■

References

Agarwal, A., Negahban, S. and Wainwright, M. (2012). Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics* **40** 1171–1197.

Amit, Y., Fink, M., Srebro, N. and Ullman, S. (2007). Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine Learning*. ACM.

Anderson, T. (1958). *An introduction to multivariate statistical analysis*. Wiley New York.

Anderson, T. (1999). Asymptotic distribution of the reduced rank regression estimator under general conditions. *The Annals of Statistics* **27** 1141–1154.

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research* **6** 1817–1853.

Argyriou, A., Evgeniou, T. and Pontil, M. (2008). Convex multi-task feature learning. *Machine Learning* **73** 243–272.

Argyriou, A., Micchelli, C. A. and Pontil, M. (2010). On spectral learning. *The Journal of Machine Learning Research* **11** 935–953.

Baxter, J. (2000). A model of inductive bias learning. *Journal of Artificial Intelligence Research* **12** 149–198.

Beck, A. and Teboulle, M. (2009a). Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. *IEEE Transactions on Image Processing* **18** 2419–2434.

Beck, A. and Teboulle, M. (2009b). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences* **2** 183–202.

Belloni, A., Chernozhukov, V. and Wang, L. (2011). Square-root lasso: pivotal recovery of sparse signals via conic programming. *Biometrika* **98** 791–806.

Bickel, P. J., Ritov, Y. and Tsybakov, A. B. (2009). Simultaneous analysis of lasso and dantzig selector. *The Annals of Statistics* **37** 1705–1732.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning* **3** 1–122.

Breiman, L. and Friedman, J. (2002). Predicting multivariate responses in multiple linear regression. *Journal of the Royal Statistical Society: Series B* **59** 3–54.

Bunea, F. and Barbu, A. (2009). Dimension reduction and variable selection in case control studies via regularized likelihood optimization. *Electronic Journal of Statistics* **3** 1257–1287.

Bunea, F., Lederer, J. and She, Y. (2013). The group square-root lasso: Theoretical properties and fast algorithms. *IEEE Transactions on Information Theory* **60** 1313 – 1325.

Bunea, F., She, Y. and Wegkamp, M. (2012). Joint variable and rank selection for parsimonious estimation of high dimensional matrices. *The Annals of Statistics* **40** 2359–2388.

Bunea, F., She, Y. and Wegkamp, M. H. (2011). Optimal selection of reduced rank estimators of high-dimensional matrices. *The Annals of Statistics* **39** 1282–1309.

Caruana, R. (1997). Multitask learning. *Machine Learning* **28** 41–75.

Caruana, R., Baluja, S., Mitchell, T. et al. (1996). Using the future to "sort out" the present: Rankprop and multitask learning for medical risk evaluation. In *Advances in Neural Information Processing Systems*.

Chen, J., Zhou, J. and Ye, J. (2011). Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Chen, S., Donoho, D. and Saunders, M. (1998). Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing* **20** 33–61.

Chen, X., Lin, Q., Kim, S., Carbonell, J. G. and Xing, E. P. (2012). Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics* **6** 719–752.

Davidson, K. R. and Szarek, S. J. (2001). Local operator theory, random matrices and Banach spaces. *Handbook of the geometry of Banach spaces* **1** 317–366.

Evgeniou, T., Micchelli, C. A. and Pontil, M. (2006). Learning multiple tasks with kernel methods. *Journal of Machine Learning Research* **6** 615.

Foygel, R. and Srebro, N. (2011). Concentration-based guarantees for low-rank matrix reconstruction. In *24th Annual Conference on Learning Theory*, vol. 19.

Geer, S. (2014). Weakly decomposable regularization penalties and structured sparsity. *Scandinavian Journal of Statistics* **41** 72–86.

Giraud, C. (2011). Low rank multivariate regression. *Electronic Journal of Statistics* **5** 775–799.

Gong, P., Ye, J. and Zhang, C. (2012). Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

Gong, P., Zhou, J., Fan, W. and Ye, J. (2014). Efficient multi-task feature learning with calibration. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM.

Heskes, T. (2000). Empirical Bayes for learning to learn. In *Proceedings of the 17th International Conference on Machine Learning*.

Izenman, A. (1975). Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis* **5** 248–264.

Izenman, A. J. (2008). *Modern multivariate statistical techniques: regression, classification, and manifold learning*. Springer.

Jalali, A., Ravikumar, P., Sanghavi, S. and Ruan, C. (2010). A dirty model for multi-task learning. In *Advances in Neural Information Processing Systems*.

Johnson, R. and Zhang, T. (2008). Graph-based semi-supervised learning and spectral kernel design. *IEEE Transactions on Information Theory* **54** 275–288.

Johnstone, I. M. (2001). Chi-square oracle inequalities. *Lecture Notes-Monograph Series* 399–418.

Klopp, O. (2011). High dimensional matrix estimation with unknown variance of the noise. Tech. rep., Université Paris Ouest Nanterre La Défense.
URL `http://arxiv.org/abs/1112.3055`

Kolar, M., Lafferty, J. and Wasserman, L. (2011). Union support recovery in multi-task learning. *Journal of Machine Learning Research* **12** 2415–2435.

Ledoux, M. and Talagrand, M. (2011). *Probability in Banach Spaces: isoperimetry and processes*. Springer.

Lee, J., Sun, Y. and Taylor, J. E. (2013). On model selection consistency of penalized m-estimators: a geometric theory. In *Advances in Neural Information Processing Systems*.

Liu, H., Palatucci, M. and Zhang, J. (2009a). Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery. In *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM.
URL `http://arxiv.org/abs/1209.2437`

Liu, H., Wang, L. and Zhao, T. (2014a). Multivariate regression with calibration. In *Advances in Neural Information Processing Systems*.

Liu, H., Wang, L. and Zhao, T. (2014b). Sparse covariance matrix estimation with eigenvalue constraints. *Journal of Computational and Graphical Statistics* **23** 439–459.

Liu, J., Ji, S. and Ye, J. (2009b). Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Liu, J. and Ye, J. (2010). Efficient ℓ_1/ℓ_q norm regularization. Tech. rep., Arizona State University.
URL `http://arxiv.org/abs/1009.4766`

Lounici, K., Pontil, M., Van De Geer, S. and Tsybakov, A. (2011). Oracle inequalities and optimal inference under group sparsity. *The Annals of Statistics* **39** 2164–2204.

Meier, L., Van De Geer, S. and Bühlmann, P. (2008). The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B* **70** 53–71.

Meinshausen, N. and Bühlmann, P. (2010). Stability selection. *Journal of the Royal Statistical Society: Series B* **72** 417–473.

Mitchell, T., Shinkareva, S., Carlson, A., Chang, K., Malave, V., Mason, R. and Just, M. (2008). Predicting human brain activity associated with the meanings of nouns. *Science* **320** 1191–1195.

Mukherjee, A., Wang, N. and Zhu, J. (2012). Degrees of freedom of the reduced rank regression. Tech. rep., University of Michigan Ann Arbor.
URL `http://arxiv.org/abs/1210.2464`

Negahban, S. and Wainwright, M. J. (2011). Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics* **39** 1069–1097.

Negahban, S. N., Ravikumar, P., Wainwright, M. J. and Yu, B. (2012). A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science* **27** 538–557.

Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming* **103** 127–152.

Obozinski, G., Taskar, B. and Jordan, M. (2010). Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing* **20** 231–252.

Raskutti, G., Wainwright, M. J. and Yu, B. (2010). Restricted eigenvalue properties for correlated gaussian designs. *The Journal of Machine Learning Research* **11** 2241–2259.

Reinsel, G. (2003). *Elements of multivariate time series analysis*. Springer Verlag.

Reinsel, G. and Velu, R. (1998). *Multivariate reduced-rank regression: theory and applications*. Springer New York.

Rohde, A. and Tsybakov, A. B. (2011). Estimation of high-dimensional low-rank matrices. *The Annals of Statistics* **39** 887–930.

Rothman, A., Levina, E. and Zhu, J. (2010). Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics* **19** 947–962.

Salakhutdinov, R. and Srebro, N. (2010). Collaborative filtering in a non-uniform world: Learning with the weighted trace norm. In *Advances in Neural Information Processing Systems*.

Sun, T. and Zhang, C. (2012). Scaled sparse linear regression. *Biometrika* **101** 1–20.

Teh, Y. W., Seeger, M. and Jordan, M. I. (2005). Semiparametric latent factor models. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, vol. 10.

Thrun, S. (1996). Is learning the n-th thing any easier than learning the first? In *Advances in Neural Information Processing Systems*.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* **58** 267–288.

Toh, K.-C. and Yun, S. (2010). An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization* **6** 15.

Turlach, B., Venables, W. and Wright, S. (2005). Simultaneous variable selection. *Technometrics* **47** 349–363.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *Compressed Sensing, Theory and Applications* 210–268.

Yu, K., Tresp, V. and Schwaighofer, A. (2005). Learning Gaussian processes from multiple tasks. In *Proceedings of the 22nd international conference on Machine Learning*.

Yuan, M., Ekici, A., Lu, Z. and Monteiro, R. (2007). Dimension reduction and coefficient estimation in multivariate linear regression. *Journal of the Royal Statistical Society: Series B* **69** 329–346.

Yuan, M. and Lin, Y. (2005). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B* **68** 49–67.

Zhang, J. (2006). *A probabilistic framework for multi-task learning*. Ph.D. thesis, Carnegie Mellon University, Language Technologies Institute, School of Computer Science.

Zhang, J., Ghahramani, Z. and Yang, Y. (2006). Learning multiple related tasks using latent independent component analysis. In *Advances in Neural Information Processing Systems*.

Zhao, T. and Liu, H. (2012). Sparse additive machine. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*.

Zhao, T. and Liu, H. (2014). Calibrated precision matrix estimation for high-dimensional elliptical distributions. *IEEE Transactions on Information Theory* **60** 7874–7887.

Zhao, T., Liu, H. and Zhang, T. (2014a). A general theory of pathwise coordinate optimization. *arXiv preprint arXiv:1412.7477* .

Zhao, T., Roeder, K. and Liu, H. (2014b). Positive semidefinite rank-based correlation matrix estimation with application to semiparametric graph estimation. *Journal of Computational and Graphical Statistics* **23** 895–922.

Zhao, T., Yu, M., Wang, Y., Arora, R. and Liu, H. (2014c). Accelerated mini-batch randomized block coordinate descent method. In *Advances in Neural Information Processing Systems*.

Zhou, J., Chen, J. and Ye, J. (2012). MALSAR: Multi-task learning via structural regularization. Tech. rep., Arizona State University.
URL `http://www.public.asu.edu/~jye02/Software/MALSAR`

Journal of Machine Learning Research 16 (2015) 1607-1627 Submitted 7/14; Published 8/15

Bayesian Nonparametric Crowdsourcing

Pablo G. Moreno PGMORENO@TSC.UC3M.ES
Antonio Artés-Rodríguez ANTONIO@TSC.UC3M.ES
Gregorio Marañón Health Research Institute
Department of Signal Theory and Communications
Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Leganés (Madrid, Spain)

Yee Whye Teh Y.W.TEH@STATS.OX.AC.UK
Department of Statistics
1 South Parks Road
Oxford OX1 3TG, UK

Fernando Perez-Cruz FERNANDO@TSC.UC3M.ES
Gregorio Marañón Health Research Institute FERNANDO.PEREZ-CRUZ@ALCATEL-LUCENT.COM
Department of Signal Theory and Communications
Universidad Carlos III de Madrid
Avda. de la Universidad, 30
28911 Leganés (Madrid, Spain)
Bell Labs (Alcatel-Lucent)
600 Mountain Avenue
New Providence, NJ 07974

Editor: David Blei

Abstract

Crowdsourcing has been proven to be an effective and efficient tool to annotate large data-sets. User annotations are often noisy, so methods to combine the annotations to produce reliable estimates of the ground truth are necessary. We claim that considering the existence of clusters of users in this combination step can improve the performance. This is especially important in early stages of crowdsourcing implementations, where the number of annotations is low. At this stage there is not enough information to accurately estimate the bias introduced by each annotator separately, so we have to resort to models that consider the statistical links among them. In addition, finding these clusters is interesting in itself as knowing the behavior of the pool of annotators allows implementing efficient active learning strategies. Based on this, we propose in this paper two new fully unsupervised models based on a Chinese restaurant process (CRP) prior and a hierarchical structure that allows inferring these groups jointly with the ground truth and the properties of the users. Efficient inference algorithms based on Gibbs sampling with auxiliary variables are proposed. Finally, we perform experiments, both on synthetic and real databases, to show the advantages of our models over state-of-the-art algorithms.

Keywords: multiple annotators, Bayesian nonparametrics, Dirichlet process, hierarchical clustering, Gibbs sampling

1. Introduction

Crowdsourcing services are becoming very popular as a mean of outsourcing tasks to a large crowd of users. The best-known tool is Mechanical Turk (Amazon, 2005), in which *requesters* are able to post small tasks for *providers* registered in the system, who complete them for a monetary payment set by the requester. In machine learning, crowdsourcing allows to distribute the labeling of a data-set among a pool of users, so each user only labels a subset of the instances. The advantage is the ability to gather large data-sets in a short time and, generally, at a low cost. Successful examples include, but it is not limited to, LabelMe (Russell et al., 2008) or GalazyZoo (Lintott et al., 2010).

The quality of the labels retrieved by crowdsourcing is uneven. Unlike traditional ways of gathering a labeled data-set in which labels are provided by a small set of motivated experts, we deal now with a large number of users who are not necessarily experts nor motivated. Further, we might have little or no information about them to perform quality control tests. This motivates the development of statistical models for reliably estimating ground truth from noisy and biased labels provided by users.

Another problem that has received significant attention of late, is the detection of groupings among the labelers (Simpson et al., 2011, 2013). In most crowdsourcing applications we can identify several types of users: experts, novices, spammers and even malicious or adversarial annotators. Identifying these groups of users and learning about their properties is useful to design efficient crowdsourcing strategies that minimize the overall cost, selecting the most suitable users for a labeling task. For example, if we could identify spammers we could ban them from the system and avoid wasting resources. In the same way, if a user is identified as an expert in a particular task, we could reward him by increasing his pay-off or giving him preference over other users when the time to select new tasks comes.

Usually, the detection of grouping of labelers is tackled in a post-processing step, after the ground truth has been estimated from user annotations (see Section 4). In particular Simpson et al. (2011) were the first to tackle the join problem. They estimated the ground truth using a previous model called Independent Bayesian Combination of Classifiers (iBCC) (Kim and Ghahramani, 2012) and then, as a post-processing step they infer the different clusters of users. Therefore, the estimation of the ground truth is done without considering the clustering structure of the users.

In this paper, we propose two unsupervised Bayesian nonparametric models to combine the labels provided by the users in a crowdsourcing scenario, taking into account the presence of clusters of users. Our models jointly solve the problem of the estimation of ground truth and the problem of identification of clusters of users and their properties. The estimation of the ground truth improves the clustering of the users and vice-versa, thus performing better than current state-of-the-art (Kim and Ghahramani, 2012; Simpson et al., 2011). The overall improvement in both tasks is particularly important in the early stages of a crowdsourcing project, when the number of annotations provided by the users is very low. In this case, algorithms that estimate the properties of each user independently, without considering the dependencies among them, tend to provide poor estimates, and may perform worse than majority voting (see Section 5).

In the first model, we propose a clustering structure using a CRP prior (Pitman, 2002) which allows flexible modeling of the number of clusters of users. In this model, all the users that belong to the same cluster share the same parameters governing the way they label instances, and therefore, they have the same behavior. Forcing all the users to share the same exact parameters, is a strong assumption that might lead to groupings with a large number of cluster. Therefore, these groupings

are difficult to interpret and not very useful. To relax this assumption we propose a second model in which users that belong to the same cluster are modeled as having similar parameters, but allows each user to have its own parameters using a hierarchical Bayesian approach.

In this paper, we rely on a Bayesian nonparametric model, because we are not only interesting in having an accurate model but also in having an interpretable one. In the experiments (Section 5) we show that the error rates between the two proposed models are not significantly different. However, the second one is interpretable, in the sense that it perfectly identifies each kind of clusters and it reports the least number of them. The interpretability of the model is principal to us, because we want to use the model to identify the 'good' annotators and be able to reward them accordingly, while other models are not able to provide this information.

The rest of the paper is organized as follows. In Section 2, we present the two new generative models for crowdsourcing that take into account the clustering structure of the users. In Section 3, we propose efficient Markov chain Monte Carlo (MCMC) inference algorithms for estimating the different groups of users as well as the ground truth. In Section 4, we review related literature on crowdsourcing and the identification of user clusters in the context of crowdsourcing. In Section 5, we validate our model on synthetic data and we perform several experiments on real data-sets to show the advantages of our models over state-of-the-art algorithms. Finally, we conclude this paper in Section 6 and present possible extensions for the future.

2. Hierarchical Bayesian Combination of Classifiers

In this section, we propose two different models. Both algorithms receive as input a set of noisy labels $\boldsymbol{Y} \in \{0,...,C\}^{N\times L}$ provided by L users for N instances. The element $y_{i\ell}$ represents the label given by the user ℓ to the instance i and it is 0 if the user did not label the corresponding instance. Notice that this matrix $\boldsymbol{Y}$ is highly sparse in the early stages of a crowdsourcing application. This is known as the *cold start problem* (Schein et al., 2002), i.e. the difficulty of drawing any inferences due to the lack of information. Notice that $\boldsymbol{Y}$ is the only observed variable in the models.

The output of the algorithms is the set of true but unknown labels of the instances $\boldsymbol{z} \in \{1,...,C\}^N$, where z_i indicates the true label estimate of the instance i.

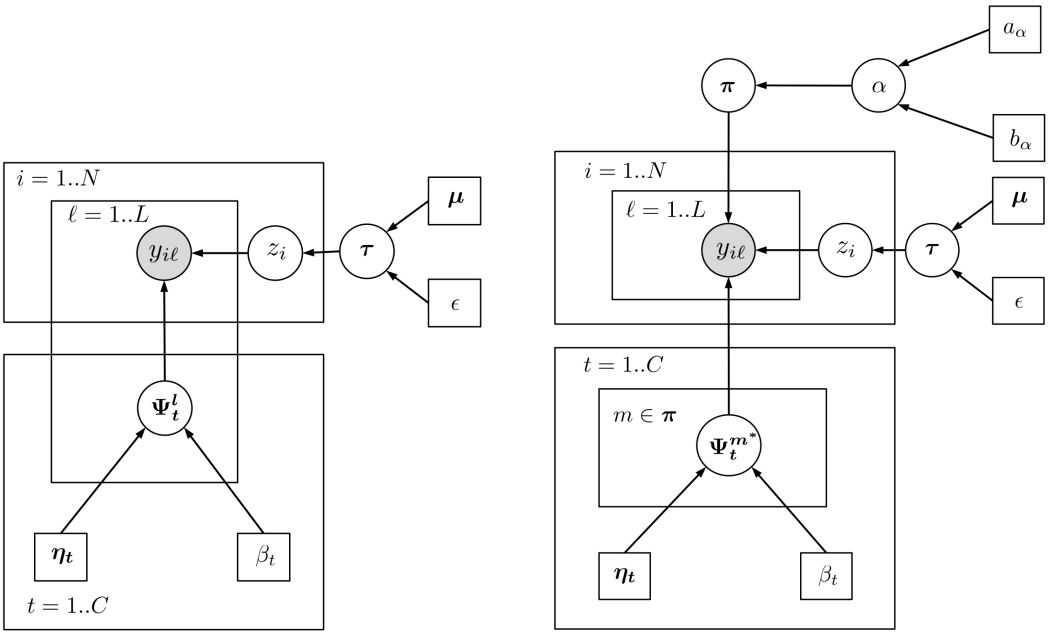
We denote by $[L] = \{1,2,...,L\}$ the set of indices of the users and by $\boldsymbol{\pi}_L$ a partition of $[L]$. A partition is a collection of mutually exclusive, mutually exhaustive and non-empty subsets called clusters. We denote the cluster assignment of the user ℓ with a variable q_ℓ such that $q_\ell = m$ denotes the event that the user ℓ is assigned to cluster $m \in \boldsymbol{\pi}_L$.

2.1 Clustering Based Bayesian Combination of Classifiers

Firstly, we propose a model for users in which they can belong to different clusters. In each cluster all the users have the same properties. We name it Clustering based Bayesian Combination of Classifiers (cBCC) (see Figure 1b) and it has the following observation model

$$y_{i\ell}|z_i, \boldsymbol{\pi}, \boldsymbol{\Psi} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\Psi}^{q_\ell}_{z_i}),$$
$$z_i|\boldsymbol{\tau} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\tau}).$$

We assume that all the users that belong to cluster $m \in \boldsymbol{\pi}$ share the same properties, i.e. the same confusion matrix $\boldsymbol{\Psi}^m \in [0,1]^{C\times C}$, where Ψ^m_{tc} is the probability that a user allocated in cluster m labels an instance as $y = c$ when the ground truth is $z = t$. We use the notation $\boldsymbol{\Psi}^m_t \in \mathscr{S}^C$ to denote



(a) iBCC model (b) cBCC model

Figure 1: Graphical model representation of the iBCC and cBCC models

the row t of $\boldsymbol{\Psi}^m$, where $\mathscr{S}^C$ is the C-dimensional probability simplex. The component τ_t of $\boldsymbol{\tau} \in \mathscr{S}^C$ is the probability of the ground truth z being equal to $t \in \{1,...,C\}$.

We also need to define priors to complete the Bayesian model. In particular, we choose conjugate priors

$$\boldsymbol{\Psi}_t^m | \boldsymbol{\beta}, \boldsymbol{\eta} \sim \text{Dir}(\beta_t \boldsymbol{\eta_t}),$$
$$\boldsymbol{\tau} | \varepsilon, \boldsymbol{\mu} \sim \text{Dir}(\varepsilon \boldsymbol{\mu}),$$

where we use a Dirichlet prior on each of the rows of the confusion matrices in which $\boldsymbol{\eta}_t \in \mathscr{S}^C$ is the mean value of $\boldsymbol{\Psi}_t^m$ while $\beta_t \in \mathbb{R}_+$ is related to its precision. Notice that this is an over parametrization of the Dirichlet distribution, which only needs C parameters to be fully determined. However, this decomposition is useful to interpret the results as well as for the development of the inference algorithms in Section 3. Likewise, we set a Dirichlet prior on $\boldsymbol{\tau}$, where $\boldsymbol{\mu} \in \mathscr{S}^C$ is the mean and $\varepsilon \in \mathbb{R}_+$ relates to the precision.

We could use a parametric model in which the cardinality of the partition $M = |\boldsymbol{\pi}|$ is fixed a priori. Unfortunately, in this case the inferences are sensitive to the value of M chosen. In the limiting case $M = 1$ the model is equivalent to majority voting. If M is too large the model does not take advantage of the presence of clusters of users. In the limiting case $M = L$ each user becomes a singleton cluster, and the model does not capture the dependencies among the users.

To find M we could use traditional model selection strategies like cross-validation (Stone, 1974) or Bayesian Information Criterion (Fraly and Raftery, 1998). This approach has two limitations. First, we usually do not have access to a validation set for which $\boldsymbol{z}$ is known. Second, is the high computational complexity. An alternative pathway is to set a prior on the space of partitions.

We denote by $\mathscr{P}_L$ the space of all partitions $\pi_L \in \mathscr{P}_L$. In a Bayesian setting, we have to set the prior without observing the number of users. One option is to set a prior on an infinite number of users, i.e. on partition $\pi \in \mathscr{P}_\infty$. To build such a prior we further assume that the observations are exchangeable and that we deal with consistent random partitions (Pitman, 2002). A (exchangeable and consistent) prior on $\mathscr{P}_\infty$ is the CRP introduced by Blackwell and Macqueen (1973) and that can be seen as the induced distribution over the partition space by a Dirichlet Process (DP) (Ferguson, 1973; Antoniak, 1974). We place a CRP prior over the users' partitions

$$\pi|\alpha \sim \text{CRP}(\alpha). \tag{1}$$

We can generate samples from this prior using the following conditional distributions

$$p(q_\ell = m|\pi^{\neg\ell}, \alpha) \propto \begin{cases} |m|^{\neg\ell}, & m \in \pi^{\neg\ell} \\ \alpha, & m = \emptyset \end{cases},$$

where $|m|$ represents the number of users in cluster m and $|m|^{\neg\ell}$ is equal to $|m|$ excluding user ℓ. We denote by $\pi^{\neg\ell}$ the partition with the user ℓ removed and $q_\ell = \emptyset$ denotes the event that user ℓ is assigned to a new cluster. α is the so called concentration parameter and control the a priori probability of generating new clusters. We further place a gamma prior over the concentration parameter α

$$\alpha|a_\alpha, b_\alpha \sim \text{Gamma}(a_\alpha, b_\alpha). \tag{2}$$

If α tends to infinity, every user is allocated to a singleton cluster. If α tends to 0, all the users share the same confusion matrix and the model produces a majority voting solution

In general, the CRP assigns more mass to partitions with a small number of clusters. This is sensible in our case, because when the number of annotations is scarce, majority voting may perform better than more elaborate algorithms since there is no enough information to estimate the individual properties of the users. In this case, the CRP prior dominates and therefore all users are allocated to the same cluster. When the number of annotations increase, the likelihood term dominates and different clusters of users are created.

Finally, we analyze the correlation structure that it is introduced among the users as a consequence of this clustering. The correlation a priori among two users ℓ and ℓ' is

$$\text{Corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)|z_i = t) = \begin{cases} -\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\sqrt{\frac{\eta_{ta}\eta_{tb}}{(1-\eta_{ta})(1-\eta_{tb})}} & a \neq b \\ \left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right) & a = b \end{cases}. \tag{3}$$

Here, $\mathbb{I}(.)$ represents the indicator function. The proof is in the supplementary material. In Section 4, we show how this model relates to other state-of-the-art algorithms.

2.2 Hierarchical Clustering Based Bayesian Combination of Classifiers

In the cBCC model, the users that belong to the same cluster share the same confusion matrix. However, in a practical situation, each user has a behavior that is different from every other user, but it is in some sense similar to the behavior of users that are allocated to its cluster.

To capture this behavior, we propose a hierarchical extension of the cBCC model called hcBCC (hierarchical cBCC) depicted in Figure 2. The observation model is the following

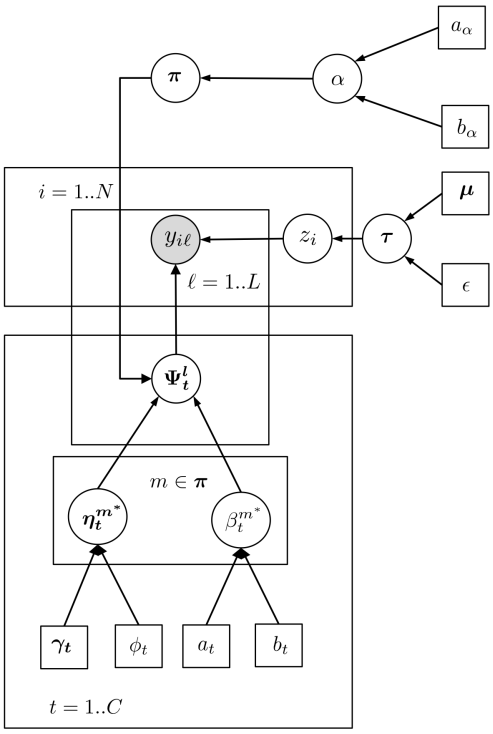


Figure 2: Graphical model representation of the hcBCC model

$$y_{i\ell}|z_i, \boldsymbol{\Psi} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\Psi}^{\ell}_{z_i}),$$
$$z_i|\boldsymbol{\tau} \overset{i.i.d}{\sim} \text{Discrete}(\boldsymbol{\tau}).$$

Now each user has its own confusion matrix $\boldsymbol{\Psi}^{\ell}$ in contrast to the cBCC model where we had a confusion matrix per cluster $\boldsymbol{\Psi}^{m}$. To capture the similarity between users that belong to the same cluster we use the following hierarchical prior:

$$\boldsymbol{\Psi}^{\ell}_t|\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\eta} \sim \text{Dir}(\beta_t^{q_\ell}\boldsymbol{\eta}_t^{q_\ell}),$$
$$\beta_t^m|a_t, b_t \sim \text{Gamma}(a_t, b_t),$$
$$\boldsymbol{\eta}_t^m|\boldsymbol{\phi}, \boldsymbol{\gamma} \sim \text{Dir}(\boldsymbol{\phi}_t\boldsymbol{\gamma}_t),$$
$$\boldsymbol{\tau}|\varepsilon, \boldsymbol{\mu} \sim \text{Dir}(\varepsilon\boldsymbol{\mu}).$$

In this way, the confusion matrices of all users that belong to the same cluster m are generated from the same distribution. In particular, each of the rows of the confusion matrices of all the users that belong to cluster m, i.e. $\{\boldsymbol{\Psi}^{\ell}_t : q_\ell = m\}$, are i.i.d. samples from the same Dirichlet distribution whose parameters are β_t^m and $\boldsymbol{\eta}_t^m$. A Dirichlet prior is set on the vector $\boldsymbol{\eta}_t^m$ while a gamma prior is set on the scalar β_t^m. Finally, for $\boldsymbol{\pi}$ and α we, respectively, use the same priors given by Equations 1 and 2. With this we have a model where we no longer cluster the confusion matrices of the users, but the distributions that generate them.

Notice that the vector $\boldsymbol{\beta}^m$ governs the variability among the users that belong to the same cluster m. The bigger are these values, the lower is the intra-cluster variability. If we make each of the components of $\boldsymbol{\beta}^m$ tend to infinity, then the variability among the users tend to 0 and the model becomes equivalent to the cBCC model. In this way, this model can be seen as a generalization of some state-of-the-art methods (see Section 4).

3. Inference

Computing the posterior distribution of the clusters allocation, the properties of the users and the estimated ground is intractable, so we have to resort to approximate inference. Since the proposal of the DP by Ferguson (1973), approximate inference schemes based on Markov Chain Monte Carlo (MCMC) methods have played a crucial role (Escobar, 1994; MacEachern and Müller, 1998; Neal, 2000; Ishwaran and James, 2001; Walker, 2007; Kalli et al., 2011) among others. In this section we propose to use Gibbs sampling together with the corresponding auxiliary variables whenever it is not possible to compute the conditional distributions due to non-conjugacies.

3.1 CBCC

We use a collapsed Gibbs sampling algorithm where we integrate out the variables $\boldsymbol{\Psi}^m$ and $\boldsymbol{\tau}$, obtaining the following new set of equations

$$p(\boldsymbol{Y}|\boldsymbol{\pi},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_m \prod_t \left[\frac{\Gamma(\beta_t)}{\Gamma(n_{mt}+\beta_t)} \prod_c \frac{\Gamma(n_{mtc}+\beta_t\eta_{tc})}{\Gamma(\beta_t\eta_{tc})} \right],$$
$$p(\boldsymbol{z}|\boldsymbol{\varepsilon},\boldsymbol{\mu}) = \frac{\Gamma(\varepsilon)}{\Gamma(N+\varepsilon)} \prod_t \frac{\Gamma(n_m+\varepsilon\mu_t)}{\Gamma(\varepsilon\mu_t)},$$

where $\Gamma(\cdot)$ denotes the gamma function. We denote $n_{i\ell mtc} = \mathbb{I}(z_i = t, y_{i\ell} = c, y_{i\ell} \neq 0, q_\ell = m)$, and when an index of this variable is omitted we assume it is summed out. For example, n_{mtc} represents the number of annotations equal to c provided by the users of cluster m for the set of instances whose ground truth is equal to t. We use Gibbs sampling to infer the value of the ground truth $\boldsymbol{z}$, the clusters of annotators $\boldsymbol{\pi}$, as well as the hyper parameters of the CRP, conditioned on the observed variables $\boldsymbol{Y}$.

Firstly, to update the cluster assignment of annotator ℓ, we need the conditional distribution of q_ℓ given the rest of the variables

$$p(q_\ell = m|\text{rest}) \propto \begin{cases} n_m^{\neg\ell} \times \prod_t \frac{\Gamma(n_{mt}^{\neg\ell}+\beta_t)}{\Gamma(n_{mt}^{\neg\ell}+n_{\ell t}+\beta_t)} \prod_t \prod_c \frac{\Gamma(n_{mtc}^{\neg\ell}+n_{\ell tc}+\beta_t\eta_{tc})}{\Gamma(n_{mtc}^{\neg\ell}+\beta_t\eta_{tc})}, & m \in \boldsymbol{\pi}^{\neg\ell} \\ \alpha \times \prod_t \frac{\Gamma(\beta_t)}{\Gamma(n_{\ell t}+\beta_t)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t\eta_{tc})}{\Gamma(\beta_t\eta_{tc})}, & m = \emptyset \end{cases},$$

where $q_\ell = \emptyset$ denotes the event that user ℓ is assigned to a new cluster. The quantities $n_{mt}^{\neg\ell}$ and $n_{mtc}^{\neg\ell}$ are defined in the same way as n_{mt} and n_{mtc} respectively, but excluding the annotator ℓ. The complexity of updating the $\boldsymbol{q}$ variables is $O(LMTC)$. To sample the estimate of the ground truth z_i of each instance conditioned on the rest of the variables, the required conditional distribution is

$$p(z_i = t|\text{rest}) \propto \left(n_t^{\neg i} + \varepsilon\mu_t\right) \times \prod_m \left[\frac{\Gamma(n_{mt}^{\neg i}+\beta_t)}{\Gamma(n_{mt}^{\neg i}+n_{im}+\beta_t)} \prod_c \frac{\Gamma(n_{mtc}^{\neg i}+n_{imc}+\beta_t\eta_{tc})}{\Gamma(n_{mtc}^{\neg i}+\beta_t\eta_{tc})} \right].$$

The quantities $n_{mt}^{\neg i}$ and $n_{mtc}^{\neg i}$ again correspond to n_{mt} and n_{mtc} but excluding the instance i. The complexity of updating the $\boldsymbol{z}$ variables is $O(NMTC)$

Finally, we sample the concentration parameter α following the procedure proposed by Escobar (1994).

3.2 HCBCC

As in the cBCC we start by integrating out the $\boldsymbol{\Psi}^{\ell}$ and $\boldsymbol{\tau}$ variables

$$p(\boldsymbol{Y}|\boldsymbol{z},\boldsymbol{\pi},\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_m \prod_{\ell:q_\ell=m} \prod_t \left[\frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t \eta_{tc}^m)} \right],$$
$$p(\boldsymbol{z}|\boldsymbol{\varepsilon},\boldsymbol{\mu}) = \frac{\Gamma(\boldsymbol{\varepsilon})}{\Gamma(N+\boldsymbol{\varepsilon})} \prod_t \frac{\Gamma(n_m+\boldsymbol{\varepsilon}\mu_t)}{\Gamma(\boldsymbol{\varepsilon}\mu_t)}.$$

The variables we need to sample from are $\boldsymbol{\pi}$ and the ground truth estimate $\boldsymbol{z}$. Note however that we cannot marginalize out the cluster parameters $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$, as the Dirichlet prior and the Gamma prior are not conjugate to the likelihoods given above, so that these variables will have to be sampled as well.

The conditional distribution of $p(q_\ell = m|rest)$ when $m \in \boldsymbol{\pi}^{\neg \ell}$ can be computed like in the cBCC model. However, to compute $p(q_\ell = m|rest)$ when $m = \emptyset$ we need to integrate the parameters of the new clusters, i .e . $\boldsymbol{\beta}$ and $\boldsymbol{\eta}$. In this case, due to the non-conjugacy we cannot solve this integral analytically. Instead, we use the recently proposed *Reuse algorithm* (Favaro and Teh, 2012). This algorithm is similar to the well-known Algorithm 8 (Neal, 2000), where the idea is to use a set of h auxiliary empty clusters H_{empty} to approximate the integral. However, the *reuse algorithm* is more efficient as it requires less simulations from the prior over the cluster parameters. For each cluster $m \in \boldsymbol{\pi} \cup H_{empty}$ we keep track of the parameters $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$. The conditional distribution of q_ℓ is then

$$p(q_\ell = m|\text{rest}) \propto \begin{cases} n_m^{\neg \ell} \times \prod_t \frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t^m \eta_{tc}^m)}, & m \in \boldsymbol{\pi}^{\neg \ell} \\ \frac{\alpha}{h} \times \prod_t \frac{\Gamma(\beta_t^m)}{\Gamma(n_{\ell t}+\beta_t^m)} \prod_t \prod_c \frac{\Gamma(n_{\ell tc}+\beta_t^m \eta_{tc}^m)}{\Gamma(\beta_t^m \eta_{tc}^m)}, & m \in H_{empty} \end{cases}.$$

If an auxiliary empty cluster is chosen, it is moved into the partition $\boldsymbol{\pi}$, and a new empty cluster is created in its place by sampling from the prior over cluster parameters. If a cluster in $\boldsymbol{\pi}$ is emptied as a result of sampling q_ℓ, it is moved into H, displacing one of the empty clusters (picked uniformly at random). In addition, at regular intervals the parameters of the empty clusters are refreshed by simulating them from their priors, while those in $\boldsymbol{\pi}$ are updated. The complexity of updating the $\boldsymbol{q}$ variables is $O(LMTC)$.

Again, due to the non-conjugacy of the Dirichlet and Gamma priors, the conditional distributions of the parameters $\boldsymbol{\eta}^m$ and $\boldsymbol{\beta}^m$ for $m \in \boldsymbol{\pi}$ cannot be computed analytically. To solve this, we use an auxiliary variable method similar to the one proposed by Escobar (1994) and Teh et al. (2003). Specifically, we introduce two auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ (see the supplementary material for fur-

ther details), and apply the following Gibbs updates that leave invariant the posterior distribution:

$$\nu_{\ell t} \sim \text{Beta}(\beta_t^{q_\ell}), \qquad s_{\ell tc} \sim \text{Antoniak}(n_{\ell tc}, \beta_t^{q_\ell}\eta_{tc}^{q_\ell}),$$
$$\eta_{t:}^m \sim \text{Dir}\left(\sum_{\{\ell:q_\ell=m\}} s_{\ell t:} + \phi_t \gamma_{t:}\right),$$
$$\beta_t^m \sim \text{Gamma}\left(\sum_{\{\ell:q_\ell=m\}}\sum_c s_{\ell tc} + a_t, b_t - \sum_{\{\ell:q_\ell=m\}} \log(\nu_{\ell t})\right).$$

Here the Antoniak distribution introduced by Antoniak (1974) is simply the distribution of the number of clusters in a partition of $n_{\ell tc}$ items under a CRP with concentration parameter $\beta_t^{q_\ell}\eta_{tc}^{q_\ell}$.

To update z_i, we compute its conditional distribution given the rest of the variables:

$$p(z_i = t|\text{rest}) \propto \left(n_t^{\neg i} + \varepsilon\mu_t\right)\prod_m \prod_{\{\ell:q_\ell=m\}} \frac{\prod_c (n_{\ell tc}^{\neg i} + \beta_t\eta_{tc})^{\mathbb{I}(y_{i\ell}=c)}}{(n_{\ell t}^{\neg i} + \beta_t)^{\mathbb{I}(y_{i\ell}\neq 0)}}.$$

The complexity of updating the $\boldsymbol{z}$ variables is $O(NLTC)$. Finally, we use the same scheme as the one applied in Section 3.1 to update α.

4. Related Work

Dawid and Skene in a seminal work proposed a model in which each user is characterized by a confusion matrix, and they use the EM algorithm to estimate the most likely values of both the parameters governing the behavior of each user and the ground truth (Dawid and Skene, 1979). Similar models have been applied to depression diagnosis (Young et al., 1983) and myocardial infarction (Rindskopf and Rindskopf, 1986), among other areas.

Ghahramani and Kim (2003); Kim and Ghahramani (2012) proposed a Bayesian extension of the method proposed by Dawid and Skene (1979) called Independent Bayesian Combination of Classifiers (iBCC), whose graphical model is shown in Figure 1a. In our cBCC model, if α tends to infinity, every user is allocated in a different cluster, and it becomes equivalent to the iBCC model. We see that in this case, the correlation a priori among two users (Equation 3) is zero. Also, in the hcBCC model, if each component of ϕ tends to infinity, and we also make the quantities a_t and b_t tend to infinity with a fixed $\frac{a_t}{b_t}$ ratio for all t, then we recover the iBCC model with $\eta_t = \gamma_t$ and $\beta_t = \frac{a_t}{b_t}$. If α tends to ∞, then the model is equivalent to the iBCC model, but with additional priors on η and β. To sum up, we can see each the cBCC and the iBCC as particularizations of the hcBCC model, which capture more complex relationships among the users.

Ghahramani and Kim (2003); Kim and Ghahramani (2012) also presented two extensions. The first one uses a latent variable that categorizes the instances in two classes: easy and difficult to classify. The assumption is that the annotators have the same behavior regarding the easy instances, while they are different for the difficult ones. In the second one, they propose a more flexible correlation model based on a factor graph. However, these models do not identify groupings of users.

Simpson et al. (2011) extend the proposal of Ghahramani and Kim (2003); Kim and Ghahramani (2012) in two directions. First, they derived a variational inference algorithm for the iBCC, which is more efficient for large data-sets. Second, they apply community detection algorithms to the estimated confusion matrices to detect clusters of users with the same behavior. Recently, they have

extended the model to the case in which the properties of the users can vary in time (Simpson et al., 2013). In both cases, the detection of groups of users is made in a post-hoc manner and therefore, this information is not used to improve the estimation of the confusion matrices of the users or the ground truth estimate. To the extent of our knowledge, only Kajino et al. (2013) perform the inference of the groups of users and the ground truth at the same time, using convex optimization. However, the performance depends on a constant that controls the strength of the clustering and for tuning this constant, the authors rely on a labeled validation set. Our algorithm, on the other hand, is fully unsupervised and therefore can be apply to the standard problem presented by Ghahramani and Kim (2003); Kim and Ghahramani (2012).

Recently, a paper on the inconsistency of the DP Mixture Model to estimate the true number of components was published (Miller and Harrison, 2013). However, we are not interested in estimating the "true" number of users' clusters, specially since this is not a well defined measure in a real crowdsourcing application. Instead, we look for identifying a clustering of the users that improves the performance and helps us to better understand the different types of users that are present in the crowdsourcing application.

Another research line that is related to the problem is relaxing the assumption of the existence of one single gold standard, which is a limiting assumption when the tasks involved in the crowdsourcing problem are subjective and accept multiple reasonable answers (Wauthier and Jordan, 2011; Tian and Zhu, 2012). In this paper, we focus on a crowdsourcing scenario with a well defined gold standard that we aim to predict.

5. Experiments

In this section, we firstly use synthetic data-sets based on different assumptions to validate our models. In the second part we use publicly available real data-sets to compare our two models with state-of-the-art algorithms highlighting their advantages.

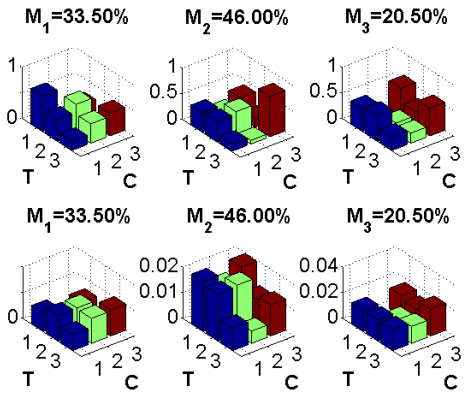
5.1 Synthetic Data-sets

We generate three different data-sets following respectively the assumptions of the hcBCC, cBCC and iBCC models. In order to analyze the properties of the algorithms, we apply each of them to each of the generated data-sets.

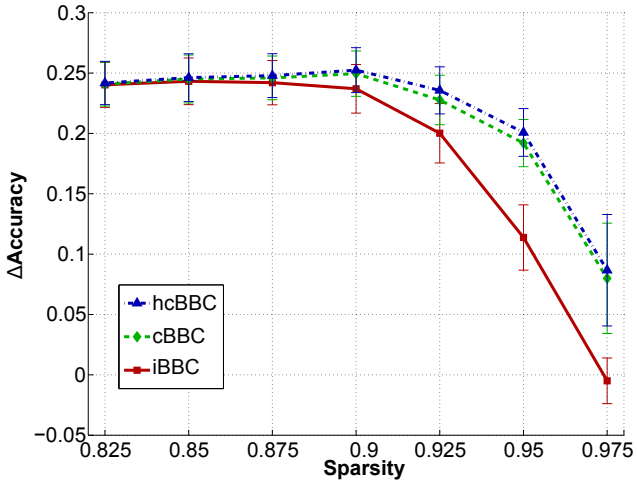
Firstly, we generate a synthetic database called data-set1 following the generative model for the hcBCC model. This data-set has 500 labeled instances provided by 200 users. The number of categories is $C = 3$. These users belong to 3 clusters with properties shown in Figure 3a, where we can see the mean of each cluster, their variances and the percentage of users allocated to each of them.

We analyze the behavior of the different algorithms with respect to the sparsity of the input matrix $\boldsymbol{Y}$. In particular, we randomly erase a percentage of the entries from 82.5% missing entries to 97.5% in steps of 2.5%. This high sparsity levels are typical in crowdsourcing applications, where the idea is to distribute the load of labeling a data-set among many users, and therefore each of them only labels a small subset of the data-set.

In the iBCC, the diagonal elements of $\boldsymbol{\eta}$ are 0.7 while the off diagonal are 0.3, which reflect our prior belief that users perform better than random. All the elements of $\boldsymbol{\beta}$ are 3. In the cBCC model, the hyper parameters of $\boldsymbol{\alpha}$ are $a_\alpha = 1$ and $b_\alpha = 10$. This values agree with our prior belief that if the annotations are very scarce, simpler algorithm like majority voting are more suitable and therefore,



(a) Users' properties for data-set1. (Upper row) Mean of the clusters. (Lower row) Variance of the clusters.



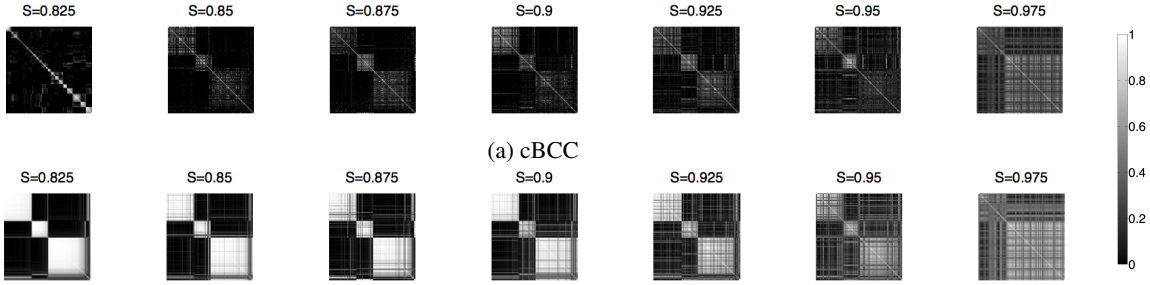
(b) Performance for data-set1

Figure 3: Results for data-set1. a) Characteristics of the users' clusters present in data-set1. M_i denotes the percentage of users allocated to cluster i, T is the ground truth label, and C is the user label. b) Results for data-set 1. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels.

we should favor partitions with a small number of clusters. For these parameters, in the limiting case when the sparsity of $\boldsymbol{Y}$ tends to 100%, the average number of clusters tends to 1. Finally, in the hcBCC model, we set γ and ϕ to the values of η and β in the cBCC model respectively. All the components of a_t are set to 30 while all the components of b_t are set to 2. This reflects our prior belief that the variability among the users inside the clusters should be less than the variability across clusters.

We run the MCMC for 10,000 iterations. After the first 3,000 we collect 7,000 samples to compute z and π. In the cBCC and hcBCC, we set to five the number of iterations used to sample α following the algorithm proposed by Escobar (1994). In the hcBCC we fix the number of auxiliary clusters used by the *Reuse Algorithm* to $h = 10$.

The increment in accuracy of ours proposals and the iBCC algorithm with respect to majority voting is shown in Figure 3b. The two proposed models outperform iBCC as expected. This improvement of both methods cBCC and hcBCC is particularly significant when the level of sparsity is high, which is a situation that we face in the early stages of a crowdsourcing project. In this case there is not enough information to accurately estimate the confusion matrix of every user independently. We can see that the performance of iBCC drops below the performance of the majority voting algorithm, which assumes all users are similar. Therefore, identifying a clustering structure that allows to share some parameters among the users helps to increase the accuracy of the estimates. Notice that the performance obtained by Simpson et al. (2011) would be equal to the performance of the iBCC model given that it identifies the users' clusters after the ground truth has been estimated, so it does not affect the performance of the algorithm.



(a) cBCC

(b) hcBCC

Figure 4: Co-occurrence matrices of the users

To further analyze the cluster structure identified by the algorithms, in Figure 4 we represent the co-occurrence matrix of the users. The position (ℓ, ℓ') is the probability of ℓ and ℓ' belonging to the same cluster. We can see that the clusters identified by the hcBCC are more useful than the ones extracted by the cBCC, because in a practical situation we are not normally interested in finding users with exactly the same behavior, but users with similar characteristics. For example, we can see that when 82.5% of the annotations are missing, the hcBCC algorithm identifies the 3 main groups of users while the cBCC algorithm identifies instead a much larger number of groups because of the constraint that all users of a cluster must have the same properties. So, although both algorithms' performance is similar, the clustering provided by the hBCC is easier to interpret and gives a simpler explanation of the data.

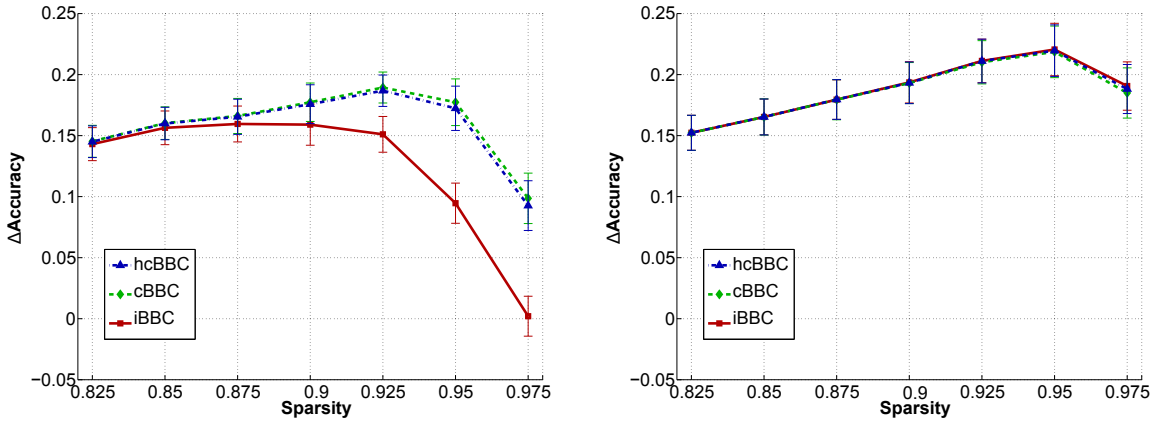
Finally, we test with data-sets that are generated following the iBCC and cBCC models. First, we create a new data-set (data-set2) in which the mean confusion matrix of each cluster is the same as in data-set1 which is shown in Figure 3a. However, in this case the variability of the confusion matrices inside each cluster is zero. Therefore, this new data-set follows the assumptions made by the cBCC model. Again, the performance of the cBCC and the hcBCC models outperforms iBCC as expected (see Figure 5a). However, even though data is generated from the cBCC which is a simpler model than the hcBCC, hcBCC is able to discover the underlying structure of the users and gets a performance which is on par with the cBCC. The hcBCC does not degrade the solution although it is more flexible.

In the last database called data-set3, we generate all the instances from the same clustering (M_2 in Figure 3a). In this case there is no different clusters of users and each of them has its own confusion matrix. Therefore, this data-set fulfill the assumptions of the iBCC. In Figure 5b we see that the performance of the two proposed models is identical to iBCC. To sum up, we see that the performances of cBCC and hcBCC dominate iBCC under all conditions tested.

5.2 Real Data-sets

In this Section, we use 4 publicly available crowdsourced data-sets with $C = 2$ whose principal characteristics are described in Table 1 (Raykar and Yu, 2012).

To choose the hyper-parameters we follow the reasoning of Section 5.1. Specifically, in the iBCC the diagonal elements of $\boldsymbol{\eta}$ are 0.7 while the off diagonal are 0.3, and all the elements of $\boldsymbol{\beta}$ are set to 3. In this way, we incorporate our prior belief that users are imperfect but perform better



(a) Performance for data-set2

(b) Performance for data-set3

Figure 5: Results for data-set2 and data-set3. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels

than chance. In the cBCC model we use the same value for η and β so that the comparison is fair. Finally for the hcBCC model, γ is set to the same value used for η in the previous models. All the components of a_t are set to 20 while all the components of b_t are set to 2, reflecting our belief that the variability inside clusters should be lower than the variability across clusters. We fix $a_\alpha = 1, b_\alpha = 10$ in both, cBCC and hcBCC. We run the MCMC for 10,000 iterations and we discard the first 3,000 to compute the posterior distribution of z and π.

In Table 2, we see the performance of the different algorithms in terms of accuracy predicting the ground truth. In particular, we see that the performances of the cBCC and the hcBCC are better than that of the iBCC in the last three data-sets, i.e. rte, temp and valence. On the other hand, in the bluebird data-sets the iBCC performs better. Notice again that the performance of the algorithm described by Simpson et al. (2011) would be exactly equal to the one of the iBCC, given that the communities of users are inferred after the ground truth is inferred and therefore, it does not affect the accuracy in any way.

The performance difference between the cBCC and the hcBCC is only significant in the valence data-set. However, the main advantage of the hcBCC model over the cBCC is clear when we represent the average number of clusters (See Figure 6 and Table 2). Even though the cBCC model correctly captures the clustering structure of the users, forcing all users of a cluster to share the same confusion matrix translates into a large number of clusters, some of them with very similar properties.

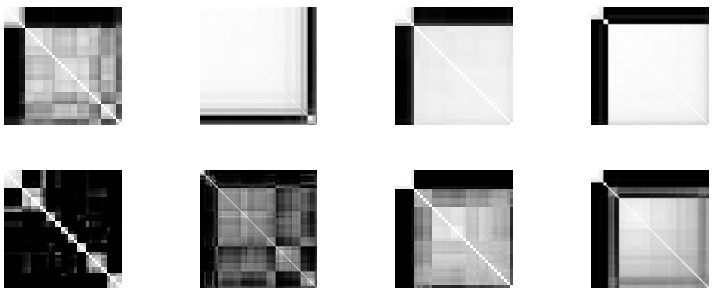
The hcBCC identifies a smaller number of clusters that are much more interpretable, in the sense that it perfectly identifies each kind of clusters thanks to its additional flexibility. We are not interested in identifying clusters of users with the exact same behavior, but what we really want is to find clusters of users that behave in a similar way, so we can establish strategies to boost the overall performance of the crowdsourcing system, i.e. by rewarding the most efficient labelers, avoiding spammers or by better defining the description of the task based on the biases identified in the clusters of users.

data-set	N	L	μ_n	μ_l	Sparsity (%)	Brief Description
bluebird	108	39	108	39	0	Identify whether there is a Indigo Bunting or Blue Grosbeak in the image
rte	800	164	49	10	93.90	Identify wether the second sentence can be inferred from the first
valence	100	38	26	10	73.68	Identify the overall positive or negative valence of the emotional content of a headline
temp	462	76	61	10	86.84	Users observe a dialogue and two verbs from the dialogue and have to identify whether the first verb event occurs before or after the second

Table 1: Description of the real data-sets. N and L denotes the number of instances and users respectively. μ_n stand for the mean number of instances labeled by a user and μ_l designate the mean number of users that label an instance.

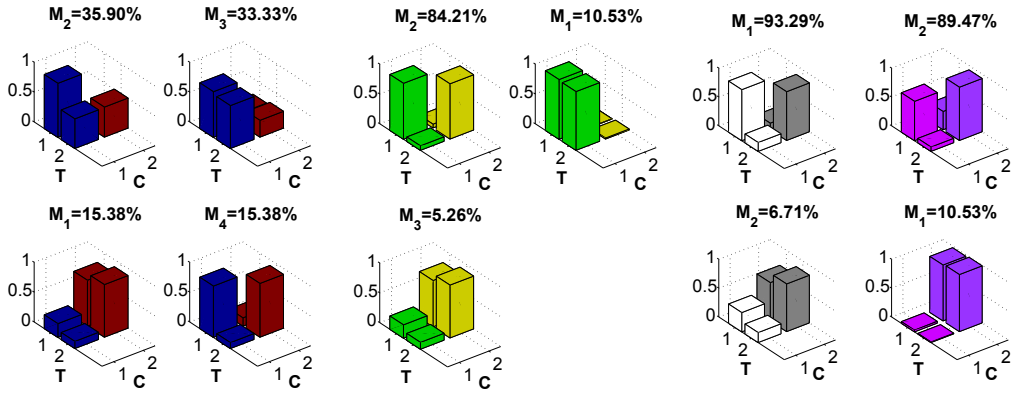
data-set	Accuracy(%)				Average number of clusters	
	Majority	iBCC	cBCC	hcBCC	cBCC	hcBCC
bluebird	75.93	**89.81**	88.89	88.89	11.32 ± 0.04	3.31 ± 0.09
rte	91.88	92.88	**93.12**	**93.12**	7.70 ± 0.07	2.30 ± 0.06
valence	80.00	85.00	88.00	**89.00**	3.5 ± 0.04	2.25 ± 0.02
temp	93.94	94.35	**94.37**	**94.37**	6.20 ± 0.03	3.2 ± 0.02

Table 2: Results for the real data. Mean accuracy of the different algorithms [2]. Average number of clusters (mean $\pm$ one standard deviation).



(a) bluebird (b) rte (c) valence (d) temp

Figure 6: Co-occurrence matrix of the users. (Upper row) hcBCC. (Lower row) cBCC.



(a) bluebird (b) temp (c) rte (d) valence

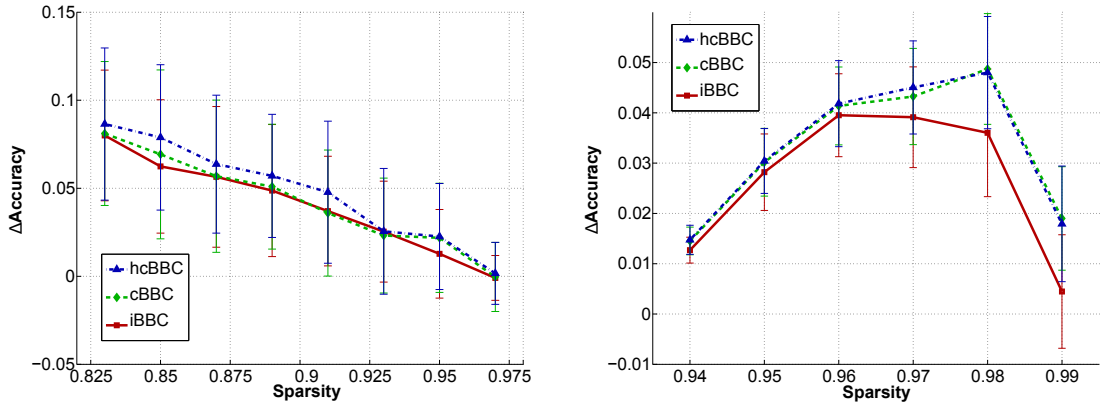
Figure 7: Mean confusion matrices of the user's clusters identified by hcBCC. M_i denotes the percentage of users allocated to cluster i, T is the ground truth label, and C is the user label.

In Figure 7 we show as an example the mean confusion matrix of the hcBCC clusters in the data-sets. It shows very interpretable clusters that are useful for the modeler. In the bluebird data-set we can clearly identify a small subset of experts ($M_4 = 15.38\%$) who shows a high performance labeling the bird images. In addition, we find that the biggest cluster ($M_2 = 35.90\%$) corresponds to users whose accuracy is high when the real class is $z = 1$ (images of Blue Grosbeak) but performs poorly when the class is $z = 2$ (images of Indigo Bunting). Finally, we have two clusters of spammers. In the first cluster ($M_1 = 15.38\%$) users tend to label all images as belonging to class $z = 2$ and in the second ($M_3 = 33.33\%$) users tend to label all images as $z = 1$. In the temp data-set, we can observe that the majority of the users ($M_2 = 84.21\%$) are experts, but there are again two small clusters of spammers. As for the rte data-set, most of the users have a good performance ($M_1 = 93.29\%$). The remaining users are bias toward labeling instances as belonging to class $z = 2$. Finally, in the valence data-set we can see that the majority of the users ($M_2 = 89.47\%$) are very accurate identifying instances belonging to class $z = 2$ and have a medium performance when $z = 1$. In addition we find a small cluster of users that have labeled almost every instance as $z = 2$. All this information about the underlying clustering structure of the users in the data-sets can be used in a real crowdsourcing application to develop efficient strategies to minimize the cost of a crowdsourcing project maximizing the performance.

To conclude this Section, we evaluate the performance of the algorithms in the real data-sets for different levels of sparsity. Following the procedure in Section 5.1, we create 50 random databases for each level of sparsity. We do that in such a way that every instance has at least one label and every user provides at least one label. The results are shown in Figure 8.

In the data-set bluebird and temp, we observe that finding clusters of users does not have a significant effect in terms of accuracy. However, the cBCC and the hcBCC models do not degrade the performance and give us some insight about the users in the crowdsourcing application (See

2. The standard deviations are less than 10^{-4} and are not shown



(a) Performance for bluebird

(b) Performance for rte

(c) Performance for valence

(d) Performance for temp

Figure 8: Results for real data-sets. Improvement in accuracy of the different methods with respect to majority voting, for different sparsity levels

Figure 7). In the rte and valence data-sets, the inference of the clustering structure of the users also translates into an improvement in terms of accuracy. In the rte data-set, this improvement is not significant for the original sparsity level, but it becomes more significant when the sparsity is increased. What happens is that when the sparsity is very high, there are very few annotations provided by each user, and the iBCC algorithm fails to infer the properties of each user separately.

In the valence data-set, we can even see that the performance of the iBCC model drops below the performance of a simple majority voting algorithm when the sparsity is increased. However, the cBCC and hcBCC outperform the majority voting algorithm for every sparsity level. Again the iBCC model does not have enough information to infer the properties of each user and a simpler model like majority voting, which assume that all users have the same level of expertise, performs better. Actually, what is happening is that the the CRP prior used in the cBCC and the hcBCC

models favors partitions with a small number of clusters. When the input matrix $\boldsymbol{Y}$ is very sparse, the prior term dominates over the likelihood and all users tend to be grouped in the same cluster.

6. Conclusions

We have proposed two new Bayesian nonparametric models to merge the information provided by the users in a crowdsourcing system. In addition, the algorithms detect clusters of users that have similar behaviors and use this information to improve the ground truth estimate. In the cBCC model, we have used a CRP to infer the partitioning of the users such that users in the same cluster are constrained to have the same properties. In the hcBCC model, we have used a hierarchical structure to increase the flexibility. In particular, each user has its own properties, but users assigned to the same cluster have similar properties. In this way, it finds smaller number of clusters that are easy to interpret.

We have shown how these new models relate to the iBCC model and analyzed the correlation structure among the users as a consequence of the clustering. We have proposed MCMC methods to infer the parameters of both models and performed several experiments with synthetic and real databases, which have shown that the algorithms outperform the current state-of-the-art.

Finally, we comment possible extensions. The ground truth estimated by the proposed algorithms, can be used to train a supervised learning algorithm. Raykar et al. (2010); Groot et al. (2011); Welinder et al. (2010) propose to train a classifier directly with the noisy labels provided by the users. It would be interesting to extend the models following this line. Also, the models assume consistent users, i.e the users have the same properties across the whole instance space. An extension would be considering users with nonuniform behavior (Zhang and Obradovic, 2011; Yan et al., 2010), i.e. a user can be an expert for a subset of the instances while can act as a novice in another subset. Also, a future research line is to propose new inference schemes that improve the scalability of the methods.

Acknowledgments

Pablo G. Moreno is supported by an FPU fellowship from the Spanish Ministry of Education (AP2009-1513). This work has been partly supported by Ministerio de Economía of Spain ('COMONSENS', id. CSD2008-00010, 'ALCIT', id. TEC2012-38800-C03-01, 'COMPREHENSION', id. TEC2012-38883-C02-01) and Comunidad de Madrid (project 'CASI-CAM-CM', id. S2013/ICE-2845). This work was also supported by the European Union 7th Framework Programme through the Marie Curie Initial Training Network "Machine Learning for Personalized Medicine" MLPM2012, Grant No. 316861. Yee Why Teh's research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) ERC grant agreement no. 617411. Finally, the authors would like to thank the anonymous reviewers for their constructive comments that helped to improve the quality of this paper

Appendix A: Correlations in the CBCC Model

In the iBCC model, the joint probability of two users given the ground truth is

$$p(y_{i\ell}, y_{i\ell'}|z_i = t) = \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0))}{\Gamma(\beta_t \eta_{tc})} \times$$
$$\times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell'} \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t \eta_{tc})} = p(y_{i\ell}|z_i = t) \times p(y_{i\ell'}|z_i = t).$$

Therefore

$$\text{corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = 0.$$

In the cBCC model, we have the following expression for the joint distribution of $y_{i\ell}$ and $y_{i\ell'}$

$$p(y_{i\ell}, y_{i\ell'}|z_i = t) = \left(\frac{1}{1+\alpha}\right) \left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0) + \mathbb{I}(y_{i\ell'} \neq 0))} \times \right.$$
$$\left. \times \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0) + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t \eta_{tc})}\right] + \left(\frac{\alpha}{1+\alpha}\right) \left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell} \neq 0))} \times \right.$$
$$\left. \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell} = c, y_{i\ell} \neq 0))}{\Gamma(\beta_t \eta_{tc})} \times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(y_{i\ell'} \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(y_{i\ell'} = c, y_{i\ell'} \neq 0))}{\Gamma(\beta_t \eta_{tc})}\right].$$

We can now compute the covariance in the following way

$$\text{cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = \mathbb{E}\{\mathbb{I}(y_{i\ell} = a)\mathbb{I}(y_{i\ell'} = b)\} - \mathbb{E}\{\mathbb{I}(y_{i\ell} = a)\}\mathbb{E}\{\mathbb{I}(y_{i\ell'} = b)\} =$$
$$= \left(\frac{1}{1+\alpha}\right) \left[\frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(a \neq 0) + \mathbb{I}(b \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(a = c, a \neq 0) + \mathbb{I}(b = c, b \neq 0))}{\Gamma(\beta_t \eta_{tc})} - \right.$$
$$\left. - \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(a \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(a = c, a \neq 0))}{\Gamma(\beta_t \eta_{tc})} \times \frac{\Gamma(\beta_t)}{\Gamma(\beta_t + \mathbb{I}(b \neq 0))} \prod_c \frac{\Gamma(\beta_t \eta_{tc} + \mathbb{I}(b = c, b \neq 0))}{\Gamma(\beta_t \eta_{tc})}\right].$$

Assuming that $a \neq 0$ and $b \neq 0$, and considering the cases where $a = b$ and $a \neq b$ we obtain the following equation for the covariance

$$\text{Cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i'\ell'} = b)|z_i = t) = \begin{cases} -\left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\eta_{ta}\eta_{tb} & a \neq b \\ \left(\frac{1}{1+\alpha}\right)\left(\frac{1}{1+\beta_t}\right)\eta_{ta}(1-\eta_{ta}) & a = b \end{cases}.$$

Here we have taken into account that $\Gamma(x+1) = x\Gamma(x)$. Once we get the expression of the covariance, we divide it by the square root of the variances to get the correlation

$$\text{Corr}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b)) = \frac{\text{Cov}(\mathbb{I}(y_{i\ell} = a), \mathbb{I}(y_{i\ell'} = b))}{\sqrt{\text{Var}(\mathbb{I}(y_{i\ell} = a))\text{Var}(\mathbb{I}(y_{i\ell'} = b))}}.$$

It is straightforward to see that $\text{Var}(\mathbb{I}(y_{i\ell} = a)) = \eta_a(1-\eta_a)$, getting the expected result.

Appendix B: Inference Details of the HCBCC Model

The posterior distribution of the parameters $\boldsymbol{\beta}^m$ and $\boldsymbol{\eta}^m$ is proportional to the following expression.

$$p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) \propto p(\boldsymbol{Y}|\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{z},\boldsymbol{\pi}) \times p(\boldsymbol{\eta},\boldsymbol{\beta}) = \prod_{\ell}\prod_{t}\left[\frac{\Gamma(\beta_t^{q_\ell})}{\Gamma(n_{\ell t}+\beta_t^{q_\ell})}\prod_{c}\frac{\Gamma(n_{\ell tc}+\beta_t^{q_\ell}\eta_{tc}^{q_\ell})}{\Gamma(\beta_t\eta_{tc}^{q_\ell})}\right]\times$$
$$\times\prod_{m}\prod_{t}\left[\frac{\Gamma(\phi_t)}{\prod_c\Gamma(\phi_t\gamma_{tc})}\prod_{c}(\eta_{tc}^m)^{\phi_t\gamma_{tc}-1}\right]\prod_{m}\prod_{t}\frac{b_t^{a_t}}{\Gamma(a_t)}(\beta_t^m)^{a_t-1}\exp(-b_t\beta_t^m).$$

We cannot compute an analytic expression for $p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi})$ because the prior on $p(\boldsymbol{\eta},\boldsymbol{\beta})$ is no longer conjugate of the likelihood of the observations. The idea is to include two auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ such that we can compute the joint distribution $p(\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{\nu},\boldsymbol{s}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi})$. To do so, we use the following relation between the gamma function and the Stirling numbers of the first kind denoted by S

$$\frac{\Gamma(x+n)}{\Gamma(x)} = (x)_n = \sum_{s=0}^{n} S(n,s)(x)^s.$$

Here $(x)_n$ denotes the Pochhammer symbol. Taking into account also the definition of the beta distribution we reach the following expression

$$p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) \propto \prod_{\ell}\prod_{t}\left[\int_0^1 \nu^{\beta_t^{q_\ell}-1}(1-\nu)^{n_{\ell t}-1}d\nu\prod_{c}\sum_{s=0}^{n_{\ell tc}}S(n_{\ell tc},s)(\beta_t^{q_\ell}\eta_{tc}^{q_\ell})^s\right]\times$$
$$\times\prod_{m}\prod_{t}\left[\frac{\Gamma(\phi_t)}{\prod_c\Gamma(\phi_t\gamma_{tc})}\prod_{c}(\eta_{tc}^m)^{\phi_t\gamma_{tc}-1}\right]\prod_{m}\prod_{t}\frac{b_t^{a_t}}{\Gamma(a_t)}(\beta_t^m)^{a_t-1}\exp(-b_t\beta_t^m).$$

And therefore we can introduce a set of auxiliary variables $\boldsymbol{\nu}$ and $\boldsymbol{s}$ such that the joint distribution is given by

$$p(\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{\nu},\boldsymbol{s}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) \propto \prod_{\ell}\prod_{t}\left[\nu_{\ell t}^{\beta_t^{q_\ell}-1}(1-\nu_{\ell t})^{n_{\ell t}-1}\prod_{c}S(n_{\ell tc},s_{\ell tc})(\beta_t^{q_\ell}\eta_{tc}^{q_\ell})^{s_{\ell tc}}\right]\times$$
$$\times\prod_{m}\prod_{t}\left[\frac{\Gamma(\phi_t)}{\prod_c\Gamma(\phi_t\gamma_{tc})}\prod_{c}(\eta_{tc}^m)^{\phi_t\gamma_{tc}-1}\right]\prod_{m}\prod_{t}\frac{b_t^{a_t}}{\Gamma(a_t)}(\beta_t^m)^{a_t-1}\exp(-b_t\beta_t^m). \tag{4}$$

and such that

$$p(\boldsymbol{\eta},\boldsymbol{\beta}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi}) = \int p(\boldsymbol{\eta},\boldsymbol{\beta},\boldsymbol{\nu},\boldsymbol{s}|\boldsymbol{Y},\boldsymbol{z},\boldsymbol{\pi})d\boldsymbol{\nu}d\boldsymbol{s}.$$

From Equation 4 it is straightforward to compute the necessary conditional distributions to implement the Gibbs sampler (See Section 3.2).

References

Amazon. Amazon mechanical turk. `http://www.mturk.com`, 2005.

C. E. Antoniak. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

D. Blackwell and J. B. Macqueen. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society*, 28(1):pp. 20–28, 1979.

M. D. Escobar. Estimating normal means with a dirichlet process prior. *Journal of the American Statistical Association*, 89(425):pp. 268–277, 1994.

S. Favaro and Y. W. Teh. Mcmc for normalized random measure mixture models. *Preprint*, 2012.

T. S. Ferguson. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1 (2):209–230, 1973.

C. Fraly and A. E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Z. Ghahramani and H-C. Kim. Bayesian classifier combination. *Technical Report*, 2003.

P. Groot, A. Birlutiu, and T. Heskes. Learning from multiple annotators with gaussian processes. In *International Conference on Artificial Neural Networks*, pages 159–164. Springer-Verlag, 2011.

H. Ishwaran and L. F. James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):pp. 161–173, 2001.

H. Kajino, Y. Tsubo, and H. Kashima. Clustering crowds. In *Association for the Advancement of Artificial Intelligence*, pages 1120–1127, 2013.

M. Kalli, J. E. Griffin, and S. G. Walker. Slice sampling mixture models. *Statistics and Computing*, 21(1):93–105, January 2011.

H-C. Kim and Z. Ghahramani. Bayesian classifier combination. *Journal of Machine Learning Research*, 22:619–627, 2012.

C. Lintott, K. Schawinski, S. Bamford, A. Slosar, K. Land, D. Thomas, E. Edmondson, K. Masters, R. Nichol, J. Raddick, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy zoo 1 : data release of morphological classifications for nearly 900,000 galaxies. 2010.

S. N. MacEachern and P. Müller. Estimating mixture of dirichlet process models. *Journal of Computational and Graphical Statistics*, 7(2):pp. 223–238, 1998.

J. W. Miller and M. T. Harrison. A simple example of dirichlet process mixture inconsistency for the number of components. In *Neural Information Processing Systems*, 2013.

R. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

J. Pitman. Combinatorial stochastic processes. Technical Report 621, Department of Statistics, U.C. Berkeley, 2002. Lecture notes for St. Flour course.

V. C. Raykar and S. Yu. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, pages 491–518, 2012.

V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010.

D. Rindskopf and W. Rindskopf. The value of latent class analysis in medical diagnosis. *Statistics in Medicine*, 5(1):21–27, 1986.

B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1-3):157–173, 2008.

A. I. Schein, A. Popescul, L. H. Ungar, and D. M. Pennock. Methods and metrics for cold-start recommendations. In *Special Interest Group on Information Retrieval*, Special Interest Group on Information Retrieval, pages 253–260, New York, NY, USA, 2002.

E. Simpson, S. J. Roberts, A. Smith, and C. Lintott. Bayesian combination of multiple, imperfect classifiers. In *Neural Information Processing Systems*, pages 1–8, Oxford, 2011. University of Oxford.

E. Simpson, S. Roberts, I. Psorakis, and A. Smith. Dynamic bayesian combination of multiple imperfect classifiers. In *Decision making and imperfection*, volume 474 of *Studies in Computational Intelligence*, pages 1–35. 2013.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):111–147, 1974.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101, 2003.

Y. Tian and J. Zhu. Learning from crowds in the presence of schools of thought. In *Knowledge Discovery and Data Mining*, pages 226–234, 2012.

S. G. Walker. Sampling the dirichlet mixture model with slices. *Communications in Statistics - Simulation and Computation*, 2007.

F. L. Wauthier and M. I. Jordan. Bayesian bias mitigation for crowdsourcing. In *Neural Information Processing Systems*, pages 1800–1808, 2011.

P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In *Neural Information Processing Systems*, pages 2424–2432, 2010.

Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, and J. Dy. Modeling annotator expertise: learning when everybody knows a bit of something. *International Conference on Artificial Intelligence and Statistics*, 2010.

M.A. Young, R. Abrams, M.A. Taylor, and H.Y. Meltzer. Establishing diagnostic criteria for mania. *The Journal of Nervous and Mental Disease*, 171(11):676–682, 1983.

P. Zhang and Z. Obradovic. Learning from inconsistent and unreliable annotators by a gaussian mixture model and bayesian information criterion. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 553–568, 2011.

Journal of Machine Learning Research 16 (2015) 1629-1676 Submitted 7/13; Revised 6/14; Published 8/15

Approximate Modified Policy Iteration and its Application to the Game of Tetris

Bruno Scherrer Bruno.Scherrer@inria.fr
INRIA Nancy - Grand Est, Team Maia,
615 rue du Jardin Botanique,
54600 Vandœuvre-ls-Nancy, France

Mohammad Ghavamzadeh Mohammad.Ghavamzadeh@inria.fr
Adobe Research & INRIA Lille
321 Park Avenue
San Jose, CA 95110, USA

Victor Gabillon Victor.Gabillon@inria.fr
INRIA Lille - Nord Europe, Team SequeL,
40 avenue Halley,
59650 Villeneuve d'Ascq, France

Boris Lesner lesnerboris@gmail.com
INRIA Nancy - Grand Est, Team Maia,
615 rue du Jardin Botanique,
54600 Vandœuvre-ls-Nancy, France

Matthieu Geist matthieu.geist@centralesupelec.fr
CentraleSupélec, IMS-MaLIS Research Group & UMI 2958 (GeorgiaTech-CNRS),
2 rue Edouard Belin,
57070 Metz, France

Editor: Shie Mannor

Abstract

Modified policy iteration (MPI) is a dynamic programming (DP) algorithm that contains the two celebrated policy and value iteration methods. Despite its generality, MPI has not been thoroughly studied, especially its approximation form which is used when the state and/or action spaces are large or infinite. In this paper, we propose three implementations of approximate MPI (AMPI) that are extensions of the well-known approximate DP algorithms: fitted-value iteration, fitted-Q iteration, and classification-based policy iteration. We provide error propagation analysis that unify those for approximate policy and value iteration. We develop the finite-sample analysis of these algorithms, which highlights the influence of their parameters. In the classification-based version of the algorithm (CBMPI), the analysis shows that MPI's main parameter controls the balance between the estimation error of the classifier and the overall value function approximation. We illustrate and evaluate the behavior of these new algorithms in the Mountain Car and Tetris problems. Remarkably, in Tetris, CBMPI outperforms the existing DP approaches by a large margin, and competes with the current state-of-the-art methods while using fewer samples.[1]

1. This paper is a significant extension of two conference papers by the authors (Scherrer et al., 2012; Gabillon et al., 2013). Here we discuss better the relation of the AMPI algorithms with other approximate DP methods, and provide more detailed description of the algorithms, proofs of the theorems, and report

Keywords: approximate dynamic programming, reinforcement learning, Markov decision processes, finite-sample analysis, performance bounds, game of tetris

1. Introduction

Modified Policy Iteration (MPI) (Puterman, 1994, Chapter 6, and references therein for a detailed historical account) is an iterative algorithm to compute the optimal policy and value function of a Markov Decision Process (MDP). Starting from an arbitrary value function v_0, it generates a sequence of value-policy pairs

$$\pi_{k+1} = \mathcal{G}\, v_k \qquad \text{(greedy step)} \tag{1}$$

$$v_{k+1} = (T_{\pi_{k+1}})^m v_k \qquad \text{(evaluation step)} \tag{2}$$

where $\mathcal{G}\, v_k$ is a *greedy* policy w.r.t. (with respect to) v_k, T_{π_k} is the Bellman operator associated to the policy π_k, and $m \geq 1$ is a parameter. MPI generalizes the well-known dynamic programming algorithms: Value Iteration (VI) and Policy Iteration (PI) for the values $m = 1$ and $m = \infty$, respectively. MPI has less computation per iteration than PI (in a way similar to VI), while enjoys the faster convergence (in terms of the number of iterations) of the PI algorithm (Puterman, 1994). In problems with large state and/or action spaces, approximate versions of VI (AVI) and PI (API) have been the focus of a rich literature (see *e.g.*, Bertsekas and Tsitsiklis 1996; Szepesvári 2010). Approximate VI (AVI) generates the next value function as the approximation of the application of the Bellman optimality operator to the current value (Singh and Yee, 1994; Gordon, 1995; Bertsekas and Tsitsiklis, 1996; Munos, 2007; Ernst et al., 2005; Antos et al., 2007; Munos and Szepesvári, 2008). On the other hand, approximate PI (API) first finds an approximation of the value of the current policy and then generates the next policy as greedy w.r.t. this approximation (Bertsekas and Tsitsiklis, 1996; Munos, 2003; Lagoudakis and Parr, 2003a; Lazaric et al., 2010b, 2012). Another related algorithm is λ-policy iteration (Bertsekas and Ioffe, 1996), which is a rather complicated variation of MPI. It involves computing a fixed-point at each iteration, and thus, suffers from some of the drawbacks of the PI algorithms. This algorithm has been analyzed in its approximate form by Thiery and Scherrer (2010a); Scherrer (2013). The aim of this paper is to show that, similarly to its exact form, approximate MPI (AMPI) may represent an interesting alternative to AVI and API algorithms.

In this paper, we propose three implementations of AMPI (Section 3) that generalize the AVI implementations of Ernst et al. (2005); Antos et al. (2007); Munos and Szepesvári (2008) and the classification-based API algorithms of Lagoudakis and Parr (2003b); Fern et al. (2006); Lazaric et al. (2010c); Gabillon et al. (2011). We then provide an error propagation analysis of AMPI (Section 4), which shows how the L_p-norm of its performance loss

$$\ell_k = v_{\pi_*} - v_{\pi_k}$$

of using the policy π_k computed at some iteration k instead of the optimal policy π_* can be controlled through the errors at each iteration of the algorithm. We show that the error

of the experimental results, especially in the game of Tetris. Moreover, we report new results in the game Tetris that were obtained after the publication of our paper on this topic (Gabillon et al., 2013).

propagation analysis of AMPI is more involved than that of AVI and API. This is due to the fact that neither the contraction nor monotonicity arguments, that the error propagation analysis of these two algorithms rely on, hold for AMPI. The analysis of this section unifies those for AVI and API and is applied to the AMPI implementations presented in Section 3. We then detail the analysis of the three algorithms of Section 3 by providing their finite-sample analysis in Section 5. Interestingly, for the classification-based implementation of MPI (CBMPI), our analysis indicates that the parameter m allows us to balance the estimation error of the classifier with the overall quality of the value approximation. Finally, we evaluate the proposed algorithms and compare them with several existing methods in the Mountain Car and Tetris problems in Section 6. The game of Tetris is particularly challenging as the DP methods that are only based on approximating the value function have performed poorly in this domain. An important contribution of this work is to show that the classification-based AMPI algorithm (CBMPI) outperforms the existing DP approaches by a large margin, and competes with the current state-of-the-art methods while using fewer samples.

2. Background

We consider a discounted MDP $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is a finite action space, $P(ds'|s,a)$, for all state-action pairs (s,a), is a probability kernel on $\mathcal{S}$, the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is bounded by $R_{\max}$, and $\gamma \in (0,1)$ is a discount factor. A deterministic stationary policy (for short thereafter: a policy) is defined as a mapping $\pi : \mathcal{S} \to \mathcal{A}$. For a policy π, we may write $r_\pi(s) = r\big(s, \pi(s)\big)$ and $P_\pi(ds'|s) = P\big(ds'|s, \pi(s)\big)$. The value of the policy π in a state s is defined as the expected discounted sum of rewards received by starting at state s and then following the policy π, i.e.,

$$v_\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_\pi(s_t) \mid s_0 = s, s_{t+1} \sim P_\pi(\cdot|s_t)\right].$$

Similarly, the action-value function of a policy π at a state-action pair (s,a), $Q_\pi(s,a)$, is the expected discounted sum of rewards received by starting at state s, taking action a, and then following the policy π, i.e.,

$$Q_\pi(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) \mid s_0 = s, a_0 = a, s_{t+1} \sim P(\cdot|s_t,a_t), a_{t+1} = \pi(s_{t+1})\right].$$

Since the rewards are bounded by $R_{\max}$, the values and action-values are bounded by $V_{\max} = Q_{\max} = R_{\max}/(1-\gamma)$.

For any distribution μ on $\mathcal{S}$, μP_π is a distribution given by $(\mu P_\pi)(ds') = \int P_\pi(ds'|ds)\mu(ds)$. For any integrable function v on $\mathcal{S}$, $P_\pi v$ is a function defined as $(P_\pi v)(s) = \int v(s') P_\pi(ds'|s)$. The product of two kernels is naturally defined as $(P_{\pi'} P_\pi)(ds''|s) = \int P_{\pi'}(ds''|s') P_\pi(ds'|s)$. In analogy with the discrete space case, we write $(I - \gamma P_\pi)^{-1}$ to denote the kernel that is defined as $\sum_{t=0}^{\infty} (\gamma P_\pi)^t$.

The Bellman operator T_π of policy π takes an integrable function f on $\mathcal{S}$ as input and returns the function $T_\pi f$ defined as

$$\forall s \in \mathcal{S}, \quad [T_\pi f](s) = \mathbb{E}\big[r_\pi(s) + \gamma f(s') \mid s' \sim P_\pi(.|s)\big],$$

or in compact form, $T_\pi f = r_\pi + \gamma P_\pi f$. It is known that $v_\pi = (I - \gamma P_\pi)^{-1} r_\pi$ is the unique fixed-point of T_π. Given an integrable function f on $\mathcal{S}$, we say that a policy π is greedy w.r.t. f, and write $\pi = \mathcal{G}\, f$, if

$$\forall s \in \mathcal{S}, \quad [T_\pi f](s) = \max_a [T_a f](s),$$

or equivalently $T_\pi f = \max_{\pi'} [T_{\pi'} f]$. We denote by v_* the optimal value function. It is also known that v_* is the unique fixed-point of the Bellman optimality operator $T : v \to \max_\pi T_\pi v = T_{\mathcal{G}(v)} v$, and that a policy π_* that is greedy w.r.t. v_* is optimal and its value satisfies $v_{\pi_*} = v_*$.

We now define the concentrability coefficients (Munos, 2003, 2007; Munos and Szepesvári, 2008; Farahmand et al., 2010; Scherrer, 2013) that measure the stochasticity of an MDP, and will later appear in our analysis. For any integrable function $f : \mathcal{S} \to \mathbb{R}$ and any distribution μ on $\mathcal{S}$, the μ-weighted L_p norm of f is defined as

$$\|f\|_{p,\mu} \triangleq \left[\int |f(x)|^p \mu(dx) \right]^{1/p}.$$

Given some distributions μ and ρ that will be clear in the context of the paper, for all integers j and q, we shall consider the following Radon-Nikodym derivative based quantities

$$c_q(j) \triangleq \max_{\pi_1, \ldots, \pi_j} \left\| \frac{d(\rho P_{\pi_1} P_{\pi_2} \cdots P_{\pi_j})}{d\mu} \right\|_{q,\mu}, \tag{3}$$

where $\pi_1, \ldots, \pi_j$ is any set of policies defined in the MDP, and with the understanding that if $\rho P_{\pi_1} P_{\pi_2} \cdots P_{\pi_j}$ is not absolutely continuous with respect to μ, then we take $c_q(j) = \infty$. These coefficients measure the mismatch between some reference measure μ and the distribution $\rho P_{\pi_1} P_{\pi_2} \cdots P_{\pi_j}$ obtained by starting the process from distribution ρ and then making j steps according to π_1, π_2, ... π_j, respectively. Since the bounds we shall derive will be based on these coefficients, they will be informative only if these coefficients are finite. We refer the reader to Munos (2007); Munos and Szepesvári (2008); Farahmand et al. (2010) for more discussion on this topic. In particular, the interested reader may find a simple MDP example for which these coefficients are reasonably small in Munos (2007, Section 5.5 and 7).

3. Approximate MPI Algorithms

In this section, we describe three approximate MPI (AMPI) algorithms. These algorithms rely on a function space $\mathcal{F}$ to approximate value functions, and in the third algorithm, also on a policy space Π to represent greedy policies. In what follows, we describe the iteration k of these iterative algorithms.

3.1 AMPI-V

The first and most natural AMPI algorithm presented in the paper, called AMPI-V, is described in Figure 1. In AMPI-V, we assume that the values v_k are represented in a

function space $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{S}}$. In any state s, the action $\pi_{k+1}(s)$ that is greedy w.r.t. v_k can be estimated as follows:

$$\pi_{k+1}(s) \in \arg\max_{a \in \mathcal{A}} \frac{1}{M} \sum_{j=1}^{M} \big(\widehat{T}_a^{(j)} v_k\big)(s), \tag{4}$$

$$\text{with} \qquad \big(\widehat{T}_a^{(j)} v_k\big)(s) = r_a^{(j)} + \gamma v_k(s_a^{(j)}),$$

where for all $a \in \mathcal{A}$ and $1 \leq j \leq M$, $r_a^{(j)}$ and $s_a^{(j)}$ are samples of rewards and next states when action a is taken in state s. Thus, approximating the greedy action in a state s requires $M|\mathcal{A}|$ samples. The algorithm works as follows. We sample N states from a distribution μ on $\mathcal{S}$, and build a rollout set $\mathcal{D}_k = \{s^{(i)}\}_{i=1}^{N}$, $s^{(i)} \sim \mu$. We denote by $\widehat{\mu}$ the empirical distribution corresponding to μ. From each state $s^{(i)} \in \mathcal{D}_k$, we generate a rollout of size m, *i.e.*, $\big(s^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}, \ldots, a_{m-1}^{(i)}, r_{m-1}^{(i)}, s_m^{(i)}\big)$, where $a_t^{(i)}$ is the action suggested by π_{k+1} in state $s_t^{(i)}$, computed using Equation 4, and $r_t^{(i)}$ and $s_{t+1}^{(i)}$ are sampled reward and next state induced by this choice of action. For each $s^{(i)}$, we then compute a rollout estimate

$$\widehat{v}_{k+1}(s^{(i)}) = \sum_{t=0}^{m-1} \gamma^t r_t^{(i)} + \gamma^m v_k(s_m^{(i)}), \tag{5}$$

which is an unbiased estimate of $\big[(T_{\pi_{k+1}})^m v_k\big](s^{(i)})$. Finally, v_{k+1} is computed as the best fit in $\mathcal{F}$ to these estimates, *i.e.*, it is a function $v \in \mathcal{F}$ that minimizes the empirical error

$$\widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; v) = \frac{1}{N} \sum_{i=1}^{N} \big(\widehat{v}_{k+1}(s^{(i)}) - v(s^{(i)})\big)^2, \tag{6}$$

with the goal of minimizing the true error

$$\mathcal{L}_k^{\mathcal{F}}(\mu; v) = \Big|\Big|\big[(T_{\pi_{k+1}})^m v_k\big] - v\Big|\Big|_{2,\mu}^2 = \int \Big(\big[(T_{\pi_{k+1}})^m v_k\big](s) - v(s)\Big)^2 \mu(ds).$$

Each iteration of AMPI-V requires N rollouts of size m, and in each rollout, each of the $|\mathcal{A}|$ actions needs M samples to compute Equation 4. This gives a total of $Nm(M|\mathcal{A}| + 1)$ transition samples. Note that the fitted value iteration algorithm (Munos and Szepesvári, 2008) is a special case of AMPI-V when $m = 1$.

3.2 AMPI-Q

In AMPI-Q, we replace the value function $v : \mathcal{S} \to \mathbb{R}$ with the action-value function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. Figure 2 contains the pseudocode of this algorithm. The Bellman operator for a policy π at a state-action pair (s, a) can then be written as

$$[T_\pi Q](s, a) = \mathbb{E}\big[r(s, a) + \gamma Q(s', \pi(s')) \mid s' \sim P(\cdot|s, a)\big],$$

and the greedy operator is defined as

$$\pi \in \mathcal{G}\, Q \quad \Longleftrightarrow \quad \forall s, \quad \pi(s) = \arg\max_{a \in \mathcal{A}} Q(s, a).$$

Input: Value function space $\mathcal{F}$, state distribution μ
Initialize: Let $v_0 \in \mathcal{F}$ be an arbitrary value function
for $k = 0, 1, \ldots$ **do**
- ***Perform rollouts:***

Construct the rollout set $\mathcal{D}_k = \{s^{(i)}\}_{i=1}^N,\ s^{(i)} \overset{\text{iid}}{\sim} \mu$
for all states $s^{(i)} \in \mathcal{D}_k$ **do**
Perform a rollout (using Equation 4 for each action)
$\widehat{v}_{k+1}(s^{(i)}) = \sum_{t=0}^{m-1} \gamma^t r_t^{(i)} + \gamma^m v_k(s_m^{(i)})$
end for
- ***Approximate value function:***

$v_{k+1} \in \underset{v \in \mathcal{F}}{\operatorname{argmin}}\, \widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; v)$ **(regression)** (see Equation 6)
end for

Figure 1: The pseudo-code of the AMPI-V algorithm.

In AMPI-Q, action-value functions Q_k are represented in a function space $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{S}\times\mathcal{A}}$, and the greedy action w.r.t. Q_k at a state s, *i.e.*, $\pi_{k+1}(s)$, is computed as

$$\pi_{k+1}(s) \in \arg\max_{a \in \mathcal{A}} Q_k(s, a). \tag{7}$$

The *evaluation step* is similar to that of AMPI-V, with the difference that now we work with state-action pairs. We sample N state-action pairs from a distribution μ on $\mathcal{S} \times \mathcal{A}$ and build a rollout set $\mathcal{D}_k = \{(s^{(i)}, a^{(i)})\}_{i=1}^N,\ (s^{(i)}, a^{(i)}) \sim \mu$. We denote by $\widehat{\mu}$ the empirical distribution corresponding to μ. For each $(s^{(i)}, a^{(i)}) \in \mathcal{D}_k$, we generate a rollout of size m, *i.e.*, $\big(s^{(i)}, a^{(i)}, r_0^{(i)}, s_1^{(i)}, a_1^{(i)}, \cdots, s_m^{(i)}, a_m^{(i)}\big)$, where the first action is $a^{(i)}$, $a_t^{(i)}$ for $t \geq 1$ is the action suggested by π_{k+1} in state $s_t^{(i)}$ computed using Equation 7, and $r_t^{(i)}$ and $s_{t+1}^{(i)}$ are sampled reward and next state induced by this choice of action. For each $(s^{(i)}, a^{(i)}) \in \mathcal{D}_k$, we then compute the rollout estimate

$$\widehat{Q}_{k+1}(s^{(i)}, a^{(i)}) = \sum_{t=0}^{m-1} \gamma^t r_t^{(i)} + \gamma^m Q_k(s_m^{(i)}, a_m^{(i)}),$$

which is an unbiased estimate of $\big[(T_{\pi_{k+1}})^m Q_k\big](s^{(i)}, a^{(i)})$. Finally, Q_{k+1} is the best fit to these estimates in $\mathcal{F}$, *i.e.*, it is a function $Q \in \mathcal{F}$ that minimizes the empirical error

$$\widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; Q) = \frac{1}{N} \sum_{i=1}^N \big(\widehat{Q}_{k+1}(s^{(i)}, a^{(i)}) - Q(s^{(i)}, a^{(i)})\big)^2, \tag{8}$$

with the goal of minimizing the true error

$$\mathcal{L}_k^{\mathcal{F}}(\mu; Q) = \Big|\Big|\big[(T_{\pi_{k+1}})^m Q_k\big] - Q\Big|\Big|_{2,\mu}^2 = \int \Big(\big[(T_{\pi_{k+1}})^m Q_k\big](s, a) - Q(s, a)\Big)^2 \mu(dsda).$$

Each iteration of AMPI-Q requires Nm samples, which is less than that for AMPI-V. However, it uses a hypothesis space on state-action pairs instead of states (a larger space than that used by AMPI-V). Note that the fitted-Q iteration algorithm (Ernst et al., 2005; Antos et al., 2007) is a special case of AMPI-Q when $m = 1$.

Input: Value function space $\mathcal{F}$, state distribution μ
Initialize: Let $Q_0 \in \mathcal{F}$ be an arbitrary value function
for $k = 0, 1, \ldots$ **do**
- ***Perform rollouts:***

Construct the rollout set $\mathcal{D}_k = \{(s^{(i)}, a^{(i)}\}_{i=1}^{N},\ (s^{(i)}, a^{(i)}) \overset{\text{iid}}{\sim} \mu$
for all states $(s^{(i)}, a^{(i)}) \in \mathcal{D}_k$ **do**
Perform a rollout (using Equation 7 for each action)
$\widehat{Q}_{k+1}(s^{(i)}, a^{(i)}) = \sum_{t=0}^{m-1} \gamma^t r_t^{(i)} + \gamma^m Q_k(s_m^{(i)}, a_m^{(i)})$,
end for
- ***Approximate action-value function:***

$Q_{k+1} \in \underset{Q \in \mathcal{F}}{\operatorname{argmin}}\, \widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; Q)$ **(regression)** (see Equation 8)
end for

Figure 2: The pseudo-code of the AMPI-Q algorithm.

3.3 Classification-Based MPI

The third AMPI algorithm presented in this paper, called classification-based MPI (CBMPI), uses an explicit representation for the policies π_k, in addition to the one used for the value functions v_k. The idea is similar to the classification-based PI algorithms (Lagoudakis and Parr, 2003b; Fern et al., 2006; Lazaric et al., 2010c; Gabillon et al., 2011) in which we search for the greedy policy in a policy space Π (defined by a classifier) instead of computing it from the estimated value or action-value function (similar to AMPI-V and AMPI-Q).
In order to describe CBMPI, we first rewrite the MPI formulation (Equations 1 and 2) as

$$v_k = (T_{\pi_k})^m v_{k-1} \qquad \text{(evaluation step)} \tag{9}$$

$$\pi_{k+1} = \mathcal{G}\left[(T_{\pi_k})^m v_{k-1}\right] \qquad \text{(greedy step)} \tag{10}$$

Note that in this equivalent formulation both v_k and π_{k+1} are functions of $(T_{\pi_k})^m v_{k-1}$. CBMPI is an approximate version of this new formulation. As described in Figure 3, CBMPI begins with arbitrary initial policy $\pi_1 \in \Pi$ and value function $v_0 \in \mathcal{F}$.[2] At each iteration k, a new value function v_k is built as the best approximation of the m-step Bellman operator $(T_{\pi_k})^m v_{k-1}$ in $\mathcal{F}$ (*evaluation step*). This is done by solving a regression problem whose target function is $(T_{\pi_k})^m v_{k-1}$. To set up the regression problem, we build a rollout set $\mathcal{D}_k$ by sampling N states i.i.d. from a distribution μ.[3] We denote by $\widehat{\mu}$ the empirical distribution corresponding to μ. For each state $s^{(i)} \in \mathcal{D}_k$, we generate a rollout $\left(s^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}, \ldots, a_{m-1}^{(i)}, r_{m-1}^{(i)}, s_m^{(i)}\right)$ of size m, where $a_t^{(i)} = \pi_k(s_t^{(i)})$, and $r_t^{(i)}$ and $s_{t+1}^{(i)}$ are sampled reward and next state induced by this choice of action. From this rollout, we compute an unbiased estimate $\widehat{v}_k(s^{(i)})$ of $\left[(T_{\pi_k})^m v_{k-1}\right](s^{(i)})$ as in Equation 5:

$$\widehat{v}_k(s^{(i)}) = \sum_{t=0}^{m-1} \gamma^t r_t^{(i)} + \gamma^m v_{k-1}(s_m^{(i)}), \tag{11}$$

2. Note that the function space $\mathcal{F}$ and policy space Π are automatically defined by the choice of the regressor and classifier, respectively.
3. Here we used the same sampling distribution μ for both regressor and classifier, but in general different distributions may be used for these two components of the algorithm.

and use it to build a training set $\left\{\left(s^{(i)}, \widehat{v}_k(s^{(i)})\right)\right\}_{i=1}^{N}$. This training set is then used by the regressor to compute v_k as an estimate of $(T_{\pi_k})^m v_{k-1}$. Similar to the AMPI-V algorithm, the regressor here finds a function $v \in \mathcal{F}$ that minimizes the empirical error

$$\widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; v) = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{v}_k(s^{(i)}) - v(s^{(i)})\right)^2, \tag{12}$$

with the goal of minimizing the true error

$$\mathcal{L}_k^{\mathcal{F}}(\mu; v) = \left|\left|\left[(T_{\pi_k})^m v_{k-1}\right] - v\right|\right|_{2,\mu}^2 = \int\left(\left[(T_{\pi_k})^m v_{k-1}\right](s) - v(s)\right)^2 \mu(ds).$$

The *greedy step* at iteration k computes the policy π_{k+1} as the best approximation of $\mathcal{G}\left[(T_{\pi_k})^m v_{k-1}\right]$ by solving a cost-sensitive classification problem. From the definition of a greedy policy, if $\pi = \mathcal{G}\left[(T_{\pi_k})^m v_{k-1}\right]$, for each $s \in \mathcal{S}$, we have

$$\left[T_\pi (T_{\pi_k})^m v_{k-1}\right](s) = \max_{a\in\mathcal{A}}\left[T_a (T_{\pi_k})^m v_{k-1}\right](s). \tag{13}$$

By defining $Q_k(s,a) = \left[T_a (T_{\pi_k})^m v_{k-1}\right](s)$, we may rewrite Equation 13 as

$$Q_k\big(s, \pi(s)\big) = \max_{a\in\mathcal{A}} Q_k(s,a). \tag{14}$$

The cost-sensitive error function used by CBMPI is of the form

$$\mathcal{L}^{\Pi}_{\pi_k, v_{k-1}}(\mu; \pi) = \int\Big[\max_{a\in\mathcal{A}} Q_k(s,a) - Q_k\big(s, \pi(s)\big)\Big]\mu(ds). \tag{15}$$

To simplify the notation we use $\mathcal{L}_k^{\Pi}$ instead of $\mathcal{L}^{\Pi}_{\pi_k, v_{k-1}}$. To set up this cost-sensitive classification problem, we build a rollout set $\mathcal{D}'_k$ by sampling N' states i.i.d. from a distribution μ. For each state $s^{(i)} \in \mathcal{D}'_k$ and each action $a \in \mathcal{A}$, we build M independent rollouts of size $m+1$, *i.e.*,[4]

$$\left(s^{(i)}, a, r_0^{(i,j)}, s_1^{(i,j)}, a_1^{(i,j)}, \ldots, a_m^{(i,j)}, r_m^{(i,j)}, s_{m+1}^{(i,j)}\right)_{j=1}^{M},$$

where for $t \geq 1$, $a_t^{(i,j)} = \pi_k(s_t^{(i,j)})$, and $r_t^{(i,j)}$ and $s_{t+1}^{(i,j)}$ are sampled reward and next state induced by this choice of action. From these rollouts, we compute an unbiased estimate of $Q_k(s^{(i)}, a)$ as $\widehat{Q}_k(s^{(i)}, a) = \frac{1}{M}\sum_{j=1}^{M} R_k^j(s^{(i)}, a)$ where

$$R_k^j(s^{(i)}, a) = \sum_{t=0}^{m}\gamma^t r_t^{(i,j)} + \gamma^{m+1} v_{k-1}(s_{m+1}^{(i,j)}). \tag{16}$$

Given the outcome of the rollouts, CBMPI uses a cost-sensitive classifier to return a policy π_{k+1} that minimizes the following *empirical error*

$$\widehat{\mathcal{L}}_k^{\Pi}(\widehat{\mu}; \pi) = \frac{1}{N'}\sum_{i=1}^{N'}\Big[\max_{a\in\mathcal{A}}\widehat{Q}_k(s^{(i)}, a) - \widehat{Q}_k\big(s^{(i)}, \pi(s^{(i)})\big)\Big], \tag{17}$$

4. In practice, one may implement CBMPI in more sample-efficient way by reusing the rollouts generated for the greedy step in the evaluation step, but we do not consider this here because it makes the forthcoming analysis more complicated.

Input: Value function space $\mathcal{F}$, policy space Π, state distribution μ
Initialize: Let $\pi_1 \in \Pi$ be an arbitrary policy and $v_0 \in \mathcal{F}$ an arbitrary value function
for $k = 1, 2, \ldots$ **do**
- ***Perform rollouts:***

Construct the rollout set $\mathcal{D}_k = \{s^{(i)}\}_{i=1}^{N},\ s^{(i)} \overset{\text{iid}}{\sim} \mu$
for all states $s^{(i)} \in \mathcal{D}_k$ **do**
Perform a rollout and return $\widehat{v}_k(s^{(i)})$ (using Equation 11)
end for
Construct the rollout set $\mathcal{D}'_k = \{s^{(i)}\}_{i=1}^{N'},\ s^{(i)} \overset{\text{iid}}{\sim} \mu$
for all states $s^{(i)} \in \mathcal{D}'_k$ and actions $a \in \mathcal{A}$ **do**
for $j = 1$ to M **do**
Perform a rollout and return $R_k^j(s^{(i)}, a)$ (using Equation 16)
end for
$\widehat{Q}_k(s^{(i)}, a) = \frac{1}{M} \sum_{j=1}^{M} R_k^j(s^{(i)}, a)$
end for
- ***Approximate value function:***

$v_k \in \underset{v \in \mathcal{F}}{\operatorname{argmin}}\, \widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; v)$ **(regression)** (see Equation 12)
- ***Approximate greedy policy:***

$\pi_{k+1} \in \underset{\pi \in \Pi}{\operatorname{argmin}}\, \widehat{\mathcal{L}}_k^{\Pi}(\widehat{\mu}; \pi)$ **(classification)** (see Equation 17)

end for

Figure 3: The pseudo-code of the CBMPI algorithm.

with the goal of minimizing the true error $\mathcal{L}_k^{\Pi}(\mu; \pi)$ defined by Equation 15.

Each iteration of CBMPI requires $Nm + M|\mathcal{A}|N'(m+1)$ (or $M|\mathcal{A}|N'(m+1)$ in case we reuse the rollouts, see Footnote 4) transition samples. Note that when m tends to ∞, we recover the DPI algorithm proposed and analyzed by Lazaric et al. (2010c).

3.4 Possible Approaches to Reuse the Samples

In all the proposed AMPI algorithms, we generate fresh samples for the rollouts, and even for the starting states, at each iteration. This may result in relatively high sample complexity for these algorithms. In this section, we propose two possible approaches to circumvent this problem and to keep the number of samples independent of the number of iterations.

One approach would be to use a fixed set of starting samples $(s^{(i)})$ or $(s^{(i)}, a^{(i)})$ for all iterations, and think of a tree of depth m that contains all the possible outcomes of m-steps choices of actions (this tree contains $|A|^m$ leaves). Using this tree, all the trajectories with the same actions share the same samples. In practice, it is not necessarily to build the entire depth m tree, it is only needed to add a branch when the desired action does not belong to the tree. Using this approach, that is reminiscent of the work by Kearns et al. (2000), the sample complexity of the algorithm no longer depends on the number of iterations. For example, we may only need $NM|A|^m$ transitions for the CBMPI algorithm.

We may also consider the case where we do not have access to a generative model of the system, and all we have is a set of trajectories of size m generated by a behavior policy π_b that is assumed to choose all actions a in each state s with a positive probability (*i.e.*, $\pi_b(a|s) >$

$0,\ \forall s, \forall a$) (Precup et al., 2000, 2001; Geist and Scherrer, 2014). In this case, one may still compute an unbiased estimate of the application of $(T_\pi)^m$ operator to value and action-value functions. For instance, given a m-step sample trajectory $(s, a_0, r_0, s_1, \ldots, s_m, a_m)$ generated by π_b, an unbiased estimate of $[(T_\pi)^m v](s)$ may be computed as (assuming that the distribution μ has the following factored form $p(s, a_0|\mu) = p(s)\pi_b(a_0|s)$ at state s)

$$y = \sum_{t=0}^{m-1} \alpha_t \gamma^t r_t + \alpha_m \gamma^m v(s_m), \qquad \text{where} \qquad \alpha_t = \prod_{j=1}^{t} \frac{1_{a_j=\pi(s_j)}}{\pi_b(a_j|s_j)}$$

is an importance sampling correction factor that can be computed along the trajectory. Note that this process may increase the variance of such an estimate, and thus, requires many more samples to be accurate—the price to pay for the absence of a generative model.

4. Error Propagation

In this section, we derive a general formulation for propagation of errors through the iterations of an AMPI algorithm. The line of analysis for error propagation is different in VI and PI algorithms. VI analysis is based on the fact that this algorithm computes the fixed point of the Bellman optimality operator, and this operator is a γ-contraction in max-norm (Bertsekas and Tsitsiklis, 1996; Munos, 2007). On the other hand, it can be shown that the operator by which PI updates the value from one iteration to the next is not a contraction in max-norm in general. Unfortunately, we can show that the same property holds for MPI when it does not reduce to VI (*i.e.*, for $m > 1$).

Proposition 1 *If $m > 1$, there exists no norm for which the operator that MPI uses to update the values from one iteration to the next is a contraction.*

Proof We consider the MDP with two states $\{s_1, s_2\}$, two actions $\{change, stay\}$, rewards $r(s_1) = 0,\ r(s_2) = 1$, and transitions $P_{ch}(s_2|s_1) = P_{ch}(s_1|s_2) = P_{st}(s_1|s_1) = P_{st}(s_2|s_2) = 1$. Consider two value functions $v = (\epsilon, 0)$ and $v' = (0, \epsilon)$ with $\epsilon > 0$. Their corresponding greedy policies are $\pi = (st, ch)$ and $\pi' = (ch, st)$, and the next iterates of v and v' can be computed as $(T_\pi)^m v = \begin{pmatrix} \gamma^m \epsilon \\ 1 + \gamma^m \epsilon \end{pmatrix}$ and $(T_{\pi'})^m v' = \begin{pmatrix} \frac{\gamma - \gamma^m}{1-\gamma} + \gamma^m \epsilon \\ \frac{1-\gamma^m}{1-\gamma} + \gamma^m \epsilon \end{pmatrix}$. Thus, $(T_{\pi'})^m v' - (T_\pi)^m v = \begin{pmatrix} \frac{\gamma - \gamma^m}{1-\gamma} \\ \frac{\gamma - \gamma^m}{1-\gamma} \end{pmatrix}$, while $v' - v = \begin{pmatrix} -\epsilon \\ \epsilon \end{pmatrix}$. Since ϵ can be arbitrarily small, the norm of $(T_{\pi'})^m v' - (T_\pi)^m v$ can be arbitrarily larger than the norm of $v - v'$ as long as $m > 1$. ■

We also know that the analysis of PI usually relies on the fact that the sequence of the generated values is non-decreasing (Bertsekas and Tsitsiklis, 1996; Munos, 2003). Unfortunately, it can be easily shown that for m finite, the value functions generated by MPI may decrease (it suffices to take a very high initial value). It can be seen from what we just described and Proposition 1 that for $m \neq 1$ and ∞, MPI is neither contracting nor non-decreasing, and thus, a new proof is needed for the propagation of errors in this algorithm.

To study error propagation in AMPI, we introduce an abstract algorithmic model that accounts for potential errors. AMPI starts with an arbitrary value v_0 and at each iteration

$k \geq 1$ computes the greedy policy w.r.t. v_{k-1} with some error ϵ'_k, called the *greedy step error*. Thus, we write the new policy π_k as

$$\pi_k = \widehat{\mathcal{G}}_{\epsilon'_k} v_{k-1}. \tag{18}$$

Equation 18 means that for any policy π', we have $T_{\pi'} v_{k-1} \leq T_{\pi_k} v_{k-1} + \epsilon'_k$. AMPI then generates the new value function v_k with some error ϵ_k, called the *evaluation step error*

$$v_k = (T_{\pi_k})^m v_{k-1} + \epsilon_k. \tag{19}$$

Before showing how these two errors are propagated through the iterations of AMPI, let us first define them in the context of each of the algorithms presented in Section 3 separately.

AMPI-V: The term ϵ_k is the error when fitting the value function v_k. This error can be further decomposed into two parts: the one related to the approximation power of $\mathcal{F}$ and the one due to the finite number of samples/rollouts. The term ϵ'_k is the error due to using a finite number of samples M for estimating the greedy actions.

AMPI-Q: In this case $\epsilon'_k = 0$ and ϵ_k is the error in fitting the state-action value function Q_k.

CBMPI: This algorithm iterates as follows:

$$\begin{aligned} v_k &= (T_{\pi_k})^m v_{k-1} + \epsilon_k \\ \pi_{k+1} &= \widehat{\mathcal{G}}_{\epsilon'_{k+1}} \left[(T_{\pi_k})^m v_{k-1} \right]. \end{aligned}$$

Unfortunately, this does not exactly match the model described in Equations 18 and 19. By introducing the auxiliary variable $w_k \stackrel{\Delta}{=} (T_{\pi_k})^m v_{k-1}$, we have $v_k = w_k + \epsilon_k$, and thus, we may write

$$\pi_{k+1} = \widehat{\mathcal{G}}_{\epsilon'_{k+1}} \left[w_k \right]. \tag{20}$$

Using $v_{k-1} = w_{k-1} + \epsilon_{k-1}$, we have

$$w_k = (T_{\pi_k})^m v_{k-1} = (T_{\pi_k})^m (w_{k-1} + \epsilon_{k-1}) = (T_{\pi_k})^m w_{k-1} + (\gamma P_{\pi_k})^m \epsilon_{k-1}. \tag{21}$$

Now, Equations 20 and 21 exactly match Equations 18 and 19 by replacing v_k with w_k and ϵ_k with $(\gamma P_{\pi_k})^m \epsilon_{k-1}$.

The rest of this section is devoted to show how the errors ϵ_k and ϵ'_k propagate through the iterations of an AMPI algorithm. We only outline the main arguments that will lead to the performance bounds of Theorems 7 and 8 and report most technical details of the proof in Appendices A to C. To do this, we follow the line of analysis developed by Scherrer and Thiéry (2010), and consider the following three quantities:

1) The distance between the optimal value function and the value before approximation at the k^{th} iteration:

$$d_k \stackrel{\Delta}{=} v_* - (T_{\pi_k})^m v_{k-1} = v_* - (v_k - \epsilon_k).$$

2) The shift between the value before approximation and the value of the policy at the k^{th} iteration:

$$s_k \stackrel{\Delta}{=} (T_{\pi_k})^m v_{k-1} - v_{\pi_k} = (v_k - \epsilon_k) - v_{\pi_k}.$$

3) The (approximate) Bellman residual at the k^{th} iteration:

$$b_k \stackrel{\Delta}{=} v_k - T_{\pi_{k+1}} v_k.$$

We are interested in finding an upper bound on the **loss**

$$l_k \stackrel{\Delta}{=} v_* - v_{\pi_k} = d_k + s_k.$$

To do so, we will upper bound d_k and s_k, which requires a bound on the Bellman residual b_k. More precisely, the core of our analysis is to prove the following point-wise inequalities for our three quantities of interest.

Lemma 2 *Let $k \geq 1$, $x_k \stackrel{\Delta}{=} (I - \gamma P_{\pi_k})\epsilon_k + \epsilon'_{k+1}$ and $y_k \stackrel{\Delta}{=} -\gamma P_{\pi_*}\epsilon_k + \epsilon'_{k+1}$. We have:*

$$\begin{aligned} b_k &\leq (\gamma P_{\pi_k})^m b_{k-1} + x_k, \\ d_{k+1} &\leq \gamma P_{\pi_*} d_k + y_k + \sum_{j=1}^{m-1} (\gamma P_{\pi_{k+1}})^j b_k, \\ s_k &= (\gamma P_{\pi_k})^m (I - \gamma P_{\pi_k})^{-1} b_{k-1}. \end{aligned}$$

Proof See Appendix A. ■

Since the stochastic kernels are non-negative, the bounds in Lemma 2 indicate that the loss l_k will be bounded if the errors ϵ_k and ϵ'_k are controlled. In fact, if we define ϵ as a uniform upper-bound on the pointwise absolute value of the errors, $|\epsilon_k|$ and $|\epsilon'_k|$, the first inequality in Lemma 2 implies that $b_k \leq O(\epsilon)$, and as a result, the second and third inequalities gives us $d_k \leq O(\epsilon)$ and $s_k \leq O(\epsilon)$. This means that the loss will also satisfy $l_k \leq O(\epsilon)$.

Our bound for the loss l_k is the result of careful expansion and combination of the three inequalities in Lemma 2. Before we state this result, we introduce some notations that will ease our formulation and significantly simplify our proofs compared to those in the similar existing work (Munos, 2003, 2007; Scherrer, 2013).

Definition 3 *For a positive integer n, we define $\mathbb{P}_n$ as the smallest set of discounted transition kernels that are defined as follows:*

1) *for any set of n policies $\{\pi_1, \ldots, \pi_n\}$, $(\gamma P_{\pi_1})(\gamma P_{\pi_2}) \ldots (\gamma P_{\pi_n}) \in \mathbb{P}_n$,*

2) *for any $\alpha \in (0, 1)$ and $(P_1, P_2) \in \mathbb{P}_n \times \mathbb{P}_n$, $\alpha P_1 + (1 - \alpha) P_2 \in \mathbb{P}_n$.*

Furthermore, we use the somewhat abusive notation Γ^n for denoting any element of $\mathbb{P}_n$. For example, if we write a transition kernel P as $P = \alpha_1 \Gamma^i + \alpha_2 \Gamma^j \Gamma^k = \alpha_1 \Gamma^i + \alpha_2 \Gamma^{j+k}$, it should be read as: "there exist $P_1 \in \mathbb{P}_i$, $P_2 \in \mathbb{P}_j$, $P_3 \in \mathbb{P}_k$, and $P_4 \in \mathbb{P}_{k+j}$ such that $P = \alpha_1 P_1 + \alpha_2 P_2 P_3 = \alpha_1 P_1 + \alpha_2 P_4$."

Using the notation in Definition 3, we now derive a point-wise bound on the loss.

Lemma 4 *After k iterations, the losses of AMPI-V and AMPI-Q satisfy*

$$l_k \leq 2 \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon_{k-i}| + \sum_{i=0}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon'_{k-i}| + h(k),$$

while the loss of CBMPI satisfies

$$l_k \leq 2\sum_{i=1}^{k-2}\sum_{j=i+m}^{\infty}\Gamma^j|\epsilon_{k-i-1}| + \sum_{i=0}^{k-1}\sum_{j=i}^{\infty}\Gamma^j|\epsilon'_{k-i}| + h(k),$$

where $h(k) \triangleq 2\sum_{j=k}^{\infty}\Gamma^j|d_0|$ *or* $h(k) \triangleq 2\sum_{j=k}^{\infty}\Gamma^j|b_0|$.

Proof See Appendix B. ■

Remark 5 *A close look at the existing point-wise error bounds for AVI (Munos, 2007, Lemma 4.1) and API (Munos, 2003, Corollary 10) shows that they do not consider error in the greedy step (*i.e., $\epsilon'_k = 0$*) and have the following form:*

$$\limsup_{k\to\infty} l_k \leq 2\limsup_{k\to\infty}\sum_{i=1}^{k-1}\sum_{j=i}^{\infty}\Gamma^j|\epsilon_{k-i}|.$$

This indicates that the bound in Lemma 4 not only unifies the analysis of AVI and API, but it generalizes them to the case of error in the greedy step and to a finite number of iterations k. Moreover, our bound suggests that the way the errors are propagated in the whole family of algorithms, VI/PI/MPI, is independent of m at the level of abstraction suggested by Definition 3.[5]

An important immediate consequence of the point-wise bound of Lemma 4 is a simple guarantee on the performance of the algorithms. Let us define $\epsilon = \sup_{j\geq 1}\|\epsilon_j\|_\infty$ and $\epsilon' = \sup_{j\geq 1}\|\epsilon'_j\|_\infty$ as uniform bounds on the evaluation and greedy step errors. Now by taking the max-norm (using the fact that for all i, $\|\Gamma^i\|_\infty = \gamma^i$) and limsup when k tends to infinity, we obtain

$$\limsup_{k\to\infty}\|l_k\|_\infty \leq \frac{2\gamma\epsilon + \epsilon'}{(1-\gamma)^2}. \tag{22}$$

Such a bound is a generalization of the bounds for AVI ($m = 1$ and $\epsilon' = 0$) and API ($m = \infty$) in Bertsekas and Tsitsiklis (1996). This bound can be read as follows: if we can control the max-norm of the evaluation and greedy errors at all iterations, then we can control the loss of the policy returned by the algorithm w.r.t. the optimal policy. Conversely, another interpretation of the above bound is that errors should not be too big if we want to have a performance guarantee. Since the loss is always bounded by $2V_{\max}$, the bound stops to be informative as soon as $2\gamma\epsilon + \epsilon' > 2(1-\gamma)^2 V_{\max} = 2(1-\gamma)R_{\max}$.

Assume we use (max-norm) regression and classification for the evaluation and greedy steps. Then, the above result means that one can make a *reduction* from the RL problem to these regression and classification problems. Furthermore, if any significant breakthrough is made in the literature for these (more standard problems), the RL setting can automatically benefit from it. The error terms ϵ and ϵ' in the above bound are expressed in terms of the

5. Note however that the dependence on m will reappear if we make explicit what is hidden in Γ^j terms.

max-norm. Since most regressors and classifiers, including those we have described in the algorithms, control some weighted quadratic norm, the practical range of a result like Equation 22 is limited. The rest of this section addresses this specific issue, by developing a somewhat more complicated but more useful error analysis in L_p-norm.

We now turn the point-wise bound of Lemma 4 into a bound in weighted L_p-norm, which we recall, for any function $f : \mathcal{S} \to \mathbb{R}$ and any distribution μ on $\mathcal{S}$ is defined as $\|f\|_{p,\mu} \triangleq \left[\int |f(x)|^p \mu(dx)\right]^{1/p}$. Munos (2003, 2007); Munos and Szepesvári (2008), and the recent work of Farahmand et al. (2010), which provides the most refined bounds for API and AVI, show how to do this process through quantities, called *concentrability coefficients*. These coefficients use the Radon-Nikodym coefficients introduced in Section 2 and measure how a distribution over states may concentrate through the dynamics of the MDP. We now state a technical lemma that allows to convert componentwise bounds to L_p-norm bounds, and that generalizes the analysis of Farahmand et al. (2010) to a larger class of concentrability coefficients.

Lemma 6 *Let $\mathcal{I}$ and $(\mathcal{J}_i)_{i\in\mathcal{I}}$ be sets of non-negative integers, $\{\mathcal{I}_1, \ldots, \mathcal{I}_n\}$ be a partition of $\mathcal{I}$, and f and $(g_i)_{i\in\mathcal{I}}$ be functions satisfying*

$$|f| \leq \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{J}_i} \Gamma^j |g_i| = \sum_{l=1}^{n} \sum_{i\in\mathcal{I}_l} \sum_{j\in\mathcal{J}_i} \Gamma^j |g_i|.$$

Then for all p, q and q' such that $\frac{1}{q} + \frac{1}{q'} = 1$, and for all distributions ρ and μ, we have

$$\|f\|_{p,\rho} \leq \sum_{l=1}^{n} \left(\mathcal{C}_q(l)\right)^{1/p} \sup_{i\in\mathcal{I}_l} \|g_i\|_{pq',\mu} \sum_{i\in\mathcal{I}_l} \sum_{j\in\mathcal{J}_i} \gamma^j,$$

with the following concentrability coefficients

$$\mathcal{C}_q(l) \triangleq \frac{\sum_{i\in\mathcal{I}_l} \sum_{j\in\mathcal{J}_i} \gamma^j c_q(j)}{\sum_{i\in\mathcal{I}_l} \sum_{j\in\mathcal{J}_i} \gamma^j},$$

where $c_q(j)$ is defined by Equation 3.

Proof See Appendix C. ■

We now derive an L_p-norm bound for the loss of the AMPI algorithm by applying Lemma 6 to the point-wise bound of Lemma 4.

Theorem 7 *For all q, l, k and d, define the following concentrability coefficients:*

$$\mathcal{C}_q^{l,k,d} \triangleq \frac{(1-\gamma)^2}{\gamma^l - \gamma^k} \sum_{i=l}^{k-1} \sum_{j=i}^{\infty} \gamma^j c_q(j+d),$$

with $c_q(j)$ given by Equation 3. Let ρ and μ be distributions over states. Let p, q, and q' be such that $\frac{1}{q} + \frac{1}{q'} = 1$. After k iterations, the loss of AMPI satisfies

$$\|l_k\|_{p,\rho} \leq 2 \sum_{i=1}^{k-1} \frac{\gamma^i}{1-\gamma} \left(\mathcal{C}_q^{i,i+1,0}\right)^{\frac{1}{p}} \|\epsilon_{k-i}\|_{pq',\mu} + \sum_{i=0}^{k-1} \frac{\gamma^i}{1-\gamma} \left(\mathcal{C}_q^{i,i+1,0}\right)^{\frac{1}{p}} \|\epsilon'_{k-i}\|_{pq',\mu} + g(k),$$

while the loss of CBMPI satisfies

$$\|l_k\|_{p,\rho} \leq 2\gamma^m \sum_{i=1}^{k-2} \frac{\gamma^i}{1-\gamma} \left(\mathcal{C}_q^{i,i+1,m}\right)^{\frac{1}{p}} \|\epsilon_{k-i-1}\|_{pq',\mu} + \sum_{i=0}^{k-1} \frac{\gamma^i}{1-\gamma} \left(\mathcal{C}_q^{i,i+1,0}\right)^{\frac{1}{p}} \|\epsilon'_{k-i}\|_{pq',\mu} + g(k),$$

where $g(k) \stackrel{\Delta}{=} \frac{2\gamma^k}{1-\gamma} \left(\mathcal{C}_q^{k,k+1,0}\right)^{\frac{1}{p}} \min\left(\|d_0\|_{pq',\mu}, \|b_0\|_{pq',\mu}\right)$.

Proof We only detail the proof for AMPI, the proof is similar for CBMPI. We define $\mathcal{I} = \{1, 2, \ldots, 2k\}$ and the (trivial) partition $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots, \mathcal{I}_{2k}\}$, where $\mathcal{I}_i = \{i\}$, $i \in \{1, \ldots, 2k\}$. For each $i \in \mathcal{I}$, we also define

$$g_i = \begin{cases} 2\epsilon_{k-i} & \text{if} \quad 1 \leq i \leq k-1, \\ \epsilon'_{k-(i-k)} & \text{if} \quad k \leq i \leq 2k-1, \\ 2d_0 \text{ (or } 2b_0) & \text{if} \quad i = 2k, \end{cases}$$

$$\text{and} \quad \mathcal{J}_i = \begin{cases} \{i, \cdots\} & \text{if} \quad 1 \leq i \leq k-1, \\ \{i-k \cdots\} & \text{if} \quad k \leq i \leq 2k-1, \\ \{k\} & \text{if} \quad i = 2k. \end{cases}$$

With the above definitions and the fact that the loss l_k is non-negative, Lemma 4 may be rewritten as

$$|l_k| \leq \sum_{l=1}^{2k} \sum_{i \in \mathcal{I}_l} \sum_{j \in \mathcal{J}_i} \Gamma^j |g_i|.$$

The result follows by applying Lemma 6 and noticing that $\sum_{i=i_0}^{k-1} \sum_{j=i}^{\infty} \gamma^j = \frac{\gamma^{i_0} - \gamma^k}{(1-\gamma)^2}$. ■

Similar to the results of Farahmand et al. (2010), this bound shows that the last iterations have the highest influence on the loss and the influence decreases at the exponential rate γ towards the initial iterations. This phenomenon is related to the fact that the DP algorithms progressively forget about the past iterations. This is similar to the fact that exact VI and PI converge to the optimal limit independently of their initialization.

We can group the terms differently and derive an alternative L_p-norm bound for the loss of AMPI and CBMPI. This also shows the flexibility of Lemma 6 for turning the point-wise bound of Lemma 4 into L_p-norm bounds.

Theorem 8 *With the notations of Theorem 7, and writing* $\epsilon = \sup_{1 \leq j \leq k-1} \|\epsilon_j\|_{pq',\mu}$ *and* $\epsilon' = \sup_{1 \leq j \leq k} \|\epsilon'_j\|_{pq',\mu}$*, the loss of AMPI satisfies*

$$\|l_k\|_{p,\rho} \leq \frac{2(\gamma - \gamma^k) \left(\mathcal{C}_q^{1,k,0}\right)^{\frac{1}{p}}}{(1-\gamma)^2} \epsilon + \frac{(1-\gamma^k) \left(\mathcal{C}_q^{0,k,0}\right)^{\frac{1}{p}}}{(1-\gamma)^2} \epsilon' + g(k), \tag{23}$$

while the loss of CBMPI satisfies

$$\|l_k\|_{p,\rho} \leq \frac{2\gamma^m(\gamma - \gamma^{k-1}) \left(\mathcal{C}_q^{2,k,m}\right)^{\frac{1}{p}}}{(1-\gamma)^2} \epsilon + \frac{(1-\gamma^k) \left(\mathcal{C}_q^{0,k,0}\right)^{\frac{1}{p}}}{(1-\gamma)^2} \epsilon' + g(k). \tag{24}$$

Proof We only give the details of the proof for AMPI, the proof is similar for CBMPI. Defining $\mathcal{I} = \{1, 2, \cdots, 2k\}$ and g_i as in the proof of Theorem 7, we now consider the partition $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \mathcal{I}_3\}$ as $\mathcal{I}_1 = \{1, \ldots, k-1\}$, $\mathcal{I}_2 = \{k, \ldots, 2k-1\}$, and $\mathcal{I}_3 = \{2k\}$, where for each $i \in \mathcal{I}$

$$\mathcal{J}_i = \begin{cases} \{i, i+1, \cdots\} & \text{if} \quad 1 \leq i \leq k-1, \\ \{i-k, i-k+1, \cdots\} & \text{if} \quad k \leq i \leq 2k-1, \\ \{k\} & \text{if} \quad i = 2k. \end{cases}$$

The proof ends similar to that of Theorem 7. ■

By sending the iteration number k to infinity, one obtains the following bound for AMPI:

$$\limsup_{k \to \infty} \|l_k\|_{p,\rho} \leq \frac{2\gamma \left(\mathcal{C}_q^{1,\infty,0}\right)^{\frac{1}{p}} \epsilon + \left(\mathcal{C}_q^{0,\infty,0}\right)^{\frac{1}{p}} \epsilon'}{(1-\gamma)^2}.$$

Compared to the simple max-norm bound of Equation 22, we can see that the price that we must pay to have an error bound in L_p-norm is the appearance of the concentrability coefficients $\mathcal{C}_q^{1,\infty,0}$ and $\mathcal{C}_q^{0,\infty,0}$. Furthermore, it is easy to see that the above bound is more general, i.e., by sending p to infinity, we recover the max-norm bound of Equation 22.

Remark 9 *We can balance the influence of the concentrability coefficients (the bigger the q, the higher the influence) and the difficulty of controlling the errors (the bigger the q', the greater the difficulty in controlling the $L_{pq'}$-norms) by tuning the parameters q and q', given that $\frac{1}{q} + \frac{1}{q'} = 1$. This potential leverage is an improvement over the existing bounds and concentrability results that only consider specific values of these two parameters: $q = \infty$ and $q' = 1$ in Munos (2007) and Munos and Szepesvári (2008), and $q = q' = 2$ in Farahmand et al. (2010).*

Remark 10 *It is important to note that our loss bound for AMPI does not "directly" depend on m (although as we will discuss in the next section, it "indirectly" does through ϵ_k). For CBMPI, the parameter m controls the influence of the value function approximator, cancelling it out in the limit when m tends to infinity (see Equation 24). Assuming a fixed budget of sample transitions, increasing m reduces the number of rollouts used by the classifier, and thus, worsens its quality. In such a situation, m allows making a trade-off between the estimation error of the classifier and the overall value function approximation.*

The arguments we developed globally follow those originally developed for λ-policy iteration (Scherrer, 2013). With respect to that work, our proof is significantly simpler thanks to the use of the Γ^n notation (Definition 3) and the fact that the AMPI scheme is itself much simpler than λ-policy iteration. Moreover, the results are deeper since we consider a possible error in the greedy step and more general concentration coefficients. Canbolat and Rothblum (2012) recently (and independently) developed an analysis of an approximate form of MPI. While Canbolat and Rothblum (2012) only consider the error in the greedy step, our work is more general since it takes into account both this error and the error in the value update. Note that it is required to consider both sources of error for the analysis of CBMPI. Moreover, Canbolat and Rothblum (2012) provide bounds when the errors are

controlled in max-norm, while we consider the more general L_p-norm. At a more technical level, Theorem 2 in Canbolat and Rothblum (2012) bounds the norm of the distance $v_* - v_k$, while we bound the loss $v_* - v_{\pi_k}$. Finally, if we derive a bound on the loss (using *e.g.*, Theorem 1 in Canbolat and Rothblum 2012), this leads to a bound on the loss that is looser than ours. In particular, this does not allow to recover the standard bounds for AVI and API, as we may obtain here (in Equation 22).

The results that we just stated (Theorem 7 and 8) can be read as follows: if one can control the errors ϵ_k and ϵ'_k in L_p-norm, then the performance loss is also controlled. The main limitation of this result is that in general, even if there is no sampling noise (*i.e.*, $N = \infty$ for all the algorithms and $M = \infty$ for AMPI-V), the error ϵ_k of the *evaluation step* may grow arbitrarily and make the algorithm diverge. The fundamental reason is that the composition of the approximation and the Bellman operator T_π is not necessarily contracting. Since the former is contracting with respect to the μ-norm, another reason for this issue is that T_π is in general not contracting for that norm. A simple well-known pathological example is due to Tsitsiklis and Van Roy (1997) and involves a two-state uncontrolled MDP and a linear projection onto a 1-dimensional space (that contains the real value function). Increasing the parameter m of the algorithm makes the operator $(T_\pi)^m$ used in Equation 19 more contracting and can in principle address this issue. For instance, if we consider that we have a state space of finite size $|\mathcal{S}|$, and take the uniform distribution μ, it can be easily seen that for any v and v', we have

$$\begin{aligned}\|(T_\pi)^m v - (T_\pi)^m v'\|_{2,\mu} &= \gamma^m \|(P_\pi)^m (v - v')\|_{2,\mu} \\ &\leq \gamma^m \|(P_\pi)^m\|_{2,\mu} \|v - v'\|_{2,\mu} \\ &\leq \gamma^m \sqrt{|\mathcal{S}|} \|v - v'\|_{2,\mu}.\end{aligned}$$

In other words, $(T_\pi)^m$ is contracting w.r.t. the μ-weighted norm as soon as $m > \frac{\log |\mathcal{S}|}{2 \log \frac{1}{\gamma}}$. In particular, it is sufficient for m to be exponentially smaller than the size of the state space in order to solve this potential divergence problem.

5. Finite-Sample Analysis of the Algorithms

In this section, we first show how the error terms ϵ_k and ϵ'_k appeared in Theorem 8 (Equations 23 and 24) can be bounded in each of the three proposed algorithms, and then use the obtained results and derive finite-sample performance bounds for these algorithms. We first bound the *evaluation step error* ϵ_k. In AMPI-V and CBMPI, the evaluation step at each iteration k is a regression problem with the target $(T_{\pi_k})^m v_{k-1}$ and a training set of the form $\left\{\left(s^{(i)}, \widehat{v}_k(s^{(i)})\right)\right\}_{i=1}^N$ in which the states $s^{(i)}$ are i.i.d. samples from the distribution μ and $\widehat{v}_k(s^{(i)})$'s are unbiased estimates of the target computed using Equation 5. The situation is the same for AMPI-Q, except everything is in terms of action-value function Q_k instead of value function v_k. Therefore, in the following we only show how to bound ϵ_k in AMPI-V and CBMPI, the extension to AMPI-Q is straightforward.

We may use linear or non-linear function space $\mathcal{F}$ to approximate $(T_{\pi_k})^m v_{k-1}$. Here we consider a linear architecture with parameters $\alpha \in \mathbb{R}^d$ and bounded (by L) basis functions $\{\varphi_j\}_{j=1}^d$, $\|\varphi_j\|_\infty \leq L$. We denote by $\phi : \mathcal{X} \to \mathbb{R}^d$, $\phi(\cdot) = \left(\varphi_1(\cdot), \ldots, \varphi_d(\cdot)\right)^\top$ the feature

vector, and by $\mathcal{F}$ the linear function space spanned by the features φ_j, *i.e.*, $\mathcal{F} = \{f_\alpha(\cdot) = \phi(\cdot)^\top \alpha : \alpha \in \mathbb{R}^d\}$. Now if we define v_k as the truncation (by $V_{\max}$) of the solution of the above linear regression problem, we may bound the *evaluation step error* ϵ_k using the following lemma.

Lemma 11 (Evaluation step error) *Consider the linear regression setting described above, then we have*

$$\|\epsilon_k\|_{2,\mu} \leq 4 \inf_{f \in \mathcal{F}} \|(T_{\pi_k})^m v_{k-1} - f\|_{2,\mu} + e_1(N,\delta) + e_2(N,\delta),$$

with probability at least $1-\delta$*, where*

$$e_1(N,\delta) = 32 V_{\max} \sqrt{\frac{2}{N} \log\Big(\frac{27(12e^2N)^{2(d+1)}}{\delta}\Big)}\,,$$
$$e_2(N,\delta) = 24\Big(V_{\max} + \|\alpha_*\|_2 \cdot \sup_x \|\phi(x)\|_2\Big)\sqrt{\frac{2}{N}\log\frac{9}{\delta}}\,,$$

and α_* *is such that* f_{α_*} *is the best approximation (w.r.t.* μ*) of the target function* $(T_{\pi_k})^m v_{k-1}$ *in* $\mathcal{F}$*.*

Proof See Appendix D. ■

After we showed how to bound the *evaluation step error* ϵ_k for the proposed algorithms, we now turn our attention to bounding the *greedy step error* ϵ'_k, that contrary to the evaluation step error, varies more significantly across the algorithms. While the greedy step error equals to zero in AMPI-Q, it is based on sampling in AMPI-V, and depends on a classifier in CBMPI. To bound the *greedy step error* in AMPI-V and CBMPI, we assume that the action space $\mathcal{A}$ contains only two actions, i.e., $|\mathcal{A}| = 2$. The extension to more than two actions is straightforward along the same line of analysis as in Section 6 of Lazaric et al. (2010a). The main difference w.r.t. the two action case is that the VC-dimension of the policy space is replaced with its Natarajan dimension. We begin with AMPI-V.

Lemma 12 (Greedy step error of AMPI-V) *Let* μ *be a distribution over the state space* $\mathcal{S}$ *and* N *be the number of states in the rollout set* $\mathcal{D}_k$ *drawn i.i.d. from* μ*. For each state* $s \in \mathcal{D}_k$ *and each action* $a \in \mathcal{A}$*, we sample* M *states resulted from taking action* a *in state* s*. Let* h *be the VC-dimension of the policy space obtained by Equation 4 from the truncation (by* $V_{\max}$*) of the function space* $\mathcal{F}$*. For any* $\delta > 0$*, the greedy step error* ϵ'_k *in the AMPI-V algorithm is bounded as*

$$||\epsilon'_k(s)||_{1,\mu} \leq e'_3(N,\delta) + e'_4(M,N,\delta) + e'_5(M,N,\delta),$$

with probability at least $1-\delta$*, with*

$$e'_3(N,\delta) = 16 V_{\max}\sqrt{\frac{2}{N}(h\log\frac{eN}{h} + \log\frac{24}{\delta})}\,,$$
$$e'_4(N,M,\delta) = 8 V_{\max}\sqrt{\frac{2}{MN}\big(h\log\frac{eMN}{h} + \log\frac{24}{\delta}\big)}\,, \qquad e'_5(M,N,\delta) = V_{\max}\sqrt{\frac{2\log(3N/\delta)}{M}}\,.$$

Proof See Appendix E. ■

We now show how to bound ϵ'_k in CBMPI. From the definitions of ϵ'_k (Equation 20) and $\mathcal{L}^{\Pi}_k(\mu;\pi)$ (Equation 15), it is easy to see that $\|\epsilon'_k\|_{1,\mu} = \mathcal{L}^{\Pi}_{k-1}(\mu;\pi_k)$. This is because

$$\begin{aligned}\epsilon'_k(s) &= \max_{a\in\mathcal{A}} \big[T_a(T_{\pi_{k-1}})^m v_{k-2}\big](s) - \big[T_{\pi_k}(T_{\pi_{k-1}})^m v_{k-2}\big](s) && \text{(see Equation 13)}\\ &= \max_{a\in\mathcal{A}} Q_{k-1}(s,a) - Q_{k-1}\big(s,\pi_k(s)\big). && \text{(see Equations 14 and 15)}\end{aligned}$$

Lemma 13 (Greedy step error of CBMPI) *Let the policy space Π defined by the classifier have finite VC-dimension $h = VC(\Pi) < \infty$, and μ be a distribution over the state space $\mathcal{S}$. Let N' be the number of states in $\mathcal{D}'_{k-1}$ drawn i.i.d. from μ, M be the number of rollouts per state-action pair used in the estimation of $\widehat{Q}_{k-1}$, and $\pi_k = \operatorname{argmin}_{\pi\in\Pi} \widehat{\mathcal{L}}^{\Pi}_{k-1}(\widehat{\mu},\pi)$ be the policy computed at iteration $k-1$ of CBMPI. Then, for any $\delta > 0$, we have*

$$\|\epsilon'_k\|_{1,\mu} = \mathcal{L}^{\Pi}_{k-1}(\mu;\pi_k) \le \inf_{\pi\in\Pi} \mathcal{L}^{\Pi}_{k-1}(\mu;\pi) + 2\big(e'_1(N',\delta) + e'_2(N',M,\delta)\big),$$

with probability at least $1-\delta$, where

$$\begin{aligned}e'_1(N',\delta) &= 16 Q_{\max}\sqrt{\frac{2}{N'}\big(h\log\frac{eN'}{h} + \log\frac{32}{\delta}\big)}\,,\\ e'_2(N',M,\delta) &= 8 Q_{\max}\sqrt{\frac{2}{MN'}\big(h\log\frac{eMN'}{h} + \log\frac{32}{\delta}\big)}\,.\end{aligned}$$

Proof See Appendix F. ■

From Lemma 11, we have a bound on $\|\epsilon_k\|_{2,\mu}$ for all the three algorithms. Since $\|\epsilon_k\|_{1,\mu} \le \|\epsilon_k\|_{2,\mu}$, we also have a bound on $\|\epsilon_k\|_{1,\mu}$ for all the algorithms. On the other hand, from Lemmas 12 and 13, we have a bound on $\|\epsilon'_k\|_{1,\mu}$ for the AMPI-V and CMBPI algorithms. This means that for AMPI-V, AMPI-Q ($\epsilon'_k = 0$ for this algorithm), and CBMPI, we can control the right hand side of Equations 23 and 24 in L_1-norm, which in the context of Theorem 8 means $p = 1$, $q' = 1$, and $q = \infty$. This leads to the main result of this section, finite-sample performance bounds for the three proposed algorithms.

Theorem 14 *Let*

$$d' = \sup_{g\in\mathcal{F},\pi'\in\Pi} \inf_{\pi\in\Pi} \mathcal{L}^{\Pi}_{\pi',g}(\mu;\pi) \qquad \text{and} \qquad d_m = \sup_{g\in\mathcal{F},\pi\in\Pi} \inf_{f\in\mathcal{F}} \|(T_\pi)^m g - f\|_{2,\mu}$$

where $\mathcal{F}$ is the function space used by the algorithms and Π is the policy space used by CBMPI with the VC-dimension h. With the notations of Theorem 8 and Lemmas 11-13, after k iterations, and with probability $1-\delta$, the expected losses $\mathbb{E}_\rho[l_k] = \|l_k\|_{1,\rho}$ of the proposed AMPI algorithms satisfy:[6]

6. Note that the bounds of AMPI-V and AMPI-Q may also be written with $(p = 2, q' = 1, q = \infty)$, and $(p = 1, q' = 2, q = 2)$.

AMPI-V:
$$\|l_k\|_{1,\rho} \leq \frac{2(\gamma-\gamma^k)\mathcal{C}_\infty^{1,k,0}}{(1-\gamma)^2}\left(d_m + e_1(N,\frac{\delta}{k}) + e_2(N,\frac{\delta}{k})\right) + \frac{(1-\gamma^k)\mathcal{C}_\infty^{0,k,0}}{(1-\gamma)^2}\left(e'_3(N,\frac{\delta}{k}) + e'_4(N,M,\frac{\delta}{k}) + e'_5(N,M,\frac{\delta}{k})\right) + g(k),$$

AMPI-Q:
$$\|l_k\|_{1,\rho} \leq \frac{2(\gamma-\gamma^k)\mathcal{C}_\infty^{1,k,0}}{(1-\gamma)^2}\left(d_m + e_1(N,\frac{\delta}{k}) + e_2(N,\frac{\delta}{k})\right) + g(k),$$

CBMPI:
$$\|l_k\|_{1,\rho} \leq \frac{2\gamma^m(\gamma-\gamma^{k-1})\mathcal{C}_\infty^{2,k,m}}{(1-\gamma)^2}\left(d_m + e_1(N,\frac{\delta}{2k}) + e_2(N,\frac{\delta}{2k})\right) + \frac{(1-\gamma^k)\mathcal{C}_\infty^{1,k,0}}{(1-\gamma)^2}\left(d' + e'_1(N',\frac{\delta}{2k}) + e'_2(N',M,\frac{\delta}{2k})\right) + g(k).$$

Remark 15 *Assume that we run AMPI-Q with a total fixed budget B that is equally divided between the K iterations.*[7] *Recall from Theorem 8 that $g(k) = \gamma^k \mathcal{C}_q^{k,k+1,0} C_0$, where $C_0 = \min\left(\|d_0\|_{pq',\mu}, \|b_0\|_{pq',\mu}\right) \leq V_{\max}$ measures the quality of the initial value/policy pair. Then, up to constants and logarithmic factors, one can see that the bound has the form*

$$\|l_k\|_{1,\mu} \leq O\left(d_m + \sqrt{\frac{K}{B}} + \gamma^K C_0\right).$$

We deduce that the best choice for the number of iterations K can be obtained as a compromise between the quality of the initial value/policy pair and the estimation errors of the value estimation step.

Remark 16 *The CBMPI bound in Theorem 14 allows to turn the* qualitative *Remark 10 into a* quantitative *one. Assume that we have a fixed budget per iteration $B = Nm + N'M|\mathcal{A}|(m+1)$ that is equally divided over the classifier and regressor. Note that the budget is measured in terms of the number of calls to the generative model. Then up to constants and logarithmic factors, the bound has the form*

$$\|l_k\|_{1,\mu} \leq O\left(\gamma^m\left(d_m + \sqrt{\frac{m}{B}}\right) + d' + \sqrt{\frac{|\mathcal{A}|mM}{B}}\right).$$

This shows a trade-off in tuning the parameter m: a large value of m makes the influence (in the final error) of the regressor's error (both approximation and estimation errors) smaller, and at the same time the influence of the estimation error of the classifier larger.

6. Experimental Results

The main objective of this section is to present experiments for the new algorithm that we think is the most interesting, CBMPI, but we shall also illustrate AMPI-Q (we do not

7. Similar reasoning can be done for AMPI-V and CBMPI, we selected AMPI-Q for the sake of simplicity. Furthermore, one could easily relax the assumption that the budget is *equally* divided by using Theorem 7.

illustrate AMPI-V that is close to AMPI-Q but significantly less efficient to implement). We consider two different domains: **1)** the *mountain car* problem and **2)** the more challenging game of *Tetris*. In several experiments, we compare the performance of CBMPI with the DPI algorithm (Lazaric et al., 2010c), which is basically CBMPI without value function approximation.[8] Note that comparing DPI and CBMPI allows us to highlight the role of the value function approximation.

As discussed in Remark 10, the parameter m in CBMPI balances between the errors in evaluating the value function and the policy. The value function approximation error tends to zero for large values of m. Although this would suggest to have large values for m, as mentioned in Remark 16, the size of the rollout sets $\mathcal{D}$ and $\mathcal{D}'$ would correspondingly decreases as $N = O(B/m)$ and $N' = O(B/m)$, thus decreasing the accuracy of both the regressor and classifier. This leads to a trade-off between long rollouts and the number of states in the rollout sets. The solution to this trade-off strictly depends on the capacity of the value function space $\mathcal{F}$. A rich value function space would lead to solve the trade-off for small values of m. On the other hand, when the value function space is poor, or, as in the case of DPI, when there is no value function, m should be selected in a way to guarantee large enough rollout sets (parameters N and N'), and at the same time, a sufficient number of rollouts (parameter M).

One of the objectives of our experiments is to show the role of these parameters in the performance of CBMPI. However, since we almost always obtained our best results with $M = 1$, we only focus on the parameters m and N in our experiments. Moreover, as mentioned in Footnote 3, we implement a more sample-efficient version of CBMPI by reusing the rollouts generated for the classifier in the regressor. More precisely, at each iteration k, for each state $s^{(i)} \in \mathcal{D}'_k$ and each action $a \in \mathcal{A}$, we generate one rollout of length $m+1$, i.e., $\big(s^{(i)}, a, r_0^{(i)}, s_1^{(i)}, a_1^{(i)}, \ldots, a_m^{(i)}, r_m^{(i)}, s_{m+1}^{(i)}\big)$. We then take the rollout of action $\pi_k(s^{(i)})$, select its last m steps, i.e., $\big(s_1^{(i)}, a_1^{(i)}, \ldots, a_m^{(i)}, r_m^{(i)}, s_{m+1}^{(i)}\big)$ (note that all the actions here have been taken according to the current policy π_k), use it to estimate the value function $\widehat{v}_k(s_1^{(i)})$, and add it to the training set of the regressor. This process guarantees to have $N = N'$.

In each experiment, we run the algorithms with the same budget B per iteration. The budget B is the number of next state samples generated by the generative model of the system at each iteration. In DPI and CBMPI, we generate a rollout of length $m+1$ for each state in $\mathcal{D}'$ and each action in $\mathcal{A}$, so, $B = (m+1)N|\mathcal{A}|$. In AMPI-Q, we generate one rollout of length m for each state-action pair in $\mathcal{D}$, and thus, $B = mN$.

6.1 Mountain Car

Mountain Car (MC) is the problem of driving a car up to the top of a one-dimensional hill (see Figure 4). The car is not powerful enough to accelerate directly up the hill, and thus, it must learn to oscillate back and forth to build up enough inertia. There are three possible actions: forward (+1), reverse (−1), and stay (0). The reward is −1 for all the states but the goal state at the top of the hill, where the episode ends with a reward 0. The discount factor is set to $\gamma = 0.99$. Each state s consists of the pair $(x_s, \dot{x}_s)$, where x_s is the

8. DPI, as it is presented by Lazaric et al. (2010c), uses infinitely long rollouts and is thus equivalent to CBMPI with $m = \infty$. In practice, implementations of DPI use rollouts that are truncated after some horizon H, and is then equivalent to CBMPI with $m = H$ and $v_k = 0$ for all the iterations k.

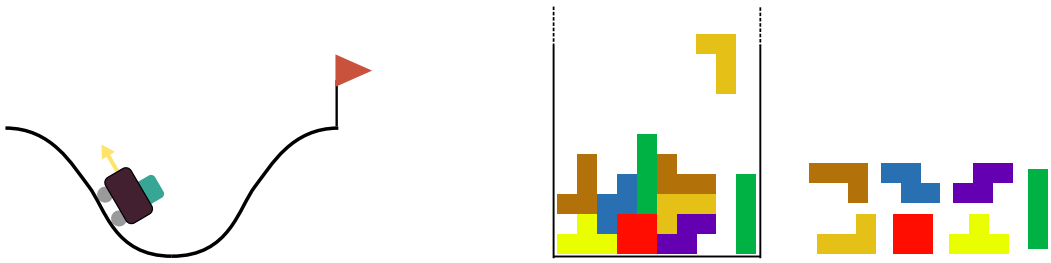


Figure 4: ***(Left)*** The Mountain Car (MC) problem in which the car needs to learn to oscillate back and forth in order to build up enough inertia to reach the top of the one-dimensional hill. ***(Right)*** A screen-shot of the game of Tetris and the seven pieces (shapes) used in the game.

position of the car and $\dot{x}_s$ is its velocity. We use the formulation described in Dimitrakakis and Lagoudakis (2008) with uniform noise in $[-0.2, 0.2]$ added to the actions.

In this section, we report the empirical evaluation of CBMPI and AMPI-Q and compare it to DPI and LSPI (Lagoudakis and Parr, 2003a) in the MC problem. In our experiments, we show that CBMPI, by combining policy and value function approximation, can improve over AMPI-Q, DPI, and LSPI.

6.1.1 Experimental Setup

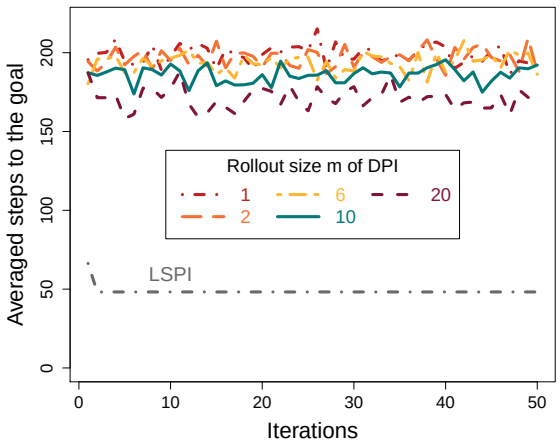
The value function is approximated using a linear space spanned by a set of radial basis functions (RBFs) evenly distributed over the state space. More precisely, we uniformly divide the 2-dimensional state space into a number of regions and place a Gaussian function at the center of each of them. We set the standard deviation of the Gaussian functions to the width of a region. The function space to approximate the action-value function in LSPI is obtained by replicating the state-features for each action. We run LSPI off-policy (i.e., samples are collected once and reused through the iterations of the algorithm).

The policy space Π (classifier) is defined by a regularized support vector classifier (C-SVC) using the LIBSVM implementation by Chang and Lin (2011). We use the RBF kernel $\exp(-|u-v|^2)$ and set the cost parameter $C = 1000$. We minimize the classification error instead of directly solving the cost-sensitive multi-class classification step as in Figure 3. In fact, the classification error is an upper-bound on the empirical error defined by Equation 17. Finally, the rollout set is sampled uniformly over the state space.

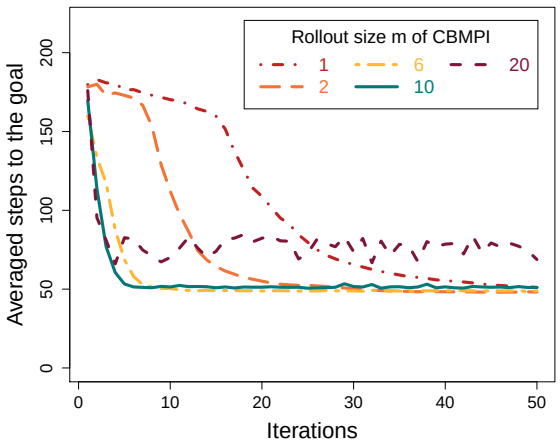
In our MC experiments, the policies learned by the algorithms are evaluated by the number of steps-to-go (average number of steps to reach the goal with a maximum of 300) averaged over $4,000$ independent trials. More precisely, we define the possible starting configurations (positions and velocities) of the car by placing a 20×20 uniform grid over the state space, and run the policy 6 times from each possible initial configuration. The performance of each algorithm is represented by a learning curve whose value at each iteration is the average number of steps-to-go of the policies learned by the algorithm at that iteration in $1,000$ separate runs of the algorithm.

We tested the performance of DPI, CBMPI, and AMPI-Q on a wide range of parameters (m, M, N), but only report their performance for the best choice of M (as mentioned earlier, $M = 1$ was the best choice in all the experiments) and different values of m.

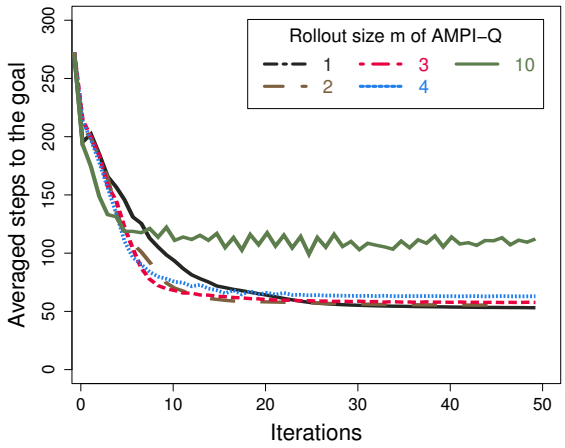
6.1.2 Experimental Results



(a) Performance of DPI (for different values of m) and LSPI.



(b) Performance of CBMPI for different values of m.

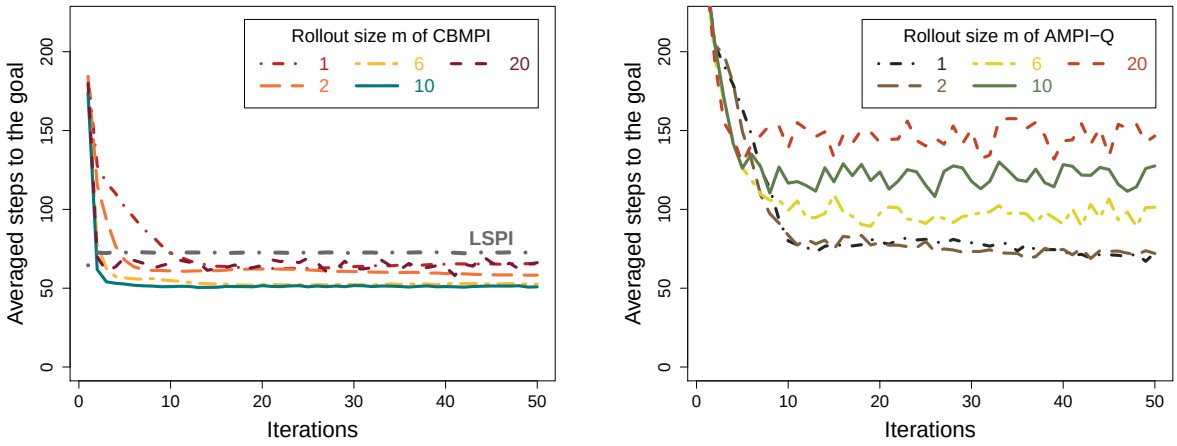


(c) Performance of AMPI-Q for different values of m.

Figure 5: Performance of the policies learned by **(a)** DPI and LSPI, **(b)** CBMPI, and **(c)** AMPI-Q algorithms in the Mountain Car (MC) problem, when we use a 3×3 RBF grid to approximate the value function. The results are averaged over $1,000$ runs. The total budget B is set to $4,000$ per iteration.

Figure 5 shows the learning curves of DPI, CBMPI, AMPI-Q, and LSPI algorithms with budget $B = 4,000$ per iteration and the function space $\mathcal{F}$ composed of a 3×3 RBF grid.

We notice from the results that this space is rich enough to provide a good approximation for the value function components (e.g., in CBMPI, for $(T_\pi)^m v_{k-1}$ defined by Equation 19). Therefore, LSPI and DPI obtain the best and worst results with about 50 and 160 steps to reach the goal, respectively. The best DPI results are obtained with the large value of $m = 20$. DPI performs better for large values of m because the reward function is constant everywhere except at the goal, and thus, a DPI rollout is only *informative* if it reaches the goal. We also report the performance of CBMPI and AMPI-Q for different values of m. The value function approximation is very accurate, and thus, CBMPI and AMPI-Q achieve performance similar to LSPI for $m < 20$. However when m is large ($m = 20$), the performance of these algorithms is worse, because in this case, the rollout set does not have enough elements (N small) to learn the greedy policy and value function well. Note that as we increase m (up to $m = 10$), CBMPI and AMPI-Q converge faster to a good policy.



(a) Performance of CBMPI (for different values of m) and LSPI.

(b) Performance of AMPI-Q for different values of m.

Figure 6: Performance of the policies learned by **(a)** CBMPI and LSPI and **(b)** AMPI-Q algorithms in the Mountain Car (MC) problem, when we use a 2×2 RBF grid to approximate the value function. The results are averaged over $1,000$ runs. The total budget B is set to $4,000$ per iteration.

Although this experiment shows that the use of a critic in CBMPI compensates for the truncation of the rollouts (CBMPI performs better than DPI), most of this advantage is due to the richness of the function space $\mathcal{F}$ (LSPI and AMPI-Q perform as well as CBMPI—LSPI even converges faster). Therefore, it seems that it would be more efficient to use LSPI instead of CBMPI in this case.

In the next experiment, we study the performance of these algorithms when the function space $\mathcal{F}$ is less rich, composed of a 2×2 RBF grid. The results are reported in Figure 6. Now, the performance of LSPI and AMPI-Q (for the best value of $m = 1$) degrades to 75 and 70 steps, respectively. Although $\mathcal{F}$ is not rich, it still helps CBMPI to outperform DPI. We notice the effect of (a weaker) $\mathcal{F}$ in CBMPI when we observe that it no longer converges to its best performance (about 50 steps) for small values of $m = 1$ and $m = 2$. Note that

CMBPI outperforms all the other algorithms for $m = 10$ (and even for $m = 6$), while still has a sub-optimal performance for $m = 20$, mainly due to the fact that the rollout set would be too small in this case.

6.2 Tetris

Tetris is a popular video game created by Alexey Pajitnov in 1985. The game is played on a grid originally composed of 20 rows and 10 columns, where pieces of 7 different shapes fall from the top (see Figure 4). The player has to choose where to place each falling piece by moving it horizontally and rotating it. When a row is filled, it is removed and all the cells above it move one line down. The goal is to remove as many rows as possible before the game is over, i.e., when there is no space available at the top of the grid for the new piece. This game constitutes an interesting optimization benchmark in which the goal is to find a controller (policy) that maximizes the average (over multiple games) number of lines removed in a game (score).[9] This optimization problem is known to be computationally hard. It contains a huge number of board configurations (about $2^{200} \simeq 1.6 \times 10^{60}$), and even in the case that the sequence of pieces is known in advance, finding the strategy to maximize the score is a NP hard problem (Demaine et al., 2003). Here, we consider the variation of the game in which the player only knows the current falling piece and none of the next several coming pieces.

Approximate dynamic programming (ADP) and reinforcement learning (RL) algorithms including approximate value iteration (Tsitsiklis and Van Roy, 1996), λ-policy iteration (λ-PI) (Bertsekas and Ioffe, 1996; Scherrer, 2013), linear programming (Farias and Van Roy, 2006), and natural policy gradient (Kakade, 2002; Furmston and Barber, 2012) have been applied to this very setting. These methods formulate Tetris as a MDP (with discount factor $\gamma = 1$) in which the state is defined by the current board configuration plus the falling piece, the actions are the possible orientations of the piece and the possible locations that it can be placed on the board,[10] and the reward is defined such that maximizing the expected sum of rewards from each state coincides with maximizing the score from that state. Since the state space is large in Tetris, these methods use value function approximation schemes (often linear approximation) and try to tune the value function parameters (weights) from game simulations. Despite a long history, ADP/RL algorithms, that have been (almost) entirely based on approximating the value function, have not been successful in finding good policies in Tetris. On the other hand, methods that search directly in the space of policies by learning the policy parameters using black-box optimization, such as the cross entropy (CE) method (Rubinstein and Kroese, 2004), have achieved the best reported results in this game (see e.g., Szita and Lőrincz 2006; Thiery and Scherrer 2009b). This makes us conjecture that Tetris is a game in which good policies are easier to represent, and thus to learn, than their corresponding value functions. So, in order to obtain a good performance with ADP in Tetris, we should use those ADP algorithms that search in a policy space, like CBMPI and DPI, instead of the more traditional ones that search in a value function space.

9. Note that this number is finite because it was shown that Tetris is a game that ends with probability one (Burgiel, 1997).
10. The total number of actions at a state depends on the shape of the falling piece, with the maximum of 32 actions in a state, i.e., $|\mathcal{A}| \leq 32$.

In this section, we evaluate the performance of CBMPI in Tetris and compare it with DPI, λ-PI, and CE. In these experiments, we show that CBMPI improves over all the previously reported ADP results. Moreover, it obtains the best results reported in the literature for Tetris in both small 10×10 and large 10×20 boards. Although the CBMPI's results are similar to those achieved by the CE method in the large board, it uses considerably fewer samples (call to the generative model of the game) than CE.

6.2.1 Experimental Setup

In this section, we briefly describe the algorithms used in our experiments: the cross entropy (CE) method, our particular implementation of CBMPI, and its slight variation DPI. We refer the readers to Scherrer (2013) for λ-PI. We begin by defining some terms and notations. A state s in Tetris consists of two components: the description of the board b and the type of the falling piece p. All controllers rely on an evaluation function that gives a value to each possible action at a given state. Then, the controller chooses the action with the highest value. In ADP, algorithms aim at tuning the weights such that the evaluation function approximates well the value function, which coincides with the optimal expected future score from each state. Since the total number of states is large in Tetris, the evaluation function f is usually defined as a linear combination of a set of features ϕ, i.e., $f(\cdot) = \phi(\cdot)^\top \theta$. Alternatively, we can think of the parameter vector θ as a policy (controller) whose performance is specified by the corresponding evaluation function $f(\cdot) = \phi(\cdot)^\top \theta$. The features used in Tetris for a state-action pair (s, a) may depend on the description of the board b' resulting from taking action a in state s, e.g., the maximum height of b'. Computing such features requires to exploit the knowledge of the game's dynamics (this dynamics is indeed known for tetris). We consider the following sets of features, plus a constant offset feature:[11]

(i) Bertsekas Features: First introduced by Bertsekas and Tsitsiklis (1996), this set of 22 features has been mainly used in the ADP/RL community and consists of: *the number of holes in the board*, *the height of each column*, *the difference in height between two consecutive columns*, and *the maximum height of the board*.

(ii) Dellacherie-Thiery (D-T) Features: This set consists of the six features of Dellacherie (Fahey, 2003), i.e., *the landing height of the falling piece*, *the number of eroded piece cells*, *the row transitions*, *the column transitions*, *the number of holes*, and *the number of board wells*; plus 3 additional features proposed in Thiery and Scherrer (2009b), i.e., *the hole depth*, *the number of rows with holes*, and *the pattern diversity feature*. Note that the best policies reported in the literature have been learned using this set of features.

(iii) RBF Height Features: These new 5 features are defined as $\exp(\frac{-|c-ih/4|^2}{2(h/5)^2})$, $i = 0, \ldots, 4$, where c is the average height of the columns and $h = 10$ or 20 is the total number of rows in the board.

11. For a precise definition of the features, see Thiery and Scherrer (2009a) or the documentation of their code (Thiery and Scherrer, 2010b). Note that the constant offset feature only plays a role in value function approximation, and has no effect in modeling polices.

Input: parameter space Θ, number of parameter vectors n, proportion $\zeta \leq 1$, noise η
Initialize: Set the mean and variance parameters $\boldsymbol{\mu} = (0, 0, \ldots, 0)$ and $\boldsymbol{\sigma}^2 = (100, 100, \ldots, 100)$
for $k = 1, 2, \ldots$ **do**
  Generate a random sample of n parameter vectors $\{\theta_i\}_{i=1}^{n} \sim \mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$
  For each θ_i, play G games and calculate the average number of rows removed (score) by the controller
  Select $\lfloor \zeta n \rfloor$ parameters with the highest score $\theta'_1, \ldots, \theta'_{\lfloor \zeta n \rfloor}$
  Update $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$: $\mu(j) = \frac{1}{\lfloor \zeta n \rfloor} \sum_{i=1}^{\lfloor \zeta n \rfloor} \theta'_i(j)$ and $\sigma^2(j) = \frac{1}{\lfloor \zeta n \rfloor} \sum_{i=1}^{\lfloor \zeta n \rfloor} [\theta'_i(j) - \mu(j)]^2 + \eta$
end for

Figure 7: The pseudo-code of the cross-entropy (CE) method used in our experiments.

The Cross Entropy (CE) Method: CE (Rubinstein and Kroese, 2004) is an iterative method whose goal is to optimize a function f parameterized by a vector $\theta \in \Theta$ by direct search in the parameter space Θ. Figure 7 contains the pseudo-code of the CE algorithm used in our experiments (Szita and Lőrincz, 2006; Thiery and Scherrer, 2009b). At each iteration k, we sample n parameter vectors $\{\theta_i\}_{i=1}^{n}$ from a multivariate Gaussian distribution with diagonal covariance matrix $\mathcal{N}(\boldsymbol{\mu}, \mathrm{diag}(\boldsymbol{\sigma}^2))$. At the beginning, the parameters of this Gaussian have been set to cover a wide region of Θ. For each parameter θ_i, we play G games and calculate the average number of rows removed by this controller (an estimate of the expected score). We then select $\lfloor \zeta n \rfloor$ of these parameters with the highest score, $\theta'_1, \ldots, \theta'_{\lfloor \zeta n \rfloor}$, and use them to update the mean $\boldsymbol{\mu}$ and variance $\mathrm{diag}(\boldsymbol{\sigma}^2)$ of the Gaussian distribution, as shown in Figure 7. This updated Gaussian is used to sample the n parameters at the next iteration. The goal of this update is to sample more parameters from the promising parts of Θ at the next iteration, and hopefully converge to a good maximum of f. In our experiments, in the pseudo-code of Figure 7, we set $\zeta = 0.1$ and $\eta = 4$, the best parameters reported in Thiery and Scherrer (2009b). We also set $n = 1,000$ and $G = 10$ in the small board (10×10) and $n = 100$ and $G = 1$ in the large board (10×20).

Our Implementation of CBMPI (DPI): We use the algorithm whose pseudo-code is shown in Figure 3. We sampled states from the trajectories generated by a good policy for Tetris, namely the DU controller obtained by Thiery and Scherrer (2009b). Since this policy is good, this set is biased towards boards with small height. The rollout set is then obtained by subsampling this set so that the board height distribution is more uniform. We noticed from our experiments that this subsampling significantly improves the performance. We now describe how we implement the regressor and the classifier.

- **Regressor:** We use linear function approximation for the value function, i.e., $\widehat{v}_k(s^{(i)}) = \phi(s^{(i)})\alpha$, where $\phi(\cdot)$ and α are the feature and weight vectors, and minimize the empirical error $\widehat{\mathcal{L}}_k^{\mathcal{F}}(\widehat{\mu}; v)$ using the standard least-squares method.

- **Classifier:** The training set of the classifier is of size N with $s^{(i)} \in \mathcal{D}'_k$ as input and $\big(\max_a \widehat{Q}_k(s^{(i)}, a) - \widehat{Q}_k(s^{(i)}, a_1), \ldots, \max_a \widehat{Q}_k(s^{(i)}, a) - \widehat{Q}_k(s^{(i)}, a_{|\mathcal{A}|})\big)$ as output. We use the policies of the form $\pi_\beta(s) = \mathrm{argmax}_a \, \psi(s, a)^\top \beta$, where ψ is the policy

feature vector (possibly different from the value function feature vector ϕ) and $\beta \in \mathcal{B}$ is the policy parameter vector. We compute the next policy π_{k+1} by minimizing the empirical error $\widehat{\mathcal{L}}_k^{\Pi}(\widehat{\mu}; \pi_\beta)$, defined by (17), using the covariance matrix adaptation evolution strategy (CMA-ES) algorithm (Hansen and Ostermeier, 2001). In order to evaluate a policy $\beta \in \mathcal{B}$ in CMA-ES, we only need to compute $\widehat{\mathcal{L}}_k^{\Pi}(\widehat{\mu}; \pi_\beta)$, and given the training set, this procedure does not require further simulation of the game.

We set the initial value function parameter to $\alpha = (0, 0, \ldots, 0)$ and select the initial policy π_1 (policy parameter β) randomly. We also set the CMA-ES parameters (classifier parameters) to $\zeta = 0.5$, $\eta = 0$, and n equal to 15 times the number of features.

6.2.2 Experiments

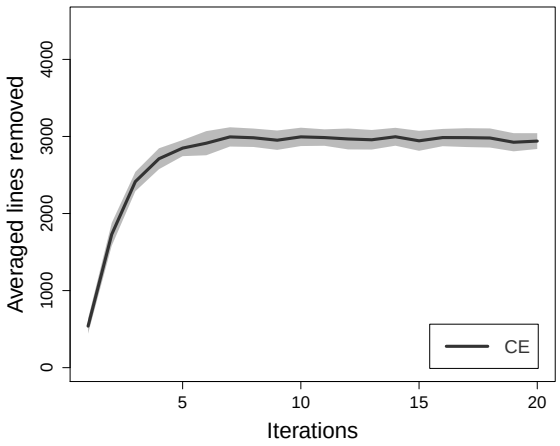
In our Tetris experiments, the policies learned by the algorithms are evaluated by their score (average number of rows removed in a game started with an empty board) averaged over 200 games in the small 10×10 board and over 20 games in the large 10×20 board (since the game takes much more time to complete in the large board). The performance of each algorithm is represented by a learning curve whose value at each iteration is the average score of the policies learned by the algorithm at that iteration in 100 separate runs of the algorithm. The curves are wrapped in their confidence intervals that are computed as three time the standard deviation of the estimation of the performance at each iteration. In addition to their score, we also evaluate the algorithms by the number of samples they use. In particular, we show that CBMPI/DPI use 6 times fewer samples than CE in the large board. As discussed in Section 6.2.1, this is due the fact that although the classifier in CBMPI/DPI uses a direct search in the space of policies (for the greedy policy), it evaluates each candidate policy using the empirical error of Equation 17, and thus, does not require any simulation of the game (other than those used to estimate the $\widehat{Q}_k$'s in its training set). In fact, the budget B of CBMPI/DPI is fixed in advance by the number of rollouts NM and the rollout's length m as $B = (m+1)NM|\mathcal{A}|$. On the contrary, CE evaluates a candidate policy by playing several games, a process that can be extremely costly (sample-wise), especially for good policies in the large board.

We first run the algorithms on the small board to study the role of their parameters and to select the best features and parameters, and then use the selected features and parameters and apply the algorithms to the large board. Finally, we compare the best policies found in our experiments with the best controllers reported in the literature (Tables 1 and 2).

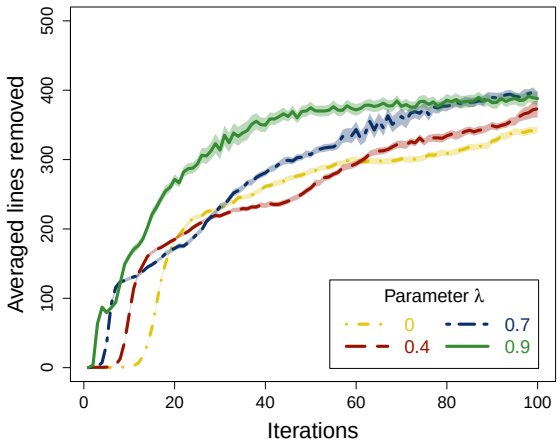
6.2.2.1 Small ($\mathbf{10 \times 10}$) Board

Here we run the algorithms with two different feature sets: *Dellacherie-Thiery (D-T)* and *Bertsekas*, and report their results.

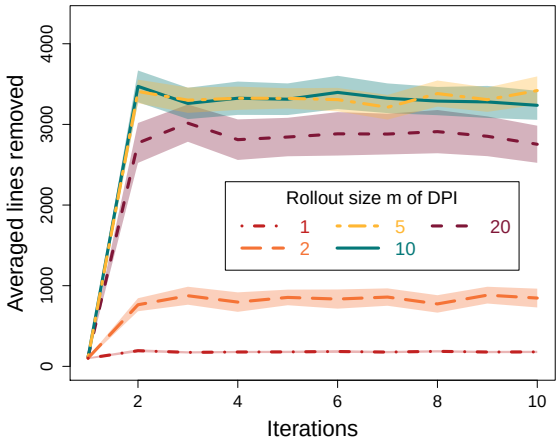
D-T Features: Figure 8 shows the learning curves of CE, λ-PI, DPI, and CBMPI. Here we use the D-T features for the evaluation function in CE, the policy in DPI and CBMPI, and the value function in λ-PI (in the last case we also add the constant offset feature). For the value function of CBMPI, we tried different choices of features and "D-T plus the 5 RBF features and constant offset" achieved the best performance (see Figure 8(d)). The budget



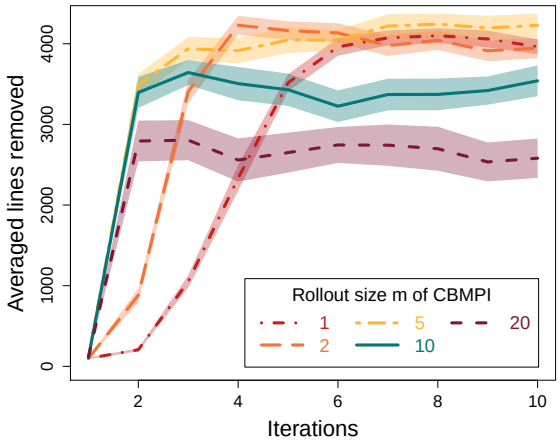
(a) The cross-entropy (CE) method.



(b) λ-PI with $\lambda = \{0, 0.4, 0.7, 0.9\}$.



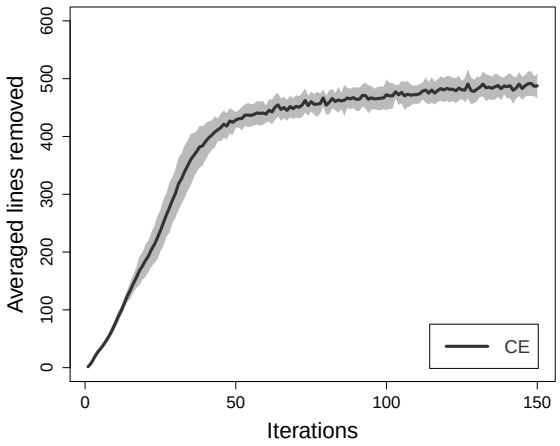
(c) DPI with budget $B = 8,000,000$ per iteration and $m = \{1, 2, 5, 10, 20\}$.



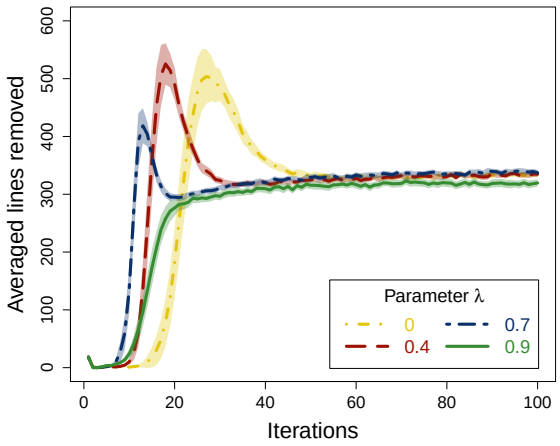
(d) CBMPI with budget $B = 8,000,000$ per iteration and $m = \{1, 2, 5, 10, 20\}$.

Figure 8: Learning curves of CE, λ-PI, DPI, and CBMPI using the 9 Dellacherie-Thiery (D-T) features on the small 10×10 board. The results are averaged over 100 runs of the algorithms.

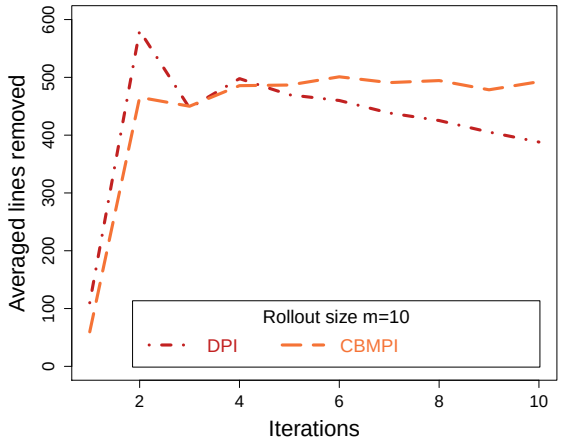
of CBMPI and DPI is set to $B = 8,000,000$ per iteration. The CE method reaches the score $3,000$ after 10 iterations using an average budget $B = 65,000,000$. λ-PI with the best value of λ only manages to score 400. In Figure 8(c), we report the performance of DPI for different values of m. DPI achieves its best performance for $m = 5$ and $m = 10$ by removing $3,400$ lines on average. As explained in Section 6.1, having short rollouts ($m = 1$) in DPI leads to poor action-value estimates $\widehat{Q}$, while having too long rollouts ($m = 20$) decreases the size of the training set of the classifier N. CBMPI outperforms the other algorithms, including CE, by reaching the score of $4,200$ for $m = 5$. This value of $m = 5$ corresponds to $N = \frac{8000000}{(5+1)\times 32} \approx 42,000$. Note that unlike DPI, CBMPI achieves good performance with very short rollouts $m = 1$. This indicates that CBMPI is able to approximate the value



(a) The cross-entropy (CE) method.



(b) λ-PI with $\lambda = \{0, 0.4, 0.7, 0.9\}$.



(c) DPI (dash-dotted line) & CBMPI (dash line) with budget $B = 80,000,000$ per iteration and $m = 10$.

Figure 9: (a)-(c) Learning curves of CE, λ-PI, DPI, and CBMPI algorithms using the 22 Bertsekas features on the small 10×10 board.

function well, and as a result, build a more accurate training set for its classifier than DPI. Despite this improvement, the good results obtained by DPI in Tetris indicate that with small rollout horizons like $m = 5$, one already has fairly accurate action-value estimates in order to detect greedy actions accurately (at each iteration).

Overall, the results of Figure 8 show that an ADP algorithm, namely CBMPI, outperforms the CE method using a similar budget (80 vs. 65 millions after 10 iterations). Note that CBMPI takes less iterations to converge than CE. More generally, Figure 8 confirms the superiority of the policy search and classification-based PI methods to value-function based ADP algorithms (λ-PI). This suggests that the D-T features are more suitable to represent policies than value functions in Tetris.

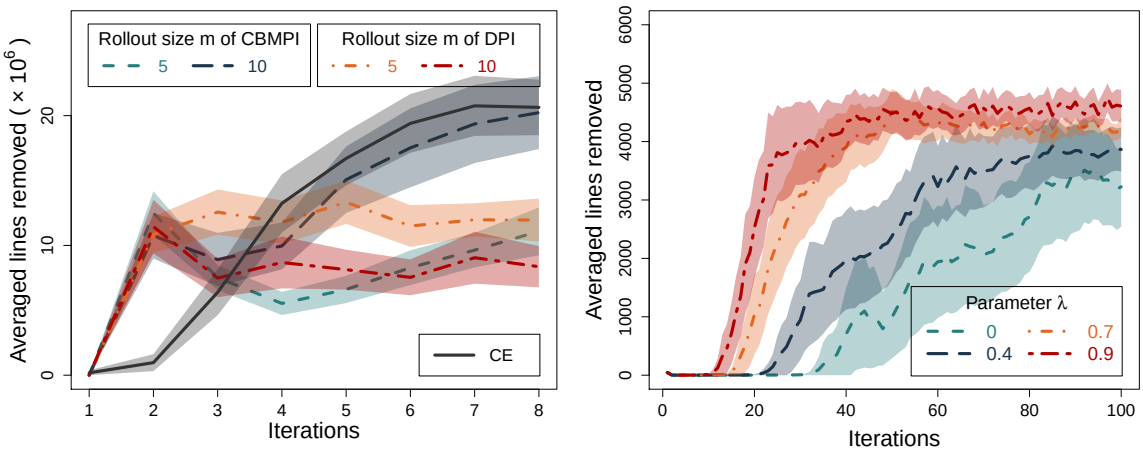


Figure 10: Learning curves of CBMPI, DPI and CE (left) using the 9 features listed in Table 2, and λ-PI (right) using the Bertsekas features (those for which λ-PI achieves here its best performance) on the large 10×20 board. The total budget B of CBMPI and DPI is set to 16,000,000 per iteration.

Bertsekas Features: Figures 9(a)-(c) show the performance of CE, λ-PI, DPI, and CBMPI. Here all the approximations in the algorithms are with the Bertsekas features plus constant offset. CE achieves the score 500 after about 60 iterations and outperforms λ-PI with score 350. It is clear that the Bertsekas features lead to much weaker results than those obtained by the D-T features (Figure 8) for all the algorithms. We may conclude then that the D-T features are more suitable than the Bertsekas features to represent both value functions and policies in Tetris. In DPI and CBMPI, we managed to obtain results similar to CE, only after multiplying the per iteration budget B used in the D-T experiments by 10. Indeed, CBMPI and DPI need more samples to solve the classification and regression problems in this 22-dimensional weight vector space than with the 9 D-T features. Moreover, in the classifier, the minimization of the empirical error through the CMA-ES method (see Equation 12) was converging most of the times to a local minimum. To solve this issue, we run multiple times the minimization problem with different starting points and small initial covariance matrices for the Gaussian distribution in order to force local exploration of different parts of the weight vector areas. However, CBMPI and CE require the same number of samples, $150,000,000$, to reach their best performance, after 2 and 60 iterations, respectively (see Figure 9). Note that DPI and CBMPI obtain the same performance, which means that the use of a value function approximation by CBMPI does not lead to a significant performance improvement over DPI. We tried several values of m in this setting among which $m = 10$ achieved the best performance for both DPI and CBMPI.

6.2.2.2 Large ($\mathbf{10 \times 20}$) Board

We now use the best parameters and features in the small board experiments, run CE, DPI, and CBMPI in the large board, and report their results in Figure 10 (left). We also report

the results of λ-PI in the large board in Figure 10 (right). The per iteration budget of DPI and CBMPI is set to $B = 32,000,000$. While λ-PI with per iteration budget $100,000$, at its best, achieves the score of $2,500$, DPI and CBMPI, with $m = 5$ and $m = 10$, reach the scores of $12,000,000$ and $20,000,000$ after 3 and 8 iterations, respectively. CE matches the performances of CBMPI with the score of $20,000,000$ after 8 iterations, this is achieved with almost 6 times more samples: after 8 iterations, CBMPI and CE use $256,000,000$ and $1,700,000,000$ samples, respectively.

6.2.2.3 Comparison of the Best Policies

So far the reported scores for each algorithm was averaged over the policies learned in 100 separate runs. Here we select the best policies observed in all our experiments and compute their scores more accurately by averaging over $10,000$ games. We then compare these results with the best policies reported in the literature, i.e., DU and BDU (Thiery and Scherrer, 2009b) in both small and large boards in Table 1. The DT-10 and DT-20 policies, whose weights and features are given in Table 2, are policies learned by CBMPI with D-T features in the small and large boards, respectively.[12] As shown in Table 1, DT-10 removes $5,000$ lines and outperforms DU, BDU, and DT-20 in the small board. Note that DT-10 is the only policy among these four that has been learned in the small board. In the large board, DT-20 obtains the score of $51,000,000$ and not only outperforms the other three policies, but also achieves the best reported result in the literature (to the best of our knowledge). We observed in our experiments that the learning process in CBMPI has more variance in its performance than the one of CE. We believe this is why in the large board, although the policies learned by CE have similar performance to CBMPI (see Figure 10 (left)), the best policy learned by CBMPI outperforms BDU, the best one learned by CE (see Table 1).

Boards \ Policies	**DU**	**BDU**	**DT-10**	**DT-20**
Small (10×10) board	3800	4200	5000	4300
Large (10×20) board	$31,000,000$	$36,000,000$	$29,000,000$	$51,000,000$

Table 1: Average (over $10,000$ games) score of DU, BDU, DT-10, and DT-20 policies.

feature	**weight**		**feature**	**weight**		**feature**	**weight**	
landing height	-2.18	-2.68	column transitions	-3.31	-6.32	hole depth	-0.81	-0.43
eroded piece cells	2.42	1.38	holes	0.95	2.03	rows w/ holes	-9.65	-9.48
row transitions	-2.17	-2.41	board wells	-2.22	-2.71	diversity	1.27	0.89

Table 2: The weights of the 9 D-T features in the DT-10 (left) and DT-20 (right) policies.

12. Note that in the standard code by Thiery and Scherrer (2010b), there exist two versions of the feature "board wells" numbered 6 and -6. In our experiments, we used the feature -6 as it is the more computationally efficient of the two.

7. Conclusion

In this paper, we considered a dynamic programming (DP) scheme for Markov decision processes, known as modified policy iteration (MPI). We proposed three original approximate MPI (AMPI) algorithms that are extensions of the existing approximate DP (ADP) algorithms: fitted-value iteration, fitted-Q iteration, and classification-based policy iteration. We reported a general error propagation analysis for AMPI that unifies those for approximate policy and value iteration. We instantiated this analysis for the three algorithms that we introduced, which led to a finite-sample analysis of their guaranteed performance. For the last introduced algorithm, CBMPI, our analysis indicated that the main parameter of MPI controls the balance of errors (between value function approximation and estimation of the greedy policy). The role of this parameter was illustrated for all the algorithms on two benchmark problems: Mountain Car and Tetris. Remarkably, in the game of Tetris, CBMPI showed advantages over all previous approaches: it significantly outperforms previous ADP approaches, and is competitive with black-box optimization techniques—the current state of the art for this domain—while using fewer samples. In particular, CBMPI led to what is to our knowledge the currently best Tetris controller, removing $51,000,000$ lines on average. Interesting future work includes **1)** the adaptation and precise analysis of our three algorithms to the computation of non-stationary policies—we recently showed that considering a variation of AMPI for computing non-stationary policies allows improving the $\frac{1}{(1-\gamma)^2}$ constant (Lesner and Scherrer, 2013)—and **2)** considering problems with large action spaces, for which the methods we have proposed here are likely to have limitation.

Acknowledgments

The experiments were conducted using Grid5000 (`https://www.grid5000.fr`).

Appendix A. Proof of Lemma 2

Before we start, we recall the following definitions:

$$\begin{aligned}
b_k &= v_k - T_{\pi_{k+1}} v_k, \\
d_k &= v_* - (T_{\pi_k})^m v_{k-1} = v_* - (v_k - \epsilon_k), \\
s_k &= (T_{\pi_k})^m v_{k-1} - v_{\pi_k} = (v_k - \epsilon_k) - v_{\pi_k}.
\end{aligned}$$

A.1 Bounding b_k

$$\begin{aligned}
b_k &= v_k - T_{\pi_{k+1}} v_k \\
&= v_k - T_{\pi_k} v_k + T_{\pi_k} v_k - T_{\pi_{k+1}} v_k \\
&\overset{(a)}{\leq} v_k - T_{\pi_k} v_k + \epsilon'_{k+1} \\
&= v_k - \epsilon_k - T_{\pi_k} v_k + \gamma P_{\pi_k} \epsilon_k + \epsilon_k - \gamma P_{\pi_k} \epsilon_k + \epsilon'_{k+1} \\
&\overset{(b)}{=} v_k - \epsilon_k - T_{\pi_k}(v_k - \epsilon_k) + (I - \gamma P_{\pi_k})\epsilon_k + \epsilon'_{k+1}. \qquad (25)
\end{aligned}$$

Using the definition of x_k, i.e.,

$$x_k \stackrel{\Delta}{=} (I - \gamma P_{\pi_k})\epsilon_k + \epsilon'_{k+1}, \tag{26}$$

we may write Equation 25 as

$$\begin{aligned} b_k &\leq v_k - \epsilon_k - T_{\pi_k}(v_k - \epsilon_k) + x_k \\ &\stackrel{(c)}{=} (T_{\pi_k})^m v_{k-1} - T_{\pi_k}(T_{\pi_k})^m v_{k-1} + x_k \\ &= (T_{\pi_k})^m v_{k-1} - (T_{\pi_k})^m (T_{\pi_k} v_{k-1}) + x_k \\ &\stackrel{(d)}{=} (\gamma P_{\pi_k})^m (v_{k-1} - T_{\pi_k} v_{k-1}) + x_k \\ &= (\gamma P_{\pi_k})^m b_{k-1} + x_k. \end{aligned} \tag{27}$$

(a) From the definition of ϵ'_{k+1}, we have $\forall \pi' \;\; T_{\pi'} v_k \leq T_{\pi_{k+1}} v_k + \epsilon'_{k+1}$, thus this inequality holds also for $\pi' = \pi_k$.
(b) This step is due to the fact that for every v and v', we have $T_{\pi_k}(v + v') = T_{\pi_k} v + \gamma P_{\pi_k} v'$.
(c) This is from the definition of ϵ_k, i.e., $v_k = (T_{\pi_k})^m v_{k-1} + \epsilon_k$.
(d) This step is due to the fact that for every v and v', any m, we have $(T_{\pi_k})^m v - (T_{\pi_k})^m v' = (\gamma P_{\pi_k})^m (v - v')$.

A.2 Bounding d_k

Define

$$g_{k+1} \stackrel{\Delta}{=} T_{\pi_{k+1}} v_k - (T_{\pi_{k+1}})^m v_k. \tag{28}$$

Then,

$$\begin{aligned} d_{k+1} &= v_* - (T_{\pi_{k+1}})^m v_k \\ &= T_{\pi_*} v_* - T_{\pi_*} v_k + T_{\pi_*} v_k - T_{\pi_{k+1}} v_k + T_{\pi_{k+1}} v_k - (T_{\pi_{k+1}})^m v_k \\ &\stackrel{(a)}{\leq} \gamma P_{\pi_*}(v_* - v_k) + \epsilon'_{k+1} + g_{k+1} \\ &= \gamma P_{\pi_*}(v_* - v_k) + \gamma P_{\pi_*} \epsilon_k - \gamma P_{\pi_*} \epsilon_k + \epsilon'_{k+1} + g_{k+1} \\ &\stackrel{(b)}{=} \gamma P_{\pi_*}\big(v_* - (v_k - \epsilon_k)\big) + y_k + g_{k+1} \\ &= \gamma P_{\pi_*} d_k + y_k + g_{k+1} \\ &\stackrel{(c)}{=} \gamma P_{\pi_*} d_k + y_k + \sum_{j=1}^{m-1} (\gamma P_{\pi_{k+1}})^j b_k. \end{aligned} \tag{29}$$

(a) This step is from the definition of ϵ'_{k+1} (see step **(a)** in bounding b_k) and that of g_{k+1} in Equation 28.
(b) This is from the definition of y_k, i.e.,

$$y_k \stackrel{\Delta}{=} -\gamma P_{\pi_*} \epsilon_k + \epsilon'_{k+1}. \tag{30}$$

(c) This step comes from rewriting g_{k+1} as

$$g_{k+1} = T_{\pi_{k+1}} v_k - (T_{\pi_{k+1}})^m v_k$$

$$
\begin{aligned}
&= \sum_{j=1}^{m-1} \left[(T_{\pi_{k+1}})^j v_k - (T_{\pi_{k+1}})^{j+1} v_k \right] \\
&= \sum_{j=1}^{m-1} \left[(T_{\pi_{k+1}})^j v_k - (T_{\pi_{k+1}})^j (T_{\pi_{k+1}} v_k) \right] \\
&= \sum_{j=1}^{m-1} (\gamma P_{\pi_{k+1}})^j (v_k - T_{\pi_{k+1}} v_k) \\
&= \sum_{j=1}^{m-1} (\gamma P_{\pi_{k+1}})^j b_k. \qquad (31)
\end{aligned}
$$

A.3 Bounding s_k

With some slight abuse of notation, we have

$$v_{\pi_k} = (T_{\pi_k})^\infty v_k$$

and thus:

$$
\begin{aligned}
s_k &= (T_{\pi_k})^m v_{k-1} - v_{\pi_k} \\
&\overset{(a)}{=} (T_{\pi_k})^m v_{k-1} - (T_{\pi_k})^\infty v_{k-1} \\
&= (T_{\pi_k})^m v_{k-1} - (T_{\pi_k})^m (T_{\pi_k})^\infty v_{k-1} \\
&= (\gamma P_{\pi_k})^m \big(v_{k-1} - (T_{\pi_k})^\infty v_{k-1} \big) \\
&= (\gamma P_{\pi_k})^m \sum_{j=0}^{\infty} \left[(T_{\pi_k})^j v_{k-1} - (T_{\pi_k})^{j+1} v_{k-1} \right] \\
&= (\gamma P_{\pi_k})^m \sum_{j=0}^{\infty} \left[(T_{\pi_k})^j v_{k-1} - (T_{\pi_k})^j T_{\pi_k} v_{k-1} \right] \\
&= (\gamma P_{\pi_k})^m \Big(\sum_{j=0}^{\infty} (\gamma P_{\pi_k})^j \Big) (v_{k-1} - T_{\pi_k} v_{k-1}) \\
&= (\gamma P_{\pi_k})^m (I - \gamma P_{\pi_k})^{-1} (v_{k-1} - T_{\pi_k} v_{k-1}) \\
&= (\gamma P_{\pi_k})^m (I - \gamma P_{\pi_k})^{-1} b_{k-1}. \qquad (32)
\end{aligned}
$$

(a) For any v, we have $v_{\pi_k} = (T_{\pi_k})^\infty v$. This step follows by setting $v = v_{k-1}$, i.e., $v_{\pi_k} = (T_{\pi_k})^\infty v_{k-1}$.

Appendix B. Proof of Lemma 4

We begin by focusing our analysis on AMPI. Here we are interested in bounding the loss $l_k = v_* - v_{\pi_k} = d_k + s_k$.

By induction, from Equations 27 and 29, we obtain

$$b_k \leq \sum_{i=1}^{k} \Gamma^{m(k-i)} x_i + \Gamma^{mk} b_0, \tag{33}$$

$$d_k \leq \sum_{j=0}^{k-1} \Gamma^{k-1-j} \Big(y_j + \sum_{l=1}^{m-1} \Gamma^l b_j \Big) + \Gamma^k d_0. \tag{34}$$

in which we have used the notation introduced in Definition 3. In Equation 34, we also used the fact that from Equation 31, we may write $g_{k+1} = \sum_{j=1}^{m-1} \Gamma^j b_k$. Moreover, we may rewrite Equation 32 as

$$s_k = \Gamma^m \sum_{j=0}^{\infty} \Gamma^j b_{k-1} = \sum_{j=0}^{\infty} \Gamma^{m+j} b_{k-1}. \tag{35}$$

B.1 Bounding l_k

From Equations 33 and 34, we may write

$$\begin{aligned} d_k &\leq \sum_{j=0}^{k-1} \Gamma^{k-1-j} \left(y_j + \sum_{l=1}^{m-1} \Gamma^l \Big(\sum_{i=1}^{j} \Gamma^{m(j-i)} x_i + \Gamma^{mj} b_0 \Big) \right) + \Gamma^k d_0 \\ &= \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{j=0}^{k-1} \sum_{l=1}^{m-1} \sum_{i=1}^{j} \Gamma^{k-1-j+l+m(j-i)} x_i + z_k, \end{aligned} \tag{36}$$

where we used the following definition

$$z_k \stackrel{\Delta}{=} \sum_{j=0}^{k-1} \sum_{l=1}^{m-1} \Gamma^{k-1+l+j(m-1)} b_0 + \Gamma^k d_0 = \sum_{i=k}^{mk-1} \Gamma^i b_0 + \Gamma^k d_0.$$

The triple sum involved in Equation 36 may be written as

$$\begin{aligned} \sum_{j=0}^{k-1} \sum_{l=1}^{m-1} \sum_{i=1}^{j} \Gamma^{k-1-j+l+m(j-i)} x_i &= \sum_{i=1}^{k-1} \sum_{j=i}^{k-1} \sum_{l=1}^{m-1} \Gamma^{k-1+l+j(m-1)-mi} x_i \\ &= \sum_{i=1}^{k-1} \sum_{j=mi+k-i}^{mk-1} \Gamma^{j-mi} x_i \\ &= \sum_{i=1}^{k-1} \sum_{j=k-i}^{m(k-i)-1} \Gamma^{j} x_i \\ &= \sum_{i=1}^{k-1} \sum_{j=i}^{mi-1} \Gamma^{j} x_{k-i}. \end{aligned} \tag{37}$$

Using Equation 37, we may write Equation 36 as

$$d_k \leq \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \sum_{j=i}^{mi-1} \Gamma^j x_{k-i} + z_k. \tag{38}$$

Similarly, from Equations 35 and 33, we have

$$\begin{aligned}
s_k &\leq \sum_{j=0}^{\infty} \Gamma^{m+j} \Big(\sum_{i=1}^{k-1} \Gamma^{m(k-1-i)} x_i + \Gamma^{m(k-1)} b_0 \Big) \\
&= \sum_{j=0}^{\infty} \Big(\sum_{i=1}^{k-1} \Gamma^{m+j+m(k-1-i)} x_i + \Gamma^{m+j+m(k-1)} b_0 \Big) \\
&= \sum_{i=1}^{k-1} \sum_{j=0}^{\infty} \Gamma^{j+m(k-i)} x_i + \sum_{j=0}^{\infty} \Gamma^{j+mk} b_0 = \sum_{i=1}^{k-1} \sum_{j=0}^{\infty} \Gamma^{j+mi} x_{k-i} + \sum_{j=mk}^{\infty} \Gamma^j b_0 \\
&= \sum_{i=1}^{k-1} \sum_{j=mi}^{\infty} \Gamma^j x_{k-i} + z'_k,
\end{aligned} \tag{39}$$

where we used the following definition

$$z'_k \stackrel{\Delta}{=} \sum_{j=mk}^{\infty} \Gamma^j b_0.$$

Finally, using the bounds in Equations 38 and 39, we obtain the following bound on the loss

$$\begin{aligned}
l_k &\leq d_k + s_k \\
&\leq \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \Big(\sum_{j=i}^{mi-1} \Gamma^j + \sum_{j=mi}^{\infty} \Gamma^j \Big) x_{k-i} + z_k + z'_k \\
&= \sum_{i=1}^{k} \Gamma^{i-1} y_{k-i} + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j x_{k-i} + \eta_k,
\end{aligned} \tag{40}$$

where we used the following definition

$$\eta_k \stackrel{\Delta}{=} z_k + z'_k = \sum_{j=k}^{\infty} \Gamma^j b_0 + \Gamma^k d_0. \tag{41}$$

Note that we have the following relation between b_0 and d_0

$$\begin{aligned}
b_0 &= v_0 - T_{\pi_1} v_0 \\
&= v_0 - v_* + T_{\pi_*} v_* - T_{\pi_*} v_0 + T_{\pi_*} v_0 - T_{\pi_1} v_0 \\
&\leq (I - \gamma P_{\pi_*})(-d_0) + \epsilon'_1,
\end{aligned} \tag{42}$$

In Equation 42, we used the fact that $v_* = T_{\pi_*} v_*$, $\epsilon_0 = 0$, and $T_{\pi_*} v_0 - T_{\pi_1} v_0 \leq \epsilon'_1$ (this is because the policy π_1 is ϵ'_1-greedy w.r.t. v_0). As a result, we may write $|\eta_k|$ either as

$$\begin{aligned}
|\eta_k| &\leq \sum_{j=k}^{\infty} \Gamma^j \left[(I - \gamma P_{\pi_*}) |d_0| + |\epsilon'_1| \right] + \Gamma^k |d_0| \\
&\leq \sum_{j=k}^{\infty} \Gamma^j \left[(I + \Gamma^1) |d_0| + |\epsilon'_1| \right] + \Gamma^k |d_0| \\
&= 2 \sum_{j=k}^{\infty} \Gamma^j |d_0| + \sum_{j=k}^{\infty} \Gamma^j |\epsilon'_1|, \qquad (43)
\end{aligned}$$

or using the fact that from Equation 42, we have $d_0 \leq (I - \gamma P_{\pi_*})^{-1} (-b_0 + \epsilon'_1)$, as

$$\begin{aligned}
|\eta_k| &\leq \sum_{j=k}^{\infty} \Gamma^j |b_0| + \Gamma^k \sum_{j=0}^{\infty} (\gamma P_{\pi_*})^j \left(|b_0| + |\epsilon'_1| \right) \\
&= \sum_{j=k}^{\infty} \Gamma^j |b_0| + \Gamma^k \sum_{j=0}^{\infty} \Gamma^j \left(|b_0| + |\epsilon'_1| \right) \\
&= 2 \sum_{j=k}^{\infty} \Gamma^j |b_0| + \sum_{j=k}^{\infty} \Gamma^j |\epsilon'_1|. \qquad (44)
\end{aligned}$$

Now, using the definitions of x_k and y_k in Equations 26 and 30, the bound on $|\eta_k|$ in Equation 43 or 44, and the fact that $\epsilon_0 = 0$, we obtain

$$\begin{aligned}
|l_k| &\leq \sum_{i=1}^{k} \Gamma^{i-1} \left[\Gamma^1 |\epsilon_{k-i}| + |\epsilon'_{k-i+1}| \right] + \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j \left[(I + \Gamma^1) |\epsilon_{k-i}| + |\epsilon'_{k-i+1}| \right] + |\eta_k| \\
&= \sum_{i=1}^{k-1} \left(\Gamma^i + \sum_{j=i}^{\infty} (\Gamma^j + \Gamma^{j+1}) \right) |\epsilon_{k-i}| + \Gamma^k |\epsilon_0| \qquad (45) \\
&\quad + \sum_{i=1}^{k-1} \left(\Gamma^{i-1} + \sum_{j=i}^{\infty} \Gamma^j \right) |\epsilon'_{k-i+1}| + \Gamma^{k-1} |\epsilon'_1| + \sum_{j=k}^{\infty} \Gamma^j |\epsilon'_1| + h(k) \\
&= 2 \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon_{k-i}| + \sum_{i=1}^{k-1} \sum_{j=i-1}^{\infty} \Gamma^j |\epsilon'_{k-i+1}| + \sum_{j=k-1}^{\infty} \Gamma^j |\epsilon'_1| + h(k) \\
&= 2 \sum_{i=1}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon_{k-i}| + \sum_{i=0}^{k-1} \sum_{j=i}^{\infty} \Gamma^j |\epsilon'_{k-i}| + h(k), \qquad (46)
\end{aligned}$$

where we used the following definition

$$h(k) \triangleq 2 \sum_{j=k}^{\infty} \Gamma^j |d_0| \quad \text{or} \quad h(k) \triangleq 2 \sum_{j=k}^{\infty} \Gamma^j |b_0|,$$

depending on whether one uses Equation 43 or Equation 44.

We end this proof by adapting the error propagation to CBMPI. As expressed by Equations 20 and 21 in Section 4, an analysis of CBMPI can be deduced from that we have just done by replacing v_k with the auxiliary variable $w_k = (T_{\pi_k})^m v_{k-1}$ and ϵ_k with $(\gamma P_{\pi_k})^m \epsilon_{k-1} = \Gamma^m \epsilon_{k-1}$. Therefore, using the fact that $\epsilon_0 = 0$, we can rewrite the bound of Equation 46 for CBMPI as follows:

$$\begin{aligned} l_k &\leq 2\sum_{i=1}^{k-1}\sum_{j=i}^{\infty}\Gamma^{j+m}|\epsilon_{k-i-1}| + \sum_{i=0}^{k-1}\sum_{j=i}^{\infty}\Gamma^j|\epsilon'_{k-i}| + h(k) \\ &= 2\sum_{i=1}^{k-2}\sum_{j=m+i}^{\infty}\Gamma^j|\epsilon_{k-i-1}| + \sum_{i=0}^{k-1}\sum_{j=i}^{\infty}\Gamma^j|\epsilon'_{k-i}| + h(k). \end{aligned} \tag{47}$$

Appendix C. Proof of Lemma 6

For any integer t and vector z, the definition of Γ^t and Hölder's inequality imply that

$$\rho\Gamma^t|z| = \left\|\Gamma^t|z|\right\|_{1,\rho} \leq \gamma^t c_q(t)\|z\|_{q',\mu} = \gamma^t c_q(t)\left(\mu|z|^{q'}\right)^{\frac{1}{q'}}. \tag{48}$$

We define

$$K \triangleq \sum_{l=1}^{n}\xi_l\left(\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\gamma^j\right),$$

where $\{\xi_l\}_{l=1}^n$ is a set of non-negative numbers that we will specify later. We now have

$$\begin{aligned} \|f\|_{p,\rho}^p &= \rho|f|^p \\ &\leq K^p\rho\left(\frac{\sum_{l=1}^n\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\Gamma^j|g_i|}{K}\right)^p = K^p\rho\left(\frac{\sum_{l=1}^n\xi_l\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\Gamma^j\left(\frac{|g_i|}{\xi_l}\right)}{K}\right)^p \\ &\overset{(a)}{\leq} K^p\rho\frac{\sum_{l=1}^n\xi_l\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\Gamma^j\left(\frac{|g_i|}{\xi_l}\right)^p}{K} = K^p\frac{\sum_{l=1}^n\xi_l\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\rho\Gamma^j\left(\frac{|g_i|}{\xi_l}\right)^p}{K} \\ &\overset{(b)}{\leq} K^p\frac{\sum_{l=1}^n\xi_l\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\gamma^j c_q(j)\left(\mu\left(\frac{|g_i|}{\xi_l}\right)^{pq'}\right)^{\frac{1}{q'}}}{K} \\ &= K^p\frac{\sum_{l=1}^n\xi_l\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\gamma^j c_q(j)\left(\frac{\|g_i\|_{pq',\mu}}{\xi_l}\right)^p}{K} \\ &\leq K^p\frac{\sum_{l=1}^n\xi_l\left(\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\gamma^j c_q(j)\right)\left(\frac{\sup_{i\in\mathcal{I}_l}\|g_i\|_{pq',\mu}}{\xi_l}\right)^p}{K} \\ &\overset{(c)}{=} K^p\frac{\sum_{l=1}^n\xi_l\left(\sum_{i\in\mathcal{I}_l}\sum_{j\in\mathcal{J}_i}\gamma^j\right)\mathcal{C}_q(l)\left(\frac{\sup_{i\in\mathcal{I}_l}\|g_i\|_{pq',\mu}}{\xi_l}\right)^p}{K}, \end{aligned}$$

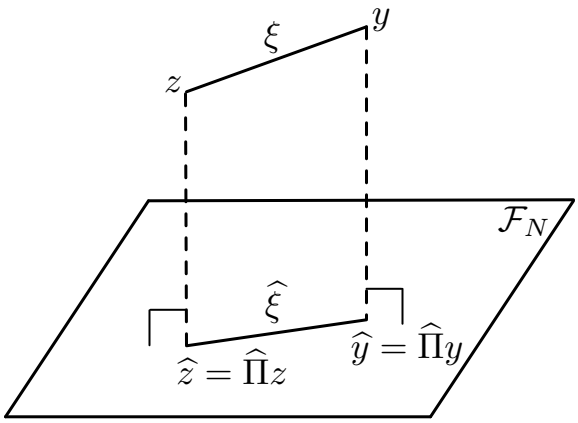


Figure 11: The notations used in the proof.

where **(a)** results from Jensen's inequality, **(b)** from Equation 48, and **(c)** from the definition of $\mathcal{C}_q(l)$. Now, by setting $\xi_l = \big(\mathcal{C}_q(l)\big)^{1/p} \sup_{i \in \mathcal{I}_l} \|g_i\|_{pq',\mu}$, we obtain

$$\|f\|_{p,\rho}^p \leq K^p \frac{\sum_{l=1}^n \xi_l \left(\sum_{i \in \mathcal{I}_l} \sum_{j \in \mathcal{J}_i} \gamma^j \right)}{K} = K^p,$$

where the last step follows from the definition of K.

Appendix D. Proof of Lemma 11

Let $\widehat{\mu}$ be the empirical distribution corresponding to states $s^{(1)}, \ldots, s^{(n)}$. Let us define two N-dimensional vectors $z = \Big(\big[(T_{\pi_k})^m v_{k-1}\big](s^{(1)}), \ldots, \big[(T_{\pi_k})^m v_{k-1}\big](s^{(N)}) \Big)^\top$ and $y = \big(\widehat{v}_k(s^{(1)}), \ldots, \widehat{v}_k(s^{(N)})\big)^\top$ and their orthogonal projections onto the vector space $\mathcal{F}_N$ as $\widehat{z} = \widehat{\Pi} z$ and $\widehat{y} = \widehat{\Pi} y = \big(\widetilde{v}_k(s^{(1)}), \ldots, \widetilde{v}_k(s^{(N)})\big)^\top$, where $\widetilde{v}_k$ is the result of linear regression and its truncation (by $V_{\max}$) is v_k, i.e., $v_k = \mathbb{T}(\widetilde{v}_k)$ (see Figure 11). What we are interested in is to find a bound on the regression error $\|z - \widehat{y}\|$ (the difference between the target function z and the result of the regression $\widehat{y}$). We may decompose this error as

$$\|z - \widehat{y}\|_{2,\widehat{\mu}} \leq \|\widehat{z} - \widehat{y}\|_{2,\widehat{\mu}} + \|z - \widehat{z}\|_{2,\widehat{\mu}} = \|\widehat{\xi}\|_{2,\widehat{\mu}} + \|z - \widehat{z}\|_{2,\widehat{\mu}}, \tag{49}$$

where $\widehat{\xi} = \widehat{z} - \widehat{y}$ is the projected noise (estimation error) $\widehat{\xi} = \widehat{\Pi}\xi$, with the noise vector $\xi = z - y$ defined as $\xi_i = \big[(T_{\pi_k})^m v_{k-1}\big](s^{(i)}) - \widehat{v}_k(s^{(i)})$. It is easy to see that noise is zero mean, i.e., $\mathbb{E}[\xi_i] = 0$ and is bounded by $2V_{\max}$, i.e., $|\xi_i| \leq 2V_{\max}$. We may write the estimation error as

$$\|\widehat{z} - \widehat{y}\|_{2,\widehat{\mu}}^2 = \|\widehat{\xi}\|_{2,\widehat{\mu}}^2 = \langle \widehat{\xi}, \widehat{\xi} \rangle = \langle \xi, \widehat{\xi} \rangle,$$

where the last equality follows from the fact that $\widehat{\xi}$ is the orthogonal projection of ξ. Since $\widehat{\xi} \in \mathcal{F}_N$, let $f_\alpha \in \mathcal{F}$ be any function in the function space $\mathcal{F}$,[13] whose values at $\{s^{(i)}\}_{i=1}^N$

13. We should discriminate between the linear function space $\mathcal{F} = \big\{ f_\alpha \mid \alpha \in \mathbb{R}^d \text{ and } f_\alpha(\cdot) = \phi(\cdot)^\top \alpha \big\}$, where $\phi(\cdot) = \big(\varphi_1(\cdot), \ldots, \varphi_d(\cdot)\big)^\top$, and its corresponding linear vector space $\mathcal{F}_N = \{\Phi\alpha,\ \alpha \in \mathbb{R}^d\} \subset \mathbb{R}^N$, where $\Phi = \big[\phi(s^{(1)})^\top; \ldots; \phi(s^{(N)})^\top\big]$.

equals to $\{\widehat{\xi}_i\}_{i=1}^N$. By application of a variation of Pollard's inequality (Györfi et al., 2002), we obtain

$$\langle \xi, \widehat{\xi} \rangle = \frac{1}{N} \sum_{i=1}^{N} \xi_i f_\alpha(s^{(i)}) \leq 4V_{\max} \|\widehat{\xi}\|_{2,\widehat{\mu}} \sqrt{\frac{2}{N} \log\left(\frac{3(9e^2N)^{d+1}}{\delta'}\right)},$$

with probability at least $1-\delta'$. Thus, we have

$$\|\widehat{z} - \widehat{y}\|_{2,\widehat{\mu}} = \|\widehat{\xi}\|_{2,\widehat{\mu}} \leq 4V_{\max} \sqrt{\frac{2}{N} \log\left(\frac{3(9e^2N)^{d+1}}{\delta'}\right)}. \tag{50}$$

From Equations 49 and 50, we have

$$\|(T_{\pi_k})^m v_{k-1} - \widetilde{v}_k\|_{2,\widehat{\mu}} \leq \|(T_{\pi_k})^m v_{k-1} - \widehat{\Pi}(T_{\pi_k})^m v_{k-1}\|_{2,\widehat{\mu}} + 4V_{\max} \sqrt{\frac{2}{N} \log\left(\frac{3(9e^2N)^{d+1}}{\delta'}\right)}. \tag{51}$$

Now in order to obtain a random design bound, we first define $f_{\widehat{\alpha}_*} \in \mathcal{F}$ as $f_{\widehat{\alpha}_*}(s^{(i)}) = \big[\widehat{\Pi}(T_{\pi_k})^m v_{k-1}\big](s^{(i)})$, and then define $f_{\alpha_*} = \Pi(T_{\pi_k})^m v_{k-1}$ that is the best approximation (w.r.t. μ) of the target function $(T_{\pi_k})^m v_{k-1}$ in $\mathcal{F}$. Since $f_{\widehat{\alpha}_*}$ is the minimizer of the empirical loss, any function in $\mathcal{F}$ different than $f_{\widehat{\alpha}_*}$ has a bigger empirical loss, thus we have

$$\begin{aligned} \|f_{\widehat{\alpha}_*} - (T_{\pi_k})^m v_{k-1}\|_{2,\widehat{\mu}} &\leq \|f_{\alpha_*} - (T_{\pi_k})^m v_{k-1}\|_{2,\widehat{\mu}} \\ &\leq 2\|f_{\alpha_*} - (T_{\pi_k})^m v_{k-1}\|_{2,\mu} \\ &\quad + 12\Big(V_{\max} + \|\alpha_*\|_2 \sup_x \|\phi(x)\|_2\Big)\sqrt{\frac{2}{N}\log\frac{3}{\delta'}} \end{aligned} \tag{52}$$

with probability at least $1-\delta'$, where the second inequality is the application of a variation of Theorem 11.2 in Györfi et al. (2002) with $\|f_{\alpha_*} - (T_{\pi_k})^m v_{k-1}\|_\infty \leq V_{\max} + \|\alpha^*\|_2 \sup_x \|\phi(x)\|_2$. Similarly, we can write the left-hand-side of Equation 51 as

$$\begin{aligned} 2\|(T_{\pi_k})^m v_{k-1} - \widetilde{v}_k\|_{2,\widehat{\mu}} &\geq 2\|(T_{\pi_k})^m v_{k-1} - \mathbb{T}(\widetilde{v}_k)\|_{2,\widehat{\mu}} \\ &\geq \|(T_{\pi_k})^m v_{k-1} - \mathbb{T}(\widetilde{v}_k)\|_{2,\mu} - 24V_{\max}\sqrt{\frac{2}{N}\Lambda(N,d,\delta')} \end{aligned} \tag{53}$$

with probability at least $1-\delta'$, where $\Lambda(N,d,\delta') = 2(d+1)\log N + \log\frac{e}{\delta'} + \log\big(9(12e)^{2(d+1)}\big)$. Putting together Equations 51, 52, and 53 and using the fact that $\mathbb{T}(\widetilde{v}_k) = v_k$, we obtain

$$\begin{aligned} \|\epsilon_k\|_{2,\mu} &= \|(T_{\pi_k})^m v_{k-1} - v_k\|_{2,\mu} \\ &\leq 2\Bigg(2\|(T_{\pi_k})^m v_{k-1} - f_{\alpha_*}\|_{2,\mu} \\ &\quad + 12\Big(V_{\max} + \|\alpha_*\|_2 \sup_x \|\phi(x)\|_2\Big)\sqrt{\frac{2}{N}\log\frac{3}{\delta'}} + 4V_{\max}\sqrt{\frac{2}{N}\log\left(\frac{3(9e^2N)^{d+1}}{\delta'}\right)}\Bigg) \\ &\quad + 24V_{\max}\sqrt{\frac{2}{N}\Lambda(N,d,\delta')}. \end{aligned}$$

The result follows by setting $\delta = 3\delta'$ and some simplifications.

Appendix E. Proof of Lemma 12

Proof We prove the following series of inequalities:

$$
\begin{aligned}
||\epsilon'_k||_{1,\mu} &\overset{(a)}{\leq} ||\epsilon'_k||_{1,\widehat{\mu}} + e'_3(N,\delta') && \text{w.p. } 1-\delta' \\
&\overset{(b)}{=} \frac{1}{N}\sum_{i=1}^{N}\left[\max_{a\in\mathcal{A}}\big(T_a v_{k-1}\big)(s^{(i)}) - \big(T_{\pi_k} v_{k-1}\big)(s^{(i)})\right] + e'_3(N,\delta') \\
&\overset{(c)}{\leq} \frac{1}{N}\sum_{i=1}^{N}\left[\max_{a\in\mathcal{A}}\big(T_a v_{k-1}\big)(s^{(i)}) - \frac{1}{M}\sum_{j=1}^{M}\big(\widehat{T}^{(j)}_{\pi_k} v_{k-1}\big)(s^{(i)})\right] + e'_3(N,\delta') + e'_4(N,M,\delta') && \text{w.p. } 1-2\delta' \\
&\overset{(d)}{=} \frac{1}{N}\sum_{i=1}^{N}\left[\max_{a\in\mathcal{A}}\big(T_a v_{k-1}\big)(s^{(i)}) - \max_{a'\in\mathcal{A}}\frac{1}{M}\sum_{j=1}^{M}\big(\widehat{T}^{(j)}_{a'} v_{k-1}\big)(s^{(i)})\right] + e'_3(N,\delta') + e'_4(N,M,\delta') \\
&\overset{(e)}{\leq} \frac{1}{N}\sum_{i=1}^{N}\left\{\max_{a\in\mathcal{A}}\left[\big(T_a v_{k-1}\big)(s^{(i)}) - \frac{1}{M}\sum_{j=1}^{M}\big(\widehat{T}^{(j)}_{a} v_{k-1}\big)(s^{(i)})\right]\right\} + e'_3(N,\delta') + e'_4(N,M,\delta') \\
&\overset{(f)}{\leq} e'_3(N,\delta') + e'_4(N,M,\delta') + e'_5(M,N,\delta') && \text{w.p. } 1-3\delta'
\end{aligned}
$$

(a) This step is the result of the following lemma.

Lemma 17 *Let Π be the policy space of the policies obtained by Equation 4 from the truncation (by $V_{\max}$) of the function space $\mathcal{F}$, with finite VC-dimension $h = VC(\Pi) < \infty$. Let $N > 0$ be the number of states in the rollout set $\mathcal{D}_k$, drawn i.i.d. from the state distribution μ. Then, we have*

$$
\mathbb{P}_{\mathcal{D}_k}\left[\sup_{\pi\in\Pi}\Big|||\epsilon'_k(\pi)||_{1,\widehat{\mu}} - ||\epsilon'_k(\pi)||_{1,\mu}\Big| > e'_3(N,\delta')\right] \leq \delta' \,,
$$

with $e'_3(N,\delta') = 16 V_{\max}\sqrt{\frac{2}{N}(h\log\frac{eN}{h} + \log\frac{8}{\delta'})}$.

Proof The proof is similar to the proof of Lemma 1 in Lazaric et al. (2010c). ■

(b) This is from the definition of $||\epsilon'_k||_{1,\widehat{\mu}}$.

(c) This step is the result of bounding

$$
\sup_{\pi\in\Pi}\left[\frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big(\widehat{T}^{(j)}_{\pi} v_{k-1}\big)(s^{(i)}) - \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}\big(T_{\pi} v_{k-1}\big)(s^{(i)})\right]
$$

by $e'_4(N,M,\delta')$. The supremum over all the policies in the policy space Π is due to the fact that π_k is a random object whose randomness comes from all the randomly generated samples at the k'th iteration (i.e., the states in the rollout set and all the generated rollouts). We bound this term using the following lemma.

Lemma 18 *Let Π be the policy space of the policies obtained by Equation 4 from the truncation (by $V_{\max}$) of the function space $\mathcal{F}$, with finite VC-dimension $h = VC(\Pi) < \infty$. Let $\{s^{(i)}\}_{i=1}^{N}$ be N states sampled i.i.d. from the distribution μ. For each sampled state $s^{(i)}$,*

we take the action suggested by policy π, M times, and observe the next states $\{s^{(i,j)}\}_{j=1}^{M}$. Then, we have

$$\mathbb{P}\left[\sup_{\pi\in\Pi}\left|\frac{1}{N}\sum_{i=1}^{N}\frac{1}{M}\sum_{j=1}^{M}\left[r\big(s^{(i)},\pi(s^{(i)})\big)+\gamma v_{k-1}\big(s^{(i,j)}\big)\right]-\frac{1}{N}\sum_{i=1}^{N}\big(T_\pi v_{k-1}\big)(s^{(i)})\right|>e_4'(N,M,\delta')\right]\leq\delta'\,,$$

with $e_4'(N,M,\delta')=8V_{\max}\sqrt{\frac{2}{MN}\big(h\log\frac{eMN}{h}+\log\frac{8}{\delta'}\big)}$.

Proof The proof is similar to the proof of Lemma 4 in Lazaric et al. (2010a). ■

(d) This step is from the definition of π_k in the AMPI-V algorithm (Equation 4).

(e) This step is algebra, replacing two maximums with one.

(f) This step follows from applying Chernoff-Hoeffding to bound

$$\big(T_{a_*^{(i)}}v_{k-1}\big)(s^{(i)})-\frac{1}{M}\sum_{j=1}^{M}\big(\widehat{T}_{a_*^{(i)}}^{(j)}v_{k-1}\big)(s^{(i)}),$$

for each $i=1,\ldots,N$, by $e_5'(M,\delta'')=V_{\max}\sqrt{\frac{2\log(1/\delta'')}{M}}$, followed by a union bound, which gives us $e_5'(M,N,\delta')=V_{\max}\sqrt{\frac{2\log(N/\delta')}{M}}$. Note that the fixed action $a_*^{(i)}$ is defined as

$$a_*^{(i)}=\operatorname*{argmax}_{a\in\mathcal{A}}\Big[\big(T_a v_{k-1}\big)(s^{(i)})-\frac{1}{M}\sum_{j=1}^{M}\big(\widehat{T}_a^{(j)}v_{k-1}\big)(s^{(i)})\Big].$$

The final statement of the theorem follows by setting $\delta=3\delta'$. ■

Appendix F. Proof of Lemma 13

The proof of this lemma is similar to the proof of Theorem 1 in Lazaric et al. (2010c). Before stating the proof, we report the following two lemmas that are used in the proof.

Lemma 19 *Let Π be a policy space with finite VC-dimension $h=VC(\Pi)<\infty$ and N' be the number of states in the rollout set $\mathcal{D}'_{k-1}$ drawn i.i.d. from the state distribution μ. Then we have*

$$\mathbb{P}_{\mathcal{D}'_{k-1}}\left[\sup_{\pi\in\Pi}\left|\mathcal{L}_{k-1}^{\Pi}(\widehat{\mu};\pi)-\mathcal{L}_{k-1}^{\Pi}(\mu;\pi)\right|>\epsilon\right]\leq\delta\,,$$

with $\epsilon=16Q_{\max}\sqrt{\frac{2}{N'}\big(h\log\frac{eN'}{h}+\log\frac{8}{\delta}\big)}$.

Proof This is a restatement of Lemma 1 in Lazaric et al. (2010c). ■

Lemma 20 *Let Π be a policy space with finite VC-dimension $h = VC(\Pi) < \infty$ and $s^{(1)}, \dots, s^{(N')}$ be an arbitrary sequence of states. Assume that at each state, we simulate M independent rollouts. We have*

$$\mathbb{P}\left[\sup_{\pi\in\Pi}\Big|\frac{1}{N'}\sum_{i=1}^{N'}\frac{1}{M}\sum_{j=1}^{M}R^j_{k-1}\big(s^{(i,j)},\pi(s^{(i,j)})\big)-\frac{1}{N'}\sum_{i=1}^{N'}Q_{k-1}\big(s^{(i,j)},\pi(s^{(i,j)})\big)\Big|>\epsilon\right]\le\delta\,,$$

with $\epsilon = 8Q_{\max}\sqrt{\frac{2}{MN'}\big(h\log\frac{eMN'}{h}+\log\frac{8}{\delta}\big)}$.

Proof The proof is similar to the one for Lemma 19. ■

Proof (Lemma 13) Let $a^*(\cdot)\in\operatorname{argmax}_{a\in\mathcal{A}}Q_{k-1}(\cdot,a)$ be a greedy action. To simplify the notation, we remove the dependency of a^* on states and use a^* instead of $a^*(s^{(i)})$ in the following. We prove the following series of inequalities:

$$\begin{aligned}
\mathcal{L}^{\Pi}_{k-1}(\mu;\pi_k) &\overset{(a)}{\le} \mathcal{L}^{\Pi}_{k-1}(\widehat{\mu};\pi_k)+e'_1(N',\delta) && \text{w.p. } 1-\delta'\\
&= \frac{1}{N'}\sum_{i=1}^{N'}\Big[Q_{k-1}(s^{(i)},a^*)-Q_{k-1}\big(s^{(i)},\pi_k(s^{(i)})\big)\Big]+e'_1(N',\delta)\\
&\overset{(b)}{\le} \frac{1}{N'}\sum_{i=1}^{N'}\Big[Q_{k-1}(s^{(i)},a^*)-\widehat{Q}_{k-1}\big(s^{(i)},\pi_k(s^{(i)})\big)\Big]+e'_1(N',\delta)+e'_2(N',M,\delta) && \text{w.p. } 1-2\delta'\\
&\overset{(c)}{\le} \frac{1}{N'}\sum_{i=1}^{N'}\Big[Q_{k-1}(s^{(i)},a^*)-\widehat{Q}_{k-1}\big(s^{(i)},\tilde{\pi}(s^{(i)})\big)\Big]+e'_1(N',\delta)+e'_2(N',M,\delta)\\
&\overset{(b)}{\le} \frac{1}{N'}\sum_{i=1}^{N'}\Big[Q_{k-1}(s^{(i)},a^*)-Q_{k-1}\big(s^{(i)},\tilde{\pi}(s^{(i)})\big)\Big]+e'_1(N',\delta)+2e'_2(N',M,\delta) && \text{w.p. } 1-3\delta'\\
&= \mathcal{L}^{\Pi}_{k-1}(\widehat{\mu};\tilde{\pi})+e'_1(N',\delta)+2e'_2(N',M,\delta)\\
&\overset{(a)}{\le} \mathcal{L}^{\Pi}_{k-1}(\mu;\tilde{\pi})+2\big(e'_1(N',\delta)+e'_2(N',M,\delta)\big) && \text{w.p. } 1-4\delta'\\
&= \inf_{\pi\in\Pi}\mathcal{L}^{\Pi}_{k-1}(\mu;\pi)+2\big(e'_1(N',\delta)+e'_2(N',M,\delta)\big).
\end{aligned}$$

The statement of the theorem is obtained by setting $\delta' = \delta/4$.

(a) This follows from Lemma 19.
(b) Here we introduce the estimated action-value function $\widehat{Q}_{k-1}$ by bounding

$$\sup_{\pi\in\Pi}\left[\frac{1}{N'}\sum_{i=1}^{N'}\widehat{Q}_{k-1}\big(s^{(i)},\pi(s^{(i)})\big)-\frac{1}{N'}\sum_{i=1}^{N'}Q_{k-1}\big(s^{(i)},\pi(s^{(i)})\big)\right]$$

using Lemma 20.
(c) From the definition of π_k in CBMPI, we have

$$\pi_k = \operatorname*{argmin}_{\pi\in\Pi}\widehat{\mathcal{L}}^{\Pi}_{k-1}(\widehat{\mu};\pi) = \operatorname*{argmax}_{\pi\in\Pi}\frac{1}{N'}\sum_{i=1}^{N'}\widehat{Q}_{k-1}\big(s^{(i)},\pi(s^{(i)})\big),$$

thus, $-1/N' \sum_{i=1}^{N'} \widehat{Q}_{k-1}\big(s^{(i)}, \pi_k(s^{(i)})\big)$ can be maximized by replacing π_k with any other policy, particularly with

$$\tilde{\pi} = \operatorname*{argmin}_{\pi \in \Pi} \int_{\mathcal{S}} \left(\max_{a \in \mathcal{A}} Q_{k-1}(s,a) - Q_{k-1}\big(s, \pi(s)\big) \right) \mu(ds).$$

■

References

A. Antos, R. Munos, and Cs. Szepesvári. Fitted q-iteration in continuous action-space MDPs. In *Proceedings of the Advances in Neural Information Processing Systems 19*, pages 9–16, 2007.

D. Bertsekas and S. Ioffe. Temporal differences-based policy iteration and applications in neuro-dynamic programming. Technical report, MIT, 1996.

D. Bertsekas and J. Tsitsiklis. *Neuro-dynamic programming*. Athena Scientific, 1996.

H. Burgiel. How to lose at tetris. *Mathematical Gazette*, 81:194–200, 1997.

Pelin Canbolat and Uriel Rothblum. (Approximate) iterated successive approximations algorithm for sequential decision processes. *Annals of Operations Research*, pages 1–12, 2012. ISSN 0254-5330.

C. Chang and C. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011. ISSN 2157-6904. doi: 10.1145/1961189.1961199. URL `http://doi.acm.org/10.1145/1961189.1961199`.

E. Demaine, S. Hohenberger, and D. Liben-Nowell. Tetris is hard, even to approximate. In *Proceedings of the Ninth International Computing and Combinatorics Conference*, pages 351–363, 2003.

C. Dimitrakakis and M. Lagoudakis. Rollout sampling approximate policy iteration. *Machine Learning Journal*, 72(3):157–171, 2008.

D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005.

C. Fahey. Tetris AI, computer plays tetris, 2003. `http://colinfahey.com/tetris/tetris.html`.

A. Farahmand, R. Munos, and Cs. Szepesvári. Error propagation for approximate policy and value iteration. In *Proceedings of the Advances in Neural Information Processing Systems 22*, pages 568–576, 2010.

V. Farias and B. Van Roy. *Tetris: A study of randomized constraint sampling*. Springer-Verlag, 2006.

A. Fern, S. Yoon, and R. Givan. Approximate policy iteration with a policy language bias: Solving relational Markov decision processes. *Journal of Artificial Intelligence Research*, 25:75–118, 2006.

T. Furmston and D. Barber. A unifying perspective of parametric policy search methods for Markov decision processes. In *Proceedings of the Advances in Neural Information Processing Systems 24*, pages 2726–2734, 2012.

V. Gabillon, A. Lazaric, M. Ghavamzadeh, and B. Scherrer. Classification-based policy iteration with a critic. In *Proceedings of the Twenty-Eighth International Conference on Machine Learning*, pages 1049–1056, 2011.

V. Gabillon, M. Ghavamzadeh, and B. Scherrer. Approximate dynamic programming finally performs well in the game of tetris. In *Proceedings of Advances in Neural Information Processing Systems 26*, pages 1754–1762, 2013.

M. Geist and B. Scherrer. Off-policy learning with eligibility traces: A survey. *Journal of Machine Learning Research*, 14, April 2014.

G.J. Gordon. Stable function approximation in dynamic programming. In *ICML*, pages 261–268, 1995.

L. Györfi, M. Kolher, M. Krzyżak, and H. Walk. *A distribution-free theory of nonparametric regression*. Springer-Verlag, 2002.

N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9:159–195, 2001.

S. Kakade. A natural policy gradient. In *Proceedings of the Advances in Neural Information Processing Systems 14*, pages 1531–1538, 2002.

M. Kearns, Y. Mansour, and A. Ng. Approximate planning in large pomdps via reusable trajectories. In *Proceedings of the Advances in Neural Information Processing Systems 12*, pages 1001–1007. MIT Press, 2000.

M. Lagoudakis and R. Parr. Least-squares policy iteration. *Journal of Machine Learning Research*, 4:1107–1149, 2003a.

M. Lagoudakis and R. Parr. Reinforcement learning as classification: Leveraging modern classifiers. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 424–431, 2003b.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Analysis of a classification-based policy iteration algorithm. Technical Report inria-00482065, INRIA, 2010a.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Finite-sample analysis of LSTD. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 615–622, 2010b.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Analysis of a classification-based policy iteration algorithm. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 607–614, 2010c.

A. Lazaric, M. Ghavamzadeh, and R. Munos. Finite-sample analysis of least-squares policy iteration. *Journal of Machine Learning Research*, 13:3041–3074, 2012.

Boris Lesner and Bruno Scherrer. Tight performance bounds for approximate modified policy iteration with non-stationary policies. *CoRR*, abs/1304.5610, 2013.

R. Munos. Error bounds for approximate policy iteration. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 560–567, 2003.

R. Munos. Performance bounds in ℓ_p-norm for approximate value iteration. *SIAM J. Control and Optimization*, 46(2):541–561, 2007.

R. Munos and Cs. Szepesvári. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9:815–857, 2008.

D. Precup, R. Sutton, and S. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 759–766, 2000.

D. Precup, R. Sutton, and S. Dasgupta. Off-policy temporal difference learning with function approximation. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 417–424, 2001.

M. Puterman. *Markov decision processes.* Wiley, New York, 1994.

R. Rubinstein and D. Kroese. *The cross-entropy method: A unified approach to combinatorial optimization, Monte-Carlo simulation, and machine learning.* Springer-Verlag, 2004.

B. Scherrer. Performance bounds for λ-policy iteration and application to the game of tetris. *Journal of Machine Learning Research*, 14:1175–1221, 2013.

B. Scherrer and C. Thiéry. Performance bound for approximate optimistic policy iteration. Technical report, INRIA, 2010.

B. Scherrer, M. Ghavamzadeh, V. Gabillon, and M. Geist. Approximate modified policy iteration. In *Proceedings of the Twenty Ninth International Conference on Machine Learning*, pages 1207–1214, 2012.

S. Singh and R. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16-3:227–233, 1994.

Cs. Szepesvári. Reinforcement learning algorithms for mdps. In *Wiley Encyclopedia of Operations Research.* Wiley, 2010.

I. Szita and A. Lőrincz. Learning tetris using the noisy cross-entropy method. *Neural Computation*, 18(12):2936–2941, 2006.

C. Thiery and B. Scherrer. Building controllers for tetris. *International Computer Games Association Journal*, 32:3–11, 2009a. URL `http://hal.inria.fr/inria-00418954`.

C. Thiery and B. Scherrer. Improvements on learning tetris with cross entropy. *International Computer Games Association Journal*, 32, 2009b. URL `http://hal.inria.fr/inria-00418930`.

C. Thiery and B. Scherrer. Least-squares λ-policy iteration: bias-variance trade-off in control problems. In *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, pages 1071–1078, 2010a.

C. Thiery and B. Scherrer. MDPTetris features documentation, 2010b. `http://mdptetris.gforge.inria.fr/doc/feature_functions_8h.html`.

J. Tsitsiklis and B Van Roy. Feature-based methods for large scale dynamic programming. *Machine Learning*, 22:59–94, 1996.

J. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997.

Journal of Machine Learning Research 16 (2015) 1677-1681 Published 9/15

Preface to this Special Issue

Alex Gammerman ALEX@CS.RHUL.AC.UK
Vladimir Vovk V.VOVK@RHUL.AC.UK
Computer Learning Research Centre, Department of Computer Science
Royal Holloway, University of London

This issue of JMLR is devoted to the memory of Alexey Chervonenkis. Over the period of a dozen years between 1962 and 1973 he and Vladimir Vapnik created a new discipline of statistical learning theory—the foundation on which all our modern understanding of pattern recognition is based. Alexey was 28 years old when they made their most famous and original discovery, the uniform law of large numbers. In that short period Vapnik and Chervonenkis also introduced the main concepts of statistical learning theory, such as VC-dimension, capacity control, and the Structural Risk Minimization principle, and designed two powerful pattern recognition methods, Generalised Portrait and Optimal Separating Hyperplane, later transformed by Vladimir Vapnik into Support Vector Machine—arguably one of the best tools for pattern recognition and regression estimation. Thereafter Alexey continued to publish original and important contributions to learning theory. He was also active in research in several applied fields, including geology, bioinformatics, medicine, and advertising.

Alexey tragically died in September 2014 after getting lost during a hike in the Elk Island park on the outskirts of Moscow. Vladimir Vapnik suggested to prepare an issue of JMLR to be published at the first anniversary of the death of his long-term collaborator and close friend. Vladimir and the editors contacted a few dozen leading researchers in the fields of machine learning related to Alexey's research interests and had many enthusiastic replies. In the end eleven papers were accepted. This issue also contains a first attempt at a complete bibliography of Alexey Chervonenkis's publications.

Simultaneously with this special issue will appear Alexey's Festschrift (Vovk et al., 2015), to which the reader is referred for information about Alexey's research, life, and death. The Festschrift is based in part on a symposium held in Pathos, Cyprus, in 2013 to celebrate Alexey's 75th anniversary. Apart from research contributions, it contains Alexey's reminiscences about his early work on statistical learning with Vladimir Vapnik, a reprint of their seminal 1971 paper, a historical chapter by R. M. Dudley, reminiscences of Alexey's and Vladimir's close colleague Vasily Novoseltsev, and three reviews of various measures of complexity used in machine learning ("Measures of Complexity" is both the name of the symposium and the title of the book). Among Alexey's contributions to machine learning (mostly joint with Vladimir Vapnik) discussed in the book are:

- derivation of necessary and sufficient conditions for the uniform convergence of the frequencies of events to their probabilities (later developed into necessary and sufficient conditions for the uniform convergence of means to expectations);
- introduction of a new characteristic of classes of sets, which they called capacity (емкость) and which was later renamed as VC-dimension;

- development of powerful pattern recognition algorithms, Generalized Portrait and Optimal Separating Hyperplane;
- applying the theory of machine learning to diverse fields; e.g., a computer system using methods of machine learning was developed and installed at the world's largest open gold pit in Murun-Tau (Uzbekistan), which won him the State Prize of the USSR.

This Special Issue opens with the paper by Vladimir Vapnik and Rauf Izmailov "V-matrix method of solving statistical inference problems", which proposes new ways of solving learning problems such as estimating conditional probabilities. The authors ask whether the new methods can replace SVM in the problem of pattern recognition. Solving the problem of pattern recognition via estimating conditional probabilities might appear to contradict Vapnik's [1995, 1998] Imperative that the problem of interest (in this case pattern recognition) should be solved directly, without solving a more general problem (in this case estimating conditional probabilities) as an intermediate step. However, the authors explain that there is no real contradiction.

The paper "Batch learning from logged bandit feedback through Counterfactual Risk Minimization" by Adith Swaminathan and Thorsten Joachims extends Vapnik and Chervonenkis's Structural Risk Minimization principle to the situation where only partial feedback, determined by the prediction, is available. Such situations are ubiquitous in, e.g., advertisement placement, an active area of Alexey's research during the last years of his life.

The next paper, "Optimal estimation of low rank density matrices" by Vladimir Koltchinskii and Dong Xia, concerns Alexey's other major interest, quantum mechanics, which the authors mention at the beginning of their contribution. This interest is not reflected in Alexey's bibliography (published at the end of this Special Issue), and we know about it from reminiscences of his colleagues and relatives and from his technical report (Chervonenkis, 2001), in which he computes the covariance function for the solution to Schrödinger's equation. The paper by Koltchinskii and Xia is devoted to the estimation of density matrices, describing states of quantum systems, which has important applications in quantum tomography.

The paper "Fast rates in statistical and online learning" by van Erven, Peter D. Grünwald, Nishant A. Mehta, Mark D. Reid, and Robert C. Williamson explores conditions that make fast learning possible finding unexpected similarities between two styles of learning, statistical and online. It is interesting that Vapnik and Chervonenkis started their joint work in a non-statistical setting (Chervonenkis, 2015), although they quickly moved to their well-known statistical one, which is now standard in machine learning. The authors show us that the difference between the two styles of learning is smaller than it appears.

In their paper "On the asymptotic normality of an estimate of a regression functional" László Györfi and Harro Walk deal with the standard problem of regression in statistical learning theory but are interested in the quality (as measured by the square loss function) of the regression function rather than the regression function itself. They prove a surprisingly robust result about the asymptotic distribution of the main component of this measure of quality.

Pierre Bellec and Alexandre B. Tsybakov in "Sharp oracle bounds for monotone and convex regression through aggregation" consider important restricted versions of regression, in which the regression function is assumed to be either monotone (isotonic regression)

or convex. In evaluating their procedures the authors follow Vapnik and Chervonenkis's criterion of minimax loss, their main tools are different kinds of predictor aggregation, and their non-asymptotic performance guarantees are sharp.

The paper "Exceptional rotations of random graphs: a VC theory" by Louigi Addario-Berry, Shankar Bhamidi, Sébastien Bubeck, Luc Devroye, Gábor Lugosi, and Roberto Imbuzeiro Oliveira develops a fascinating analogue of the Vapnik–Chervonenkis statistical learning theory adapting it to random graphs.

The next paper, "Semi-supervised interpolation in an anticausal learning scenario" by Dominik Janzing and Bernhard Schölkopf, represents the area of causal inference and sheds new light on the situations in which seeing unlabelled observations does not provide any useful information (and so semi-supervised learning does not work). As the second author modestly says in a slightly different context elsewhere (Schölkopf, 2014), such results may not be as beautiful as those in the field that Alexey co-founded; however, this is compensated by their practical and philosophical importance.

It appears that unsupervised learning was one of the few fields of machine learning in which Alexey did not work directly, despite its importance in many applications: no one mind, even as versatile as his, can embrace everything. In their "Towards an axiomatic approach to hierarchical clustering of measures", Philipp Thomann, Ingo Steinwart, and Nico Schmid study the foundations of unsupervised learning. They show how the user's choice of a "clustering base" in conjunction with several natural axioms determines a clustering method.

Mark Herbster, Stephen Pasteris, and Massimiliano Pontil's "Predicting a switching sequence of graph labellings" is another paper devoted to online learning. The authors design new online prediction algorithms on graphs that can cope with switching labellings and multitask prediction problems.

The last research paper in this Special Issue is Vladimir Vapnik and Rauf Izmailov's "Learning using privileged information: Similarity control and knowledge transfer", which complements the standard protocol of statistical learning with an Intelligent Teacher providing the Student with privileged information. Such information is present in many real-world applications of machine learning and can be very useful.

The last part of the Special Issue is Alexey Chervonenkis's bibliography. His publications are listed in the chronological order, starting from the fundamental papers by Vapnik and Chervonenkis on the method of Generalized Portrait and the foundations of statistical learning theory, and then branching into countless fields including applied linguistics, geology, medicine and bioinformatics, and advertisement placement. They attest to his great role as discoverer and inventor. His tragic death a year ago was a great loss not only to his relatives, friends, and colleagues, who remember his wonderful warmth as a person, but also to the whole machine learning community and science in general.

References

Alexey Chervonenkis. Covariance function for Shrodinger equations decision. Technical Report CLRC-TR-01-03, Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, UK, April 2001. URL `www.clrc.rhul.ac.uk/publications/files/tr0103.ps`. Accessed in September 2015.

Alexey Chervonenkis. Chervonenkis's recollections. In Vovk et al. (2015), pages 3–8.

Bernhard Schölkopf, 2014. URL `http://people.tuebingen.mpg.de/bs/chervonenkis.html`. Post in memory of Alexey Chervonenkis. Accessed in September 2015.

Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995. Second edition: 2000.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

Vladimir Vovk, Harris Papadopoulos, and Alex Gammerman, editors. *Measures of Complexity: Festschrift for Alexey Chervonenkis*. Springer, Berlin, 2015.

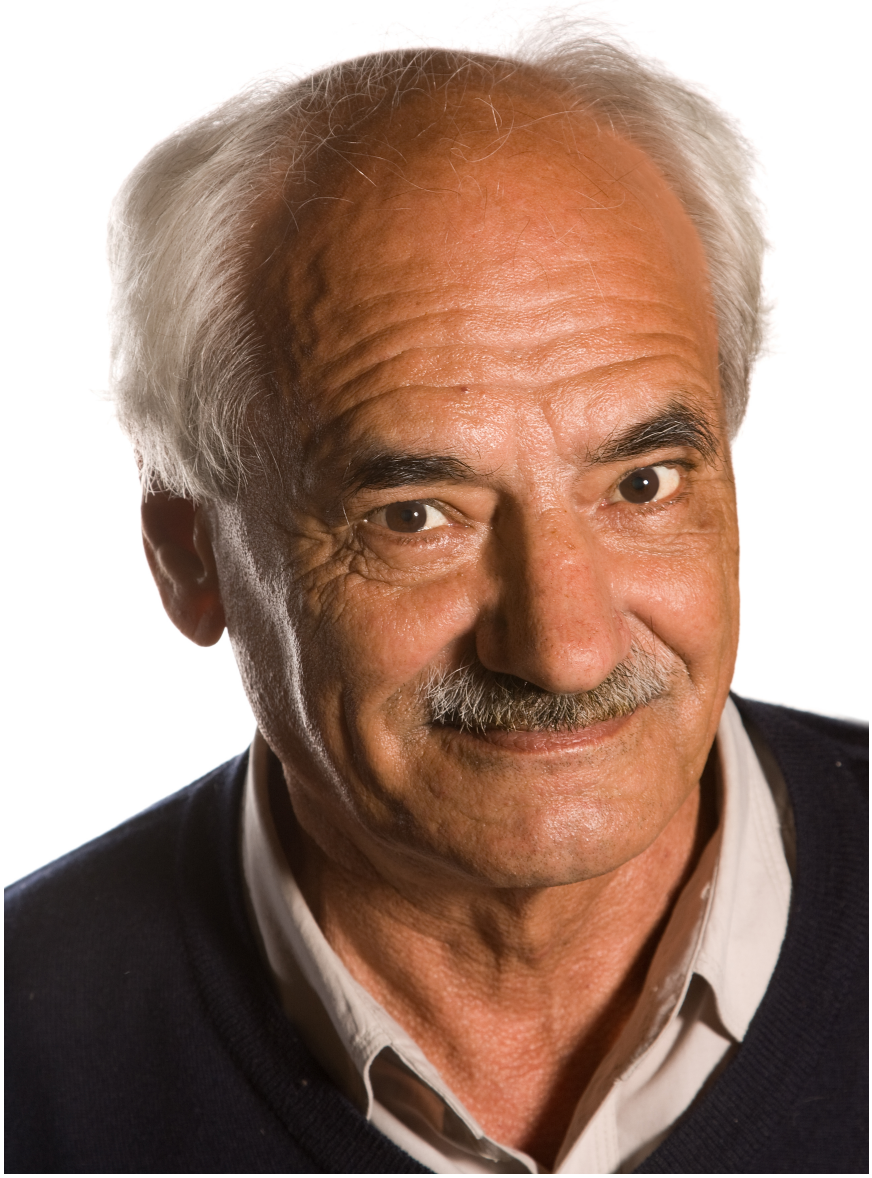


Figure 1: Alexey Chervonenkis (1938–2014)

Journal of Machine Learning Research 16 (2015) 1683-1730 Submitted 7/15; Published 9/15

In memory of Alexey Chervonenkis

V-Matrix Method of Solving Statistical Inference Problems

Vladimir Vapnik VLADIMIR.VAPNIK@GMAIL.COM
Columbia University
New York, NY 10027, USA
Facebook AI Research
New York, NY 10017, USA

Rauf Izmailov RIZMAILOV@APPCOMSCI.COM
Applied Communication Sciences
Basking Ridge, NJ 07920-2021, USA

Editors: Alex Gammerman and Vladimir Vovk

Abstract

This paper presents direct settings and rigorous solutions of the main Statistical Inference problems. It shows that rigorous solutions require solving multidimensional Fredholm integral equations of the first kind in the situation where not only the right-hand side of the equation is an approximation, but the operator in the equation is also defined approximately. Using Stefanuyk-Vapnik theory for solving such ill-posed operator equations, constructive methods of empirical inference are introduced. These methods are based on a new concept called V-matrix. This matrix captures geometric properties of the observation data that are ignored by classical statistical methods.

Keywords: conditional probability, regression, density ratio, ill-posed problem, mutual information, reproducing kernel Hilbert space · function estimation, interpolation function, support vector machines, data adaptation, data balancing, conditional density

1. Basic Concepts of Classical Statistics

In the next several sections, we describe main concepts of Statistics. We first outline these concepts for the one-dimensional case and then generalize them for the multidimensional case.

1.1 Cumulative Distribution Function

The basic concept of *Theoretical Statistics* and *Probability Theory* is the so-called *Cumulative Distribution Function* (CDF)

$$F(x) = P\{X \leq x\}.$$

This function defines the probability of the random variable X not exceeding x. Different CDFs describe different statistical environments, so CDF (defining the probability measure) is the main characteristic of the random events. In this paper, we consider the important case when $F(x)$ is a *continuous* function.

1.2 General Problems of Probability Theory and Statistics

The general problem of Probability Theory can be defined as follows:

> *Given a cumulative distribution function $F(x)$, describe outcomes of random experiments for a given theoretical model.*

The general problem of Statistics can be defined as follows:

> *Given iid observations of outcomes of the same random experiments, estimate the statistical model that defines these observations.*

In Section 2, we discuss several main problems of Statistics. Next, we consider the basic one: estimation of CDF.

1.3 Empirical Cumulative Distribution Functions

In order to estimate CDF, one introduces the so-called *Empirical Cumulative Distribution function* (ECDF) constructed for iid observations obtained according to $F(x)$:

$$X_1, ..., X_\ell.$$

The ECDF function has the form

$$F_\ell(x) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(x - X_i),$$

where $\theta(x - X_i)$ is the step-function

$$\theta(x - X_i) = \begin{cases} 1, & \text{if } x \geq X_i, \\ 0, & \text{if } x < X_i. \end{cases}$$

Classical statistical theory is based on convergence of ECDF converges to CDF when the number ℓ of observations increases.

1.4 The Glivenko-Cantelli Theorem and Kolmogorov Type Bounds

In 1933, the following theorem was proven (Glivenko-Cantelli theorem).

Theorem. *Empirical cumulative distribution functions converge uniformly to the true cumulative distribution function:*

$$\lim_{\ell \to \infty} P\{\sup_x |F(x) - F_\ell(x)| \geq \varepsilon\} = 0, \quad \forall \varepsilon > 0.$$

In 1933, Kolmogorov derived asymptotical exact rate of convergence of ECDF to CDF for continuous functions $F(x)$:

$$\lim_{\ell \to \infty} P\{\sqrt{\ell} \sup_x |F(x) - F_\ell(x)| \geq \varepsilon\} = 2 \sum_{k=1}^{\infty} (-1)^{k-1} \exp\{-2\varepsilon^2 k^2\}. \tag{1}$$

Later, Dvoretzky, Kiefer, Wolfowitz, and Massart showed the existence of exponential type of bounds for any ℓ:

$$P\{\sup_x |F(x) - F_\ell(x)| \geq \varepsilon\} \leq 2\exp\{-2\varepsilon^2\ell\}. \tag{2}$$

Bound (2) is defined by the first term of the right-hand side of Kolmogorov asymptotic equality (1).

Glivenko-Cantelli theorem and bounds (1), (2) can be considered as a foundation of statistical science since they claim that:

1. It is possible to estimate the true statistical distribution from iid data.

2. The ECDF strongly converges to the true CDF, and this convergence is fast.

1.5 Generalization to Multidimensional Case

Let us generalize the main concepts described above to the multidimensional case. We start with CDF.

Joint cumulative distribution function. For the multivariate random variable $x = (x^1, ..., x^d)$, the joint cumulative distribution function $F(x)$, $x \in R^d$ is defined by the function

$$F(x) = P\{X^1 \leq x^1, ..., X^d \leq x^d\}. \tag{3}$$

As in the one-dimensional case, the main problem of Statistics is as follows: estimate CDF, as defined in (3), based on random multivariate iid observations

$$X_1, ..., X_\ell, \quad X_i \in R^d, \quad i = 1, \ldots, \ell..$$

In order to solve this problem, one uses the same idea of empirical distribution function

$$F_\ell(x) = \frac{1}{\ell}\sum_{i=1}^{\ell} \theta(x - X_i),$$

where $x = (x^1, ..., x^d) \in R^d$, $X_i = (X_i^1, ..., X_i^d) \in R^d$ and

$$\theta(x - X_i) = \prod_{k=1}^{d} \theta(x^k - X_i^k).$$

Note that

$$F(x) = E_u\theta(x - u) = \int \theta(x - u)dF(u),$$

and the generalized (for the multidimensional case) Glivenko-Cantelli theorem has the form

$$\lim_{\ell\to\infty} P\left\{\sup_x \left| E_u\theta(x - u) - \frac{1}{\ell}\sum_{i=1}^{\ell} \theta(x - X_i)\right| \geq \varepsilon\right\} = 0.$$

This equation describes the uniform convergence of the empirical risks to their expectation over vectors $u \in R^d$ for the parametric set of multidimensional step functions $\theta(x-u)$ (here

$x, u \in R^d$, and x is a vector of parameters). Since VC dimension of this set of functions is equal[1] to one, according to the VC theory (Vapnik and Chervonenkis, 1974), (Vapnik, 1995), (Vapnik, 1998), the corresponding rate of convergence is bounded as follows:

$$P\left\{\sup_x \left|E_u\theta(x-u) - \frac{1}{\ell}\sum_{i=1}^{\ell}\theta(x-X_i)\right| \geq \varepsilon\right\} \leq \exp\left\{-\left(\varepsilon^2 - \frac{\ln \ell}{\ell}\right)\ell\right\}. \tag{4}$$

According to this bound, for sufficiently large values of ℓ, the convergence of ECDF to the actual CDF does not depend on the dimensionality of the space. This fact has important consequences for Applied Statistics.

2. Main Problems of Statistical Inference

The main target of statistical inference theory is estimation (from the data) of specific models of random events, namely:

1. conditional probability function;
2. conditional density function;
3. regression function;
4. density ratio function.

2.1 Conditional Density, Conditional Probability, Regression, and Density Ratio Functions

Let $F(x)$ be a cumulative distribution function of random variable x. We call non-negative function $p(x)$ the probability density function if

$$\int_{-\infty}^{x} p(x^*)dx^* = F(x).$$

Similarly, let $F(x, y)$ be the joint probability distribution function of variables x and y. We call non-negative $p(x, y)$ the joint probability density function of two variables x and y if

$$\int_{-\infty}^{y}\int_{-\infty}^{x} p(x^*, y^*)dx^*dy^* = F(x, y).$$

1. Let $p(x, y)$ and $p(x)$ be probability density functions for pairs (x, y) and vectors x. Suppose that $p(x) > 0$. The function

$$p(y|x) = \frac{p(x, y)}{p(x)}$$

is called the *Conditional Density Function*. It defines, for any fixed $x = x_0$, the probability density function $p(y|x = x_0)$ of random value $y \in R^1$. The estimation of the conditional density function from data

$$(y_1, X_1), ..., (y_\ell, X_\ell) \tag{5}$$

1. Since the set of d-dimensional parametric (with respect to parameter x) functions $\theta(x - u)$ can shatter, at most, one vector.

is the most difficult problem in our list of statistical inference problems.

2. Along with estimation of the conditional density function, the important problem is to estimate the so-called *Conditional Probability Function*. Let variable y be discrete, say, $y \in \{0, 1\}$. The function defined by the ratio

$$p(y = 1|x) = \frac{p(x, y = 1)}{p(x)}, \quad p(x) > 0$$

is called *Conditional Probability Function*. For any given vector $x = x_0$, this function defines the probability that y is equal to one; correspondingly, $p(y = 0|x = x_0) = 1 - p(y = 1|x = x_0)$. The problem is to estimate the conditional probability function, given data (5) where $y \in \{0, 1\}$.

3. As mentioned above, estimation of the conditional density function is a difficult problem; a much easier problem is the problem of estimating the so-called *Regression Function* (conditional expectation of the variable y):

$$r(x) = \int y p(y|x) dy,$$

which defines expected value $y \in R^1$ for a given vector x.

4. In this paper, we also consider a problem, which is important for applications: estimating the ratio of two probability densities (Sugiyama et al., 2012). Let $p_{\text{num}}(x)$ and $p_{\text{den}}(x) > 0$ be two different density functions (subscripts *num* and *den* correspond to numerator and denominator of the density ratio). Our goal is to estimate the function

$$R(x) = \frac{p_{\text{num}}(x)}{p_{\text{den}}(x)}$$

given iid data

$$X_1, ..., X_{\ell_{\text{den}}},$$

distributed according to $p_{\text{den}}(x)$, and iid data

$$X'_1, ..., X'_{\ell_{\text{num}}},$$

distributed according to $p_{\text{num}}(x)$.

In the next sections, we introduce direct settings for these four statistical inference problems.

2.2 Direct Constructive Setting for Conditional Density Estimation

By definition, conditional density $p(y|x)$ is the ratio of two densities

$$p(y|x) = \frac{p(x, y)}{p(x)}, \quad p(x) > 0 \tag{6}$$

or, equivalently,

$$p(y|x)p(x) = p(x, y).$$

This expression leads to the following equivalent one:

$$\int \int \theta(y - y')\theta(x - x')f(x', y')dF(x')dy' = F(x, y), \tag{7}$$

where $f(x, y) = p(y|x)$, function $F(x)$ is the cumulative distribution function of x and $F(x, y)$ is the joint cumulative distribution function of x and y.

Therefore, our setting of the condition density estimation problem is as follows:

Find the solution of integral equation (7) in the set of nonnegative functions $f(x, y) = p(y|x)$ when the cumulative probability distribution functions $F(x, y)$ and $F(x)$ are unknown but iid data

$$(y_1, X_1), ..., (y_\ell, X_\ell)$$

are given.

In order to solve this problem, we use empirical estimates

$$F_\ell(x, y) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(y - y_i)\theta(x - X_i), \tag{8}$$

$$F_\ell(x) = \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(x - X_i) \tag{9}$$

of the unknown cumulative distribution functions $F(x, y)$ and $F(x)$. Therefore, we have to solve an integral equation where not only its right-hand side is defined approximately ($F_\ell(x, y)$ instead of $F(x, y)$), but also the data-based approximation

$$A_\ell f(x, y) = \int \int \theta(y - y')\theta(x - x')f(x', y')dy'dF_\ell(x')$$

is used instead of the exact integral operator

$$Af(x, y) = \int \int \theta(y - y')\theta(x - x')f(x', y')dy'dF(u').$$

Taking into account (9), our goal is thus to find the solution of approximately defined equation

$$\sum_{i=1}^{\ell} \theta(x - X_i) \int_{-\infty}^{y} f(X_i, y')dy' \approx \frac{1}{\ell} \sum_{i=1}^{\ell} \theta(y - y_i)\theta(x - X_i). \tag{10}$$

Taking into account definition (6), we have

$$\int_{-\infty}^{\infty} p(y|x)dy = 1, \quad \forall x \in \mathcal{X}.$$

Therefore, the solution of equation (10) has to satisfy the constraint $f(x, y) \geq 0$ and the constraint

$$\int_{-\infty}^{\infty} f(y', x)dy' = 1, \quad \forall x \in \mathcal{X}.$$

We call this setting *the direct constructive setting* since it is based on direct definition of conditional density function (7) and uses theoretically justified approximations (8), (9) of unknown functions.

2.3 Direct Constructive Setting for Conditional Probability Estimation

The problem of estimation of the conditional probability function can be considered analogously to the conditional density estimation problem. The conditional probability is defined as

$$p(y=1|x) = \frac{p(x, y=1)}{p(x)}, \quad p(x) > 0 \tag{11}$$

or, equivalently,

$$p(y=1|x)p(x) = p(x, y=1).$$

We can rewrite it as

$$\int \theta(x - x')f(x')dF(x') = F(x, y=1), \tag{12}$$

where $f(x) = p(y=1|x)$ and $F(x, y=1) = P\{X \le x, y=1\}$.

Therefore, the problem of estimating the conditional probability is formulated as follows.

In the set of bounded functions $0 \le f(x) \le 1$, find the solution of equation (12) if cumulative distribution functions $F(x)$ and $F(x, y=1)$ are unknown but iid data

$$(y_1, X_1), ..., (y_\ell, X_\ell), \qquad y \in \{0, 1\}, \quad x \in \mathcal{X},$$

generated according to $F(x, y)$, are given.

As before, instead of unknown cumulative distribution functions we use their empirical approximations

$$F_\ell(x) = \frac{1}{\ell}\sum_{i=1}^{\ell} \theta(x - X_i), \tag{13}$$

$$F_\ell(x, y=1) = p_\ell F_\ell(x|y=1) = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i \theta(x - X_i), \tag{14}$$

where p_ℓ is the ratio of the number of examples with $y = 1$ to the total number ℓ of the observations.

Therefore, one has to solve integral equation (12) with approximately defined right-hand side (13) and approximately defined operator (14):

$$A_\ell f(x) = \frac{1}{\ell}\sum_{i=1}^{\ell} \theta(x - X_i)f(X_i).$$

Since the probability takes values between 0 and 1, our solution has to satisfy the bounds

$$0 \le f(x) \le 1, \quad \forall x \in \mathcal{X}.$$

Also, definition (11) implies that

$$\int f(x)dF(x) = p(y=1),$$

where $p(y=1)$ is the probability of $y = 1$.

2.4 Direct Constructive Setting for Regression Estimation

By definition, regression is the conditional mathematical expectation

$$r(x) = \int yp(y|x)dy = \int y\frac{p(x,y)}{p(x)}dy.$$

This can be rewritten in the form

$$r(x)p(x) = \int yp(x,y)dy. \tag{15}$$

From (15), one obtains the equivalent equation

$$\int \theta(x-x')r(x')dF(x') = \int \theta(x-x')\int ydF(x',y'). \tag{16}$$

Therefore, the direct constructive setting of regression estimation problem is as follows:

In a given set of functions $r(x)$, find the solution of integral equation (16) if cumulative probability distribution functions $F(x,y)$ and $F(x)$ are unknown but iid data (5) are given.

As before, instead of these functions, we use their empirical estimates. That is, we construct the approximation

$$A_\ell r(x) = \frac{1}{\ell}\sum_{i=1}^{\ell}\theta(x-X_i)r(X_i)$$

instead of the actual operator in (16), and the approximation of the right-hand side

$$F_\ell(x) = \frac{1}{\ell}\sum_{j=1}^{\ell}y_j\theta(x-X_j)$$

instead of the actual right-hand side in (16), based on the observation data

$$(y_1,X_1),...,(y_\ell,X_\ell), \quad y\in R^1, \quad x\in\mathcal{X}. \tag{17}$$

2.5 Direct Constructive Setting of Density Ratio Estimation Problem

Let $F_{\text{num}}(x)$ and $F_{\text{den}}(x)$ be two different cumulative distribution functions defined on $\mathcal{X}\subset R^d$ and let $p_{\text{num}}(x)$ and $p_{\text{den}}(x)$ be the corresponding density functions. Suppose that $p_{\text{den}}(x)>0, x\in\mathcal{X}$. Consider the ratio of two densities:

$$R(x) = \frac{p_{\text{num}}(x)}{p_{\text{den}}(x)}.$$

The problem is to estimate the ratio $R(x)$ when densities are unknown, but iid data

$$X_1,...,X_{\ell_{\text{den}}} \sim F_{\text{den}}(x), \tag{18}$$

generated according to $F_{\text{den}}(x)$, and iid data

$$X'_1,...,X'_{\ell_{\text{num}}} \sim F_{\text{num}}(x), \tag{19}$$

generated according to $F_{\text{num}}(x)$, are given.

As before, we introduce the constructive setting of this problem: solve the integral equation

$$\int \theta(x-u)R(u)dF_{\text{den}}(u) = F_{\text{num}}(x)$$

when cumulative distribution functions $F_{\text{den}}(x)$ and $F_{\text{num}}(x)$ are unknown, but data (18) and (19) are given. As before, we approximate the unknown cumulative distribution functions $F_{\text{num}}(x)$ and $F_{\text{den}}(x)$ using empirical distribution functions

$$F_{\ell_{\text{num}}}(x) = \frac{1}{\ell_{\text{num}}} \sum_{j=1}^{\ell_{\text{num}}} \theta(x - X'_j)$$

for $F_{\text{num}}(x)$, and

$$F_{\ell_{\text{den}}}(x) = \frac{1}{\ell_{\text{den}}} \sum_{j=1}^{\ell_{\text{den}}} \theta(x - X_j)$$

for $F_{\text{den}}(x)$.

Since $R(x) \geq 0$ and $\lim_{x\to\infty} F_{\text{num}}(x) = 1$, our solution has to satisfy the constraints

$$R(x) \geq 0, \quad \forall x \in \mathcal{X},$$

$$\int R(x)dF_{\text{den}}(x) = 1.$$

Therefore, all main empirical inference problems can be represented via (multidimensional) Fredholm integral equation of the first kind with approximately defined elements. Although approximations converge to the true functions, these problems are computationally difficult due to their ill-posed nature. Thus they require rigorous solutions.[2]

In Section 5, we consider methods for solving ill-posed operator equations, which we apply in Section 6 to our problems of inference. Before that, however, we present a general form for all statistical inference problems in the next subsections.

2.6 General Form of Statistical Inference Problems

Consider the multidimensional Fredholm integral equation

$$\int \theta(z-z')f(z')dF_A(z') = F_B(z),$$

where the kernel of operator equation is defined by the step function $\theta(z-z')$, the cumulative distribution functions $F_A(z)$ and $F_B(z)$ are unknown but the corresponding iid data

$$Z_1, ..., Z_{\ell_A} \sim F_A(z)$$

$$Z_1, ..., Z_{\ell_B} \sim F_B(z)$$

are given. In the different inference problems, the elements $f(z), F_A(z), F_B(z)$ of the equation have different meanings (Table 1):

2. Various statistical methods exist for solving these inference problems. Our goal is to find general rigorous solutions that take into account all the available characteristics of the problems.

	Conditional density	**Conditional probability**	**Density ratio**	**Regression**
z	(x, y)	x	x	(x, y), where $y \geq 0$
$f(z)$	$p(y\|x)$	$p(y = 1\|x)$	$\frac{p_{num}(x)}{p_{den}(x)}$	$\hat{y}^{-1}R(x),\ (R(x) = \int yp(y\|x)dy)$
$F_A(z)$	$F(x)$	$F(x)$	$F_{num}(x)$	$F(x)$
$F_B(z)$	$F(x, y)$	$F(x\|y = 1)p(y = 1)$	$F_{den}(x)$	$\hat{y}^{-1} \int \theta(x - x')y' dF(x', y')$

Table 1: Vector z, solution $f(z)$, and functions $F_A(z)$, $F_B(z)$ for different statistical inference problems.

1. In the problem of conditional density estimation, vector z is the pair (x, y), the solution $f(z)$ is $p(y|x)$, the cumulative distribution function $F_A(z)$ is $F(x)$ and the cumulative distribution function $F_B(z)$ is $F(x, y)$.

2. In the problem of conditional probability $p(y = 1|x)$ estimation, vector z is x, the solution $f(z)$ is $p(y = 1|x)$, the cumulative distribution function $F_A(z)$ is $F(x)$, the cumulative distribution function $F_B(z)$ is $F(x|y = 1)p(y = 1)$, where $p(y = 1)$ is the probability of class $y = 1$.

3. In the problem of density ratio estimation, the vector z is x, the solution $f(z)$ is $p_{num}(x)/p_{den}(x)$, the cumulative function $F_A(z)$ is $F_{num}(x)$, the cumulative function $F_B(z)$ is $F_{den}(x)$.

4. In the problem of regression $R(x) = \int yp(y|x)dy$ estimation, the vector z is (x, y), where $y \geq 0$, the solution $f(z)$ is $\hat{y}^{-1}R(x),\ (R(x) = \int yp(y|x)dy)$, the cumulative function $F_A(z)$ is $F(x)$, the cumulative function $F_B(z)$ is $\hat{y}^{-1} \int \theta(x' - x')y' dF(x', y')$.

Since statistical inference problems have the same kernel of the integral equations (i.e., the step-function) and the same right-hand side (i.e., the cumulative distribution function), it allows us to introduce (in Section 5) a common standard method (called V-matrix method) for solving all inference problems.

3. Solution of Ill-Posed Operator Equations

In this section, we consider ill-posed operator equations and their solutions.

3.1 Fredholm Integral Equations of the First Kind

In this section, we consider the linear operator equations

$$Af = F, \tag{20}$$

where A maps elements of the metric space $f \in \mathcal{M} \subset E_1$ into elements of the metric space $F \in \mathcal{N} \subset E_2$. Let f be a continuous one-to-one operator and $f(\mathcal{M}) = \mathcal{N}$. Let the solution of such operator equation exist and be unique. Then

$$\mathcal{M} = A^{-1}\mathcal{N}.$$

The crucial question is whether this inverse operator A^{-1} is continuous. If it is, then close functions in $\mathcal{N}$ correspond to close functions in $\mathcal{M}$. That is, "small" changes in the right-hand side of (20) cause "small" changes of its solution. In this case, we call the operator A^{-1} *stable* (Tikhonov and Arsenin, 1977).

If, however, the inverse operator is discontinuous, then "small" changes in the right-hand side of (20) can cause significant changes of the solution. In this case, we call the operator A^{-1} *unstable*.

Solution of equation (20) is called *well-posed* if this solution

1. *exists;*
2. *is unique;*
3. *is stable.*

Otherwise we call the solution *ill-posed*.

We are interested in the situation when the solution of operator equation *exists*, and *is unique*. In this case, the effectiveness of solution of equation (20) is defined by the *stability* of the operator A^{-1}. If the operator is unstable, then, generally speaking, the numerical solution of equation is impossible.

Here we consider linear integral operator

$$Af(x) = \int_a^b K(x,u)f(u)du$$

defined by the kernel $K(t,u)$, which is continuous almost everywhere on $a \le t \le b, \quad c \le x \le d$. This kernel maps the set of functions $\{f(t)\}$, continuous on $[a,b]$, unto the set of functions $\{F(x)\}$, also continuous on $[c,d]$. The corresponding Fredholm equation of the first kind

$$\int_a^b K(x,u)f(u)du = F(x)$$

requires finding the solution $f(u)$ given the right-hand side $F(x)$.

In this paper, we consider integral equation defined by the so-called convolution kernel

$$K(x,u) = K(x-u).$$

Moreover, we consider the specific convolution kernel of the form

$$K(x-u) = \theta(x-u).$$

As stated in Section 2.2, this kernel covers all settings of empirical inference problems.

First, we show that the solution of equation

$$\int_0^1 \theta(x-u)f(u)du = x \tag{21}$$

is indeed ill-posed[3]. It is easy to check that

$$f(x) = 1$$

is the solution of this equation. Indeed,

$$\int_0^1 \theta(x-u)du = \int_0^x du = x. \tag{22}$$

It is also easy to check that the function

$$f^*(x) = 1 + \cos nx \tag{23}$$

is a solution of the equation

$$\int_0^1 \theta(x-u)f^*(u)du = x + \frac{\sin nx}{n}. \tag{24}$$

That is, when n increases, the right-hand sides of equations (22) and (24) are getting close to each other, but their solutions (21) and (23) are not.

The problem is how one can solve an ill-posed equation when its right-hand side is defined imprecisely.

3.2 Methods of Solving Ill-Posed Problems

In this subsection, we consider methods for solving ill-posed operator equations.

3.2.1 Inverse Operator Lemma

The following classical inverse operator lemma (Tikhonov and Arsenin, 1977) is the key enabler for solving ill-posed problems.

Lemma. *If A is a continuous one-to-one operator defined on a compact set $\mathcal{M}^* \subset \mathcal{M}$, then the inverse operator A^{-1} is continuous on the set $\mathcal{N}^* = A\mathcal{M}^*$.*

Therefore, the conditions of existence and uniqueness of the solution of an operator equation imply that the problem is well-posed on the compact $\mathcal{M}^*$. The third condition (stability of the solution) is automatically satisfied. This lemma is the basis for all constructive ideas of solving ill-posed problems. We now consider one of them.

3.2.2 Regularization Method

Suppose that we have to solve the operator equation

$$Af = F \tag{25}$$

3. Using the same arguments, one can show that the problem of solving any Fredholm equation of the first kind is ill-posed.

defined by continuous one-to-one operator A mapping $\mathcal{M}$ into $\mathcal{N}$, and assume the solution of (25) exists. Also suppose that, instead of the right-hand side $F(x)$, we are given its approximation $F_\delta(x)$, where

$$\rho_{E_2}(F(x), F_\delta(x)) \le \delta.$$

Our goal is to find the solution of equation

$$Af = F_\delta$$

when $\delta \to 0$.

Consider a lower semi-continuous functional $W(f)$ (called the *regularizer*) that has the following three properties:

1. the solution of the operator equation (25) belongs to the domain $D(W)$ of the functional $W(f)$;
2. functional $W(f)$ is non-negative values in its domain;
3. all sets
$$\mathcal{M}_c = \{f : W(f) \le c\}$$
are compact for any $c \ge 0$.

The idea of regularization is to find a solution for (25) as an element minimizing the so-called regularized functional

$$R_\gamma(\hat{f}, F_\delta) = \rho^2_{E_2}(A\hat{f}, F_\delta) + \gamma_\delta W(\hat{f}), \qquad \hat{f} \in D(W) \tag{26}$$

with *regularization parameter* $\gamma_\delta > 0$.

The following theorem holds true (Tikhonov and Arsenin, 1977).

Theorem 1 *Let E_1 and E_2 be metric spaces, and suppose for $F \in \mathcal{N}$ there exists a solution of (25) that belongs to $\mathcal{M}_c$. Suppose that, instead of the exact right-hand side F in (25), its approximations[4] $F_\delta \in E_2$ in (26) are given such that $\rho_{E_2}(F, F_\delta) \le \delta$. Consider the sequence of parameters γ such that*

$$\gamma(\delta) \longrightarrow 0 \quad \text{for} \quad \delta \longrightarrow 0,$$
$$\lim_{\delta \longrightarrow 0} \frac{\delta^2}{\gamma(\delta)} \le r < \infty. \tag{27}$$

Then the sequence of solutions $f_\delta^{\gamma(\delta)}$ minimizing the functionals $R_{\gamma(\delta)}(f, F_\delta)$ on $D(W)$ converges to the exact solution f (in the metric of space E_1) as $\delta \longrightarrow 0$.

In a Hilbert space, the functional $W(f)$ may be chosen as $||f||^2$ for a linear operator A. Although the sets $\mathcal{M}_c$ are (only) weakly compact in this case, regularized solutions converge to the desired one. Such a choice of regularized functional is convenient since its domain $D(W)$ is the whole space E_1. In this case, however, the conditions on the parameters γ are more restrictive than in the case of Theorem 1: namely, γ should converge to zero slower than δ^2.

Thus the following theorem holds true (Tikhonov and Arsenin, 1977).

Theorem 2 *Let E_1 be a Hilbert space and $W(f) = ||f||^2$. Then for $\gamma(\delta)$ satisfying (27) with $r = 0$, the regularized element $f_\delta^{\gamma(\delta)}$ converges to the exact solution f in metric E_1 as $\delta \to 0$.*

4. The elements F_δ do not have to belong to the set $\mathcal{N}$.

4. Stochastic Ill-Posed Problems

In this section, we consider the problem of solving the operator equation

$$Af = F, \tag{28}$$

where not only its right-hand side is defined approximately ($F_\ell(x)$ instead of $F(x)$), but the operator Af is also defined approximately. Such problem are called *stochastic ill-posed problems.*

In the next subsections, we describe the conditions under which it is possible to solve equation (28), where both the right-hand side and the operator are defined approximately. We first discuss the general theory for solving stochastic ill-posed problems and then consider specific operators describing particular problems, i.e., empirical inference problems described in Sections 2.3, 2.4, and 2.5. For all these problems, the operator has the form

$$A_\ell f = \int \theta(x-u) f(u) dF_\ell(u).$$

We show that rigorous solutions of stochastic ill-posed problem with this operator leverage the so-called V-matrix, which captures some geometric properties of the data; we also describe specific algorithms for solution of our empirical inference problems.

4.1 Regularization of Stochastic Ill-Posed Problems

Consider the problem of solving the operator equation

$$Af = F$$

under the condition where (random) approximations are given not only for the function on the right-hand side of the equation but for the operator as well (*the stochastic ill-posed problem).*

We assume that, instead of the true operator A, we are given a sequence of random continuous operators $A_\ell,\ \ell = 1, 2, ...$ that converges in probability to the operator A (the definition of closeness between two operators will be defined later).

First, we discuss general conditions under which the solution of stochastic ill-posed problem is possible; after that, we consider specific operator equations corresponding to each empirical inference problem.

As before, we consider the problem of solving the operator equation by the regularization method, i.e., by minimizing the functional

$$R^*_{\gamma_\ell}(f, F_\ell, A_\ell) = \rho^2_{E_2}(A_\ell f, F_\ell) + \gamma_\ell W(f). \tag{29}$$

For this functional, there exists a minimum (perhaps, not unique). We define the closeness of operator A and operator A_ℓ as the distance

$$||A_\ell - A|| = \sup_{f \in D} \frac{||A_\ell f - Af||_{E_2}}{W^{1/2}(f)}.$$

The main result for solving stochastic ill-posed problems via regularization method (29) is provided by the following Theorem (Stefanyuk, 1986), (Vapnik, 1998).

Theorem. *For any $\varepsilon > 0$ and any constants $C_1, C_2 > 0$ there exists a value $\gamma_0 > 0$ such that for any $\gamma_\ell \le \gamma_0$ the inequality*

$$P\{\rho_{E_1}(f_\ell, f) > \varepsilon\} \le P\{\rho_{E_2}(F_\ell, F) > C_1\sqrt{\gamma_\ell}\} + P\{||A_\ell - A|| > C_2\sqrt{\gamma_\ell}\} \tag{30}$$

holds true.

Corollary. As follows from this theorem, if the approximations $F_\ell(x)$ of the right-hand side of the operator equation converge to the true function $F(x)$ in E_2 with the rate of convergence $r(\ell)$, and the approximations A_ℓ converge to the true operator A in the metric in E_1 defined in (30) with the rate of convergence $r_A(\ell)$, then there exists such a function

$$r_0(\ell) = \max\{r(\ell),\ r_A(\ell)\}; \qquad \lim_{\ell\to\infty} r_0(\ell) = 0,$$

that the sequence of solutions to the equation converges in probability to the true one if

$$\lim_{\ell\to\infty} \frac{r_0(\ell)}{\sqrt{\gamma_\ell}} = 0, \quad \lim_{\ell\to\infty} \gamma_\ell = 0.$$

4.2 Solution of Empirical Inference Problems

In this section, we consider solutions of the integral equation

$$Af = F,$$

where operator A has the form

$$Af = \int \theta(x-u) f(u) dF_1(u),$$

and the right-hand side of the equation is $F_2(x)$. That is, our goal is to solve the integral equation

$$\int \theta(x-u) f(u) dF_1(x) = F_2(x).$$

We consider the case where $F_1(x)$ and $F_2(x)$ are two different cumulative distribution functions. (This integral equation also includes, as a special case, the problem of regression estimation, where $F_2(x) = \int y dP(x,y)$ for non-negative y). This equation defines the main empirical inference problem described in Section 2. The problem of density ratio estimation requires solving this equation when both functions $F_1(x)$ and $F_2(x)$ are unknown but the iid data

$$X_1^1, ..., X_{\ell_1}^1 \sim F_1 \tag{31}$$

$$X_1^1, ..., X_{\ell_2}^1 \sim F_2 \tag{32}$$

are available. In order to solve this equation, we use empirical approximations instead of actual distribution functions, thus obtaining

$$A_{\ell_1} f = \int \theta(x-u) dF_{\ell_1}(u) \tag{33}$$

$$F_{\ell_k}(x) = \frac{1}{\ell_k}\sum_{i=1}^{\ell_k}\theta(x - X_i^k), \quad k = 1, 2,$$

where $F_{\ell_1}(u)$ are and $F_{\ell_2}(x)$ are the empirical distribution functions obtained from data (31) and (32), respectively.

One can show (see (Vapnik, 1998), Section 7.7) that, for sufficiently large ℓ, the inequality

$$||A_\ell - A|| = \sup_f \frac{||A_\ell f - Af||_{E_2}}{W^{1/2}(f)} \leq ||F_\ell - F||_{E_2}$$

holds true for the smooth solution $f(x)$ of our equations.

From this inequality, bounds (4), and the Theorem of Section 4.1, it follows that the regularized solutions of our operator equations converge to the actual function

$$\rho_{E_1}(f_\ell, f) \to_{\ell\to\infty} 0$$

with probability one.

Therefore, to solve our inference problems, we minimize the functional

$$R_\gamma(f, F_\ell, A_{\ell_1}) = \rho^2_{E_2}(A_{\ell_1} f, F_{\ell_2}) + \gamma_\ell W(f). \tag{34}$$

In order to do this well and find the unique solution of this problem, we have to define three elements of (34):

1. The distance $\rho_{E_2}(F_1, F_2)$ between functions $F_1(x)$ and $F_2(x)$ in E_2.
2. The regularization functional $W(f)$ in the space of functions $f \in E_1$.
3. The rule for selecting the regularization constant γ_ℓ.

In the next sections, we consider the first two elements.

5. Solving Statistical Inference Problems with V-matrix

Consider the explicit form of the functional for solving our inference problems. In order to do this, we specify expressions for the squared distance and regularization functional in expression (34).

5.1 The V-matrix

In this subsection, we consider the key element of our approach, the V-matrix.

5.1.1 Definition of Distance

Let our distance in E_2 be defined by the L_2 metric

$$\rho^2_{E_2}(F_1(x), F_2(x)) = \int (F_1(x) - F_2(x))^2 \sigma(x) d\mu(x),$$

where $\sigma(x)$ is a *known* positive function and $\mu(x)$ is a *known* measure defined on $\mathcal{X}$. To define distance, one can use any non-negative measurable function $\sigma(x)$ and any measure

$\mu(x)$. For example, if our equation is defined in the box domain $[0,1]^d$, we can use uniform measure in this domain and $\sigma(x) = 1$.

Below we define the measure $\mu(x)$ as

$$d\mu(x) = \prod_{k=1}^{d} dF_\ell(x^k), \tag{35}$$

where each $F_\ell(x^k)$ is the marginal empirical cumulative distribution function of the coordinate x^k estimated from data.

We also choose function $\sigma(x)$ in the form

$$\sigma(x) = \prod_{k=1}^{n} \sigma_k(x^k). \tag{36}$$

In this paper, we consider several weight functions $\sigma(x^k)$:

1. The function
$$\sigma(x^k) = 1.$$

2. For the problem of conditional probability estimation, we consider the function
$$\sigma(x^k) = \frac{1}{F_\ell(x^k|y=1)(1 - F_\ell(x^k|y=1)) + \epsilon}, \tag{37}$$
where $\varepsilon > 0$ is a small constant.

3. For the problem of regression estimation, we consider the case where $y \geq 0$ and, instead of $F_\ell(x^k|y=1)$ in (37), the monotonic function
$$F_\ell(x^k) = \frac{1}{\ell \hat{y}_{av}} \sum_{i=1}^{\ell} y_i \theta(x^k - X_i^k)$$
is used, where $\hat{y}_{av}$ is the average value of y in the training data. This function has properties of ECDF.

4. For the problem of density ratio estimation, we consider an estimate of function $F_{\text{num}}(x)$ instead of the estimate of function $F(x|y=1)$ in (37).

Remark. In order to explain choice (37) for function $\sigma(x)$, consider the problem of one-dimensional conditional probability estimation. Let $f_0(x)$ be the true conditional probability. Consider the function $\hat{f}_0(x) = p_1 f_0(x)$. Then the solution of integral equation

$$\int \theta(x-u)\hat{f}(u)dF(u) = F(x|y=1)$$

defines the conditional probability $\hat{f}_0(x) = p_1 f_0(x)$. Consider two functions: the estimate of the right-hand side of equation $F_\ell(x|y=1)$ and the actual right-hand side

$$F_0(x|y=1) = \int_{-\infty}^{x} \hat{f}_0(t)dt.$$

The deviation

$$\Delta = F_0(x|y=1) - F_\ell(x|y=1)$$

between these two functions has different values of variance for different x. The variance is small (equal to zero) at the end points of an interval and is large somewhere inside it. To obtain the *uniform relative deviation* of approximation from the actual function over the whole interval, we adjust the distance in any point of interval proportionally to the inverse of variance. Since for any fixed x the variance is

$$\mathrm{Var}(x) = F(x|y=1)(1 - F(x|y=1)), \tag{38}$$

we normalize the squared deviation Δ^2 by (38). The expression (37) realizes this idea.

5.1.2 Definition of Distance for Conditional Probability Estimation Problem

Consider the problem of conditional probability estimation.

For this problem, the squared distance between approximations of the right-hand side and the left-hand side of equation

$$F_\ell(x, y=1) = p_\ell F_\ell(x|y=1) = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i \theta(x - X_i)$$

can be written as follows:

$$\rho^2(A_\ell f, F_\ell) = \int \left(\int \theta(x-u) f(u) dF_\ell(u) - \int y_i \theta(x-u) dF_\ell(u) \right)^2 \sigma(x) d\mu(x),$$

where $y_i \in \{0, 1\}$ and $F_\ell(x)$ is the empirical distribution function estimated from training vectors X_i. Therefore, we obtain the expression

$$\begin{aligned} \rho^2(A_\ell f, F_\ell) = \quad & \frac{1}{\ell^2}\sum_{i,j=1}^{\ell} f(X_i) f(X_j) \int \theta(x - X_i)\theta(x - X_j)\sigma(x) d\mu(x) - \\ & \frac{2}{\ell^2}\sum_{i,j=1}^{\ell} f(X_i) y_j \int \theta(x - X_i)\theta(x - X_j)\sigma(x) d\mu(x) + \\ & \frac{1}{\ell^2}\sum_{i,j=1}^{\ell} y_i y_j \int \theta(x - X_i)\theta(x - X_j)\sigma(x) d\mu(x), \end{aligned} \tag{39}$$

where the last term does not depend on function $f(x)$.

Since both $\sigma(x)$ and $\mu(x)$ are products of one-dimensional functions, each integral in (39) has the form

$$V_{i,j} = \int \theta(x - X_i)\theta(x - X_j)\sigma(x)\, d\mu(x) = \prod_{k=1}^{d} \int \theta(x^k - X_i^k)\theta(x^k - X_j^k)\sigma_k(x^k) d\mu(x^k). \tag{40}$$

This $(\ell \times \ell)$-dimensional matrix of elements $V_{i,j}$ we call V-matrix of the sample $X_1, ..., X_\ell$, where $X_i = (X_i^1, \ldots X_i^d), \quad \forall i = 1, \ldots, \ell$.

Consider three cases:

Case 1. Data belongs to the upper-bounded support $(-\infty, B]^d$ for some B and $\sigma(x) = 1$ on this support. Then the elements $V_{i,j}$ of V-matrix have the form

$$V_{i,j} = \prod_{k=1}^{d} (B - \max\{X_i^k, X_j^k\}).$$

Case 2. Case where $\sigma(x^k) = 1$ and μ defined as (35). Then the elements $V_{i,j}$ of V-matrix have the form

$$V_{i,j} = \prod_{k=1}^{d} \nu(X^k > \max\{X_i^k, X_j^k\}).$$

where $\nu(X^k > \max\{X_i^k, X_j^k\})$ is the frequency of X^k from the given data with the values larger than $\max\{X_i^k, X_j^k\}$.

Case 3. Case where $\sigma(x)$ is defined as (36), (37) and $\mu(x)$ as (35). In this case, the values $V_{i,j}$ also can be easily computed numerically (since both functions are piecewise constant, the integration (40) is reduced to a summation of constants).

To rewrite the expression for the distance in a compact form, we introduce the ℓ-dimensional vector Φ

$$\Phi = (f(X_1), ..., f(X_\ell))^T.$$

Then, taking into account (39), we rewrite the first summand of functional (34) as

$$\rho^2(A_\ell f, F_\ell) = \frac{1}{\ell^2} \left(\Phi^T V \Phi - 2\Phi^T V Y + Y^T V Y \right), \tag{41}$$

where Y denotes the ℓ-dimensional vector $(y_1, ..., y_\ell)^T$, $y_i \in \{0, 1\}$.

5.1.3 Distance for Regression Estimation Problem

Repeating the same derivation for regression estimation problem, we obtain the same expression for the distance

$$\rho^2(A_\ell f, F_\ell) = \frac{1}{\ell^2} \left(\Phi^T V \Phi - 2\Phi^T V Y + Y^T V Y \right),$$

where coordinates of vector Y are values $y \in R^1$ given in examples (17) for regression estimation problem.

5.1.4 Distance for Density Ratio Estimation Problem

In the problem of density ratio estimation, we have to solve the integral equation

$$\int \theta(x - u) R(u) dF_{\text{den}}(u) = F_{\text{num}}(x),$$

where cumulative distribution functions $F_{\text{den}}(x)$ and $F_{\text{num}}(x)$ are unknown but iid data

$$X_1, ..., X_{\ell_{\text{den}}} \sim F_{\text{den}}(x)$$

and iid data

$$X'_1, ..., X'_{\ell_{\rm num}} \sim F_{\rm num}(x)$$

are available.

Using the empirical estimates

$$F_{\ell_{\rm num}}(x) = \frac{1}{\ell_{\rm num}} \sum_{j=1}^{\ell_{\rm num}} \theta(x - X'_j)$$

and

$$F_{\ell_{\rm den}}(x) = \frac{1}{\ell_{\rm den}} \sum_{i=1}^{\ell_{\rm den}} \theta(x - X_i)$$

instead of unknown cumulative distribution $F_{\rm num}(x)$ and $F_{\rm den}(x)$ and repeating the same distance computations as in the problems of conditional probability estimation and regression estimation, we obtain

$$\begin{aligned}
\rho^2 = & \int \left(\int \theta(x-u) R(u) dF_{\ell_{\rm den}}(u) - F_{\ell_{\rm num}}(x) \right)^2 \sigma(x) d\mu(x) = \\
& \frac{1}{\ell^2_{\rm den}} \sum_{i,j=1}^{\ell_{\rm den}} R(X_i) R(X_j) \int \theta(x - X_j) \theta(x - X_j) \sigma(x) d\mu(x) - \\
& \frac{2}{\ell_{\rm num} \ell_{\rm den}} \sum_{i=1}^{\ell_{\rm num}} \sum_{j=1}^{\ell_{\rm den}} R(X_i) R(X'_j) \int \theta(x - X_i) \theta(x - X'_j) \sigma(x) d\mu(x) + \\
& \frac{1}{\ell^2_{\rm num}} \sum_{i,j=1}^{\ell_{\rm num}} \int \theta(x - X'_j) \theta(x - X'_j) \sigma(x) d\mu(x) = \frac{1}{\ell^2_{\rm num}} \sum_{i,j=1}^{\ell_{\rm num}} V^{**}_{i,j} + \\
& \frac{1}{\ell^2_{\rm den}} \sum_{i,j=1}^{\ell_{\rm den}} R(X_i) R(X_j) V_{i,j} - \frac{2}{\ell_{\rm num} \ell_{\rm den}} \sum_{i=1}^{\ell_{\rm den}} \sum_{j=1}^{\ell_{\rm num}} R(X_i) R(X'_j) V^*_{i,j},
\end{aligned}$$

where the values $V_{i,j}, V^*_{i,j}, V^{**}_{i,j}$ are calculated as

$$\begin{cases}
V_{i,j} = & \int \theta(x - X_i) \theta(x - X_j) \sigma(x) d\mu(x), \quad i, j = 1, ..., \ell_{\rm den}, \\
V^*_{i,j} = & \int \theta(x - X_i) \theta(x - X'_j) \sigma(x) d\mu(x), \quad i = 1, ..., \ell_{\rm num}, \ j = 1, ..., \ell_{\rm den}, \\
V^{**}_{i,j} = & \int \theta(x - X'_i) \theta(x - X'_j) \sigma(x) d\mu(x), \quad i, j = 1, ..., \ell_{\rm num}.
\end{cases}$$

We denote by V, V^*, and V^{**} (respectively, $(\ell_{\rm den} \times \ell_{\rm den})$-dimensional, $(\ell_{\rm den} \times \ell_{\rm num})$-dimensional, and $(\ell_{\rm num} \times \ell_{\rm num})$-dimensional) the matrices of corresponding elements $V_{i,j}$, $V^*_{i,j}$, and $V^{**}_{i,j}$. We also denote by $1_{\ell_{\rm num}}$ the $\ell_{\rm num}$-dimensional vector of ones, and by R – the $\ell_{\rm den}$-dimensional column vector of $R(X_i)$, $i = 1, \ldots, \ell_{\rm den}$.

Using these notations, we can rewrite the distance as follows:

$$\rho^2 = \frac{1}{\ell^2_{\rm den}} \left(R^T V R - 2 \left(\frac{\ell_{\rm den}}{\ell_{\rm num}} \right) R^T V^* 1_{\ell_{\rm num}} + \left(\frac{\ell_{\rm den}}{\ell_{\rm num}} \right)^2 1^T_{\ell_{\rm num}} V^{**} 1_{\ell_{\rm num}} \right).$$

5.2 The Regularization Functionals in RKHS

For each of our inference problems, we now look for its solution in Reproducing Kernel Hilbert Space (RKHS).

5.2.1 Reproducing Kernel Hilbert Space

According to Mercer theorem, any positive semi-definite kernel has a representation

$$K(x,z)=\sum_{k=1}^{\infty}\lambda_k\phi_k(x)\phi_k(z),\quad x,z\in\mathcal{X},$$

where $\{\phi_k(x)\}$ is a system of orthonormal functions and $\lambda_k\geq 0\ \ \forall k$.

Consider the set of functions

$$f(x;a)=\sum_{k=1}^{\infty}a_k\phi_k(x). \tag{42}$$

We say that set of functions (42) belongs to RKHS of kernel $K(x,z)$ if we can define the inner product (f_1,f_2) in this space such that

$$(f_1(x),K(x,y))=f_1(y). \tag{43}$$

It is easy to check that the inner product

$$(f(x,a),f(x,b))=\sum_{k=1}^{\infty}\frac{a_kb_k}{\lambda_k},$$

where a_k and b_k are the coefficients of expansion of functions $f(x,a)$, and $f(x,b)$, satisfies the reproducing property (43). In particular, the equality

$$(K(x_1,z),K(x_2,z))=K(x_1,x_2) \tag{44}$$

holds true for the kernel $K(x,x^*)$ that defines RKHS.

The remarkable property of RKHS is the so-called Representer Theorem (Kimeldorf and Wahba, 1971), (Kimeldorf and Wahba, 1970), (Schölkopf et al., 2001), which states that any function $f(x)$ from RKHS that minimizes functional (34) can be represented as

$$f(x)=\sum_{i=1}^{\ell}c_iK(X_i,x),$$

where $c_i,\ i=1,...,\ell$ are parameters and $X_i,\ i=1,...,\ell$ are vectors of observations.

5.2.2 Explicit Form of Regularization Functional.

In all our Statistical Inference problems, we are looking for solutions in RKHS, where we use the squared norm as the regularization functional:

$$W(f)=(f,f)=||f||^2. \tag{45}$$

That is, we are looking for solution in the form

$$f(x) = \sum_{i=1}^{\ell} \alpha_i K(X_i, x), \tag{46}$$

where X_i are elements of the observation. Using property (44), we define the functional (45) as

$$W(f) = \sum_{i,j=1}^{\ell} \alpha_i \alpha_j K(x_i, x_j).$$

In order to use the matrix form of (34), we introduce the following notations:

1. K is the $(\ell \times \ell)$-dimensional matrix of elements $K(X_i, X_j),\ i, j = 1, ..., \ell$.
2. $\mathcal{K}(x)$ is the ℓ-dimensional vector of functions $K(X_i, x),\quad i = 1, ..., \ell$.
3. A is the ℓ-dimensional vector $A = (\alpha_1, ..., \alpha_\ell)^T$ of elements $\alpha_i,\quad i = 1, ..., \ell$.

In these notations, the regularization functional has the form

$$W(f) = A^T K A, \tag{47}$$

and its solution has the form

$$f(x) = A^T \mathcal{K}(x). \tag{48}$$

6. Solution of Statistical Inference Problems

In this section, we formulate our statistical inference problems as optimization problems.

6.1 Estimation of Conditional Probability Function

Here we present an explicit form of the optimization problem for estimating conditional probability function.

We are looking for the solution in form (48), where we have to find vector A. In order to find it, we have to minimize the objective function

$$T(A) = A^T K V K A - 2 A^T K V Y + \gamma A^T K A, \tag{49}$$

where Y is a binary vector (with coordinates $y \in \{0, 1\}$) defined by the observations. The first two terms of the objective function come from distance (41), the last term is regularization functional (47). (The third term from (49) was omitted in the target functional since it does not depend on the unknown function.) Since the conditional probability has values between 0 and 1, we have to minimize this objective function subject to the constraint

$$0 \le A^T \mathcal{K}(x) \le 1, \quad \forall x \in X. \tag{50}$$

We also know that

$$\int A^T \mathcal{K}(x) dF(x) = p_0, \tag{51}$$

where p_0 is the probability of class $y = 1$.

Minimization of (49) subject to constraints (50), (51) is a difficult optimization problem. To simplify this problem, we minimize the functional subject to the constraints

$$0 \le A^T \mathcal{K}(X_i) \le 1, \ i = 1, ..., \ell, \tag{52}$$

defined only at the vectors X_i of observations[5].

Also, we can approximate equality (51) using training data

$$\frac{1}{\ell} \sum_{i=1}^{\ell} A^T \mathcal{K}(X_i) = p_\ell, \tag{53}$$

where p_ℓ is the frequency of class $y = 1$ estimated from data. Using matrix notation, the constraints (52) and (53) can be rewritten as follows:

$$0_\ell \le KA \le 1_\ell, \tag{54}$$

$$\frac{1}{\ell} A^T K 1_\ell = p_\ell. \tag{55}$$

where K is the matrix of elements $K(X_i, X_j), \ i, j = 1, ..., \ell$ and 0_ℓ, 1_ℓ are ℓ-dimensional vectors of zeros and ones, respectively.

Therefore, we are looking for the solution in form (48), where parameters of vector A minimize functional (49) subject to constraints (54) and (55). This is a quadratic optimization problem with one linear equality constraint and 2ℓ *general* linear inequality constraints.

In Section 6.4, we simplify this optimization problem by reducing it to a quadratic optimization problem with one linear equality constraint and several *box* constraints.

6.2 Estimation of Regression Function

Similarly, we can formulate the problem of regression function estimation, which has the form (48). To find the vector A, we minimize the functional

$$T(A) = A^T KVKA - 2A^T KVY + \gamma A^T KA, \tag{56}$$

where Y is a real-valued vector (with coordinates $y_i \in R^1$ of observations (5)).

Suppose that we have the following knowledge about the regression function:

1. Regression $y = f(x) = A^T \mathcal{K}(x)$ takes values inside an interval $[a, b]$:

$$a \le A^T \mathcal{K}(x) \le b, \quad \forall x \in \mathcal{X}. \tag{57}$$

2. We know the expectation of the values of the regression function:

$$\int A^T \mathcal{K}(x) dF(x) = c. \tag{58}$$

5. One can find the solution in closed form $A = (VK + \gamma I)^{-1} VY$ if constraints (52), (53) are ignored; here I is the identity matrix.

Then we can solve the following problem: minimize functional (56) subject to constraints (57), (58).

Usually we do not have knowledge (57), (58), but we can approximate it from the training data. Specifically, we can approximate a by the smallest value a_ℓ of y_i, while b can be approximated by the largest value b_ℓ of y_i from the training set:

$$a_\ell = \min\{y_1, ..., y_\ell\}, \quad b_\ell = \max\{y_1, ..., y_\ell\}.$$

We then consider constraint (57) applied only for the training data:

$$a_\ell \le A^T \mathcal{K}(X_i) \le b_\ell, \quad i = 1, ..., \ell. \tag{59}$$

Also, we can approximate (58) with the equality constraint

$$\frac{1}{\ell} \sum_{i=1}^{\ell} A^T \mathcal{K}(X_i) = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i. \tag{60}$$

Constraints (59), (60) can be written in matrix notation

$$a_\ell 1_\ell \le KA \le 1_\ell b_\ell, \tag{61}$$

$$\frac{1}{\ell} A^T K 1_\ell = \hat{y}_{av}, \tag{62}$$

where $\hat{y}_{av}$ is the right-hand side of (60). If these approximations[6] are reasonable, the problem of estimating the regression can be stated as minimization of functional (56) subject to constraints (61), (62). This is a quadratic optimization problem with one linear equality constraint and 2ℓ general linear inequality constraints.

6.3 Estimation of Density Ratio Function

To solve the problem of estimating density ratio function in the form

$$R(x) = A^T \mathcal{K}(x),$$

where A is the ℓ_{den}-dimensional vector of parameters and $\mathcal{K}(x)$ is the ℓ_{den}-dimensional vector of functions $K(X_1, x), ..., K(X_{\ell_{\text{den}}}, x)$, we have to minimize the functional

$$T(A) = A^T KVKA - 2\left(\frac{\ell_{\text{den}}}{\ell_{\text{num}}}\right) A^T KV^* \mathbf{1}_{\ell_{\text{num}}} + \gamma A^T KA, \tag{63}$$

where K is the $(\ell_{\text{den}} \times \ell_{\text{den}})$-dimensional matrix of elements $K(X_i, X_j)$ subject to the constraints

$$A^T \mathcal{K}(x) \ge 0, \quad \forall x \in X,$$

$$\int A^T \mathcal{K}(x) dF_{\text{den}}(x) = 1.$$

6. Without constraints, the solution has the closed form (see footnote 5), where $y \in R^1$ are elements of training data for regression.

As above, we replace these constraints with their approximations

$$KA \geq \mathbf{0}_{\ell_{\text{den}}},$$

$$\frac{1}{\ell_{\text{den}}} A^T K V^* 1_{\ell_{\text{num}}} = 1.$$

Here K is $(\ell_{\text{den}} \times \ell_{\text{den}})$-dimensional matrix of observations from $F_{\text{den}}(x)$, and V^* is $(\ell_{\text{den}} \times \ell_{\text{num}})$-dimensional matrix defined in Section 5.1.

6.4 Two-Stage Method for Function Estimation: Data Smoothing and Data Interpolation

Solutions of Statistical Inference problems considered in the previous sections require numerical treatment of the general quadratic optimization problem: minimization of quadratic form subject to one linear equality constraint and 2ℓ linear inequality constraints of general type (ℓ linear inequality constraints for density ratio estimation problem).

Numerical solution for such problems can be computationally hard (especially when ℓ is large). In this section, we simplify the problem by splitting it into two stages:

1. Estimating function values at ℓ observation points, that is, the estimating vector $\Phi = (f(X_1), ..., f(X_\ell))^T$.

2. Interpolating the values of function known at the ℓ observation points to other points in the space $\mathcal{X}$.

6.4.1 Stage 1: Estimating Function Values at Observation Points

In order to find the function values at the training data points, we rewrite the regularization functional in objective functions (49), (56), (63) in a different form. In order to do this, we use the equality

$$K = KK^+K,$$

where K^+ is the *generalized inverse* matrix of matrix[7] K.

In our regularization term of objective functions, we use the equality

$$A^T KA = A^T KK^+ KA.$$

1. Estimation of values of conditional probability. For the problem of estimating the values of conditional probability at ℓ observation points, we rewrite objective function (49) in the form

$$W(\Phi) = \Phi^T V\Phi - 2\Phi^T VY + \gamma\Phi^T K^+\Phi, \tag{64}$$

where we have denoted

$$\Phi = KA. \tag{65}$$

In the problem of estimating conditional probability, Y is a binary vector.

7. Along with generalized inverse matrix, pseudoinverse matrix is also used. Pseudoinverse matrix M^+ of the matrix M (not necessarily symmetric) satisfies the following four conditions: (1) $MM^+M = M$, (2) $M^+MM^+ = M^+$, (3) $(MM^+)^T = MM^+$, and (4) $(M^+M)^T = M^+M$. If matrix M is invertible, then $M^+ = M^{-1}$. Pseudoinverse exists and is unique for any matrix.

In order to find vector Φ, we minimize functional (64) subject to box constraints

$$\mathbf{0}_\ell \leq \Phi \leq 1_\ell,$$

and equality constraint

$$\frac{1}{\ell}\Phi^T 1_\ell = p_\ell.$$

2. Estimating values of regression. In order to estimate the vector Φ of values of regression at ℓ observation points, we minimize functional (64) (where Y is a real-valued vector), subject to the box constraints

$$a_\ell 1_\ell \leq \Phi \leq b_\ell 1_\ell,$$

and the equality constraint

$$\frac{1}{\ell}\Phi^T 1_\ell = \hat{y}_{av}.$$

3. Estimating values of density ratio function. In order to estimate the vector Φ of values of density ratio function at ℓ_{den} observation points $X_1, ..., X_{\ell_{\text{den}}}$, we minimize the functional

$$\Phi^T V \Phi - 2\left(\frac{\ell_{\text{den}}}{\ell_{\text{num}}}\right)\Phi^T V^* \mathbf{1}_{\ell_{\text{num}}} + \gamma \Phi^T K^+ \Phi$$

subject to the box constraints

$$\Phi \geq \mathbf{0}_{\ell_{\text{den}}},$$

and the equality constraint

$$\frac{1}{\ell_{\text{den}}}\Phi^T V^* 1_{\ell_{\text{num}}} = 1.$$

6.4.2 Stage 2: Function Interpolation

In the second stage of our two-stage procedure, we use the estimated function values at the points of training set to define the function in input space. That is, we solve the problem of function interpolation.

In order to do this, consider representation (65) of vector Φ^*:

$$\Phi^* = KA^*. \tag{66}$$

We also consider the RKHS representation of the desired function:

$$f(x) = A^{*T}\mathcal{K}(x). \tag{67}$$

If the inverse matrix K^{-1} exists, then

$$A^* = K^{-1}\Phi^*.$$

If K^{-1} does not exist, there are many different A^* satisfying (66). In this situation, the best interpolation of Φ^* is a (linear) function (67) that belongs to the subset of functions with the smallest bound on VC dimension (Vapnik, 1998). According to Theorem 10.6

in (Vapnik, 1998), such a function either satisfies equation (66) with the smallest L_2 norm of A^* or it satisfies equation (66) with the smallest L_0 norm of A^*.

Efficient computational implementations for both L_0 and L_2 norms are available in the popular scientific software package Matlab.

Note that the obtained solutions in all our problems satisfy the corresponding constraints only on the training data, but they do not have to satisfy these constraints at any $x \in \mathcal{X}$. Therefore, we truncate the obtained solution functions as

$$f_{tr}(x) = [A^{*T}\mathcal{K}(x)]_a^b,$$

where

$$[u]_a^b = \begin{cases} a, & \text{if } u < a \\ u, & \text{if } a \le u \le b \\ b, & \text{if } u > b \end{cases}$$

Remark. For conditional probability estimation, the choice of $a > 0$, $b < 1$ (for constraints in training and truncation in test) is an additional tool for regularization that can leverage prior knowledge.

6.4.3 Additional Considerations

For many problems, it is useful to consider the solutions in the form of a function from a set of RKHS functions with a bias term:

$$f(x) = \sum_{i=1}^{\ell} \alpha_i K(X_i, x) + c = A^T\mathcal{K}(x) + c.$$

Using this set of functions, our quadratic optimization formulation for estimating the function values at training data points for the problem of conditional probability and regression estimation is as follows: minimize the functional (over vectors Φ)

$$(\Phi + c1_\ell)^T V(\Phi + c1_\ell) - 2(\Phi + c1_\ell)^T VY + \gamma\Phi^T K^+\Phi$$

subject to the constraints

$$(a - c1_\ell) \le \Phi \le (b - c1_\ell),$$

(where $a = 0,\ b = 1$ for conditional probability problem, and $a = a_\ell,\ b = b_\ell$ for regression problem).

$$\frac{1}{\ell}1_\ell^T + c = \hat{y}_{av}$$

where we denoted

$$\hat{y}_{av} = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i.$$

For estimating the values of density ratio function at points $(X_1, \ldots, X_{\ell_{\text{den}}})$, we minimize the functional

$$(\Phi + c1_{\ell_{\text{den}}})^T V(\Phi + c1_{\ell_{\text{den}}}) - 2\left(\frac{\ell_{\text{den}}}{\ell_{\text{num}}}\right)(\Phi + c1_{\ell_{\text{den}}})^T V^* 1_{\ell_{\text{num}}} + \gamma\Phi^T K^+\Phi$$

subject to the constraints

$$-c1_{\ell_{\text{den}}} \le \Phi,$$
$$\Phi^T 1_{\ell_{\text{den}}} + c\ell_{\text{den}} = \ell_{\text{den}}.$$

7. Applications of Density Ratio Estimation

Here we describe three applications of density ratio estimation (Sugiyama et al., 2012), (Kawahara and Sugiyama, 2009), specifically,

– Data adaptation and correction of solution for unbalanced data.

– Estimation of mutual information and problem of feature selection.

– Change point detection.

It is important to note that, in all these problems, it is required to estimate not the function $R(x)$, but rather the values $R(X_i)$ of density ratio function at the points $X_1, ..., X_{\ell_{\text{den}}}$ (generated by probability measure $F_{\text{den}}(x)$).

Below we consider the first two problems in the pattern recognition setting and then consider two new applications:

1) Learning from data with unbalanced classes

2) Learning of local rules.

7.1 Data Adaptation Problem

Let the iid data

$$(y_1, X_1), ..., (y_\ell, X_\ell) \tag{68}$$

be defined by a fixed unknown density function $p(x)$ and a fixed unknown conditional density function $p(y|x)$ generated according to an unknown joint density function $p(x, y) = p(y|x)p(x)$. Suppose now that one is given data

$$X_1^*, ..., X_{\ell_1}^* \tag{69}$$

defined by another fixed unknown density function $p_*(x)$. This density function, together with conditional density $p(y|x)$ (the same one as for Equation 68), defines the joint density function $p_*(x, y) = p(y|x)p_*(x)$.

It is required, using data (68) and (69), to find in a set of functions $f(x, \alpha),\ \alpha \in \Lambda$, the one that minimizes the functional

$$T(\alpha) = \int L(y, f(x, \alpha))p_*(x, y)dydx, \tag{70}$$

where $L(\cdot, \cdot)$ is a known loss function.

This setting is an important generalization of the classical function estimation problem where the functional dependency between variables y and x is the same (the function $p(y|x)$ which is the part of composition of $p(x, y)$ and $p_*(x, y)$), but the environments (defined by densities $p(x)$ and $p_*(x)$) are different.

It is required, by observing examples from one environment (with $p(x)$), to define the rule for another environment (with $p^*(x)$). Let us denote

$$R(x) = \frac{p_*(x)}{p(x)}, \quad p(x) > 0.$$

Then functional (70) can be rewritten as

$$T(\alpha) = \int L(y, f(x,\alpha))R(x)p(x,y)dydx,$$

and we have to minimize the functional

$$T_\ell(\alpha) = \sum_{i=1}^{\ell} L(y_i, f(X_i,\alpha))R(x_i),$$

where $X_i,\ y_i$ are data points from (68). In this equation, we have multipliers $R(X_i)$ that define the adaptation of data (69) generated by joint density $p(x,y) = p(y|x)p(x)$ to the data generated by the density $p_*(x,y) = p(y|x)p_*(x)$. Knowledge of density ratio values $R(X_i)$ leads to a modification of classical algorithms.

For SVM method in pattern recognition (Vapnik, 1995), (Vapnik, 1998), this means that we have to minimize the functional

$$T_\ell(w) = (w,w) + C\sum_{i=1}^{\ell} R(X_i)\xi_i \tag{71}$$

(C is a tuning parameter) subject to the constraints

$$y_i((w,z_i)+b) \geq 1-\xi_i, \quad \xi \geq 0, \quad y_i \in \{-1,+1\}, \tag{72}$$

where z_i is the image of vector $X_i \in \mathcal{X}$ in feature space $\mathcal{Z}$.

This leads to the following dual-space SVM solution: maximize the functional

$$T_\ell(\alpha) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j K(X_i,X_j), \tag{73}$$

where $(z_i,z_j) = K(X_i,X_j)$ is Mercer kernel that defines the inner product (z_i,z_j) subject to the constraint

$$\sum_{i=1}^{\ell} y_i\alpha_i = 0 \tag{74}$$

and the constraints

$$0 \leq \alpha_i \leq CR(X_i). \tag{75}$$

The adaptation to new data is given by the values $R(x_i),\ i = 1,...,\ell$; these values are set to 1 in standard SVM (71).

7.2 Estimation of Mutual Information.

Consider k-class pattern recognition problem $y \in \{a_1,...,a_k\}$.

The *entropy* of nominal random variable y (level of uncertainty for y with no information about corresponding x) is defined by

$$H(y) = -\sum_{t=1}^{k} p(y=a_t)\log_2 p(y=a_t).$$

Similarly, the *conditional entropy* given fixed value x_* (level of uncertainty of y given information x_*) is defined by the value

$$H(y|x_*) = -\sum_{t=1}^{k} p(y = a_t|x_*) \log_2 p(y = a_t|x_*).$$

For any x^*, the difference (decrease in uncertainty)

$$\Delta H(y|x_*) = H(y) - H(y|x_*)$$

defines the amount of information about y contained in vector x_*. The expectation of this value (with respect to x)

$$I(x, y) = \int \Delta H(y|x) dF(x)$$

is called the *mutual information* between variables y and vectors x. It describes how much information vector x caries about variable y. The mutual information can be rewritten in the form

$$I(x, y) = \sum_{t=1}^{k} p(y = a_t) \int \left(p(x, y = a_t) \log_2 \frac{p(x, y = a_t)}{p(x)p(y = a_t)} \right) dF(x) \tag{76}$$

(see (Cover and Thomas, 2006) for details).

For two densities ($p(x|y = a_t)$ and $p(x)$), the density ratio function is

$$R(x, y = a_t) = \frac{p(x|y = a_t)}{p(x)}.$$

Using this notation, one can rewrite expression (76) as

$$I(x, y) = \sum_{t=1}^{k} p(y = a_t) \int R(y = a_t, x) \log_2 R(y = a_t, x) dF(x), \tag{77}$$

where $F(x)$ is cumulative distribution function of x.

Our goal is to use data

$$(y_1, X_1), ..., (y_\ell, X_\ell)$$

to estimate $I(x, y)$. Using in (77) the empirical distribution function $F_\ell(x)$ and the values $p_\ell(y = a_t)$ estimated from the data, we obtain the approximation $I_\ell(x, y)$ of mutual information (77):

$$I_\ell(x, y) = \frac{1}{\ell} \sum_{t=1}^{m} p(y = a_t) \sum_{i=1}^{\ell} R(X_i, y = a_t) \log_2 R(X_i, y = a_t).$$

Therefore, in order to estimate the mutual information for k-class classification problem, one has to solve the problem of values of density ratio estimation problem k times at the observation points $R(X_i, y = a_t),\ i = 1, ..., \ell$ and use these values in (77).

Feature selection problem and mutual information. Estimates of mutual information play important role in the problem of feature selection. Indeed, the problem of selecting k features from the set of n features require to find among n features $x^1, .., x^n$ such k elements $x^{k_1}, ..., x^{k_k}$ which contain *maximal information* about variable y generated according to $p(y|x)$, $x = (x^1, ..., x^n)$. That means to find the subset of k elements with maximal mutual information. This is a hard computational problem: even if one can estimate the mutual information from data well, one still needs to solve mutual information estimation problem C_n^k times to chose the best subset.

Therefore some heuristic methods are used (Brown et al., 2012) to chose the subset with best features. There are two heuristic approaches to the problem of estimating best features:

1. To estimate mutual information $I(y, x^t)$ or $I(x^t, x^m)$ of scalar values and then combine (heuristically) the results.

2. To estimate mutual information between the value of y and two features $x_{t,m} = (x^t, x^m)$, obtaining n^2 elements of matrix $I(y, x_{t,m})$ $t, m = 1, \ldots, n$ and choose from this matrix the minor with the largest score (say, the sum of its elements).

All these procedures require accurate estimates of mutual information.

7.3 Unbalanced Classes in Pattern Recognition

An important application of data adaptation method is the case of binary classification problem with unbalanced training data (du Plessis and Sugiyama, 2012). In this case, the numbers of training examples for both classes differ significantly (often, by orders of magnitude). For instance, for diagnosis of rare diseases, the number of samples from the first class (patients suffering from the disease) is much smaller than the number of samples from the second class (patients without that disease).

Classical pattern recognition algorithms applied to unbalanced data can lead to large false positive or false negative error rates. We would like to construct a method that would allow to control the balance of both error rates. Formally, this means that training data are generated according to some probability measure

$$p(x) = p(x|y = 1)p + p(x|y = 0)(1 - p),$$

where $0 \le p \le 1$ is a fixed parameter that defines probability of the event of the first class. Learning algorithms are developed to minimize the expectation of error for this generator of random events.

Our goal, however, is to minimize the expected error for another generator

$$p_*(x) = p(x|y = 1)p_* + p(x|y = 0)(1 - p_*),$$

where p_* defines different probability of the first class (in the rare disease example, we minimize the expected error if this disease is not so rare); that is, for parameter $p = p_*$.

To solve this problem, we have to estimate the values of density ratio function

$$R(x) = \frac{p_*(x)}{p(x)}$$

from available data. Suppose we are given observations

$$(y_1, X_1), ..., (y_\ell, X_\ell). \tag{78}$$

Let us denote by X_i^1 and X_j^0 vectors from (78) corresponding to $y = 1$ and $y = 0$, respectively. We rewrite elements of x from (78) generated by $p(x)$ as

$$X_{i_1}^1, ..., X_{i_m}^1, X_{i_{m+1}}^0, ..., X_{i_\ell}^0$$

Consider the new training set that imitates iid observations generated by $p_*(x)$ by having the elements of the first class to have frequency p_*:

$$X_{i_1}^1, ..., X_{i_m}^1, X_{j_1}^1, ...X_{j_s}^1, X_{i_{m+1}}^0, ..., X_{i_\ell}^0, \tag{79}$$

where $X_{j_1}^1, \ldots, X_{j_s}^1$ are the result of random sampling from $X_{i_1}^1, \ldots, X_{i_m}^1$ with replacement. Now, in order to estimate values $R(X_i), i = 1, ..., \ell$, we construct function $F_{\ell_{\text{den}}}(x)$ from data (78) and function $F_{\ell_{\text{num}}}(x)$ from data (79) and use the algorithm for density ratio estimation. For SVM method, in order to balance data, we have to maximize (73) subject to constraints (74) and (75).

8. Problem of Local Learning

In 1992, the following problem of local learning was formulated (Bottou and Vapnik, 1992): given data

$$(x_1, y_1), ..., (x_\ell, y_\ell) \tag{80}$$

generated according to an unknown density function

$$p_0(y, x) = p_0(y|x)p_0(x),$$

find the decision rule that minimizes risk *in a vicinity of the given point* x_0. Using some heuristic concept of vicinity of given points, the corresponding algorithm was developed. It was demonstrated (Bottou and Vapnik, 1992), (Vapnik and Bottou, 1993) that local learning is often more accurate than the global learning.

In this Section, we present a reasonable definition of the concept of locality, and we solve the problem of constructing local rules. Our goal is to use data (80) for constructing a rule that is accurate for vectors distributed according to some $p_{loc}(x)$, for example, according to the multidimensional normal distribution

$$p_{loc}(x) = N(x_0, \sigma I) = \frac{1}{(2\pi)^{m/2}\sigma^m} \prod_{k=1}^{m} \exp\left\{-\frac{(x^k - x_0^k)}{2\sigma^2}\right\},$$

where $x_0 = (x_0^1, ..., x_0^m)$ is the vector of means, $\sigma > 0$ and identity matrix I are known parameters of multi-dimensional normal distribution (they are specified by our concept of vicinity point x_0). We denote by $p_{loc}(y, x)$ the density function

$$p_{loc}(y, x) = p_0(y|x)N(x_0, \sigma I).$$

Therefore, the goal of local learning is to find, in the set of functions $f(x,\alpha), \alpha \in \Lambda$, the function $f(x,\alpha_n)$ that minimizes the functional

$$T_{loc}(\alpha) = \int (y - f(x,\alpha))^2 p_{loc}(y,x) dy dx$$

instead of the functional

$$T_0(\alpha) = \int (y - f(x,\alpha))^2 p_0(y,x) dy dx,$$

as it is formulated in classical (global) learning paradigm.

We rewrite functional T_{loc} as follows:

$$T_{loc}(\alpha) = \int (y - f(x,\alpha))^2 R(x) p_0(y,x) dy dx,$$

where we have denoted

$$R(x) = \frac{p_{loc}(x)}{p_0(x)}.$$

To minimize the functional T_{loc} given data obtained according to $p_0(y,x)$, we minimize the empirical risk

$$\hat{R}(\alpha) = \frac{1}{\ell} \sum_{i=1}^{\ell} (y_i - f(x_i,\alpha))^2 R(x_i).$$

To define this functional explicitly, we have to estimate the ratio of two densities, one of which (the density that specifies vicinity of point x_0) is known, and another one is unknown but elements x of data obtained according to that unknown density $p_0(x)$ are available from the training set.

This problem of density ratio estimation, however, differs from the one considered in Section 6.3. It requires solving the integral equation when the right-hand side of equation is precisely defined, whereas the operator is defined approximately:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} \theta(x - X_i) R(x_i) \approx \int_{-\infty}^{x} N(x_0, \sigma I) dx.$$

The integral in the right-hand can be expressed as the product of m (where m is the dimensionality of space X) functions

$$\mathrm{Erf}(x_* | x_0, \sigma I) = \frac{2^m}{(\pi)^{m/2}} \prod_{k=1}^{m} \int_0^{x_*^k} \exp\left\{ -\frac{(x^k - x_0^k)^2}{2\sigma^2} dx^k \right\}.$$

Note that function $\mathrm{Erf}(x|x_0,\sigma I)$ can be easily computed using the so-called function $\mathrm{erf}(x)$:

$$\mathrm{erf}(x^k) = \frac{2}{\sqrt{\pi}} \int_0^{x^k} \exp\{-t^2\} dt,$$

which is tabulated.

Using the method of estimation of density ratio, in order to estimate vector $R = (R(x_1), ..., R(x_\ell))$, one has to minimize the functional

$$R^T V R - 2\ell R^T U + \gamma R^T K^+ R,$$

where we have denoted by $U = (U_1, ..., U_\ell)^T$ the vector with coordinates

$$U_i = \prod_{k=1}^{m} \sum_{t=1}^{\ell} \theta(x_t^k - x_i^k) \mathrm{Erf}(x_t^k | x_0, \sigma)$$

subject to constraints

$$R^T 1 = \ell$$

and constraints

$$R \geq 0_\ell.$$

In SVM technology, the V-matrix method requires to estimate values $R(X_i)$ in the points of observations first, and then to solve the SVM problem itself using the data adaptation technique described in Section 6.3.

9. Comparison with Classical Methods

In this paper, we introduced a new unified approach to solution of statistical inference problems based on their direct settings. We used rigorous mathematical techniques to solve them. Surprisingly, all these problems are amenable to relatively simple solutions.

One can see that elements of such solutions already exist in the basic classical statistical methods, for instance, in estimation of linear regression and in SVM pattern recognition problems.

9.1 Comparison with Linear Methods

Estimation of linear regression function is an important part of classical statistics. It is based on iid data

$$(y_1, X_1), ..., (y_\ell, X_\ell), \tag{81}$$

where y is distributed according to an unknown function $p(y|x)$. Distribution over vectors x is a subject of special discussions: it could be either defined by an unknown $p(x)$ or by known fixed vectors. It is required to estimate the linear regression function

$$y = w_0^T x.$$

Linear estimator. To estimate this function, classical statistics uses *ridge regression method* that minimizes the functional

$$R(w) = (Y - \mathbf{X}w)^T (Y - \mathbf{X}w) + \gamma(w, w), \tag{82}$$

where $\mathbf{X}$ is the $(\ell \times n)$-dimensional matrix of observed vectors X, and Y is the $(\ell \times 1)$-dimensional matrix of observations y. This approach also covers the least squares method (for which $\gamma = 0$).

When observed vectors X in $\mathbf{X}$ are distributed according to an unknown $p(x)$, method (81) is consistent under very general conditions.

The minimum of this functional has the form

$$w_\ell = (\mathbf{X}^T\mathbf{X} + \gamma I)^{-1}\mathbf{X}^T Y. \tag{83}$$

However, estimate (82) is not necessarily the best possible one.

The main theorem of linear regression theory, the *Gauss-Markov* theorem, assumes that input vectors X in $\mathbf{X}$ (81) are fixed (they are not random!). Below we formulate it in a slightly more general form.

Theorem. *Suppose that the random values* $(y_i - w_0^T X_i)$ *and* $(y_j - w_0^T X_j)$ *are uncorrelated and that the bias of estimate (82)*

$$\mu = E_y(w_\ell - w_0).$$

Then, among all linear[8] *estimates with bias*[9] μ*, estimate (82) has the smallest expectation of squared deviation:*

$$E_y(w_0 - w_\ell)^2 \le E_y(w_0 - w)^2, \quad \forall w.$$

Generalized linear estimator. Gauss-Markov model can be extended in the following way. Let ℓ-dimensional vector of observations Y be defined by fixed vectors X and additive random noise $\Omega = (\varepsilon_1, ..., \varepsilon_\ell)^T$ so that

$$Y = \mathbf{X}w_0 + \Omega,$$

where the noise vector $\Omega = (\varepsilon_1, ..., \varepsilon_\ell)^T$ is such that

$$E\Omega = 0, \tag{84}$$

$$E\Omega\Omega^T = \Sigma. \tag{85}$$

Here, the noise values at the different points X_i and X_j of matrix $\mathbf{X}$ are correlated and the correlation matrix Σ is *known* (in the classical Gauss-Markov model, it is identity matrix $\Sigma = I$). Then, instead of estimator (82) minimizing functional (81), one minimizes the functional

$$R(w) = (Y - \mathbf{X}w)^T\Sigma^{-1}(Y - \mathbf{X}w) + \gamma(w, w). \tag{86}$$

This functional is obtained as the result of de-correlation of noise in (83), (84). The minimum of (85) has the form

$$\hat{w}_* = (\mathbf{X}^T\Sigma^{-1}\mathbf{X} + \gamma I)^{-1}\mathbf{X}^T\Sigma^{-1}Y. \tag{87}$$

This estimator of parameters w is an improvement of (82) for correlated noise vector.

V-matrix estimator of linear functions. The method of solving regression estimation problem (ignoring constraints) with V matrix leads to the estimate

$$\hat{w}_{**} = (\mathbf{X}^T V\mathbf{X} + \gamma I)^{-1}\mathbf{X}^T VY.$$

8. Note that estimate (83) is linear only if matrix X is fixed.
9. Note that when $\gamma = 0$ in (82), the estimator (82) with $\gamma = 0$ is unbiased.

The structure of the V-matrix based estimate is the same as those of linear regression estimates (82) and (86), except that the V-matrix replaces identity matrix in (82) and inverse covariance matrix in (86).

The significant difference, however, is that both classical models were developed for the known (fixed) vectors X, while V-matrix is defined for random vectors X and is computed using these vectors. It takes into account the information that classical methods ignore: the domain of regression function and the geometry of observed data points. The complete solution also takes into accounts the constraints that reflects the belief in estimated prior knowledge about the solution.

9.2 Comparison with L_2-SVM (Non-Linear) Methods

For simplicity, we discuss in this section only pattern recognition problem; we can use the same approach for the non-linear regression estimation problem.

The pattern recognition problem can be viewed as a special case of the problem of conditional probability estimation. Using an estimate of conditional probability $p(y = 1|x)$, one can easily obtain the classification rule

$$f(x) = \theta(p(y = 1|x) - 1/2).$$

We now compare the solution $f(x)$ with

$$f(x) = A^T \mathcal{K}(x)$$

obtained for conditional probability problem with the same form of solution that defines SVM.

The coefficients A for L_2-SVM have the form (Saunders et al., 1998), (Suykens and Vandewalle, 1999)

$$A = (K + \gamma I)^{-1} Y. \tag{88}$$

If V-matrix method ignores the prior knowledge about the properties of conditional probability function, the coefficients of expansion have the form

$$A = (KV + \gamma I)^{-1} VY. \tag{89}$$

It is easy, however, to incorporate the existing constraints into both solutions.

In order to find the standard hinge-loss SVM solution (Vapnik, 1995), we have to minimize the quadratic form

$$-A^T \mathcal{Y} K \mathcal{Y} A + 2A^T \mathbf{1}_\ell$$

with respect to A subject to the box constraint

$$\mathbf{0}_\ell \leq A \leq C\mathbf{1}_\ell$$

and the equality constraint

$$A^T \mathcal{Y} \mathbf{1}_\ell = 0,$$

where C is the (penalty) parameter of the algorithm, and $\mathcal{Y}$ is $(\ell \times \ell)$-dimensional diagonal matrix with $y_i \in \{-1, +1\}$ from training data on its diagonal (see formulas (71), (72) , (73), (74), and (75) with $R(x_i) = 1$ in (71) and (75)).

In order to find the values of conditional probability, we also have to minimize the quadratic form

$$\Phi^T(V + \gamma K^+)\Phi - 2\Phi VY,$$

with respect to Φ subject to the box constraints[10]

$$\mathbf{0}_\ell \le \Phi \le \mathbf{1}_\ell$$

and the equality constraint

$$\Phi^T \mathbf{1}_\ell = \ell p_\ell,$$

where $\gamma > 0$ is the (regularization) parameter of the algorithm in the objective function (as C for SVM).

The essential difference between SVM and V-matrix method is that the constraints in SVM method appear due to necessary technicalities (related to Lagrange multiplier method[11]) while in V-matrix method they appear as a result of incorporating existing prior knowledge about the solution: the classical setting of pattern recognition problem does not include such prior knowledge[12].

The discussion above indicates that, on one hand, the computational complexity of estimation of conditional probability is not higher than that of standard SVM classification, while, on the other hand, the V-estimate of conditional probability takes into account not only the information about the geometry of training data (incorporated in V-matrix) but also the existing prior knowledge about solution (incorporated in the constraints above).

This leads to the following question:

Can V-matrix method replace SVM method for pattern recognition?

The answer to this question is not obvious. In the mid-1990s, the following Imperative was formulated (Vapnik, 1995), (Vapnik, 1998):

> *While solving problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you need, but not a more general one. It is quite possible that you have enough information to solve a particular problem of interest well, but not enough information to solve a general problem.*

10. Often one has stronger constraints

$$\mathbf{a}_\ell \le \Phi \le \mathbf{b}_\ell,$$

where $\mathbf{0}_\ell \le \mathbf{a}_\ell$ and $\mathbf{b}_\ell \le \mathbf{1}_\ell$ are given (by experts) as additional prior information.

11. The Lagrange multiplier method was developed to find the solution in the *dual optimization space* and constraints in SVM method are related to Lagrange multipliers. Computationally, it is much easier to obtain the solution in the dual space given by (73), (74), (75) than in the *primal space* given by (71), (72). As shown by comparisons (Osuna and Girosi, 1999) of SVM solutions in primal and dual settings, (1) solution in primal space is more difficult computationally, (2) the obtained accuracies in both primal and dual spaces are about the same, (3) the primal space solution uses significantly fewer support vectors, and (4) the large number of support vectors in dual space solution is caused by the need to maintain the constraints for Lagrange multipliers.

12. The only information in SVM about the solution are the constraints $y_i f(x_i, \alpha) \ge 1 - \xi_i$, where $\xi_i \ge 0$ are (unknown) slack variables (Vapnik and Izmailov, 2015). However, this information does not contain any prior knowledge about the function f.

Solving (ill-posed) conditional probability problem instead of pattern recognition problem might appear to contradict this Imperative. However, while estimating conditional probability, one uses prior knowledge about the solution, and applies rigorous approaches, whereas the SVM setting does not take that knowledge into account and is based, instead, on justified heuristic approach of large margin. Since these two approaches leverage different factors and thus cannot be compared theoretically, it is important to compare them empirically.

9.3 Experimental Comparison of I-Matrix (L_2 SVM) and V-matrix Methods

In this section, we compare the L_2-SVM based method with V-matrix based method for estimation of one-dimensional conditional probability functions. Let the data be generated by an unknown probability density function $p(x, y) = p(y|x)p(x)$, where $x \in X,\ y \in \{0, 1\}$. Then the regression function $f_0(x)$ coincides with the conditional probability function $p(y = 1|x)$, so the problem of estimating the conditional probability in the set $\{f(x, \alpha)\}, \alpha \in \Lambda$ is equivalent to the problem of estimating the regression function on the data

$$(x_1, y_1), ..., (x_\ell, y_\ell).$$

We use L_2-SVM method for the estimation of the non-linear regression function in the set $\{f(x, \alpha)\}, \alpha \in \Lambda$ belonging to RKHS.

According to this method, in order to estimate the regression in the set of RKHS associated with the kernel $K(x_i, x)$, one has to find the parameters α_i of the function

$$f(x, \alpha) = \sum_{i=1}^{\ell} \alpha_i K(x_i, x) + \alpha_0$$

that minimize the functional

$$(Y - K\Lambda - \alpha_0 1)^T (Y - K\Lambda - \alpha_0 1) + \gamma \Lambda^T K \Lambda, \tag{90}$$

where we have denoted $Y = (y_1, ..., y_\ell)^T$, $\Lambda = (\alpha_1, ..., \alpha_\ell)^T$, by K is the matrix of elements $K(x_i, x_j),\ i, j = 1, ..., \ell$ and 1 is the ℓ-dimensional vector of ones.

Additionally, we take into account that (since regression coincides with conditional probability) the desired function satisfies $(\ell + 1)$ constraints: one constraint of equality type

$$\frac{1}{\ell} \sum_{i,j=1}^{\ell} \alpha_i K(x_i, x_j) + \alpha_0 = \frac{1}{\ell} \sum_{i=1}^{\ell} y_i, \tag{91}$$

and ℓ constraints of inequality type

$$0 \le \sum_{i=1}^{\ell} \alpha_i K(x_i, x_j) + \alpha_0 \le 1, \quad j = 1, ..., \ell, \tag{92}$$

forming L_2-SVM based method of conditional probability estimation.

The V-matrix based method of conditional probability estimation minimizes the functional

$$(Y - K\Lambda - \alpha_0 1)^T V (Y - K\Lambda - \alpha_0 1) + \gamma \Lambda^T K \Lambda \tag{93}$$

subject to the same constraints.

Therefore, the L_2-SVM method differs from V-matrix method by using identity matrix I instead of V matrix. Further, we call these method as I-matrix and V-matrix methods[13] In this Section, we present results of experimental comparisons of I-matrix and V-matrix methods. In our comparison, we consider two (one-dimensional) examples: estimating monotonic[14] and non-monotonic functions. In our experiments, we use the same kernel, namely, INK-spline of order 0:

$$K(x_i, x_j) = \min(x, x_i).$$

We can apply three versions of the solution for this problem:

1. Solutions that are defined by closed forms (ignoring the prior knowledge about the problem). These solutions are fast to obtain, without any significant computational problems.

2. Solutions that minimize the corresponding functionals while taking into account only the constraint of equality type.

   These solutions are also fast, without any significant computational problems. In this case one has to minimize functionals (90) and (93) choosing such α_0 for which equality constraints (91) holds true (Kuhn-Tucker condition)

3. Solutions that minimize functionals (90), (93) subject to all $\ell + 1$ constraints.

   These solutions require applying a full-scale quadratic optimization procedure. For large values of ℓ, it is not as simple computationally as previous two versions.

For our examples, all three solutions gave reasonably close results. Below we only report the results of the last one, the QP-solution.

Our first goal was to estimate the effect of using V-matrix (and compare it to I-matrix). To do this, we had to exclude the influence of the choice of regularization parameter γ. We did this by using two one-dimensional problems of estimating conditional probability functions: (1) monotonic function (Figure 1) and (2) non-monotonic one (Figure 2). For each problem, we generated 10,000 test examples and selected the best the possible (for the given training set) value of parameter γ. Figure 1 and Figure 2 present the result of approximation of conditional probability function for training sets of different sizes (48, 96,

13. If one ignores the constraints, both methods (I-matrix method the V-matrix method) have closed form solutions. The solutions are (for I-matrix method $V = I$)

$$A = \left(WK + \frac{\gamma}{2} I\right)^{-1} WY,$$

where

$$W = V - c^{-1}(V\mathbf{1})(\mathbf{1}^T V), \quad c = \mathbf{1}^T V \mathbf{1}.$$
$$\alpha_0 = c^{-1}\mathbf{1}^T V\,(Y - KA)\,.$$

14. Estimation of monotonic conditional probability function is important for pattern recognition problem since the VC dimension of the set of monotonically increasing (decreasing) functions equal to one independently of dimensionality.

192, 384) using the best γ for I-matrix method (left column) and V-matrix method (right column). In the figures, blue color corresponds to the true condition probability function, while black color corresponds to its approximations; red and green points in the horizontal axis correspond to two classes of the training set. In the Figures, we also show deviations of the approximations from the true conditional probability functions in both $L_1(\mu)$ and $L_2(\mu)$ metrics. In all our experiments we used the equal number of representatives of both classes.

These comparisons show that in all cases V-matrix method delivers better solution.

Subsequently, we compared V-matrix and I-matrix methods when the parameter γ is selected using the cross-validation technique on training data (6-fold cross validation based on maximum likelihood criterion): Figure 3 and Figure 4. Here also V-matrix method performs better than I-matrix method. The more training data is used, the larger is the advantage of the V-matrix method.

It is especially important that, in all our experiments, V-matrix method produced more smooth approximations to the true function than I-matrix method did. This is due to incorporation of the geometry of the training data into the solution.

Acknowledgments

This material is based upon work partially supported by AFRL and DARPA under contract FA8750-14-C-0008. Any opinions, findings and / or conclusions in this material are those of the authors and do not necessarily reflect the views of AFRL and DARPA.

We thank Professor Cherkassky, Professor Gammerman, and Professor Vovk for their helpful comments on this paper.

Appendix A. Appendix: V-Matrix for Statistical Inference

In this section, we describe some details of statistical inference algorithms using V-matrix. First, consider algorithms for conditional probability function $P(y|x)$ estimation and regression function $f(x)$ estimation given iid data

$$(y_1, X_1), ..., (y_\ell, X_\ell) \tag{94}$$

generated according to $p(x, y) = p(y|x)p(x)$. In (94), $y \in \{0, 1\}$ for the problem of conditional probability estimation, and $y \in R^1$ for the problems of regression estimation and density ratio estimation. Our V-matrix algorithm consists of the following simple steps.

A.1 Algorithms for Conditional Probability and Regression Estimation

Step 1. Find the domain of function. Consider vectors

$$X_1, ..., X_\ell \tag{95}$$

from training data. By a linear transformation in space $\mathcal{X}$, this data can be embedded into the smallest rectangular box with its edges parallel to coordinate axes. Without loss of generality, we also chose the origin of coordinate y such that all $y_i \in [0, \infty],\ i = 1, ..., \ell$ are non-negative.

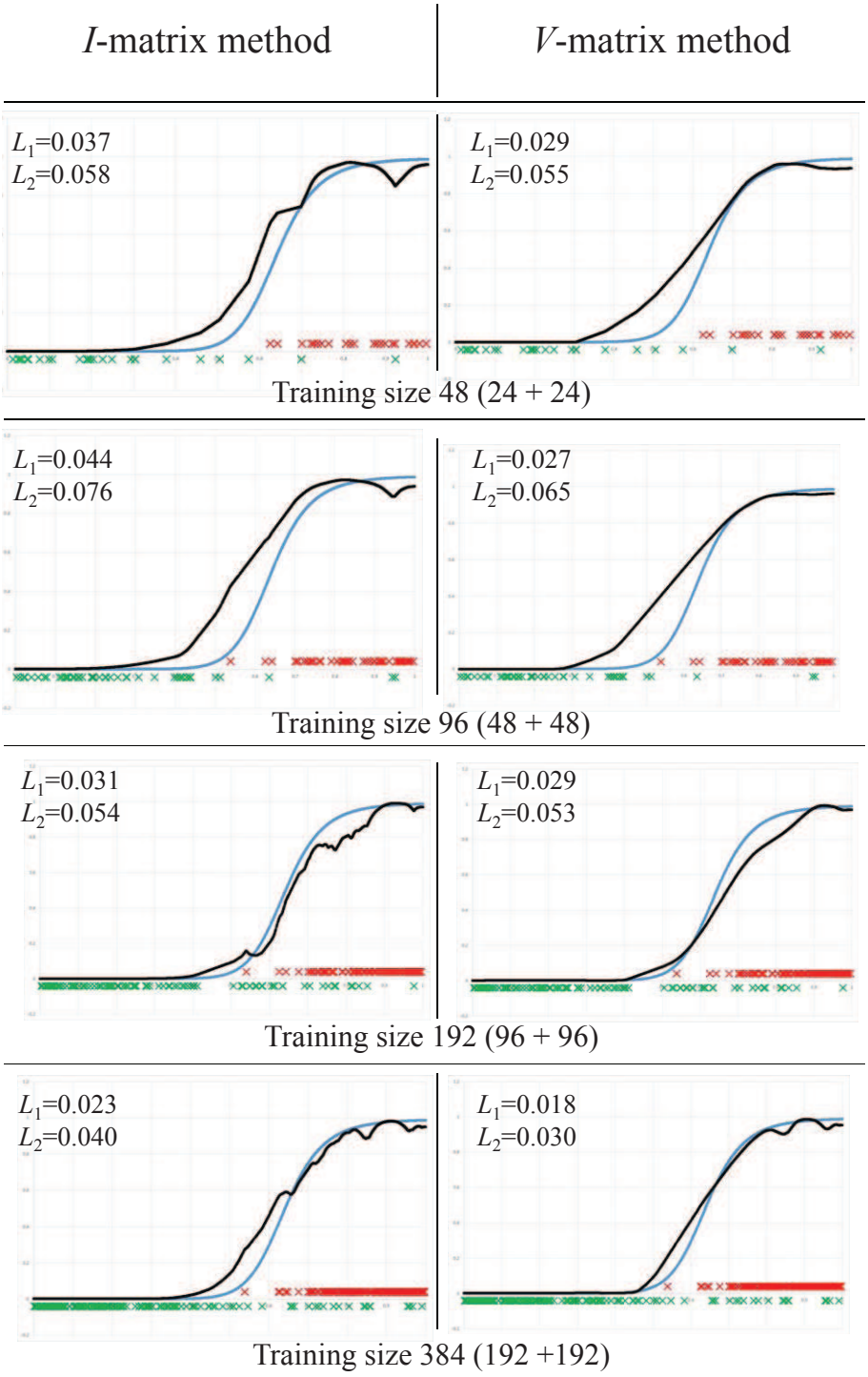


Figure 1: Comparison of I-matrix and V-matrix methods where regularization parameters γ were selected on validation set of size 10,000.

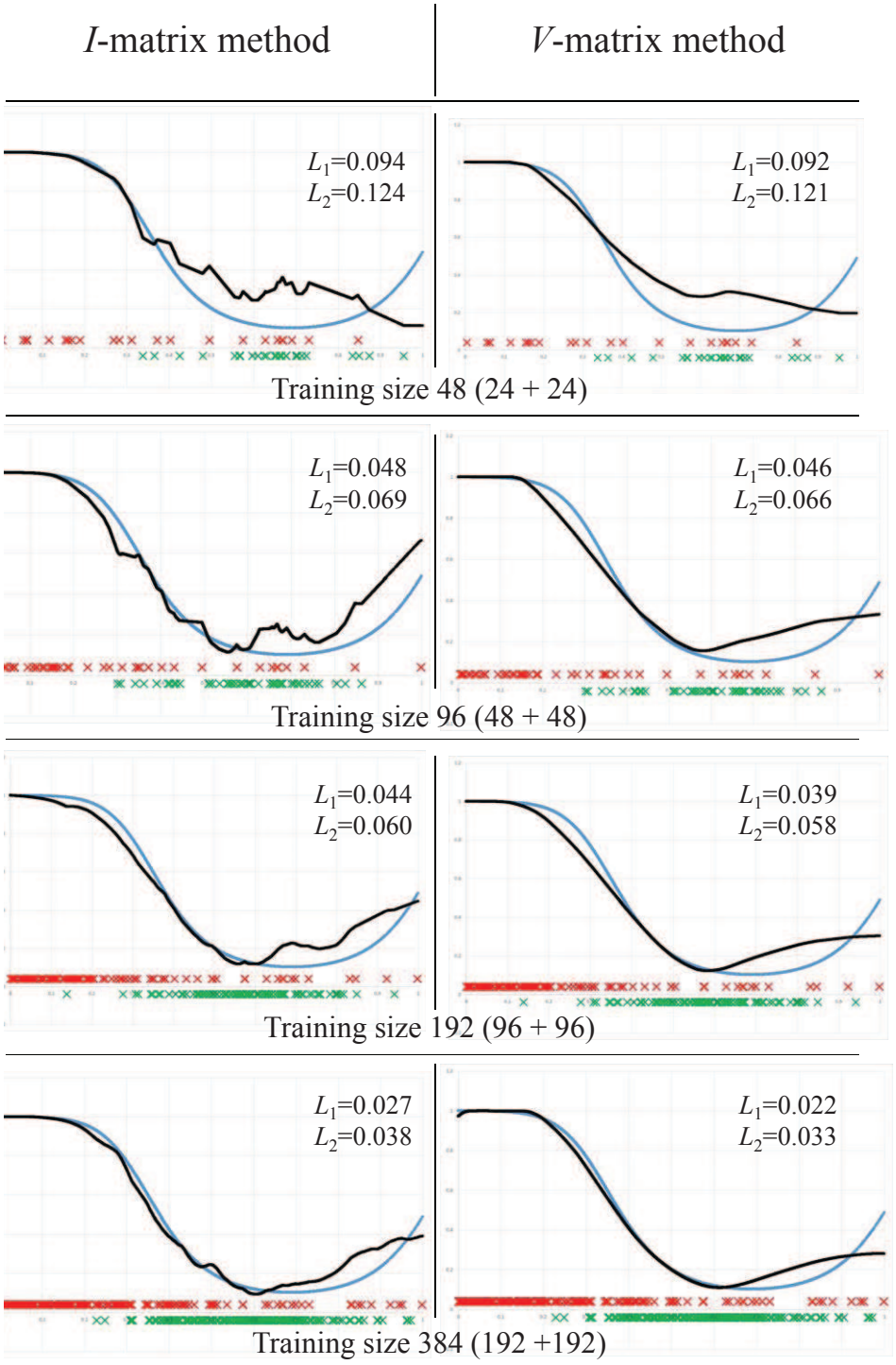


Figure 2: Comparison of I-matrix and V-matrix methods where regularization parameters γ were selected on validation set of size 10,000.

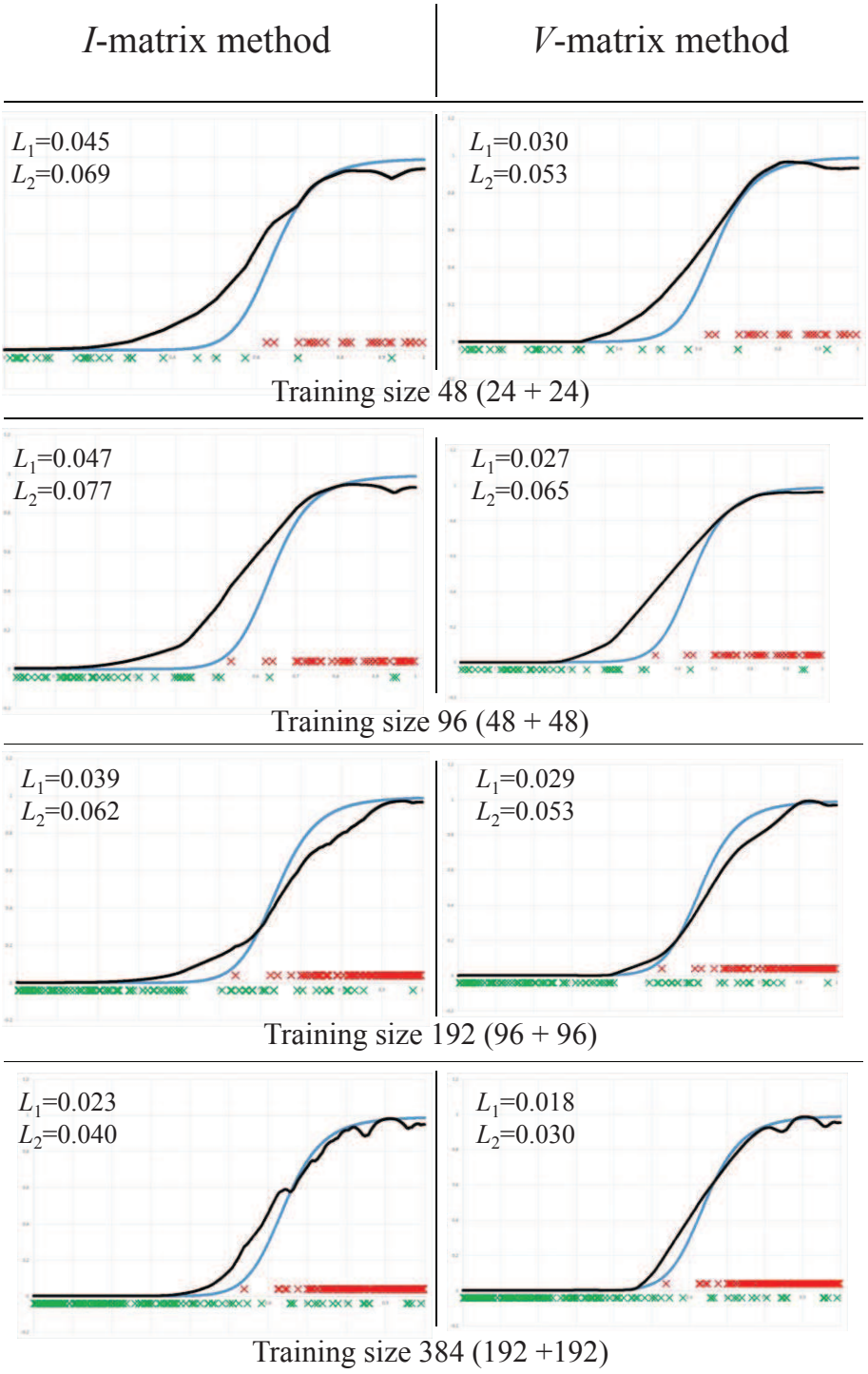


Figure 3: Comparison of I-matrix and V-matrix methods where regularization parameters γ were selected by cross-validation on training set.

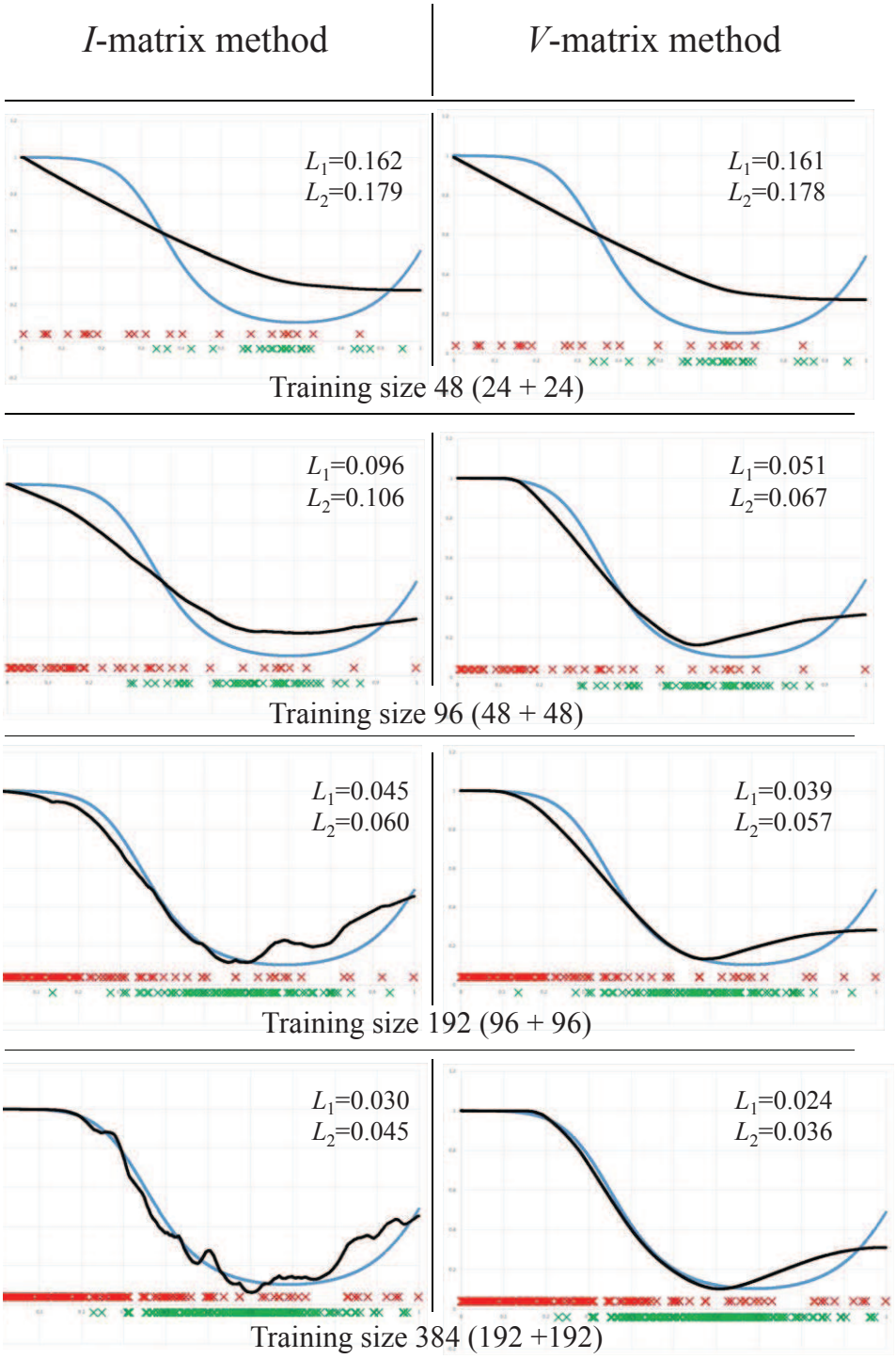


Figure 4: Comparison of I-matrix and V-matrix methods where regularization parameters γ were selected by cross-validation on training set.

Further we assume that data (95) had been preprocessed in this way.

Step 2. Find the functions $\mu(x^k)$. Using preprocessed data (95), construct for any coordinate x^k of the vector x the piecewise constant function

$$\mu_k(x) = \frac{1}{\ell}\sum_{i=1}^{\ell}\theta(x^k - X_i^k).$$

Step 3. Find functions $\sigma(x^k)$. For any coordinate of $k = 1, ..., d$ find the following:

1. The value
$$\hat{y}_{av} = \frac{1}{\ell}\sum_{i=1}^{\ell} y_i$$
(for pattern recognition problem, $\hat{y}_{av} = p_\ell$ is the fraction of training samples from class $y = 1$).

2. The piecewise constant function
$$F_*(x^k) = \frac{1}{\ell\hat{y}_{av}}\sum_{i=1}^{\ell} y_i\theta(x - X_i)$$
(For pattern recognition problem, function $F_*(x^k) = P(x^k|y = 1)$ estimates cumulative distribution function of x^k for samples from class $y = 1$).

3. The piecewise constant function
$$\sigma^k(x) = \Big(F_*(x^k)(1 - F_*(x^k)) + \varepsilon\Big)^{-1}.$$

Step 4. Find elements of V-matrix. Calculate the values

$$V_{ij}^k = \int \theta(x^k - X_i^k)\theta(x^k - X_j^k)\sigma(x^k)d\mu(x^k) = \int_{\max\{X_i^k, X_j^k\}}^{\infty} \sigma(x^k)d\mu(x^k).$$

Since both $\sigma(x^k)$ and $\mu(x^k)$ are piecewise constant functions, the last integral is a sum of constants.

Step 5. Find V-matrix. Compute elements of V-matrix as

$$V_{ij} = \prod_{k=1}^{d} V_{ij}^k.$$

Remark 1. Since V-matrix in the problems of conditional probability and regression estimation is scale-invariant, one can multiply all elements of this matrix by a fixed constant in order to keep the values of matrix elements within reasonable bounds for subsequent computations.

Remark 2. Any diagonal element V_{tt}^k is not less than elements of the corresponding row V_{tj}^k and column V_{jt}^k. Therefore, in order to compute V-matrix in multi-dimensional

case, it is reasonable to compute its diagonal elements first and, if they are small, just to replace the entries in the corresponding row and column with zeros.

It is possible (especially for large d) that V-matrix can have dominating diagonal elements. In this case, V-matrix can be approximated by a diagonal matrix. This is equivalent to the weighted least square method where weights are defined by the diagonal values V_{tt}.

Step 6. Find the values of conditional probability or the values of regression at the points of observation. Solve the quadratic optimization problem defined in the corresponding sections (in Section 6.4).

Step 7. Find the conditional probability or regression function. Solve interpolation problem defined in Section 6.4.

A.2 Algorithms for Density Ratio Estimation

For the problem of density ratio estimation, the algorithm requires the following modifications:

Step 1a. Find the domain of function. Domain of function is defined using data

$$X_1, ..., X_{\ell_{\text{den}}}, X'_1, ..., X'_{\ell_{\text{num}}}, \tag{96}$$

where training vectors X_i and X'_j are distributed according to $F_{\text{den}}(x)$ and $F_{\text{num}}(x')$, respectively.

Step 2a. Find the functions $\mu(x^k)$. Using (preprocessed) data (96), construct for coordinate $x^k,\ k = 1, ..., d$ of vector x the piecewise constant function

$$\mu_k(x) = \frac{1}{(\ell_{\text{den}} + \ell_{\text{num}})} \left(\sum_{i=1}^{\ell_{\text{den}}} \theta(x^k - X_i^k) + \sum_{i=1}^{\ell_{\text{num}}} \theta(x^k - X_i'^k) \right).$$

Step 3a. Find functions $\sigma(x^k)$. For any coordinate $x^k,\ k = 1, ..., d$ find:

– the piecewise constant function

$$F_{**}(x^k) = \frac{1}{\ell_{\text{num}}} \sum_{j=1}^{\ell_{\text{num}}} \theta(x - X'_j);$$

– the piecewise constant function

$$\sigma(x^k) = \left(F_{**}(x^k)(1 - F_{**}(x^k)) + \varepsilon \right)^{-1},$$

where $\varepsilon > 0$ is a small value.

Step 4a. Find the V-matrix and V^*-matrix. Estimate the matrices using expressions from corresponding sections.

Step 5a. Find the values of density ratio function at the points of observation. Solve the quadratic optimization problem defined in corresponding sections.

Step 6a. Find the density ratio function. Solve the interpolation problem defined in Section 6.4 (if estimated values of density ratio in ℓ_{den} points are not sufficient for the application, and the function itself has to be estimated).

A.3 Choice of Regularization Parameter

The value of regularization parameter γ can be selected using standard cross-validation techniques.

For conditional probability estimation, one can look for maximization of likelihood rather than for minimization of error rate. This leads to a more accurate estimate of conditional probability function.

References

L. Bottou and V. Vapnik. Local learning algorithms. *Neural Computation*, 4(6):888–900, 1992.

G. Brown, A. Pocock, M. Zhao, and M. Luján. Conditional likelihood maximisation: A unifying framework for information theoretic feature selection. *J. Mach. Learn. Res.*, 13: 27–66, January 2012.

T. Cover and J. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2006.

M. du Plessis and M. Sugiyama. Semi-supervised learning of class balance under class-prior change by distribution matching. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, pages 823–830, 2012.

Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In H. Park, S. Parthasarathy, H. Liu, and Z. Obradovic, editors, *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)*, pages 389–400, Sparks, Nevada, USA, Apr. 30–May 2 2009.

G. Kimeldorf and G. Wahba. A correspondence between bayesian estimation on stochastic processes and smoothing by splines. *The Annals of Mathematical Statistics*, pages 495–502, 1970.

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

E. Osuna and F. Girosi. Reducing the run-time complexity in support vector machines. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods*, pages 271–283. MIT Press, Cambridge, MA, USA, 1999.

C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 515–521, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1-55860-556-8.

B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, COLT '01/EuroCOLT '01, pages 416–426, London, UK, UK, 2001. Springer-Verlag.

A. Stefanyuk. Estimation of the likelihood ratio function in the "disorder" problem of random processes. *Automation and Remote Control*, (9):53–59, 1986.

M. Sugiyama, T. Suzuki, and T. Kanamori. *Density Ratio Estimation in Machine Learning*. Cambridge University Press, Cambridge, UK, 2012.

J. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300, June 1999.

A. Tikhonov and V. Arsenin. *Solutions of Ill-Posed Problems*. W.H. Winston, 1977.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

V. Vapnik and L. Bottou. Local algorithms for pattern recognition and dependencies estimation. *Neural Computation*, 5(6):893–909, 1993.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition (in Russian)*. Nauka, Moscow, 1974.

V. Vapnik and R. Izmailov. Learning with intelligent teacher: Similarity control and knowledge transfer. In A. Gammerman, V. Vovk, and H. Papadopoulos, editors, *Statistical Learning and Data Sciences*, volume 9047 of *Lecture Notes in Computer Science*, pages 3–32. Springer International Publishing, 2015.

Journal of Machine Learning Research 16 (2015) 1731-1755 Submitted 6/15; Published 9/15

Batch Learning from Logged Bandit Feedback through Counterfactual Risk Minimization

Adith Swaminathan ADITH@CS.CORNELL.EDU
Thorsten Joachims TJ@CS.CORNELL.EDU
Department of Computer Science
Cornell University
Ithaca, NY 14853, USA

Editor: Alex Gammerman and Vladimir Vovk

Abstract

We develop a learning principle and an efficient algorithm for batch learning from logged bandit feedback. This learning setting is ubiquitous in online systems (e.g., ad placement, web search, recommendation), where an algorithm makes a prediction (e.g., ad ranking) for a given input (e.g., query) and observes bandit feedback (e.g., user clicks on presented ads). We first address the counterfactual nature of the learning problem (Bottou et al., 2013) through propensity scoring. Next, we prove generalization error bounds that account for the variance of the propensity-weighted empirical risk estimator. In analogy to the Structural Risk Minimization principle of Wapnik and Tscherwonenkis (1979), these constructive bounds give rise to the Counterfactual Risk Minimization (CRM) principle. We show how CRM can be used to derive a new learning method—called Policy Optimizer for Exponential Models (POEM)—for learning stochastic linear rules for structured output prediction. We present a decomposition of the POEM objective that enables efficient stochastic gradient optimization. The effectiveness and efficiency of POEM is evaluated on several simulated multi-label classification problems, as well as on a real-world information retrieval problem. The empirical results show that the CRM objective implemented in POEM provides improved robustness and generalization performance compared to the state-of-the-art.

Keywords: empirical risk minimization, bandit feedback, importance sampling, propensity score matching, structured prediction

1. Introduction

Log data is one of the most ubiquitous forms of data available, as it can be recorded from a variety of systems (e.g., search engines, recommender systems, ad placement) at little cost. The interaction logs of such systems typically contain a record of the input to the system (e.g., features describing the user), the prediction made by the system (e.g., a recommended list of news articles) and the feedback (e.g., number of ranked articles the user read) (Li et al., 2010). The feedback, however, provides only partial information— "bandit feedback"— limited to the particular prediction shown by the system. The feedback for all the other predictions the system could have made is typically not known. This makes learning from log data fundamentally different from supervised learning, where "correct"

predictions (e.g., the best ranking of news articles for that user) together with a loss function provide full-information feedback.

In this paper, we address the problem of learning from logged bandit feedback. Unlike online learning with bandit feedback, batch learning with bandit feedback does not require interactive experimental control over the system. Furthermore, it enables the reuse of existing data and offline cross-validation techniques for model selection (e.g., "which features to use?", "which learning algorithm to use?", etc.).

To design algorithms for batch learning from bandit feedback, *counterfactual* estimators (Bottou et al., 2013) of a system's performance can be used to estimate how other systems would have performed if they had been in control of choosing predictions. Such estimators have been developed recently for the off-policy evaluation problem (Dudík et al., 2011; Li et al., 2011, 2014), where data collected from the interaction logs of one bandit algorithm is used to evaluate another system.

Our approach to counterfactual learning centers around the insight that, to perform robust learning, it is not sufficient to have just an unbiased estimator of the off-policy system's performance. We must also reason about how the variances of these estimators differ across the hypothesis space, and pick the hypothesis that has the best possible guarantee (tightest conservative bound) for its performance. We first prove generalization error bounds for a *stochastic hypothesis* family using an empirical Bernstein argument (Maurer and Pontil, 2009). This builds on recent approaches to deriving confidence intervals for counterfactual estimators (Bottou et al., 2013; Thomas et al., 2015). By relating the generalization error to the empirical sample variance of different hypotheses, we can effectively penalize the hypotheses with large variance during training using a data-dependant regularizer. In analogy to Structural Risk Minimization for full-information feedback (Wapnik and Tscherwonenkis, 1979), the constructive nature of these bounds suggests a general principle—Counterfactual Risk Minimization (CRM)—for designing methods for batch learning from bandit feedback.

Using the CRM principle, we derive a new learning algorithm—Policy Optimizer for Exponential Models (POEM)—for structured output prediction. The training objective is decomposed using repeated variance linearization, and optimizing it using AdaGrad (Duchi et al., 2011) yields a fast and effective algorithm. We evaluate POEM on several multi-label classification problems, verify that its empirical performance supports the theory, and demonstrates substantial gain in generalization performance over the state-of-the-art.

This paper is an extended version of Swaminathan and Joachims (2015), adding the following contributions. First, it provides the proof of the main generalization error bound upon which the CRM principle is based. Second, it derives and details the Iterated Variance Majorization Algorithm for training POEM, which was only sketched in Swaminathan and Joachims (2015). Third, the paper provides a first real-world experiment using POEM for learning a high precision classifier for information retrieval using logged click data.

The remainder of this paper is structured as follows. We review existing approaches in Section 2. The learning setting is detailed in Section 3, and contrasted with supervised learning. In Section 4, we derive the Counterfactual Risk Minimization learning principle and provide a rule of thumb for setting hyper-parameters. In Section 5, we instantiate the CRM principle for structured output prediction using exponential models and construct an efficient decomposition of the objective for stochastic optimization. Empirical evaluations

are reported in Section 6 and a real-world application is described in Section 7. We conclude with future directions and discussion in Section 8.

2. Related Work

Existing approaches for batch learning from logged bandit feedback fall into two categories. The first approach is to reduce the problem to supervised learning. In principle, since the logs give us an incomplete view of the feedback for different predictions, one could first use regression to estimate a feedback oracle for unseen predictions, and then use any supervised learning algorithm using this feedback oracle. Such a two-stage approach is known to not generalize well (Beygelzimer and Langford, 2009). More sophisticated techniques using the Offset Tree algorithm (Beygelzimer and Langford, 2009) allow us to perform batch learning when the space of possible predictions is small. In contrast, our approach generalizes structured output prediction, with exponential-sized prediction spaces. In particular, we apply our approach to multilabel classification problems. When the number of labels is K, the number of possible predictions is 2^K. A direct application of the Offset tree algorithm requires $\mathcal{O}(2^K)$ space and only guarantees regret $\mathcal{O}((2^K - 1)r)$ where r is the regret of the underlying binary classifier. Our approach directly tackles the problem using popular models from structured prediction instead, using computation and space complexity that mimics supervised approaches to the problem.

The second approach to batch learning from bandit feedback uses propensity scoring (Rosenbaum and Rubin, 1983) to derive unbiased estimators from the interaction logs (Bottou et al., 2013). These estimators are used for a small set of candidate policies, and the best estimated candidate is picked via exhaustive search. In contrast, our approach can be optimized via gradient descent, over hypothesis families (of infinite size) that are equally as expressive as those used in supervised learning. In particular, we build on recent work that develops confidence bounds for counterfactual estimators (Bottou et al., 2013; Thomas et al., 2015) using empirical Bernstein bounds. Our key insight is that these confidence intervals are not merely observable but can be efficiently optimized during training. Other recent bounds derived from analyzing Renyi divergences (Cortes et al., 2010) can analogously be co-opted in our approach to counterfactual learning.

Our approach builds on counterfactual estimators that have been developed for off-policy evaluation. The inverse propensity scoring approach can work well when we have a good model of the historical algorithm (Strehl et al., 2010; Li et al., 2014, 2015), and doubly robust estimators (Dudík et al., 2011) are even more effective when we additionally have a good model of the feedback. In our work, we focus on the inverse propensity scoring estimator, but the results we derive hold equally for the doubly robust estimators.

In the current work, we concentrate on the case where the historical algorithm was a stationary, stochastic policy. Techniques like exploration scavenging (Langford et al., 2008) and bootstrapping (Mary et al., 2014) allow us to perform counterfactual evaluation even when the historical algorithm was deterministic or adaptive.

Our strategy of picking the hypothesis with the tightest conservative bound on performance mimics similar successful approaches in other problems like supervised learning (Wapnik and Tscherwonenkis, 1979), risk averse multi-armed bandits (Galichet et al., 2013), regret minimizing contextual bandits (Langford and Zhang, 2008) and reinforcement learn-

ing (Garcia and Fernandez, 2012). Beyond the problem of batch learning from bandit feedback, our approach can have implications for several applications that require learning from logged bandit feedback data: warm-starting multi-armed bandits (Shivaswamy and Joachims, 2012) and contextual bandits (Strehl et al., 2010), pre-selecting retrieval functions for search engines (Hofmann et al., 2013), policy evaluation for contextual bandits (Li et al., 2011), and reinforcement learning (Thomas et al., 2015) to name a few.

3. Learning Setting: Batch Learning with Logged Bandit Feedback

Consider a structured output prediction problem that takes as input $x \in \mathcal{X}$ and outputs a prediction $y \in \mathcal{Y}$. For example, in multi-label document classification, x could be a news article and y a bit vector indicating the labels assigned to this article. The inputs are assumed drawn from a fixed but unknown distribution $\Pr(\mathcal{X})$, $x \overset{i.i.d.}{\sim} \Pr(\mathcal{X})$. Consider a hypothesis space $\mathcal{H}$ of *stochastic policies*. A hypothesis $h(\mathcal{Y} \mid x) \in \mathcal{H}$ defines a probability distribution over the output space $\mathcal{Y}$, and the hypothesis makes predictions by *sampling*, $y \sim h(\mathcal{Y} \mid x)$. Note that this definition also includes deterministic hypotheses, where the distributions assign probability 1 to a single y. For notational convenience, denote $h(\mathcal{Y} \mid x)$ by $h(x)$, and the probability assigned by $h(x)$ to y as $h(y \mid x)$. We will abuse notation slightly and use $(x, y) \sim h$ to refer to samples drawn from the joint distribution, $x \sim \Pr(\mathcal{X}), y \sim h(\mathcal{Y} \mid x)$. When it is clear from the context, we will drop $(x, y) \sim h$ and simply write h.

In interactive learning systems, we only observe feedback $\delta(x, y)$ for the y sampled from $h(x)$. In this work, feedback $\delta : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ is a cardinal loss that is only observed at the sampled data points. Small values for $\delta(x, y)$ indicate user satisfaction with y for x, while large values indicate dissatisfaction. The expected loss—called risk—of a hypothesis $R(h)$ is defined as

$$R(h) = \mathbb{E}_{x \sim \Pr(\mathcal{X})} \mathbb{E}_{y \sim h(x)} \left[\delta(x, y) \right] = \mathbb{E}_h \left[\delta(x, y) \right].$$

The goal of the system is to minimize risk, or equivalently, maximize expected user satisfaction. The aim of learning is to find a hypothesis $h \in \mathcal{H}$ that has minimum risk.

We wish to re-use the interaction logs of these systems for batch learning. Assume that its historical algorithm acted according to a *stationary* policy $h_0(x)$ (also called logging policy). The data collected from this system is

$$\mathcal{D} = \{(x_1, y_1, \delta_1), \ldots, (x_n, y_n, \delta_n)\},$$

where $y_i \sim h_0(x_i)$ and $\delta_i \equiv \delta(x_i, y_i)$.

Sampling bias. $\mathcal{D}$ cannot be used to estimate $R(h)$ for a new hypothesis h using the estimator typically used in supervised learning. We ideally need either full information about $\delta(x_i, \cdot)$ or need samples $y \sim h(x_i)$ to directly estimate $R(h)$. This explains why, in practice, model selection over a small set of candidate systems is typically done via A/B tests, where the candidates are deployed to collect new data sampled according to $y \sim h(x)$ for each hypothesis h. A relative comparison of the assumptions, hypotheses, and principles used in supervised learning vs. our learning setting is outlined in Table 1. Fundamentally, batch learning with bandit feedback is hard because $\mathcal{D}$ is both *biased* (predictions favored by the historical algorithm will be over-represented) and *incomplete* (feedback for other predictions will not be available) for learning.

	Supervised	Batch w/bandit
Distribution	$(x, y^*) \sim \Pr(\mathcal{X} \times \mathcal{Y})$	$x \sim \Pr(\mathcal{X}), y \sim h_0(x)$
Data $\mathcal{D}$	$\{x_i, y_i^*\}$	$\{x_i, y_i, \delta_i, p_i\}$
Hypothesis h	$y = h(x)$	$y \sim h(\mathcal{Y} \mid x)$
Loss	$\Delta(y^*, \cdot)$ known	$\delta(x, \cdot)$ unknown
Objective: argmin_h	$\hat{R}(h) + C \cdot Reg(\mathcal{H})$	$\hat{R}^M(h) + C \cdot Reg(\mathcal{H}) + \lambda \cdot \sqrt{\frac{\boldsymbol{Var}(h)}{n}}$

Table 1: Comparison of assumptions, hypotheses and learning principles for supervised learning and batch learning with bandit feedback.

4. Learning Principle: Counterfactual Risk Minimization

The distribution mismatch between h_0 and any hypothesis $h \in \mathcal{H}$ can be addressed using importance sampling, which corrects the sampling bias as

$$R(h) = \mathbb{E}_h\left[\delta(x, y)\right] = \mathbb{E}_{h_0}\left[\delta(x, y)\frac{h(y \mid x)}{h_0(y \mid x)}\right].$$

This motivates the propensity scoring approach of Rosenbaum and Rubin (1983). During the operation of the logging policy, we keep track of the propensity, $h_0(y \mid x)$ of the historical system to generate y for x. From these propensity-augmented logs

$$\mathcal{D} = \{(x_1, y_1, \delta_1, p_1), \ldots, (x_n, y_n, \delta_n, p_n)\},$$

where $p_i \equiv h_0(y_i \mid x_i)$, we can derive an unbiased estimate of $R(h)$ via Monte Carlo approximation,

$$\hat{R}(h) = \frac{1}{n}\sum_{i=1}^{n}\delta_i\frac{h(y_i \mid x_i)}{p_i}. \tag{1}$$

At first thought, one may think that directly estimating $\hat{R}(h)$ over $h \in \mathcal{H}$ and picking the empirical minimizer is a valid learning strategy. Unfortunately, there are several pitfalls.

First, this strategy is not invariant to additive transformations of the loss and will give degenerate results if the loss is not appropriately scaled. In Section 4.3, we develop intuition for why this is so, and derive the optimal scaling of δ. For now, assume that $\forall x, \forall y, \delta(x, y) \in [-1, 0]$.

Second, this estimator has unbounded variance, since $p_i \simeq 0$ in $\mathcal{D}$ can cause $\hat{R}(h)$ to be arbitrarily far away from the true risk $R(h)$. This can be fixed by "clipping" the importance sampling weights (Ionides, 2008; Strehl et al., 2010; Bottou et al., 2013; Cortes et al., 2010)

$$R^M(h) = \mathbb{E}_{h_0}\left[\delta(x, y)\min\left\{M, \frac{h(y \mid x)}{h_0(y \mid x)}\right\}\right],$$
$$\hat{R}^M(h) = \frac{1}{n}\sum_{i=1}^{n}\delta_i\min\left\{M, \frac{h(y_i \mid x_i)}{p_i}\right\}.$$

$M \geq 1$ is a hyper-parameter chosen to trade-off bias and variance in the estimate, where smaller values of M induce larger bias in the estimate. Optimizing $\hat{R}^M(h)$ through exhaustive enumeration over $\mathcal{H}$ yields the Inverse Propensity Scoring (IPS) training objective

$$\hat{h}^{IPS} = \operatorname*{argmin}_{h \in \mathcal{H}} \left\{ \hat{R}^M(h) \right\}. \tag{2}$$

This objective captures the essence of previous offline policy optimization approaches (Bottou et al., 2013; Strehl et al., 2010). These approaches differ from Equation (2) in the specific way the importance sampling weights are clipped, and frame the optimization problem as a maximization of counterfactual rewards as opposed to minimization of counterfactual risk.

Third, importance sampling typically estimates $\hat{R}^M(h)$ of different hypotheses $h \in \mathcal{H}$ with vastly different variances. Consider two hypotheses h_1 and h_2, where h_1 is similar to h_0, but where h_2 samples predictions that were not well explored by h_0. Importance sampling gives us low-variance estimates for $\hat{R}^M(h_1)$, but highly variable estimates for $\hat{R}^M(h_2)$. Intuitively, if we can develop variance-sensitive confidence bounds over the hypothesis space, optimizing a conservative confidence bound should find a h whose $R(h)$ will not be much worse, with high probability.

4.1 Generalization Error Bound

A standard analysis would give a bound that is agnostic to the variance introduced by importance sampling. Following our intuition above, we derive a higher order bound that includes the variance term using empirical Bernstein bounds (Maurer and Pontil, 2009). To develop such a generalization error bound, we first need a concept of capacity for stochastic hypothesis classes. Our strategy is to define an auxiliary deterministic function class $\mathcal{F}_{\mathcal{H}}$ for $\mathcal{H}$ and directly use covering numbers for $\mathcal{F}_{\mathcal{H}}$ conditioned on a sample $\mathcal{D}$. We start by defining the auxiliary deterministic function class $\mathcal{F}_{\mathcal{H}}$.

Definition 1 *For any stochastic class $\mathcal{H}$, define an auxiliary function class $\mathcal{F}_{\mathcal{H}} = \{f_h : \mathcal{X} \times \mathcal{Y} \mapsto [0,1]\}$. Each $h \in \mathcal{H}$ corresponds to a function $f_h \in \mathcal{F}_{\mathcal{H}}$,*

$$f_h(x,y) = 1 + \frac{\delta(x,y)}{M} \min\left\{ M, \frac{h(y \mid x)}{h_0(y \mid x)} \right\}. \tag{3}$$

Based on this auxiliary function class $\mathcal{F}_{\mathcal{H}}$, we will study the convergence of $\hat{R}^M(h) \rightarrow R^M(h)$. A key insight is the following relationship between h and f_h.

Lemma 2 *For any stochastic hypothesis h, the clipped risk $R^M(h)$ and the expected value of f_h under the data generating distribution are related as*

$$\mathbb{E}_{h_0}\left[f_h(x,y)\right] = 1 + \frac{R^M(h)}{M}. \tag{4}$$

Proof Note that f_h is a deterministic and bounded function. From the definition of f_h and by linearity of expectation,

$$\begin{aligned}\mathbb{E}_{h_0}\left[f_h(x,y)\right] &= \mathbb{E}_{h_0}\left[1+\frac{\delta(x,y)}{M}\min\left\{M,\frac{h(y\mid x)}{h_0(y\mid x)}\right\}\right]\\ &= 1+\frac{1}{M}\mathbb{E}_{h_0}\left[\delta(x,y)\min\left\{M,\frac{h(y\mid x)}{h_0(y\mid x)}\right\}\right]\\ &= 1+\frac{R^M(h)}{M}\end{aligned}$$

■

As a consequence of Lemma 2, we can use classic notions of capacity for $\mathcal{F}_\mathcal{H}$ to reason about the convergence of $\hat{R}^M(h) \to R^M(h)$. Recall the covering number $\mathcal{N}_\infty(\epsilon,\mathcal{F},n)$ for a function class $\mathcal{F}$.[1] Define an $\epsilon-$cover $\mathcal{N}(\epsilon,A,\|\cdot\|_\infty)$ for a set $A\subseteq\mathbb{R}^n$ to be the size of the smallest cardinality subset $A_0\subseteq A$ such that A is contained in the union of balls of radius ϵ centered at points in A_0, in the metric induced by $\|\cdot\|_\infty$. The covering number is,

$$\mathcal{N}_\infty(\epsilon,\mathcal{F},n) = \sup_{(x_i,y_i)\in(\mathcal{X}\times\mathcal{Y})^n}\mathcal{N}(\epsilon,\mathcal{F}(\{(x_i,y_i)\}),\|\cdot\|_\infty),$$

where $\mathcal{F}(\{(x_i,y_i)\})$ is the function class conditioned on sample $\{(x_i,y_i)\}$,

$$\mathcal{F}(\{(x_i,y_i)\}) = \{(f(x_1,y_1),\ldots,f(x_n,y_n)) : f\in\mathcal{F}\}.$$

Our measure for the capacity of our stochastic class $\mathcal{H}$ to "fit" a sample of size n shall be $\mathcal{N}_\infty(\frac{1}{n},\mathcal{F}_\mathcal{H},2n)$.

For a compact notation, define the random variable $u_h \equiv \delta(x,y)\min\left\{M,\frac{h(y|x)}{h_0(y|x)}\right\}$ with mean $\overline{u}_h = R^M(h)$. The sample $\mathcal{D}$ contains n i.i.d. random variables ${u_h}^i \equiv \delta_i\min\{M,\frac{h(y_i|x_i)}{p_i}\}$. Define the sample mean and variance of ${u_h}^i$

$$\hat{\overline{u}}_h \equiv \frac{1}{n}\sum_{i=1}^n {u_h}^i = \hat{R}^M(h),$$

$$\hat{\boldsymbol{Var}}(u_h) \equiv \frac{1}{n-1}\sum_{i=1}^n({u_h}^i - \hat{\overline{u}}_h)^2.$$

Theorem 3 *With probability at least $1-\gamma$ in the random vector $(x_1,y_1)\cdots(x_n,y_n) \overset{i.i.d.}{\sim} h_0$, with observed losses $\delta_1,\ldots,\delta_n$, for $n\geq 16$ and a stochastic hypothesis space $\mathcal{H}$ with capacity $\mathcal{N}_\infty(\frac{1}{n},\mathcal{F}_\mathcal{H},2n)$,*

$$\forall h\in\mathcal{H}:\quad R(h)\leq\hat{R}^M(h)+\sqrt{18\frac{\hat{\boldsymbol{Var}}(u_h)\mathcal{Q}_\mathcal{H}(n,\gamma)}{n}}+M\frac{15\mathcal{Q}_\mathcal{H}(n,\gamma)}{n-1},$$

$$\textit{where, } \mathcal{Q}_\mathcal{H}(n,\gamma)\equiv\log(\frac{10\cdot\mathcal{N}_\infty(\frac{1}{n},\mathcal{F}_\mathcal{H},2n)}{\gamma}),\quad 0<\gamma<1.$$

1. Refer Anthony and Bartlett (2009); Maurer and Pontil (2009) and the references therein for a comprehensive treatment of covering numbers.

Proof The proof follows from Theorem 6 of Maurer and Pontil (2009) applied to the deterministic function class $\mathcal{F}_{\mathcal{H}}$. We sketch the main argument using symmetrization and Rademacher variables here.

Define the random variable $s_h = 1 + \frac{u_h}{M}$ with mean $\mathbb{E}_{h_0}[s_h]$ and variance $\boldsymbol{Var}(s_h)$. Observe that $\mathbb{E}_{h_0}[s_h] = 1 + \frac{R^M(h)}{M}$ from Lemma 2. Let ${s_h}^i = 1 + \frac{{u_h}^i}{M}$. The sample $\mathcal{D}$ essentially contains n i.i.d. observations of s_h. Let $\hat{\bar{s}}_h$ and $\hat{\boldsymbol{Var}}(s_h)$ denote the empirical mean and variance of $\{{s_h}^i\}_{i=1}^n$ respectively. Observe that $\hat{\boldsymbol{Var}}(s_h) = \frac{\hat{\boldsymbol{Var}}(u_h)}{M^2}$. Abusing notation slightly, we will use boldface $\boldsymbol{s_h}$ to refer to the sample $\{{s_h}^i\}_{i=1}^n$.

We begin with Bennet's inequality.

For $s, \{s^i\}_{i=1}^n$ i.i.d. bounded random variables in $[0,1]$ having mean $\mathbb{E}[s]$ and variance $\boldsymbol{Var}(s)$, with probability at least $1-\gamma$ in $\{s^i\}_{i=1}^n \equiv \boldsymbol{s}$,

$$\mathbb{E}[s] - \hat{\bar{s}} \leq \sqrt{\frac{2\boldsymbol{Var}(s)\log 1/\gamma}{n}} + \frac{\log 1/\gamma}{3n}. \tag{5}$$

Intuitively, Bennet's inequality tells us that the estimate $\hat{\bar{s}}$ has lower accuracy if $\boldsymbol{Var}(s)$ is high, which exactly captures our intuition about the variance introduced by importance sampling when estimating the risk of a hypothesis "far" from h_0. However, the diameter of this confidence interval depends on the unobservable $\boldsymbol{Var}(s)$.

We recite Theorem 11 from Maurer and Pontil (2009) that gives a variance-sensitive bound with an observable confidence interval, which they call an Empirical Bernstein bound.

Under the same conditions as Bennet's inequality (5), let $n \geq 2, \hat{\boldsymbol{Var}}(s)$ represent the empirical variance of $\{s^i\}_{i=1}^n$. With probability at least $1-\gamma$,

$$\mathbb{E}[s] - \hat{\bar{s}} \leq \sqrt{\frac{2\hat{\boldsymbol{Var}}(s)\log 2/\gamma}{n}} + \frac{7\log 2/\gamma}{3(n-1)}. \tag{6}$$

This follows from confidence bounds on the sample standard deviation $\sqrt{\hat{\boldsymbol{Var}}(s)}$ compared to the true standard deviation $\mathbb{E}_{\boldsymbol{s}}\left[\sqrt{\hat{\boldsymbol{Var}}(s)}\right]$. Based on this bound, Maurer and Pontil (2009) define two Lipschitz continuous functions, $\Phi, \Psi : [0,1]^n \times \mathbb{R}_+ \to \mathbb{R}$.

$$\Phi(\boldsymbol{s}, t) = \hat{\bar{s}} + \sqrt{\frac{2\hat{\boldsymbol{Var}}(s)t}{n}} + \frac{7t}{3(n-1)}$$
$$\Psi(\boldsymbol{s}, t) = \hat{\bar{s}} + \sqrt{\frac{18\hat{\boldsymbol{Var}}(s)t}{n}} + \frac{11t}{n-1}.$$

These functions are Lipschitz continuous,

$$\Phi(\boldsymbol{s}, t) - \Phi(\boldsymbol{s'}, t) \leq (1 + 2\sqrt{\frac{t}{n}})\|\boldsymbol{s} - \boldsymbol{s'}\|_\infty$$
$$\Psi(\boldsymbol{s}, t) - \Psi(\boldsymbol{s'}, t) \leq (1 + 6\sqrt{\frac{t}{n}})\|\boldsymbol{s} - \boldsymbol{s'}\|_\infty. \tag{7}$$

The inequalities follow directly from $\sqrt{\hat{\boldsymbol{Var}}(s)} - \sqrt{\hat{\boldsymbol{Var}}(s')} \leq \sqrt{2}\|\boldsymbol{s} - \boldsymbol{s'}\|_\infty$.

For the symmetrization argument, consider two sets of n samples $\mathcal{D}$ and $\mathcal{D}'$ drawn from h_0 according to the conditions of Theorem 3 and used to estimate risk of a hypothesis h. This gives rise to two sets of n i.i.d. random variables $\boldsymbol{s}_h$ and $\boldsymbol{s}'_h$. Also define the Rademacher variables $\sigma_1, \dots \sigma_n \overset{i.i.d}{\sim} \mathcal{U}\{-1, 1\}$. Define $(\boldsymbol{\sigma}, \boldsymbol{s}_h, \boldsymbol{s}'_h)$ as the vector who's i^{th} co-ordinate is set to ${s_h}^i$ or ${s'_h}^i$ as specified by σ_i.

$$(\boldsymbol{\sigma}, \boldsymbol{s}_h, \boldsymbol{s}'_h)_i = \begin{cases} {s_h}^i & \text{if } \sigma_i = 1 \\ {s'_h}^i & \text{if } \sigma_i = -1. \end{cases}$$

For a fixed $h \in \mathcal{H}$ and a fixed double sample $\boldsymbol{s}_h, \boldsymbol{s}'_h$ as described above,

$$\Pr_{\sigma}\left[\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) \geq \Psi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t)\right] \leq 5e^{-t}. \tag{8}$$

This is simply a restatement of Lemma 14 from Maurer and Pontil (2009) and follows by decomposing the event $[\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) \geq \Psi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t)]$ as $[\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) \geq A] \wedge [A \geq \Psi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t)]$ where A uses the true mean and variance of s_h. The probability of the first event can be bounded using Bennet's inequality (5), while the second event can be bounded using the empirical Bernstein bound (6) and the confidence bounds on the sample standard deviation $\sqrt{\hat{\boldsymbol{Var}}(s)}$.

Set $t = \log \frac{2}{\gamma}$ and consider $t \geq \log 4$ (i.e. $\gamma \leq \frac{1}{2}$). Equation (6) implies, for any $h \in \mathcal{H}$,

$$\Pr(\Phi(\boldsymbol{s}_h, t) \geq \mathbb{E}[s_h]) \geq \frac{1}{2}. \tag{9}$$

Hence, for any $\rho > 0$,

$$\begin{aligned}
&\Pr_{\mathcal{D}}(\exists h \in \mathcal{H} : \mathbb{E}[s_h] > \Psi(\boldsymbol{s}_h, t) + \rho) = \mathbb{E}_{\mathcal{D}}\left[\sup_{h \in \mathcal{H}} \mathbb{I}\{\mathbb{E}[s_h] > \Psi(\boldsymbol{s}_h, t) + \rho\}\right] \\
&\leq \mathbb{E}_{\mathcal{D}}\left[\sup_{h \in \mathcal{H}} \mathbb{I}\{\mathbb{E}[s_h] > \Psi(\boldsymbol{s}_h, t) + \rho\}\right] 2\Pr(\Phi(\boldsymbol{s}'_h, t) \geq \mathbb{E}\left[s'_h\right]) && \text{Equation (9)} \\
&= 2\mathbb{E}_{\mathcal{D}}\left[\sup_{h \in \mathcal{H}} \mathbb{E}_{\mathcal{D}'}\left[\mathbb{I}\{\mathbb{E}[s_h] > \Psi(\boldsymbol{s}_h, t) + \rho \wedge \Phi(\boldsymbol{s}'_h, t) \geq \mathbb{E}[s_h]\}\right]\right] && \text{since } \mathbb{E}[s_h] = \mathbb{E}\left[s'_h\right] \\
&\leq 2\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathcal{D}'}\left[\sup_{h \in \mathcal{H}} \mathbb{I}\{\mathbb{E}[s_h] > \Psi(\boldsymbol{s}_h, t) + \rho \wedge \Phi(\boldsymbol{s}'_h, t) \geq \mathbb{E}[s_h]\}\right] \\
&\leq 2\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathcal{D}'}\left[\sup_{h \in \mathcal{H}} \mathbb{I}\{\Phi(\boldsymbol{s}'_h, t) > \Psi(\boldsymbol{s}_h, t) + \rho\}\right] \\
&= 2\mathbb{E}_{\sigma}\mathbb{E}_{\mathcal{D}}\mathbb{E}_{\mathcal{D}'}\left[\sup_{h \in \mathcal{H}} \mathbb{I}\{\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) + \rho\}\right] && \text{since } \boldsymbol{s}_h, \boldsymbol{s}'_h \text{ are i.i.d.} \\
&\leq 2 \sup_{\mathcal{D}, \mathcal{D}'} \mathbb{E}_{\sigma}\left[\sup_{h \in \mathcal{H}} \mathbb{I}\{\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) + \rho\}\right] \\
&= 2 \sup_{\mathcal{D}, \mathcal{D}'} \Pr_{\sigma}(\exists h \in \mathcal{H} : \Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s}'_h), t) + \rho).
\end{aligned}$$

For a fixed $\mathcal{D}, \mathcal{D}'$, consider the $\epsilon-$cover of $\mathcal{F}_{\mathcal{H}}$, ${\mathcal{F}_{\mathcal{H}}}^0$. Denote the set of stochastic policies that correspond to each $f_h \in {\mathcal{F}_{\mathcal{H}}}^0$ by $\mathcal{H}^0$. We know that $\left|\mathcal{H}^0\right| \leq \mathcal{N}_{\infty}(\epsilon, \mathcal{F}_{\mathcal{H}}, 2n)$ (by

definition of the covering number, and since there is a one-to-one mapping from h to f_h) and $\forall h \in \mathcal{H}, \quad \exists h' \in \mathcal{H}^0$ such that $\|\boldsymbol{s}_h - \boldsymbol{s}_{h'}\|_\infty \leq \epsilon$ and $\|\boldsymbol{s'}_h - \boldsymbol{s'}_{h'}\|_\infty \leq \epsilon$ (by definition of $\epsilon-$cover). Instantiate $\rho = \epsilon(2 + 8\sqrt{\frac{t}{n}})$ and suppose $\exists h \in \mathcal{H}$ such that $\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t) + \rho$. Since Φ and Ψ are Lipschitz continuous, as demonstrated in Equation (7), hence there must exist a $h' \in \mathcal{H}^0$ such that $\Phi((\sigma, \boldsymbol{s}_{h'}, \boldsymbol{s'}_{h'}), t) > \Psi((-\sigma, \boldsymbol{s}_{h'}, \boldsymbol{s'}_{h'}), t)$. Hence,

$$
\begin{aligned}
&\Pr_\sigma(\exists h \in \mathcal{H} : \Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t) + \epsilon(2 + 8\sqrt{\frac{t}{n}})) \\
&\leq \Pr_\sigma(\exists h \in \mathcal{H}^0 : \Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t)) \\
&\leq \sum_{h \in \mathcal{H}^0} \Pr_\sigma(\Phi((\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t) > \Psi((-\sigma, \boldsymbol{s}_h, \boldsymbol{s'}_h), t)) \\
&\leq 5e^{-t}\mathcal{N}_\infty(\epsilon, \mathcal{F}_\mathcal{H}, 2n) \qquad \text{Equation (8) .}
\end{aligned}
$$

In short,

$$
\Pr_\mathcal{D}(\exists h \in \mathcal{H} : \mathbb{E}[s_h] > \Psi(\boldsymbol{s}_h, t) + \epsilon(2 + 8\sqrt{\frac{t}{n}})) \leq 10e^{-t}\mathcal{N}_\infty(\epsilon, \mathcal{F}_\mathcal{H}, 2n).
$$

Setting $10e^{-t}\mathcal{N}_\infty(\epsilon, \mathcal{F}_\mathcal{H}, 2n) = \gamma$ we get $t_\gamma = \log \frac{10\mathcal{N}_\infty(\epsilon, \mathcal{F}_\mathcal{H}, 2n)}{\gamma} > 1$. Moreover, $\frac{2(t_\gamma+1)}{n} \leq \frac{2(t_\gamma+1)}{n-1} \leq \frac{4t_\gamma}{n-1}$ and for $n \geq 16$, $8\sqrt{\frac{t_\gamma}{n}} \leq 2t_\gamma$. Substituting $\epsilon = \frac{1}{n}$ and simplifying,

$$
\Pr_\mathcal{D}(\exists h \in \mathcal{H} : \mathbb{E}[s_h] > \hat{\bar{s}}_h + \sqrt{\frac{18\hat{\boldsymbol{Var}}(s_h)t_\gamma}{n}} + \frac{15t_\gamma}{n-1}) \leq \gamma.
$$

Finally, $\mathbb{E}[s_h] = 1 + \frac{R^M(h)}{M}, \hat{\bar{s}}_h = 1 + \frac{\hat{R}^M(h)}{M}$ and $\hat{\boldsymbol{Var}}(s_h) = \frac{\hat{\boldsymbol{Var}}(u_h)}{M^2}$. Since $\delta(\cdot, \cdot) \leq 0$, hence $R(h) \leq R^M(h)$. Putting it all together,

$$
\Pr_\mathcal{D}(\exists h \in \mathcal{H} : R(h) > \hat{R}^M(h) + \sqrt{\frac{18\hat{\boldsymbol{Var}}(u_h)t_\gamma}{n}} + \frac{15Mt_\gamma}{n-1}) \leq \gamma.
$$

■

4.2 CRM Principle

The generalization error bound from the previous section is constructive in the sense that it motivates a general principle for designing machine learning methods for batch learning from bandit feedback. In particular, a learning algorithm following this principle should jointly optimize the estimate $\hat{R}^M(h)$ as well as its empirical standard deviation, where the latter serves as a *data-dependent regularizer*.

$$
\hat{h}^{CRM} = \operatorname*{argmin}_{h \in \mathcal{H}} \left\{ \hat{R}^M(h) + \lambda\sqrt{\frac{\hat{\boldsymbol{Var}}(u_h)}{n}} \right\}. \tag{10}
$$

$M \geq 1$ and $\lambda \geq 0$ are regularization hyper-parameters. When $\lambda = 0$, we recover the Inverse Propensity Scoring objective of Equation (2). In analogy to Structural Risk Minimization (Wapnik and Tscherwonenkis, 1979), we call this principle *Counterfactual Risk Minimization*, since both pick the hypothesis with the tightest upper bound on the true risk $R(h)$.

4.3 Optimal Loss Scaling

When performing supervised learning with true labels y^* and a loss function $\Delta(y^*, \cdot)$, empirical risk minimization using the standard estimator is invariant to additive translation and multiplicative scaling of Δ. The risk estimators $\hat{R}(h)$ and $\hat{R}^M(h)$ in bandit learning, however, crucially require $\delta(\cdot, \cdot) \in [-1, 0]$.

Consider, for example, the case of $\delta(\cdot, \cdot) \geq 0$. The training objectives in Equation (2) (IPS) and Equation (10) (CRM) become degenerate! A hypothesis $h \in \mathcal{H}$ that completely avoids the sample $\mathcal{D}$ (i.e. $\forall i = 1, \ldots, n, h(y_i \mid x_i) = 0$) trivially achieves the best possible $\hat{R}^M(h)$ $(= 0)$ with 0 empirical variance. This degeneracy arises partially because when $\delta(\cdot, \cdot) \geq 0$, the objectives optimize a *lower* bound on $R(h)$, whereas what we need is an *upper* bound.

For any bounded loss $\delta(\cdot, \cdot) \in [\bigtriangledown, \bigtriangleup]$, we have, $\forall x$

$$\mathbb{E}_{y \sim h(x)}\left[\delta(x, y)\right] \leq \bigtriangleup + \mathbb{E}_{y \sim h_0(x)}\left[(\delta(x, y) - \bigtriangleup) \min\left\{M, \frac{h(y \mid x)}{h_0(y \mid x)}\right\}\right].$$

Since the optimization objectives in Equations (2),(10) are unaffected by a constant scale factor (e.g., $\bigtriangleup - \bigtriangledown$), we should transform $\delta \mapsto \delta'$ to derive a conservative training objective,

$$\delta' \equiv \{\delta - \bigtriangleup\}/\{\bigtriangleup - \bigtriangledown\}.$$

Such a transformation captures the following assumption: for an input $x \in \mathcal{D}$, if a new hypothesis $h \neq h_0$ samples an unexplored y not seen in $\mathcal{D}$, in the worst case it will incur a loss of $\bigtriangleup$. This is clearly a very conservative assumption, and we foresee future work that relaxes this using additional assumptions about $\delta(\cdot, \cdot)$ and $\mathcal{Y}$.

4.4 Selecting Hyper-Parameters

We propose selecting the hyper-parameters $M \geq 1$ and $\lambda \geq 0$ via cross validation. However, we must be careful not to set M too small or λ too big. The estimated risk $\hat{R}^M(h) \in [-M, 0]$, while the variance penalty $\sqrt{\frac{\hat{\boldsymbol{Var}}(u_h)}{n}} \in \left[0, \frac{M}{2\sqrt{n}}\right]$. If M is too small, all the importance sampling weights will be clipped and all hypotheses will have the same biased estimate of risk $M\hat{R}^M(h_0)$. Similarly, if $\lambda \gg 0$, a hypothesis $h \in \mathcal{H}$ that completely avoids $\mathcal{D}$ (i.e. $\forall i = 1, \ldots, n, h(y_i \mid x_i) = 0$) has $\hat{R}^M(h)$ $(= 0)$ with 0 empirical variance. So, it will achieve the best possible training objective of 0. As a rule of thumb, we can calibrate M and λ so that the estimator is unbiased and the objective is negative for some $h \in \mathcal{H}$. When $h_0 \in \mathcal{H}$, $M \simeq \max\{p_i\}/\min\{p_i\}$ and $\left\{\hat{R}^M(h_0) + \lambda\sqrt{\frac{\hat{\boldsymbol{Var}}(u_{h_0})}{n}}\right\} < 0$ are natural choices.

4.5 When is Counterfactual Learning Possible?

The bounds in Theorem 3 are with respect to the randomness in h_0. Known impossibility results for counterfactual evaluation using h_0 (Langford et al., 2008) also apply to counterfactual learning. In particular, if h_0 was deterministic, or even stochastic but without full support over $\mathcal{Y}$, it is easy to engineer examples involving the unexplored $y \in \mathcal{Y}$ that guarantee sub-optimal learning even as $|\mathcal{D}| \to \infty$. Similarly, lower bounds for learning under covariate shift (Cortes et al., 2010) also apply to counterfactual learning. Finally, a stochastic h_0 with heavier tails need not always allow more effective learning. From importance sampling theory (Owen, 2013), what really matters is how well h_0 explores the regions of $\mathcal{Y}$ with favorable losses.

5. Learning Algorithm: POEM

We now use the CRM principle to derive an efficient algorithm for structured output prediction using linear rules. Classic learning methods for structured output prediction based on full-information feedback, e.g. structured support vector machines (Tsochantaridis et al., 2004) and conditional random fields (Lafferty et al., 2001), predict using

$$h_w^{sup}(x) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}} \left\{ w \cdot \phi(x, y) \right\}, \tag{11}$$

where w is a $d-$dimensional weight vector, and $\phi(x, y)$ is a $d-$dimensional joint feature map. For example, in multi-label document classification, for a news article x and a possible assignment of labels y represented as a bit vector, $\phi(x, y)$ could simply be a concatenation $\overline{x} \otimes y$ of the bag-of-words features of the document ($\overline{x}$), one copy for each of the assigned labels in y. Several efficient inference algorithms have been developed to solve Equation (11).

The POEM algorithm that is derived in this section uses the same parameterization of the hypothesis space as in Equation (11). However, it considers the following expanded class of Stochastic Softmax Rules based on this parameterization, which contains the deterministic rule in Equation (11) as a limiting case.

5.1 Stochastic Softmax Rules

Consider the following stochastic family $\mathcal{H}_{lin}$, parametrized by w. A hypothesis $h_w(x) \in \mathcal{H}_{lin}$ samples y from the distribution

$$h_w(y \mid x) = \exp(w \cdot \phi(x, y))/\mathbb{Z}(x).$$

$\mathbb{Z}(x) = \sum_{y' \in \mathcal{Y}} \exp(w \cdot \phi(x, y'))$ is the partition function. This can be thought of as the "soft-max" variant of the "hard-max" rules from Equation (11). Additionally, for a *temperature* multiplier $\alpha > 1, w \mapsto \alpha w$ induces a more "peaked" distribution $h_{\alpha w}$ that preserves the modes of h_w, and intuitively is a "more deterministic" variant of h_w.

h_w lies in the exponential family of distributions, and has a simple gradient,

$$\nabla h_w(y \mid x) = h_w(y \mid x) \left\{ \phi(x, y) - \mathbb{E}_{y' \sim h_w(x)} \left[\phi(x, y') \right] \right\}.$$

5.2 POEM Training Objective

Consider a bandit structured-output data set $\mathcal{D} = \{(x_1, y_1, \delta_1, p_1), \ldots, (x_n, y_n, \delta_n, p_n)\}$. In multi-label document classification, this data could be collected from an interactive labeling system, where each y indicates the labels predicted by the system for a document x. The feedback $\delta(x, y)$ is how many labels (but not which ones) were correct. To perform learning, first we scale the losses as outlined in Section 4.3. Next, instantiating the CRM principle of Equation (10) for $\mathcal{H}_{lin}$, (using notation analogous to that in Theorem 3, adapted for $\mathcal{H}_{lin}$), yields the POEM training objective:

$$w^* = \operatorname*{argmin}_{w \in \mathbb{R}^d} \hat{\overline{u}}_w + \lambda \sqrt{\frac{\hat{\boldsymbol{Var}}(u_w)}{n}}, \tag{12}$$

$$\begin{aligned} \text{with} \qquad & {u_w}^i \equiv \delta_i \min\{M, \frac{\exp(w \cdot \phi(x_i, y_i))}{p_i \cdot \mathbb{Z}(x_i)}\}, \\ & \hat{\overline{u}}_w \equiv \frac{1}{n} \sum_{i=1}^{n} {u_w}^i, \\ & \hat{\boldsymbol{Var}}(u_w) \equiv \frac{1}{n-1} \sum_{i=1}^{n} ({u_w}^i - \hat{\overline{u}}_w)^2. \end{aligned}$$

While the objective in Equation (12) is not convex in w (even for $\lambda = 0$), we find that batch and stochastic gradient descent compute h_w that have good generalization error (e.g., L-BFGS out of the box). The key subroutine that enables us to perform efficient gradient descent is a tractable way to compute ${u_w}^i$ and $\nabla_w({u_w}^i)$—both depend on $\mathbb{Z}(x_i)$ using the formulas

$${u_w}^i = \delta_i \min\{M, \frac{\exp(w \cdot \phi(x_i, y_i))}{p_i \cdot \mathbb{Z}(x_i)}\} \tag{13}$$

$$\nabla_w({u_w}^i) = \begin{cases} 0 & \text{if } \frac{\exp(w \cdot \phi(x_i, y_i))}{p_i \cdot \mathbb{Z}(x_i)} \geq M \\ \frac{\delta_i}{p_i} {u_w}^i \left\{ \phi(x_i, y_i) - \sum_{y'} \left[\phi(x_i, y') \frac{\exp(w \cdot \phi(x_i, y'))}{\mathbb{Z}(x_i)} \right] \right\} & \text{otherwise.} \end{cases}$$

For the special case when $\phi(x, y) = \overline{x} \otimes y$, where y is a bit vector $\in \{0, 1\}^L$, $\mathbb{Z}(x)$ has a simple decomposition:

$$\exp(w \cdot \phi(x, y)) = \prod_{l=1}^{L} \exp(y_l w_l \cdot x),$$

$$\mathbb{Z}(x) = \prod_{l=1}^{L} (1 + \exp(w_l \cdot x)),$$

where L is the length of the bit vector representation of y. For the general case, several approximation schemes have been developed to handle $\mathbb{Z}(x)$ for supervised training of graphical models and we can directly co-opt these for batch learning under bandit feedback as well.

5.3 POEM Iterated Variance Majorization Algorithm

We could use standard batch gradient descent methods to minimize the POEM training objective. In particular, prior work (Yu et al., 2010; Lewis and Overton, 2013) has established theoretically sound modifications to L-BFGS for non-smooth non-convex optimization. However, the following develops a stochastic method that can be much faster.

At first glance, the POEM training objective in Equation (12), specifically the variance term resists stochastic gradient optimization in the presented form. To remove this obstacle, we now develop a Majorization-Minimization scheme, similar in spirit to recent approaches to multi-class SVMs (van den Burg and Groenen, 2014) that can be shown to converge to a local optimum of the POEM training objective. In particular, we will show how to decompose $\sqrt{\hat{\boldsymbol{Var}}(u_w)}$ as a sum of differentiable functions (e.g., $\sum_i {u_w}^i$ or $\sum_i \{{u_w}^i\}^2$) so that we can optimize the overall training objective at scale using stochastic gradient descent.

Proposition 4 *For any w_0 such that $\hat{\boldsymbol{Var}}(u_{w_0}) > 0$,*

$$\begin{aligned}
\sqrt{\hat{\boldsymbol{Var}}(u_w)} &\leq A_{w_0} \sum_{i=1}^{n} {u_w}^i + B_{w_0} \sum_{i=1}^{n} \{{u_w}^i\}^2 + C_{w_0} \\
&= G(w; w_0). \\
A_{w_0} &\equiv -\frac{\hat{\overline{u}}_{w_0}}{(n-1)\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}}, \\
B_{w_0} &\equiv \frac{1}{2(n-1)\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}}, \\
C_{w_0} &\equiv \frac{n\{\hat{\overline{u}}_{w_0}\}^2}{2(n-1)\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}} + \frac{\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}}{2}.
\end{aligned}$$

Proof Consider a first order Taylor approximation of $\sqrt{\hat{\boldsymbol{Var}}(u_w)}$ around w_0. Observe that $\sqrt{\cdot}$ is concave.

$$\begin{aligned}
\sqrt{\hat{\boldsymbol{Var}}(u_w)} &\leq \sqrt{\hat{\boldsymbol{Var}}(u_{w_0})} + \nabla_z \sqrt{z} \mid_{z=\hat{\boldsymbol{Var}}(u_{w_0})} (\hat{\boldsymbol{Var}}(u_w) - \hat{\boldsymbol{Var}}(u_{w_0})) \\
&= \sqrt{\hat{\boldsymbol{Var}}(u_{w_0})} + \frac{\hat{\boldsymbol{Var}}(u_w) - \hat{\boldsymbol{Var}}(u_{w_0})}{2\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}} \\
&= \frac{\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}}{2} + \frac{1}{2\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}} \hat{\boldsymbol{Var}}(u_w) \\
&= \frac{\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}}{2} + \frac{\sum_{i=1}^{n} \{{u_w}^i\}^2}{2(n-1)\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}} + \frac{-n\{\hat{\overline{u}}_w\}^2}{2(n-1)\sqrt{\hat{\boldsymbol{Var}}(u_{w_0})}}.
\end{aligned}$$

Again Taylor approximate $-\{\hat{\overline{u}_w}\}^2$, noting that $-\{\cdot\}^2$ is concave.

$$\begin{aligned}
-\{\hat{\overline{u}_w}\}^2 &\leq -\{\hat{\overline{u}_{w_0}}\}^2 + \nabla_z(-z^2)\mid_{z=\hat{\overline{u}_{w_0}}} (\hat{\overline{u}_w} - \hat{\overline{u}_{w_0}}) \\
&= -\{\hat{\overline{u}_{w_0}}\}^2 + 2\{\hat{\overline{u}_{w_0}}\}^2 - 2\hat{\overline{u}_{w_0}}\hat{\overline{u}_w} \\
&= \{\hat{\overline{u}_{w_0}}\}^2 - \frac{2\hat{\overline{u}_{w_0}}}{n}\sum_{i=1}^{n} {u_w}^i.
\end{aligned}$$

Substituting above and re-arranging terms, we derive the proposition. ■

Iteratively minimizing $w^{t+1} = \text{argmin}_w G(w; w^t)$ ensures that the sequence of iterates $w^1, \ldots, w^{t+1}$ are successive minimizers of $\sqrt{\hat{\boldsymbol{Var}}(u_w)}$. Hence, during an epoch t, POEM proceeds by sampling uniformly $i \sim \mathcal{D}$, computing ${u_w}^i, \nabla {u_w}^i$ and, for learning rate η, updating

$$w \leftarrow w - \eta\{\nabla {u_w}^i + \lambda\sqrt{n}(A_{w_t}\nabla {u_w}^i + 2B_{w_t}{u_w}^i\nabla {u_w}^i)\}.$$

After each epoch, $w^{t+1} \leftarrow w$, and iterated minimization proceeds until convergence.

The complete algorithm is summarized as Algorithm 1. Software implementing POEM is available at http://www.cs.cornell.edu/~adith/poem/ for download, as is all the code and data needed to run each of the experiments reported in Section 6.

6. Empirical Evaluation

We now empirically evaluate the prediction performance and computational efficiency of POEM on a broad range of scenarios. To be able to control these experiments effectively, we derive bandit feedback from existing full-information data sets. As the learning task, we consider multi-label classification with input $x \in \mathbb{R}^p$ and prediction $y \in \{0,1\}^q$. Popular supervised algorithms that solve this problem include Structured SVMs (Tsochantaridis et al., 2004) and Conditional Random Fields (Lafferty et al., 2001). In the simplest case, CRF essentially performs logistic regression for each of the q labels independently. As outlined in Section 5, we use a joint feature map: $\phi(x, y) = x \otimes y$. We conducted experiments on different multi-label data sets collected from the LibSVM repository, with different ranges for p (features), q (labels) and n (samples) represented as summarized in Table 2.

Experiment methodology. We employ the Supervised $\mapsto$ Bandit conversion (Beygelzimer and Langford, 2009) method. Here, we take a supervised data set $\mathcal{D}^* = \{(x_1, y_1^*) \ldots (x_n, y_n^*)\}$

Name	p(# features)	q(# labels)	n_{train}	n_{test}
Scene	294	6	1211	1196
Yeast	103	14	1500	917
TMC	30438	22	21519	7077
LYRL	47236	4	23149	781265

Table 2: Corpus statistics for different multi-label data sets from the LibSVM repository. LYRL was post-processed so that only top level categories were treated as labels.

Algorithm 1 POEM pseudocode. An alternative version can use separate samplers for estimating ${u_w}^i$ and $\{{u_w}^i\}^2$ on Line 24.

1: **procedure** LossGradient($\mathcal{D}_s$, $\vec{w}$) ▷ Returns ${u_w}^i, \nabla_w({u_w}^i)$ for $i \in \mathcal{D}_s$
2: **for** $i \in \mathcal{D}_s$ **do**
3: $u^i \leftarrow {u_w}^i$. ▷ Equation (13)
4: $g^i \leftarrow \nabla_w({u_w}^i)$.
return $\vec{u}, \vec{g}$.
5: **procedure** ABC($\mathcal{D}$, $\vec{w}$, λ) ▷ Returns A_w, B_w, C_w from Proposition (4)
6: $\vec{u}, \vec{g} \leftarrow LossGradient(\mathcal{D}, \vec{w})$.
7: $R \leftarrow \sum_{i \in \mathcal{D}} u_i / n$.
8: $V \leftarrow \sqrt{\sum_{i \in \mathcal{D}} (u_i - R)^2 / (n-1)}$.
9: $A \leftarrow 1 - \frac{\lambda \sqrt{n} R}{(n-1)V}$.
10: $B \leftarrow \frac{\lambda}{2(n-1)V\sqrt{n}}$.
11: $C \leftarrow \frac{\lambda V}{2\sqrt{n}} + \frac{\lambda \sqrt{n} R^2}{2(n-1)V}$.
return A, B, C.
12: **procedure** SGD($\mathcal{D}$, λ, μ) ▷ $L2$ regularizer μ
13: $\vec{w} \leftarrow [0]_d$. ▷ Initial param
14: $\vec{h} \leftarrow [1]_d$. ▷ Adagrad history
15: **while** True **do**
16: Shuffle $\mathcal{D}$.
17: $A, B, C \leftarrow ABC(\mathcal{D}, w, \lambda)$.
18: **for** $\mathcal{D}_s \subset \mathcal{D}$ **do** ▷ Minibatch $|\mathcal{D}_s| = b$
19: $\vec{u}, \vec{g} \leftarrow LossGradient(\mathcal{D}_s, \vec{w})$.
20: $\overline{u} = \sum_{i \in \mathcal{D}_s} u_i / |\mathcal{D}_s|$.
21: $\overline{g} = \sum_{i \in \mathcal{D}_s} g_i / |\mathcal{D}_s|$.
22: $h_i \leftarrow h_i + {\overline{g}_i}^2$.
23: $j_i \leftarrow \overline{g}_i / \sqrt{h_i}$.
24: $\vec{\nabla} \leftarrow A\vec{j} + 2\mu\vec{w} + 2B\overline{u}\vec{j}$.
25: **if** $\|\vec{\nabla}\| \simeq 0$ **then return** $\vec{w}$. ▷ Gradient norm convergence
26: **if** $\overline{u} >$ avg $\overline{u}$ **then return** $\vec{w}$. ▷ Progressive validation
27: $\vec{w} \leftarrow \vec{w} - \eta\vec{\nabla}$. ▷ Step size η

and simulate a bandit feedback data set from a logging policy h_0 by sampling $y_i \sim h_0(x_i)$ and collecting feedback $\Delta(y_i^*, y_i)$. In principle, we could use any arbitrary stochastic policy as h_0. We choose a CRF trained on 5% of $\mathcal{D}^*$ as h_0 using default hyper-parameters, since they provide probability distributions amenable to sampling. In all the multi-label experiments, $\Delta(y^*, y)$ is the Hamming loss between the supervised label y^* vs. the sampled label y for input x. Hamming loss is just the number of incorrectly assigned labels (both false positives and false negatives). To create bandit feedback $\mathcal{D} = \{(x_i, y_i, \delta_i \equiv \Delta(y_i^*, y_i), p_i \equiv h_0(y_i \mid x_i))\}$, we take four passes through $\mathcal{D}^*$ and sample labels from h_0. Note that each supervised label is worth $\simeq |\mathcal{Y}| = 2^q$ bandit feedback labels. We can explore different learning strategies (e.g., IPS, CRM, etc.) on $\mathcal{D}$ and obtain learnt weight vectors w_{ips}, w_{crm}, etc. On the super-

vised test set, we then report the expected loss per instance $\mathcal{R} = \frac{1}{n_{test}} \sum_i \mathbb{E}_{y \sim h_w(x_i)} \Delta(y_i^*, y)$ and compare the generalization error of these learning strategies.

Baselines and learning methods. The expected Hamming loss of h_0 is the baseline to beat. Lower loss is better. The naïve, variance-agnostic approach to counterfactual learning (Bottou et al., 2013; Strehl et al., 2010) can be generalized to handle parametric multilabel classification by optimizing Equation (12) with $\lambda = 0$. We optimize it either using L-BFGS (IPS($\mathcal{B}$)) or stochastic optimization (IPS($\mathcal{S}$)). POEM($\mathcal{S}$) uses our Iterative-Majorization approach to variance regularization as outlined in Section 5.3, while POEM($\mathcal{B}$) is a L-BFGS variant. Finally, we report results from a supervised CRF as a skyline, despite its unfair advantage of having access to the full-information examples.

We keep aside 25% of $\mathcal{D}$ as a validation set—we use the unbiased counterfactual estimator from Equation (1) for selecting hyper-parameters. $\lambda = c\lambda^*$, where λ^* is the calibration factor from Section 4.4 and $c \in \{10^{-6}, \ldots, 1\}$ in multiples of 10. The clipping constant M is similarly set to the ratio of the $90\%ile$ to the $10\%ile$ propensity score observed in the training set of $\mathcal{D}$. The reported results are not sensitive to this choice of M, any reasonably large clipping constant suffices (e.g. even a simple, problem independent choice of $M = 100$ works well). When optimizing any objective over w, we always begin the optimization from $w = 0$, which is equivalent to $h_w = \text{uniform}(\mathcal{Y})$. We use mini-batch AdaGrad (Duchi et al., 2011) with batch size = 100 and step size $\eta = 1$ to adapt our learning rates for the stochastic approaches and use progressive validation (Blum et al., 1999) and gradient norms to detect convergence. Finally, the entire experiment set-up is run 10 times (i.e. h_0 trained on randomly chosen 5% subsets, $\mathcal{D}$ re-created, and test set performance of different approaches collected) and we report the averaged test set expected error across runs.

6.1 Does Variance Regularization Improve Generalization?

Results are reported in Table 3. We statistically test the performance of POEM against IPS (batch variants are paired together, and the stochastic variants are paired together) using a one-tailed paired difference t-test at significance level of 0.05 across 10 runs of the experiment, and find POEM to be significantly better than IPS on each data set and each optimization variant. Furthermore, on all data sets POEM learns a hypothesis that substantially improves over the performance of h_0. This suggests that the CRM principle is practically useful for designing learning algorithms, and that the variance regularizer is indeed beneficial.

6.2 How Computationally Efficient is POEM?

Table 4 shows the time taken (in CPU seconds) to run each method on each data set, averaged over different validation runs when performing hyper-parameter grid search. Some of the timing results are skewed by outliers, e.g., when under very weak regularization, CRFs tend to take longer to converge. However, it is still clear that the stochastic variants are able to recover good parameter settings in a fraction of the time of batch L-BFGS optimization, and this is even more pronounced when the number of labels grows—the run-time is dominated by computation of $\mathbb{Z}(x_i)$.

$\mathcal{R}$	Scene	Yeast	TMC	LYRL
h_0	1.543	5.547	3.445	1.463
IPS($\mathcal{B}$)	1.193	4.635	2.808	0.921
POEM($\mathcal{B}$)	1.168	4.480	2.197	0.918
IPS($\mathcal{S}$)	1.519	4.614	3.023	1.118
POEM($\mathcal{S}$)	1.143	4.517	2.522	0.996
CRF	0.659	2.822	1.189	0.222

Table 3: Test set Hamming loss, $\mathcal{R}$ for different approaches to multi-label classification on different data sets, averaged over 10 runs. POEM is significantly better than IPS on each data set and each optimization variant (one-tailed paired difference t-test at significance level of 0.05).

Time(s)	Scene	Yeast	TMC	LYRL
IPS($\mathcal{B}$)	2.58	47.61	136.34	21.01
IPS($\mathcal{S}$)	1.65	2.86	49.12	13.66
POEM($\mathcal{B}$)	75.20	94.16	949.95	561.12
POEM($\mathcal{S}$)	4.71	5.02	276.13	120.09
CRF	4.86	3.28	99.18	62.93

Table 4: Average time in seconds for each validation run for different approaches to multi-label classification. CRF is implemented by scikit-learn (Pedregosa et al., 2011). On all data sets, stochastic approaches are much faster than batch gradients.

6.3 Can MAP Predictions Derived From Stochastic Policies Perform Well?

For the policies learnt by POEM as shown in Table 3, Table 5 reports the averaged performance of the deterministic predictor derived from them. For a learnt weight vector w, this simply amounts to applying Equation (11). In practice, this method of generating predictions can be substantially faster than sampling since computing the argmax does not require computation of the partition function $\mathbb{Z}(x)$ which can be expensive in structured output prediction. From Table 5, we see that the loss of the deterministic predictor is typically not far from the loss of the stochastic policy, and often better.

6.4 How Does Generalization Improve With Size Of $\mathcal{D}$?

As we collect more data under h_0, our generalization error bound indicates that prediction performance should eventually approach that of the optimal hypothesis in the hypothesis space. We can simulate $n \to \infty$ by replaying the training data multiple times, collecting samples $y \sim h_0(x)$. In the limit, we would observe every possible y in the bandit feedback data set, since $h_0(x)$ has non-zero probability of exploring each prediction y. However, the learning rate may be slow, since the exponential model family has very thin tails, and

$\mathcal{R}$	Scene	Yeast	TMC	LYRL
POEM($\mathcal{S}$)	1.143	4.517	2.522	0.996
POEM($\mathcal{S}$)$_{map}$	1.143	4.065	2.299	0.880

Table 5: Mean Hamming loss of MAP predictions from the policies in Table 3. POEM$_{map}$ is significantly better than POEM on all data sets except Scene (one-sided paired difference t-test, significance level 0.05).

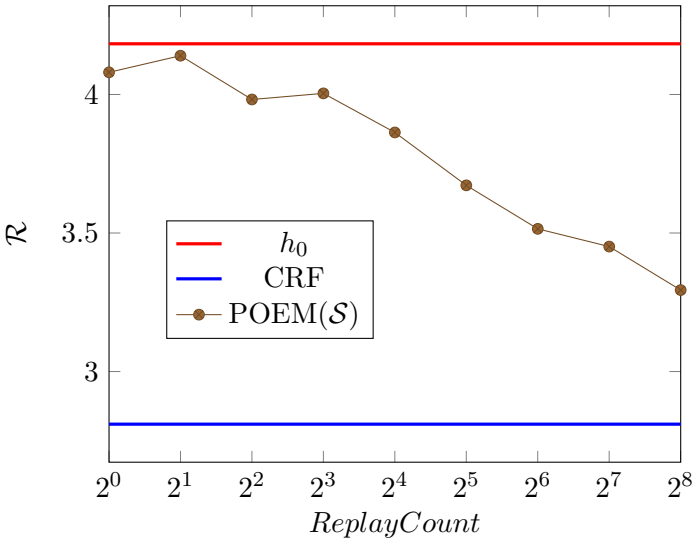


Figure 1: Generalization performance of POEM($\mathcal{S}$) as a function of n on the Yeast data set.

hence may not be an ideal logging distribution to learn from. Holding all other details of the experiment setup fixed, we vary the number of times we replayed the training set (*ReplayCount*) to collect samples from h_0, and report the performance of POEM($\mathcal{S}$) on the Yeast data set in Figure 1. As expected, performance of POEM improves with increasing sample size. Note that even with $\mathit{ReplayCount} = 2^8$, POEM($\mathcal{S}$) is learning from much less information than the CRF, where each supervised label conveys 2^{14} bandit label feedbacks.

6.5 How Does Quality of h_0 Affect Learning?

In this experiment, we change the fraction of the training set $f \cdot n_{train}$ that was used to train the logging policy—and as f is increased, the quality of h_0 improves. Intuitively, there's a trade-off: better h_0 probably samples correct predictions more often and so produces a higher quality $\mathcal{D}$ to learn from, but it should also be harder to beat h_0. We vary f from 1% to 100% while keeping all other conditions identical to the original experiment setup in Figure 2, and find that POEM($\mathcal{S}$) is able to consistently find a hypothesis at least as good

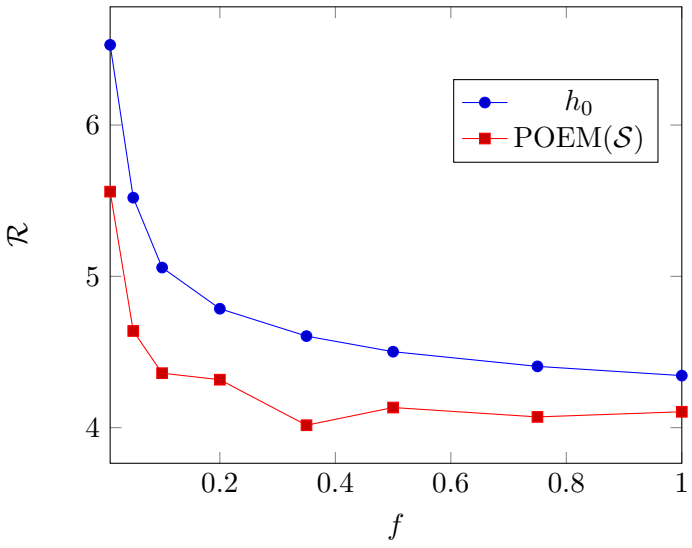


Figure 2: Performance of POEM($\mathcal{S}$) on the Yeast data set as h_0 is improved. The fraction f of the supervised training set used to train h_0 is varied to control h_0's quality. h_0 performance does not reach CRF when $f = 1$ because we do not tune hyper-parameters, and we report its expected loss, not the loss of its MAP prediction.

as h_0. Moreover, even $\mathcal{D}$ collected from a poor quality h_0 ($0.5 \leq f \leq 0.2$) allows POEM($\mathcal{S}$) to effectively learn an improved policy.

6.6 How Does Stochasticity of h_0 Affect Learning?

Finally, the theory suggests that counterfactual learning is only possible when h_0 is sufficiently stochastic (the generalization bounds hold with high probability in the samples drawn from h_0). Does CRM degrade gracefully when this assumption is violated? We test this by introducing the *temperature* multiplier $w \mapsto \alpha w, \alpha > 0$ (as discussed in Section 5) into the logging policy. For $h_0 = h_{w_0}$, we scale $w_0 \mapsto \alpha w_0$, to derive a "less stochastic" variant of h_0, and generate $\mathcal{D} \sim h_{\alpha w_0}$. We report the performance of POEM($\mathcal{S}$) on the LYRL data set in Figure 3 as we change $\alpha \in [0.5, \ldots, 32]$, compared against h_0, and the deterministic predictor— h_0 *map*—derived from h_0. So long as there is some minimum amount of stochasticity in h_0, POEM($\mathcal{S}$) is still able to find a w that improves upon h_0 and h_0 *map*. The margin of improvement is typically greater when h_0 is more stochastic. Even when h_0 is barely stochastic ($\alpha \geq 2^4$), performance of POEM($\mathcal{S}$) simply recovers h_0 *map*, suggesting that the CRM principle indeed achieves robust learning.

We observe the same trends (Figures 1, 2 and 3) across all data sets and optimization variants. They also remain unchanged when we include $l2-$regularization (analogous to supervised CRFs to capture the capacity of $\mathcal{H}_{lin}$).

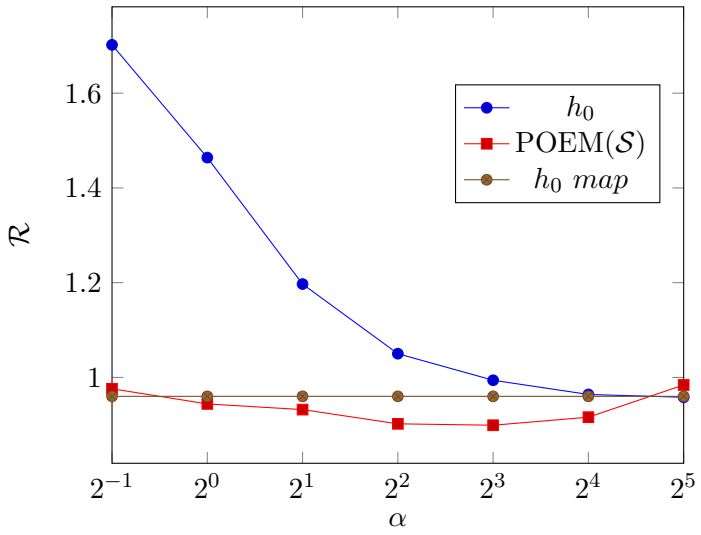


Figure 3: Performance of POEM($\mathcal{S}$) on the LYRL data set as h_0 becomes less stochastic. For $\alpha \geq 2^5$, $h_0 \equiv h_0$ *map* (within machine precision).

7. Real-World Application

We now demonstrate how POEM (and in general the CRM principle) can be instantiated effectively in real world settings. Bloomberg, the financial and media company in New York, had the following challenging retrieval problem: the task was to train a high-precision classifier that could reliably pick the best answer d^* (or none, if none answered the query) from a pool of candidate answers $\mathcal{Y}(x)$ for query x, where $\mathcal{Y}(x)$ was generated by an existing high-recall retrieval function. The challenge lay in collecting supervised labeled data that could be used to train this high-precision classifier.

Before we started our experiment with POEM, an existing high-precision classifier was already in operation. It was trained using a few labeled examples (x, d^*), but scaling up the system to achieve improved accuracy appeared challenging given the cost of acquiring new (x, d^*) pairs that mimicked what the system saw during its operation. However, it was possible to collect logs of the system, where each entry contained a query x and the features $\phi(x, d)$ describing each candidate answer $d \in \mathcal{Y}(x)$. The high-precision classifier could be modeled as a logistic regression classifier with weights w and a threshold τ. Each candidate was scored using w, $s(d) = w \cdot \phi(x, d)$. If the highest scoring candidate $s(d^*) \geq \tau$, it was selected as the answer and otherwise the system abstained.

This existing system could easily be adapted to provide $\mathcal{D}$ as needed by POEM. For each x, a dummy $d_0 \in \mathcal{Y}(x)$ is added to the candidate pool to model abstention. During the operation of the system, answers are *sampled* according to $\frac{\exp(\alpha \cdot s(d))}{\mathbb{Z}}$. $\mathbb{Z}$ is the partition function to ensure this is a valid sampling distribution, $\mathbb{Z} = \sum_{d \in \mathcal{Y}(x) \cup d_0} \exp(\alpha \cdot s(d))$. Abstention is modeled by the fact that d_0 is sampled with probability proportional to $\exp(\alpha \cdot s(d_0))$. α is a temperature constant so that the system can be tuned to sample abstentions at

roughly the same rate as its deterministic counterpart. Finally, the end-result feedback ($\delta \in \{\texttt{thumbs-up}, \texttt{thumbs-down}\}$ represented as binary feedback) was logged and provided bandit feedback for the presented answer d.

This data set was much easier to collect during the system run compared to annotating each x in the logs with the best possible d^* that would have answered the query. We argue that this is a general, practical, alternative approach to training retrieval systems: use any strategy with very high recall to construct $\mathcal{Y}$, then use the parameters w estimated using the CRM principle to search through this $\mathcal{Y}$ and find a precise answer.

On a small pilot study, we acquired $\mathcal{D}$ with $\simeq 4000$ $(x, d, \frac{\exp(\alpha \cdot s(d))}{\mathbb{Z}}, \delta)$ tuples in the training set and $\simeq 500$ tuples in the validation and test sets. We verified that the existing high-precision classifier was statistically significantly better than random baselines for the problem. POEM($\mathcal{S}$) is trained on this log data by performing gradient descent with w initialized to $w_0 = 0$ and validating $c \in \left[10^{-6}, \ldots 1\right], \lambda = c\lambda^*$ as described in Sections 4.4 and 6. POEM($\mathcal{S}$) found a w^* that improved δ feedback over the existing system by over 30%, as estimated using the unbiased counterfactual estimator of Equation (1) on the test set. Without using the variance regularizer, the IPS($\mathcal{S}$) found a w^* that degraded the system performance by 3.5% estimated counterfactually in the same way. This shows that POEM and the CRM principle can bring potential benefit even in binary-feedback multi-class classification settings where classic supervised learning approaches lack available data.

8. Conclusion

Counterfactual risk minimization serves as a robust principle for designing algorithms that can learn from a batch of bandit feedback interactions. The key insight for CRM is to expand the classical notion of a hypothesis class to include stochastic policies, reason about variance in the risk estimator, and derive a generalization error bound over this hypothesis space. The practical take-away is a simple, data-dependent regularizer that guarantees robust learning. Following the CRM principle, we developed the POEM learning algorithm for structured output prediction. POEM can optimize over rich policy families (exponential models corresponding to linear rules in supervised learning), and deal with massive output spaces as efficiently as classical supervised methods.

The CRM principle more generally applies to supervised learning with non-differentiable losses, since the objective does not require the gradient of the loss function. We also foresee extensions of the algorithm to handle ordinal or co-active feedback models for $\delta(\cdot, \cdot)$, and extensions of the generalization error bound to include adaptive or deterministic h_0, etc.

Acknowledgments

This research was funded in part through NSF Awards IIS-1247637, IIS-1217686, IIS-1513692, the JTCII Cornell-Technion Research Fund, and a gift from Bloomberg.

References

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, New York, NY, USA, 2009.

Alina Beygelzimer and John Langford. The offset tree for learning with partial labels. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 129–138, 2009.

Avrim Blum, Adam Kalai, and John Langford. Beating the hold-out: Bounds for k-fold and progressive cross-validation. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 203–208, 1999.

Léon Bottou, Jonas Peters, Joaquin Q. Candela, Denis X. Charles, Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Y. Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, pages 442–450, 2010.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1097–1104, 2011.

Nicolas Galichet, Michèle Sebag, and Olivier Teytaud. Exploration vs exploitation vs safety: Risk-aware multi-armed bandits. In *Asian Conference on Machine Learning*, pages 245–260, 2013.

J. Garcia and F. Fernandez. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012.

Katja Hofmann, Anne Schuth, Shimon Whiteson, and Maarten de Rijke. Reusing historical interaction data for faster online learning to rank for IR. In *Sixth ACM International Conference on Web Search and Data Mining*, pages 183–192, 2013.

Edward L. Ionides. Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311, 2008.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, 2001.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Proceedings of the 21st Annual Conference on Neural Information Processing Systems*, pages 817–824, 2008.

John Langford, Alexander Strehl, and Jennifer Wortman. Exploration scavenging. In *Proceedings of the 25th International Conference on Machine Learning*, pages 528–535, 2008.

Adrian S. Lewis and Michael L. Overton. Nonsmooth optimization via quasi-Newton methods. *Mathematical Programming*, 141(1-2):135–163, 2013.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, pages 661–670, 2010.

Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 297–306, 2011.

Lihong Li, Shunbao Chen, Jim Kleban, and Ankur Gupta. Counterfactual estimation and optimization of click metrics for search engines. *CoRR*, abs/1403.1891, 2014.

Lihong Li, Remi Munos, and Csaba Szepesvari. Toward minimax off-policy value estimation. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Jérémie Mary, Philippe Preux, and Olivier Nicol. Improving offline evaluation of contextual bandit algorithms via bootstrapping techniques. In *Proceedings of the 31st International Conference on Machine Learning*, pages 172–180, 2014.

Andreas Maurer and Massimiliano Pontil. Empirical bernstein bounds and sample-variance penalization. In *Proceedings of the 22nd Conference on Learning Theory*, 2009.

Art B. Owen. *Monte Carlo Theory, Methods and Examples.* 2013.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Pannagadatta K. Shivaswamy and Thorsten Joachims. Multi-armed bandit problems with history. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, pages 1046–1054, 2012.

Alexander L. Strehl, John Langford, Lihong Li, and Sham Kakade. Learning from logged implicit exploration data. In *Proceedings of the 24th Annual Conference on Neural Information Processing Systems*, pages 2217–2225, 2010.

Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Philip S. Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 3000–3006, 2015.

Ioannis Tsochantaridis, Thomas Hofmann, Thorsten Joachims, and Yasemin Altun. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the 21st International Conference on Machine Learning*, pages 104–, 2004.

G.J.J. van den Burg and P.J.F. Groenen. GenSVM: A Generalized Multiclass Support Vector Machine. Technical Report EI 2014-33, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute, 2014.

W. N. Wapnik and A. J. Tscherwonenkis. *Theorie der Zeichenerkennung*. Akademie Verlag, Berlin, 1979.

Jin Yu, S. V. N. Vishwanathan, Simon Günter, and Nicol N. Schraudolph. A quasi-Newton approach to nonsmooth convex optimization problems in machine learning. *Journal of Machine Learning Research*, 11:1145–1200, 2010.

Journal of Machine Learning Research 16 (2015) 1757-1792 Submitted 7/15; Published 9/15

Optimal Estimation of Low Rank Density Matrices

Vladimir Koltchinskii* VLAD@MATH.GATECH.EDU
Dong Xia † DXIA7@MATH.GATECH.EDU
School of Mathematics
Georgia Institute of Technology
Atlanta, GA 30332, USA.

Editor: Alex Gammerman and Vladimir Vovk

Abstract

The density matrices are positively semi-definite Hermitian matrices of unit trace that describe the state of a quantum system. The goal of the paper is to develop minimax lower bounds on error rates of estimation of low rank density matrices in trace regression models used in quantum state tomography (in particular, in the case of Pauli measurements) with explicit dependence of the bounds on the rank and other complexity parameters. Such bounds are established for several statistically relevant distances, including quantum versions of Kullback-Leibler divergence (relative entropy distance) and of Hellinger distance (so called Bures distance), and Schatten p-norm distances. Sharp upper bounds and oracle inequalities for least squares estimator with von Neumann entropy penalization are obtained showing that minimax lower bounds are attained (up to logarithmic factors) for these distances.

Keywords: quantum state tomography, low rank density matrix, minimax lower bounds

1. Introduction

This paper deals with optimality properties of estimators of density matrices, describing states of quantum systems, that are based on penalized empirical risk minimization with specially designed complexity penalties such as von Neumann entropy of the state. Alexey Chervonenkis was a co-founder of the theory of empirical risk minimization that is of crucial importance in machine learning, but he also had very broad interests that included, in particular, quantum mechanics. By the choice of the topic, we would like to honor the memory of this great man and great scientist.

Let $\mathbb{M}_m(\mathbb{C})$ be the set of all $m \times m$ matrices with complex entries and let $\mathbb{H}_m = \mathbb{H}_m(\mathbb{C}) \subset \mathbb{M}_m(\mathbb{C})$ be the set of all Hermitian matrices: $\mathbb{H}_m = \{A \in \mathbb{M}_m(\mathbb{C}) : A = A^*\}$, A^* denoting the adjoint matrix of A. For $A \in \mathbb{H}_m$, $\mathrm{tr}(A)$ denotes the trace of A and $A \succcurlyeq 0$ means that A is positively semi-definite. Let $\mathcal{S}_m := \{S \in \mathbb{H}_m : S \succcurlyeq 0, \mathrm{tr}(S) = 1\}$ be the set of all positively semi-definite Hermitian matrices of unit trace called *density matrices*. In quantum mechanics, the state of a quantum system is usually characterized by a density matrix $\rho \in \mathcal{S}_m$ (or, more generally, by a self-adjoint positively semi-definite operator of unit trace acting in an infinite-dimensional Hilbert space, called a density operator). Often, very

*. Supported in part by NSF Grants DMS-1509739, DMS-1207808, CCF-1523768 and CCF-1415498
†. Supported in part by NSF Grant DMS-1207808

large density matrices are needed to represent or to approximate the density operator of the state. For instance, for a quantum system consisting of b qubits, the density matrices are of the size $m \times m$ with $m = 2^b$, so the dimension of the density matrix grows exponentially with b. For instance, for a 10 qubit system, one has to deal with matrices that have 2^{20} entries. Thus, it becomes natural in the problems of statistical estimation of density matrix ρ to take an advantage of the fact that it might be low rank, or nearly low rank (that is, it could be well approximated by low rank matrices) which reduces the complexity of the estimation problem.

In *quantum state tomography (QST)*, the goal is to estimate an unknown state $\rho \in \mathcal{S}_m$ based on a number of specially designed measurements for the system prepared in state ρ (see Gross et al. 2010, Gross 2011, Koltchinskii 2011a, Cai et al. 2015 and references therein). Given an observable $A \in \mathbb{H}_m$ with spectral representation $A = \sum_{j=1}^{m'} \lambda_j P_j$, where $m' \leq m$, λ_j being the eigenvalues of A and P_j being the corresponding mutually orthogonal eigenprojectors, the outcome of a measurement of A for the system prepared in state ρ is a random variable Y taking values λ_j with probabilities $\mathrm{tr}(\rho P_j)$. The expectation of Y is then $\mathbb{E}_\rho Y = \mathrm{tr}(\rho A)$, so, Y could be viewed as a noisy observation of the value of linear functional $\mathrm{tr}(\rho A)$ of the unknown density matrix ρ. A common approach is to choose an observable A at random, assuming that it is the value of a random variable X with some design distribution Π in the space $\mathbb{H}_m$. More precisely, given a sample of n i.i.d. copies $X_1, \dots, X_n$ of X, n measurements are being performed for the system identically prepared n times in state ρ resulting in outcomes $Y_1, \dots, Y_n$. Based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$, the goal is to estimate the target density matrix ρ. Clearly, the observations satisfy the following model

$$Y_j = \mathrm{tr}(\rho X_j) + \xi_j, \ j = 1, \dots, n, \tag{1}$$

where $\{\xi_j\}$ is a random noise consisting of n i.i.d. random variables satisfying the condition $\mathbb{E}_\rho(\xi_j | X_j) = 0, j = 1, \dots, n$. This is a special case of so called *trace regression model* intensively studied in the recent literature (see, e.g., Koltchinskii et al. 2011, Koltchinskii 2011b and references therein).

1.1 Assumptions

A common choice of design distribution in this type of problems is so called *uniform sampling from an orthonormal basis* described in the following assumptions.

Assumption 1 *Let $\mathcal{E} = \{E_1, \dots, E_{m^2}\} \subset \mathbb{H}_m$ be an orthonormal basis of $\mathbb{H}_m$ with respect to the Hilbert–Schmidt inner product: $\langle A, B \rangle = tr(AB)$. Moreover, suppose that, for some $U > 0$,*

$$\|E_j\|_\infty \leq U, j = 1, \dots, n,$$

where $\|\cdot\|_\infty$ denotes the operator norm (the spectral norm).

Since $\|E_j\|_2 = 1$, where $\|\cdot\|_2$ denotes the Hilbert–Schmidt (or Frobenius) norm, we can assume that $U \leq 1$. Moreover, $U \geq m^{-1/2}$ since $1 = \|E_j\|_2 \leq m^{1/2}\|E_j\|_\infty \leq m^{1/2}U$.

Assumption 2 *Let Π be the uniform distribution in the finite set $\mathcal{E}$ (see Assumption 1), let X be a random variable sampled from Π and let $X_1, \dots, X_n$ be i.i.d. copies of X.*

It will be assumed in what follows that assumptions 1 and 2 hold (unless it is stated otherwise). Under these assumptions, $Y_1, \ldots, Y_n$ could be viewed as noisy observations of a random sample of Fourier coefficients $\langle \rho, X_1 \rangle, \ldots, \langle \rho, X_n \rangle$ of the target density matrix ρ in the basis $\mathcal{E}$. The above model (in which $X_1, \ldots, X_n$ are uniformly sampled from an orthonormal basis and $Y_1, \ldots, Y_n$ are the outcomes of measurements of the observables $X_1, \ldots, X_n$ for the system being identically prepared n times in the same state ρ) will be called in what follows the *standard QST model*. It is a special case of *trace regression model with bounded response*:

Assumption 3 (Trace regression with bounded responce) *Suppose that Assumption 1 holds and let (X, Y) be a random couple such that X is sampled from the uniform distribution Π in an orthonormal basis $\mathcal{E} \subset \mathbb{H}_m$. Suppose also that, for some $\rho \in \mathcal{S}_m$, $\mathbb{E}(Y|X) = \langle \rho, X \rangle$ a.s. and, for some $\bar{U} > 0$, $|Y| \leq \bar{U}$ a.s.. The data $(X_1, Y_1), \ldots (X_n, Y_n)$ consists of n i.i.d. copies of (X, Y).*

We are also interested in the *trace regression model with Gaussian noise*:

Assumption 4 (Trace regression with Gaussian noise) *Suppose Assumption 1 holds and let (X, Y) be a random couple such that X is sampled from the uniform distribution Π in an orthonormal basis $\mathcal{E} \subset \mathbb{H}_m$ and, for some $\rho \in \mathcal{S}_m$, $Y = \langle \rho, X \rangle + \xi$, where ξ is a normal random variable with mean 0 and variance σ_ξ^2, ξ and X being independent. The data $(X_1, Y_1), \ldots (X_n, Y_n)$ consists of n i.i.d. copies of (X, Y).*

Note that this model is not directly applicable to the "standard QST problem" described above, where the response variable Y is discrete. However, if the measurements are repeated multiple times for each observable X_j and the resulting outcomes are averaged to reduce the variance, the noise of such averaged measurements becomes approximately Gaussian and it is of interest to characterize the estimation error in terms of the variance of the noise.

An important example of an orthonormal basis used in quantum state tomography is so called *Pauli basis*, see, e.g., Gross et al. (2010), Gross (2011). The Pauli basis in the space $\mathbb{H}_2$ of 2×2 Hermitian matrices (observables in a single qubit system) consists of four matrices W_1, W_2, W_3, W_4 defined as $W_i = \frac{1}{\sqrt{2}}\sigma_i,\ i = 1, \ldots, 4$, where

$$\sigma_1 := \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_2 := \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 := \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_4 := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

It is easy to check that $\{W_0, W_1, W_2, W_3\}$ indeed forms an orthonormal basis in $\mathbb{H}_2$. The Pauli basis in the space $\mathbb{H}_m$ for $m = 2^b$ (the space of observables for a b qubits system) is defined by tensorisation, namely, it consists of 4^b tensor products $W_{i_1} \otimes \ldots \otimes W_{i_b}, (i_1, \ldots, i_b) \in \{1, 2, 3, 4\}^b$. Let us write these matrices as $E_1, \ldots, E_{m^2}$ with $E_1 = W_1 \otimes \ldots \otimes W_1$. It is easy to see that each of them has eigenvalues $\pm\frac{1}{\sqrt{m}}$ and $\|E_j\|_\infty = m^{-1/2}$, so, for this basis, $U = m^{-1/2}$. The fact that, for the Pauli basis, the operator norms of basis matrices are as small as possible plays an important role in quantum state tomography (Gross et al., 2010; Gross, 2011; Liu, 2011). Let $E_j = \frac{1}{\sqrt{m}}Q_j^+ - \frac{1}{\sqrt{m}}Q_j^-$ be the spectral representation of E_j. Then, an outcome of a measurement of E_j in state ρ is a random variable τ_j taking values

$\pm\frac{1}{\sqrt{m}}$ with probabilities $\langle \rho, Q_t^{\pm}\rangle$. Its expectation is $\mathbb{E}_\rho \tau_j = \langle \rho, E_j\rangle$. Of course, there exists a unique representation of density matrix ρ in the Pauli basis that can be written as follows: $\rho = \sum_{j=1}^{m^2} \frac{\alpha_j}{\sqrt{m}} E_j$ with $\alpha_1 = 1$. Then, we clearly have $\mathbb{E}_\rho \tau_j = \frac{\alpha_j}{\sqrt{m}}$ and $\mathbb{P}_\rho\Big\{\tau_j = \pm\frac{1}{\sqrt{m}}\Big\} = \frac{1\pm\alpha_j}{2}$ (for $j = 1$, this gives $\mathbb{P}_\rho\Big\{\tau_1 = \frac{1}{\sqrt{m}}\Big\} = 1$). As a consequence, $\mathrm{Var}_\rho(\tau_j) = \frac{1-\alpha_j^2}{m}$. Note that $\sum_{j=1}^{m^2} \frac{\alpha_j^2}{m} = \|\rho\|_2^2 \le \mathrm{tr}^2(\rho) = 1$. This implies that there exists j such that $\alpha_j^2 \le \frac{1}{2}$ and $\mathrm{Var}_\rho(\tau_j) \ge \frac{1}{2m}$. In fact, the number of such j must be large, say, at least $\frac{m^2}{2}$ (provided that $m > 4$). Thus, for "most" of the values of j, $\mathrm{Var}_\rho(\tau_j) \asymp \frac{1}{m}$. A way to reduce the variance is to repeat the measurement of each observable X_j K times (for a system identically prepared in state ρ) and to average the outcomes of such K measurements. The resulting response variable is $Y_j = \langle \rho, X_j\rangle + \xi_j$, where $\mathbb{E}_\rho(\xi_j|X_j) = 0$ and $\mathbb{E}_\rho(\xi_j^2|X_j) = \mathrm{Var}_\rho(Y_j|X_j) = \frac{1-\alpha_{\nu_j}^2}{Km}$, ν_j being defined by the relationship $X_j = E_{\nu_j}$.

1.2 Preliminaries and Notations

Some notations will be used throughout the paper. The Euclidean norm in $\mathbb{C}^m$ will be denoted by $\|\cdot\|$ and the notation $\langle\cdot,\cdot\rangle$ will be used for both the Euclidean inner product in $\mathbb{C}^m$ and for the Hilbert–Schmidt inner product in $\mathbb{H}_m$. $\|\cdot\|_p, p \ge 1$ will be used to denote the *Schatten* p-norm in $\mathbb{H}_m$, namely $\|A\|_p^p = \sum\limits_j^m |\lambda_j(A)|^p,\ A \in \mathbb{H}_m$, $\lambda_1(A) \ge \ldots \ge \lambda_m(A)$ being the eigenvalues of A. In particular, $\|\cdot\|_2$ denotes the Hilbert–Schmidt (or Frobenius) norm, $\|\cdot\|_1$ denotes the nuclear (or trace) norm and $\|\cdot\|_\infty$ denotes the operator (or spectral) norm: $\|A\|_\infty = \max_{1\le j\le m} |\lambda_j(A)| = |\lambda_1(A)|$. The following well known *interpolation inequality* for Schatten p-norms will be used to extend the bounds proved for some values of p to the whole range of its values. It easily follows from similar bounds for ℓ_p-spaces.

Lemma 1 (Interpolation inequality) *For $1 \le p < q < r \le \infty$, and let $\mu \in [0,1]$ be such that*

$$\frac{\mu}{p} + \frac{1-\mu}{r} = \frac{1}{q}.$$

Then, for all $A \in \mathbb{H}_m$,

$$\|A\|_q \le \|A\|_p^{\mu} \|A\|_r^{1-\mu}.$$

Given $A \in \mathbb{H}_m$, define a function $f_A : \mathbb{H}_m \mapsto \mathbb{R} : f_A(x) := \langle A, x\rangle, x \in \mathbb{H}_m$. For a given random variable X in $\mathbb{H}_m$ with a distribution Π, we have $\|f_A\|_{L_2(\Pi)}^2 = \mathbb{E} f_A^2(X) = \mathbb{E}\langle A, X\rangle^2$. Sometimes, with a minor abuse of notation, we might write $\|A\|_{L_2(\Pi)}^2 = \int_{\mathbb{H}_m} \langle A, x\rangle^2 \Pi(dx) = \|f_A\|_{L_2(\Pi)}^2$. In what follows, Π will be typically the uniform distribution in an orthonormal basis $\mathcal{E} = \{E_1, \ldots, E_{m^2}\} \subset \mathbb{H}_m$, implying that

$$\|f_A\|_{L_2(\Pi)}^2 = \|A\|_{L_2(\Pi)}^2 = m^{-2}\|A\|_2^2,$$

so, the $L_2(\Pi)$-norm is just a rescaled Hilbert–Schmidt norm.

Consider $A \in \mathbb{H}_m$ with spectral representation $A = \sum_{j=1}^{m'} \lambda_j P_j$, $m' \le m$ with distinct non-zero eigenvalues λ_j. Denote by $\mathrm{sign}(A) := \sum_{j=1}^{m'} \mathrm{sign}(\lambda_j) P_j$ and by $\mathrm{supp}(A)$ the linear

span of the images of projectors $P_j, j = 1, \dots, m'$ (the subspace $\mathrm{supp}(A) \subset \mathbb{C}^m$ will be called *the support* of A).

Given a subspace $L \subset \mathbb{C}^m$, $L^\perp$ denotes the orthogonal complement of L and P_L denotes the orthogonal projection onto L. Let $\mathcal{P}_L, \mathcal{P}_L^\perp$ be orthogonal projection operators in the space $\mathbb{H}_m$ (equipped with the Hilbert–Schmidt inner product), defined as follows:

$$\mathcal{P}_L^\perp(A) = P_{L^\perp} A P_{L^\perp}, \quad \mathcal{P}_L(A) = A - P_{L^\perp} A P_{L^\perp}.$$

These two operators split any Hermitian matrix A into two orthogonal parts, $\mathcal{P}_L(A)$ and $\mathcal{P}_L^\perp(A)$, the first one being of rank at most $2\dim(L)$.

For a convex function $f : \mathbb{H}_m \mapsto \mathbb{R}$, $\partial f(A)$ denotes the subdifferential of f at the point $A \in \mathbb{H}_m$. It is well known that

$$\partial \|A\|_1 = \Big\{ \mathrm{sign}(A) + \mathcal{P}_L^\perp(M) : M \in \mathbb{H}_m, \|M\|_\infty \le 1 \Big\}, \tag{2}$$

where $L = \mathrm{supp}(A)$ (see Koltchinskii 2011b, p. 240 and references therein).

C, C_1, C', c, c', etc will denote constants (that do not depend on parameters of interest such as m, n, etc) whose values could change from line to line (or, even, within the same line) without further notice. For nonnegative A and B, $A \lesssim B$ (equivalently, $B \gtrsim A$) means that $A \le CB$ for some absolute constant $C > 0$, and $A \asymp B$ means that $A \lesssim B$ and $B \lesssim A$. Sometimes, symbols $\lesssim, \gtrsim$ and $\asymp$ could be provided with subscripts (say, $A \lesssim_\gamma B$) to indicate that constant C may depend on a parameter (say, γ).

In what follows, P denotes the distribution of (X, Y) and P_n denotes the corresponding empirical distribution based on the sample $(X_1, Y_1), \dots, (X_n, Y_n)$ of n i.i.d. observations. Similarly, Π is the distribution of X (typically, uniform in an orthonormal basis) and Π_n is the corresponding empirical distribution based on the sample $(X_1, \dots, X_n)$. We will use standard notations $Pf = \mathbb{E} f(X, Y), P_n f = n^{-1} \sum_{j=1}^n f(X_j, Y_j)$ and $\Pi g = \mathbb{E} g(X), P_n g = n^{-1} \sum_{j=1}^n g(X_j)$.

1.3 Estimation Methods

Recall that the central problem in quantum state tomography is to estimate a large density matrix ρ based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$ satisfying the trace regression model. Often, the goal is to develop adaptive estimators with optimal dependence of the estimation error (measured by various statistically relevant distances) on the unknown rank of the target matrix ρ under the assumption that ρ is low rank, or on other complexity parameters in the case when the target matrix ρ can be well approximated by low rank matrices.

The simplest estimation procedure for density matrix ρ is the least squares estimator defined by the following convex optimization problem:

$$\hat{\rho} := \mathop{\arg\min}_{S \in \mathcal{S}_m} \frac{1}{n} \sum_{j=1}^n \left(Y_j - \langle S, X_j \rangle \right)^2. \tag{3}$$

Since, for all $S \in \mathcal{S}_m$, $\|S\|_1 = \mathrm{tr}(S) = 1$, we have that

$$\hat{\rho} = \hat{\rho}^\varepsilon := \mathop{\arg\min}_{S \in \mathcal{S}_m} \left[\frac{1}{n} \sum_{j=1}^n \left(Y_j - \langle S, X_j \rangle \right)^2 + \varepsilon \|S\|_1 \right], \quad \varepsilon \ge 0. \tag{4}$$

Thus, in the case of density matrices, the least squares estimator $\hat{\rho}$ coincides with the *matrix LASSO* estimator $\hat{\rho}^{\varepsilon}$ with nuclear norm penalty and arbitrary value of regularization parameter ε. The nuclear norm penalty is used as a proxy of the rank that provides a convex relaxation for rank penalized least squares method. Matrix LASSO is a standard method of low rank estimation in trace regression models that has been intensively studied in the recent years, see, for instance, Candés and Plan (2011), Rohde and Tsybakov (2011), Koltchinskii (2011b), Koltchinskii et al. (2011), Negahban and Wainwright (2010) and references therein. In the case of estimation of density matrices, due to their positive semidefiniteness and trace constraint, the nuclear norm penalization is present implicitly even in the case of a non-penalized least squares estimator $\hat{\rho}$ (see also Koltchinskii 2013a, Kalev et al. 2015 where similar ideas were used).

Note that the estimator $\hat{\rho}$ can be also rewritten as

$$\hat{\rho} := \mathop{\arg\min}_{S\in\mathcal{S}_m}\biggl[\|S\|^2_{L_2(\Pi_n)} - \frac{2}{n}\sum_{j=1}^n Y_j\langle S, X_j\rangle\biggr]. \tag{5}$$

Replacing the empirical $\|\cdot\|_{L_2(\Pi_n)}$-norm with the "true" $\|\cdot\|_{L_2(\Pi)}$-norm (which could make sense in the case when the design distribution Π is known) yields the following *modified least squares* estimator studied in Koltchinskii et al. (2011), Koltchinskii (2013a):

$$\check{\rho} := \mathop{\arg\min}_{S\in\mathcal{S}_m}\biggl[\|S\|^2_{L_2(\Pi)} - \frac{2}{n}\sum_{j=1}^n Y_j\langle S, X_j\rangle\biggr]. \tag{6}$$

Another estimator was proposed in Koltchinskii (2011a) and it is based on an idea of using so called *von Neumann entropy* as a penalizer in least squares method. Von Neumann entropy is a canonical extension of Shannon's entropy to the quantum setting. For a density matrix $S \in \mathcal{S}_m$, it is defined as $\mathcal{E}(S) := -\mathrm{tr}(S\log S)$. The estimator proposed in Koltchinskii (2011a) is defined as follows

$$\tilde{\rho}^{\varepsilon} := \mathop{\arg\min}_{S\in\mathcal{S}_m}\biggl[\frac{1}{n}\sum_{j=1}^n (Y_j - \langle S, X_j\rangle)^2 + \varepsilon\mathrm{tr}(S\log S)\biggr]. \tag{7}$$

Essentially, it is based on a trade-off between fitting the model via the least squares method in the class of all density matrices and maximizing the entropy of the quantum state. Note that (7) is also a convex optimization problem (due to concavity of von Neumann entropy, see Nielsen and Chuang 2000) and its solution $\tilde{\rho}^{\varepsilon}$ is a full rank matrix (see Koltchinskii 2011a, the proof of Proposition 3). It should be also mentioned that the idea of estimation of a density matrix of a quantum state by maximizing the von Neumann entropy subject to constraints based on the data has been used in quantum state tomography earlier (see Bužek 2004 and references therein).

1.4 Distances between Density Matrices

The main purpose of this paper is to study the optimality properties of estimator $\tilde{\rho}^{\epsilon}$ with respect to a variety of statistically meaningful distances, in the case when the underlying density matrix ρ is low rank. These distances include Schatten p-norm distances for $p \in$

[1, 2],[1] but also quantum versions of Hellinger distance and Kullback-Leibler divergence that are of importance in quantum statistics and quantum information. A version of the (squared) Hellinger distance that will be studied is defined as

$$H^2(S_1, S_2) := 2 - 2\mathrm{tr}\sqrt{S_1^{\frac{1}{2}} S_2 S_1^{\frac{1}{2}}}$$

for $S_1, S_2 \in \mathcal{S}_m$ (see also Nielsen and Chuang 2000). Clearly, $0 \leq H^2(S_1, S_2) \leq 2$. In quantum information literature, it is usually called Bures distance and it does not coincide with $\mathrm{tr}(\sqrt{S_1} - \sqrt{S_2})^2$ (which is another possible non-commutative extension of the classical Hellinger distance). In fact, $H^2(S_1, S_2) \leq \mathrm{tr}(\sqrt{S_1} - \sqrt{S_2})^2, S_1, S_2 \in \mathcal{S}_m$, but the opposite inequality does not necessarily hold. The quantity $\mathrm{tr}\sqrt{S_1^{\frac{1}{2}} S_2 S_1^{\frac{1}{2}}}$ in the right hand side of the definition of H^2 is a quantum version of Hellinger affinity.

The noncommutative Kullback-Leibler divergence (or relative entropy distance) $K(\cdot\|\cdot)$ is defined as (see also Nielsen and Chuang 2000):

$$K(S_1\|S_2) := \langle S_1, \log S_1 - \log S_2\rangle.$$

If $\log S_2$ is not well-defined (for instance, some of the eigenvalues of S_2 are equal to 0) we set $K(S_1\|S_2) = +\infty$. The symmetrized version of Kullback-Leibler divergence is defined as

$$K(S_1; S_2) := K(S_1\|S_2) + K(S_2\|S_1) = \langle S_1 - S_2, \log S_1 - \log S_2\rangle.$$

The following very useful inequality is a noncommutative extension of similar classical inequalities for total variation, Hellinger and Kullback-Leibler distances. It follows from representing the "noncommutative distances" involved in the inequality as suprema of the corresponding classical distances between the distributions of outcomes of measurements for two states S_1, S_2 over all possible measurements represented by positive operator valued measures (see, Nielsen and Chuang 2000, Klauck et al. 2007, Koltchinskii 2011a, Section 3 and references therein).

Lemma 2 *For all $S_1, S_2 \in \mathcal{S}_m$, the following inequalities hold:*

$$\frac{1}{4}\|S_1 - S_2\|_1^2 \leq H^2(S_1, S_2) \leq \big(K(S_1\|S_2) \wedge \|S_1 - S_2\|_1\big). \tag{8}$$

1.5 Matrix Bernstein Inequalities

Non-commutative (matrix) versions of Bernstein inequality will be used in what follows. The most common version is stated (in a convenient form for our applications) in the following lemma.

Lemma 3 *Let $X, X_1, \dots, X_n \in \mathbb{H}_m$ be i.i.d. random matrices with $\mathbb{E}X = 0$, $\sigma_X^2 := \|\mathbb{E}X^2\|_\infty$ and $\|X\|_\infty \leq U$ a.s. for some $U > 0$. Then, for all $t \geq 0$ with probability at least $1 - e^{-t}$,*

$$\left\|\frac{1}{n}\sum_{j=1}^n X_j\right\|_\infty \leq 2\left[\sigma_X\sqrt{\frac{t + \log(2m)}{n}} \bigvee U\frac{t + \log(2m)}{n}\right].$$

1. Similar problems for estimators $\hat\rho, \check\rho$ and for Schatten p-norm distances with $p \in (2, +\infty]$ are studied in a related paper by Koltchinskii and Xia (2015+)

The proof of such bounds could be found, e.g., in Tropp (2012). Other versions on matrix Bernstein type inequalities for not necessarily bounded random matrices will be also used in what follows and they could be found in Koltchinskii (2011b), Koltchinskii (2013a). A simple consequence of the inequality of Lemma 3 is the following expectation bound:

$$\mathbb{E}\Big\|\frac{1}{n}\sum_{j=1}^{n} X_j\Big\|_{\infty} \lesssim \Big[\sigma_X\sqrt{\frac{\log(2m)}{n}} \bigvee U\frac{\log(2m)}{n}\Big].$$

It follows from the exponential bound by integrating the tail probabilities.

The paper is organized as follows. In Section 2, minimax lower bounds on estimation error of low rank density matrices are provided in Schatten p-norm, Hellinger (Bures) and Kullback-Leibler distances. In Section 3.1, sharp low rank oracle inequalities for von Neumann entropy penalized least squares estimator are derived in the case of trace regression model with bounded response. In Section 3.2, low rank oracle inequalities are established in the case of trace regression with Gaussian noise. In addition to this, in these two sections, upper bounds on estimation error with respect to Kullback-Leibler distance are obtained. In Section 3.3, they are further developed and extended to other distances (Hellinger distance, Schatten p-norm distances for $p \in [1,2]$) showing the minimax optimality (up to logarithmic factors) of the error rates of the least squares estimator with von Neumann entropy penalization.

2. Minimax Lower Bounds

In this section, we provide main results on the minimax lower bounds on the risk of estimation of density matrices with respect to Schatten p-norm (or, rather q-norm in the notations used below) distances as well as Hellinger-Bures distance and Kullback-Leibler divergence.

Minimax lower bounds will be derived for the class $\mathcal{S}_{r,m} := \{S \in \mathcal{S}_m : \mathrm{rank}(S) \le r\}$ consisting of all density matrices of rank at most r (the low rank case). We will start with the case of trace regression with Gaussian noise. Given that the sample $(X_1, Y_1), \dots, (X_n, Y_n)$ satisfies Assumption 4 with the target density matrix $\rho \in \mathcal{S}_m$ and noise variance σ_ξ^2, let $\mathbb{P}_\rho$ denote the corresponding probability distribution.

Note that Ma and Wu (2013) developed a method of deriving minimax lower bounds for distances based on unitary invariant norms, including Schatten p-norms in matrix problems, and obtained such lower bounds, in particular, in matrix completion problem. The approach used in our paper is somewhat different and the aim is to develop such bounds under an additional constraint that the target matrix is a density matrix. The resulting bounds are also somewhat different, they involve an additional term that does not depend on the rank, but does depend on q. Essentially, it means that the "complexity" of the problem is controlled by a "truncated rank" $r \wedge \frac{1}{\tau}$, where $\tau = \frac{\sigma_\xi m^{3/2}}{\sqrt{n}}$ rather than by the actual rank r. The upper bounds of Section 3.3 show that such a structure of the bound is, indeed, necessary. It should be also mentioned that minimax lower bounds on the nuclear norm error of estimation of density matrices have been obtained earlier in Flammia et al. (2012) (see Remark 11 below).

Theorem 4 *For all $q \in [1, +\infty]$, there exist constants $c, c' > 0$ such that, the following bounds hold:*

$$\inf_{\hat{\rho}} \sup_{\rho \in \mathcal{S}_{r,m}} \mathbb{P}_\rho \Big\{ \|\hat{\rho} - \rho\|_q \geq c \Big(\frac{\sigma_\xi m^{\frac{3}{2}} r^{1/q}}{\sqrt{n}} \bigwedge \Big(\frac{\sigma_\xi m^{3/2}}{\sqrt{n}} \Big)^{1-\frac{1}{q}} \bigwedge 1 \Big) \Big\} \geq c', \tag{9}$$

$$\inf_{\hat{\rho}} \sup_{\rho \in \mathcal{S}_{r,m}} \mathbb{P}_\rho \Big\{ H^2(\hat{\rho}, \rho) \geq c \Big(\frac{\sigma_\xi m^{\frac{3}{2}} r}{\sqrt{n}} \bigwedge 1 \Big) \Big\} \geq c', \tag{10}$$

and

$$\inf_{\hat{\rho}} \sup_{\rho \in \mathcal{S}_{r,m}} \mathbb{P}_\rho \Big\{ K(\rho \| \hat{\rho}) \geq c \Big(\frac{\sigma_\xi m^{\frac{3}{2}} r}{\sqrt{n}} \bigwedge 1 \Big) \Big\} \geq c', \tag{11}$$

where $\inf_{\hat{\rho}}$ denotes the infimum over all estimators $\hat{\rho}$ in $\mathcal{S}_m$ based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$ satisfying the Gaussian trace regression model with noise variance σ_ξ^2.

Proof A couple of preliminary facts will be needed in the proof. We start with bounds on the packing numbers of Grassmann manifold $\mathcal{G}_{k,l}$, which is the set of all k-dimensional subspaces L of the l-dimensional space $\mathbb{R}^l$. Given such a subspace $L \subset \mathbb{R}^l$ with $\dim(L) = k$, let P_L be the orthogonal projection onto L and let $\mathfrak{P}_{k,l} := \{P_L : L \in \mathcal{G}_{k,l}\}$. The set of all k-dimensional projectors $\mathfrak{P}_{k,l}$ will be equipped with Schatten q-norm distances for all $q \in [1, +\infty]$ (which also could be viewed as distances on the Grassmannian itself): $d_q(Q_1, Q_2) := \|Q_1 - Q_2\|_q, Q_1, Q_2 \in \mathfrak{P}_{k,l}$. Recall that the *$\varepsilon$-packing number* of a metric space (T, d) is defined as

$$D(T, d, \varepsilon) = \max \Big\{ n : \text{there are } t_1, \dots, t_n \in T, \text{such that } \min_{i \neq j} d(t_i, t_j) > \varepsilon \Big\}.$$

The following lemma (see Pajor 1998, Proposition 8) will be used to control the packing numbers of $\mathfrak{P}_{k,l}$ with respect to Schatten distances d_q.

Lemma 5 *For all integer $1 \leq k \leq l$ such that $k \leq l - k$, and all $1 \leq q \leq \infty$, the following bounds hold*

$$\Big(\frac{c}{\varepsilon} \Big)^d \leq D(\mathfrak{P}_{k,l}, d_q, \varepsilon k^{1/q}) \leq \Big(\frac{C}{\varepsilon} \Big)^d, \ \varepsilon > 0 \tag{12}$$

with $d = k(l-k)$ and universal positive constants c, C.

In addition to this, we need the following well known information-theoretic bound frequently used in derivation of minimax lower bounds (see Tsybakov 2008, Theorem 2.5). Let $\Theta = \{\theta_0, \theta_1, \dots, \theta_M\}$ be a finite parameter space equipped with a metric d and let $\mathcal{P} := \{\mathbb{P}_\theta : \theta \in \Theta\}$ be a family of probability distributions in some sample space. Given $\mathbb{P}, \mathbb{Q} \in \mathcal{P}$, let $K(\mathbb{P} \| \mathbb{Q}) := \mathbb{E}_{\mathbb{P}} \log \frac{d\mathbb{P}}{d\mathbb{Q}}$ be the Kullback-Leibler divergence between $\mathbb{P}$ and $\mathbb{Q}$.

Proposition 6 *Suppose that the following conditions hold:*

(i) for some $s > 0$, $d(\theta_j, \theta_k) \geq 2s > 0, 0 \leq j < k \leq M$;

(ii) for some $0 < \alpha < 1/8$, $\frac{1}{M} \sum_{j=1}^{M} K(\mathbb{P}_{\theta_j} \| \mathbb{P}_{\theta_0}) \leq \alpha \log M$

Then, for a positive constant c_α,

$$\inf_{\hat\theta}\sup_{\theta\in\Theta}\mathbb{P}_\theta\{d(\hat\theta,\theta)\ge s\}\ge c_\alpha,$$

where the infimum is taken over all estimators $\hat\theta\in\Theta$ *based on an observation sampled from* $\mathbb{P}_\theta$.

We now turn to the actual proof of Theorem 4. Under Assumption 4, the following computation is well known: for $\rho_1,\rho_2\in\mathcal{S}_{r,m}$,

$$\begin{aligned}K(\mathbb{P}_{\rho_1}\|\mathbb{P}_{\rho_2})&=\mathbb{E}_{\mathbb{P}_{\rho_1}}\log\frac{\mathbb{P}_{\rho_1}}{\mathbb{P}_{\rho_2}}\Big(X_1,Y_1,\dots,X_n,Y_n\Big)\\&=\mathbb{E}_{\mathbb{P}_{\rho_1}}\sum_{j=1}^n\Big[-\frac{(Y_j-\langle\rho_1,X_j\rangle)^2}{2\sigma_\xi^2}+\frac{(Y_j-\langle\rho_2,X_j\rangle)^2}{2\sigma_\xi^2}\Big]\\&=\mathbb{E}\sum_{j=1}^n\frac{\langle\rho_1-\rho_2,X_j\rangle^2}{2\sigma_\xi^2}=\frac{n}{2\sigma_\xi^2}\|\rho_1-\rho_2\|^2_{L_2(\Pi)}.\end{aligned}\tag{13}$$

It is enough to prove the bounds for $2\le r\le m/2$. The proof in the case $r=1$ is simpler and the case $r>m/2$ easily reduces to the case $r\le m/2$. We will use Lemma 5 to construct a well separated (with respect to d_q) subset of density matrices in $\mathcal{S}_{r,m}$. To this end, first choose a subset $\mathcal{D}_q\subset\mathfrak{P}_{r-1,m-1}$ such that $\mathrm{card}(\mathcal{D}_q)\ge 2^{(r-1)(m-r)}$ and, for some constant c', $\|Q_1-Q_2\|_q\ge c'(r-1)^{1/q}$, $Q_1,Q_2\in\mathfrak{P}_{r-1,m-1}, Q_1\ne Q_2$. Such a choice is possible due to the lower bound on the packing numbers of Lemma 5. For $Q\in\mathcal{D}_q$ (note that Q can be viewed as an $(m-1)\times(m-1)$ matrix with real entries) and $\kappa\in(0,1)$, consider the following $m\times m$ matrix

$$S=S_Q=\begin{pmatrix}1-\kappa & \mathbf{0}'\\ \mathbf{0} & \kappa\frac{Q}{r-1}\end{pmatrix}.\tag{14}$$

Note that S is symmetric positively-semidefinite real matrix of unit trace. It is straightforward to check that it defines a Hermitian positively-semidefinite operator in $\mathbb{C}^m$ of unit trace, and it can be identified with a density matrix $S\in\mathcal{S}_m$. Clearly, S is of rank r, so, $S\in\mathcal{S}_{r,m}$.

We will take $\kappa:=c_1\frac{\sigma_\xi m^{3/2}(r-1)}{\sqrt{n}}$ with a small enough absolute constant $c_1>0$ and first assume that $\kappa<1$ (as it is needed in definition Equation 14).

Let $\mathcal{S}'_q:=\{S_Q: Q\in\mathcal{D}_q\}$ and consider a family of $M+1=\mathrm{card}(\mathcal{D}_q)\ge 2^{(r-1)(m-r)}$ distributions $\{\mathbb{P}_S: S\in\mathcal{S}'_q\}$. It is immediate that for $S_1=S_{Q_1},S_2=S_{Q_2},Q_1,Q_2\in\mathcal{D}_q,Q_1\ne Q_2$, we have

$$\begin{aligned}\|S_1-S_2\|_q&=\frac{\kappa}{r-1}\|Q_1-Q_2\|_q\ge c'\kappa(r-1)^{1/q-1}\\&\ge c'c_1\frac{\sigma_\xi m^{3/2}(r-1)^{1/q}}{\sqrt{n}}\ge c\frac{\sigma_\xi m^{3/2}r^{1/q}}{\sqrt{n}}\end{aligned}\tag{15}$$

with some constant $c>0$, implying condition (i) of Proposition 6 with $s=\frac{c}{2}\frac{\sigma_\xi m^{3/2}r^{1/q}}{\sqrt{n}}$.

We will now check its condition (ii) . In view of (13), we have, for all $S_1 = S_{Q_1}, S_2 = S_{Q_2} \in \mathcal{S}'_q$,

$$\begin{aligned} K(\mathbb{P}_{S_1}\|\mathbb{P}_{S_2}) &= \frac{n}{2\sigma_\xi^2}\|S_1 - S_2\|^2_{L_2(\Pi)} = \frac{n}{2\sigma_\xi^2 m^2}\|S_1 - S_2\|_2^2 \\ &= \frac{n\kappa^2}{2\sigma_\xi^2 m^2(r-1)^2}\|Q_1 - Q_2\|_2^2 \le \frac{4n(r-1)\kappa^2}{2\sigma_\xi^2 m^2(r-1)^2} = 2c_1^2 m(r-1) \qquad (16) \\ &\le \alpha m(r-1)/\log(2)/4 \le \frac{\alpha}{2}(r-1)(m-r)\log(2) \le \alpha \log M, \end{aligned}$$

provided that constant c_1 is small enough, so, condition (ii) of Proposition 6 is also satisfied. Proposition 6 implies that, under the assumption $\kappa = c_1\frac{\sigma_\xi m^{3/2}(r-1)}{\sqrt{n}} < 1$, the following minimax lower bound holds for some $c, c' > 0$:

$$\inf_{\hat\rho}\sup_{\rho\in\mathcal{S}_{r,m}} \mathbb{P}_\rho\Big\{\|\hat\rho - \rho\|_q \ge c\frac{\sigma_\xi m^{\frac{3}{2}} r^{1/q}}{\sqrt{n}}\Big\} \ge c'. \qquad (17)$$

In the case when

$$c_1\frac{\sigma_\xi m^{3/2}}{\sqrt{n}} < 1 \le c_1\frac{\sigma_\xi m^{3/2}(r-1)}{\sqrt{n}},$$

one can choose $2 \le r' < r-1$ such that, for some constant $c_2 > 0$,

$$c_2 < c_1\frac{\sigma_\xi m^{3/2}(r'-1)}{\sqrt{n}} < 1.$$

For such a choice of r', it follows from (17) that

$$\inf_{\hat\rho}\sup_{\rho\in\mathcal{S}_{r',m}} \mathbb{P}_\rho\Big\{\|\hat\rho - \rho\|_q \ge c\frac{\sigma_\xi m^{\frac{3}{2}} (r')^{1/q}}{\sqrt{n}}\Big\} \ge c'. \qquad (18)$$

The definition of r' implies that

$$r' \asymp r' - 1 \asymp \left(\frac{\sigma_\xi m^{3/2}}{\sqrt{n}}\right)^{-1}.$$

Therefore,

$$\frac{\sigma_\xi m^{\frac{3}{2}}(r')^{1/q}}{\sqrt{n}} \asymp \left(\frac{\sigma_\xi m^{3/2}}{\sqrt{n}}\right)^{1-1/q},$$

and, since $\mathcal{S}_{r',m} \subset \mathcal{S}_{r,m}$, bound (18) yields

$$\inf_{\hat\rho}\sup_{\rho\in\mathcal{S}_{r,m}} \mathbb{P}_\rho\Big\{\|\hat\rho-\rho\|_q \ge c\Big(\frac{\sigma_\xi m^{3/2}}{\sqrt{n}}\Big)^{1-1/q}\Big\} \ge \inf_{\hat\rho}\sup_{\rho\in\mathcal{S}_{r',m}} \mathbb{P}_\rho\Big\{\|\hat\rho-\rho\|_q \ge c\Big(\frac{\sigma_\xi m^{3/2}}{\sqrt{n}}\Big)^{1-1/q}\Big\} \ge c' \qquad (19)$$

for some constants $c, c' > 0$. This allows us to recover the second term in the minimum in bound (9). Finally, in the case when $c_1\frac{\sigma_\xi m^{3/2}}{\sqrt{n}} > 1$, the minimax lower bound becomes a

constant (and the proof is based on a simplified version of the above argument that could be done for $r = 1$). This completes the proof of bound (9) for Schatten q-norms.

The proof of bound (10) for the Hellinger distance is similar. In the case $r \geq 2$, we will use a "well separated" set of density matrices $\mathcal{S}'_q \subset \mathcal{S}_{r,m}$ for $q = 1$ constructed above. We still use $\kappa := c_1 \frac{\sigma_\xi m^{3/2}(r-1)}{\sqrt{n}}$ assuming first that $\kappa \in (0,1)$. For $S_{Q_1}, S_{Q_2} \in \mathcal{S}'_q$ with $Q_1 \neq Q_2$, it follows by a simple computation and using bound (8) that, for some $c'' > 0$,

$$
\begin{aligned}
H^2(S_{Q_1}, S_{Q_2}) &= \kappa H^2\Big(\frac{Q_1}{r-1}, \frac{Q_2}{r-1}\Big) \\
&\geq \frac{1}{4}\frac{\kappa}{(r-1)^2}\|Q_1 - Q_2\|_1^2 \geq \frac{(c')^2}{4}\kappa \geq c''\frac{\sigma_\xi m^{3/2}(r-1)}{\sqrt{n}}.
\end{aligned}
$$

Repeating the argument based on Proposition 6 yields bound (10) in the case when $\kappa = c_1 \frac{\sigma_\xi m^{3/2}(r-1)}{\sqrt{n}} < 1$, and in the opposite case it is easy to see that the lower bound is a constant.

Finally, bound (11) for the Kullback–Leibler divergence follows from (10) and the inequality $K(\rho\|\hat{\rho}) \geq H^2(\hat{\rho}, \rho)$ (see inequality 8).

■

Next we state similar results in the case of trace regression model with bounded response (see Assumption 3). Denote by $\mathcal{P}_{r,m}(\bar{U})$ the class of all distributions P of (X, Y) such that Assumption 3 holds for some $\bar{U}$ and $\mathbb{E}(Y|X) = \langle \rho_P, X\rangle$ for some $\rho_P \in \mathcal{S}_{r,m}$. Given P, $\mathbb{P}_P$ denotes the corresponding probability measure (such that $(X_1, Y_1), \dots, (X_n, Y_n)$ are i.i.d. copies of (X, Y) sampled from P).

Theorem 7 *Suppose $\bar{U} \geq 2U$. For all $q \in [1, +\infty]$, there exist absolute constants $c, c' > 0$ such that the following bounds hold:*

$$
\inf_{\hat{\rho}} \sup_{P \in \mathcal{P}_{r,m}(\bar{U})} \mathbb{P}_P\Big\{\|\hat{\rho} - \rho_P\|_q \geq c\Big(\frac{\bar{U}m^{\frac{3}{2}}r^{1/q}}{\sqrt{n}} \bigwedge \Big(\frac{\bar{U}m^{3/2}}{\sqrt{n}}\Big)^{1-\frac{1}{q}} \bigwedge 1\Big)\Big\} \geq c', \tag{20}
$$

$$
\inf_{\hat{\rho}} \sup_{P \in \mathcal{P}_{r,m}(\bar{U})} \mathbb{P}_P\Big\{H^2(\hat{\rho}, \rho_P) \geq c\Big(\frac{\bar{U}m^{\frac{3}{2}}r}{\sqrt{n}} \bigwedge 1\Big)\Big\} \geq c', \tag{21}
$$

and

$$
\inf_{\hat{\rho}} \sup_{P \in \mathcal{P}_{r,m}(\bar{U})} \mathbb{P}_P\Big\{K(\rho_P\|\hat{\rho}) \geq c\Big(\frac{\bar{U}m^{\frac{3}{2}}r}{\sqrt{n}} \bigwedge 1\Big)\Big\} \geq c', \tag{22}
$$

where $\inf_{\hat{\rho}}$ denotes the infimum over all estimators $\hat{\rho}$ in $\mathcal{S}_m$ based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$.

Proof The proof relies on an idea already used in a context of matrix completion by Koltchinskii et al. (2011) (see their Theorem 7). We need the same family $\mathcal{S}'_q \subset \mathcal{S}_{r,m}$ of "well separated" density matrices of rank r as in the proof of Theorem 4. For a density matrix ρ, let (X, Y) be a random couple such that X is sampled from the uniform distribution Π in $\mathcal{E}$ and, conditionally on X, Y takes value $+\bar{U}$ with probability $p_\rho(X) := \frac{1}{2} + \frac{\langle \rho, X\rangle}{2\bar{U}}$ and value

$-\bar{U}$ with probability $q_\rho(X) := \frac{1}{2} - \frac{\langle \rho, X\rangle}{2\bar{U}}$. Since $\bar{U} \geq 2U$ and $|\langle \rho, X\rangle| \leq \|\rho\|_1 \|X\|_\infty \leq U$, we have $p_\rho(X), q_\rho(X) \in [1/4, 3/4]$ (so, they are bounded away from 0 and from 1). Clearly, $\mathbb{E}_\rho(Y|X) = \langle \rho, X\rangle$. Let P_ρ denote the distribution of such a couple and $\mathbb{P}_\rho$ denote the corresponding distribution of the data $(X_1, Y_1), \dots, (X_n, Y_n)$. Then, for all $\rho \in \mathcal{S}_{r,m}$, $P_\rho \in \mathcal{P}_{r,m}(\bar{U})$. The only difference with the proof of Theorem 4 is in the bound on Kullback-Leibler divergence $K(\mathbb{P}_{\rho_1} \| \mathbb{P}_{\rho_2})$ (see Equation 13). It is easy to see that

$$K(\mathbb{P}_{\rho_1} \| \mathbb{P}_{\rho_2}) = n\mathbb{E}\left(p_{\rho_1}(X) \log \frac{p_{\rho_1}(X)}{p_{\rho_2}(X)} + q_{\rho_1}(X) \log \frac{q_{\rho_1}(X)}{q_{\rho_2}(X)} \right). \tag{23}$$

The following simple inequality will be used: for all $a, b \in [1/4, 3/4]$,

$$a \log \frac{a}{b} + (1-a) \log \frac{1-a}{1-b} \leq 12(a-b)^2.$$

It implies that

$$K(\mathbb{P}_{\rho_1} \| \mathbb{P}_{\rho_2}) \leq 3n\mathbb{E} \frac{\langle \rho_1 - \rho_2, X\rangle^2}{\bar{U}^2} \leq \frac{3n}{\bar{U}^2} \|\rho_1 - \rho_2\|^2_{L_2(\Pi)}.$$

This bound is used instead of identity (13) from the proof of Theorem 4. The rest of the proof is the same. ■

Note that the proof requires the possible range $[-\bar{U}, \bar{U}]$ of response variable Y to be larger than the possible range $[-U, U]$ of Fourier coefficients $\langle \rho, E_j\rangle, j = 1, \dots, m^2$. This is not the case for standard QST model described in the introduction (see also the example of Pauli measurements) and it is of interest to prove a version of minimax lower bounds without this constraint, including the case when $\bar{U} = U$. The following theorem is a result in this direction.

Theorem 8 *Suppose Assumption 1 is satisfied and, moreover, for some constant $\gamma \in (0,1)$,*

$$\Big|\mathrm{tr}(E_k)\Big| \leq (1-\gamma)Um, \ k = 1, \dots, m^2. \tag{24}$$

Then, for all $q \in [1, +\infty]$, there exist constants $c_\gamma, c'_\gamma > 0$ such that the following bounds hold:

$$\inf_{\hat{\rho}} \sup_{P \in \mathcal{P}_{r,m}(U)} \mathbb{P}_P \Big\{ \|\hat{\rho} - \rho_P\|_q \geq c_\gamma \Big(\frac{Um^{\frac{3}{2}} r^{1/q}}{\sqrt{n}} \bigwedge \Big(\frac{Um^{3/2}}{\sqrt{n}} \Big)^{1-\frac{1}{q}} \bigwedge 1 \Big) \Big\} \geq c'_\gamma, \tag{25}$$

$$\inf_{\hat{\rho}} \sup_{P \in \mathcal{P}_{r,m}(U)} \mathbb{P}_P \Big\{ H^2(\hat{\rho}, \rho_P) \geq c_\gamma \Big(\frac{Um^{\frac{3}{2}} r}{\sqrt{n}} \bigwedge 1 \Big) \Big\} \geq c'_\gamma, \tag{26}$$

and

$$\inf_{\hat{\rho}} \sup_{P \in \mathcal{P}_{r,m}(U)} \mathbb{P}_P \Big\{ K(\rho_P \| \hat{\rho}) \geq c_\gamma \Big(\frac{Um^{\frac{3}{2}} r}{\sqrt{n}} \bigwedge 1 \Big) \Big\} \geq c'_\gamma, \tag{27}$$

where $\inf_{\hat{\rho}}$ denotes the infimum over all estimators $\hat{\rho}$ in $\mathcal{S}_m$ based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$.

Proof The proof is based on the following lemma:

Lemma 9 *Suppose assumption (24) holds. Let K be a sufficiently large absolute constant (to be chosen later) and let m satisfy the condition $K\frac{\log m}{\sqrt{m}} \le \frac{\gamma}{2}$ (which means that $m \ge A_\gamma$ for some constant A_γ). Then there exists $v \in \mathbb{C}^m$ with $\|v\| = 1$ such that*

$$\Big|\langle E_k v, v\rangle\Big| \le (1-\gamma/2)U, k = 1, \dots, m^2. \tag{28}$$

Proof We will prove this fact by a probabilistic argument. Namely, set $v := m^{-1/2}(\varepsilon_1, \dots, \varepsilon_m)$, where $\varepsilon_j = \pm 1$. We will show that there is a random choice of "signs" ε_j such that (28) holds. Assume that $\varepsilon_j, j = 1, \dots, m$ are i.i.d. and take values ± 1 with probability $1/2$ each. Let $E_k := (a_{ij}^{(k)})_{i,j=1,\dots,m}$. For simplicity, assume that $(a_{ij}^{(k)})_{i,j=1,\dots,m}$ is a symmetric real matrix (in the complex case, the proof can be easily modified). We have

$$\langle E_k v, v\rangle = \frac{1}{m}\sum_{i=1}^m a_{ii}^{(k)}\varepsilon_i^2 + \frac{1}{m}\sum_{i\neq j} a_{ij}^{(k)}\varepsilon_i\varepsilon_j = \frac{\mathrm{tr}(E_k)}{m} + \frac{1}{m}\sum_{i\neq j} a_{ij}^{(k)}\varepsilon_i\varepsilon_j.$$

It is well known that

$$\mathrm{Var}\Big(\sum_{i\neq j} a_{ij}^{(k)}\varepsilon_i\varepsilon_j\Big) = \mathbb{E}\Big(\sum_{i\neq j} a_{ij}^{(k)}\varepsilon_i\varepsilon_j\Big)^2 = 2\sum_{i\neq j}\Big(a_{ij}^{(k)}\Big)^2 \le 2\sum_{i,j}\Big(a_{ij}^{(k)}\Big)^2 = 2\|E_k\|_2^2 = 2.$$

Moreover, it follows from exponential inequalities for Rademacher chaos (see, e.g., Corollary 3.2.6 in de la Peña and Giné 1999) that for some absolute constant $K > 0$ and for all $t > 0$, with probability at least $1 - e^{-t}$

$$\Big|\langle E_k v, v\rangle - \frac{\mathrm{tr}(E_k)}{m}\Big| = \Big|\frac{1}{m}\sum_{i\neq j} a_{ij}^{(k)}\varepsilon_i\varepsilon_j\Big| \le \frac{Kt}{m}.$$

Taking $t = 2\log m$ and using the union bound, we conclude that with probability at least $1 - me^{-2\log m} = 1 - \frac{1}{m} > 0$,

$$\max_{1\le k\le m^2}\Big|\langle E_k v, v\rangle - \frac{\mathrm{tr}(E_k)}{m}\Big| \le \frac{K\log m}{m} \le \frac{K\log m}{\sqrt{m}}U \le \frac{\gamma}{2}U,$$

where we also used the fact that $U \ge m^{-1/2}$. Thus, there exists a choice of signs ε_j such that

$$\max_{1\le k\le m^2}\Big|\langle E_k v, v\rangle\Big| \le \max_{1\le k\le m}\Big|\frac{\mathrm{tr}(E_k)}{m}\Big| + \frac{\gamma}{2}U,$$

which, under condition (24), implies (28). ∎

We set $e_1 := v$ (where v is the unit vector introduced in Lemma 9) and construct an orthonormal basis $e_1, \dots, e_m$. Assume that matrices S_Q defined by (14) represent linear transformations in basis $e_1, \dots, e_m$. Then we have

$$\langle S_Q, E_k\rangle = (1-\kappa)\langle E_k v, v\rangle + \frac{\kappa}{r-1}\langle Q, E_k\rangle.$$

Therefore,

$$\Big|\langle S_Q, E_k\rangle\Big| \le (1-\kappa)\Big|\langle E_k v, v\rangle\Big| + \frac{\kappa}{r-1}\|E_k\|_\infty \|Q\|_1 \le (1-\kappa)(1-\gamma/2)U + \kappa U = (1-(1-\kappa)(\gamma/2))U.$$

Assuming that $\kappa \le 1/2$, we get

$$\Big|\langle S_Q, E_k\rangle\Big| \le (1-\gamma/4)U,\ k = 1, \dots, m^2. \tag{29}$$

The rest of the proof becomes similar to the proof of Theorem 7 (with $\bar{U} = U$). Namely, bound (29) implies that, for $\rho = S_Q$ and X being sampled from the orthonormal basis $\{E_1, \dots, E_{m^2}\}$, probabilities $p_\rho(X)$ and $q_\rho(X)$ are bounded away from 0 and from 1 : $p_\rho(X), q_\rho(X) \in [\gamma/8, 1-\gamma/8]$. This allows us to complete the argument of the proof of Theorem 7. ∎

Theorem 8 does not apply directly to the Pauli basis since condition (24) fails in this case. Indeed, by the definition of Pauli basis, $U = m^{-1/2}$ and $\mathrm{tr}(E_1) = \sqrt{m} = Um > (1-\gamma)Um$. Note also that $\mathrm{tr}(E_j) = 0, j = 2, \dots, m^2$. Thus, for Pauli basis, E_1 is the only matrix for which condition (24) fails. However, for this matrix $\langle \rho, E_1\rangle = m^{-1/2}\mathrm{tr}(\rho) = m^{-1/2} = U$ for all density matrices $\rho \in \mathcal{S}_m$. This immediately implies that $p_\rho(E_1) = 1$ and $q_\rho(E_1) = 0$ for all $\rho \in \mathcal{S}_m$ and, as a result, the value $X = E_1$ does not have an impact on the computation of Kullback-Leibler divergence in (23). For the rest of the matrices in the Pauli basis, condition (24) holds implying also bound (28). Therefore, if $X \neq E_1$, we still have that, for $\rho = S_Q$, $p_\rho(X), q_\rho(X) \in [\gamma/8, 1-\gamma/8]$, and the proof of Theorem 7 can be completed in this case, too. Note also that, given X sampled from the Pauli basis, the binary random variable Y taking values $\pm U = \pm\frac{1}{\sqrt{m}}$ with probabilities $p_\rho(X)$ and $q_\rho(X)$, respectively (this is exactly the random variable used in the construction of the proof of Theorem 7) coincides with an outcome of a Pauli measurement for the system prepared in state ρ. These considerations yield the following minimax lower bounds for Pauli measurements.

Theorem 10 *Let $\{E_1, \dots, E_{m^2}\}$ be the Pauli basis in the space $\mathbb{H}_m$ of $m \times m$ Hermitian matrices and let $X_1, \dots, X_n$ be i.i.d. random variables sampled from the uniform distribution in $\{E_1, \dots, E_{m^2}\}$. Let $Y_1, \dots, Y_n$ be outcomes of measurements of observables $X_1, \dots, X_n$ for the system being identically prepared n times in state ρ. The corresponding distribution of the data $(X_1, Y_1), \dots, (X_n, Y_n)$ will be denoted by $\mathbb{P}_\rho$. Then, for all $q \in [1, +\infty]$, there exist constants $c, c' > 0$ such that the following bounds hold:*

$$\inf_{\hat\rho} \sup_{\rho \in \mathcal{S}_{r,m}} \mathbb{P}_\rho\Big\{\|\hat\rho - \rho\|_q \ge c\Big(\frac{mr^{1/q}}{\sqrt{n}} \bigwedge \Big(\frac{m}{\sqrt{n}}\Big)^{1-\frac{1}{q}} \bigwedge 1\Big)\Big\} \ge c', \tag{30}$$

$$\inf_{\hat\rho} \sup_{\rho \in \mathcal{S}_{r,m}} \mathbb{P}_\rho\Big\{H^2(\hat\rho, \rho) \ge c\Big(\frac{mr}{\sqrt{n}} \bigwedge 1\Big)\Big\} \ge c', \tag{31}$$

and

$$\inf_{\hat\rho} \sup_{\rho \in \mathcal{S}_{r,m}} \mathbb{P}_\rho\Big\{K(\rho\|\hat\rho) \ge c\Big(\frac{mr}{\sqrt{n}} \bigwedge 1\Big)\Big\} \ge c', \tag{32}$$

where $\inf_{\hat\rho}$ denotes the infimum over all estimators $\hat\rho$ in $\mathcal{S}_m$ based on the data $(X_1, Y_1), \dots, (X_n, Y_n)$.

Remark 11 *Minimax lower bounds on nuclear norm error of density matrix estimation close to bound (30) for $q = 1$ (but for a somewhat different "estimation protocol" and stated in a different form) were obtained earlier in Flammia et al. (2012). This paper also contains upper bounds on the errors of matrix LASSO and Dantzig selector estimators in the nuclear norm matching the lower bounds up to log-factors.*

Remark 12 *It is easy to see that, if constant $\gamma \in (0,1)$ is small enough (namely, $\gamma < 1 - \frac{1}{\sqrt{2}}$), then, in an arbitrary orthonormal basis $\{E_1, \dots, E_{m^2}\}$, there is at most one matrix E_j such that $|\mathrm{tr}(E_j)| > (1-\gamma)Um$. Indeed, note that $\mathrm{tr}(E_j) = \langle E_j, I_m \rangle$. Since*

$$\sum_{j=1}^{m^2} \langle E_j, I_m \rangle^2 = \|I_m\|_2^2 = m$$

and $U^2 m \geq 1$, we have

$$\mathrm{card}\Big(\Big\{j : |\langle E_j, I_m\rangle| > (1-\gamma)Um\Big\}\Big) \leq \frac{1}{(1-\gamma)^2 U^2 m^2} \sum_{j=1}^{m^2} \langle E_j, I_m \rangle^2$$

$$\leq \frac{m}{(1-\gamma)^2 U^2 m^2} = \frac{1}{(1-\gamma)^2 U^2 m} \leq \frac{1}{(1-\gamma)^2} < 2,$$

provided that $\gamma < 1 - \frac{1}{\sqrt{2}}$.

Remark 13 *It will be shown in Section 3.3 that the minimax rates of theorems 4, 7, 8 and 10 are attained up to logarithmic factors for the von Neumann entropy penalized least squares estimator.*

Remark 14 *Similar minimax lower bounds could be proved in certain classes of "nearly low rank" density matrices. Consider, for instance, the following class*

$$B_p(d; m) := \Big\{ S \in \mathcal{S}_m : \sum_{j=1}^{m} |\lambda_j(S)|^p \leq d \Big\} \tag{33}$$

for some $d > 0$ and $p \in [0,1]$, where $\lambda_1(S) \geq \cdots \geq \lambda_m(S)$ denote the eigenvalues of S. This set consists of density matrices with the eigenvalues decaying at a certain rate (nearly low rank case) and, for $p = 0$, $d = r$ it coincides with $\mathcal{S}_{r,m}$. It turns out that minimax lower bounds of theorems 4 and 7 hold for the class $B_p(d; m)$ (instead of $\mathcal{S}_{r,m}$) with r replaced by

$$\bar{r} := \bar{r}(\tau, d, m, p) = d\tau^{-p} \wedge m,$$

where $\tau := \frac{\sigma_\xi m^{3/2}}{\sqrt{n}}$ in the case of trace regression with Gaussian noise and $\tau := \frac{\bar{U} m^{3/2}}{\sqrt{n}}$ in the case of trace regression with bounded response. These minimax bounds are attained up to logarithmic factors for a slightly modified von Neumann entropy penalized least squares estimator.

Note that, for $\rho \in B_p(d, m)$ with eigenvalues $\lambda_1(\rho) \geq \cdots \geq \lambda_m(\rho)$, we have $\lambda_j(\rho) \leq \frac{d^{1/p}}{j^{1/p}}, j = 1, \dots, m$. Therefore, for $j \geq \bar{r}$, $\lambda_j(\rho) \leq \tau$. Note also that τ characterizes the

minimax rate of estimation of $\rho \in \mathcal{S}_{r,m}$ in the operator norm for any value of the rank r (see bound (9) for $q = +\infty$; the corresponding upper bound also holds for the least squares estimator up to a logarithmic factor, see Koltchinskii and Xia 2015+). Roughly speaking, τ is a threshold below which the estimation of eigenvalues $\lambda_j(\rho)$ becomes impossible and $\bar{r}$ can be viewed as an "effective rank" of nearly low rank density matrices in the class $B_p(d, m)$.

3. Von Neumann Entropy Penalization: Optimality and Oracle Inequalities

The goal of this section is to study optimality properties of von Neumann entropy penalized least squares estimator $\tilde{\rho}^{\varepsilon}$ defined by (7). In particular, we establish oracle inequalities for such estimators in the cases of trace regression with bounded response (Subsection 3.1) and trace regression with Gaussian noise (Subsection 3.2), and prove upper bounds on their estimation errors measured by Schatten q-norm distances for $q \in [1, 2]$ and also by Hellinger and Kullback-Leibler distances (Subsection 3.3).

3.1 Oracle Inequalities for Trace Regression with Bounded Response

In this subsection, we prove a *sharp low rank oracle inequality* for estimator $\tilde{\rho}^{\varepsilon}$ defined by (7). It is done in the case of trace regression model with bounded response (that is, under Assumption 3). The results of this type show some form of optimality of the estimation method, namely, that the estimator provides an optimal trade-off between the "approximation error" of the target density matrix by a low rank "oracle" and the "estimation error" of the "oracle" that is proportional to its rank. Sharp oracle inequalities (in which the leading constant in front of the "approximation error" is equal to 1, so that the bound mimics precisely the approximation by the oracle) are usually harder to prove. In the case of low rank matrix completion, the first result of this type was proved by Koltchinskii et al. (2011) for a modified least squares estimator with nuclear norm penalty. A version of such inequality for empirical risk minimization with nuclear norm penalty (that includes matrix LASSO) was first proved by Koltchinskii (2013b). Low rank oracle inequalities for von Neumann entropy penalized least squares method with the leading constant larger than 1 were proved by Koltchinskii (2011a). The main result of this section refines these previous bounds by proving a sharp oracle inequality, improving the logarithmic factors and removing superfluous assumptions, but also by establishing the inequality in the whole range of values of regularization parameter $\varepsilon \geq 0$ (including the value $\varepsilon = 0$, for which $\tilde{\rho}^{\varepsilon}$ coincides with the least squares estimator $\hat{\rho}$). In addition to this, for a special choice of regularization parameter ε, the theorem below also provides an upper bound on the Kullback-Leibler error $K(\rho \| \tilde{\rho}^{\varepsilon})$ of $\tilde{\rho}^{\varepsilon}$ that matches the minimax lower bound (22) up to log-factors (and "second order terms"). It turns out that, for this choice of ε, the estimator satisfies exactly the same low rank oracle inequality as the best inequalities known for LASSO estimator and minimax optimal error rates are attained for $\tilde{\rho}^{\varepsilon}$ also with respect to Hellinger distance and Schatten q-norm distances for all $q \in [1, 2]$ (see Section 3.3). For simplicity, it will be assumed that constants U in Assumption 1 and $\bar{U}$ in Assumption 3 coincide (in the upper bounds, one can always replace U and $\bar{U}$ by $U \vee \bar{U}$).

Theorem 15 *Suppose Assumption 3 holds with constant $\bar{U} = U$ and let $\varepsilon \in [0,1]$. Then, there exists a constant $C > 0$ such that for all $t \geq 1$ with probability at least $1 - e^{-t}$*

$$\|f_{\tilde{\rho}^\varepsilon} - f_\rho\|^2_{L_2(\Pi)} \leq \inf_{S \in \mathcal{S}_m} \Big[\|f_S - f_\rho\|^2_{L_2(\Pi)} + C \Big(\mathrm{rank}(S) m^2 \varepsilon^2 \log^2(mn) + U^2 \frac{\mathrm{rank}(S) m \log(2m)}{n} + U^2 \frac{t + \log\log_2(2n)}{n} \Big) \Big]. \tag{34}$$

In particular, this implies that

$$\|f_{\tilde{\rho}^\varepsilon} - f_\rho\|^2_{L_2(\Pi)} \leq C \Big[\mathrm{rank}(\rho) m^2 \varepsilon^2 \log^2(mn) + U^2 \frac{\mathrm{rank}(\rho) m \log(2m)}{n} + U^2 \frac{t + \log\log_2(2n)}{n} \Big]. \tag{35}$$

Moreover, if

$$\varepsilon := \frac{1}{\log(mn)} \Big[U \sqrt{\frac{\log(2m)}{nm}} \bigvee U^2 \frac{\log(2m)}{n} \Big],$$

then, with some constant C and with probability at least $1 - e^{-t}$

$$\|f_{\tilde{\rho}^\varepsilon} - f_\rho\|^2_{L_2(\Pi)} \leq C \Big[U^2 \frac{\mathrm{rank}(\rho) m \log(2m)}{n} \Big(1 \bigvee U^2 \frac{m \log(2m)}{n} \Big) + U^2 \frac{t + \log\log_2(2n)}{n} \Big] \tag{36}$$

and

$$K(\rho \| \tilde{\rho}^\varepsilon) \leq CU \Big[\frac{\mathrm{rank}(\rho) m^{3/2} \sqrt{\log(2m)} \log(mn)}{\sqrt{n}} \Big(1 \bigvee U \sqrt{\frac{m \log(2m)}{n}} \Big) + \sqrt{\frac{m}{n}} \frac{(t + \log\log_2(2n)) \log(mn)}{\sqrt{\log(2m)}} \Big]. \tag{37}$$

Proof The following notations will be used in the proof. Let $\ell(y,u) := (u-y)^2, y, u \in \mathbb{R}$ be the quadratic loss function. For $f : \mathbb{H}_m \mapsto \mathbb{R}$, denote

$$(\ell \bullet f)(x,y) = (f(x) - y)^2, \quad (\ell' \bullet f)(x,y) = 2(f(x) - y)$$

and

$$P(\ell \bullet f) = \mathbb{E}(Y - f(X))^2, \quad P_n(\ell \bullet f) = n^{-1} \sum_{j=1}^n (Y_j - f(X_j))^2.$$

For $A \in \mathbb{H}_m$, let $f_A(x) = \langle A, x \rangle, x \in \mathbb{H}_m$. Since for density matrices $S \in \mathcal{S}_m$, $\|S\|_1 = \mathrm{tr}(S) = 1$, the estimator $\tilde{\rho} = \tilde{\rho}^\varepsilon$ can be equivalently defined by the following convex optimization problem:

$$\tilde{\rho} = \mathrm{argmin}_{S \in \mathcal{S}_m} L_n(S), \quad L_n(S) := \Big[P_n(\ell \bullet f_S) + \varepsilon \mathrm{tr}(S \log S) + \bar{\varepsilon} \|S\|_1 \Big]$$

for an arbitrary $\bar{\varepsilon} > 0$.

The following lemma will be crucial in the proofs of Theorem 15 as well Theorem 19 in the following subsection. Note that it does not rely on Assumption 3, only Assumptions 1 and 2 are needed.

Lemma 16 *Suppose Assumptions 1 and 2 hold. Let $\delta \in (0,1)$ and $S := (1-\delta)S' + \delta\frac{I_m}{m}$, where $S' \in \mathcal{S}_m$, $\mathrm{rank}(S') = r$ and I_m is the $m \times m$ identity matrix. Then the following bound holds:*

$$\begin{aligned}\|f_{\tilde\rho} - f_\rho\|^2_{L_2(\Pi)} + \tfrac{1}{2}\|f_{\tilde\rho} - f_S\|^2_{L_2(\Pi)} + \varepsilon K(\tilde\rho; S) + \bar\varepsilon\left\|\mathcal{P}_L^\perp(\tilde\rho)\right\|_1 \\ \le \|f_S - f_\rho\|^2_{L_2(\Pi)} + rm^2\varepsilon^2\log^2(m/\delta) + rm^2\bar\varepsilon^2 \\ +4\bar\varepsilon\delta + (P - P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S).\end{aligned} \tag{38}$$

Lemma 16 will be often used together with the following simple bound:

$$\begin{aligned}\|f_S - f_\rho\|^2_{L_2(\Pi)} = \tfrac{1}{m^2}\|S - \rho\|_2^2 \le \\ \tfrac{1}{m^2}\|S' - \rho\|_2^2 + \tfrac{2}{m^2}\|S' - \rho\|_2\|S' - S\|_2 + \tfrac{1}{m^2}\|S' - S\|_2^2 \\ \le \|f_{S'} - f_\rho\|^2_{L_2(\Pi)} + \tfrac{8\delta}{m^2} + \tfrac{4\delta^2}{m^2} \le \|f_{S'} - f_\rho\|^2_{L_2(\Pi)} + \tfrac{12\delta}{m^2}.\end{aligned} \tag{39}$$

Together, they imply that

$$\begin{aligned}\|f_{\tilde\rho} - f_\rho\|^2_{L_2(\Pi)} + \tfrac{1}{2}\|f_{\tilde\rho} - f_S\|^2_{L_2(\Pi)} + \varepsilon K(\tilde\rho; S) + \bar\varepsilon\left\|\mathcal{P}_L^\perp(\tilde\rho)\right\|_1 \\ \le \|f_{S'} - f_\rho\|^2_{L_2(\Pi)} + rm^2\varepsilon^2\log^2(m/\delta) + rm^2\bar\varepsilon^2 \\ +4\bar\varepsilon\delta + \tfrac{12\delta}{m^2} + (P - P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S).\end{aligned} \tag{40}$$

We will now give the proof of Lemma 16.

Proof By standard necessary conditions of extremum in convex problems, we get that, for all $S \in \mathcal{S}_m$ and for some $\tilde V \in \partial\|\tilde\rho\|_1$,

$$P_n(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S) + \varepsilon\langle\log\tilde\rho, \tilde\rho - S\rangle + \bar\varepsilon\langle\tilde V, \tilde\rho - S\rangle \le 0$$

(see, e.g., Aubin and Ekeland 2006, Chapter 2, Corollary 6; see also Koltchinskii 2011b, pp. 198–199; for the computation of derivative of the function $\mathrm{tr}(S\log S)$, see Lemma 1 in Koltchinskii 2011a). Replacing in the left hand side P by P_n, we get

$$P(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S) + \varepsilon\langle\log\tilde\rho, \tilde\rho - S\rangle + \bar\varepsilon\langle\tilde V, \tilde\rho - S\rangle \le (P - P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S).$$

It is easy to check that for the quadratic loss

$$P(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S) = P(\ell \bullet f_{\tilde\rho}) - P(\ell \bullet f_S) + \|f_{\tilde\rho} - f_S\|^2_{L_2(\Pi)},$$

implying that

$$\begin{aligned}P(\ell \bullet f_{\tilde\rho}) - P(\ell \bullet f_S) + \|f_{\tilde\rho} - f_S\|^2_{L_2(\Pi)} + \varepsilon\langle\log\tilde\rho, \tilde\rho - S\rangle + \bar\varepsilon\langle\tilde V, \tilde\rho - S\rangle \\ \le (P - P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S).\end{aligned}$$

Also, for the quadratic loss,

$$P(\ell \bullet f) - P(\ell \bullet f_\rho) = \|f - f_\rho\|^2_{L_2(\Pi)}.$$

Therefore,

$$\|f_{\tilde\rho}-f_\rho\|^2_{L_2(\Pi)}+\|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)}+\varepsilon\langle\log\tilde\rho,\tilde\rho-S\rangle+\bar\varepsilon\langle\tilde V,\tilde\rho-S\rangle$$
$$\le\|f_S-f_\rho\|^2_{L_2(\Pi)}+(P-P_n)(\ell'\bullet f_{\tilde\rho})(f_{\tilde\rho}-f_S).$$

Recall that we have set $S=(1-\delta)S'+\delta\frac{I_m}{m}$, where $S'\in\mathcal{S}_m$, $\mathrm{rank}(S')=r$, $\delta\in(0,1)$. Clearly,

$$\Big|\langle\tilde V,S-S'\rangle\Big|\le\|\tilde V\|_\infty\|S-S'\|_1\le\|S-S'\|_1=\delta\Big\|S'-\frac{I_m}{m}\Big\|_1\le 2\delta,$$

where we used the fact that $\|\tilde V\|_\infty\le 1$ for $\tilde V\in\partial\|\tilde\rho\|_1$. This implies

$$\begin{aligned}\|f_{\tilde\rho}-f_\rho\|^2_{L_2(\Pi)}+\|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)}+\varepsilon\langle\log\tilde\rho,\tilde\rho-S\rangle+\bar\varepsilon\langle\tilde V,\tilde\rho-S'\rangle\\ \le\|f_S-f_\rho\|^2_{L_2(\Pi)}+2\bar\varepsilon\delta+(P-P_n)(\ell'\bullet f_{\tilde\rho})(f_{\tilde\rho}-f_S).\end{aligned}\tag{41}$$

Recall formula (2) for the subdifferential of nuclear norm. Let $L=\mathrm{supp}(S')$. By the duality between the operator and nuclear norms, there exists $M\in\mathbb{H}_m$ with $\|M\|_\infty\le 1$ such that

$$\langle\mathcal{P}_L^\perp(M),\tilde\rho-S'\rangle=\langle M,\mathcal{P}_L^\perp(\tilde\rho-S')\rangle=\Big\|\mathcal{P}_L^\perp(\tilde\rho-S')\Big\|_1=\Big\|\mathcal{P}_L^\perp(\tilde\rho)\Big\|_1.$$

With $V=\mathrm{sign}(S')+\mathcal{P}_L^\perp(M)\in\partial\|S'\|_1$, by monotonicity of subdifferential, we get that

$$\langle\mathrm{sign}(S'),\tilde\rho-S'\rangle+\Big\|\mathcal{P}_L^\perp(\tilde\rho)\Big\|_1=\langle V,\tilde\rho-S'\rangle\le\langle\tilde V,\tilde\rho-S'\rangle.\tag{42}$$

In addition to this, we have

$$\langle\log\tilde\rho,\tilde\rho-S\rangle=\langle\log\tilde\rho-\log S,\tilde\rho-S\rangle+\langle\log S,\tilde\rho-S\rangle=K(\tilde\rho;S)+\langle\log S,\tilde\rho-S\rangle.\tag{43}$$

Substituting (42) and (43) into (41), we get

$$\begin{aligned}\|f_{\tilde\rho}-f_\rho\|^2_{L_2(\Pi)}+\|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)}+\varepsilon K(\tilde\rho;S)+\bar\varepsilon\Big\|\mathcal{P}_L^\perp(\tilde\rho)\Big\|_1\\ \le\|f_S-f_\rho\|^2_{L_2(\Pi)}+\varepsilon\langle\log S,S-\tilde\rho\rangle+\bar\varepsilon\langle\mathrm{sign}(S'),S'-\tilde\rho\rangle\\ +2\bar\varepsilon\delta+(P-P_n)(\ell'\bullet f_{\tilde\rho})(f_{\tilde\rho}-f_S).\end{aligned}\tag{44}$$

The following bound on $\bar\varepsilon\langle\mathrm{sign}(S'),S'-\tilde\rho\rangle$ is straightforward:

$$\begin{aligned}\bar\varepsilon\langle\mathrm{sign}(S'),S'-\tilde\rho\rangle\le\bar\varepsilon\langle\mathrm{sign}(S'),S-\tilde\rho\rangle+\bar\varepsilon\|\mathrm{sign}(S')\|_\infty\|S-S'\|_1\\ \le\bar\varepsilon\|\mathrm{sign}(S')\|_2\|S-\tilde\rho\|_2+2\bar\varepsilon\delta\le\bar\varepsilon\sqrt{r}m\|f_S-f_{\tilde\rho}\|_{L_2(\Pi)}+2\bar\varepsilon\delta\\ \le rm^2\bar\varepsilon^2+\tfrac14\|f_S-f_{\tilde\rho}\|^2_{L_2(\Pi)}+2\bar\varepsilon\delta.\end{aligned}\tag{45}$$

A similar bound on $\varepsilon\langle\log S,S-\tilde\rho\rangle$ is only slightly more complicated. Suppose S' has the following spectral representation: $S'=\sum_{k=1}^r\lambda_kP_k$ with eigenvalues $\lambda_k\in(0,1]$ (repeated with their multiplicities) and one-dimensional orthogonal eigenprojectors P_k. We will extend $P_j,j=1,\dots,r$ to the complete orthogonal resolution of the identity $P_j,j=1,\dots,m$. Then

$$\log S=\log\Big((1-\delta)S'+\delta\frac{I_m}{m}\Big)=\sum_{j=1}^r\log\Big((1-\delta)\lambda_j+\delta/m\Big)P_j+\sum_{j=r+1}^m\log(\delta/m)P_j$$

$$= \sum_{j=1}^{r} \log\Big(1 + (1-\delta)m\lambda_j/\delta\Big) P_j + \log(\delta/m) I_m$$

and

$$\langle \log S, S - \tilde{\rho}\rangle = \Big\langle \sum_{j=1}^{r} \log\Big(1 + (1-\delta)m\lambda_j/\delta\Big) P_j, S - \tilde{\rho}\Big\rangle + \log(\delta/m)\langle I_m, S - \tilde{\rho}\rangle$$

$$= \Big\langle \sum_{j=1}^{r} \log\Big(1 + (1-\delta)m\lambda_j/\delta\Big) P_j, S - \tilde{\rho}\Big\rangle$$

where we used the fact that $\langle I_m, S - \tilde{\rho}\rangle = \mathrm{tr}(S) - \mathrm{tr}(\tilde{\rho}) = 0$. Therefore,

$$\varepsilon\langle \log S, S - \tilde{\rho}\rangle \le \varepsilon \Big\| \textstyle\sum_{j=1}^{r} \log\Big(1 + (1-\delta)m\lambda_j/\delta\Big) P_j \Big\|_2 \, \|S - \tilde{\rho}\|_2 \tag{46}$$

$$= \varepsilon m \Big(\textstyle\sum_{j=1}^{r} \log^2\Big(1 + (1-\delta)m\lambda_j/\delta\Big)\Big)^{1/2} \|f_S - f_{\tilde{\rho}}\|_{L_2(\Pi)}$$

$$\le \varepsilon\sqrt{r} m \log(m/\delta)\|f_S - f_{\tilde{\rho}}\|_{L_2(\Pi)} \le rm^2\varepsilon^2 \log^2(m/\delta) + \tfrac{1}{4}\|f_S - f_{\tilde{\rho}}\|^2_{L_2(\Pi)},$$

where it was used that for $\lambda_j \in [0,1]$

$$\log\Big(1 + (1-\delta)m\lambda_j/\delta\Big) \le \log\Big(\frac{\delta + (1-\delta)m}{\delta}\Big) \le \log(m/\delta).$$

Substituting bounds (45) and (46) in (44) we easily get bound (38), as claimed in the lemma. ■

We will also need the following simple lemma that provides a bound on $K(S'\|\tilde{\rho})$ in terms of $K(S\|\tilde{\rho})$.

Let

$$h(\delta) := \delta \log\frac{1}{\delta} + (1-\delta)\log\frac{1}{1-\delta}.$$

Observe that

$$h(\delta) = \delta\log\frac{1}{\delta} + (1-\delta)\log\Big(1 + \frac{\delta}{1-\delta}\Big) \le \delta\log\frac{1}{\delta} + (1-\delta)\frac{\delta}{1-\delta} \le \delta\log\frac{e}{\delta}$$

(this bound will be used in what follows).

Lemma 17 *Let $\delta \in (0,1)$, $S' \in \mathcal{S}_m$ with $\mathrm{rank}(S') = r$ and $S = (1-\delta)S' + \delta\frac{I_m}{m}$. Then, for any $U \in \mathcal{S}_m$,*

$$K(S'\|U) \le \frac{K(S\|U) + h(\delta)}{1-\delta}.$$

Proof The following identities are straightforward:

$$K(S\|U) = \mathrm{tr}(S(\log S - \log U))$$
$$= (1-\delta)\mathrm{tr}(S'(\log S - \log U)) + \delta\mathrm{tr}((I_m/m)(\log S - \log U))$$
$$= (1-\delta)\mathrm{tr}(S'(\log S' - \log U)) + (1-\delta)\mathrm{tr}(S'(\log S - \log S'))$$
$$+\delta\mathrm{tr}((I_m/m)(\log S - \log(I_m/m))) + \delta\mathrm{tr}((I_m/m)(\log(I_m/m) - \log U))$$
$$= (1-\delta)K(S'\|U) - (1-\delta)K(S'\|S) + \delta K(I_m/m\|U) - \delta K(I_m/m\|S).$$

Since $K(I_m/m\|U) \geq 0$, it follows that

$$K(S'\|U) \leq \frac{K(S\|U)}{1-\delta} + K(S'\|S) + \frac{\delta}{1-\delta}K(I_m/m\|S). \tag{47}$$

Assuming that S' has spectral representation $S' = \sum_{j=1}^r \lambda_j P_j$ with eigenvalues $\lambda_j > 0$ and one-dimensional projectors P_j, we get

$$-K(S'\|S) = \sum_{j=1}^r \lambda_j \log \frac{(1-\delta)\lambda_j + \delta/m}{\lambda_j}$$

$$= \sum_{j=1}^r \lambda_j \log\Big(1 - \delta + \frac{\delta}{m\lambda_j}\Big) \geq \log(1-\delta)\sum_{j=1}^r \lambda_j = \log(1-\delta),$$

implying that $K(S'\|S) \leq \log \frac{1}{1-\delta}$. On the other hand,

$$K(I_m/m\|S) = \frac{1}{m}\sum_{j=1}^m \log \frac{1/m}{(1-\delta)\lambda_j + \delta/m} \leq \frac{1}{m}\sum_{j=1}^m \log\frac{1}{\delta} = \log\frac{1}{\delta}.$$

Substituting these bounds in (47) yields the result. ■

To complete the proof of Theorem 15, we need to control the empirical process $(P - P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S)$ in the right hand side of bound (38). Our approach is based on the following empirical processes bound that is a slight modification of Lemma 1 in Koltchinskii (2013b). As before, we assume that $S = (1-\delta)S' + \delta\frac{I_m}{m}$ with $S' \in \mathcal{S}_m$, $\mathrm{rank}(S') = r$. We will set $\delta := \frac{1}{m^2n^2}$.

Let $\Xi_\varepsilon := n^{-1}\sum_{j=1}^n \varepsilon_j X_j$, where ε_j are i.i.d. Rademacher random variables (that is, ε_j takes values $+1$ and -1 with probability $1/2$ each) and $\{\varepsilon_j\}, \{X_j\}$ are independent.

Lemma 18 *Given $\delta_1, \delta_2 > 0$, denote*

$$\alpha_n(\delta_1,\delta_2) := \sup\Big\{\Big|(P_n - P)(\ell' \bullet f_A)(f_A - f_S)\Big| : A \in \mathcal{S}_m, \|f_A - f_S\|_{L_2(\Pi)} \leq \delta_1, \|\mathcal{P}_L^\perp A\|_1 \leq \delta_2\Big\}.$$

Let $0 < \delta_1^- < \delta_1^+, 0 < \delta_2^- < \delta_2^+$. For $t \geq 1$, denote

$$\bar t := t + \log\Big([\log_2(\delta_1^+/\delta_1^-)] + 2\Big) + \log\Big([\log_2(\delta_2^+/\delta_2^-)] + 2\Big) + \log 3.$$

Then, with probability at least $1 - e^{-t}$, for all $\delta_1 \in [\delta_1^-, \delta_1^+], \delta_2 \in [\delta_2^-, \delta_2^+]$,

$$\alpha_n(\delta_1,\delta_2) \leq C_1 U\mathbb{E}\|\Xi_\varepsilon\|_\infty\Big(\sqrt{r}m\delta_1 + \delta_2 + \delta\Big) + C_2 U\delta_1\sqrt{\frac{\bar t}{n}} + C_3U^2\frac{\bar t}{n},$$

where $C_1, C_2, C_3 > 0$ are constants.

We will use this lemma to control the term $(P-P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho}-f_S)$ in bound (38). Let $\delta_1 := \|f_{\tilde\rho}-f_S\|_{L_2(\Pi)}$ and $\delta_2 := \|\mathcal{P}_L^\perp \tilde\rho\|_1$. Define also

$$\delta_1^+ := \frac{2}{m},\ \delta_2^+ := 1,\ \delta_1^- = \delta_2^- := \frac{1}{mn},$$

so that $\bar t \le t + 2\log(\log_2(mn)+3) + \log 3$. It is easy to see that $\delta_1 \le \delta_1^+$ and $\delta_2 \le \delta_2^+$. If, in addition, $\delta_1 \ge \delta_1^-$, $\delta_2 \ge \delta_2^-$, the bound of Lemma 18 implies that with probability at least $1-e^{-t}$

$$(P-P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho}-f_S) \le \alpha_n(\delta_1,\delta_2)$$

$$\le C_1 U \mathbb{E}\|\Xi_\varepsilon\|_\infty \Big(\sqrt{r} m \delta_1 + \delta_2 + \delta\Big) + C_2 U \delta_1 \sqrt{\frac{\bar t}{n}} + C_3 U^2 \frac{\bar t}{n}$$

If $\bar\varepsilon \ge C_1 U \mathbb{E}\|\Xi_\varepsilon\|_\infty$, the last bound implies that

$$\begin{aligned}(P-P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho}-f_S)\\ \le \tfrac14 \|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)} + r m^2 \bar\varepsilon^2 + \bar\varepsilon \|\mathcal{P}_L^\perp \tilde\rho\|_1 + \bar\varepsilon\delta\\ + \tfrac14 \|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)} + (C_2^2 + C_3) U^2 \tfrac{\bar t}{n}.\end{aligned} \tag{48}$$

Substituting this bound in the right hand side of (40), we get

$$\begin{aligned}\|f_{\tilde\rho}-f_\rho\|^2_{L_2(\Pi)} + \varepsilon K(\tilde\rho; S)\\ \le \|f_{S'}-f_\rho\|^2_{L_2(\Pi)} + r m^2 \varepsilon^2 \log^2(m/\delta) + 2 r m^2 \bar\varepsilon^2\\ + 5\bar\varepsilon\delta + C U^2 \tfrac{\bar t}{n} + \tfrac{12\delta}{m^2},\end{aligned} \tag{49}$$

where $C := C_2^2 + C_3$.

In the case when $\delta_1 = \|f_{\tilde\rho}-f_S\|_{L_2(\Pi)} \le \delta_1^- = \frac{1}{mn}$ or $\delta_2 = \|\mathcal{P}_L^\perp\tilde\rho\|_1 \le \delta_2^- = \frac{1}{mn}$, we can replace the terms $\frac14\|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)}$ or $\|\mathcal{P}_L^\perp\tilde\rho\|_1$ in bound (48) by their respective upper bounds ($\frac14(\delta_1^-)^2 = \frac{1}{4m^2n^2}$, or $\delta_2^- = \frac{1}{mn}$), which would be smaller than $CU^2\frac{\bar t}{n}$ for large enough $C>0$, so bound (49) still holds (recall that $U \ge m^{-1/2}$). Note also that $\frac{12\delta}{m^2} = 12\frac{1}{m^4n^2} \le 12U^2\frac{\bar t}{n}$. Thus, increasing the value of constant C, one can rewrite (49) in a simpler form as

$$\begin{aligned}\|f_{\tilde\rho}-f_\rho\|^2_{L_2(\Pi)} + \varepsilon K(\tilde\rho; S)\\ \le \|f_{S'}-f_\rho\|^2_{L_2(\Pi)} + r m^2 \varepsilon^2 \log^2(m/\delta) + 2 r m^2 \bar\varepsilon^2\\ + 5\bar\varepsilon\delta + C U^2 \tfrac{\bar t}{n}.\end{aligned} \tag{50}$$

The following expectation bound is a consequence of a matrix version of Bernstein inequality for $\|\Xi_\varepsilon\|_\infty$ (it follows by integrating out its exponential tails):

$$\mathbb{E}\|\Xi_\varepsilon\|_\infty \le 4\Big[\sqrt{\frac{\log(2m)}{nm}} \bigvee U\frac{\log(2m)}{n}\Big]$$

(it is also used in this computation that, in the case of uniform sampling from an orthonormal basis, $\sigma^2_{\varepsilon X} = \|\mathbb{E}X^2\|_\infty = \frac1m$, a simple fact often used in the literature; see, e.g., Koltchinskii 2011a, Section 5). Let

$$\bar\varepsilon := D' U \sqrt{\frac{\log(2m)}{nm}}$$

for some constant D'. If D' is sufficiently large and

$$U\frac{\log(2m)}{n} \le \sqrt{\frac{\log(2m)}{nm}}, \tag{51}$$

then the condition $\bar{\varepsilon} \ge C_1 U\mathbb{E}\|\Xi_\varepsilon\|_\infty$ is satisfied and bound (50) holds with probability at least $1-e^{-t}$. Moreover, $\bar{\varepsilon}\delta \lesssim_{D'} \delta \lesssim_{D'} U^2\frac{\bar{t}}{n}$, implying that the term $5\bar{\varepsilon}\delta$ in (50) can be dropped at a price of further increasing the value of constant C.

If (51) does not hold, we still have that

$$\|f_{\tilde{\rho}} - f_\rho\|^2_{L_2(\Pi)} = \frac{\|\tilde{\rho}-\rho\|_2^2}{m^2} \le \frac{2}{m^2} \le CU^2\frac{\bar{t}}{n}.$$

Recalling that $\bar{t} \le t + 2\log(\log_2(mn)+3)$ and $\log(m/\delta) \lesssim \log(mn)$, we deduce from (50) that with some constant C and with probability at least $1-e^{-t}$

$$\|f_{\tilde{\rho}} - f_\rho\|^2_{L_2(\Pi)} \le \|f_{S'} - f_\rho\|^2_{L_2(\Pi)} + C\Big[rm^2\varepsilon^2\log^2(mn) + U^2\frac{rm\log(2m)}{n} + U^2\frac{t+\log(\log_2(mn)+3)}{n}\Big]. \tag{52}$$

Note that, for $n \ge 2$,

$$\log(\log_2(mn)+3) = \log\Big(\log_2(4m)+\log_2(2n)\Big) \le \log\log_2(4m) + \log\log_2(2n), \tag{53}$$

since $\log_2(4m)+\log_2(2n) \le \log_2(4m)\log_2(2n)$. Since also, for $r \ge 1$,

$$U^2\frac{t+\log\log_2(4m)}{n} \lesssim U^2\frac{rm\log(2m)}{n}, \tag{54}$$

we can replace in bound (52) the term $U^2\frac{t+\log(\log_2(mn)+3)}{n}$ with the term $U^2\frac{t+\log\log_2(2n)}{n}$ (increasing the value of the constant C accordingly). This yields bound (34) of the theorem. For $S'=\rho$, it yields bound (35), and, moreover, for $S'=\rho$ and $S=(1-\delta)\rho+\delta\frac{I_m}{m}$ with $\delta=\frac{1}{m^2n^2}$, bound (50) also implies that

$$\varepsilon K(\tilde{\rho};S) \le \mathrm{rank}(\rho)m^2\varepsilon^2\log^2(m/\delta) + 2\mathrm{rank}(\rho)m^2\bar{\varepsilon}^2 + 5\bar{\varepsilon}\delta + CU^2\frac{\bar{t}}{n}. \tag{55}$$

We will now take

$$\bar{\varepsilon} := D'\Big[U\sqrt{\frac{\log(2m)}{nm}} \bigvee U^2\frac{\log(2m)}{n}\Big]$$

for a large enough constant D' so that $\bar{\varepsilon} \ge C_1 U\mathbb{E}\|\Xi_\varepsilon\|_\infty$. Assume that

$$\varepsilon := \frac{1}{\log(mn)}\Big[U\sqrt{\frac{\log(2m)}{nm}} \bigvee U^2\frac{\log(2m)}{n}\Big].$$

As before, the term $\bar{\varepsilon}\delta$ in bound (55) will be absorbed by the term $CU^2\frac{\bar{t}}{n}$ with a larger value of C and also

$$\operatorname{rank}(\rho)m^2\varepsilon^2\log^2(m/\delta) \asymp_{D'} \operatorname{rank}(\rho)m^2\bar{\varepsilon}^2 \asymp_{D'} U^2\frac{\operatorname{rank}(\rho)m\log(2m)}{n}\left(1\bigvee U^2\frac{m\log(2m)}{n}\right).$$

As a result, taking into account (53), (54), bound (55) can be rewritten as follows:

$$\varepsilon K(\tilde{\rho};S) \le CU^2\left[\frac{\operatorname{rank}(\rho)m\log(2m)}{n}\left(1\bigvee U^2\frac{m\log(2m)}{n}\right)\right. \tag{56}$$
$$\left.+\frac{t+\log\log_2(2n)}{n}\right].$$

Using the bound of Lemma 17 along with the bound

$$h(\delta) \le \delta\log(e/\delta) = \frac{1}{m^2n^2}\log(em^2n^2) \lesssim U\sqrt{\frac{m}{n}}\frac{(t+\log\log_2(2n))\log(mn)}{\sqrt{\log(2m)}},$$

we easily get that (37) holds. ■

3.2 Oracle Inequalities for Trace Regression with Gaussian Noise

In this subsection, we establish oracle inequalities for the von Neumann entropy penalized least squares estimator $\tilde{\rho}^{\varepsilon}$ in the case of trace regression model with Gaussian noise (Assumption 4). Unlike in the case of Theorem 15 of the previous section, our aim is not to obtain sharp oracle inequality, but rather to get a clean main term of the random error bound part of the inequality, namely, the term $\sigma_\xi^2\frac{\operatorname{rank}(S)m(t+\log(2m))}{n}$ in inequality (58) below. Note that this term depends only on the variance of the noise σ_ξ^2, but not on the constant U from Assumption 1 (the constant U is involved only in the higher order $O(n^{-2})$ terms of the bound). Note also that there are no constraints on the variance σ_ξ^2 that could be arbitrarily small, or even equal to 0 (in which case only higher order terms are present in the bound). This improvement comes at a price of having the leading constant 2 in the oracle inequality and also of imposing assumption (57) that requires the regularization parameter ε to be bounded away from 0 (again, unlike Theorem 15, where it could be arbitrarily small). As in the previous section, we also obtain a bound on Kullback–Leibler divergence $K(\rho\|\tilde{\rho}^{\varepsilon})$.

Theorem 19 *Let $t \ge 1$. Suppose*

$$\varepsilon \in \left[DU^2\frac{t+\log^3 m\log^2 n}{n}, \frac{D_1\sigma_\xi}{\log(mn)}\sqrt{\frac{t+\log(2m)}{nm}}\bigvee DU^2\frac{t+\log^3 m\log^2 n}{n}\right] \tag{57}$$

with large enough constants $D, D_1 > 0$. There exists a constant $C > 0$ such that with probability at least $1-e^{-t}$

$$\|f_{\tilde{\rho}^{\varepsilon}} - f_\rho\|^2_{L_2(\Pi)} \le \inf_{S\in\mathcal{S}_m}\left[2\|f_S - f_\rho\|^2_{L_2(\Pi)} + C\left(\sigma_\xi^2\frac{\operatorname{rank}(S)m(t+\log(2m))}{n}\right.\right.$$
$$\left.\left.+\sigma_\xi^2U^2\frac{\operatorname{rank}(S)m^2(t+\log(2m))^2\log(2m)}{n^2} + U^4\frac{\operatorname{rank}(S)m^2(t+\log^3 m\log^2 n)^2\log^2(mn)}{n^2}\right)\right]. \tag{58}$$

In particular,

$$\|f_{\tilde{\rho}^{\varepsilon}} - f_{\rho}\|^2_{L_2(\Pi)} \le C\Big[\sigma_{\xi}^2 \tfrac{\mathrm{rank}(\rho)m(t+\log(2m))}{n} \tag{59}$$

$$+\sigma_{\xi}^2 U^2 \tfrac{\mathrm{rank}(\rho)m^2(t+\log(2m))^2\log(2m)}{n^2} + U^4 \tfrac{\mathrm{rank}(\rho)m^2(t+\log^3 m\log^2 n)^2\log^2(mn)}{n^2}\Big].$$

Moreover, if

$$\varepsilon := \frac{D_1\sigma_{\xi}}{\log(mn)}\sqrt{\frac{t+\log(2m)}{nm}} \bigvee DU^2\frac{t+\log^3 m\log^2 n}{n}$$

for large enough constants D, D_1, then with some constant C and with the same probability both (59) and the following bound hold:

$$K(\rho\|\tilde{\rho}^{\varepsilon}) \le C\Big[\sigma_{\xi}\tfrac{\mathrm{rank}(\rho)m^{3/2}(t+\log(2m))^{1/2}\log(mn)}{\sqrt{n}} \tag{60}$$

$$+\sigma_{\xi}^2\tfrac{\mathrm{rank}(\rho)m^2(t+\log(2m))\log(2m)}{n} + U^2\tfrac{\mathrm{rank}(\rho)m^2(t+\log^3 m\log^2 n)\log^2(mn)}{n}\Big].$$

Proof As in in the proof of Theorem 15, we rely on Lemma 16, but we use a different approach to bounding the empirical process $(P-P_n)(\ell'\bullet f_{\tilde{\rho}})(f_{\tilde{\rho}}-f_S)$. The following identity follows from the definition of quadratic loss ℓ

$$(\ell'\bullet f)(x,y)(f(x)-f_S(x)) = 2(f(x)-f_S(x))^2 + 2(f_S(x)-y)(f(x)-f_S(x))$$

and it implies that

$$(P-P_n)(\ell'\bullet f_{\tilde{\rho}})(f_{\tilde{\rho}}-f_S) = -2(P_n-P)(f_{\tilde{\rho}}-f_S)^2 - 2\langle \Xi, \tilde{\rho}-S\rangle \tag{61}$$

where

$$\Xi := n^{-1}\sum_{j=1}^n (f_S(X_j)-Y_j)X_j - \mathbb{E}(f_S(X)-Y)X.$$

We will bound $(P_n-P)(f_{\tilde{\rho}}-f_S)^2$ in representation (61) as follows:

$$\Big|(P_n-P)(f_{\tilde{\rho}}-f_S)^2\Big| \le \|\tilde{\rho}-S\|_1^2\beta_n\Big(\frac{\|f_{\tilde{\rho}}-f_S\|_{L_2(\Pi)}}{\|\tilde{\rho}-S\|_1}\Big), \tag{62}$$

where

$$\beta_n(\Delta) := \sup\Big\{\Big|(P_n-P)(f_A^2)\Big| : A\in\mathbb{H}_m, \|A\|_1\le 1, \|f_A\|_{L_2(\Pi)}\le\Delta\Big\}.$$

The next lemma provides a bound on $\beta_n(\Delta)$. Its proof is somewhat involved and it will not be given here. It is based on Rudelson's $L_\infty(P_n)$ generic chaining bound for empirical processes indexed by squares of functions and on the ideas of the paper by Guédon et al. (2008) combined with Talagrand's concentration inequality (see also Aubrun 2009, Liu 2011 and Theorem 3.16, Lemma 9.8 and Proposition 9.2 in Koltchinskii 2011b for similar arguments).

Lemma 20 *Given $0 < \delta^- < \delta^+$ and $t \geq 1$, let*

$$\bar{t} := t + \log\Big(\log_2(\delta^+/\delta^-) + 3\Big).$$

Then, with some constant C and with probability at least $1 - e^{-t}$, the following bound holds for all $\Delta \in [\delta^-, \delta^+]$:

$$\beta_n(\Delta) \leq C\Big[\Delta U \frac{\log^{3/2} m \log n}{\sqrt{n}} + U^2 \frac{\log^3 m \log^2 n}{n} + \Delta U \sqrt{\frac{\bar{t}}{n}} + U^2 \frac{\bar{t}}{n}\Big]. \tag{63}$$

We will use Lemma 20 to control $\beta_n(\Delta)$ for $\Delta := \frac{\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}}{\|\tilde\rho - S\|_1}$. Let $\delta^+ := \frac{1}{m}$ and $\delta^- := \frac{1}{mn}$. With this choice, $\bar{t} \leq t + \log(\log_2 n + 3)$. Note that for $A = \frac{\tilde\rho - S}{\|\tilde\rho - S\|_1}$, $\|f_A\|_{L_2(\Pi)} = \frac{\|A\|_2}{m} \leq \frac{\|A\|_1}{m} = m^{-1} = \delta^+$. If also $\|f_A\|_{L_2(\Pi)} \geq \delta^-$, then we can substitute bound (63) on $\beta_n(\Delta)$ into (62) that yields:

$$\begin{aligned}
\Big|(P_n - P)(f_{\tilde\rho} - f_S)^2\Big| &\leq C\Big[\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)} \|\tilde\rho - S\|_1 U \tfrac{\log^{3/2} m \log n}{\sqrt{n}} \\
&\quad + \|\tilde\rho - S\|_1^2 U^2 \tfrac{\log^3 m \log^2 n}{n} + \|f_{\tilde\rho} - f_S\|_{L_2(\Pi)} \|\tilde\rho - S\|_1 U \sqrt{\tfrac{\bar{t}}{n}} \\
&\quad + \|\tilde\rho - S\|_1^2 U^2 \tfrac{\bar{t}}{n}\Big] \\
&\leq \tfrac{1}{32}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 + 8(C^2 + C/8) U^2 \tfrac{\log^3 m \log^2 n}{n} \|\tilde\rho - S\|_1^2 \\
&\quad + \tfrac{1}{32}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 + 8(C^2 + C/8) U^2 \tfrac{\bar{t}}{n} \|\tilde\rho - S\|_1^2 \\
&\leq \tfrac{1}{16}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 + C' U^2 \tfrac{\log^3 m \log^2 n + \bar{t}}{n} \|\tilde\rho - S\|_1^2,
\end{aligned} \tag{64}$$

where $C' := 8(C^2 + C/8)$. If, on the other hand, $\|f_A\|_{L_2(\Pi)} \leq \delta^- = \frac{1}{mn}$, then $\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}$ in the above bound can be replaced by $\frac{1}{mn}\|\tilde\rho - S\|_1$ and the proof that follows only simplifies since

$$\frac{1}{16}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 \leq \frac{1}{16}\frac{1}{m^2 n^2}\|\tilde\rho - S\|_1^2 \leq \frac{1}{16} U^2 \frac{\log^3 m \log^2 n + \bar{t}}{n} \|\tilde\rho - S\|_1^2.$$

Another term in the right hand side of representation (61) to be controlled is $\langle \Xi, \tilde\rho - S\rangle$. Note that $\Xi = \Xi_1 + \Xi_2$, where

$$\Xi_1 := -n^{-1} \sum_{j=1}^n \xi_j X_j$$

and

$$\Xi_2 := n^{-1} \sum_{j=1}^n (f_S(X_j) - f_\rho(X_j)) X_j - \mathbb{E}(f_S(X) - f_\rho(X)) X.$$

Recall that $S = (1-\delta)S' + \delta \frac{I_m}{m}$ with $S' \in \mathcal{S}_m$, $\mathrm{rank}(S') = r$, $\mathrm{supp}(S') = L$ and $\delta = \frac{1}{m^2 n^2}$.

The term with Ξ_1 is controlled as follows:

$$\begin{aligned}
&\Big|\langle \Xi_1, \tilde\rho - S\rangle\Big| \\
&\le \Big|\langle \mathcal{P}_L(\Xi_1), \tilde\rho - S'\rangle\Big| + \Big|\langle \Xi_1, \mathcal{P}_L^\perp(\tilde\rho - S')\rangle\Big| + \Big|\langle \mathcal{P}_L^\perp(\Xi_1), S' - S\rangle\Big| \\
&\le \|\mathcal{P}_L(\Xi_1)\|_2 \|\tilde\rho - S'\|_2 + \|\Xi_1\|_\infty \|\mathcal{P}_L^\perp(\tilde\rho)\|_1 + \Big\|\mathcal{P}_L^\perp(\Xi_1)\Big\|_\infty \|S' - S\|_1 \\
&\le 2\sqrt{2r}m\|\Xi_1\|_\infty \|f_{\tilde\rho} - f_S\|_{L_2(\Pi)} + \|\Xi_1\|_\infty \|\mathcal{P}_L^\perp(\tilde\rho)\|_1 + 4\delta\|\Xi_1\|_\infty \qquad (65)\\
&\le 32rm^2\|\Xi_1\|_\infty^2 + \tfrac{1}{16}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 \\
&\quad + \|\Xi_1\|_\infty \|\mathcal{P}_L^\perp(\tilde\rho)\|_1 + 4\delta\|\Xi_1\|_\infty.
\end{aligned}$$

We also have

$$\begin{aligned}
\Big|\langle \Xi_2, \tilde\rho - S\rangle\Big| &\le \|\Xi_2\|_\infty \|\tilde\rho - S\|_1 \le \|\Xi_2\|_\infty \|\tilde\rho - S'\|_1 + \|\Xi_2\|_\infty \|S' - S\|_1 \\
&\le \|\Xi_2\|_\infty \|\tilde\rho - S'\|_1 + 2\delta\|\Xi_2\|_\infty. \qquad (66)
\end{aligned}$$

Thus,

$$\begin{aligned}
&\Big|\langle \Xi, \tilde\rho - S\rangle\Big| \le 32rm^2\|\Xi_1\|_\infty^2 + \tfrac{1}{16}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 \\
&+ \|\Xi_1\|_\infty \|\mathcal{P}_L^\perp(\tilde\rho)\|_1 + 4\delta\|\Xi_1\|_\infty + \|\Xi_2\|_\infty \|\tilde\rho - S'\|_1 + 2\delta\|\Xi_2\|_\infty. \qquad (67)
\end{aligned}$$

It follows from (61), (64) and (67) that with some constant C'

$$\begin{aligned}
&(P - P_n)(\ell' \bullet f_{\tilde\rho})(f_{\tilde\rho} - f_S) \le \\
&\tfrac{1}{4}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 + C'U^2 \tfrac{\log^3 m \log^2 n + \bar t}{n} \|\tilde\rho - S\|_1^2 \qquad (68)\\
&+ 64rm^2\|\Xi_1\|_\infty^2 + 2\|\Xi_1\|_\infty \|\mathcal{P}_L^\perp(\tilde\rho)\|_1 + 8\delta\|\Xi_1\|_\infty \\
&+ 2\|\Xi_2\|_\infty \|\tilde\rho - S'\|_1 + 4\delta\|\Xi_2\|_\infty.
\end{aligned}$$

This bound will be substituted in (38). Note that, if assumption (57) on ε holds with a sufficiently large constant D, then we have

$$\varepsilon \ge 8C'U^2 \frac{\log^3 m \log^2 n + \bar t}{n}$$

(this follows from the fact that $\bar t \le t + \log(\log_2 n + 3) \le t + c\log^3 m \log^2 n$ for some constant $c > 0$). Assume also that $\bar\varepsilon \ge 4\|\Xi_1\|_\infty$ and recall that $K(\tilde\rho; S) \ge \frac{1}{4}\|\tilde\rho - S\|_1^2$ (see inequality 8). Taking all this into account, (38) implies that

$$\begin{aligned}
&\|f_{\tilde\rho} - f_\rho\|_{L_2(\Pi)}^2 + \tfrac{1}{4}\|f_{\tilde\rho} - f_S\|_{L_2(\Pi)}^2 + \tfrac{\varepsilon}{2}K(\tilde\rho; S) + \tfrac{\bar\varepsilon}{2}\|\mathcal{P}_L^\perp \tilde\rho\|_1 \\
&\le \|f_S - f_\rho\|_{L_2(\Pi)}^2 + rm^2\varepsilon^2 \log^2(m/\delta) + 5rm^2\bar\varepsilon^2 + 6\bar\varepsilon\delta \qquad (69)\\
&\quad + 2\|\Xi_2\|_\infty \|\tilde\rho - S'\|_1 + 4\|\Xi_2\|_\infty \delta.
\end{aligned}$$

It remains to control $\|\Xi_1\|_\infty$ and $\|\Xi_2\|_\infty$. To this end, we use matrix versions of Bernstein inequality. To bound $\|\Xi_2\|_\infty$, we use its standard version which yields that with probability

at least $1-e^{-t}$

$$\|\Xi_2\|_\infty \le 2\Big[\Big\|\mathbb{E}(f_S(X)-f_\rho(X))^2X^2\Big\|_\infty^{1/2}\sqrt{\tfrac{t+\log(2m)}{n}} \bigvee\Big\|(f_S(X)-f_\rho(X))\|X\|_\infty\Big\|_{L_\infty}\tfrac{t+\log(2m)}{n}\Big],$$

where $\|\cdot\|_{L_\infty}$ denotes the essential supremum norm in the space of random variables. Since

$$\Big\|\mathbb{E}(f_S(X)-f_\rho(X))^2X^2\Big\|_\infty \le U^2\|f_S-f_\rho\|^2_{L_2(\Pi)}$$

and

$$\Big\|(f_S(X)-f_\rho(X))\|X\|_\infty\Big\|_{L_\infty}\le 2U^2,$$

we get

$$\|\Xi_2\|_\infty \le 4\Big[\|f_S-f_\rho\|_{L_2(\Pi)}U\sqrt{\tfrac{t+\log(2m)}{n}}+U^2\tfrac{t+\log(2m)}{n}\Big]. \tag{70}$$

This implies that

$$\begin{aligned}2\|\Xi_2\|_\infty\|\tilde\rho-S'\|_1 \le \|f_S-f_\rho\|^2_{L_2(\Pi)}+16U^2\tfrac{t+\log(2m)}{n}\|\tilde\rho-S'\|_1^2 \\ +8U^2\tfrac{t+\log(2m)}{n}\|\tilde\rho-S'\|_1.\end{aligned} \tag{71}$$

Note that

$$\begin{aligned}&16U^2\tfrac{t+\log(2m)}{n}\|\tilde\rho-S'\|_1^2\\ &\le 16U^2\tfrac{t+\log(2m)}{n}\|\tilde\rho-S\|_1^2+16U^2\tfrac{t+\log(2m)}{n}(4\delta+\delta^2)\end{aligned} \tag{72}$$

and

$$\begin{aligned}&8U^2\tfrac{t+\log(2m)}{n}\|\tilde\rho-S'\|_1\\ &\le 8U^2\tfrac{t+\log(2m)}{n}\|\mathcal{P}_L^\perp\tilde\rho\|_1+8U^2\tfrac{t+\log(2m)}{n}\|\mathcal{P}_L(\tilde\rho-S')\|_1\\ &\le 8U^2\tfrac{t+\log(2m)}{n}\|\mathcal{P}_L^\perp\tilde\rho\|_1+8U^2\tfrac{t+\log(2m)}{n}\|\mathcal{P}_L(\tilde\rho-S)\|_1+16U^2\tfrac{t+\log(2m)}{n}\delta.\end{aligned} \tag{73}$$

Since, for some constant $C''>0$,

$$\begin{aligned}&8U^2\tfrac{t+\log(2m)}{n}\|\mathcal{P}_L(\tilde\rho-S)\|_1\le 8\sqrt{2}U^2\tfrac{t+\log(2m)}{n}\sqrt{r}\|\mathcal{P}_L(\tilde\rho-S)\|_2\\ &\le 8\sqrt{2}U^2\tfrac{t+\log(2m)}{n}\sqrt{r}m\|f_{\tilde\rho}-f_S\|_{L_2(\Pi)}\le\tfrac14\|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)}+C''U^4\tfrac{rm^2(t+\log(2m))^2}{n^2},\end{aligned}$$

it follows from (71), (72) and (73) that

$$\begin{aligned}2\|\Xi_2\|_\infty\|\tilde\rho-S'\|_1 \le \|f_S-f_\rho\|^2_{L_2(\Pi)}+\\ +16U^2\tfrac{t+\log(2m)}{n}\|\tilde\rho-S\|_1^2+16U^2\tfrac{t+\log(2m)}{n}(4\delta+\delta^2)\\ +8U^2\tfrac{t+\log(2m)}{n}\|\mathcal{P}_L^\perp\tilde\rho\|_1+16U^2\tfrac{t+\log(2m)}{n}\delta\\ +\tfrac14\|f_{\tilde\rho}-f_S\|^2_{L_2(\Pi)}+C''U^4\tfrac{rm^2(t+\log(2m))^2}{n^2}.\end{aligned} \tag{74}$$

Note that (70) also implies that

$$\|\Xi_2\|_\infty \le 4\left[\tfrac{2U}{m}\sqrt{\tfrac{t+\log(2m)}{n}} + U^2\tfrac{t+\log(2m)}{n}\right] \tag{75}$$

(since $\|f_S - f_\rho\|_{L_2(\Pi)} \le m^{-1}\|S-\rho\|_2 \le 2m^{-1}$). Let us substitute (74) and (75) in the last line of (69). Assume that

$$\bar\varepsilon \ge 16U^2\frac{t+\log(2m)}{n}$$

and that constant D in assumption (57) is large enough so that

$$16U^2\frac{t+\log(2m)}{n}\|\tilde\rho - S\|_1^2 \le \frac{\varepsilon}{4}K(\tilde\rho, S)$$

(recall inequality 8). It easily follows that with some constants C_1, C_2,

$$\begin{aligned}&\|f_{\tilde\rho} - f_\rho\|^2_{L_2(\Pi)} + \tfrac{\varepsilon}{4}K(\tilde\rho; S)\\ &\le 2\|f_S - f_\rho\|^2_{L_2(\Pi)} + C_1 rm^2\varepsilon^2\log^2(m/\delta) + 5rm^2\bar\varepsilon^2\\ &\quad + C_2\bar\varepsilon\delta + 32\tfrac{U}{m}\sqrt{\tfrac{t+\log(2m)}{n}}\delta\end{aligned} \tag{76}$$

(note that the term $C''U^4\frac{rm^2(t+\log(2m))^2}{n^2}$ of bound (74) is "absorbed" by the term $C_1 rm^2\varepsilon^2\log^2(m/\delta)$ of bound (76) provided that constant C_1 is large enough). Since

$$\delta = \frac{1}{m^2n^2} \le U^2\frac{t+\log(2m)}{n} \le \bar\varepsilon$$

(recall that $U^2 \ge m^{-1}$), we have $\bar\varepsilon\delta \le \bar\varepsilon^2$. Also, since $U \ge m^{-1/2}$,

$$\frac{U}{m}\sqrt{\frac{t+\log(2m)}{n}}\delta = U\sqrt{\frac{t+\log(2m)}{n}}\frac{1}{m^3n^2} \le U^4\Big(\frac{t+\log(2m)}{n}\Big)^2 \le \bar\varepsilon^2.$$

Therefore, (76) implies that with some constant C

$$\begin{aligned}&\|f_{\tilde\rho} - f_\rho\|^2_{L_2(\Pi)} + \tfrac{\varepsilon}{4}K(\tilde\rho; S)\\ &\le 2\|f_S - f_\rho\|^2_{L_2(\Pi)} + C\Big(rm^2\varepsilon^2\log^2(m/\delta) + rm^2\bar\varepsilon^2\Big).\end{aligned} \tag{77}$$

To bound $\|\Xi_1\|_\infty$, we use a version of matrix Bernstein type inequality due to Koltchinskii (2011b) (see bound (2.7) of Theorem 2.7). Its version for $\alpha = 2$ (with $U^{(\alpha)} \asymp U\sigma_\xi$) implies that for some constant $K > 0$ with probability at least $1 - e^{-t}$

$$\|\Xi_1\|_\infty \le K\Big[\sigma_\xi\sqrt{\frac{t+\log(2m)}{nm}} \bigvee \sigma_\xi U\frac{(t+\log(2m))\log^{1/2}(2Um^{1/2})}{n}\Big]. \tag{78}$$

We choose

$$\bar\varepsilon := D_2\Big[\sigma_\xi\sqrt{\frac{t+\log(2m)}{nm}} \bigvee (\sigma_\xi \vee U)U\frac{(t+\log(2m))\log^{1/2}(2m)}{n}\Big]$$

with a sufficiently large constant D_2 to satisfy the condition $\|\Xi_1\|_\infty \le 4\bar\varepsilon$ with probability at least $1-e^{-t}$ (the rest of the assumptions we made on $\bar\varepsilon$ are also satisfied with this choice).

Bound (77) then implies that with some constant C and with probability at least $1-3e^{-t}$ the following inequality holds:

$$\begin{aligned}\|f_{\tilde\rho^\varepsilon} - f_\rho\|^2_{L_2(\Pi)} &\le 2\|f_S - f_\rho\|^2_{L_2(\Pi)}\\ &+ C\Big[\sigma_\xi^2\frac{rm(t+\log(2m))}{n} + \sigma_\xi^2 U^2\frac{rm^2(t+\log(2m))^2\log(2m)}{n^2}\\ &+ U^4\frac{rm^2(t+\log^3 m\log^2 n)^2\log^2(mn)}{n^2}\Big].\end{aligned} \tag{79}$$

Using bound (39) to replace S in $\|f_S - f_\rho\|^2_{L_2(\Pi)}$ with S' and adjusting the value of constant C to rewrite the probability bound as $1-e^{-t}$, it is easy to complete the proof of (58). If $S' = \rho$, this also yields bound (59). Moreover, with a larger value of regularization parameter

$$\varepsilon := \frac{D_1\sigma_\xi}{\log(mn)}\sqrt{\frac{t+\log(2m)}{nm}} \bigvee DU^2\frac{t+\log^3 m\log^2 n}{n},$$

bound (77) and Lemma 17 easily imply bound (60). ∎

3.3 Optimality Properties of von Neumann Entropy Penalized Estimator $\tilde\rho^\epsilon$

We start with upper bounds on the error of estimator $\tilde\rho^\epsilon$ (von Neumann entropy penalized least squares estimator defined by (7)) in Hellinger, Kullback-Leibler and Schatten q-norm distances for $q \in [1,2]$ for the trace regression model with Gaussian noise (Assumption 4). To avoid the impact of "second order terms" on the upper bounds, we will make the following simplifying assumptions:

$$U\sqrt{\frac{m}{n}}\log m \lesssim 1 \quad\text{and}\quad U^2\sqrt{\frac{m}{n}}\log^{5/2} m\log^2 n\log(mn) \lesssim \sigma_\xi. \tag{80}$$

Recall that, for the Pauli basis, $U = m^{-1/2}$, so, the above assumptions hold if $n \gtrsim \log^2 m$ and σ_ξ is larger than $\frac{1}{\sqrt{mn}}$ (times a logarithmic factor). We will choose regularization parameter ε as follows:

$$\varepsilon := \frac{D_1\sigma_\xi}{\log(mn)}\sqrt{\frac{\log(2m)}{nm}} \tag{81}$$

with a sufficiently large constant $D_1 > 0$. The next result shows that minimax rates of Theorem 4 are attained up to logarithmic factors for the estimator $\tilde\rho^\varepsilon$.

Theorem 21 *There exists a constant $C > 0$ such that the following bounds hold for all $r = 1,\dots,m$, for all $\rho \in \mathcal{S}_{r,m}$ and for all $q \in [1,2]$ with probability at least $1 - m^{-2}$:*

$$\|\tilde\rho^\varepsilon - \rho\|_q \le C\Big(\frac{\sigma_\xi m^{\frac{3}{2}} r^{1/q}}{\sqrt{n}}\sqrt{\log m}\log^{(2-q)/q}(mn) \bigwedge \Big(\frac{\sigma_\xi m^{3/2}}{\sqrt{n}}\Big)^{1-\frac{1}{q}}(\log m)^{\frac{1}{2}-\frac{1}{2q}}\Big) \bigwedge 2, \tag{82}$$

$$H^2(\tilde{\rho}^{\varepsilon},\rho)\le C\frac{\sigma_{\xi}m^{\frac{3}{2}}r}{\sqrt{n}}\sqrt{\log m}\log(mn)\bigwedge 2 \tag{83}$$

and

$$K(\rho\|\tilde{\rho}^{\varepsilon})\le C\frac{\sigma_{\xi}m^{\frac{3}{2}}r}{\sqrt{n}}\sqrt{\log m}\log(mn). \tag{84}$$

Proof We will need the following simple lemma.

Lemma 22 *For all $\rho\in\mathcal{S}_m$ and all $l=1,\dots,m$, there exists $\rho'\in\mathcal{S}_{l,m}$ such that*

$$\|\rho-\rho'\|_2^2\le\frac{1}{l}.$$

Proof Suppose that $\rho=\sum_{j=1}^m\lambda_jP_j$, where λ_j are the eigenvalues of ρ repeated with their multiplicities and P_j are orthogonal one-dimensional projectors. Note that $\{\lambda_j:j=1,\dots,m\}$ is a probability distribution on the set $\{1,\dots,m\}$. Let ν be a random variable sampled from this distribution and $\nu_1,\dots,\nu_l$ be its i.i.d. copies. Then $\mathbb{E}P_{\nu}=\rho$ and

$$\mathbb{E}\left\|l^{-1}\sum_{j=1}^l P_{\nu_j}-\rho\right\|_2^2=\frac{\mathbb{E}\|P_{\nu}-\rho\|_2^2}{l}=\frac{\mathbb{E}\|P_{\nu}\|_2^2-\|\rho\|_2^2}{l}=\frac{1-\|\rho\|_2^2}{l}\le\frac{1}{l}.$$

Therefore, there exists a realization $\nu_1=k_1,\dots,\nu_l=k_l$ of r.v. $\nu_1,\dots,\nu_l$ such that

$$\left\|l^{-1}\sum_{j=1}^l P_{k_j}-\rho\right\|_2^2\le\frac{1}{l}.$$

Denote $\rho':=l^{-1}\sum_{j=1}^l P_{k_j}$. Then, $\rho'\in\mathcal{S}_{l,m}$ and $\|\rho-\rho'\|_2^2\le\frac{1}{l}$. ■

First, we will prove bound (82) for $q=2$. To this end, we use oracle inequality (58) with $t=2\log m+\log 2$ and with oracle $S=\rho'\in\mathcal{S}_{l,m}$ such that $\|\rho-\rho'\|_2^2\le\frac{1}{l}$. Under simplifying assumptions (80) it yields that with probability at least $1-\frac{1}{2}m^{-2}$

$$\|\tilde{\rho}^{\varepsilon}-\rho\|_2^2=m^2\|f_{\tilde{\rho}^{\varepsilon}}-f_{\rho}\|_{L_2(\Pi)}^2\lesssim\left[\frac{1}{l}+\tau^2l\log m\right],$$

where $\tau:=\frac{\sigma_{\xi}m^{3/2}}{\sqrt{n}}$. On the other hand, using the same inequality with $S=\rho\in\mathcal{S}_{r,m}$ yields the bound

$$\|\tilde{\rho}^{\varepsilon}-\rho\|_2^2\lesssim\tau^2r\log m$$

that also holds with probability at least $1-\frac{1}{2}m^{-2}$. Therefore, with probability at least $1-m^{-2}$

$$\|\tilde{\rho}^{\varepsilon}-\rho\|_2^2\lesssim\Big(\frac{1}{l}+\tau^2l\log m\Big)\bigwedge\tau^2r\log m. \tag{85}$$

Let $\bar{l}=\frac{1}{\tau\sqrt{\log m}}$. If $\bar{l}\in[1,m]$, set $l:=[\bar{l}]$. Otherwise, if $\bar{l}>m$, set $l:=m$ and, if $\bar{l}<1$, set $l:=1$. An easy computation shows that with such a choice of l bound (85) implies (82) for $q=2$.

Next we use bound (60) that, for $t = 2\log m$, implies under assumptions (80) that with some constant C and with probability at least $1 - m^{-2}$

$$K(\rho\|\tilde{\rho}^{\varepsilon}) \le C\sigma_{\xi}\frac{rm^{3/2}\sqrt{\log m}\log(mn)}{\sqrt{n}}, \tag{86}$$

which is bound (84). Bound (83) also holds in view of inequality (8).

Now, we prove bound (82) for $q = 1$ (the bound for $q \in [1, 2]$ will then follow by interpolation). To this end, we will use the following lemma (see Proposition 1 in Koltchinskii 2011a) that shows that if two density matrices are close in Hellinger distance and one of them is "concentrated around a subspace" L, then another one is also "concentrated around" L.

Lemma 23 *For any $L \subset \mathbb{C}^m$ and all $S_1, S_2 \in \mathcal{S}_m$,*

$$\|\mathcal{P}_L^{\perp} S_1\|_1 \le 2\|\mathcal{P}_L^{\perp} S_2\|_1 + 2H^2(S_1, S_2).$$

We apply this lemma to $S_1 = \tilde{\rho}^{\varepsilon}$, $S_2 = \rho$ and $L = \mathrm{supp}(\rho)$ so that $\mathcal{P}_L^{\perp}\rho = 0$. It yields that

$$\|\mathcal{P}_L^{\perp}\tilde{\rho}^{\varepsilon}\|_1 \le 2H^2(\tilde{\rho}^{\varepsilon}, \rho).$$

Therefore,

$$\|\tilde{\rho}^{\varepsilon}-\rho\|_1 \le \|\mathcal{P}_L(\tilde{\rho}^{\varepsilon}-\rho)\|_1+\|\mathcal{P}_L^{\perp}(\tilde{\rho}^{\varepsilon}-\rho)\|_1 \le \sqrt{2r}\|\tilde{\rho}^{\varepsilon}-\rho\|_2+\|\mathcal{P}_L^{\perp}\tilde{\rho}^{\varepsilon}\|_1 \le \sqrt{2r}\|\tilde{\rho}^{\varepsilon}-\rho\|_2+2H^2(\tilde{\rho}^{\varepsilon}, \rho). \tag{87}$$

Using bounds (82) for $q = 2$ and (83), we get from (87) that

$$\|\tilde{\rho}^{\varepsilon} - \rho\|_1 \le C\frac{\sigma_{\xi}m^{\frac{3}{2}}r}{\sqrt{n}}\sqrt{\log m}\log(mn)\bigwedge 2, \tag{88}$$

which is equivalent to (82) for $q = 1$. Note that by choosing $t = 2\log m + \log 2 + 2$ (which might have an impact only on the constant), we could make probability bounds in (82) for $q = 2$ and (83) to be at least $1 - \frac{1}{2}m^{-2}$ implying that (88) holds with probability at least $1 - m^{-2}$, as it is claimed in the theorem.

To complete the proof, it is enough to use the interpolation inequality of Lemma 1. It follows that, for $q \in (1, 2)$,

$$\|\tilde{\rho}^{\varepsilon} - \rho\|_q \le \|\tilde{\rho}^{\varepsilon} - \rho\|_1^{\frac{2}{q}-1}\|\tilde{\rho}^{\varepsilon} - \rho\|_2^{2-\frac{2}{q}}.$$

Substituting bound (82) for $q = 1$ and $q = 2$ into the last inequality yields the result for an arbitrary $q \in (1, 2)$. ■

Similarly, in the case of trace regression with bounded response (see Assumption 3), minimax rates of Theorem 7 are also attained for the estimator $\tilde{\rho}^{\varepsilon}$ (up to log factors). In this case, assume that Assumption 3 holds with $\bar{U} = U$ and, in addition, let us make the following simplifying assumptions:

$$U\sqrt{\frac{m\log m}{n}} \lesssim 1 \quad \text{and} \quad \log\log_2 n \lesssim m\log m. \tag{89}$$

For the Pauli basis ($U = m^{-1/2}$), the first assumption holds if $n \gtrsim \log m$. The second assumption does hold unless n is extremely large ($n \sim 2^{\exp\{m \log m\}}$). Under these assumptions, we will use the following value of regularization parameter ε :

$$\varepsilon := \frac{U}{\log(mn)} \sqrt{\frac{\log(2m)}{nm}}.$$

The following version of Theorem 21 holds in the bounded regression case (with a similar proof).

Theorem 24 *There exists a constant $C > 0$ such that the following bounds hold for all $r = 1, \dots, m$, for all $\rho \in \mathcal{S}_{r,m}$ and for all $q \in [1, 2]$ with probability at least $1 - m^{-2}$:*

$$\|\tilde{\rho}^{\varepsilon} - \rho\|_q \le C \Big(\frac{U m^{\frac{3}{2}} r^{1/q}}{\sqrt{n}} \sqrt{\log m} \log^{(2-q)/q}(mn) \bigwedge \Big(\frac{U m^{3/2}}{\sqrt{n}} \Big)^{1-\frac{1}{q}} (\log m)^{\frac{1}{2}-\frac{1}{2q}} \Big) \bigwedge 2, \quad (90)$$

$$H^2(\tilde{\rho}^{\varepsilon}, \rho) \le C \frac{U m^{\frac{3}{2}} r}{\sqrt{n}} \sqrt{\log m} \log(mn) \bigwedge 2 \quad (91)$$

and

$$K(\rho \| \tilde{\rho}^{\varepsilon}) \le C \frac{U m^{\frac{3}{2}} r}{\sqrt{n}} \sqrt{\log m} \log(mn). \quad (92)$$

Remark 25 *In the case of Pauli basis, the minimax optimal rates (up to constants and logarithmic factors) are: $\frac{m r^{1/q}}{\sqrt{n}} \wedge (\frac{m}{\sqrt{n}})^{1-\frac{1}{q}} \wedge 2$ for Schatten q-norm distances for $q \in [1, 2]$; $\frac{mr}{\sqrt{n}}$ for nuclear norm, squared Hellinger and Kullback-Leibler distances (provided the $mr \lesssim \sqrt{n}$).*

References

Jean-Pierre Aubin and Ivar Ekeland. *Applied Nonlinear Analysis*. Courier Corporation, 2006.

Guillaume Aubrun. On almost randomizing channels with a short Kraus decomposition. *Communications in Mathematical Physics*, 288(3):1103–1116, 2009.

Vladimír Bužek. Quantum tomography from incomplete data via maxent principle. In *Quantum State Estimation*, pages 189–234. Springer, 2004.

Tony Cai, Donggyu Kim, Yazhen Wang, Ming Yuan, and Harrison H Zhou. Optimal large-scale quantum state tomography with Pauli measurements. `http://www-stat.wharton.upenn.edu/~tcai/paper/Estimating-Density-Matrix-Pauli.pdf`, 2015.

Emmanuel J Candés and Yaniv Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Transactions on Information Theory*, 57(4):2342–2359, 2011.

Victor H. de la Peña and Evarist Giné. *Decoupling. From Dependence to Independence*. Springer, 1999.

Steven T Flammia, David Gross, Yi-Kai Liu, and Jens Eisert. Quantum tomography via compressed sensing: error bounds, sample complexity and efficient estimators. *New Journal of Physics*, 14(9):095022, 2012.

David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

David Gross, Yi-Kai Liu, Steven T Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical Review Letters*, 105(15):150401, 2010.

Olivier Guédon, Shahar Mendelson, Alain Pajor, and Nicole Tomczak-Jaegermann. Majorizing measures and proportional subsets of bounded orthonormal systems. *Revista Matemática Iberoamericana*, 24(3):1075–1095, 2008.

Amir Kalev, Robert L Kosut, and Ivan H Deutsch. Informationally complete measurements from compressed sensing methodology. *arXiv preprint arXiv:1502.00536*, 2015.

Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. Interaction in quantum communication. *IEEE Transactions on Information Theory*, 53(6):1970–1982, 2007.

Vladimir Koltchinskii. von Neumann entropy penalization and low-rank matrix estimation. *The Annals of Statistics*, 39(6):2936–2973, 2011a.

Vladimir Koltchinskii. *Oracle Inequalities in Empirical Risk Minimization and Sparse Recovery Problems: École d'Été de Probabilités de Saint-Flour XXXVIII-2008*. Springer, 2011b.

Vladimir Koltchinskii. A remark on low rank matrix recovery and noncommutative Bernstein type inequalities. In *From Probability to Statistics and Back: High-Dimensional Models and Processes–A Festschrift in Honor of Jon A. Wellner*, pages 213–226. Institute of Mathematical Statistics, 2013a.

Vladimir Koltchinskii. Sharp oracle inequalities in low rank estimation. In *Empirical Inference*, pages 217–230. Springer, 2013b.

Vladimir Koltchinskii and Dong Xia. Schatten p-norm distances in low rank density matrix estimation. 2015+.

Vladimir Koltchinskii, Karim Lounici, and Alexandre B Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

Yi-Kai Liu. Universal low-rank matrix recovery from Pauli measurements. In *Advances in Neural Information Processing Systems*, pages 1638–1646, 2011.

Zongming Ma and Yihong Wu. Volume ratio, sparsity, and minimaxity under unitarily invariant norms. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium*, pages 1027–1031. IEEE, 2013.

Sahand Negahban and Martin J. Wainwright. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 2010.

M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

Alain Pajor. Metric entropy of the Grassmann manifold. *Convex Geometric Analysis*, 34: 181–188, 1998.

Angelika Rohde and Alexandre B Tsybakov. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887–930, 2011.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation.* Springer, 2008.

Journal of Machine Learning Research 16 (2015) 1793-1861 Submitted 7/15; Published 9/15

Fast Rates in Statistical and Online Learning

Tim van Erven* TIM@TIMVANERVEN.NL
Mathematisch Instituut, Universiteit Leiden
Leiden, 2300 RA, The Netherlands

Peter D. Grünwald PETER.GRUNWALD@CWI.NL
Centrum voor Wiskunde en Informatica and MI, Universiteit Leiden
Amsterdam, NL-1090 GB, The Netherlands

Nishant A. Mehta† MEHTA@CWI.NL
Centrum voor Wiskunde en Informatica
Amsterdam, NL-1090 GB, The Netherlands

Mark D. Reid MARK.REID@ANU.EDU.AU
Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
Australian National University and NICTA
Canberra, ACT 2601 Australia.

Editor: Alex Gammerman and Vladimir Vovk

Abstract

The speed with which a learning algorithm converges as it is presented with more data is a central problem in machine learning — a fast rate of convergence means less data is needed for the same level of performance. The pursuit of fast rates in online and statistical learning has led to the discovery of many conditions in learning theory under which fast learning is possible. We show that most of these conditions are special cases of a single, unifying condition, that comes in two forms: the *central condition* for 'proper' learning algorithms that always output a hypothesis in the given model, and *stochastic mixability* for online algorithms that may make predictions outside of the model. We show that under surprisingly weak assumptions both conditions are, in a certain sense, equivalent. The central condition has a re-interpretation in terms of convexity of a set of pseudoprobabilities, linking it to density estimation under misspecification. For bounded losses, we show how the central condition enables a direct proof of fast rates and we prove its equivalence to the *Bernstein* condition, itself a generalization of the *Tsybakov margin condition*, both of which have played a central role in obtaining fast rates in statistical learning. Yet, while the Bernstein condition is two-sided, the central condition is one-sided, making it more suitable to deal with unbounded losses. In its stochastic mixability form, our condition generalizes both a *stochastic exp-concavity* condition identified by Juditsky, Rigollet and Tsybakov and Vovk's notion of *mixability*. Our unifying conditions thus provide a substantial step towards a characterization of fast rates in statistical learning, similar to how classical mixability characterizes constant regret in the sequential prediction with expert advice setting.

Keywords: statistical learning theory, fast rates, Tsybakov margin condition, mixability, exp-concavity

∗. Authors listed alphabetically. Preliminary versions of some parts of this work were presented at NIPS 2012 and at NIPS 2014 (see acknowledgments on page 1845).

†. Work performed while at ANU and NICTA.

1. Introduction

Alexey Chervonenkis jointly achieved several significant milestones in the theory of machine learning: the characterization of uniform convergence of relative frequencies of events to their probabilities (Vapnik and Chervonenkis, 1971), the uniform convergence of means to their expectations (Vapnik and Chervonenkis, 1981), and the 'key theorem in learning theory' showing the relationship between the consistency of empirical risk minimization (ERM) and the uniform one-sided convergence of means to expectations (Vapnik and Chervonenkis, 1991); (Vapnik, 1998, Chapter 3). Two outstanding features of these contributions are that they *characterized* the phenomenon in question, and the quantitative results are *parametrization independent* in the sense that they do not depend upon how elements of the hypothesis class $\mathcal{F}$ are parameterized, only on global (effectively geometric) properties of $\mathcal{F}$. With his co-author Vladimir Vapnik, Alexey Chervonenkis also presented quantitative bounds on the deviation between the empirical and expected risk as a function of the sample size n. These are used for the theoretical analysis of the statistical convergence of ERM algorithms, which are central to machine learning. According to Vapnik (1998, p. 695), in his 1974 book co-authored by Chervonenkis (Vapnik and Chervonenkis, 1974) they presented 'slow' and 'fast' bounds for ERM when used with 0-1 loss. They showed that in the realizable or 'optimistic' case (where there is an $f \in \mathcal{F}$ that almost surely predicts correctly, so that the minimum achievable risk is zero) one can achieve fast $O(1/n)$ convergence as opposed to the 'pessimistic' case where one does not have such an f in the hypothesis class and the best *uniform* bound is $O(1/\sqrt{n})$ (Vapnik, 1998, page 127). This difference is important because if one is in such a 'fast rate' regime, one can achieve good performance with less data.

The present paper makes several further contributions along this path first delineated by Vapnik and Chervonenkis. We focus upon the distinction between slow and fast learning. As shown in the special case of squared loss by Lee et al. (1998) and log loss by Li (1999), if the hypothesis class is *convex*, one can still attain fast $O(1/n)$ convergence even in the agnostic (pessimistic) setting.[1] Such convergence results, like those of Vapnik and Chervonenkis, are uniform — they hold for all possible target distributions. When the hypothesis class is not convex, one cannot attain a uniform fast bound for ERM (Mendelson, 2008a), and it is not known whether fast rates are possible for any algorithm at all; however, one can obtain a *non-uniform* bound (Mendelson and Williamson, 2002; Mendelson, 2008b). Such bounds are necessarily dependent upon the relationships between the components $(\ell, \mathcal{P}, \mathcal{F})$ of a statistical decision problem or learning task. Here ℓ is the loss, $\mathcal{F}$ the hypothesis class, and $\mathcal{P}$ the (possibly singleton) class of distributions which, by assumption, contains the unknown data-generating distribution. Often one can assume large classes of $\mathcal{P}$ and still obtain bounds that are *relatively* uniform, i.e. uniform over all $P \in \mathcal{P}$. We identify a *central condition* on decision problems $(\ell, \mathcal{P}, \mathcal{F})$ — where ℓ may be unbounded — that, in its strongest form, allows $O(1/n)$ rates for so-called 'proper' learning algorithms that always output a member of $\mathcal{F}$. In weaker forms, it allows rates in between $O(1/\sqrt{n})$ and $O(1/n)$.

1. Throughout this work, implicit in our statements about rates is that the function class is not too large; we assume classes with at most logarithmic universal metric entropy, which includes finite classes, VC classes, and VC-type classes.

As a second contribution, we connect the above line of work (within the traditional stochastic setting) to a parallel development in the worst-case online sequence prediction setting. There, one makes no probabilistic assumptions at all, and one measures convergence of the regret, that is, the difference between the cumulative loss attained by a given algorithm on a particular sequence with the best possible loss attainable on that sequence (Cesa-Bianchi and Lugosi, 2006). This work, due in large part to Vovk (1990, 1998, 2001), shares one aspect of Vapnik and Chervonenkis' approach — it achieves a *characterization* of when fast learning is possible in the online individual sequence-setting. Since there is no $\mathcal{P}$ in this setting, the characterization depends only upon the loss ℓ, and in particular whether the loss is *mixable*. As shown in Section 4, our second key condition, *stochastic mixability*, is a generalization of Vovk's earlier notion. Briefly, when $\mathcal{P}$ is the set of all distributions on a domain, stochastic mixability is equivalent to Vovk's classical mixability. Stochastic mixability of $(\ell, \mathcal{P}, \mathcal{F})$ for general $\mathcal{P}$ then indicates that fast rates are possible in a stochastic on-line setting, in the worst-case over all $P \in \mathcal{P}$.

The main contribution in this paper is to show, first, that a range of existing conditions for fast rates (such as the Bernstein condition, itself a generalization of the Tsybakov condition) are either special cases of our central condition, or special cases of stochastic mixability (such as original mixability and (stochastic) exp-concavity); and second, to show that under surprisingly weak conditions the central condition and stochastic mixability are in fact equivalent — thus there emerges essentially a *single* condition that implies fast rates in a wide variety of situations. Our central and stochastic mixability condition improve in several ways on the existing conditions that they generalize and unify. For example, like the uniform convergence condition in Vapnik and Chervonenkis' original 'key theorem of learning theory' (Vapnik and Chervonenkis, 1991), but unlike the Bernstein fast rate condition, our conditions are *one-sided* which, as forcefully argued by Mendelson (2014), seems as it should be; Example 5.7 explains and illustrates the difference between the two- and one-sided conditions. Like Vapnik and Chervonenkis' uniform convergence condition and Vovk's classical mixability, but unlike the stochastic and individual-sequence exp-concavity conditions, our conditions are *parametrization independent* (Section 4.2.2). Finally, unlike the assumptions for classical mixability (Vovk, 1998), we do not require compactness of the loss function's domain. We hasten to add though that for unbounded losses, several important issues are still unresolved — for example, if under some $P \in \mathcal{P}$ and with some $f \in \mathcal{F}$ the distribution of the loss has polynomial tails, then some of our equivalences break down (Section 5.2).

One final historical precursor deserves mention. Statistical convergence bounds rely on bounds on the tails of certain random variables. In Section 7 we show how, for bounded losses, the central condition (4) directly controls the behaviour of the cumulant generating function of the excess loss random variable. The geometric insight behind this result, Figure 3, previously was used, unbeknownst to us when carrying out the work originally (Mehta and Williamson, 2014), by Claude Shannon (1956). It is fitting that our tribute to Alexey Chervonenkis can trace its history to another such giant of the theory of information processing.

1.1 Why Read This Paper? Our Most Important Results

Below, we highlight the core contributions of this work. A more comprehensive overview is in Section 2 and the diagram on page 1798, which summarizes all results from the paper.

- We introduce the *v-stochastic mixability* condition on decision problems (Equation 8, Definition 4.1 and 5.9), a strict generalization of Vovk's *classical mixability* (Vovk, 1990, 1998, 2001; van Erven et al., 2012a), *exp-concavity* (Kivinen and Warmuth, 1999; Cesa-Bianchi and Lugosi, 2006) and *stochastic exp-concavity*, a condition identified implicitly by Juditsky et al. (2008) and used by e.g. Dalalyan and Tsybakov (2012). Here $v : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a nondecreasing nonnegative function. In the important special case that $v \equiv \eta$ is constant, we say that *(strong) stochastic mixability holds*. Proposition 4.5 shows that in that case, with finite $\mathcal{F}$, Vovk's aggregating algorithm for on-line prediction in combination with an online-to-batch conversion achieves a learning rate of $O(1/n)$; if the v-condition holds for sublinear v with $v(0) = 0$, intermediate rates between $O(1/\sqrt{n})$ and $O(1/n)$ are obtained. These results hold under no further conditions at all, in particular for unbounded losses. *Interest:* the condition being a strict generalization of earlier ones, it shows that we can get fast rates for some situations for which this was was hitherto unknown.

- We introduce the *v-central condition* (Equations 4, 5, 6, 10, Definitions 3.1 and 5.3). As we show in Theorem 5.4, for bounded losses and v of the form $v(x) = Cx^\alpha$, it generalizes the *Bernstein condition* (Bartlett and Mendelson, 2006), itself a generalization of the *Tsybakov margin condition* (Tsybakov, 2004). If $v \equiv \eta$ is constant, we just say that the (strong) *central condition holds*. In that case, with (unbounded) log-loss, it generalizes a (typically nameless) condition used to obtain fast rates in Bayesian and *minimum description length* (MDL) density estimation in misspecification contexts (Li, 1999; Zhang, 2006a,b; Kleijn and van der Vaart, 2006; Grünwald, 2011; Grünwald and van Ommen, 2014). These are all conditions that allow for fast rates for *proper* learning, in which the learning algorithm always outputs an element of $\mathcal{F}$.

  (i) For convex $\mathcal{F}$, we prove that the strong η-central condition and the strong η-stochastic mixability are equivalent, under weak conditions (Theorem 4.17 in conjunction with Proposition 4.11 and Theorem 3.10 in conjunction with Proposition 4.12). *Interest:* This shows that existing fast rate conditions for $O(1/n)$ rates in online learning are related to fast rate conditions for $O(1/n)$ rates for proper learning algorithms such as ERM — even though such conditions superficially look very different and have very different interpretations: existence of a 'substitution function' (mixability) vs. the exponential moment of a loss difference constituting a supermartingale (central condition).

  (ii) We prove (a) that for bounded losses, the strong central condition always implies fast $O(1/n)$ rates for ERM and the v-central condition implies intermediate rates (Theorem 7.6). The equivalence between η-mixability and the central condition and Proposition 4.5 mentioned above imply that, (b), the central condition implies fast rates in many more conditions, even with unbounded losses. We also show (c) that there exist decision problems with unbounded losses in which the central condition holds, the Bernstein condition does not hold, and we do get fast rates. *Interest:*

first, while fast and intermediate rates under the v-central condition with bounded loss can also be derived from existing results, our proof is directly in terms of the central condition and yields better constants. Second, results (a)-(c) above lead us to *conjecture* that there exist some very weak condition (much weaker than bounded loss) such that for sublinear v, the v-central condition together with this extra condition *always* implies sublinear rates. Establishing such a result is a major goal for future work.

- Under mild conditions, the v-central condition is equivalent to a third condition, the *pseudoprobability convexity (PPC) condition* — (7) and Definition 3.2 and 5.3. *Interest:* for the constant $v \equiv \eta$ case ($O(1/n)$ rates), the PPC condition provides a clear *geometric* and a *data-compression* interpretation of the v-central condition. For bounded losses and general v, it implies that a problem must have unique minimizers in a certain sense (Proposition 5.11), giving further insight into the fast rates phenomenon.

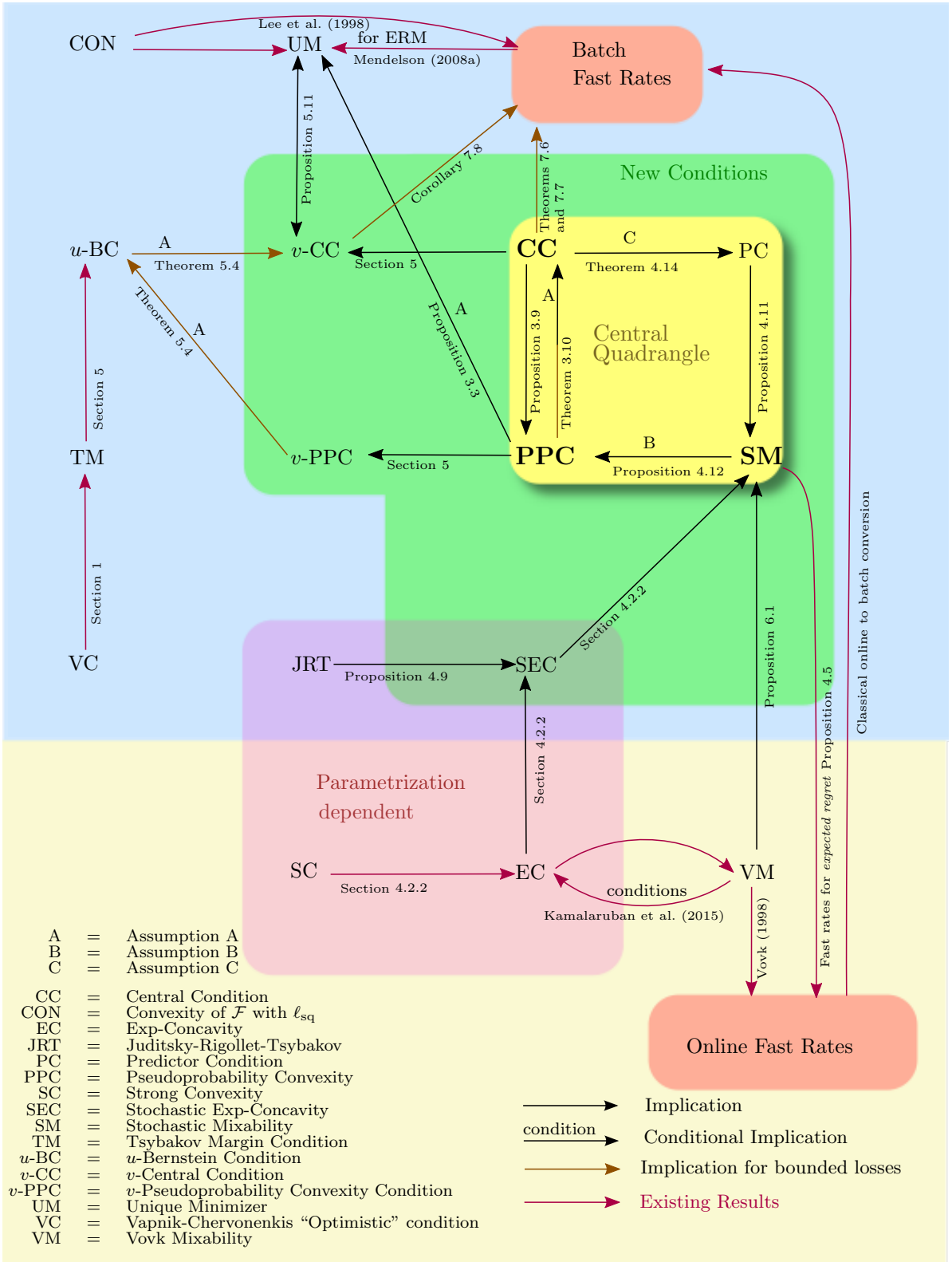
- In some cases with nonconvex $\mathcal{F}$, ERM and other proper learning algorithms achieve a suboptimal $O(1/\sqrt{n})$ rate, whereas online methods combined with an online-to-batch convergence get $O(1/n)$ rates in expectation (Audibert, 2007). Now the implication 'strong stochastic mixability $\Rightarrow$ strong central condition ' (Theorem 3.10 in conjunction with Proposition 4.12, already mentioned under 2(i)) holds *whenever the risk minimizer within $\mathcal{F}$ coincides with the risk minimizer within the convex hull of $\mathcal{F}$.* Thus, as long as this is the case, there is no inherent rate advantage in improper learning — if η-stochastic mixability holds so that (improper) online methods achieve an $O(1/n)$-rate, so will the (proper) ERM method. Theorem 7.6 implies this for bounded losses; we conjecture that the same holds for unbounded losses. *Interest:* This insight helps understand when improper learning can and cannot be helpful for general losses, something that was hitherto only well-understood for the squared loss on a bounded domain (Lecué, 2011).

2. Introduction to and Overview of Results

To facilitate reading of this long paper, we provide an introductory summary of all our results. By reading this section alongside the 'map' of conditions and their relationships on page 1798, the reader should get a good overview of our results. We start below with some notational and conceptual preliminaries, and continue in Section 2.2 with a discussion of the central condition, followed by a section-by-section description of the paper.

2.1 Decision Problems and Risk

We consider decision problems which, in their most general form, can be specified as a four-tuple $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ where $\mathcal{P}$ is a set of distributions on a sample space $\mathcal{Z}$, and the goal is to make decisions that are essentially as good as the best decision in the *model* $\mathcal{F}$ ($\mathcal{F}$ is often called an 'hypothesis space' in machine learning). We will allow the decision maker to make decisions in a *decision set* $\mathcal{F}_{\mathrm{d}}$ which is usually taken equal to, or a superset of, $\mathcal{F}$ but for mathematical convenience is also allowed to be a subset of $\mathcal{F}$. The quality of


CON
Lee et al. (1998)
UM
for ERM
Mendelson (2008a)
Batch
Fast Rates
Proposition 5.11
Corollary 7.8
Theorems 7.6 and 7.7
New Conditions
u-BC
A
Theorem 5.4
v-CC
Section 5
CC
C
Theorem 4.14
PC
Theorem 5.4
A
A
Proposition 3.3
Proposition 3.9
A
Theorem 3.10
Central Quadrangle
Proposition 4.11
Section 5
TM
v-PPC
Section 5
PPC
B
Proposition 4.12
SM
Section 1
VC
JRT
Proposition 4.9
SEC
Section 4.2.2
Proposition 6.1
Classical online to batch conversion
Parametrization dependent
Section 4.2.2
SC
Section 4.2.2
EC
conditions
Kamalaruban et al. (2015)
VM
Vovk (1998)
Fast rates for expected regret Proposition 4.5
Online Fast Rates
A = Assumption A
B = Assumption B
C = Assumption C
CC = Central Condition
CON = Convexity of $\mathcal{F}$ with ℓ_{sq}
EC = Exp-Concavity
JRT = Juditsky-Rigollet-Tsybakov
PC = Predictor Condition
PPC = Pseudoprobability Convexity
SC = Strong Convexity
SEC = Stochastic Exp-Concavity
SM = Stochastic Mixability
TM = Tsybakov Margin Condition
u-BC = u-Bernstein Condition
v-CC = v-Central Condition
v-PPC = v-Pseudoprobability Convexity Condition
UM = Unique Minimizer
VC = Vapnik-Chervonenkis "Optimistic" condition
VM = Vovk Mixability
Implication
condition
Conditional Implication
Implication for bounded losses
Existing Results

decisions will be measured by a *loss* function $\ell\colon \mathcal{F}_\ell \times \mathcal{Z} \to [-B, \infty]$ for arbitrary $B \geq 0$ where a smaller loss means better predictions, and $\mathcal{F}_\ell \supseteq \mathcal{F} \cup \mathcal{F}_\mathrm{d}$ is the *domain* of the loss. As further notation we introduce the component functions $\ell_f(z) = \ell(f, z)$ and for any set $\mathcal{G}$ we let $\Delta(\mathcal{G})$ denote the set of distributions on $\mathcal{G}$ (implicitly assuming that $\mathcal{G}$ is a measurable set, equipped with an appropriate σ-algebra). A loss function ℓ is called *bounded* if for some $B \geq 0$, for all $f \in \mathcal{F}_\ell$ and all $P \in \mathcal{P}$, we have $|\ell_f(Z)| \leq B$ almost surely when $Z \sim P$. When $\mathcal{F}_\ell$ is a set for which this is well-defined, for any $\mathcal{F} \subset \mathcal{F}_\ell$ we denote by $\mathrm{co}(\mathcal{F}) \subseteq \mathcal{F}_\ell$ the convex hull of $\mathcal{F}$.

Now fix some decision problem $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_\mathrm{d})$. The *risk* of a predictor $f \in \mathcal{F}_\ell$ with respect to $P \in \mathcal{P}$ is defined, as usual, as

$$R(P, f) = \mathop{\mathbf{E}}_{Z \sim P}[\ell_f(Z)], \tag{1}$$

where Z is a random variable mapping to outcomes in $\mathcal{Z}$ and, in general, $R(P, f)$ may be infinite. However, for the remainder of the paper we will only consider tuples $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_\mathrm{d})$ such that for all $P \in \mathcal{P}$, there exists[2] at least one $f^\circ \in \mathcal{F}$ with $R(P, f^\circ) < \infty$ and hence $P(\ell_{f^\circ}(Z) = \infty) = 0$. A *learning algorithm* or *estimator* is a (computable) function from $\cup_{n \geq 0} \mathcal{Z}^n$ to $\mathcal{F}_\mathrm{d}$ that, upon observing data $Z_1, \ldots, Z_n$, outputs some $\hat{f}_n \in \mathcal{F}_\mathrm{d}$. Following standard terminology, we call a learning algorithm *proper* (Lee et al., 1996; Alekhnovich et al., 2004; Urner and Ben-David, 2014) if its outputs are restricted to the set $\mathcal{F}$, i.e. $\mathcal{F} = \mathcal{F}_\mathrm{d}$. Examples of this setting, which has also been called *in-model estimation* (Grünwald and van Ommen, 2014), include ERM and Bayesian *maximum a posteriori* (MAP) density estimation. For notational convenience, in such cases we identify a decision problem with the triple $(\ell, \mathcal{P}, \mathcal{F})$. We only consider $\mathcal{F} \neq \mathcal{F}_\mathrm{d}$ in Section 4 and 6 on on-line learning, where $\mathcal{F}_\mathrm{d}$ is often taken to be $\mathrm{co}(\mathcal{F})$; for example, $\mathcal{F}$ may be a set of probability densities (Example 2.2) and the algorithm may be Bayesian prediction, which predicts with the Bayes predictive distribution (Section 3.3), a mixture of elements of $\mathcal{F}$ which is hence in $\mathrm{co}(\mathcal{F})$. One of our main insights, discussed in Section 4.3.3, is understanding when the *weaker* conditions that allow fast rates for improper learning transfer to the proper learning setting. In the stochastic setting, the *rate* (in expectation) of a learning algorithm is the quantity

$$\sup_{P \in \mathcal{P}} \left\{ \mathop{\mathbf{E}}_{\mathbf{Z} \sim P} \left[R(P, \hat{f}_n) \right] - \inf_{f \in \mathcal{F}} R(P, f) \right\}, \tag{2}$$

where $\mathbf{Z} = (Z_1, \ldots, Z_n)$ are n i.i.d. copies of Z. The rate of a learning algorithm can usually be bounded, up to $\log n$ factors, as $(\mathrm{COMP}_n(\mathcal{F})/n)^\alpha$ for some α between $1/2$ and 1. Here $\mathrm{COMP}_n(\mathcal{F})$ is some measure of the complexity of $\mathcal{F}$ which may or may not depend on n, such as its codelength, its VC-dimension in classification, an upper bound on the KL-divergence between prior and posterior in PAC-Bayesian approaches, or the logarithm of the number of elements of an ε-net, with ε determined by sample size, and so on. In the simplest case, with $\mathcal{F}$ finite, complexity is invariably bounded independently of n (usually as $\log |\mathcal{F}|$), and whenever for a decision problem $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_\mathrm{d})$ with finite $\mathcal{F}$ there exists a learning algorithm achieving the rate $O(1/n)$, we say that the problem *allows for fast rates*.

In the remainder of this section we make the following simplifying assumption.

2. We allow the loss itself to be infinite which makes random variables and their expectations undefined when they evaluate to $\infty - \infty$ with positive probability. The requirement that f° exists for all P ensures that we never encounter this situation in any of our formulas.

Assumption A (Minimal Risk Achieved) *For all $P \in \mathcal{P}$, the minimal risk $R(P, f)$ over $\mathcal{F}$ is achieved by some $f^* \in \mathcal{F}$ depending on P, i.e.*

$$R(P, f^*) = \inf_{f \in \mathcal{F}} R(P, f). \tag{3}$$

Assumption A is essentially a closure property that holds in many cases of interest. We will call such f^* *$\mathcal{F}$-optimal for P* or simply *$\mathcal{F}$-optimal*. When $P \in \mathcal{P}$ and $\mathcal{F}$ are clear from context, we will also simply say that f^* is the *best predictor*.

Example 2.1 (Regression, Classification, (Relatively) Well-Specified and Misspecified Models) In the standard statistical learning problems of *classification* and *regression*, we have $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ for some 'feature' or 'covariate' space $\mathcal{X}$ and $\mathcal{F}$ is a set of functions from $\mathcal{X}$ to $\mathcal{Y}$. In classification, $\mathcal{Y} = \{0, 1\}$ and one usually takes the standard classification loss $\ell_f^{\text{class}}((x, y)) = |y - f(x)|$; in regression, one takes $\mathcal{Y} = \mathbb{R}$ and the squared error loss $\ell_f^{\text{reg}}((x, y)) = \frac{1}{2}(y - f(x))^2$. In Example 2.2 we show that density estimation also fits in our setting. For losses with bounded range $[0, B]$, if the optimal f^* that exists by Assumption A has 0 risk, we are in what Vapnik and Chervonenkis (1974) call the 'optimistic' setting, more commonly known as the 'deterministic' or 'realizable' case (VC in Figure 1 on page 1798). We never make this strong an assumption and are thus always in the 'agnostic' case. A strictly weaker assumption would be to assume that f^* is the Bayes decision rule, minimizing the risk $R(P, f^*)$ over the loss function's full domain $\mathcal{F}_\ell$; in classification this means that f^* is the *Bayes classifier* (minimizing risk over all functions from $\mathcal{X}$ to $\mathcal{Y}$), in regression it implies that f^* is the *true regression function*, i.e. $f^*(x) = \mathbf{E}_{(X,Y)\sim P}[Y \mid X = x]$, in density estimation (see below) that f^* is the density of the 'true' P. Borrowing terminology from statistics, we then say that the model $\mathcal{F}$ is *well-specified*, or simply *correct*. Although this assumption is often made in statistics and sometimes in statistical learning (e.g. in the original Tsybakov condition (Tsybakov, 2004) and in the analysis of strictly convex surrogate loss functions for 0/1-loss (Bartlett et al., 2006)), all of our results are applicable to incorrect, *misspecified* $\mathcal{F}$ as well. We will, however, in some cases make the much weaker Assumption B (page 1820) that $\mathcal{F}$ is well-specified *relative to* $\mathcal{F}_{\text{d}}$, or equivalently $\mathcal{F}$ is *as good as* $\mathcal{F}_{\text{d}}$, meaning that for all $P \in \mathcal{P}$, $\min_{f \in \mathcal{F}_{\text{d}}} R(P, f) = \min_{f \in \mathcal{F}} R(P, f)$. In all our examples, if $\mathcal{F} \neq \mathcal{F}_{\text{d}}$ we can take, without loss of generality, $\mathcal{F}_{\text{d}} = \text{co}(\mathcal{F})$, and then a sufficient (but by no means necessary) condition for relative well-specification is that $\mathcal{F}$ is either convex or correct. ■

We now turn to an overview of the main results and concepts of this paper, which are also highlighted in Figure 1 on page 1798.

2.2 Main Concept: The Central Condition

We focus on decision problems $(\ell, \mathcal{P}, \mathcal{F})$ satisfying the simplifying Assumption A by fixing any such decision problem and letting $P \in \mathcal{P}$ and f^* be $\mathcal{F}$-optimal for P. We may now ask this f^* to satisfy a stronger, supermartingale-type property where for some $\eta > 0$ we require

$$\mathop{\mathbf{E}}_{Z \sim P}\left[e^{\eta\left(\ell_{f^*}(Z) - \ell_f(Z)\right)}\right] \leq 1 \qquad \text{for all } f \in \mathcal{F}. \tag{4}$$

This type of property plays a fundamental role in the study of fast rates because it controls the higher moments of the negated excess loss $\ell_{f^*}(Z) - \ell_f(Z)$. Note that by our conventions regarding infinities (Section 2.1) this implies that $P(\ell_{f^*}(Z) = \infty) = 0$.

There are several motivations for studying the requirement in (4). In the case of classification loss, it can be seen to be a special, extreme case of the *Bernstein condition* (see below). In the case of log loss, the requirement becomes a standard (but usually unnamed) condition which we call the *Bayes-MDL Condition* which is used in proving convergence rates of Bayesian and MDL density estimation (Example 2.2). Finally, under a bounded loss assumption the condition (4) implies one our main results, Theorem 7.6, a fast rates result for statistical learning over finite classes (the situation for unbounded losses is more complicated and is discussed after Example 2.2).

Note that to satisfy Assumption A it is sufficient to require that the property (4) holds for *some* $f^* \in \mathcal{F}$ since, by Jensen's inequality, this f^* must then automatically be $\mathcal{F}$-optimal as in (3). We will require (4) to hold for all $P \in \mathcal{P}$ (where f^* may depend on P). This is the simplest form of our central condition, which we call the *the η-central condition*. We note that if (4) holds for all $f \in \mathcal{F}$ then it must also hold in expectation for all *distributions* on $\mathcal{F}$. Thus, the η-central condition can be restated as follows:

$$\forall P \in \mathcal{P}\ \exists f^* \in \mathcal{F}\ \forall \Pi \in \Delta(\mathcal{F}) : \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\Pi} \left[e^{\eta\left(\ell_{f^*}(Z) - \ell_f(Z)\right)} \right] \leq 1. \tag{5}$$

This rephrasing of the central condition will be useful when comparing it to conditions introduced later in the paper.

The central condition is easiest to interpret for density estimation with the logarithmic loss. In this case the condition for $\eta = 1$ is implied by $\mathcal{F}$ being either well-specified or convex, as the following example shows.

Example 2.2 (Density estimation under well-specified or convex models) Let $\mathcal{F}$ be a set of probability densities on $\mathcal{Z}$ and take ℓ to be log loss, so that $\ell_f(z) = -\log f(z)$.

For log loss, statistical learning becomes equivalent to density estimation. Satisfying the central condition then becomes equivalent to, for all $P \in \mathcal{P}$, finding an $f^* \in \mathcal{F}$ such that

$$\mathop{\mathbf{E}}_{Z\sim P} \left(\frac{f(Z)}{f^*(Z)} \right)^{\eta} \leq 1 \tag{6}$$

for all $f \in \mathcal{F}$. If the model $\mathcal{F}$ is correct, it trivially holds that $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the 1-central condition as we choose f^* to be the density of P, so that the densities in the expectation and the denominator cancel. Even when the model is misspecified, Li (1999) showed that (6) holds for $\eta = 1$ provided the model is convex. We will recover this result in Example 3.12 in Section 3, where we review the central role that (6) plays in convergence proofs of MDL and Bayesian estimation. Even if the set of densities is neither correct nor convex, the central condition often still holds for some $\eta \neq 1$. In Example 3.6 we explore this for the set of normal densities with variance τ^2 when the true distribution is either Gaussian with a different variance, or subgaussian. ■

We show in Section 7 that for bounded losses the η-central condition implies fast $O(1/n)$ rates for finite $\mathcal{F}$. But what about unbounded losses such as log loss? In the log loss/density

estimation case, as shown by Barron and Cover (1991); Zhang (2006a); Grünwald (2007) and others, fast rates can be obtained in a weaker sense. Specifically, in the worst-case over $P \in \mathcal{P}$, the squared Hellinger distance or Rényi divergences between $\hat{f}_n$ and the optimal f^* converge as $O(1/n)$ for ERM when $\mathcal{F}$ is finite, and like $O(\text{COMP}_n/n)$ for general $\mathcal{F}$ and for 2-part MDL and Bayes MAP-style algorithms. If the goal is to obtain fast rates in the stronger sense (2) for general unbounded loss functions some additional assumptions are needed. Zhang (2006a,b) provides such results for penalized ERM and randomized estimators (see also the discussion in Section 8). Importantly, as explained by Grünwald (2012), the proofs for fast rates in all the works mentioned here crucially, though sometimes implicitly, employ the η-central condition at some point.

2.3 Overview of the Paper

Section 3 —Fast Rates for Proper Learning: PPC Condition, Bayesian Interpretation, Relation to Bayes-MDL Condition.

In Section 3, we give a second condition, the *pseudoprobability convexity (PPC) condition*, a variation of (5) stating that:

$$\forall P \in \mathcal{P}\ \forall \Pi \in \Delta(\mathcal{F})\ \exists f^* \in \mathcal{F}:\ \mathop{\mathbf{E}}_{Z\sim P}[\ell_{f^*}(Z)] \leq \mathop{\mathbf{E}}_{Z\sim P}\left[-\frac{1}{\eta}\log \mathop{\mathbf{E}}_{f\sim\Pi} e^{-\eta \ell_f(Z)}\right]. \tag{7}$$

Clearly, if the condition holds, then it will hold by choosing, for every $P \in \mathcal{P}$, f^* to be $\mathcal{F}$-optimal relative to P. The name 'pseudoprobability' stems from the interpretation of $p_f(Z) := e^{-\ell_f(Z)}$ as 'pseudo-probability associated with f, similar to the 'entropification' of f introduced by Grünwald (1999). The full 'pseudoprobability convexity' stems from the interpretation illustrated by and explained around Figure 2 on page 1813. We show that, under simplifying Assumption A, the central and PPC conditions are equivalent. One direction of this equivalence is trivial, while the other direction is our first main result, Theorem 3.10. We also explain how the rightmost expression in (7) strongly resembles the expected log-loss of a Bayes predictive distribution, and how this leads to a 'pseudo-Bayesian' or 'pseudo-data compression' interpretation of the pseudoprobability convexity condition, and hence of the central condition. Versions of this interpretation were highlighted earlier by Grünwald (2012); Grünwald and van Ommen (2014). Thus, we can think of both conditions as a single condition with dual interpretations: a frequentist one in terms of exponentially small deviation probabilities (which follow by applying Markov's inequality to $\mathbf{E}_{Z\sim P}[e^{\eta(\ell_{f^*(Z)} - \ell_f(Z))}]$), and a pseudo-Bayesian one in terms of convexity properties of $\mathcal{F}$. Further, we give a few more examples of the central/PPC condition in this section, and we discuss in detail its special case, the Bayes-MDL condition (Example 2.2).

Crucially, all algorithms that we are aware of for which fast rates have been proven by means of the η-central condition are 'proper' in that they always output a (possibly randomized) element of $\mathcal{F}$ itself. This includes ERM, two-part MDL, Bayes MAP and randomized Bayes algorithms (Barron and Cover, 1991; Zhang, 2006a,b; Grünwald, 2007) and PAC-Bayesian methods (Audibert, 2004; Catoni, 2007). Thus, the central condition is appropriate for *proper learning*. This is in contrast to the stochastic mixability condition which is defined and studied in Section 4.

Section 4 — Fast Rates for Online Learning: (Stochastic) Mixability and Exp-Concavity.

In online learning with bounded losses, *strong convexity* of the loss is an oft-used condition to obtain fast rates because it is naturally related to gradient and mirror descent methods (Hazan et al., 2007, 2008; Shalev-Shwartz and Singer, 2007). If we allow more general algorithms, however, then fast rates are also possible under the condition of *exp-concavity* which is weaker than strong convexity (Hazan et al., 2007). Exp-concavity in turn is a special case of Vovk's classical mixability condition (Vovk, 2001), the main difference being that the definition of exp-concavity depends on the choice of parametrization of the loss function whereas the definition of classical mixability does not. Whether classical mixability can really be strictly weaker than exp-concavity in an 'optimal' parametrization is an open question (Kamalaruban et al., 2015; van Erven, 2012). Strong convexity, exp-concavity and classical mixability are all individual sequence notions, allowing for fast rates in the sense that, if $\mathcal{F}$ is finite, then there exist (improper) learning algorithms for which the worst-case cumulative regret over all sequences, that is $\sup_{z_1,\ldots,z_n \in \mathcal{Z}^n} \left\{ \sum_{i=1}^n \left(\ell_{\hat{f}_{i-1}}(z_i) \right) - \inf_{f \in \mathcal{F}} \sum_{i=1}^n \ell_f(z_i) \right\}$, is bounded by a constant. This implies that the worst-case cumulative regret per outcome at time n is $O(1/n)$.

One may obtain learning algorithms for statistical learning by converting algorithms for online learning using a process called *online-to-batch conversion* (Cesa-Bianchi et al., 2004; Barron, 1987; Yang and Barron, 1999). This process preserves rates, in the sense that if the worst-case regret per outcome at time n of a method is r_n then the rate of the resulting learning algorithm in the sense of (2) will also be r_n. However, for this purpose, it suffices to use a much weaker stochastic analogue of mixability that only holds in expectation instead of holding for all outcomes. This analogue is *η-stochastic mixability*, which we define (note the similarity to (7)) as

$$\forall \Pi \in \Delta(\mathcal{F}) \; \exists f^* \in \mathcal{F}_{\mathrm{d}} \; \forall P \in \mathcal{P} : \; \mathop{\mathbf{E}}_{Z \sim P}[\ell_{f^*}(Z)] \leq \mathop{\mathbf{E}}_{Z \sim P} \left[-\frac{1}{\eta} \log \mathop{\mathbf{E}}_{f \sim \Pi} e^{-\eta \ell_f(Z)} \right]. \tag{8}$$

Under this condition, Vovk's Aggregating Algorithm (AA) achieves fast rates in expectation under any $P \in \mathcal{P}$ in sequential on-line prediction, without any further conditions on $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$; in particular there are no boundedness restrictions on the loss. If we take $\mathcal{P}$ to be the set of all distributions on $\mathcal{Z}$, we recover Vovk's original individual-sequence η-mixability. Note that, based on data $Z_1, \ldots, Z_n$, the AA outputs f that are not necessarily in $\mathcal{F}$ but can be in some different set $\mathcal{F}_{\mathrm{d}}$ (in all applications we are aware of, $\mathcal{F}_{\mathrm{d}} = \mathrm{co}(\mathcal{F})$, the convex hull of $\mathcal{F}$). Online-to-batch conversion has been used, amongst others, by Juditsky et al. (2008); Dalalyan and Tsybakov (2012) and Audibert (2009) to obtain fast rates in model selection aggregation. In Sections 4.2.3 and 4.2.4 we relate their conditions to stochastic mixability. We show that results by Juditsky et al. (2008) employ a *stochastic exp-concavity* condition, a special case of our stochastic mixability condition, in a manner similar to the way exp-concavity is a special case of classical mixability. Given these applications to statistical learning, it is not surprising that stochastic mixability is closely related to the conditions for statistical learning discussed above. We will show in Proposition 4.12 that under certain assumptions it is equivalent to our central condition (5) and hence also the PPC condition (7). The proposition shows that this holds unconditionally in the proper learning setting: stochastic mixability implies the pseudoprobability convexity condition

which, in turn, implies the central condition under some weak restrictions. The proposition also gives a condition under which these relationships continue to hold in the more challenging case when $\mathcal{F} \neq \mathcal{F}_{\mathrm{d}}$. In general, making predictions in $\mathcal{F}_{\mathrm{d}}$ gives more power, and the central condition can only be used to infer fast rates for proper learning algorithms which always play in $\mathcal{F}$. Thus, if η-stochastic mixability for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ implies η-PPC for $(\ell, \mathcal{P}, \mathcal{F})$ then there is no rate improvement for learning algorithms that are allowed to predict in $\mathcal{F}_{\mathrm{d}}$ instead of $\mathcal{F}$. Proposition 4.12 gives a central insight of this paper by showing that this implication holds under Assumption B: *η-stochastic mixability for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ implies the η-PPC and η-central conditions for $(\ell, \mathcal{P}, \mathcal{F})$ whenever $\mathcal{F}$ is well-specified* relative *to* $\mathcal{F}_{\mathrm{d}}$ — relative well-specification was defined in Example 2.1, where we indicated that this a much weaker condition than mere correctness of $\mathcal{F}$; in all cases we are aware of, a sufficient condition is that $\mathcal{F}$ is convex. In Example 4.13 we explore the implications of Proposition 4.12 for the question whether fast rates can be obtained both in expectation and in probability — as is the case under the central condition — or only in expectation — as is sometimes the case under stochastic mixability.

For the implication from the central condition to stochastic mixability, we first define an intermediate, slightly stronger generalization of classical mixability that we call the *η-predictor condition*, which looks like the central condition, but with its universal quantifiers interchanged:

$$\forall \Pi \in \Delta(\mathcal{F})\ \exists f^* \in \mathcal{F}_{\mathrm{d}}\ \forall P \in \mathcal{P}:\ \mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{f\sim\Pi}\left[e^{\eta\left(\ell_{f^*}(Z)-\ell_f(Z)\right)}\right] \leq 1. \tag{9}$$

In our second main result, Theorem 4.17, we show that the central condition implies the predictor condition whenever the decision problem satisfies a certain minimax identity, which holds under Assumption C or its weakening Assumption D. And since (by a trivial application of Jensen's inequality) the predictor condition in turn implies stochastic mixability, we come full circle and see that, under some restrictions, all four of our conditions in the 'central quadrangle' of Figure 1 (page 1798) are really equivalent.

Section 5 — Intermediate Rates: Weakening to v-central condition, connection to Bernstein and Tsybakov Conditions — can be read independently from Section 4.

In Section 5, we weaken the η-central condition to a condition which we call the v-central condition: rather than requiring that a fixed η exists such that (4) holds, we only require that it holds (for all $P \in \mathcal{P}$) up to some 'slack' ε, where we require that the slack must go to 0 as $\eta \downarrow 0$. Specifically, we require that there is some increasing nonnegative function v such that

$$\mathop{\mathbf{E}}_{Z\sim P}\left[e^{\eta\left(\ell_{f^*}(Z)-\ell_f(Z)\right)}\right] \leq e^{\eta\varepsilon} \qquad \text{for all } f \in \mathcal{F}, \text{ all } \varepsilon > 0, \text{ with } \eta := v(\varepsilon). \tag{10}$$

As shown in this section (Example 5.5), the v-central condition is associated with rates of order $w(C/n)$ where $C > 0$ is some constant, and w is the inverse of $x \mapsto xv(x)$ — taking constant $v(x) = \eta$ we see that this generalizes the situation for the η-central condition which for fixed η allows rates of order $O(1/n)$. In our third main result, Theorem 5.4, we then show that, for bounded loss functions, this condition is equivalent to a *generalized Bernstein condition* (see Definition 5.2), which itself is a generalization of the Tsybakov

margin condition (Tsybakov, 2004) to classification settings in which $\mathcal{F}$ may be misspecified, and to loss functions different from 0/1-loss (Bartlett and Mendelson, 2006). Specifically, for given function v, a decision problem satisfies the v-central condition if and only if it satisfies the u-generalized Bernstein condition for a function

$$u(x) \asymp \frac{x}{v(x)}, \tag{11}$$

where for functions a, b from $[0, \infty)$ to $[0, \infty)$, $a(x) \asymp b(x)$ denotes that there exist constants $c, C > 0$ such that, for all $x \geq 0$, $ca(x) \leq b(x) \leq Ca(x)$.

Example 2.3 (Classification) Let $(\ell, \mathcal{P}, \mathcal{F})$ represent a classification problem with ℓ the 0/1-loss that satisfies the v-central condition for $v(x) \asymp x^{1-\beta}$, $0 \leq \beta \leq 1$. Then (11) holds with u of form $u(x) = Bx^{\beta}$. This is equivalent to the standard (β, B)-Bernstein condition (which, if $\mathcal{F}$ is well-specified, corresponds to the Tsybakov margin condition with exponent $\beta/(1-\beta)$), which is known to guarantee rates of $O\left(n^{-1/(2-\beta)}\right)$. This is consistent with the rate $w(C/n)$ above, since if $v(x) \asymp x^{1-\beta}$, then its inverse w satisfies $w(x) \asymp x^{1/(2-\beta)}$. ■

For the case of unbounded losses, the generalized Bernstein and central conditions are not equivalent. Example 5.7 gives a simple case in which the Bernstein condition does not hold whereas, due to its one-sidedness, the central condition does hold and fast rates for ERM are easy to verify; Example 5.8 shows that the opposite can happen as well.

In this section we also extend η-stochastic mixability to v-stochastic-mixability and show that another fast-rate condition identified by Juditsky et al. (2008) is a special case. For unbounded losses, the v-stochastic mixability and the v-central condition become quite different, and it may be that the u-Bernstein condition does imply v-mixability; whether this is so is an open problem. Finally, using Theorem 5.4, we characterize the relationship between the η-central condition and the existence of unique risk minimizers for bounded losses.

Section 6 —From Actions to Predictors.

The classical mixability literature usually considers the *unconditional* setting where observations and actions are points from $\mathcal{Z}$ and $\mathcal{A}$, respectively. For example, one may consider the squared loss with $\ell_a(y) = (y-a)^2$ for $y, a \in [0, 1]$. It is often easy to establish stochastic mixability for a decision problem in this unconditional setting. An interesting question is whether this automatically implies that stochastic mixability (and hence, under further conditions, also the central condition) holds in the corresponding *conditional* setting where $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and the decision set contains predictors $f : \mathcal{X} \to \mathcal{A}$ that map features $x \in \mathcal{X}$ to actions. Here, an example loss function might be $\ell_f^{\text{reg}}((x, y)) = \frac{1}{2}(y - f(x))^2$ as considered in Example 2.1. In this section, we show that the answer is a qualified 'yes' — in general, the set $\mathcal{F}_{\text{d}}$ may need to be a large set such as $\mathcal{A}^{\mathcal{X}}$, but with some additional assumptions it remains manageable.

Section 7 — Fast Rate Theorem.

In Section 7, we show how for bounded losses the central condition enables a direct proof of fast rates in statistical learning over finite classes. The path to our fast rates result,

Theorem 7.6, involves showing that, for each function $f \in \mathcal{F}$, the central condition implies that the empirical excess loss of f exhibits one-sided concentration at a scale related to the excess loss of f. This one-sided concentration result is achieved by way of the Cramér-Chernoff method (Boucheron et al., 2013) combined with an upper bound on the *cumulant generating function* (CGF) of the negative excess loss of f evaluated at a specific point. The upper bound on the CGF is given in Theorem 7.3 which shows that if the absolute value of the excess loss random variable is bounded by 1, its CGF evaluated at some $-\eta < 0$ takes the value 0, and its mean μ is positive, then the central condition implies that the CGF evaluated at $-\eta/2$ is upper bounded by a universal constant times $-\eta\mu$. By way of a careful localization argument, the fast rates result for finite classes also extends to VC-type classes, as presented in Theorem 7.7.

Final Section — Discussion.

The paper ends with a discussion of what has been achieved and a list of open problems.

3. The Central Condition in General and a Bayesian Interpretation via the PPC Condition

In this section we first generalize the definitions of the central and pseudoprobability convexity (PPC) conditions beyond the case of the simplifying Assumption A. We give a few examples and list some of their basic properties. We then show that the central condition trivially implies the PPC condition, under no conditions on the decision problem at all. Additionally, in our first main theorem, we show that if Assumption A holds or the loss is bounded, then the converse result is also true. Importantly, this equivalence between the central condition and the PPC condition allows us to interpret the PPC condition as the requirement that a particular set of *pseudoprobabilities* is convex on the side that 'faces' the data-generating distribution P (Figure 2). This leads to a (pseudo)-Bayesian interpretation, which says that the (pseudo)-Bayesian predictive distribution is not allowed to be better than the best element of the model.

3.1 The Central and Pseudoprobability Convexity Conditions in General

We now extend the definition (4) of the central condition to the case that our simplifying Assumption A may not hold. In such cases, it may be that there is no fixed comparator that satisfies (4), but there does exist a sequence of comparators $f_1^*, f_2^*, \ldots$ that satisfies (5) in the limit. By introducing a function ϕ that maps P to f^* this leads to the following definition of the general η-central condition:

Definition 3.1 (Central Condition) *Let $\eta > 0$ and $\varepsilon \geq 0$. We say that $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the* η-central condition up to ε *if there exists a* comparator selection *function $\phi\colon \mathcal{P} \to \mathcal{F}$ such that*

$$\mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\Pi} \left[e^{\eta\left(\ell_{\phi(P)}(Z) - \ell_f(Z)\right)} \right] \leq e^{\eta\varepsilon} \qquad \text{for all } P \in \mathcal{P} \text{ and distributions } \Pi \in \Delta(\mathcal{F}). \tag{12}$$

If it satisfies the η-central condition up to 0, we say that the strong η-central condition *or simply the* η-central condition *holds. If it satisfies the η-central condition up to ε for all*

$\varepsilon > 0$, *we say that the* weak η-central condition *holds; this is equivalent to*

$$\sup_{P \in \mathcal{P}} \inf_{f^* \in \mathcal{F}} \sup_{\Pi \in \Delta(\mathcal{F})} \mathop{\mathbf{E}}_{Z \sim P} \mathop{\mathbf{E}}_{f \sim \Pi} \left[e^{\eta \left(\ell_{f^*}(Z) - \ell_f(Z) \right)} \right] \leq 1. \tag{13}$$

Note that we explicitly identify the situation in which the condition does not actually hold in the strong sense but will if some slack $\varepsilon > 0$ is introduced. We will do the same for the other fast rate conditions identified in this paper, and we will also establish relations between the 'up to $\varepsilon > 0$' versions. This will become useful throughout Section 5 and, in particular, Section 5.3.

The PPC condition generalizes analogously to the central condition and features

$$m_\Pi^\eta(z) = -\frac{1}{\eta} \log \mathop{\mathbf{E}}_{f \sim \Pi} \left[e^{-\eta \ell_f(z)} \right], \tag{14}$$

a quantity that plays a crucial role in the analysis of online learning algorithms (Vovk, 1998, 2001), (Cesa-Bianchi and Lugosi, 2006, Theorem 2.2) and has been called the *mix loss* in that context by De Rooij et al. (2014).

Definition 3.2 (Pseudoprobability convexity condition) *Let $\eta > 0$ and $\varepsilon \geq 0$. We say that $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the* η-pseudoprobability convexity condition up to ε *if there exists a function $\phi \colon \mathcal{P} \to \mathcal{F}$ such that*

$$\mathop{\mathbf{E}}_{Z \sim P} \left[\ell_{\phi(P)}(Z) \right] \leq \mathop{\mathbf{E}}_{Z \sim P} \left[m_\Pi^\eta(Z) \right] + \varepsilon \qquad \text{for all } P \in \mathcal{P} \text{ and } \Pi \in \Delta(\mathcal{F}). \tag{15}$$

If it satisfies the η-pseudoprobability convexity condition up to 0, we say that the strong η-pseudoprobability convexity condition *or simply the* η-pseudoprobability convexity condition *holds. If it satisfies the η-pseudoprobability convexity condition up to ε for all $\varepsilon > 0$, we say that the* weak η-pseudoprobability convexity condition *holds; this is equivalent to*

$$\sup_{\Pi \in \Delta(\mathcal{F})} \sup_{P \in \mathcal{P}} \inf_{f \in \mathcal{F}} \mathop{\mathbf{E}}_{Z \sim P} \left[\ell_f(Z) - m_\Pi^\eta(Z) \right] \leq 0. \tag{16}$$

Under Assumption A this condition simplifies and implies the essential uniqueness of optimal predictors (cf. Section 3.3).

Proposition 3.3 (PPC condition implies uniqueness of risk minimizers) *Suppose that Assumption A holds, and that $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the weak η-pseudoprobability convexity condition. Then it also satisfies the strong η-pseudoprobability convexity condition, and for all $P \in \mathcal{P}$, the $\mathcal{F}$-optimal f^* satisfying (3) is essentially unique, in the sense that, for any $g^* \in \mathcal{F}$ with $R(P, g^*) = R(P, f^*)$, we have that $\ell_{g^*}(Z) = \ell_{f^*}(Z)$ holds P-almost surely.*

Proof Assumption A implies that if (15) holds at all, then it also holds with $\phi(P)$ equal to any $\mathcal{F}$-risk minimizer f^* as in (3). Thus, if it holds for all $\varepsilon > 0$, it holds for all $\varepsilon > 0$ with the fixed choice f^*, and hence it must also hold for $\varepsilon = 0$ with the same f^*.

As to the second part, consider a distribution Π that puts mass $1/2$ on f^* and $1/2$ on g^*. Then the strong η-pseudoprobability condition implies that

$$\begin{aligned} \min_{f \in \mathcal{F}} \mathop{\mathbf{E}}_{Z \sim P} [\ell_f(Z)] &\leq \mathop{\mathbf{E}}_{Z \sim P} \left[-\frac{1}{\eta} \log \left(\frac{1}{2} e^{-\eta \ell_{f^*}(Z)} + \frac{1}{2} e^{-\eta \ell_{g^*}(Z)} \right) \right] \\ &\leq \mathop{\mathbf{E}}_{Z \sim P} \left[\frac{1}{2} \ell_{f^*}(Z) + \frac{1}{2} \ell_{g^*}(Z) \right] = \min_{f \in \mathcal{F}} \mathop{\mathbf{E}}_{Z \sim P} [\ell_f(Z)], \end{aligned}$$

where we used convexity of $-\log$ and Jensen's inequality. Hence both inequalities must hold with equality. By strict convexity of $-\log$, we know that for the second inequality this can only be the case if $\ell_{f^*} = \ell_{g^*}$ almost surely, which was to be shown. ■

Finally, we will often make use of the following trivial but important fact.

Fact 3.4 *Fix $\eta > 0, \varepsilon \geq 0$ and let $(\ell, \mathcal{P}, \mathcal{F})$ be an arbitrary decision problem that satisfies the η-central condition up to ε. Then for any $0 < \eta' \leq \eta$ and any $\varepsilon' \geq \varepsilon$ and for any $\mathcal{P}' \subseteq \mathcal{P}$, $(\ell, \mathcal{P}', \mathcal{F})$ satisfies the η'-central condition up to ε'. The same holds with 'central' replaced by 'PPC'.*

We proceed to give some examples.

Example 3.5 (Squared Loss, Unrestricted Domain) Consider squared loss $\ell_f^{\mathrm{sq}}(z) = \frac{1}{2}(z-f)^2$ with $\mathcal{Z} = \mathcal{F} = \mathbb{R}$, and let $\mathcal{P} = \{\mathcal{N}(\mu, 1) : \mu \in \mathbb{R}\}$ be the set of normal distributions with unit variance and arbitrary means μ. Estimating the mean of a normal model is a standard inference problem for which a squared error risk of order $O(1/n)$ is obtained by the sample mean. We would therefore expect the central condition to be satisfied and, indeed, this is the case for $\eta \leq 1$ via a reduction to Example 2.2. To see this, consider the well-specified setting for the log loss $\ell_{f'}^{\log}$ with densities $f' \in \mathcal{F}' = \mathcal{P}$, and note that the squared loss for f equals the log loss for f' up to a constant when f is the mean of f':

$$\ell_f^{\mathrm{sq}}(z) = -\log e^{-(z-f)^2/2} = \ell_{f'}^{\log}(z) - \log\sqrt{2\pi}.$$

Since the log loss satisfies the 1-central condition in the well-specified case (see Example 2.2), the squared loss must also satisfy the 1-central condition. ■

Not surprisingly, the central condition still holds if we replace the Gaussian assumption by a subgaussian assumption.

Example 3.6 For $\sigma^2 > 0$ let $\mathcal{P}_{\sigma^2}$ be an arbitrary subgaussian collection of distributions over $\mathbb{R}$. That is, for all $t \in \mathbb{R}$ and $P \in \mathcal{P}_{\sigma^2}$

$$\mathop{\mathbf{E}}_{Z\sim P}\left[e^{t(Z-\mu_P)}\right] \leq e^{\sigma^2 t^2/2}, \tag{17}$$

where $\mu_P = \mathbf{E}_{Z\sim P}[Z]$ is the mean of Z. Now consider the squared loss $\ell_f^{\mathrm{sq}}(z) = \frac{1}{2}(z-f)^2$ again, with $\mathcal{F} = \mathcal{Z} = \mathbb{R}$. Then

$$\ell_f^{\mathrm{sq}}(z) - \ell_{f'}^{\mathrm{sq}}(z) = \frac{1}{2}\delta(2(z-f)-\delta), \qquad \text{where } \delta = f' - f. \tag{18}$$

Taking $f = \mu_P$ gives

$$\mathop{\mathbf{E}}_{Z\sim P}\left[e^{\eta\left(\ell_f^{\mathrm{sq}}(Z)-\ell_{f'}^{\mathrm{sq}}(Z)\right)}\right] = e^{-\eta\delta^2/2}\mathop{\mathbf{E}}_{Z\sim P}\left[e^{\eta\delta(Z-\mu_P)}\right] \leq e^{-\eta\delta^2/2}e^{\sigma^2\eta^2\delta^2/2}. \tag{19}$$

The right-hand side is at most 1 if $\eta \leq 1/\sigma^2$, and hence to satisfy the strong η-central condition with substitution function $\phi(P) = \mu_P$, it suffices to take $\eta \leq 1/\sigma^2$. Note that

ϕ maps P to the $\mathcal{F}$-optimal predictor for $\mathcal{P}$ — a fact which holds generally, as shown in Proposition 3.3 above. Note also that, just like Example 3.5, the example can be reduced to the log-loss setting in which the densities are all normal densities with means in $\mathbb{R}$ and variance equal to 1. In Example 5.8 we shall see that if $\mathcal{P}$ contains P with polynomially large tails, then the η-central condition may fail. ■

Example 3.7 (Subgaussian Regression) Examples 2.2, 3.5 and 3.6 all deal with the unconditional setting (cf. page 1805) of estimating a mean without covariate information. The corresponding conditional setting is regression, in which $\mathcal{F}$ is a set of functions $f : \mathcal{X} \to \mathcal{Y}$, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $\mathcal{Y} = \mathbb{R}$ and $\ell_f^{\mathrm{reg}}((x,y)) := \ell_{f(x)}^{\mathrm{sq}}(y)$. Analogously to Example 3.6, fix $\sigma^2 > 0$ and let $\mathcal{P}$ be a set of distributions on $\mathcal{X} \times \mathcal{Y}$ such that for each $P \in \mathcal{P}$ and $x \in \mathcal{X}$, $P(Y \mid X = x)$ is subgaussian in the sense of (17). Now consider a decision problem $(\ell^{\mathrm{reg}}, \mathcal{P}, \mathcal{F})$. Example 3.6 applies to this regression setting, provided that, for each $P \in \mathcal{P}$, the model $\mathcal{F}$ contains the true regression function $f_P^*(x) := \mathbf{E}_{(X,Y)\sim P}[Y \mid X = x]$. To see this, note that then for all $P \in \mathcal{P}$, all $f' \in \mathcal{F}$,

$$\begin{aligned}\mathop{\mathbf{E}}_{(X,Y)\sim P}\left[e^{\eta\left(\ell^{\mathrm{reg}}_{f_P^*}(X,Y)-\ell^{\mathrm{reg}}_{f'}(X,Y)\right)}\right] &= \mathop{\mathbf{E}}_{P(X)}\mathop{\mathbf{E}}_{P(Y|X)}\left[e^{\eta\left(\ell^{\mathrm{sq}}_{f_P^*(X)}(Y)-\ell^{\mathrm{sq}}_{f'(X)}(Y)\right)}\right] \\ &\le \mathop{\mathbf{E}}_{P(X)}\left[e^{-\eta\delta^2/2}e^{\sigma^2\eta^2\delta^2/2}\right] \le 1,\end{aligned}$$

where the final inequality holds as long as $\eta \le 1/\sigma^2$. Thus the $1/\sigma^2$-central condition holds. Although it is often made, the assumption that $\mathcal{F}$ contains the Bayes decision rule (*i.e.*, the true regression function) is quite strong. In Section 6 we will encounter Example 6.2 where, under a compactness restriction on $\mathcal{P}$, the central condition still holds even though $\mathcal{F}$ may be misspecified. ■

Example 3.8 (Bernoulli, $0/1$-loss and the margin condition) Let $\mathcal{Z} = \mathcal{F} = \{0,1\}$, for any $0 \le \delta \le 1/2$ let $\mathcal{P}_\delta$ be the set of distributions P on $\mathcal{Z}$ with $|P(Z=1) - 1/2| \ge \delta$, and let ℓ^{01} be the $0/1$-loss with $\ell^{01}(y,f) = |y-f|$. For every $\delta > 0$, there is an $\eta > 0$ such that the η-central condition holds for $(\ell^{01}, \mathcal{P}_\delta, \mathcal{F})$. To see this, let f^* be the Bayes act for P, *i.e.*, $f^* = 1$ if and only if $P(Z=1) > 1/2$, and, for $f \neq f^*$, define $A(\eta) = \mathbf{E}_{Z\sim P}\left[e^{\eta(\ell^{01}_{f^*}(Z) - \ell^{01}_f(Z))}\right]$. Then $A(0) = 1$ and the derivative $A'(0)$ is easily seen to be negative, which implies the result. However, as $\delta \downarrow 0$, so does the largest η for which the central condition holds. For $\delta = 0$, the central condition does not hold any more. Since the central condition and the PPC condition are equivalent, this also follows from Proposition 3.3: if $\delta = 0$, then there exist $P \in \mathcal{P}$ with $P(Z=1) = 1/2$, and for this P both $f \in \mathcal{F} = \{0,1\}$ have equal risk so there is no unique minimum. For each $\delta > 0$, the restriction to $\mathcal{P}_\delta$ may also be understood as saying that a *Tsybakov margin condition* (Tsybakov, 2004) holds with noise exponent ∞, the most stringent case of this condition that has long been known to ensure fast rates. As will be seen in Example 5.5 the Tsybakov margin condition can also be thought of as a Bernstein condition with $\beta = 0$ and $B \uparrow \infty$ as $\delta \downarrow 0$ (in practice, however, this condition is

usually applied in the conditional setting with covariates X). Finally, just like the squared loss examples, this example can be recast in terms of log-loss as well. Fix $\beta > 0$ and let $\mathcal{F}_\beta$ be the subset of the Bernoulli model containing two symmetric probability mass functions, p_1 and p_0, where $p_1(1) = p_0(0) = e^\beta/(1+e^\beta) > 1/2$. Then the log loss Bayes act for P is p_1 if and only if $P(Z=1) > 1/2$. For $P \in \mathcal{P}_\delta$ and $f' \neq f^*$, $\mathbf{E}_{Z\sim P}\left[e^{\eta(\ell^{\log}_{f^*}(Z) - \ell^{\log}_{f}(Z))}\right] = A(\beta\eta)$, which by the same argument as above can be made < 1 if $\eta > 0$ is chosen small enough (provided $\delta > 0$). ■

3.2 Equivalence of Central and Pseudoprobability Convexity Conditions

The following result shows that no additional assumptions are required for the central condition to imply the pseudoprobability convexity condition.

Proposition 3.9 *Fix an arbitrary decision problem $(\ell, \mathcal{P}, \mathcal{F})$ and $\varepsilon \geq 0$. If the η-central condition holds up to ε then the η-pseudoprobability convexity condition holds up to ε. In particular the (strong) η-central condition implies the (strong) η-pseudoprobability convexity condition.*

Proof Let $P \in \mathcal{P}$ and $\Pi \in \Delta(\mathcal{F})$ be arbitrary. Assume the η-central condition holds up to ε. Then

$$\begin{aligned}\mathop{\mathbf{E}}_{Z\sim P}\left[\ell_{\phi(P)}(Z) - m^\eta_\Pi(Z)\right] &= \frac{1}{\eta}\mathop{\mathbf{E}}_{Z\sim P}\log\mathop{\mathbf{E}}_{f\sim\Pi}\left[e^{\eta\left(\ell_{\phi(P)}(Z) - \ell_f(Z)\right)}\right] \\ &\leq \frac{1}{\eta}\log\mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{f\sim\Pi}\left[e^{\eta\left(\ell_{\phi(P)}(Z) - \ell_f(Z)\right)}\right] \leq \varepsilon.\end{aligned}$$

where the first inequality is Jensen's and the second inequality follows from the central condition (12). ■

To obtain the reverse implication we require either Assumption A (*i.e.*, that minimum risk within $\mathcal{F}$ is achieved) or, if Assumption A does not hold, the boundedness of the loss[3]. Below we use the term 'essentially unique' in the sense of Proposition 3.3 and call any g^* such that $\ell_{g^*}(Z) = \ell_{f^*}(Z)$ occurs P-almost-surely a *version* of f^*.

Theorem 3.10 *Let $(\ell, \mathcal{P}, \mathcal{F})$ be a decision problem. Then the following statements both hold:*

1. *If ℓ is bounded, then the weak η-pseudoprobability convexity condition implies the weak η-central condition.*

2. *Moreover, if Assumption A holds, then (irrespective of whether the loss is bounded) the weak η-pseudoprobability convexity condition implies the strong η-central condition with comparator function $\phi(P) := f^*$ for $\mathcal{F}$-optimal f^*. That is, f^* can be any version of the essentially unique element of $\mathcal{F}$ that satisfies (3).*

3. We suspect this latter requirement can be weakened, at the cost of considerably complicating the proof.

The proof of Theorem 3.10 is deferred to Appendix A.1. It generalizes a result for log loss from the PhD thesis of Li (1999, Theorem 4.3) and Barron (2001).[4] Theorem 3.10 leads to the following useful consequence.

Corollary 3.11 *Consider a decision problem $(\ell, \mathcal{P}, \mathcal{F})$ and suppose that Assumption A holds. Then the following are equivalent:*

1. *The weak η-central condition is satisfied.*
2. *The strong η-central condition is satisfied with comparator function ϕ as given by Theorem 3.10.*
3. *The weak η-pseudoprobability convexity condition is satisfied.*
4. *The strong η-pseudoprobability convexity condition is satisfied.*

If any of these statements hold, then for all $P \in \mathcal{P}$, the corresponding optimal f^* is essentially unique in the sense of Proposition 3.3.

Proof Suppose that the η-(weak) pseudoprobability convexity condition holds and that Assumption A holds. This implies that the infimum in (16) is always achieved, from which it follows that the strong η-pseudoprobability convexity condition holds. The assumption also lets us apply Theorem 3.10 which implies that the strong η-central condition holds with ϕ as described. This immediately implies the weak η-central condition which, via Proposition 3.9, implies the weak η-pseudoprobability convexity condition. ■

The corollary establishes the equivalence of the weak and strong central and pseudoprobability convexity conditions which we assumed in Section 2.2. The result prompts the question whether *non*-uniqueness of the optimal f^* might imply that the four conditions do *not* hold. While this is not true in general, at least for bounded losses it is 'almost' true if we replace the η-fast rate conditions by the weaker notion of v–fast rate conditions of Section 5 (see Proposition 5.11).

3.3 Interpretation as Convexity of the Set of Pseudoprobabilities and a Bayesian Interpretation

As we will now explain both the pseudoprobability convexity condition and, by the equivalence from the previous section, the central condition may be interpreted as a partial convexity requirement. For simplicity, we restrict ourselves to the setting of Assumption A from Section 2.2. We first present this interpretation for the logarithmic loss from Example 2.2 on page 1801, for which it is most natural and can also be given a Bayesian interpretation.

Example 3.12 (Example 2.2 continued: convexity interpretation for log loss) Let $P \in \mathcal{P}$ be arbitrary. Under Assumption A the strong 1-pseudoprobability convexity

4. Under Assumption A, the proof of Theorem 3.10 shows that it is actually sufficient if the weak pseudoprobability convexity condition only holds for distributions Π on f^* and single $f \in \mathcal{F}$. Via Proposition 3.9 we then see that this actually implies weak pseudoprobability convexity for all distributions Π.

condition for log loss says that

$$\mathop{\mathbf{E}}_{Z\sim P}[-\log f^*(Z)] \leq \min_{\Pi\in\Delta(\mathcal{F})} \mathop{\mathbf{E}}_{Z\sim P}\left[-\log \mathop{\mathbf{E}}_{f\sim\Pi}[f(Z)]\right], \quad \textit{i.e.},$$
$$\min_{f\in\mathcal{F}} \mathop{\mathbf{E}}_{Z\sim P}[-\log f(Z)] = \min_{f\in\mathrm{co}(\mathcal{F})} \mathop{\mathbf{E}}_{Z\sim P}[-\log f(Z)], \tag{20}$$

where $f^* = \phi(P)$ and $\mathrm{co}(\mathcal{F})$ denotes the convex hull of $\mathcal{F}$ (*i.e.*, the set of all mixtures of densities in $\mathcal{F}$). This may be interpreted as the requirement that a convex combination of elements of the model $\mathcal{F}$ is never better than the best element in the model. This means that the model is essentially convex with respect to P (*i.e.*, 'in the direction facing' P — see Figure 2).

In particular, in the context of Bayesian inference, the *Bayesian predictive distribution* after observing data $Z_1, \ldots, Z_n$ is a mixture of elements of the model according to the posterior distribution, and therefore must be an element of $\mathrm{co}(\mathcal{F})$. The pseudoprobability convexity condition thus rules out the possibility that the predictive distribution is strictly better (in terms of expected log loss or, equivalently, KL-divergence) than the best single element in the model. This might otherwise be possible if the posterior was spread out over different parts of the model. This interpretation is explained at length by Grünwald and van Ommen (2014) who provide a simple regression example in which (20) does not hold and the Bayes predictive distribution is, with substantial probability, better than the best single element f^* in the model, and the Bayesian posterior does not concentrate around this optimal f^* at all. ■

For log loss, the convexity requirement (20) is, by Corollary 3.11, equivalent to the strong 1-central condition and can thus be written as

$$\mathop{\mathbf{E}}_{Z\sim P}\left[\frac{f(Z)}{f^*(Z)}\right] \leq 1 \tag{21}$$

for all $f \in \mathcal{F}$. Recognizing (6) we therefore also recover the result by Li (1999) mentioned in Example 2.2.

Example 3.13 (Bayes-MDL Condition) The 1-central condition (21) for log loss plays a fundamental role in establishing consistency and fast rates for Bayesian and related methods. Due to its use in a large number of papers on convergence of MDL-based methods (Grünwald, 2007) and Bayesian methods and lack of a standard name, we will henceforth call it the *Bayes-MDL condition*. Most of the papers using this condition make the traditional assumption that the model is well-specified, *i.e.*, for every $P \in \mathcal{P}$, $\mathcal{F}$ contains the density of P. As already mentioned in Example 2.2, the condition then holds automatically, so one does not see (21) stated in those papers as an explicit condition. Yet, if one tries to generalize the results of such papers to the misspecified case, one invariably sees that the only step in the proofs needing adjustment is the step where (21) is implicitly employed. If the model is incorrect yet (21) holds, then the proofs invariably still go through, establishing convergence towards the f^* that minimizes KL divergence to the true P. This happens, for example, in the MDL convergence proofs of Barron and Cover (1991); Zhang (2006a);

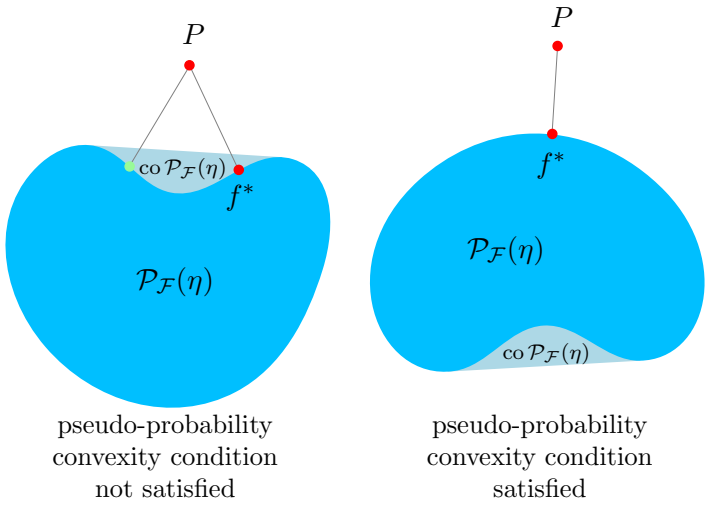


Figure 2: The pseudoprobability convexity condition interpreted as convexity of the set of pseudoprobabilities with respect to P.

Grünwald (2007) as well as in the pioneering paper by Doob (1949) on Bayesian consistency. The dependence on (21) becomes more explicit in works explicitly dealing with misspecification such as those by Li (1999); Kleijn and van der Vaart (2006); Grünwald (2011). For example, in order to guarantee convergence of the posterior around the best element f^* of misspecified models, Kleijn and van der Vaart (2006) impose a highly technical condition on $(\ell, \mathcal{P}, \mathcal{F})$. If, however, (21) holds then this complicated condition simplifies to the standard, much simpler condition from (Ghosal et al., 2000) which is sufficient for convergence in the well-specified case. The same phenomenon is seen in results by Ramamoorthi et al. (2013); De Blasi and Walker (2013). Grünwald and Langford (2004) and Grünwald and van Ommen (2014) give examples in which the condition does not hold, and Bayes and MDL estimators fail to converge. ■

The convexity interpretation for log loss may be generalized to other loss functions via loss dependent 'pseudoprobabilities'. These play a crucial role both in online learning (Vovk, 2001) and the PAC-Bayesian analysis of the Bayes posterior and the MDL estimator by Zhang (2006a). For log loss, we may express the ordinary densities in terms of the loss as $f(z) = e^{-\ell_f(z)}$. This generalizes to other loss functions by letting $\eta\ell_f(z)$ play the role of the log loss, where $\eta > 0$ is the scale factor that appears in all our definitions. We thus obtain the set of *pseudoprobabilities*

$$\mathcal{P}_{\mathcal{F}}(\eta) = \left\{ z \mapsto e^{-\eta\ell_f(z)} : f \in \mathcal{F} \right\},$$

which are non-negative, but do not necessarily integrate to 1. The only feature we need of these pseudoprobabilities is that their log loss is equal to η times the original loss, because, analogously to (20), this allows us to write the strong η-pseudoprobability convexity

condition as

$$\min_{f \in \mathcal{P}_{\mathcal{F}}(\eta)} \mathop{\mathbf{E}}_{Z \sim P} \left[-\log f(Z) \right] \leq \min_{f \in \mathrm{co}(\mathcal{P}_{\mathcal{F}}(\eta))} \mathop{\mathbf{E}}_{Z \sim P} \left[-\log f(Z) \right].$$

Figure 2 provides a graphical illustration of this condition. Thus, for any loss function we can interpret the pseudoprobability convexity condition as the requirement that the set of pseudoprobabilities is essentially convex with respect to P. As suggested by Vovk (2001); Zhang (2006a), one can also run Bayes on such pseudoprobabilities, and then the pseudoprobability convexity condition again implies that the resulting pseudo-Bayesian predictive distribution cannot be strictly better than the single best element of the model. The log loss achieved with such pseudoprobabilities, and hence η times the original loss, can be given a code length interpretation, essentially allowing arbitrary loss functions to be recast as versions of logarithmic loss (Grünwald, 2008).

4. Online Learning

In this section, we discuss conditions for fast rates that are related to online learning. Our key concept is introduced in Section 4.1, where we define *stochastic mixability*, the natural stochastic generalization of Vovk's notion of mixability, and show (in Section 4.2) how it unifies existing conditions in the literature. Section 4.3 contains the main results for this section, which connect stochastic mixability to the central condition and to pseudoprobability convexity. As an intermediate step, these results use a fourth condition called the *predictor condition*, which is related to the central condition via a minimax identity. We show that, under appropriate assumptions, all four conditions are equivalent. This equivalence is important because it relates the generic condition for fast rates in online learning (stochastic mixability) to the generic condition that enables fast rates for proper in-model estimators in statistical learning (the central condition).

4.1 Stochastic Mixability in General

Stochastic mixability generalizes from (8) similarly to the way we have generalized the central condition and pseudoprobability convexity. Let $m^{\eta}_{\Pi}(z)$ be the mix loss, as defined in (14).

Definition 4.1 (The Stochastic Mixability Condition) *Let $\eta > 0$ and $\varepsilon \geq 0$. We say that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is* η-stochastically mixable *up to ε if there exists a* substitution function *$\psi \colon \Delta(\mathcal{F}) \to \mathcal{F}_{\mathrm{d}}$ such that*

$$\mathop{\mathbf{E}}_{Z \sim P} \left[\ell_{\psi(\Pi)}(Z) \right] \leq \mathop{\mathbf{E}}_{Z \sim P} \left[m^{\eta}_{\Pi}(Z) \right] + \varepsilon \qquad \text{for all } P \in \mathcal{P} \text{ and } \Pi \in \Delta(\mathcal{F}). \tag{22}$$

If it is η-stochastically mixable up to 0, we say that it is strongly η-stochastically mixable *or simply* η-stochastically mixable*. If it is η-stochastically mixable up to ε for all $\varepsilon > 0$, we say that it is* weakly η-stochastically mixable*; this is equivalent to*

$$\sup_{\Pi \in \Delta(\mathcal{F})} \inf_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{Z \sim P} \left[\ell_f(Z) - m^{\eta}_{\Pi}(Z) \right] \leq 0. \tag{23}$$

Unlike for the central and pseudoprobability convexity conditions (see Corollary 3.11), for stochastic mixability it is not clear whether the weak and strong versions become equivalent under the simplifying Assumption A. We do have a trivial yet important extension of Fact 3.4:

Fact 4.2 *Fix $\eta > 0, \varepsilon \geq 0$ and let $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ be an arbitrary decision problem that is η-stochastically mixable up to ε. Then for any $0 < \eta' \leq \eta$, any $\varepsilon' \geq \varepsilon$ and for any $\mathcal{P}' \subseteq \mathcal{P}$, $\mathcal{F}' \subseteq \mathcal{F}$ and $\mathcal{F}'_{\mathrm{d}} \supseteq \mathcal{F}_{\mathrm{d}}$, $(\ell, \mathcal{P}', \mathcal{F}', \mathcal{F}'_{\mathrm{d}})$ is η'-stochastically mixable up to ε'.*

4.2 Relations to Conditions in the Literature

As explained next, stochastic mixability generalizes Vovk's notion of (non-stochastic) mixability, and correspondingly implies fast rates. Its most important special case is stochastic exp-concavity, for which Juditsky et al. (2008) give sufficient conditions, and which is used by, e.g., Dalalyan and Tsybakov (2012). Stochastic mixability is also equivalent to a special case of a condition introduced by Audibert (2009).

4.2.1 Generalization of Vovk's Mixability and Fast Rates for Stochastic Prediction with Expert Advice

If we take $\varepsilon = 0$ and let $\mathcal{P}$ be the set of all possible distributions, then (22) reduces to

$$\ell_{\psi(\Pi)}(z) \leq m^{\eta}_{\Pi}(z) \qquad \text{for all } z \in \mathcal{Z} \text{ and } \Pi \in \Delta(\mathcal{F}), \tag{24}$$

which is Vovk's original definition of (non-stochastic) mixability (Vovk, 2001). It follows that Vovk's mixability implies strong stochastic mixability for all sets $\mathcal{P}$.

Example 4.3 (Mixable Losses) Losses that are classically mixable in Vovk's sense, include the squared loss $\ell^{\mathrm{sq}}(f, z) = \frac{1}{2}(z - f)^2$ on a bounded domain $\mathcal{Z} = \mathcal{F}_{\mathrm{d}} \supseteq \mathcal{F} = [-B, B]$, which is $1/B^2$-mixable (Vovk, 2001, Lemma 3)[5], and the logarithmic loss, which is 1-mixable for $\mathcal{F}_{\mathrm{d}} \subseteq \mathrm{co}(\mathcal{F})$ with substitution function equal to the mean $\psi(\Pi) = \mathbf{E}_{f \sim \Pi}[f]$. The *Brier score* is also 1-mixable (Vovk and Zhdanov, 2009; van Erven et al., 2012b); this loss function is defined for all possible probability distributions $\mathcal{F}_{\mathrm{d}} = \mathcal{F}$ on a finite set of outcomes $\mathcal{Z}$ according to $\ell^{\mathrm{Brier}}_f(z) = \sum_{z' \in \mathcal{Z}} (f(z') - \delta_z(z'))^2$, where δ_z denotes a point-mass at z. ■

Example 4.4 (0/1 Loss: Example 3.8, Continued) Fix $0 \leq \delta \leq 1/2$ and consider a decision problem $(\ell^{01}, \mathcal{P}_\delta, \mathcal{F})$ where ℓ^{01} is the 0/1-loss, $\mathcal{Z} = \mathcal{F} = \{0, 1\}$ and $\mathcal{P}_\delta$ is as in Example 3.8. The 0/1-loss is not η-mixable for any $\eta > 0$ (Vovk, 1998), and it is also easily shown that $(\ell^{01}, \mathcal{P}_\delta, \mathcal{F}, \mathcal{F})$ is not η-stochastically mixable for any $\eta > 0$; nevertheless, if $\delta > 0$, then $(\ell^{01}, \mathcal{P}_\delta, \mathcal{F})$ does satisfy the η-central condition for some $\eta > 0$. In Section 4.3 we show that, under some conditions, the η-central condition and η-stochastic mixability coincide, but this example shows that this cannot always be the case. ■

5. Taking into account the factor of $\frac{1}{2}$ difference between his definition of squared loss as $(z - f)^2$ and ours.

Vovk defines the *aggregating algorithm* (AA) and shows that it achieves constant regret in the setting of prediction with expert advice, which is the online learning equivalent of fast rates, provided that (24) is satisfied. In prediction with expert advice, the data $Z_1, \ldots, Z_n$ are chosen by an adversary, but one may define a stochastic analogue by letting the adversary instead choose $P_1, \ldots, P_n \in \mathcal{P}$, where the choice of P_i may depend on the player's predictions on rounds $1, \ldots, i-1$, and letting $Z_i \sim P_i$ for all $i = 1, \ldots, n$. It turns out that under no further conditions, stochastic mixability implies fast rates for the expected regret under $P_1, \ldots, P_n$ in this stochastic version of prediction with expert advice. In particular, there is no requirement that losses are bounded.

Proposition 4.5 *Let $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ be η-stochastically mixable up to ε with substitution function ψ. Assume the data $Z_1, \ldots, Z_n$ are distributed as $Z_j \sim P_j \in \mathcal{P}$ for each $j \in [n]$, where the P_j can be adversarially chosen. Then the AA, playing $f_j \in \mathcal{F}_{\mathrm{d}}$ in round j, achieves, for all $f \in \mathcal{F}$, regret*

$$\sum_{j=1}^{n} \mathop{\mathbf{E}}_{Z_j \sim P_j} \left[\ell_{f_j}(Z_j) - \ell_f(Z_j) \right] \leq \frac{\log |\mathcal{F}|}{\eta} + n\varepsilon.$$

In particular, in the statistical learning (stochastic i.i.d.) setting where $P_1, \ldots, P_n$ all equal the same P, online-to-batch conversion yields the bound $\frac{\log |\mathcal{F}|}{\eta n} + \varepsilon$ on the expected regret and hence on the rate (2) of the AA is $O(\frac{\log |\mathcal{F}|}{\eta n} + \varepsilon)$.

Proof For $\varepsilon = 0$, the first result follows by replacing every occurrence of mixability with stochastic mixability in Vovk's proof (see Section 4 of Vovk (1998) or the proof of Proposition 3.2 of Cesa-Bianchi and Lugosi (2006)). The case of $\varepsilon > 0$ is handled simply by adding a slack of ε to the RHS of the first equation after equation (18) of Vovk (1998). The online-to-batch conversion of the second result is well-known and can be found e.g. in the proof of Lemma 4.3 of Audibert (2009). ■

4.2.2 Special Case: Stochastic Exp-concavity

In online convex optimization, an important sufficient condition for fast rates requires the loss to be *η-exp-concave* in f (Hazan et al., 2007), meaning that $\mathcal{F} = \mathcal{F}_{\mathrm{d}}$ is convex and that

$$e^{-\eta \ell_f(z)} \qquad \text{is concave in } f \text{ for all } z \in \mathcal{Z}. \tag{25}$$

We may equivalently express this requirement as

$$e^{-\eta \ell_{\mathbf{E}_{f \sim \Pi}[f]}(z)} \geq \mathop{\mathbf{E}}_{f \sim \Pi} \left[e^{-\eta \ell_f(z)} \right], \quad \text{or}$$
$$\ell_{\mathbf{E}_{f \sim \Pi}[f]}(z) \leq m_{\Pi}^{\eta}(z),$$

for all distributions $\Pi \in \Delta(\mathcal{F})$ and all $z \in \mathcal{Z}$. This shows that exp-concavity is a special case of mixability, where we require the function ψ to map Π to its mean:

$$\psi(\Pi) = \mathop{\mathbf{E}}_{f \sim \Pi}[f].$$

Because the mean $\mathbf{E}_{f\sim\Pi}[f]$ depends not only on the losses ℓ_f, but also on the choice of parameters f, we therefore see that exp-concavity is *parametrization-dependent*, whereas in general the property of being mixable is unaffected by the choice of parametrization. The parametrization dependent nature of exp-concavity is explored in detail by Vernet et al. (2011); Kamalaruban et al. (2015); see also van Erven et al. (2012b); van Erven (2012).

Example 4.6 (Exp-concavity) Consider again the mixable losses from Example 4.3. Then the log loss is 1-exp concave. The squared loss, in its standard parametrization, is *not* $1/B^2$-exp-concave, but it is $1/(4B^2)$-exp-concave, losing a factor of 4 (Vovk, 2001, Remark 3). By continuously reparametrising the squared loss, however, it can be made $1/B^2$-exp-concave after all (Kamalaruban et al., 2015; van Erven, 2012). It is not known whether there exists a parametrization that makes the Brier score 1-exp-concave. ■

The natural generalization of exp-concavity to stochastic exp-concavity becomes:

Definition 4.7 *Suppose $\mathcal{F}_{\mathrm{d}} \supseteq \mathrm{co}(\mathcal{F})$. Then we say that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is η-stochastically exp-concave up to ε or strongly/weakly η-stochastically exp-concave if it satisfies the corresponding case of stochastic mixability with substitution function $\psi(\Pi) = \mathbf{E}_{f\sim\Pi}[f]$.*

4.2.3 The JRT Conditions Imply Stochastic Exp-concavity

Juditsky, Rigollet, and Tsybakov (2008) introduced two conditions that guarantee fast rates in model selection aggregation. For now we focus on the following condition, mentioned in their Theorem 4.2, which we henceforth refer to as the *JRT-II condition*, returning to the JRT-I condition, mentioned in their Theorem 4.1, in Section 5.3.

Definition 4.8 (JRT-II condition) *Let $\eta > 0$. We say that $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the* η-JRT-II condition *if there exists a function $\gamma : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ satisfying (a) for all $f \in \mathcal{F}$, $\gamma(f, f) = 1$, (b) for all $f \in \mathcal{F}$, the function $g \mapsto \gamma(f, g)$ is concave, and (c)*

$$\text{for all } P \in \mathcal{P} \text{ and } f, g \in \mathcal{F}\colon \mathop{\mathbf{E}}_{Z\sim P}\left[e^{\eta\left(\ell_f(Z)-\ell_g(Z)\right)}\right] \leq \gamma(f, g). \tag{26}$$

This condition has been used to obtain fast $O(1/n)$ rates for the mirror averaging estimator in model selection aggregation, which is statistical learning against a finite class of functions $\mathcal{F} = \{f_1, \ldots, f_m\}$ (Juditsky et al., 2008). One may interpret their approach as using Vovk's aggregating algorithm to get $O(1)$ expected regret, and then applying online-to-batch conversion (Cesa-Bianchi et al., 2004; Barron, 1987; Yang and Barron, 1999), which leads to an estimator whose risk is upper bounded by the expected regret divided by n. This use of the AA is allowed, because, if $\mathcal{F}_{\mathrm{d}} \supseteq \mathrm{co}(\mathcal{F})$, then the JRT-II condition implies strong stochastic exp-concavity, as already shown by Audibert (2009) as part of the proof of his Corollary 5.1:

Proposition 4.9 *If $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the η-JRT-II condition, then $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies the strong η-stochastic exp-concavity condition for any $\mathcal{F}_{\mathrm{d}} \supseteq \mathrm{co}(\mathcal{F})$.*

Proof From the JRT-II condition, for all $P \in \mathcal{P}$ and $\Pi \in \Delta(\mathcal{F})$

$$\mathop{\mathbf{E}}_{g\sim\Pi}\mathop{\mathbf{E}}_{Z\sim P} e^{\eta(\ell_{\psi(\Pi)}(Z)-\ell_g(Z))} \leq \mathop{\mathbf{E}}_{g\sim\Pi}\gamma(\psi(\Pi), g),$$

which from the concavity of γ in its second argument is at most

$$\gamma\left(\psi(\Pi), \mathop{\mathbf{E}}_{g\sim\Pi} g\right) = \gamma\big(\psi(\Pi), \psi(\Pi)\big) = 1,$$

by the definition of ψ and part (a) of the JRT-II condition. Thus, we have

$$\mathop{\mathbf{E}}_{g\sim\Pi}\mathop{\mathbf{E}}_{Z\sim P} e^{\eta(\ell_{\psi(\Pi)}(Z)-\ell_g(Z))} \leq 1.$$

Applying Jensen's inequality to the exponential function completes the proof. ■

Juditsky et al. (2008) use the JRT-II condition in the proof of their Theorem 4.2 as a sufficient condition for another condition, which is then shown to imply $O(1/n)$ rates for finite classes $\mathcal{F}$. After some basic rewriting, this other condition (which requires the formula below Eq. (4.1) in their paper to be ≤ 0) is seen to be equivalent to strong stochastic exp-concavity as defined in Definition 4.7, i.e. it requires that (22) holds with $\varepsilon = 0$ and substitution function $\psi(\Pi) = \mathbf{E}_{f\sim\Pi}[f]$. The JRT-I condition, which we define in Section 5.3, can be related to stochastic exp-concavity with nonzero ε, thus we may say that the *underlying* condition that JRT work with is equivalent to our stochastic exp-concavity condition, albeit that they restrict themselves to a finite class of functions.

4.2.4 Relation to Audibert's Condition

Audibert (2009, p. 1596) presented a condition which he called the *variance inequality*. It is defined relative to a tuple $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ and has the following requirement as a special case (in Audibert's notation, this corresponds to $\delta_\lambda = 0$ and $\hat{\Pi}$ a Dirac distribution on some $f \in \mathcal{F}_{\mathrm{d}}$):

$$\forall \Pi \in \Delta(\mathcal{F})\ \exists f \in \mathcal{F}_{\mathrm{d}}\ \sup_{P\in\mathcal{P}}\ \mathop{\mathbf{E}}_{Z\sim P} \log \mathop{\mathbf{E}}_{g\sim\Pi}\left[e^{\eta(\ell_f(Z)-\ell_g(Z))}\right] \leq 0.$$

Rewriting

$$\mathop{\mathbf{E}}_{Z\sim P} \log \mathop{\mathbf{E}}_{g\sim\Pi}\left[e^{\eta(\ell_f(Z)-\ell_g(Z))}\right] = \eta \mathop{\mathbf{E}}_{Z\sim P}[\ell_f(Z) - m^{\eta}_{\Pi}(Z)],$$

this is seen to be precisely equivalent to strong stochastic mixability.

4.3 Relations with Central and Pseudoprobability Convexity Conditions

We now turn to the relations between stochastic mixability and the two main conditions from Section 3: the central condition and pseudoprobability convexity. We first define the predictor condition, which will act as an intermediate step, and then show the following implications:

$$\text{predictor} \Rightarrow \text{stochastic mixability} \Rightarrow \text{PPC} \Rightarrow \text{CC} \Rightarrow \text{predictor} \qquad \text{(under assumptions.)}$$

The implication from pseudoprobability convexity to the central condition was shown in Theorem 3.10 from Section 3.2; we will consider the other ones in turn in this section.

The second implication is of special interest since, in the online setting, there is extra power because predictions may take place in a set $\mathcal{F}_{\mathrm{d}}$ that can be larger than $\mathcal{F}$. The conditions of the second implication will identify situations in which this additional power is not helpful.

4.3.1 The Predictor Condition in General

We define the general predictor condition as follows:

Definition 4.10 (Predictor Condition) *Let $\eta > 0$ and $\varepsilon \geq 0$. We say that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies the* η-predictor condition *up to ε if there exists a* prediction function $\psi \colon \Delta(\mathcal{F}) \to \mathcal{F}_{\mathrm{d}}$ *such that*

$$\mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\Pi} \left[e^{\eta\left(\ell_{\psi(\Pi)}(Z) - \ell_f(Z)\right)} \right] \leq e^{\eta\varepsilon} \qquad \text{for all } P \in \mathcal{P} \text{ and distributions } \Pi \text{ on } \mathcal{F}. \tag{27}$$

If it satisfies the η-predictor condition up to 0, we say that the strong η-predictor condition *or simply the* η-predictor condition *holds. If it satisfies the η-predictor condition up to ε for all $\varepsilon > 0$, we say that the* weak η-predictor condition *holds; this is equivalent to*

$$\sup_{\Pi\in\Delta(\mathcal{F})} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{g\sim\Pi} \left[e^{\eta\left(\ell_f(Z) - \ell_g(Z)\right)} \right] \leq 1. \tag{28}$$

Comparing (28) to the central condition, we see that the predictor condition looks similar, except that the suprema over Π and P are interchanged. We note that, trivially, Fact 4.2 extends from η-stochastic mixability to the η-predictor condition.

4.3.2 Predictor Implies Stochastic Mixability

By an application of Jensen's inequality, the predictor condition always implies stochastic mixability, without any assumptions:

Proposition 4.11 *Suppose that $(\mathcal{P}, \ell, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies the η-predictor condition up to some $\varepsilon \geq 0$. Then it is η-stochastically mixable up to ε. In particular, the (strong) η-predictor condition implies (strong) η-stochastic mixability.*

Proof Let $P \in \mathcal{P}, \Pi \in \Delta(\mathcal{F})$ and $\varepsilon \geq 0$ be arbitrary. Then, by Jensen's inequality, the η-predictor condition up to ε implies

$$e^{\eta\varepsilon} \geq \mathop{\mathbf{E}}_{\substack{Z\sim P \\ f\sim\Pi}} \left[e^{\eta\left(\ell_{\psi(\Pi)}(Z) - \ell_f(Z)\right)} \right] = \mathop{\mathbf{E}}_{Z\sim P} \left[e^{\eta\left(\ell_{\psi(\Pi)}(Z) - m^{\eta}_{\Pi}(Z)\right)} \right] \geq e^{\eta\, \mathbf{E}_{Z\sim P}\left[\ell_{\psi(\Pi)}(Z) - m^{\eta}_{\Pi}(Z)\right]}.$$

Taking logarithms on both sides leads to $\mathbf{E}_{Z\sim P}\left[\ell_{\psi(\Pi)}(Z)\right] \leq \mathbf{E}_{Z\sim P}\left[m^{\eta}_{\Pi}(Z)\right] + \varepsilon$, which is η-stochastic mixability up to ε. ∎

4.3.3 Stochastic Mixability Implies Pseudoprobability Convexity

In Proposition 4.12 below, we show that, under the right assumptions, stochastic mixability implies pseudoprobability convexity.

A complication in establishing this implication is that stochastic mixability is defined relative to a four-tuple $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$, and allows us to play in a decision set that is different from $\mathcal{F}$, whereas the pseudoprobability convexity is defined relative to the triple $(\ell, \mathcal{P}, \mathcal{F})$. The proposition automatically holds if one takes $\mathcal{F} = \mathcal{F}_{\mathrm{d}}$, and then the implication follows trivially. In practice, however, we may have a non-convex model $\mathcal{F}$ — as is quite usual in e.g. density estimation — whereas the decision set $\mathcal{F}_{\mathrm{d}}$ for which we can establish that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is η-stochastically mixable is equal to the convex hull of $\mathcal{F}$. It would be quite disappointing if, in such cases, there would be no hope of getting fast rates for in-model statistical learning algorithms. The second part of the proposition shows that, luckily, fast rates are still possible under the following assumption:

Assumption B (model $\mathcal{F}$ and decision set $\mathcal{F}_{\mathrm{d}}$ equally good — $\mathcal{F}$ well-specified *relative* to $\mathcal{F}_{\mathrm{d}}$) *We say that Assumption B holds weakly for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$, if, for all $P \in \mathcal{P}$,*

$$\inf_{f \in \mathcal{F}} R(P, f) = \inf_{f \in \mathcal{F}_{\mathrm{d}}} R(P, f). \tag{29}$$

We say that Assumption B holds strongly if additionally, for all $P \in \mathcal{P}$, both infima are achieved: $\min_{f \in \mathcal{F}} R(P, f) = \min_{f \in \mathcal{F}_{\mathrm{d}}} R(P, f)$.

The strong version of Assumption B implies Assumption A and will be used further on in Theorem 4.14. In a typical application of the proposition below, the weak Assumption B would be assumed relative to a $\mathcal{F}_{\mathrm{d}}$ such that $\mathcal{F} \subset \mathcal{F}_{\mathrm{d}}$.

Proposition 4.12 *Suppose that Assumption B holds weakly for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$. If $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is η-stochastically mixable up to some $\varepsilon \geq 0$, then $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the η-pseudoprobability convexity condition up to δ for any $\delta > \varepsilon$; in particular, weak η-stochastic mixability of $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ implies the weak η-PPC condition for $(\ell, \mathcal{P}, \mathcal{F})$. Moreover, if Assumption A also holds and $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies strong η-stochastic mixability, then $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the strong η-PPC condition.*

If Assumption A and the weak version of Assumption B both hold, then, using this proposition, if we have η-stochastic mixability for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ we can directly conclude from Theorem 3.10 that we also have the η-central condition for $(\ell, \mathcal{P}, \mathcal{F})$. So when does Assumption B hold? Let us assume that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies η-stochastic mixability. In all cases we are aware of, it then also satisfies η-stochastic mixability for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}'_{\mathrm{d}})$, where $\mathcal{F}'_{\mathrm{d}}$ is equal to, or an arbitrary superset of, $\mathrm{co}(\mathcal{F})$ — in the special case of η-stochastic exp-concavity this actually follows by definition. An extreme case occurs if we take $\mathcal{F}'_{\mathrm{d}} := \mathcal{F}_{\ell}$ to be the set of all functions that can be defined on a domain (Example 2.1). Then Assumption B expresses that the model $\mathcal{F}$ is well-specified. But the assumption is weaker: assuming again that $\mathcal{F}_{\mathrm{d}}$ can be taken to be the convex hull of $\mathcal{F}$, it also holds if $\mathcal{F}$ is itself convex and contains, for all $P \in \mathcal{P}$, a risk minimizer; and also, if, more weakly still, $\mathcal{F}$ is convex 'in the direction facing P'. Note that, for the log-loss, we already knew that the 1-central condition holds under this condition, from the Bayesian interpretation in Section 3.3. There we also established a generalization to other loss functions: the η-central condition holds if the set of pseudoprobabilities $\mathcal{P}_{\mathcal{F}}$ is convex 'in the direction facing P' (Figure 2). But, for all loss functions except log-loss, that was a condition involving *pseudo*probabilities and *artificial* (mix) losses. The novelty of Proposition 4.12 is that, if η-stochastic mixability holds for

$(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ with $\mathcal{F}_{\mathrm{d}} = \mathrm{co}(\mathcal{F})$ (as e.g. when we have η-stochastic exp-concavity), then the result generalizes further to 'the η-central condition holds if the set $\mathcal{F}$ *itself* (rather than the artificial set $\mathcal{P}_{\mathcal{F}}$) is convex in the direction facing P'.

Example 4.13 (Fast Rates in Expectation rather than Probability) Fast rate results proved under the η-central condition, such as our result in Section 7 and the various results by Zhang (2006b) generally hold both in expectation and in probability. The situation is different for η-stochastic mixability: extending the analysis of Vovk's Aggregating Algorithm to tuples $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ and using the online-to-batch conversion, we can only prove a fast rate result in expectation, and not in probability. Audibert (2007) provides a by now well-known example $(\ell^{\mathrm{sq}}, \mathcal{P}, \mathcal{F}, \mathrm{co}(\mathcal{F}))$ with squared loss in which the rate obtained by the exponentially weighted forecaster (the aggregating algorithm applied with $\psi(\Pi) = \mathbf{E}_{f\sim\Pi}[f]$) followed by online-to-batch conversion is $O(1/n)$ in expectation, yet only $\asymp 1/\sqrt{n}$ in probability; and ERM also gives a rate, both in-probability and in-expectation of $1/\sqrt{n}$ (Theorem 2 of (Audibert, 2007)). As might then be expected, in Audibert's decision problem η-exp-concavity holds for some $\eta > 0$ yet the central condition does not hold for any $\eta > 0$. Proposition 4.12 then implies that Assumption B must be violated: the best $f \in \mathrm{co}(\mathcal{F})$ is better than the best $f \in \mathcal{F}$. Inspection of the example shows that this indeed the case (a related point was made earlier by Lecué (2011)). ■

Proof (of Proposition 4.12) Note that (22), the definition of η-stochastic mixability up to ε, can be rewritten as

$$\forall \Pi \in \Delta(\mathcal{F})\ \exists f \in \mathcal{F}_{\mathrm{d}}\ \forall P \in \mathcal{P}:\ \mathop{\mathbf{E}}_{Z\sim P}[\ell_f(Z)] \leq \mathop{\mathbf{E}}_{Z\sim P}\left[m^{\eta}_{\Pi}(Z)\right] + \varepsilon.$$

This trivially implies

$$\forall \Pi \in \Delta(\mathcal{F})\ \forall P \in \mathcal{P}\ \exists f \in \mathcal{F}_{\mathrm{d}}:\ \mathop{\mathbf{E}}_{Z\sim P}[\ell_f(Z)] \leq \mathop{\mathbf{E}}_{Z\sim P}\left[m^{\eta}_{\Pi}(Z)\right] + \delta, \tag{30}$$

for any $\delta \geq \varepsilon$. This implies that for any $\delta > \varepsilon$, we can assume that the choice of f in (30) only depends on P and not on Π. We would therefore obtain η-pseudoprobability convexity up to any $\delta > \varepsilon$ of $(\ell, \mathcal{P}, \mathcal{F})$ if we could replace $\mathcal{F}_{\mathrm{d}}$ by $\mathcal{F}$, which is trivial if $\mathcal{F}_{\mathrm{d}} = \mathcal{F}$ and allowed under Assumption B because it implies that, for any $f \in \mathcal{F}_{\mathrm{d}}$ we can find $f' \in \mathcal{F}$ such that $\mathbf{E}_{Z\sim P}\left[\ell_{f'}(Z)\right] - \mathbf{E}_{Z\sim P}\left[\ell_f(Z)\right] \leq \delta - \varepsilon$.

For the final implication, note that under Assumption A we can choose $\delta = \varepsilon$, and by Corollary 3.11 we can choose $\varepsilon = 0$. ■

4.3.4 The Central Condition Implies the Predictor Condition

We proceed to study when the central condition implies the predictor condition (with $\mathcal{F}_{\mathrm{d}} = \mathcal{F}$), which requires the strongest assumptions among the implications we consider. We first identify a minimax identity (32) that is sufficient by itself (Theorem 4.14), but difficult to verify directly. We therefore weaken Theorem 4.14 to Theorem 4.17 by providing sufficient conditions (Assumption D) for the minimax identity.

For any Π and η, define the function

$$S^{\eta}_{\Pi}(P,f) = \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{g\sim\Pi} \left[e^{\eta\left(\ell_f(Z)-\ell_g(Z)\right)} \right],$$

which is the main quantity in the definitions of both the central and the predictor condition.

Assumption C (Minimax Assumption) *For given $\eta > 0$, we say that the* η-minimax assumption *is satisfied for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ if, for all $\Pi \in \Delta(\mathcal{F})$ and for all $C \geq 1$, the following implication holds:*

$$\sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} S^{\eta}_{\Pi}(P,f) \leq C \qquad \Longrightarrow \qquad \inf_{f\in\mathcal{F}_{\mathrm{d}}} \sup_{P\in\mathcal{P}} S^{\eta}_{\Pi}(P,f) \leq C. \tag{31}$$

We call this the minimax assumption, because (31) is implied by the minimax identity

$$\sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} S^{\eta}_{\Pi}(P,f) = \inf_{f\in\mathcal{F}_{\mathrm{d}}} \sup_{P\in\mathcal{P}} S^{\eta}_{\Pi}(P,f). \tag{32}$$

Theorem 4.14 below implies that Assumption C is sufficient for the central condition to imply the predictor condition, with $\mathcal{F}_{\mathrm{d}} = \mathcal{F}$. Intuitively, Assumption C should hold under broad conditions — just like standard minimax theorems hold under broad conditions. Below we will identify the specific, less elegant but more easily verifiable Assumption D that implies Assumption C. However, like conditions for standard minimax theorems, in some cases Assumption D requires $\mathcal{F}_{\mathrm{d}} \subset \mathbb{R}$ to be compact, yet we want to apply the theorem also in cases where $\mathcal{F} = \mathbb{R}$. As shown in Example 4.21, in this case we can sometimes still use Part (b) of the result, which implies that the assumption is still sufficient if we take a smaller set $\mathcal{F}_{\mathrm{d}} \subset \mathcal{F}$ that satisfies Assumption B. Note that Assumption B also played a crucial role in going from stochastic mixability of $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ to the PPC condition for $(\ell, \mathcal{P}, \mathcal{F})$.

Theorem 4.14 *Consider a decision problem $(\ell, \mathcal{P}, \mathcal{F})$. Suppose that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is such that the the η-minimax assumption (Assumption C) holds. Then*

(a) if $\mathcal{F} = \mathcal{F}_{\mathrm{d}}$ and the η-central condition holds up to some $\varepsilon \geq 0$ for $(\ell, \mathcal{P}, \mathcal{F})$, then the η-predictor condition holds up to any $\delta > \varepsilon$ for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$. In particular, the weak η-central condition implies the weak η-predictor condition. Moreover,

(b) if $\mathcal{F} \supseteq \mathcal{F}_{\mathrm{d}}$ and $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies the strong version of Assumption B, then the weak η-central condition for $(\ell, \mathcal{P}, \mathcal{F})$ implies the weak η-predictor condition for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ and therefore also for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F})$.

Once we establish that the η-predictor condition holds for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ with $\mathcal{F}_{\mathrm{d}} \subset \mathcal{F}$, by Fact 4.2 we can also infer that the η-predictor condition holds for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}'_{\mathrm{d}})$ for any $\mathcal{F}'_{\mathrm{d}} \supset \mathcal{F}_{\mathrm{d}}$, in particular for $\mathcal{F}'_{\mathrm{d}} = \mathcal{F}$.

Proof For Part (a), from the η-central condition up to ε and the fact that the sup inf never exceeds the inf sup and that $\mathcal{F} = \mathcal{F}_{\mathrm{d}}$, we get

$$e^{\eta\varepsilon} \geq \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \sup_{\Pi\in\Delta(\mathcal{F})} S^{\eta}_{\Pi}(P,f) \geq \sup_{P\in\mathcal{P}} \sup_{\Pi\in\Delta(\mathcal{F})} \inf_{f\in\mathcal{F}_{\mathrm{d}}} S^{\eta}_{\Pi}(P,f) = \sup_{\Pi\in\Delta(\mathcal{F})} \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} S^{\eta}_{\Pi}(P,f). \tag{33}$$

This establishes that the premise of (31) holds with $C = e^{\eta\varepsilon}$ for all $\Pi \in \Delta(\mathcal{F})$. Hence Assumption C tells us that the conclusion of (31) must also hold for all $\Pi \in \Delta(\mathcal{F})$, and therefore

$$\sup_{\Pi\in\Delta(\mathcal{F})} \inf_{f\in\mathcal{F}} \sup_{P\in\mathcal{P}} S^{\eta}_{\Pi}(P,f) \leq e^{\eta\varepsilon}.$$

Since we are not guaranteed that the infimum over f is achieved, this implies the η-predictor condition up to any $\delta > \varepsilon$, but not necessarily for $\delta = \varepsilon$. We thus obtain the first part of the theorem.

For Part (b), we note that, by the premise, Assumption A must hold and we can apply Corollary 3.11 which tells us that for all $P \in \mathcal{P}$, the $f^*_P \in \mathcal{F}$ minimizing $R(P,f)$ is essentially unique and that the strong η-central condition holds, i.e. for all $P \in \mathcal{P}$, (4) holds. As explained below (4), this implies that $f'_P = \phi(P)$ is $\mathcal{F}$-optimal for P, hence it follows that $f'_P = f^*_P$, P-almost surely. The strong version of Assumption B then implies that $\mathcal{F}_{\mathrm{d}}$ contains a g^*_P with $P(\ell_{f^*_P} = \ell_{g^*_P}) = 1$. We now have, by the strong η-central condition, that for all $\Pi \in \Delta(\mathcal{F})$,

$$\begin{aligned} 1 &\geq \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}} \sup_{\Pi\in\Delta(\mathcal{F})} S^{\eta}_{\Pi}(P,f) = \sup_{P\in\mathcal{P}} \sup_{\Pi\in\Delta(\mathcal{F})} S^{\eta}_{\Pi}(P,f'_P) = \sup_{P\in\mathcal{P}} \sup_{\Pi\in\Delta(\mathcal{F})} S^{\eta}_{\Pi}(P,g^*_P) \\ &\geq \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \sup_{\Pi\in\Delta(\mathcal{F})} S^{\eta}_{\Pi}(P,f). \end{aligned}$$

We have thus established the first inequality of (33) with $\varepsilon = 0$; we can now proceed as in the first part. ■

We proceed to identify more concrete conditions that are sufficient for Assumption C. To this end, we will endow the set of finite measures (including all probability measures) on $\mathcal{Z}$ with the *weak topology* (Billingsley, 1968; Van der Vaart and Wellner, 1996), for which convergence of a sequence of measures $P_1, P_2, \ldots$ to P means that

$$\mathop{\mathbf{E}}_{Z\sim P_n}[h(Z)] \to \mathop{\mathbf{E}}_{Z\sim P}[h(Z)] \tag{34}$$

for any bounded, continuous function $h\colon \mathcal{Z} \to \mathbb{R}$. To make continuity of h well-defined, we then also need to assume a topology on $\mathcal{Z}$. It is standard to assume that $\mathcal{Z}$ is a *Polish space* (i.e. that it is a complete separable metric space), because then, from Prokhorov (1956), there exists a metric for which the set of finite measures on $\mathcal{Z}$ is a Polish space as well and for which convergence in this metric is equivalent to (34). The weak topology is the topology induced by this metric.

We shall also assume that $\mathcal{P}$ is *tight*, which means that, for any $\varepsilon > 0$, there must exist a compact event $A \subseteq \mathcal{Z}$ such that $P(A) \geq 1 - \varepsilon$ for all $P \in \mathcal{P}$. This is a weaker condition than assuming that the whole space $\mathcal{Z}$ is compact because it allows some probability mass outside of the compact event A.

Assumption D *Suppose the set of possible outcomes $\mathcal{Z}$ is a Polish space. Let $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$, $\Pi \in \Delta(\mathcal{F})$ and $\eta > 0$ be given. Then assume that all of the following are satisfied:*

1. *For all $f \in \mathcal{F} \cup \mathcal{F}_{\mathrm{d}}$, $\ell_f(z)$ is continuous in z and $\ell_f(z) \geq 0$.*

2. *The set* $\mathcal{F}_{\mathrm{d}}$ *is convex and, for any* $z \in \mathcal{Z}$, $e^{\eta \ell_f(z)}$ *is convex in* f *on* $\mathcal{F}_{\mathrm{d}}$.
3. *The set* $\mathcal{P}$ *is convex and tight.*
4. *Either a)* $\mathcal{P}$ *is closed in the weak topology; or b)* $\mathcal{F}_{\mathrm{d}}$ *is a totally bounded metric space, and, for every compact subset* $\mathcal{Z}'$ *of* $\mathcal{Z}$, *the family of functions* $\{f \mapsto \ell_f(z) : z \in \mathcal{Z}'\}$ *is uniformly equicontinuous on* $\mathcal{F}_{\mathrm{d}}$.
5. *The random variables* $\xi_{Z,f} = \mathbf{E}_{g \sim \Pi}\left[e^{\eta\left(\ell_f(Z) - \ell_g(Z)\right)}\right]$ *are* uniformly integrable *over* $f \in \mathcal{F}_{\mathrm{d}}, P \in \mathcal{P}$ *in the sense that*

$$\lim_{b \to \infty} \sup_{f \in \mathcal{F}_{\mathrm{d}}, P \in \mathcal{P}} \mathop{\mathbf{E}}_{Z \sim P} \left[\xi_{Z,f} \left[\!\left[\xi_{Z,f} \geq b\right]\!\right]\right] = 0. \tag{35}$$

While these assumptions may look daunting, they actually hold in many situations even with unbounded losses, as our examples below illustrate. In D.1, continuity is automatic for finite and countable $\mathcal{Z}$ as long as we take the discrete topology. In D.2, convexity of $e^{\eta \ell_f(z)}$ in f is implied by convexity of $\ell_f(z)$ in f. Regarding the fourth requirement, D.4: the condition that $\mathcal{P}$ is weakly closed is easily stated but hard to verify for general $\mathcal{Z}$ and $\mathcal{P}$; the alternative condition is hard to state but often straightforward to verify. And finally, D.5 will automatically hold for all bounded loss functions and for many unbounded losses as well; for a discussion of uniform integrability as used in D.5, see Shiryaev (1996, pp. 188–190). In particular, Lemma 3 on p. 190, specialised to our context, implies the following sufficient condition:

Lemma 4.15 (Sufficient Condition for D.5) *For a fixed choice of* $\Pi \in \Delta(\mathcal{F})$, *let* $\xi_{Z,f}$ *be as in Assumption D.5. Then* (35) *is satisfied if*

$$\sup_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{Z \sim P} \left[G(\xi_{Z,f})\right] < \infty$$

for any function $G \colon [0, \infty) \to \mathbb{R}$ *that is bounded below and is such that*

$$\frac{G(t)}{t} \text{ is increasing,} \quad \text{and} \quad \frac{G(t)}{t} \to \infty. \tag{36}$$

We may, for instance, take $G(t) = t^2$ or $G(t) = t \log t$.

Proof Without loss of generality, we may assume that G is non-negative. Otherwise replace $G(t)$ by $\max\{G(t), 0\}$, which preserves (36) and adds at most $-\inf_t G(t) < \infty$ to $\sup_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathbf{E}_{Z \sim P}\left[G(\xi_{Z,f})\right]$.

Now let $M = \sup_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathbf{E}_{Z \sim P}\left[G(\xi_{Z,f})\right]$ and, for any $\varepsilon > 0$, take $b > 0$ large enough that $G(t)/t \geq M/\varepsilon$ for all $t \geq b$. Then

$$\begin{aligned} 0 \leq \sup_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{Z \sim P} \left[\xi_{Z,f} \left[\!\left[\xi_{Z,f} \geq b\right]\!\right]\right] &\leq \frac{\varepsilon}{M} \sup_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{Z \sim P} \left[G(\xi_{Z,f}) \left[\!\left[\xi_{Z,f} \geq b\right]\!\right]\right] \\ &\leq \frac{\varepsilon}{M} \sup_{f \in \mathcal{F}_{\mathrm{d}}} \sup_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{Z \sim P} \left[G(\xi_{Z,f})\right] \leq \varepsilon, \end{aligned}$$

from which (35) follows by letting ε tend to 0. ■

Assumption D is sufficient for the minimax assumption, as our main technical result of this section (proof deferred to Appendix A.2) shows:

Lemma 4.16 *Fix $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ and $\eta > 0$. If Assumption D is satisfied for a given $\Pi \in \Delta(\mathcal{F})$, then (32) also holds. Consequently, if Assumption D is satisfied for all $\Pi \in \Delta(\mathcal{F})$, then that implies Assumption C.*

Together, Theorem 4.14 and Lemma 4.16 prove the following theorem.

Theorem 4.17 (**Central to Predictor**) *Let $\eta > 0$ and suppose Assumption D holds for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ for all $\Pi \in \Delta(\mathcal{F})$. If either $\mathcal{F} = \mathcal{F}_{\mathrm{d}}$ or the strong version of Assumption B holds and $\mathcal{F} \supset \mathcal{F}_{\mathrm{d}}$, then the weak η-central condition implies the weak η-predictor condition.*

We now provide some examples which indicate that while Assumption D covers several non-trivial cases — including non-compact $\mathcal{F}$ — it is probably still significantly more restrictive than needed.

Example 4.18 (Logarithmic Loss) Consider a set of distributions $\mathcal{P}$ on some set $\mathcal{Z}$ and let $\mathcal{F}$ either be the densities or mass functions corresponding to $\mathcal{P}$ or an arbitrary convex set of densities on $\mathcal{Z}$. By Example 2.2, $(\ell^{\log}, \mathcal{P}, \mathcal{F})$ satisfies the 1-central condition. If we further assume that $\mathcal{P}$ is convex and tight and that there is a $\delta > 0$ such that for all $z \in \mathcal{Z}$, all $f \in \mathcal{F}$, $f(z) \geq \delta$ (so that the densities are bounded from below), then Assumption D is readily verified and we can conclude from the theorem that the 1-predictor condition and hence 1-stochastic mixability holds for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F})$. We know however, because log-loss is 1-(Vovk-) mixable, that 1-stochastic mixability must even hold if $\mathcal{P}$ is neither convex nor tight; Assumption D is not weak enough to handle this case, so the example suggests that a further weakening might be possible. Also, we know that 1-stochastic mixability continues to hold if $\delta = 0$; verification of Assumption D is not straightforward in this case, which suggests that a simplification of the assumption is desirable. ■

Example 4.19 (0/1-Loss , Example 3.8, Continued.) Consider the setting of Example 3.8 and Example 4.4 with decision problem $(\ell^{01}, \mathcal{P}_\delta, \mathcal{F})$ and $\delta > 0$. We established in Example 3.8 that the η-central condition then holds for some $\eta > 0$, but also, in Example 4.4, that $(\ell^{01}, \mathcal{P}_\delta, \mathcal{F}, \mathcal{F})$ is not η-stochastically mixable. We would thus expect Assumption D to fail here, which it does, since $\mathcal{F} = \mathcal{F}_{\mathrm{d}}$ is not convex. ■

Example 4.20 (Squared Loss, Restricted Domain) Let ℓ be the squared loss $\ell_f^{\mathrm{sq}}(z) := \frac{1}{2}(z - f)^2$ on the restricted spaces $\mathcal{Z} = \mathcal{F} = \mathcal{F}_{\mathrm{d}} = [-B, B]$ as in Example 4.3, and take $\mathcal{P}$ to be the set of all possible distributions on $\mathcal{Z}$. Then the first three requirements of Assumption D may be verified by observing that $\ell_f^{\mathrm{sq}}(z)$ (and therefore also $e^{\eta \ell_f^{\mathrm{sq}}(z)}$) is convex in f, and that $\mathcal{P}$ is trivially tight by taking $A = \mathcal{Z}$. Now $\mathcal{P}$ is actually closed in the weak topology, but, in order to satisfy the fourth condition, we might also use that the mappings $\{f \mapsto \ell_f^{\mathrm{sq}}(z) : z \in \mathcal{Z}\}$ are all Lipschitz with the same Lipschitz constant $(2B)$, which implies that they are also uniformly equicontinuous. Finally, to see that the fifth requirement is satisfied for any $\Pi \in \Delta(\mathcal{F})$, we may appeal to Lemma 4.15 with $G(t) = t^2$ and use that ℓ^{sq} is uniformly bounded.

Then all parts of Assumption D are satisfied for all $\Pi \in \Delta(\mathcal{F})$. We know from Example 4.3 that in this case classical η-mixability holds for $\eta = 1/B^2$. This implies strong η-stochastic mixability, which implies the strong η-pseudoprobability convexity condition (by Proposition 4.12). Since Assumption A holds, this in turn implies the strong η-central condition (by Theorem 3.10), and by applying Theorem 4.17 one can then infer the weak η-predictor condition. ■

In the example above, the set $\mathcal{P}$ was convex and, by boundedness of $\mathcal{Z}$, automatically tight and thus the η-central condition and η-stochastic mixability both hold. In Example 3.5 we established the η-central condition for a set $\mathcal{P}$ that is neither convex nor tight, so Assumption D fails and we cannot apply Theorem 4.17 to jump from the η-central to the η-predictor condition as in Example 4.20. However, as the next example shows, if we replace $\mathcal{P}$ by its convex hull for a restricted range of μ, then we can recover the predictor condition via Theorem 4.17 after all; restriction of $\mathcal{F}$, however, is not needed.

Example 4.21 (Squared Loss, Unrestricted Domain: Example 3.5, Continued.) Consider the squared loss $\ell_f^{\mathrm{sq}}(z) = \frac{1}{2}(z-f)^2$, and let $\mathcal{Z} = \mathbb{R}$, $\mathcal{F} = [-B, B]$ (later we will consider $\mathcal{F} = \mathbb{R}$), and let $\mathcal{P} = \mathrm{co}(\{\mathcal{N}(\mu, 1) : \mu \in [-M, M]\})$ be the convex hull of the set of normal distributions with unit variance and means bounded by $M \leq B$. We may represent any $P \in \mathcal{P}$ as a mixture of $\mathcal{N}(\mu, 1)$ under some distribution w on μ. Let μ_P be the mean of P. Then, for all $P \in \mathcal{P}$ with corresponding w and all $t \in \mathbb{R}$,

$$\mathop{\mathbf{E}}_{Z\sim P}\left[e^{t(Z-\mu_P)}\right] = \int_{-M}^{M} \mathop{\mathbf{E}}_{Z\sim\mathcal{N}(\mu,1)}\left[e^{t(Z-\mu_P)}\right] \mathrm{d}w(\mu) = e^{t^2/2}\int_{-M}^{M} e^{t(\mu-\mu_P)}\mathrm{d}w(\mu) \leq e^{t^2/2}e^{t^2M^2/2},$$

where the last inequality follows from Hoeffding's bound on the moment generating function and the observation that $\mu_P = \mathbf{E}_{\mu\sim w}[\mu]$. Thus the elements of $\mathcal{P}$ are all subgaussian with variance $\sigma^2 = 1 + M^2$. Hence, by the argument in Example 3.6, the strong η-central condition is satisfied for $\eta \leq 1/(1+M^2)$ and with substitution function $\phi(P) = \mu_P$.

In order to also get the predictor condition via Theorem 4.17, we need to verify Assumption D. The first three parts of this assumption may be readily verified, and part b) of D.4 also holds, because the mappings $\{f \mapsto \frac{1}{2}(z-f)^2 : z \in [-A, A]\}$ are all $(2A)$-Lipschitz, which implies their uniform equicontinuity, for any choice of A. Finally, Assumption D.5 follows from Lemma 4.15 with $G(t) = t^2$ and Jensen's inequality:

$$\begin{aligned}
&\sup_{f\in\mathcal{F}}\sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P}\left[\mathop{\mathbf{E}}_{g\sim\Pi}[e^{\eta(\ell_f^{\mathrm{sq}}(Z)-\ell_g^{\mathrm{sq}}(Z))}]\right]^2 \leq \sup_{f\in\mathcal{F}}\sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{g\sim\Pi}\left[e^{2\eta(\ell_f^{\mathrm{sq}}(Z)-\ell_g^{\mathrm{sq}}(Z))}\right] \\
&\quad\leq \sup_{f,g\in\mathcal{F}}\sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P}\left[e^{2\eta(\ell_f^{\mathrm{sq}}(Z)-\ell_g^{\mathrm{sq}}(Z))}\right] = \sup_{f,g\in\mathcal{F}}\sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P}\left[e^{2\eta(f^2+2Z(g-f)-g^2)}\right] \\
&\quad\leq e^{2\eta B^2}\sup_{f,g\in\mathcal{F}}\sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P}\left[e^{4\eta Z(g-f)}\right] \overset{(*)}{\leq} e^{2\eta B^2}\sup_{f,g\in\mathcal{F}}\sup_{P\in\mathcal{P}} e^{8\eta^2(g-f)^2(1+M^2)+4\eta(g-f)\mu_P} < \infty,
\end{aligned}$$

where $(*)$ follows from $(1+M^2)$-subgaussianity. Thus, Theorem 4.17 can be applied to establish the weak η-predictor condition for squared loss on an unbounded domain $\mathcal{Z} = \mathbb{R}$ for the choices of η, $\mathcal{F}_{\mathrm{d}} = \mathcal{F}$ and $\mathcal{P}$ described above.

Now consider the case where we set $\mathcal{F} = \mathbb{R} = \mathcal{Z}$ and leave everything else unchanged. Then by the argument in Example 3.6, the strong η-central condition is still satisfied for $\eta \leq 1/(1+M^2)$, but we cannot directly use Theorem 4.17 to establish the weak predictor condition for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F})$. All steps of the above reasoning go through except part b) of D.4, since $\mathcal{F}$ is no longer compact. However, if we take $\mathcal{F}_{\mathrm{d}} = [-B, B]$ for $B \geq M$, then Assumption D.4 (which only refers to $\mathcal{F}_{\mathrm{d}}$, not to $\mathcal{F}$) holds after all. Moreover, the strong version of Assumption B also holds, because $\arg\min_{f \in \mathbb{R}} \mathbf{E}_{Z \sim P}(Z-f)^2 = \mu_P$. We can thus use Theorem 4.17 to conclude that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ satisfies the weak η-predictor condition. It then follows by Fact 4.2 that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F})$ satisfies the weak η-predictor condition as well. We conclude that the implication η-central $\Rightarrow$ weak η-predictor goes through, even though $\mathcal{F}$ is not compact. ■

This final example shows how Theorem 4.17 allows us to find assumptions on $\mathcal{P}$ that are sufficient for establishing the weak predictor condition, and therefore weak stochastic mixability, for squared loss on the unbounded domain $\mathbb{R}$. As discussed by Vovk (2001, Section 5), this is a case where the classical mixability analysis does not apply.

5. Intermediate Rates: The Central Condition, the Margin Condition and the Bernstein condition

In this section, we weaken the η-central and η-PPC conditions to the v-central and v-PPC conditions, which allow $\eta = v(\varepsilon)$ to depend on ε according to a function v that is allowed to go to 0 as ε goes to 0. In the main result of this section, Theorem 5.4 in Section 5.1, we establish that for bounded loss functions, these weakened versions of our conditions are essentially equivalent to a generalized *Bernstein condition* which has been used before to characterize fast rates. Section 5.2 shows that, for unbounded loss functions, the one-sidedness of our conditions allows them to capture situations in which fast rates are attainable yet the Bernstein condition does not hold — although there are also situations in which the Bernstein condition holds whereas the v-central condition does not for any allowed v (although the v-PPC condition does). Thus, as a corollary we find that the equivalence between the central and PPC condition breaks for the weaker, v-versions of these conditions. Section 5.3 illustrates that η-stochastic mixability can be weakened similarly to v-stochastic mixability and relates this to a condition identified by Juditsky et al. (2008). Finally, in Section 5.4 we apply Theorem 5.4 to show how the central condition is related to (non-) existence of unique risk minimizers.

5.1 The v-Conditions and the Bernstein Condition

Empirical risk minimization (ERM) achieves fast rates if the random deviations of the empirical excess risk are small compared to the true excess risk. As shown by Tsybakov (2004), this is the case in classification if the Bayes-optimal classifier is in the model $\mathcal{F}$ and the so-called *margin*, which measures the difference between the conditional probabilities of the labels given the features and the uniform distribution, is large. Technically, the random deviations can be controlled in this case, because the second moment of the excess loss can be bounded in terms of the first moment. In fact, as shown by Bartlett and Mendelson

(2006), this condition, which they call the *Bernstein condition*, is sufficient for fast rates for bounded losses in general, even if the Bayes-optimal decision is not in the model. Precisely, the standard Bernstein condition is defined as follows:

Definition 5.1 (Bernstein Condition) *Let $\beta \in (0,1]$ and $B \geq 1$. Then $(\ell, P, \mathcal{F})$ satisfies the* (β, B)-Bernstein condition *if there exists an $f^* \in \mathcal{F}$ such that*

$$\mathop{\mathbf{E}}_{Z\sim P}\left[\left(\ell_f(Z) - \ell_{f^*}(Z)\right)^2\right] \leq B\left(\mathop{\mathbf{E}}_{Z\sim P}\left[\ell_f(Z) - \ell_{f^*}(Z)\right]\right)^\beta \qquad \text{for all } f \in \mathcal{F}. \tag{37}$$

This standard definition bounds the second moment in terms of the polynomial function $u(x) = Bx^\beta$ of the first moment.[6] The exponent β is most important, because it determines the order of the rates, whereas the scaling factor B only matters for the constants. To draw the connection with the central condition, however, it will be clearer to allow general functions u instead of $x \mapsto Bx^\beta$. Following Koltchinskii (2006) and Arlot and Bartlett (2011), we then bound the variance instead of the second moment, which is equivalent with respect to the rates that can be obtained:

Definition 5.2 (Generalized Bernstein Condition) *Let $u : [0,\infty) \to [0,\infty)$ be a non-decreasing function such that $u(x) > 0$ for all $x > 0$, and $u(x)/x$ is non-increasing. We say that $(\ell, \mathcal{P}, \mathcal{F})$ satisfies the* u-Bernstein condition *if, for all $P \in \mathcal{P}$, there exists an $\mathcal{F}$-optimal $f^* \in \mathcal{F}$ (satisfying (3)) such that*

$$\mathop{\mathbf{Var}}_{Z\sim P}\left(\ell_f(Z) - \ell_{f^*}(Z)\right) \leq u\left(\mathop{\mathbf{E}}_{Z\sim P}\left[\ell_f(Z) - \ell_{f^*}(Z)\right]\right) \qquad \text{for all } f \in \mathcal{F}. \tag{38}$$

In particular $u(x) = Bx^\beta$ is allowed for $\beta \in [0,1]$, or, more generally, it is sufficient if $u(0) = 0$ and u is a non-decreasing concave function, because then the slope $u(x)/x = (u(x) - u(0))/x$ is non-increasing; for a concrete example see Example 5.5 below.

Similar generalizations have been proposed by Koltchinskii (2006) and Arlot and Bartlett (2011)[7]. For bounded losses, our generalized Bernstein condition is equivalent to a generalization of the central condition in which $\eta = v(\varepsilon)$ is allowed to depend on ε according to some function v, which in turn is equivalent to the analogous generalization of the pseudoprobability-convexity condition. We first introduce these generalized concepts and then show how they are related to the Bernstein condition. They are defined as immediate generalizations of their corresponding definitions, Definition 3.1, Equation (12) and Definition 3.2, Equation (15):

Definition 5.3 (v-Central Condition and v-PPC Condition) *Let $v\colon [0,\infty) \to [0,\infty)$ be a bounded, non-decreasing function satisfying $v(x) > 0$ for all $x > 0$. We say that*

6. The Tsybakov condition with exponent q (Tsybakov, 2004) is the special case that the (β, B)-Bernstein condition holds for $B < \infty$, $q = \beta/(1-\beta)$, additionally requiring ℓ to be classification loss and $\mathcal{F}$ to contain the Bayes classifier for P.

7. They require u to be of the form w^2 where w is a concave increasing function with $w(0) = 0$. In their examples, w^2 is also concave, a case which is subsumed by our condition, but they additionally allow concave w with convex $u = w^2$, which is not covered by our condition. On the other hand, our condition allows u with non-concave $\sqrt{u}$, which is not covered by theirs. For example, $u(x) = (x - 1/3)^3 + 1/27$ for $x \leq 1/2$ and $u(x) = x/12$ for $x > 1/2$ satisfies our condition, but $\sqrt{u(x)}$ is nonconcave. So, in general, the conditions are incomparable.

$(\ell, \mathcal{P}, \mathcal{F})$ *satisfies the* v-central condition *if, for all* $\varepsilon \geq 0$*, there exists a function* $\phi : \mathcal{P} \to \mathcal{F}$ *such that* (12) *is satisfied with* $\eta = v(\varepsilon)$*. We say that* $(\ell, \mathcal{P}, \mathcal{F})$ *satisfies the* v-pseudoprobability convexity (PPC) condition *if, for all* $\varepsilon \geq 0$*, there exists a function* $\psi : \mathcal{P} \to \mathcal{F}$ *such that* (15) *is satisfied with* $\eta = v(\varepsilon)$*.*

If $v(x) = \eta$ for all $x > 0$ and $v(0) = 0$, then the v-central condition is equivalent to the weak η-central condition. If $v(x) = \eta$ for all $x \geq 0$, then it is equivalent to the strong η-central condition.

Now consider a decision problem $(\ell, \mathcal{P}, \mathcal{F})$ such that Assumption A holds. Theorem 5.4 below in conjunction with Proposition 3.9 implies that the generalized Bernstein condition with function u, the v-central condition and the v-PPC condition are then all equivalent for bounded losses in the sense that one implies the other if

$$v(x) \cdot u(x) = c \cdot x \qquad \text{for all sufficiently small } x, \tag{39}$$

where c is a constant whose value depends on whether we are going from Bernstein to central or the other way around. In particular, if we ignore the unimportant difference between the second moment of $\ell_f(Z) - \ell_{f^*}(Z)$ and its variance, we see that the $(1, B)$-Bernstein condition and the η-central condition are equivalent for $\eta = c/B$.

Define the function $\kappa(x) := (e^x - x - 1)/x^2$ for $x \neq 0$, extended by continuity to $\kappa(0) = 1/2$, which is positive and increasing (Freedman, 1975).

Theorem 5.4 *For given* $(\ell, \mathcal{P}, \mathcal{F})$*, suppose that the losses* ℓ_f *take values in* $[0, a]$*.*

1. *If the* u*-Bernstein condition holds for a function* u *satisfying the requirements of Definition 5.2 (so that Assumption A holds), then*

   (a) *The* v*-central condition holds for*

   $$v(x) = \frac{c_1^b x}{u(x)} \wedge b,$$

   where $b > 0$ *can be any finite constant and* $c_1^b = 1/\kappa(2ba)$*; and if* $u(0) = 0$ *we read* $0/u(0)$ *as* $\liminf_{x \downarrow 0} x/u(x)$*.*

   (b) *Additionally, for each* $P \in \mathcal{P}$*, any* $\mathcal{F}$*-optimal* f^* *for* P*, and any* $\delta > 0$*, we have* $\mathbf{E}_{Z \sim P}[e^{\eta(\ell_{f^*}(Z) - \ell_f(Z))}] \leq 1$ *for all* f *with* $R(P, f) - R(P, f^*) \geq \delta$*, where* $\eta = v(\delta)$*.*

2. *On the other hand, suppose that Assumption A holds. If the* v*-pseudoprobability convexity condition holds for a function* v *satisfying the requirements of Definition 5.3 such that* $x/v(x)$ *is nondecreasing, then the* u*-Bernstein condition holds for*

   $$u(x) = \frac{c_2 x}{v(x)},$$

   where $c_2 = 6/\kappa(-2ba)$ *for* $b = \sup_x v(x) < \infty$*; and if* $v(0) = 0$ *we read* $0/v(0)$ *as* $\lim_{x \downarrow 0} x/v(x)$*.*

We are mainly interested in Part 1(a) of the theorem and its essential converse, Part 2. Part 1(b) is a by-product of the proof of 1(a) that will be useful for the proof of Proposition 5.11 below as well as the proof of the later-appearing Corollary 7.8. Part 2 assumes that the v-PPC condition holds for v such that $\sup_{x\geq 0} v(x) < \infty$. This boundedness requirement is without essential loss of generality, since we already assume that losses are in $[0, a]$. From the definition this trivially implies that, if the v-condition holds at all, then also the v'-condition holds for $v'(x) = v(x) \wedge a'$, for any $a' \geq a$.

Example 5.5 (Example 2.3 and 3.8, Continued) Let ℓ be a bounded loss function and suppose that the u-Bernstein condition holds with $u(x) = Bx^\beta$ for some $\beta \in [0, 1]$. We first note that if $\beta = 0$, then the condition holds trivially for large enough B. Theorem 5.4 shows that, in this case, we have the v-central condition for some v being linear in a neighborhood of 0, in particular $\liminf_{x\downarrow 0} v(x)/x < \infty$. Thus, for bounded losses, the v-central condition always holds for such v. Thus we will say that the v-central condition holds *nontrivially* if it holds for v with $\liminf_{x\downarrow 0} v(x)/x = \infty$. Since the trivial v-condition always holds, it provides no information and therefore, under this condition, one can only prove (using Hoeffding's inequality) the standard slow rate of $O(1/\sqrt{n})$. The other extreme is when we have the η-central condition, i.e. the v-condition holds with constant v, which as we show in Theorem 7.6 leads to rates of order $O(1/n)$. Moreover, as we show in Corollary 7.8, it also is possible to recover intermediate rates under the general case of the v-central condition. Specifically, under the v-central condition, we get in-probability rates of $O\left(w(1/n)\right)$, where we recall that w is the inverse of the function $x \mapsto xv(x)$. In the special case of $v : \varepsilon \mapsto \varepsilon^{1-\beta}$ (for which the behavior in terms of ε corresponds to the (β, B)-Bernstein condition as shown by Theorem 5.4), we get the rate $O(n^{-1/(2-\beta)})$, just as we do from the (β, B)-Bernstein condition. ■

The proof of Theorem 5.4 is deferred until Appendix A.3. It is based on the following lemma, which adds a (non-surprising) lower bound to a well-known upper bound used e.g. by Freedman (1975) in the context of concentration inequalities. Since most authors only require the upper bound, we have been unable to find a reference for the lower bound, except for Lemma C.4 in our own work (Koolen et al., 2014). Interestingly, the Lemma is applied in the proof of Theorem 5.4 with a 'frequentist' expectation over $Z \in \mathcal{Z}$ to prove the first part, and a 'Bayesian' expectation over $f \in \mathcal{F}$ to prove the second part.

Lemma 5.6 *For any random variable X taking values in $[-a, a]$,*

$$\kappa(-2a)\,\mathbf{Var}(X) \leq \mathbf{E}[X] + \log \mathbf{E}[e^{-X}] \leq \kappa(2a)\,\mathbf{Var}(X), \tag{40}$$

where the function κ is as defined above Theorem 5.4.

Proof Define the auxiliary function $\kappa'(x) = e^x - x - 1$. Then

$$\mathbf{E}[X] + \log \mathbf{E}[e^{-X}] = \min_{\mu\in[-a,a]} \mathbf{E}[\kappa'(\mu - X)],$$

as may be checked by observing that $\mathbf{E}[\kappa'(\mu - X)] = e^\mu\,\mathbf{E}[e^{-X}] - \mu + \mathbf{E}[X] - 1$ is minimized at $\mu = -\log \mathbf{E}[e^{-X}]$. Since $\kappa'(x) = \kappa(x)x^2$ and $\kappa(x)$ is increasing (Freedman, 1975), we

further have

$$\mathbf{E}[\kappa'(\mu - X)] \begin{cases} \leq \max_{\mu', x \in [-a,a]} \kappa(\mu' - x)\, \mathbf{E}[(\mu - X)^2] = \kappa(2a)\, \mathbf{E}[(\mu - X)^2] \\ \geq \min_{\mu', x \in [-a,a]} \kappa(\mu' - x)\, \mathbf{E}[(\mu - X)^2] = \kappa(-2a)\, \mathbf{E}[(\mu - X)^2], \end{cases} \tag{41}$$

from which the lemma follows upon observing that $\min_{\mu \in [-a,a]} \mathbf{E}[(\mu - X)^2] = \mathbf{Var}(X)$. ■

5.2 Bernstein vs. Central Condition for Unbounded Losses - Two-sided vs. One-sided Conditions

Applying Proposition 3.9 with $\eta = v(\varepsilon)$ for all $\varepsilon > 0$ immediately gives that, under no further assumptions, the v-central condition implies the v-pseudoprobability convexity condition. Combined with Theorem 5.4 this shows that the central condition and the Bernstein condition are essentially equivalent for bounded losses, so it is natural to ask how the v-versions of our conditions are related to the Bernstein conditions for unbounded losses. In that case there are two essential differences. One difference is that the variance or second moment in the Bernstein condition is *two-sided* in the sense that it is large both if the excess loss $\ell_f(Z) - \ell_{f^*}(Z)$ gets largely negative with significant probability, but also if the excess loss is large, whereas the central condition is *one-sided* in that large excess losses only make it easier to satisfy. This difference is illustrated by Example 5.7 below, where fast rates can be obtained and the central condition holds, but the Bernstein condition fails to be satisfied. The second difference is that the v-central condition essentially requires the probability that $\ell_{f^*}(Z) - \ell_f(Z)$ is large is *exponentially* small. Hence, if the loss is unbounded and has only polynomial tails, then the v-central condition cannot hold. Yet Example 5.8 shows that in such a case, the u-Bernstein condition can very well hold for nontrivial u. However, we should note that the v-PPC condition and the v-stochastic mixability conditions (introduced in the next subsection) also do not require exponential tails; hence it may still be that whenever the u-Bernstein condition holds, v-stochastic mixability also holds with $u(x) \cdot v(x) \asymp x$; we do not know whether this is the case.

Example 5.7 (Central without Bernstein for Unbounded Loss) Consider density estimation for the log loss. For f_μ the univariate normal density with mean μ and variance 1, let $\mathcal{P}$ be the normal location family and let $\mathcal{F} = \{f_\mu : \mu \in \mathbb{R}\}$ be the set of densities of the distributions in $\mathcal{P}$. Then, for any $P \in \mathcal{P}$ with density f_ν, the risk $R(P, f)$ is minimized by $f^* = f_\nu$, since the model is well-specified.

Let $Z_1, \ldots, Z_n$ be an iid sample from $P \in \mathcal{P}$. Then, as can be verified by direct calculation, the empirical risk minimizer/maximum likelihood estimator relative to $\mathcal{F}$, $\hat{\gamma}_n := \frac{1}{n} \sum_{j=1}^n Z_j$, satisfies $\mathbf{E}_{Z_1,\ldots,Z_n \sim P}(\hat{\gamma}_n - \nu)^2 = 1/n$, which translates into an expected excess risk of

$$\mathop{\mathbf{E}}_{Z_1,\ldots,Z_n, Z \sim P}[-\log f_{\hat{\gamma}_n}(Z) + \log f^*(Z)] = \frac{1}{2n},$$

such that ERM obtains a fast rate in expectation. One would therefore want a condition that aims to capture fast rates to be satisfied as well. For the central condition, this is the case with $\eta = 1$, as follows from Example 2.2. However, as we show next, the $(1, B)$-Bernstein condition does not hold for any constant B.

Consider $P \in \mathcal{P}$ with density f_ν, and abbreviate $U_\mu(z) = -\log f_\mu(z) + \log f_\nu(z) = \frac{\mu^2-\nu^2}{2} + z(\nu-\mu)$. Then

$$\begin{aligned}
\mathop{\mathbf{E}}_{Z\sim P}[U_\mu(Z)] &= \frac{\mu^2+\nu^2}{2} - \mu\nu \\
\mathop{\mathbf{E}}_{Z\sim P}[U_\mu^2(Z)] &= (\nu-\mu)^2 \mathop{\mathbf{E}}_{Z\sim P}[Z^2] + 2(\nu-\mu)\mathop{\mathbf{E}}_{Z\sim P}[Z]\frac{\mu^2-\nu^2}{2} + \left(\frac{\mu^2-\nu^2}{2}\right)^2 \\
&= (\nu-\mu)^2(1+\nu^2) + (\nu-\mu)\nu(\mu^2-\nu^2) + \left(\frac{\mu^2-\nu^2}{2}\right)^2.
\end{aligned}$$

First consider the case that the 'true' mean $\nu \geq 0$. Then for all constants B the $(1,B)$-Bernstein condition fails to hold. To see this, first observe that for any μ satisfying $\mu \leq 0$ and $-\mu \geq \nu$, we have $\mathbf{E}_{Z\sim P}[U_\mu^2(Z)] \geq \left(\frac{\mu^2-\nu^2}{2}\right)^2$ since $\nu - \mu \geq 0$ and $\nu \geq 0$. Second, observe that $\mathbf{E}_{Z\sim P}[U_\mu(Z)] \leq \mu^2+\nu^2$ since $-\mu\nu \leq \frac{\mu^2+\nu^2}{2}$. Hence, the following condition is weaker than the $(1,B)$-Bernstein condition:

$$(\mu^2-\nu^2)^2 \leq 4B(\mu^2+\nu^2).$$

Choosing μ to satisfy $\nu \leq \frac{\mu^2}{2}$ leads to the even weaker condition $\left(\frac{\mu^2}{2}\right)^2 \leq 4B(2\mu^2)$ which fails as soon as $|\mu| > \sqrt{32B}$. It remains to show that the $(1,B)$-Bernstein also fails to hold for all B if the true mean $\nu < 0$; this is shown using a symmetric argument by considering $\mu > 0$ and $-\mu < \nu$. The result follows. ■

Critically, the Bernstein condition cannot hold because of the two-sided nature of the second moment, which is large, not just if some f_μ is better than f^* with significant probability, but also if it is much worse. Thus, the fact that certain f_μ are so highly suboptimal that they suffer high empirical excess risk with high probability (and hence are easily avoided by ERM) ironically is what causes the Bernstein condition to fail; a related point is made by Mendelson (2014). The next example shows that, if Z has two-sided, polynomial tails then the opposite phenomenon can also occur: the v-central condition does not hold for any v, but we do have the u-Bernstein condition for constant u.

Example 5.8 Let $\mathcal{P}$ be an arbitrary collection of distributions over $\mathbb{R}$ such that for all $P \in \mathcal{P}$, the mean $\mu_P := \mathbf{E}_{Z\sim P}[Z] \in [-1,1]$. Consider the squared loss $\ell_f^{\mathrm{sq}}(z) = \frac{1}{2}(z-f)^2$, with $\mathcal{F} = [-1,1]$. Assume that $\mathcal{P}$ contains a distribution P^* with $\mu_{P^*} = 0$ and, for some constants $c_1, c_2 > 0$, for all $z \in \mathbb{R}$ with $|z| > c_1$, the density p^* of P^* satisfies $p^*(z) \geq c_2/z^6$. The predictor in $\mathcal{F}$ that minimizes risk is given by $f^* = 0$. Now with such a $\mathcal{P}$, for all $\eta > 0$, all $\mu \neq 0$, and using that $\ell_{f^*}^{\mathrm{sq}} - \ell_\mu^{\mathrm{sq}} = 2Z\mu - \mu^2$, we find for $c_3 = c_2 \cdot \exp(-\eta\mu^2)$,

$$\mathop{\mathbf{E}}_{Z\sim P}\left[e^{\eta\left(\ell_{f^*}^{\mathrm{sq}}(Z) - \ell_\mu^{\mathrm{sq}}(Z)\right)}\right] \geq \int_{c_1}^{\infty} \frac{c_3}{z^6} e^{\eta 2z|\mu|} \mathrm{d}z = \infty, \tag{42}$$

so that the v-central condition fails for all v of the form required in Definition 5.3. Hence the v-central condition does not hold — although from Example 5.10 below we see that v-stochastic mixability (and hence the v-PPC condition) does hold for $v(x) \asymp \sqrt{x}$.

Now consider a $\mathcal{P}$ with means in $[-1,1]$ and containing a P^* as above such that additionally for all $P \in \mathcal{P}$, the fourth moment is uniformly bounded, i.e. there is an $A > 0$ such that for all $P \in \mathcal{P}$, $\mathbf{E}_{Z\sim P}[Z^4] < A$. Clearly we can construct such a $\mathcal{P}$ and by the above it will not satisfy the v-central condition for any allowed v. However, the u-Bernstein condition holds with $u(x) = (4A^{1/2}+1)x$, since, using again $\ell_\mu^{\mathrm{sq}}(Z) - \ell_{f^*}^{\mathrm{sq}}(Z) = -2Z\mu + \mu^2$, we find

$$\mathop{\mathbf{E}}_{Z\sim P^*}\left(\ell_\mu^{\mathrm{sq}}(Z) - \ell_{f^*}^{\mathrm{sq}}(Z)\right)^2 = \mathbf{E}\left[4Z^2\mu^2 + \mu^4 - 4Z\mu^3\right] \leq 4\sqrt{A}\mu^2 + \mu^4 \leq u(\mu^2) = u\left(\mathop{\mathbf{E}}_{Z\sim P^*}\left(\ell_\mu^{\mathrm{sq}}(Z) - \ell_{f^*}^{\mathrm{sq}}(Z)\right)\right).$$

■

5.3 v-Stochastic Mixability and the JRT Conditions

Just as Definition 5.3 weakened the η-central and PPC conditions to the v-central and PPC conditions, we similarly may weaken the main conditions of Section 4, stochastic mixability and its special case stochastic exp-concavity, to their v-versions:

Definition 5.9 (v-Stochastic Mixability and v-Stochastic Exp-Concavity) *Let $v\colon [0,\infty) \to [0,\infty)$ be a bounded, non-decreasing function satisfying $v(x) > 0$ for all $x > 0$. We say that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is* v-stochastically mixable *if, for all $\varepsilon \geq 0$, there exists a function $\phi : \mathcal{P} \to \mathcal{F}_{\mathrm{d}}$ such that (22) is satisfied with $\eta = v(\varepsilon)$. If $\mathcal{F}_{\mathrm{d}} \supseteq \mathrm{co}(\mathcal{F})$ and this holds for the function $\psi(\Pi) = \mathbf{E}_{f\sim\Pi}[f]$ for all $\varepsilon > 0$, then we say that $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is* v-stochastically-exp-concave.

The main insight of Sections 3 and 4 was that the η-central condition, η-PPC condition and η-stochastic mixability are all equivalent under some assumptions. One may of course conjecture that the same holds for their weaker v-versions. We shall defer discussion of this issue to Section 8 and for now focus on the usefulness of v-stochastic exp-concavity, which can lead to intermediate rates even for unbounded losses.

A special case of v-stochastic exp-concavity, which we will call the JRT-I condition, was stated by Juditsky et al. (2008); recall that we discussed the JRT-II condition in Section 4.2.3. The JRT-I condition[8] states that, for every $\eta > 0$, the excess loss can be decomposed as

$$\ell_f(z) - \ell_{f^*}(z) \geq \ell_\eta^{(2)}(z, f, f^*) - r_\eta(z) \qquad \text{for all } z, \text{ any } f, f^* \in \mathrm{co}(\mathcal{F}),$$

where $r_\eta\colon \mathcal{Z} \to \mathbb{R}$ does not depend on f, f^*, and, for any $f^* \in \mathrm{co}(\mathcal{F})$, $\ell_\eta^{(2)}(z, f^*, f^*) = 0$ and $\ell_\eta^{(2)}(z, f, f^*)$ is 1-exponentially concave as a function of $f \in \mathrm{co}(\mathcal{F})$ (*i.e.*, (25) holds with $\eta\ell_f(z) = \ell_\eta^{(2)}(z, f, f^*)$). Note that the choice of $\ell_\eta^{(2)}$ and r_η in general depends on η. Juditsky et al. (2008) show that, under this condition, fast rates can be obtained in, for

8. The assumption is stated in basic form in their Theorem 4.1; their Q_2 is our $\ell^{(2)}$ and their R is our r_η; the dependence of r_η on η (their $1/\beta$) is made explicit in their Corollary 5.1.

example, regression problems with a finite number of regression functions, where the rate depends on how $\varepsilon_\eta := \sup_{P\in\mathcal{P}} \mathbf{E}_{Z\sim P}\left[r_\eta(Z)\right]$ varies with η.

We now connect the JRT-I assumption to v-stochastic exp-concavity. Consider again the substitution function $\psi(\Pi) := \mathbf{E}_{g\sim\Pi}[g]$ as in Definition 4.7. Letting $\bar{g} = \psi(\Pi)$, the JRT-I assumption implies that

$$\begin{aligned}
&\mathop{\mathbf{E}}_{Z\sim P}\left[\ell_{\mathbf{E}_{g\sim\Pi}[g]}(Z) + \frac{1}{\eta}\log \mathop{\mathbf{E}}_{g\sim\Pi} e^{-\eta\ell_g(Z)}\right] = \mathop{\mathbf{E}}_{Z\sim P}\left[\frac{1}{\eta}\log \mathop{\mathbf{E}}_{g\sim\Pi} e^{\eta\ell_{\bar{g}}(Z)-\eta\ell_g(Z)}\right] \\
&\quad\le \mathop{\mathbf{E}}_{Z\sim P}\left[\frac{1}{\eta}\log \mathop{\mathbf{E}}_{g\sim\Pi} e^{-\eta\ell^{(2)}(Z,g,\bar{g})+\eta r_\eta(Z)}\right] \\
&\quad\overset{(a)}{\le} \mathop{\mathbf{E}}_{Z\sim P}\left[\frac{1}{\eta}\log e^{-\eta\ell^{(2)}(Z,\bar{g},\bar{g})+\eta r_\eta(Z)}\right] = \mathop{\mathbf{E}}_{Z\sim P}\left[r_\eta(Z)\right] \le \varepsilon_\eta,
\end{aligned}$$

where (a) follows by the η-exp-concavity of $\ell^{(2)}$. The derivation shows that, if the JRT-I condition holds for each η with function $r_\eta(z)$ then we have η-stochastic exp-concavity up to $\varepsilon_\eta := \sup_{P\in\mathcal{P}} \mathbf{E}_{Z\sim P}[r_\eta(Z)]$. In their Theorem 4.1 they go on to show that, for finite $\mathcal{F}$, by applying the aggregating algorithm at learning rate η and an on-line to batch conversion, one can obtain rates of order $O(\log|\mathcal{F}|/(n\eta) + \varepsilon_\eta)$, for each η. They go on to calculate ε_η as function of η in various examples (regression, classification with surrogate loss functions, density estimation) and, in each example, optimize η as a function of n so as to minimize the rate. Now for each function ε_η in their examples, there is a corresponding inverse function v that maps ε to η rather than vice versa, so that if the JRT-I condition holds for ε_η, then v-stochastic exp-concavity holds. Rather than formalizing this in general, we illustrate it informally using their regression example (Juditsky et al., 2008, Section 5.1):

Example 5.10 (JRT-I Condition and Regression) JRT consider a regression problem in which $\mathcal{F}$ is finite and $\sup_{P\in\mathcal{P}} \|f\|_{P,\infty} < \infty$ for all $f \in \mathcal{F}$, where $\|\cdot\|_{P,\infty}$ denotes the $L_\infty(P_X)$-norm. They further assume that a weak moment assumption holds: for all $P \in \mathcal{P}$, $\mathbf{E}_{(X,Y)\sim P}[|Y|^s] < \infty$ for some $s \ge 2$. They show that in this setting there exist constants $c_1, c_2, c_3, c_4 > 0$ such that for all $y \in \mathbb{R}$, $r_\eta(y) \le c_1|y| \cdot [\![|y| > c_2/\eta]\!] + \eta c_3 y^2 \cdot [\![|y| \ge c_4/\sqrt{\eta}]\!]$. Bounding expectations of the form $|y|^a \cdot [\![|y| > b]\!]$ in the same way as one bounds expectations of indicator variables $[\![|y| > b]\!]$ in the proof of Markov's inequality, this gives that

$$\varepsilon_\eta = O\left(\eta^{s/2}\right),$$

which is strictly increasing in η. Thus, the inverse $v(\varepsilon)$ of ε_η is well-defined on $\varepsilon > 0$ and satisfies $v(\varepsilon) = O(\varepsilon^{2/s})$. Since the JRT-I condition implies that, for all $\eta > 0$, we have η-stochastic exp-concavity up to ε if $\varepsilon \ge \varepsilon_\eta$, it follows that for all $\varepsilon > 0$, we must have η-stochastic exp-concavity up to ε for $\eta \le v(\varepsilon)$. It follows that v-stochastic exp-concavity holds with $v(\varepsilon) = O(\varepsilon^{2/s})$. In this unbounded loss case, we can easily obtain a rate by using the aggregating algorithm with online-to-batch conversion. Applying Proposition 4.5 with the optimal choice of ε yields a rate of $2\left(\frac{\log|\mathcal{F}|}{n}\right)^{-s/(s+2)}$, which coincides with the rate obtained by Juditsky, Rigollet, and Tsybakov (2008) in their Corollary 5.2 and the minimax rate for this problem (Audibert, 2009). ■

5.4 The v-Central Condition and Existence of Unique Risk-Minimizers

Corollary 3.11 showed that, under Assumption A, strong η-fast rate (i.e. central and PPC) conditions imply uniqueness of optimal f^*'s. Here we extend this result, for bounded loss, to the v-fast rate conditions, and also provide a converse, thus completely characterizing uniqueness of f^* in terms of the v-central condition, for bounded losses. To understand the proposition, note that for two predictors with the same risk, $R(P,f) = R(P,f^*)$, it holds that f and f^* achieve the same loss almost surely, so they essentially coincide, if and only if $\mathbf{Var}_{Z\sim P}[\ell_f(Z) - \ell_{f^*}(Z)] = 0$. In the proposition we use $\mathcal{F}_\varepsilon = \{f^*\} \cup \{f \in \mathcal{F} : \mathbf{Var}_{Z\sim P}[\ell_f(Z) - \ell_{f^*}(Z)] \geq \varepsilon\}$ to denote the subset of $\mathcal{F}$ where all f's that are very similar to, but not identical with, f^* have been taken out.

Proposition 5.11 **(v-central condition and (non-)uniqueness of risk minimizers)** *Fix $(\ell, \{P\}, \mathcal{F})$ such that the loss ℓ is bounded and Assumption A holds, and let f^* be an $\mathcal{F}$-risk minimizer for P. Exactly one of the following two situations is the case:*

1. *The v-central condition holds for some v that is sublinear at 0, i.e. $\lim_{x\downarrow 0} v(x)/x = \infty$. In this case, f^* is essentially unique, in the sense that for every sequence $f_1, f_2, \ldots \in \mathcal{F}$ such that $\mathbf{E}_{Z\sim P}[\ell_{f_j}(Z)] \to \mathbf{E}_{Z\sim P}[\ell_{f^*}(Z)]$, we have $\mathbf{Var}_{Z\sim P}\left[\ell_{f_j}(Z) - \ell_{f^*}(Z)\right] \to 0$. Moreover, for every $\varepsilon > 0$, $(\ell, \{P\}, \mathcal{F}_\varepsilon)$ satisfies the η-central condition for some $\eta > 0$.*

2. *The v-central condition only holds trivially in the sense of Example 5.5, i.e. it does not hold for any v with $\lim_{x\downarrow 0} v(x)/x = \infty$. In this case, f^* is essentially* non-*unique, in the sense that there exists $\varepsilon > 0$ and a sequence $f_1, f_2, \ldots \in \mathcal{F}$ (possibly identical for all large j) such that $\mathbf{E}_{Z\sim P}[\ell_{f_j}(Z)] \to \mathbf{E}_{Z\sim P}[\ell_{f^*}(Z)]$, but, for all sufficiently large j, $\mathbf{Var}_{Z\sim P}\left[\ell_{f_j}(Z) - \ell_{f^*}(Z)\right] \geq \varepsilon$. Moreover, for some $\varepsilon > 0$, $(\ell, \{P\}, \mathcal{F}_\varepsilon)$ does not satisfy the η-central condition for any $\eta > 0$.*

Proof For Part 1, Proposition 3.9 implies that the v-PPC condition holds. Now Part 2 of Theorem 5.4 implies that the u-Bernstein condition holds with u such that $\lim_{x\downarrow 0} u(x) = \lim_{x\downarrow 0} x/v(x) = 0$ by assumption. Then it follows from the definition of the u-Bernstein condition that f^* is essentially unique. Moreover, by Part 1(b) of Theorem 5.4, there exists a function v' with $v'(x) > 0$ for $x > 0$, such that for every $\delta > 0$, $(\ell, \{P\}, \{f^*\} \cup \mathcal{G})$ satisfies the η-central condition with $\eta = v'(\delta) > 0$ for any subset $\mathcal{G} \subseteq \{f \in \mathcal{F} : R(P,f) - R(P,f^*) \geq \delta\}$. Now since the u-Bernstein condition holds with $\lim_{x\downarrow 0} u(x) = 0$, we know that, for every $\varepsilon > 0$, there is a $\delta > 0$ such that $\mathbf{Var}_{Z\sim P}[\ell_f(Z) - \ell_{f^*}(Z)] \geq \varepsilon$ implies $R(P,f) - R(P,f^*) > \delta$. For this δ, $\mathcal{G} = \{f \in \mathcal{F} : \mathbf{Var}_{Z\sim P}[\ell_f(Z) - \ell_{f^*}(Z)] \geq \varepsilon\}$ is a subset of $\{f \in \mathcal{F} : R(P,f) - R(P,f^*) \geq \delta\}$, and consequently, as already established, $(\ell, \{P\}, \{f^*\} \cup \mathcal{G})$ must satisfy the η-central condition for $\eta > 0$, which is what we had to prove.

For Part 2, to show non-uniqueness of f^*, note that by Theorem 5.4, Part 1, the u-Bernstein condition cannot hold for any u with $\lim_{x\downarrow 0} u(x) = 0$. This already shows that there exists a sequence as required, for some $\varepsilon > 0$, so that f^* is essentially non-unique. Since $\mathbf{Var}_{Z\sim P}[\ell_{f_j}(Z) - \ell_{f^*}(Z)] \geq \varepsilon$ for all elements of the sequence and $R(P,f_j) \to R(P)$, the first inequality of Lemma 5.6 applied with $X = \eta(\ell_{f_j}(Z) - \ell_{f^*}(Z))$ now gives that, for all $\eta > 0$, there exists f_j such that $\log \mathbf{E}_{Z\sim P}\, e^{\eta(\ell_{f^*}(Z) - \ell_{f_j}(Z))} > 0$, so that the η-central condition does not hold. ■

6. From Fast Rates for Actions to Fast Rates for Functions

Let $\ell\colon \mathcal{A}\times\mathcal{Y}\to\mathbb{R}$ be a loss function, where $\mathcal{Y}$ is a set of possible outcomes and $\mathcal{A}$ is a set of possible *actions*. Then our abstract formulation in terms of $(\ell,\mathcal{P},\mathcal{F},\mathcal{F}_{\mathrm{d}})$ can accommodate *unconditional* problems, where distributions $P\in\mathcal{P}$ are on $\mathcal{Z}=\mathcal{Y}$ and both $\mathcal{F}$ and $\mathcal{F}_{\mathrm{d}}$ are subsets of $\mathcal{A}$; but it can also capture the *conditional* setting, where we observe additional *features* from a covariate space $\mathcal{X}$. In that case, outcomes are pairs (X,Y) from $\mathcal{Z}'=\mathcal{X}\times\mathcal{Y}$, the model $\mathcal{F}'$ and decision set $\mathcal{F}'_{\mathrm{d}}$ are both sets of functions $\{f\colon\mathcal{X}\to\mathcal{F}\}$ from features to actions, and the loss is commonly defined in terms of the unconditional loss as $\ell'\big(f,(x,y)\big)=\ell(f(x),y)$.

It may often be easier to establish properties like stochastic mixability for the unconditional setting than for the conditional setting. In this section we therefore consider when we can lift conditions for unconditional problems with loss ℓ to the conditional setting with loss ℓ'. For the condition of being η-stochastically mixable, this is done by Proposition 6.1 below. And, in Example 6.2, it will be seen that, in some cases, this also allows us to obtain the η-central condition for the conditional setting.

Proposition 6.1 is based on the construction of a substitution function $\psi'\colon\Delta(\mathcal{F}')\to\mathcal{F}'_{\mathrm{d}}$ for the conditional setting from the substitution function $\psi\colon\Delta(\mathcal{F})\to\mathcal{F}_{\mathrm{d}}$ for the unconditional setting. This works by applying ψ conditionally on every $x\in\mathcal{X}$: first, note that any distribution Π on functions $f\in\mathcal{F}'$, induces, for every $x\in\mathcal{X}$, a distribution Π_x on actions $\mathcal{A}$ by drawing $f\sim\Pi$ and then evaluating $f(x)$. We may therefore define $\psi'(\Pi)=f_\Pi$ with f_Π the function

$$f_\Pi(x)=\psi(\Pi_x). \tag{43}$$

The conditions of the proposition then amount to the requirement that this is a valid substitution function in the conditional setting.

Proposition 6.1 *Let $(\ell,\mathcal{P},\mathcal{F},\mathcal{F}_{\mathrm{d}})$ and $(\ell',\mathcal{P}',\mathcal{F}',\mathcal{F}'_{\mathrm{d}})$ correspond to the unconditional and conditional settings described above, and assume all of the following:*

- *$(\ell,\mathcal{P},\mathcal{F},\mathcal{F}_{\mathrm{d}})$ satisfies η-stochastic mixability up to ε with substitution function ψ;*
- *$P(Y|X)\in\mathcal{P}$ for every $P\in\mathcal{P}'$;*
- *the function f_Π from (43) is measurable and contained in $\mathcal{F}'_{\mathrm{d}}$, for every $\Pi\in\Delta(\mathcal{F}')$.*

Then η-stochastic mixability up to ε is satisfied in the conditional setting. In particular, f_Π is contained in $\mathcal{F}'_{\mathrm{d}}$ if:

- *$\mathcal{F}'_{\mathrm{d}}$ is the set of* all *measurable functions from $\mathcal{X}$ to $\mathcal{A}$; or*
- *$(\ell,\mathcal{P},\mathcal{F},\mathcal{F}_{\mathrm{d}})$ is η-stochastically exp-concave up to ε, and $\mathcal{F}'_{\mathrm{d}}$ contains the convex hull of $\mathcal{F}'$. In this case, $(\ell',\mathcal{P}',\mathcal{F}',\mathcal{F}'_{\mathrm{d}})$ is also η-stochastically exp-concave up to ε.*

We recall from Section 4.2.2 that η-stochastic exp-concavity is the special case of η-stochastic mixability where the substitution function maps Π to its mean. In addition, for η-stochastic exp-concavity the weak and strong versions of the condition coincide.

Proof We verify η-stochastic mixability up to ε for $(\ell', \mathcal{P}', \mathcal{F}', \mathcal{F}'_{\mathrm{d}})$ by using η-stochastic mixability for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ conditional on each $x \in \mathcal{X}$: for any $P \in \mathcal{P}'$ and $\Pi \in \Delta(\mathcal{F}')$,

$$\begin{aligned}
\mathop{\mathbf{E}}_{P(X,Y)}\left[\ell'_{\psi'(\Pi)}(X,Y)\right] &= \mathop{\mathbf{E}}_{P(X)}\mathop{\mathbf{E}}_{P(Y|X)}\left[\ell_{\psi(\Pi_X)}(Y)\right] \\
&\leq \mathop{\mathbf{E}}_{P(X)}\mathop{\mathbf{E}}_{P(Y|X)}\left[-\tfrac{1}{\eta}\log \mathop{\mathbf{E}}_{\Pi_X(A)}\left[e^{-\eta\ell_A(Y)}\right]\right] + \varepsilon \\
&= \mathop{\mathbf{E}}_{P(X,Y)}\left[-\tfrac{1}{\eta}\log \mathop{\mathbf{E}}_{\Pi(f)}\left[e^{-\eta\ell'_f(X,Y)}\right]\right] + \varepsilon,
\end{aligned}$$

which was to be shown.

Verifying that $f_\Pi \in \mathcal{F}'_{\mathrm{d}}$ is trivial if $\mathcal{F}'_{\mathrm{d}}$ is the set of all measurable functions. And if $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$ is η-stochastically exp-concave up to ε, then $f_\Pi(x) = \mathbf{E}_\Pi[f(x)]$ for all x, and therefore f_Π is the mean of Π also in the conditional setting. ■

The most important application is when $\mathcal{P}$ contains all possible distributions on $\mathcal{Y}$, which means that the unconditional problem is classically mixable in the sense of Vovk (see Section 4.2.1). Then the requirement that $P(Y \mid X) \in \mathcal{P}$ is automatically satisfied.

Example 6.2 (Squared Loss for Misspecified Model) As discussed in Example 4.6, the squared loss is η-exp-concave in the unconditional setting on a bounded domain $\mathcal{F}_{\mathrm{d}} \supseteq \mathcal{F} = \mathcal{Z} = [-B, B]$, for $\eta = 1/4B^2$. If we make the setting conditional by adding features, and consider any set of regression functions $\mathcal{F}'$ and any set of joint distributions $\mathcal{P}'$, then Proposition 6.1 implies that we still have exp-concavity as long as we allow ourselves to make decisions in the convex hull of $\mathcal{F}'$, i.e. if $\mathcal{F}'_{\mathrm{d}} \supseteq \mathrm{co}(\mathcal{F}')$. Note that this holds even if the model $\mathcal{F}$ is misspecified in that it does not contain the true regression function $x \mapsto \mathbf{E}[Y \mid X = x]$. If, furthermore, the model $\mathcal{F}'$ is itself convex and satisfies Assumption A relative to $\mathcal{P}'$, i.e. the minimum risk $\min_{f \in \mathcal{F}'} \mathbf{E}_{(X,Y)\sim P}(Y - f(X))^2$ is achieved for all $P \in \mathcal{P}'$, then we may take $\mathcal{F}'_{\mathrm{d}} = \mathcal{F}'$ and recover the setting considered by Lee et al. (1998). Even though this does not require $\mathcal{F}'$ to be well-specified, the strong version of Assumption B (which implies Assumption A) is then still satisfied, and hence Proposition 4.12 and Theorem 3.10 tell us that $(\ell', \mathcal{P}, \mathcal{F})$ satisfies both the strong η-pseudoprobability convexity condition and the strong η-central condition. ■

The example raises the question whether we cannot directly conclude, under appropriate conditions, that, if the η-central condition holds for some unconditional $(\ell, \mathcal{P}, \mathcal{F})$, then it should also hold for the corresponding conditional $(\ell', \mathcal{P}', \mathcal{F}')$. We can indeed prove a trivial analogue of Proposition 6.1 for this case, as long as $\mathcal{F}'$ contains *all* measurable functions from $\mathcal{X}$ to $\mathcal{Y}$; we implicitly used this result in Example 3.7. Example 6.2, however, shows that, if one can first establish η-stochastic exp-concavity for $(\ell, \mathcal{P}, \mathcal{F}, \mathcal{F}_{\mathrm{d}})$, one can sometimes reach the stronger conclusion that $(\ell', \mathcal{P}', \mathcal{F}')$ satisfies the η-central condition as long as $\mathcal{F}'$ is merely convex, rather than the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$.

7. The Central Condition Implies Fast Rates

In this section, we show how a statistical learning problem's satisfaction of the strong η-central condition implies fast rates of $O(1/n)$ under a bounded losses assumption. Theorem 7.6 herein establishes via a rather direct argument that the strong η-central condition implies an exact oracle inequality (i.e. with leading constant 1) with a fast rate for finite function classes, and Theorem 7.7 extends this result to VC-type classes. We emphasize that the implication of fast rates from the strong η-central condition under a bounded losses assumption is not itself new. Specifically, for bounded losses, the central condition is essentially equivalent to the Bernstein condition by Theorem 5.4, and therefore implies fast rates via existing fast rate results for the Bernstein condition. For instance, for finite classes Theorem 4.2 of Zhang (2006b) implies a fast $O(1/n)$ rate by letting ℓ_θ be our excess loss $\ell_f - \ell_{f^*}$ assumed to satisfy the bounded loss condition therein, setting $\alpha = 0$, taking Π to be the uniform prior over a finite class $\mathcal{F}$, and taking ρ as $\frac{C}{KM}$ for some sufficiently small constant C. In addition, Audibert (2004) showed fast rates for classification under the Bernstein condition[9]; see for example Theorem 3.4 of Audibert (2004) along with the discussion of how the variant of the (CA3) condition needed there is related to the (CA1) condition connected to VC-classes. However, since we posit the one-sided central condition rather than the two-sided Bernstein condition as our main condition, it is interesting to take a direct route based on the central condition itself, rather than proceeding via the Bernstein condition. As an added benefit, this approach turns out to give better constants and a better dependence on the upper bound on the loss.

We proceed via the standard Cramér-Chernoff method, which also lies at the heart of many standard (and advanced) concentration inequalities (Boucheron et al., 2013). This method requires an upper bound on the cumulant generating function. We solve this subproblem by solving an optimization problem that is an instance of the general moment problem, a problem on which Kemperman (1968) has conducted a detailed geometric study. This strategy leads to a fast rates bound for finite classes, which can be extended to parametric (VC-type) classes, as shown in Section 7.3.

7.1 The Strong Central Condition and ERM

For the remainder of Section 7, we will consider the conditional setting, where the loss $\ell_f(Z)$ takes values in the bounded range $[0, V]$ for outcomes $Z = (X, Y) \in \mathcal{X} \times \mathcal{Y}$ and functions f from $\mathcal{F} = \{f \colon \mathcal{X} \to \mathcal{A}\}$. We take $\mathcal{P} = \{P\}$ to be a single fixed distribution and we will assume throughout that $(\ell, \{P\}, \mathcal{F})$ satisfies the strong η-central condition for some $\eta > 0$. That is, there exists $f^* \in \mathcal{F}$ such that

$$\log \mathop{\mathbf{E}}_{Z \sim P} \exp(-\eta W_f) \leq 0 \qquad \text{for all } f \in \mathcal{F}, \tag{44}$$

where we have abbreviated the excess loss by $W_f(Z) = \ell_f(Z) - \ell_{f^*}(Z)$; for brevity we further abbreviate $W_f(Z)$ to W_f in this section. Then, by Jensen's inequality, f^* is $\mathcal{F}$-optimal for P. We let η^* denote the largest η for which (44) holds.

9. Audibert actually introduces multiple conditions, referred to as variants of the margin condition, but these actually are closer to Bernstein-type conditions as they take into account the function class $\mathcal{F}$.

An empirical measure P_n associated with an n-sample $\mathbf{Z}$, comprising n *independent, identically distributed* (iid) observations $(Z_1, \ldots, Z_n) = ((X_1, Y_1), \ldots, (X_n, Y_n))$, operates on functions as $P_n f = \frac{1}{n}\sum_{j=1}^n f(X_j)$ and on losses as $P_n \ell_f = \frac{1}{n}\sum_{j=1}^n \ell_f(Z_j)$.

Cramér-Chernoff. We will bound the probability that the ERM estimator

$$\hat{f}_{\mathbf{Z}} := \operatorname*{arg\,min}_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \ell_f(Z_i) \tag{45}$$

selects a hypothesis with excess risk $R(P, f) - R(P, f^*) = \mathbf{E}[W_f]$ above $\frac{a}{n}$ for some constant $a > 0$. For any real-valued random variable X, let $\eta \mapsto \Lambda_X(\eta) = \log \mathbf{E}\, e^{\eta X}$ denote its *cumulant generating function* (CGF), which is known to be convex and satisfies $\Lambda'(0) = \mathbf{E}[X]$.

Lemma 7.1 (Cramér-Chernoff) *For any $f \in \mathcal{F}$, $\eta > 0$ and $t \in \mathbb{R}$,*

$$\mathbf{Pr}\left(\frac{1}{n}\sum_{j=1}^n \ell_f(Z_j) \le \frac{1}{n}\sum_{j=1}^n \ell_{f^*}(Z_j) + t\right) \le \exp\left(\eta n t + n\Lambda_{-W_f}(\eta)\right). \tag{46}$$

Proof Applying Markov's inequality to $e^{-\eta n P_n W_f}$ and using the fact that $\Lambda_{-n P_n W_f}(\eta) = n\Lambda_{-W_f}(\eta)$ for iid observations, yields

$$\mathbf{Pr}\left(-P_n W_f > -t\right) \le \exp\left(\eta n t + \Lambda_{-n P_n W_f}(\eta)\right) = \exp\left(\eta n t + n\Lambda_{-W_f}(\eta)\right),$$

from which the lemma follows. ■

7.2 Semi-infinite Linear Programming and the General Moment Problem

We first consider the canonical case that W_f takes values in $[-1, 1]$ (*i.e.*, $V = 1$), that $\Lambda_{-W_f}(\eta^*) = 0$ with equality (as opposed to the inequality in Equation 44) and that $\mathbf{E}[W_f] = a/n$ for some constant $a > 0$ that does not depend on f. These restrictions allow us to formulate the goal of bounding the CGF as an instance of the general moment problem of Kemperman (1968, 1987). We will later relax them to allow general V, $\Lambda_{-W_f}(\eta^*) \le 0$ and $\mathbf{E}[W_f] \ge a/n$.

As illustrated by Figure 3, our approach will be to bound $\Lambda_{-W_f}(\eta)$ at $\eta = \eta^*/2$ from above by maximizing over all possible random variables W_f subject to the given constraints. This is equivalent to minimizing $-\mathbf{E}[\exp((\eta^*/2)S)]$ over $S = -W_f$ and may be formulated as an instance of the general moment problem, which we describe next.

The general moment problem. Let $\Delta(\mathcal{S})$ be the set of all probability measures over a measurable space $\mathcal{S}$. Then for any real-valued measurable functions h, $g_1, \ldots, g_m$ on $\mathcal{S}$ and constants $k_1, \ldots, k_m$, the general moment problem is the semi-infinite linear program

$$\begin{aligned} \inf_{P \in \Delta(\mathcal{S})} \quad & \mathop{\mathbf{E}}_{S \sim P} h(S) \\ \text{subject to} \quad & \mathop{\mathbf{E}}_{S \sim P} g_j(S) = k_j, \quad j = 1, \ldots, m. \end{aligned} \tag{47}$$

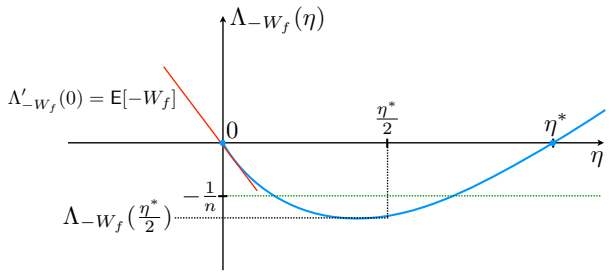


Figure 3: Control of the CGF of $-W_f$ for a function f with excess loss $\mathbf{E}[W_f]$ of order $\frac{1}{n}$. The derivative at 0 equals $-\mathbf{E}[W_f]$.

Define the vector-valued map $g\colon \mathcal{S} \to \mathbb{R}^m$ as $g(s) = (g_1(s), \ldots, g_m(s))$ and the vector $k = (k_1, \ldots, k_m)$. Then Theorem 3 of Kemperman (1968), which was also shown independently by Richter (1957) and Karlin and Studden (1966), states that, if $k \in \operatorname{int}\operatorname{co}(g(\mathcal{S}))$, the optimal value of problem (47) equals

$$\sup\left\{ d_0 + \sum_{j=1}^{m} d_j k_j : d^* = (d_0, d_1, \ldots, d_m) \in D^* \right\}, \tag{48}$$

where $D^* \subseteq \mathbb{R}^{m+1}$ is the set

$$D^* := \left\{ d^* = (d_0, d_1, \ldots, d_m) \in \mathbb{R}^{m+1} : h(s) \geq d_0 + \sum_{j=1}^{m} d_j g_j(s) \text{ for all } s \in \mathcal{S} \right\}. \tag{49}$$

Instantiating, we choose $\mathcal{S} = [-1, 1]$ and define

$$h(s) = -e^{(\eta^*/2)s}, \qquad g_1(s) = s, \qquad g_2(s) = e^{\eta^* s}, \qquad k_1 = -\frac{a}{n}, \qquad k_2 = 1,$$

which yields the following special case of problem (47):

$$\inf_{P \in \Delta([-1,1])} \quad -\mathop{\mathbf{E}}_{S \sim P} e^{(\eta^*/2)S} \tag{50a}$$

$$\text{subject to} \quad \mathop{\mathbf{E}}_{S \sim P} S = -\frac{a}{n} \tag{50b}$$

$$\mathop{\mathbf{E}}_{S \sim P} e^{\eta^* S} = 1. \tag{50c}$$

Equation 48 from the general moment problem now instantiates to

$$\sup\left\{ d_0 - \frac{a}{n} d_1 + d_2 : d^* = (d_0, d_1, d_2) \in D^* \right\}, \tag{51}$$

with D^* equal to the set

$$\left\{ d^* = (d_0, d_1, d_2) \in \mathbb{R}^3 : -e^{(\eta^*/2)s} \geq d_0 + d_1 x + d_2 e^{\eta^* s} \text{ for all } s \in [-1, 1] \right\}. \tag{52}$$

Applying Theorem 3 of Kemperman (1968) requires $k \in \operatorname{int}\operatorname{co} g([-1, 1])$. We first characterize when $k \in \operatorname{co} g([-1, 1])$ holds and handle the $\operatorname{int}\operatorname{co} g([-1, 1])$ version after Theorem 7.3. The proof of the next result, along with all subsequent results in this section, can be found in Appendix A.4.

Lemma 7.2 *For $a > 0$, the point $k = \left(-\frac{a}{n}, 1\right) \in \operatorname{co}(g([-1,1]))$ if and only if*

$$\frac{a}{n} \leq \frac{e^{\eta^*} + e^{-\eta^*} - 2}{e^{\eta^*} - e^{-\eta^*}} = \frac{\cosh(\eta^*) - 1}{\sinh(\eta^*)}. \tag{53}$$

Moreover, $k \in \operatorname{int} \operatorname{co}(g([-1,1]))$ if and only if the inequality in (53) is strict.

Note that (53) is guaranteed to hold, because otherwise the semi-infinite linear program (50) is infeasible (which in turn implies that such an excess loss random variable cannot exist).

The next theorem is a key result for using the strong central condition to control the CGF.

Theorem 7.3 *Let f be an element of $\mathcal{F}$ with $(\ell_f - \ell_{f^*})(Z)$ taking values in $[-1,1]$, $n \in \mathbb{N}$, $\mathbf{E}_{Z\sim P}(\ell_f - \ell_{f^*})(Z) = \frac{a}{n}$ for some $a > 0$, and $\Lambda_{-(\ell_f - \ell_{f^*})(Z)}(\eta^*) = 0$ for some $\eta^* > 0$. If*

$$\frac{a}{n} < \frac{\cosh(\eta^*) - 1}{\sinh(\eta^*)}, \tag{54}$$

then

$$\Lambda_{-(\ell_f - \ell_{f^*})(Z)}(\eta^*/2) \leq \frac{-0.21(\eta^* \wedge 1)a}{n}.$$

Corollary 7.4 *The result of Theorem 7.3 also holds when the strict inequality in (54) is replaced with inequality, i.e. $\frac{a}{n} \leq \frac{\cosh(\eta^*)-1}{\sinh(\eta^*)}$.*

We now present an extension of this result for losses with range $[0, V]$.

Corollary 7.5 *Let $g_1(x) = x$ and $y_2 = 1$ be common settings for the following two problems. The instantiation of problem (47) with $\mathcal{S} = [-V, V]$, $h(x) = -e^{(\eta/2)x}$, $g_2(x) = e^{\eta x}$, and $y_1 = -\frac{a}{n}$ has the same optimal value as the instantiation of problem (47) with $\mathcal{S} = [-1, 1]$, $h(x) = -e^{(V\eta/2)x}$, $g_2(x) = e^{(V\eta)x}$, and $y_1 = -\frac{a/V}{n}$.*

7.3 Fast Rates

We now show how the above results can be used to obtain an exact oracle inequality with a fast rate. We first present a result for finite classes and then present a result for VC-type classes (classes with logarithmic universal metric entropy).

Theorem 7.6 *Let $(\ell, P, \mathcal{F})$ satisfy the strong η^*-central condition, where $|\mathcal{F}| = N$, ℓ is a nonnegative loss, and $\sup_{f\in\mathcal{F}} \ell_f(Z) \leq V$ a.s. for a constant V. Then for all $n \geq 1$, with probability at least $1 - \delta$*

$$\underset{Z\sim P}{\mathbf{E}}[\ell_{\hat{f}_{\mathbf{Z}}}(Z)] \leq \underset{Z\sim P}{\mathbf{E}}[\ell_{f^*}(Z)] + \frac{5 \max\left\{V, \frac{1}{\eta^*}\right\}\left(\log\frac{1}{\delta} + \log N\right)}{n}.$$

Before presenting the result for VC-type classes, we require some definitions. For a pseudometric space $(\mathcal{G}, d)$, for any $\varepsilon > 0$, let $\mathcal{N}(\varepsilon, \mathcal{G}, d)$ be the ε-covering number of $(\mathcal{G}, d)$; that is, $\mathcal{N}(\varepsilon, \mathcal{G}, d)$ is the minimal number of balls of radius ε needed to cover $\mathcal{G}$. We will further constrain the cover (the set of centers of the balls) to be a subset of $\mathcal{G}$ (i.e. to be proper), thus ensuring that the strong central condition assumption transfers to any (proper) cover of $\mathcal{F}$. Note that the 'proper' requirement at most doubles the constant K below, as shown in Lemma 2.1 of Vidyasagar (2002).

We now present the fast rates result for VC-type classes. The proof, which can be found as the proof of Theorem 7 of Mehta and Williamson (2014), uses Theorem 6 of Mehta and Williamson (2014) and the proof of Theorem 7.6. Below, we denote the loss-composed version of a function class $\mathcal{F}$ as $\ell \circ \mathcal{F} := \{\ell_f : f \in \mathcal{F}\}$.

Theorem 7.7 *Let $(\ell, P, \mathcal{F})$ satisfy the strong η^*-central condition with $\ell \circ \mathcal{F}$ separable, where, for a constant $K \geq 1$, for each $\varepsilon \in (0, K]$ we have $\mathcal{N}(\ell \circ \mathcal{F}, L_2(P), \varepsilon) \leq \left(\frac{K}{\varepsilon}\right)^{\mathcal{C}}$, and $\sup_{f \in \mathcal{F}} \ell\big(Y, f(X)\big) \leq V$ a.s. for a constant $V \geq 1$. Then for all $n \geq 5$ and $\delta \leq \frac{1}{2}$, with probability at least $1 - \delta$,*

$$\mathop{\mathbf{E}}_{Z \sim P}[\ell_{\hat{f}_{\mathbf{Z}}}(Z)] \leq \mathop{\mathbf{E}}_{Z \sim P}[\ell_{f^*}(Z)] + \frac{1}{n} \max \left\{ \begin{array}{c} 8 \max\left\{V, \frac{1}{\eta^*}\right\} \left(\mathcal{C} \log(Kn) + \log \frac{2}{\delta}\right), \\ 2V \left(1080 \mathcal{C} \log(2Kn) + 90 \sqrt{\left(\log \frac{2}{\delta}\right) \mathcal{C} \log(2Kn)} + \log \frac{2e}{\delta}\right) \end{array} \right\} + \frac{1}{n}.$$

We have shown the fast rate of $O(1/n)$ under the best case of the v-central condition, i.e. when v is constant; however, it also is possible to recover intermediate rates for the case of general v.

Corollary 7.8 *Let $(\ell, P, \mathcal{F})$ satisfy the v-central condition hold for a finite class $\mathcal{F}$. Then, for some constant c, for all n satisfying $v\left(w^{-1}\left(\frac{5(\log \frac{1}{\delta} + \log N)}{cn}\right)\right) \leq \frac{1}{cV}$, we get an intermediate rate of $w\left(\frac{5(\log \frac{1}{\delta} + \log N)}{cn}\right)$, where w is the inverse of the function $x \mapsto xv(x)$.*

Proof From part (2) of Theorem 5.4, the v-central condition implies the u-Bernstein condition for $u(x) \asymp x/v(x)$, and from part (1b) of Theorem 5.4, we then have the η-central condition for $\eta = cv(\delta)$ for the subclass of functions with excess risk above δ, for some constant c. From here, a simple modification of the proof of Theorem 7.6 yields the desired result as follows. Let ε correspond to the excess risk threshold above which ERM should reject all functions with high probability. Then, similar to the proof of Theorem 7.6, we upper bound the probability of ERM picking a function with excess risk ε or higher:

$$\begin{aligned} N \exp(n\Lambda_{-W_f}(cv(\varepsilon)) &= N \exp(n\Lambda_{-W_f/V}(cVv(\varepsilon)) \\ &\leq N \exp\left(-0.21n\big(cVv(\varepsilon) \wedge 1\big)\frac{\varepsilon}{V}\right). \end{aligned}$$

For ε satisfying $v(\varepsilon) \leq \frac{1}{cV}$, the failure probability δ is at most $N \exp(-0.21cn\varepsilon v(\varepsilon))$, and hence by inversion we get the rate $w\left(\frac{5(\log \frac{1}{\delta} + \log N)}{cn}\right)$. ∎

8. Discussion, Open Problems and Concluding Remarks

In this paper we identified four general conditions for fast and intermediate learning rates. The two main ones, which subsumed many previously identified conditions, where the central condition and stochastic mixability. We provided sufficient assumptions under which the four conditions become equivalent via the implications

$$\eta\text{-central} \Rightarrow \eta\text{-predictor} \Rightarrow \eta\text{-stochastic mixability} \Rightarrow \eta\text{-PPC} \Rightarrow \eta\text{-central}. \tag{55}$$

In Section 3 and 4 we considered the versions of these conditions for fixed $\eta > 0$, as given by Theorem 4.17, Proposition 4.11, Proposition 4.12 and Theorem 3.10, respectively. For this fixed $\eta > 0$ case, all implications except one hold under surprisingly weak conditions, in particular allowing for unbounded loss functions. The exception is 'central $\Rightarrow$ predictor' (Theorem 4.17). Although even this result was applicable to some non-compact decision sets $\mathcal{F}$ with unbounded losses (Example 4.21), it requires tightness and convexity of the set $\mathcal{P}$, although Example 4.18 shows that sometimes the implication holds even though $\mathcal{P}$ is neither tight nor convex. An important open question is whether Theorem 4.17 still holds under weaker versions of Assumption C or Assumption D.

Another restriction of Theorem 4.17 is that, via Assumption D, it requires convexity of the decision set $\mathcal{F}_{\mathrm{d}}$, which fails for the 0/1-loss ℓ^{01} and its conditional version, the classification loss ℓ^{class}. However, we may extend the definition of ℓ^{01} to $\mathcal{F} = [0,1]$ and define the resulting *randomized* 0/1 or *absolute* loss as $\ell^{\mathrm{abs}}_f(z) := |y - f|$. This can be interpreted as the 0/1-loss a decision maker expects to make if she is allowed to randomize her decision by flipping a coin with bias f — a standard concept in PAC-Bayesian approaches (Audibert, 2004; Catoni, 2007). For the absolute loss, we can consider η-stochastic mixability for $\mathcal{F}_{\mathrm{d}} = \mathrm{co}(\mathcal{F}) = [0,1]$, which is convex; hence, the requirement of convex $\mathcal{F}_{\mathrm{d}}$ in Theorem 4.17 is not such a concern.

In Section 5 we discussed weakenings of the four conditions to their v-versions. Now for *bounded* losses, the four implications above still hold under similar conditions as for the fixed η-case. Since the first three implications in (55) were proven in an 'up to ε' form for all $\varepsilon > 0$, it immediately follows that for arbitrary functions v, the implications continue to hold under the same assumptions if the η-conditions are replaced by the corresponding v-conditions. This does not work for the fourth implication, since Theorem 3.10 is not given in an 'up to ε' form (indeed, we conjecture that it does not hold in this form). However, we can work around this issue by using instead a detour via the Bernstein condition: by using first part 2 and then part 1 in Theorem 5.4, it follows that the v-PPC condition implies the v'-central condition for $v'(\varepsilon) \asymp v(\varepsilon)$, so the four v-conditions still imply each other, under the same assumptions as before, up to constant factors. However, the Bernstein-detour works only for bounded losses, and Example 5.7, 5.8 and 5.10 together indicate that in general it cannot be made to work and indeed the analogue of (55) for the v-conditions does not hold for unbounded losses: for decision problems with polynomial rather than exponential tails on the losses, v-stochastic mixability and the v-PPC condition may hold whereas the v-central condition does not. Thus there is the question whether the central condition can be weakened such that the four implications for the v-versions continue to hold, under weak conditions, for unbounded losses — and we regard this as the main open question posed by this work. Another issue here is that, if in a decision problem $(\ell, \mathcal{P}, \mathcal{F})$

that satisfies a v-condition, we replace $\mathcal{P}$ by its convex closure, then the v-condition may very well be broken, so, once again, a weakening of Assumption D to nonconvex $\mathcal{P}$ seems required. Finally, it would be of considerable interest if one could show an analogue for unbounded losses of Proposition 5.11, which connects — for bounded losses — the central condition to the existence of a unique risk minimizer. Relatedly, it would be desirable to link this proposition to the results by Mendelson (2008a) who also connects slow rates with nonunique risk minimizers, and to Koltchinskii (2006) who gives a version of the Bernstein condition that does hold if nonunique minimizers exist, indicating that our η-central condition (which via Proposition 3.3 implies unique minimizers) might sometimes be too strong.

Apart from these implications in the 'main quadrangle' of Figure 1 on page 1798, it would be good to strengthen some of the other connections shown in that figure, such as the precise relation between η-mixability and η-exp-concavity. It would also be desirable to establish connections to results in *defensive forecasting* (Chernov et al., 2010) in which conditions similar to both the central condition and mixability play a role; their Theorem 9 is reminiscent of the special case of our Theorem 4.17 for the case that $\mathcal{Z}$ is finite and $\mathcal{P}$ consists of all distributions on $\mathcal{Z}$.

We focused on showing *equivalence* of fast rate conditions and not on showing that one can actually always *obtain* fast rates under these conditions. For stochastic mixability, this immediately follows, under no further conditions, from Proposition 4.5. For the central condition, the situation is more complicated: in this paper we only showed that it implies fast rates for bounded loss functions. We know that, for the unbounded log-loss, fast rates can be obtained under the central condition (and no additional conditions) in a weaker sense, involving Rényi and squared Hellinger distance (Section 2.2); in work in progress, we aim at showing that the central condition implies fast rates in the standard sense even for unbounded loss functions. This does appear possible, up to log-factors, however it seems that here one does need weak additional conditions such as existence of certain moments different from the exponential moment in (4).

Second, by 'fast' rates we merely meant rates of order $1/n$; it would of course be highly desirable to characterize when the rates that are achieved under our conditions by appropriate algorithms (ERM, Bayes MAP-style and MDL methods for the central condition, the aggregating algorithm for stochastic mixability) are indeed minimax optimal. Similarly, one would need examples showing that if a condition fails, then the corresponding fast or intermediate rates *cannot* be obtained in general. While several such results are available, they either focus on showing that, in the worst-case over all $P \in \mathcal{P}$, *no* learning algorithm, proper or improper, can achieve a certain rate (in particular Audibert (2009) gives very general results), or that a particular proper learning algorithm such as ERM cannot achieve a certain rate (Mendelson, 2008a). Currently unexplored, it seems, are minimax results where one looks at the optimal (not just ERM) algorithm, but within the restricted class of all proper learning algorithms.

In the spirit of Vapnik and Chervonenkis, who discovered under what conditions one can learn from a finite amount of data at all, we continue our quest for conditions under which one can learn from data using not too many examples.

Acknowledgments

We thank Olivier Catoni for raising the issue of unbounded losses discussed in Example 5.7, Wouter Koolen for suggesting the connection to minimax theorems, and Andrew Barron for various in-depth discussions over the last 16 years. Most of the results in Section 7 were published before in the conference paper by Mehta and Williamson (2014); very preliminary versions of Theorem 3.10, Theorem 4.17 (only for $\mathcal{P}$ the set of all distributions on $\mathcal{Z}$) and Theorem 5.4 were published before by van Erven et al. (2012a), in which we used the phrase '$(\ell, \mathcal{P}, \mathcal{F})$ is stochastically mixable' to denote what we now refer to as '$(\ell, \mathcal{P}, \mathcal{F})$ satisfies the central condition'. We thank both the referees of the present paper and the referees of these earlier conference papers for their useful feedback. This work was supported in part by the Australian Research Council, and by NICTA which is funded by the Australian Government, as well as by by the Netherlands Organiszation for Scientific Research (NWO) Project 639.073.904.

Appendix A. Additional Proofs

A.1 Proof of Theorem 3.10 in Section 3

Proof We first consider the case that Assumption A holds, and then the case of bounded loss.

Under Assumption A. Under our Assumption A, we can, for each $P \in \mathcal{P}$, define $\phi(P) := f^* \in \mathcal{F}$ to be optimal in the sense of (3). Note that f^* depends on P, but not on any Π. Since we also assume the weak η-pseudoprobability convexity condition, we must have that for every $\varepsilon > 0$, the η-pseudoprobability convexity condition holds up to ε for some function ϕ_ε. It follows that for all $\varepsilon > 0$, $\mathbf{E}_{Z\sim P}[\ell_{f^*}(Z)] \leq \mathbf{E}_{Z\sim P}[\ell_{\phi_\varepsilon(P)}(Z)] \leq \mathbf{E}_{Z\sim P}[m^\eta_\Pi(Z)] + \varepsilon$, so that also

$$\mathop{\mathbf{E}}_{Z\sim P}[\ell_{f^*}(Z)] \leq \mathop{\mathbf{E}}_{Z\sim P}[m^\eta_\Pi(Z)] \tag{56}$$

for all $\Pi \in \Delta(\mathcal{F})$. Now fix arbitrary $P \in \mathcal{P}$, let $f^* = \phi(P)$ and let $f \in \mathcal{F}$ be arbitrary and consider the special case that $\Pi = (1-\lambda)\delta_{f^*} + \lambda\delta_f$ for $\lambda \in [0, \frac{1}{2}]$, where δ_f is a point-mass on f. Let

$$\chi(\lambda, z) = \eta m^\eta_\Pi(z) = -\log\Big((1-\lambda)e^{-\eta\ell_{f^*}(z)} + \lambda e^{-\eta\ell_f(z)}\Big)$$

be the corresponding mix loss multiplied by η, and let

$$\chi(\lambda) = \mathop{\mathbf{E}}_{Z\sim P}[\chi(\lambda, Z)] = \eta \mathop{\mathbf{E}}_{Z\sim P}[m^\eta_\Pi(Z)]$$

be its expected value. Then from (56) it follows that $\chi(\lambda)$ is minimized at $\lambda = 0$, which implies that the right-derivative $\chi'(0)$ at 0 is nonnegative:

$$\chi'(0) \geq 0. \tag{57}$$

In order to compute $\chi'(0)$, we first observe that, for any z, $\chi(\lambda, z)$ is convex in λ, because it is the composition of the negative logarithm with a linear function. Convexity of $\chi(\lambda, z)$ in

λ implies that the slope $s(d,z) = \frac{\chi(0+d,z)-\chi(0,z)}{d}$ is non-decreasing in $d \in (0, \frac{1}{2}]$ and achieves its maximum value at $d = 1/2$, where it never exceeds $2\log 2$:

$$s(1/2, z) = 2\log \frac{e^{-\eta\ell_{f^*}(z)}}{\frac{1}{2}e^{-\eta\ell_{f^*}(z)} + \frac{1}{2}e^{-\eta\ell_f(z)}} \leq 2\log \frac{e^{-\eta\ell_{f^*}(z)}}{\frac{1}{2}e^{-\eta\ell_{f^*}(z)}} = 2\log 2.$$

Hence $\mathbf{E}_{Z\sim P}[s(\frac{1}{2}, Z)] \leq 2\log 2 < \infty$ and by the monotone convergence theorem (Shiryaev, 1996)

$$\chi'(0) = \lim_{d\downarrow 0} \mathop{\mathbf{E}}_{Z\sim P}[s(d,Z)] = \mathop{\mathbf{E}}_{Z\sim P}\left[\lim_{d\downarrow 0} s(d,Z)\right] = \mathop{\mathbf{E}}_{Z\sim P}\left[\frac{\mathrm{d}}{\mathrm{d}\lambda}\chi(\lambda, Z)|_{\lambda=0}\right] = 1 - \mathop{\mathbf{E}}_{Z\sim P}\left[\frac{e^{-\eta\ell_f(Z)}}{e^{-\eta\ell_{f^*}(Z)}}\right]. \tag{58}$$

Together with (57) and the fact that $\phi(P) = f^*$ and that P was chosen arbitrarily, this implies the strong η-central condition as required.

When the Loss is Bounded. Let $P \in \mathcal{P}$ be arbitrary. The η-pseudoprobability convexity condition implies that for any $\gamma > 0$ we can find $f^* \in \mathcal{F}$ such that

$$\mathop{\mathbf{E}}_{Z\sim P}[\ell_{f^*}(Z)] \leq \mathop{\mathbf{E}}_{Z\sim P}\left[m^\eta_\Pi(Z)\right] + \gamma$$

for all distributions $\Pi \in \Delta(\mathcal{F})$. Choose any $f \in \mathcal{F}$ and consider again the special case $\Pi = (1-\lambda)\delta_{f^*} + \lambda\delta_f$ for $\lambda \in [0, \frac{1}{2}]$, which gives

$$\chi(0) \leq \chi(\lambda) + \eta\gamma \tag{59}$$

for $\chi(\lambda)$ as above. This time $\chi(0)$ is not necessarily the exact minimum of $\chi(\lambda)$, but (59) expresses that it is close. To control $\chi'(0)$, we use that

$$\chi(\lambda, z) = \chi(0,z) + \lambda\tfrac{\mathrm{d}}{\mathrm{d}\lambda}\chi(0,z) + \tfrac{1}{2}\lambda^2\tfrac{\mathrm{d}^2}{\mathrm{d}\lambda^2}\chi(\xi, z) \qquad \text{for some } \xi \in [0,\lambda]$$

by a second-order Taylor expansion in λ, which implies that

$$\chi(\lambda) - \chi(0) - \lambda\chi'(0) \leq \frac{\lambda^2}{2}\max_{z,\lambda'}\left(\frac{e^{-\eta\ell_{f^*}(z)} - e^{-\eta\ell_f(z)}}{(1-\lambda')e^{-\eta\ell_{f^*}(z)} + \lambda' e^{-\eta\ell_f(z)}}\right)^2 \leq \frac{\lambda^2}{2}\left(e^{\eta 2B} - 1\right)^2.$$

Together with (59) the choice $\lambda = \sqrt{\gamma}$ (which requires $\gamma \leq 1/4$) then allows us to conclude that

$$\begin{aligned} -\eta\gamma \leq \chi(\sqrt{\gamma}) - \chi(0) &\leq \sqrt{\gamma}\chi'(0) + \frac{\gamma}{2}\left(e^{\eta 2B} - 1\right)^2 \\ \chi'(0) &\geq -c\sqrt{\gamma} \end{aligned}$$

for $c = \eta + \frac{1}{2}(e^{\eta 2B} - 1)^2$. Since (58) still holds, taking γ small enough that $1 + c\sqrt{\gamma} \leq e^{\eta\varepsilon}$ gives us the central condition (12) for any $\varepsilon > 0$. ■

A.2 Proof of Lemma 4.16 in Section 4

Proof Theorem 6.1 of Grünwald and Dawid (2004), itself a direct consequence of a minimax theorem due to Ferguson (1967), states the following: if a set of distributions $\bar{\mathcal{P}}$ is convex, tight and closed in the weak topology, and $L\colon \mathcal{Z} \times \mathcal{F}_{\mathrm{d}} \to \mathbb{R}$ is a function such that, for all f, $L(z,f)$ is bounded from above and upper semi-continuous in z, then

$$\sup_{P\in\bar{\mathcal{P}}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[L(Z,f)] = \inf_{\rho\in\Delta(\mathcal{F}_{\mathrm{d}})} \sup_{P\in\bar{\mathcal{P}}} \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [L(Z,f)]. \tag{60}$$

Let $\Pi \in \Delta(\mathcal{F}_{\mathrm{d}})$ be arbitrary, and observe that $S^{\eta}_{\Pi}(P,f)$ is related to $\xi_{Z,f}$ via

$$S^{\eta}_{\Pi}(P,f) = \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}],$$

so we will aim to apply (60) with $L(z,f)$ approximately equal to $\xi_{z,f}$. Although $\xi_{z,f}$ is not necessarily bounded above, rewriting

$$\xi_{z,f} = e^{\eta\ell_f(z)} \mathop{\mathbf{E}}_{g\sim\Pi}\left[e^{-\eta\ell_g(z)}\right],$$

we find that it is continuous in z, because $\ell_f(z)$ is continuous in z and $\mathbf{E}_{g\sim\Pi}\left[e^{-\eta\ell_g(z)}\right]$ is also continuous in z by continuity of $\ell_g(z)$ and the dominated convergence theorem (Shiryaev, 1996), which applies because $|e^{-\eta\ell_g(z)}| \leq 1$. Letting $a \wedge b$ denote the minimum of a and b, it follows that $\xi_{z,f} \wedge b$ is also continuous in z for any number b.

Thus we can apply (60) to the function $L(z,f) = \xi_{z,f} \wedge b$, with $\bar{\mathcal{P}}$ the closure of $\mathcal{P}$ in the weak topology, to obtain

$$\inf_{\rho\in\Delta(\mathcal{F}_{\mathrm{d}})} \sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [\xi_{Z,f}\wedge b] \leq \inf_{\rho\in\Delta(\mathcal{F}_{\mathrm{d}})} \sup_{P\in\bar{\mathcal{P}}} \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [\xi_{Z,f}\wedge b] = \sup_{P\in\bar{\mathcal{P}}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}\wedge b]. \tag{61}$$

We will show that

$$\sup_{P\in\bar{\mathcal{P}}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}\wedge b] \leq \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}\wedge b]. \tag{62}$$

If $\mathcal{P}$ is closed itself (first possibility in D.4), then $\bar{\mathcal{P}} = \mathcal{P}$ and this is immediate. The second possibility will be covered at the end of the proof.

Together, (61) and (62) imply that

$$\inf_{\rho\in\Delta(\mathcal{F}_{\mathrm{d}})} \sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [\xi_{Z,f}\wedge b] \leq \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}\wedge b] \leq \sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}]$$

for any finite b. We will show that, for every $\varepsilon > 0$, there exists a b such that

$$\mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [\xi_{Z,f}\wedge b] \geq \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [\xi_{Z,f}] - \varepsilon \qquad \text{for all } \rho\in\Delta(\mathcal{F}_{\mathrm{d}}) \text{ and } P\in\mathcal{P}. \tag{63}$$

By letting ε tend to 0, we can therefore conclude that

$$\sup_{P\in\mathcal{P}} \inf_{f\in\mathcal{F}_{\mathrm{d}}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}] \geq \inf_{\rho\in\Delta(\mathcal{F}_{\mathrm{d}})} \sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P} \mathop{\mathbf{E}}_{f\sim\rho} [\xi_{Z,f}] = \inf_{f\in\mathcal{F}_{\mathrm{d}}} \sup_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{Z\sim P}[\xi_{Z,f}], \tag{64}$$

where the identity follows from the requirement that $e^{\eta \ell_f(z)}$ is convex in f, which implies that $\xi_{Z,f}$ is also convex in f, and hence the mean of ρ is always at least as good as ρ itself: $\xi_{Z,\mathbf{E}_{f\sim\rho}[f]} \leq \mathbf{E}_{f\sim\rho}[\xi_{Z,f}]$. Since the sup inf never exceeds the inf sup, (64) implies (32), which was to be shown.

To prove (63), we observe that

$$\mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{f\sim\rho}[\xi_{Z,f} \wedge b] \geq \mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{f\sim\rho}[\xi_{Z,f}\,[\![\xi_{Z,f} < b]\!]] = \mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{f\sim\rho}[\xi_{Z,f}] - \mathop{\mathbf{E}}_{Z\sim P}\mathop{\mathbf{E}}_{f\sim\rho}[\xi_{Z,f}\,[\![\xi_{Z,f} \geq b]\!]],$$

and, by uniform integrability, we can take b large enough that $\mathbf{E}_{Z\sim P}\,\mathbf{E}_{f\sim\rho}[\xi_{Z,f}\,[\![\xi_{Z,f} \geq b]\!]] \leq \varepsilon$ for all ρ and P, as required.

Finally, it remains to establish (62) for the second possibility in Assumption D.4. To this end, let $\varepsilon > 0$ be arbitrary and let $\mathcal{Z}' \subseteq \mathcal{Z}$ be a compact set such that $P(\mathcal{Z}') \geq 1 - \varepsilon$ for all $P \in \mathcal{P}$. In addition, let $\delta > 0$ be small enough that

$$\sup_{z\in\mathcal{Z}'} |\ell_f(z) - \ell_g(z)| < \varepsilon \qquad \text{for all } f, g \in \mathcal{F}_{\mathrm{d}} \text{ such that } d(f,g) < \delta,$$

which is possible by the assumption of uniform equicontinuity. Since $\mathcal{F}_{\mathrm{d}}$ is totally bounded, it can be covered by a finite number of balls of radius δ. Let $\ddot{\mathcal{F}}_{\mathrm{d}} \subseteq \mathcal{F}_{\mathrm{d}}$ be the (finite) set of centers of those balls. Then we can bound the left-hand side of (62) as follows:

$$\sup_{P\in\bar{\mathcal{P}}}\inf_{f\in\mathcal{F}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[L(Z,f)] \leq \sup_{P\in\bar{\mathcal{P}}}\min_{f\in\ddot{\mathcal{F}}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[L(Z,f)] = \sup_{P\in\mathcal{P}}\min_{f\in\ddot{\mathcal{F}}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[L(Z,f)],$$

where the equality holds by continuity of $\mathbf{E}_{Z\sim P}[L(Z,f)]$ and hence $\min_{f\in\ddot{\mathcal{F}}_{\mathrm{d}}} \mathbf{E}_{Z\sim P}[L(Z,f)]$ in P. We now need to relate $\ddot{\mathcal{F}}_{\mathrm{d}}$ back to $\mathcal{F}_{\mathrm{d}}$, which is possible because, for every $f \in \mathcal{F}_{\mathrm{d}}$, there exists $\ddot{f} \in \ddot{\mathcal{F}}_{\mathrm{d}}$ such that $d(f,\ddot{f}) < \delta$ and hence $|\ell_{\ddot{f}}(z) - \ell_f(z)| < \varepsilon$ for all $z \in \mathcal{Z}'$. It follows that $L(z,\ddot{f}) \leq e^{\eta\varepsilon} L(z,f)$ and therefore

$$\begin{aligned}\sup_{P\in\mathcal{P}}\min_{f\in\ddot{\mathcal{F}}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[L(Z,f)] &\leq \sup_{P\in\mathcal{P}}\min_{f\in\ddot{\mathcal{F}}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[[\![Z\in\mathcal{Z}']\!]\,L(Z,f)] + \varepsilon b \\ &\leq e^{\eta\varepsilon}\sup_{P\in\mathcal{P}}\inf_{f\in\mathcal{F}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[[\![Z\in\mathcal{Z}']\!]\,L(Z,f)] + \varepsilon b \leq e^{\eta\varepsilon}\sup_{P\in\mathcal{P}}\inf_{f\in\mathcal{F}_{\mathrm{d}}}\mathop{\mathbf{E}}_{Z\sim P}[L(Z,f)] + \varepsilon b,\end{aligned}$$

and letting ε tend to 0 we obtain (62), which completes the proof. ■

A.3 Proof of Theorem 5.4 in Section 5

Proof We prove the two cases in turn.

Bernstein ⇒ Central. Fix arbitrary $P \in \mathcal{P}$, and let f^* be $\mathcal{F}$-optimal, i.e. satisfying (3). In this part of the proof, all expectations $\mathbf{E}$ are taken over $Z \sim P$.

Suppose that the u-Bernstein condition holds. Fix arbitrary $f \in \mathcal{F}$ and let $X = \ell_f(Z) - \ell_{f^*}(Z)$. Let $\varepsilon \geq 0$ and set $\eta = v(\varepsilon) \leq c_1^b \varepsilon / u(\varepsilon)$. We deal with $\varepsilon = 0$ later and for now focus on the case $\varepsilon > 0$, which implies $\eta > 0$. Then Lemma 5.6, applied to the random variable ηX, gives

$$\mathbf{E}[X] + \frac{1}{\eta}\log\mathbf{E}[e^{-\eta X}] \leq \kappa(2ba)\eta\,\mathbf{Var}(X) \leq \kappa(2ba)\eta u(\mathbf{E}[X]) \leq \frac{\varepsilon}{u(\varepsilon)}u(\mathbf{E}[X]).$$

If $\varepsilon \leq \mathbf{E}[X]$, then the assumption that $\frac{u(\varepsilon)}{\varepsilon}$ is non-increasing in ε implies that

$$\frac{\varepsilon}{u(\varepsilon)} u(\mathbf{E}[X]) \leq \frac{\mathbf{E}[X]}{u(\mathbf{E}[X])} u(\mathbf{E}[X]) = \mathbf{E}[X], \tag{65}$$

and we can conclude that $\frac{1}{\eta} \log \mathbf{E}[e^{-\eta X}] \leq 0 \leq \varepsilon$. This inequality establishes (b), and it establishes (a) for the case $0 < \varepsilon \leq \mathbf{E}[X]$. If $\varepsilon > \mathbf{E}[X]$, then the assumption that u is non-decreasing implies that

$$\frac{\varepsilon}{u(\varepsilon)} u(\mathbf{E}[X]) \leq \frac{\varepsilon}{u(\mathbf{E}[X])} u(\mathbf{E}[X]) = \varepsilon, \tag{66}$$

and, using that $\mathbf{E}[X] \geq 0$, we again find that $\frac{1}{\eta} \log \mathbf{E}[e^{-\eta X}] \leq \varepsilon$, as required for (a). To finish the proof of (a) we now consider $\varepsilon = 0$. If we also have $v(0) = 0$ then the central condition (12) holds trivially for $\varepsilon = 0$, so we may assume without loss of generality that $v(0) > 0$. Then we must have $\eta = v(0) = \liminf_{x \downarrow 0} x/u(x) > 0$. Now fix a decreasing sequence $\{\varepsilon_j\}_{j=1,2,\ldots}$ tending to 0, where the ε_j are all positive and let $\eta_j = v(\varepsilon_j)$. By the argument above, the η_j-central condition holds up to ε_j. This implies (Fact 3.4) that for all j, all $\eta \leq \eta_j$, in particular for $\eta = v(0)$, the η-central condition also holds up to ε_j. Thus, the η-central condition holds up to ε for all $\varepsilon > 0$. By Proposition 3.11 it then follows that the strong η-central condition holds, i.e. it also holds for $\varepsilon = 0$.

Pseudoprobability ⇒ Bernstein. Suppose that the v-PPC condition holds. Fix some $\varepsilon \geq 0$ and let $\eta = v(\varepsilon)$. Fix arbitrary $P \in \mathcal{P}$ and let f^* be $\mathcal{F}$-optimal for P, achieving (3). Fix arbitrary $f \in \mathcal{F}$ and let Π be the distribution on $\mathcal{F}$ assigning mass $1/2$ to f^* and mass $1/2$ to f, and let $\bar{f} \in \{f, f^*\}$ be the corresponding random variable. For $z \in \mathcal{Z}$, let $Y_{z,\bar{f}} = \eta(\ell_{\bar{f}}(z) - \ell_{f^*}(z))$ and let $\varepsilon_z = \eta^{-1} \log \mathbf{E}_{\bar{f} \sim \Pi}\left[e^{-Y_{z,\bar{f}}}\right]$. Note that $Y_{z,\bar{f}}$ is a random variable under distribution Π (not P, since z is fixed), and that

$$\mathop{\mathbf{E}}_{\bar{f} \sim \Pi}[Y_{z,\bar{f}}] = \frac{1}{2}\eta\left(\ell_f(z) - \ell_{f^*}(z)\right). \tag{67}$$

Lemma 5.6 then gives, for each $z \in \mathcal{Z}$,

$$\kappa(-2ab) \mathop{\mathbf{Var}}_{\bar{f} \sim \Pi}[Y_{z,\bar{f}}] \leq \mathop{\mathbf{E}}_{\bar{f} \sim \Pi}[Y_{z,\bar{f}}] + \log \mathop{\mathbf{E}}_{\bar{f} \sim \Pi}\left[e^{-Y_{z,\bar{f}}}\right] = \frac{1}{2}\eta\left(\ell_f(z) - \ell_{f^*}(z)\right) + \eta\varepsilon_z, \tag{68}$$

where we used the definition of Π and ε_z. We may assume from the definition of the v-pseudoprobability convexity condition that (15) holds for the given ε and η and Π; rearranging this equation it is seen to be equivalent to $\mathbf{E}_{Z \sim P}[\varepsilon_Z] \leq \varepsilon$. By taking expectations over Z on both sides of (68) this gives

$$\kappa(-2ab) \mathop{\mathbf{E}}_{Z \sim P} \mathop{\mathbf{Var}}_{\bar{f} \sim \Pi}[Y_Z] \leq \frac{1}{2}\eta \mathop{\mathbf{E}}_{Z \sim P}[\ell_f(Z) - \ell_{f^*}(Z)] + \eta\varepsilon. \tag{69}$$

The Π-variance on the left can be rewritten, using (67), as

$$\begin{aligned}
\mathop{\mathbf{Var}}_{\bar{f} \sim \Pi}\left[Y_{z,\bar{f}}\right] &= \frac{1}{2}\left(\eta(\ell_f(z) - \ell_{f^*}(z)) - \mathop{\mathbf{E}}_{\bar{f} \sim \Pi}\left[Y_{z,\bar{f}}\right]\right)^2 + \frac{1}{2}\left(\eta \cdot 0 - \mathop{\mathbf{E}}_{\bar{f} \sim \Pi}\left[Y_{z,\bar{f}}\right]\right)^2 \\
&= \frac{1}{2}\left(\frac{1}{2}\eta(\ell_f(z) - \ell_{f^*}(z))\right)^2 + \frac{1}{2}\left(-\frac{1}{2}\eta(\ell_f(z) - \ell_{f^*}(z))\right)^2 = \frac{1}{4}\eta^2(\ell_f(z) - \ell_{f^*}(z))^2.
\end{aligned}$$

Plugging this into (69) and dividing both sides by $\eta^2/(4\kappa(-2ab))$ gives

$$\mathop{\mathbf{E}}_{Z\sim P}(\ell_f(Z) - \ell_{f^*}(Z))^2 \leq \frac{2}{\kappa(-2ab)\cdot\eta}\left(\mathop{\mathbf{E}}_{Z\sim P}\left[\ell_f(Z) - \ell_{f^*}(Z)\right] + 2\varepsilon\right). \tag{70}$$

This holds for all $\varepsilon \geq 0$ and $\eta = v(\varepsilon)$, as long as $\eta = u(\varepsilon) > 0$ (if $\eta = 0$ we cannot divide by η^2 to go from (69) to (70)). Thus, we may set $\varepsilon = \mathbf{E}_{Z\sim P}\left[\ell_f(Z) - \ell_{f^*}(Z)\right] \geq 0$; if $\eta = u(\varepsilon) > 0$ then (70) must hold for ε. With these values the right-hand side becomes $6\eta^{-1}\kappa^{-1}(2ab)\varepsilon = c_2\varepsilon/v(\varepsilon) = u(\varepsilon)$, and the result follows by our choice of ε. It remains to deal with the case $\eta = 0$, which by definition of v can only happen if $\varepsilon = \mathbf{E}_{Z\sim P}\left[\ell_f(Z) - \ell_{f^*}(Z)\right] = 0$. In this case, (70) still holds for all values of $\varepsilon > 0$. We thus infer that the left-hand side of (70) is bounded by $\inf_{\varepsilon>0} 4\varepsilon/(\kappa(-2ab)v(\varepsilon)$, and the result follows by our definition of $0/v(0)$. ■

A.4 Proofs for Section 7

Lemma A.1 (Hyper-Concentrated Excess Losses) *Let Z be a random variable with probability measure P supported on $[-V, V]$. Suppose that $\lim_{\eta\to\infty}\mathbf{E}[\exp(-\eta Z)] < 1$ and $\mathbf{E}[Z] = \mu > 0$. Then there is a suitable modification Z' of Z for which $Z' \leq Z$ with probability 1, the mean of Z' is arbitrarily close to μ, and $\mathbf{E}[\exp(-\eta Z')] = 1$ for arbitrarily large η.*

Proof First, observe that $Z \geq 0$ a.s. If not, then there must be some finite $\eta > 0$ for which $\mathbf{E}[\exp(-\eta Z)] = 1$. Now, consider a random variable Z' with probability measure Q_ε, a modification of Z (with probability measure P) constructed in the following way. Define $A := [\mu, V]$ and $A^- := [-V, -\mu]$. Then for any $\varepsilon > 0$ we define Q_ε as

$$\mathrm{d}Q_\varepsilon(z) = \begin{cases} (1-\varepsilon)\mathrm{d}P(z) & \text{if } z \in A \\ \varepsilon\mathrm{d}P(-z) & \text{if } z \in A^- \\ \mathrm{d}P(z) & \text{otherwise.} \end{cases}$$

Additionally, we couple P and Q_ε such that the couple (Z, Z') is a coupling of (P, Q_ε) satisfying

$$\mathop{\mathbf{E}}_{(Z,Z')\sim(P,Q_\varepsilon)}[\![Z \neq Z']\!] = \min_{(P',Q'_\varepsilon)}\mathop{\mathbf{E}}_{(Z,Z')\sim(P',Q'_\varepsilon)}[\![Z \neq Z']\!],$$

where the min is over all couplings of P and Q_ε. This coupling ensures that $Z' \leq Z$ with probability 1; i.e. Z' is dominated by Z.

Now,

$$\begin{aligned}\mathbf{E}[\exp(-\eta Z')] &= \int_{-V}^{V} e^{-\eta z}\mathrm{d}Q_\varepsilon(z)\\ &= \int_{A^-} e^{-\eta z}\mathrm{d}Q_\varepsilon(z) + \int_A e^{-\eta z}\mathrm{d}Q_\varepsilon(z) + \int_{[0,V]\setminus A} e^{-\eta z}\mathrm{d}Q_\varepsilon(z)\\ &= \varepsilon\int_{A^-} e^{-\eta z}\mathrm{d}P(-z) + (1-\varepsilon)\int_A e^{-\eta z}\mathrm{d}P(z) + \int_{[0,V]\setminus A} e^{-\eta z}\mathrm{d}P(z)\\ &= \varepsilon\int_A e^{\eta z}\mathrm{d}P(z) + (1-\varepsilon)\int_A e^{-\eta z}\mathrm{d}P(z) + \int_{[0,V]\setminus A} e^{-\eta z}\mathrm{d}P(z)\\ &\geq \varepsilon e^{\mu\eta}P(A) + (1-\varepsilon)\int_A e^{-\eta z}\mathrm{d}P(z) + \int_{[0,V]\setminus A} e^{-\eta z}\mathrm{d}P(z). \qquad (71)\end{aligned}$$

Now, on the one hand, for any $\eta > 0$, the sum of the two right-most terms in (71) is strictly less than 1 by assumption. On the other hand, $\eta \to \varepsilon P(A)e^{\mu\eta}$ is exponentially increasing since $\varepsilon > 0$ and $\mu > 0$ (and hence $P(A) > 0$ as well) by assumption; thus, the first term in (71) can be made arbitrarily large by increasing η. Consequently, we can choose $\varepsilon > 0$ as small as desired and then choose $\eta < \infty$ as large as desired such that the mean of Z' is arbitrarily close to μ and $\mathbf{E}[\exp(-\eta Z')] = 1$ respectively. ■

Proof (of Lemma 7.2) Let W denote the convex hull of $g([-1,1])$. We need to see if $\left(-\frac{a}{n}, 1\right) \in W$. Note that W is the convex set formed by starting with the graph of $x \mapsto e^{\eta^* x}$ on the domain $[-1,1]$, including the line segment connecting this curve's endpoints $(-1, e^{-\eta^*})$ to $(1, e^{\eta^* x})$, and including all of the points below this line segment but above the aforementioned graph. That is, W is precisely the set

$$W = \left\{(x,y) \in \mathbb{R}^2 : e^{\eta^* x} \leq y \leq \frac{e^{\eta^*} + e^{-\eta^*}}{2} + \frac{e^{\eta^*} - e^{-\eta^*}}{2}x,\ x \in [-1,1]\right\}.$$

We therefore need to check that $-1 \leq -\frac{a}{n} \leq 1$ and that 1 is sandwiched between the lower and upper bounds at $x = -\frac{a}{n}$. Clearly $-1 \leq -\frac{a}{n} \leq 1$ holds since the loss is in $[0,1]$ by assumption. Using that $\cosh(\eta^*) = \frac{e^{\eta^*}+e^{-\eta^*}}{2}$ and $\sinh(\eta^*) = \frac{e^{\eta^*}-e^{-\eta^*}}{2}$, this means that $k \in W$ if and only if

$$e^{-\eta^* a/n} \leq 1 \leq \cosh(\eta^*) + \sinh(\eta^*)\frac{-a}{n}.$$

Also, since $a > 0$ the inequality $e^{-\eta^* a/n} \leq 1$ holds with *strict* inequality. Thus, we end up with a single requirement characterizing when $k \in W$, which is equivalent to condition (53). Moreover, $k \in \operatorname{int} W$ is characterized by when (53) holds strictly. ■

Proof (of Theorem 7.3) By assumption, the condition of Lemma 7.2 is satisfied, so we can apply Theorem 3 of Kemperman (1968). This gives

$$-\exp\left(\Lambda_{-(\ell_f - \ell_{f^*})(Z)}(\eta^*/2)\right) \geq d_0 - \frac{a}{n}d_1 + d_2, \qquad (72)$$

for all $d^* = (d_0, d_1, d_2) \in \mathbb{R}^3$ such that

$$d_0 + d_1 s + d_2 e^{\eta^* s} + e^{(\eta^*/2)s} \leq 0 \qquad \text{for all } s \in [-1, 1]. \tag{73}$$

To find a good choice of d^*, we will restrict attention to those d^* for which (73) holds with equality at $s = 0$, yielding the constraint

$$d_0 = -d_2 - 1. \tag{74}$$

Plugging this into (73) and changing variables to $c_1 = -d_1/\eta$,[10] and $c_2 = -d_2$, we obtain the constraint

$$u(s) := 1 + c_2(e^{\eta s} - 1) - e^{(\eta/2)s} + \eta c_1 s \geq 0 \qquad \text{for all } s \in [-1, 1].$$

A.4.1 Constraints from the Local Minimum at $\mathbf{0}$

Since $u(0) = 0$, we need $s = 0$ to be a local minimum of u, and so we require the first and second derivative to satisfy

(a) $u'(0) = 0$

(b) $u''(0) \geq 0$,

since otherwise there exists some small $\varepsilon > 0$ such that either $u(\varepsilon) < 0$ or $u(-\varepsilon) < 0$.

For (a), we compute

$$u'(s) = \eta c_2 e^{\eta s} - \frac{\eta}{2} e^{(\eta/2)s} + \eta c_1.$$

Since we require $u'(0) = 0$, we pick up the constraint

$$\eta \left(c_2 - \frac{1}{2} + c_1 \right) = 0,$$

and since $\eta > 0$ by assumption, we have

$$c_1 = \frac{1}{2} - c_2. \tag{75}$$

Thus, we can eliminate c_1 from $u(s)$:

$$u(s) = 1 + c_2(e^{\eta s} - 1) - e^{(\eta/2)s} + \eta \left(\frac{1}{2} - c_2 \right) s.$$

For (b), observe that

$$u''(s) = \eta^2 c_2 e^{\eta s} - \frac{\eta^2}{4} e^{(\eta/2)s},$$

so that $u''(0) = \eta^2 \left(c_2 - \frac{1}{4} \right) \geq 0$, and hence we require

$$c_2 \geq \frac{1}{4}. \tag{76}$$

10. We scale by η here because we are chasing a certain η-dependent rate.

A.4.2 The Other Minima of u

Thus far, we have picked up the constraints (74), (75), and (76), and it remains to choose a value of c_2 such that $u(s) \geq 0$ for all $s \in [-1, 1]$. To this end, observe that $u'(s)$ has at most two roots, because with the substitution $y = e^{(\eta/2)s}$, we have

$$u'(s) = \eta c_2 y^2 - \frac{\eta}{2} y + \eta \left(\frac{1}{2} - c_2 \right),$$

which is a quadratic equation in y with two roots:

$$y \in \left\{ \frac{1 - 2c_2}{2c_2}, 1 \right\} \quad \Rightarrow \quad s \in \left\{ \frac{2}{\eta} \log \frac{1 - 2c_2}{2c_2}, 0 \right\}.$$

Now, since we are taking $c_2 \geq \frac{1}{4}$, the first root is negative, and we find that u is non-decreasing on $[0, 1]$. As we already ensured that $u(0) = 0$, this means that u is non-negative on $[0, 1]$. On the remaining interval, $[-1, 0]$, we know that u is increasing up to $\frac{2}{\eta} \log \frac{1-2c_2}{2c_2}$ and then decreasing until $s = 0$. Since $u(0) = 0$, we therefore need to ensure only that $u(-1) \geq 0$ by finding appropriate conditions on c_2, where

$$\begin{aligned} u(-1) &= 1 + c_2(e^{-\eta} - 1) - e^{-(\eta/2)} - \eta \left(\frac{1}{2} - c_2 \right) \\ &= \left(1 - \frac{\eta}{2} \right) - e^{-(\eta/2)} + c_2 \left(e^{-\eta} - (1 - \eta) \right) \\ c_2 &\geq \frac{e^{-\eta/2} + \frac{\eta}{2} - 1}{e^{-\eta} + \eta - 1} = \frac{1}{4} \frac{\kappa(-\eta/2)}{\kappa(-\eta)}, \end{aligned}$$

where $\kappa(x) = (e^x - x - 1)/x^2$ is increasing in x, which implies that this condition always ensures that $c_2 \geq 1/4$.

We consider the cases $\eta \leq 1$ and $\eta > 1$ separately.

Case $\eta \leq 1$. For $\eta \leq 1$, we will take the value of the constraint at $\eta = 1$. That is,

$$c_2 = \frac{1}{4} \frac{\kappa(-1/2)}{\kappa(-1)} = e^{1/2} - \frac{e}{2}.$$

This is allowed because $\frac{\kappa(-\eta/2)}{\kappa(-\eta)}$ is non-decreasing, as may be verified by observing that

$$\frac{\mathrm{d}}{\mathrm{d}\eta} \frac{e^{-\eta/2} + \frac{\eta}{2} - 1}{e^{-\eta} + \eta - 1} = \frac{e^{\eta/2}(e^{\eta/2} - 1)(e^{\eta} - 1 + e^{\eta/2}\eta)}{2(1 + e^{\eta}(\eta - 1))^2},$$

which is non-negative if $g(\eta) = e^{\eta} - 1 + e^{\eta/2}\eta \geq 0$. This in turn is verified by noting that $g(0) = 0$ and $g'(\eta) = e^{\eta/2}(e^{\eta/2} - \frac{\eta}{2} - 1)$ is positive.

Case $\eta > 1$. Let $c_2 = \frac{1}{2} - \frac{\alpha}{\eta}$ for some $\alpha \geq 0$. With this substitution, we have

$$\begin{aligned} u(-1) &= 1 + c_2(e^{-\eta} - 1) - e^{-(\eta/2)} - \eta \left(\frac{1}{2} - c_2 \right) \\ &= 1 + \left(\frac{1}{2} - \frac{\alpha}{\eta} \right) (e^{-\eta} - 1) - e^{-(\eta/2)} - \alpha \\ &= \left(\frac{1 + e^{-\eta}}{2} - e^{-\eta/2} \right) + \alpha \left(-1 + \frac{1}{\eta} \left(1 - e^{-\eta} \right) \right). \end{aligned}$$

Since we want the above to be nonnegative for all $\eta > 1$, we arrive at the condition

$$\alpha \leq \inf_{\eta \geq 1} \left\{ \frac{\frac{1+e^{-\eta}}{2} - e^{-\eta/2}}{1 - \frac{1}{\eta}\left(1 - e^{-\eta}\right)} \right\}. \tag{77}$$

Plotting suggests that the minimum is attained at $\eta = 1$, with the value $\frac{1}{2}(\sqrt{e}-1)^2 = 0.2104\ldots$. We will fix α to this value and verify that

$$\left(\frac{1+e^{-\eta}}{2} - e^{-\eta/2}\right) + \left(\frac{1}{2}(\sqrt{e}-1)^2\right)\left(-1 + \frac{1}{\eta}\left(1 - e^{-\eta}\right)\right) \geq 0. \tag{78}$$

This is true with equality at $\eta = 0$. The derivative of the LHS with respect to η is

$$\frac{1}{2}e^{-\eta}\left(e^{\eta/2} - 1 - \frac{(\sqrt{e}-1)^2(e^{\eta} - \eta - 1)}{\eta^2}\right).$$

The derivative is positive at $\eta = 1$, so 0 is a candidate minimum. Eventually, $\frac{(\sqrt{e}-1)^2(e^{\eta}-\eta-1)}{\eta^2}$ grows more quickly than $e^{\eta/2}-1$ and surpasses the latter in value. The derivative is therefore negative for all sufficiently large η, and so we need only take the minimum of the LHS of (78) evaluated at $\eta = 1$ and the limiting value as $\eta \to \infty$. We have

$$\lim_{\eta\to\infty}\left(\frac{1+e^{-\eta}}{2} - e^{-\eta/2}\right) + \left(\frac{1}{2}(\sqrt{e}-1)^2\right)\left(-1 + \frac{1}{\eta}\left(1 - e^{-\eta}\right)\right) = \sqrt{e} - \frac{e}{2} \geq 0.$$

Hence, (78) indeed holds for $\alpha \leq 0.21 \leq \frac{1}{2}(\sqrt{e}-1)^2$. We conclude that $u(-1) \geq 0$ when $\alpha \leq \frac{1}{2}(\sqrt{e}-1)^2$.

A.4.3 Putting it All Together

Tracing back our substitutions, we have $d_0 + d_2 = -1$ and $d_1 = -\eta/2 + \eta c_2$, which gives

$$d_0 - \frac{a}{n}d_1 + d_2 = -1 + \frac{a\eta}{n}\left(\frac{1}{2} - c_2\right) \geq -e^{-\frac{a\eta}{n}\left(\frac{1}{2}-c_2\right)}.$$

In the regime $\eta \leq 1$, we choose $c_2 = e^{1/2} - e/2$, which leads to

$$d_0 - \frac{a}{n}d_1 + d_2 \geq -e^{-\frac{0.21\eta a}{n}}. \tag{79}$$

In the regime $\eta > 1$, we take $c_2 = \frac{1}{2} - \frac{1}{2\eta}(\sqrt{e}-1)^2$, which gives

$$d_0 - \frac{a}{n}d_1 + d_2 \geq -e^{-\frac{a}{2n}}. \tag{80}$$

Combining with (72) leads to the desired result. ■

Proof (of Corollary 7.4) Define the function $\Gamma(\eta) := \frac{\cosh(\eta)-1}{\sinh(\eta)}$. For any negative excess loss random variable S', let $\eta_{S'}$ be the maximum η for which $-S'$ is stochastically mixable.

Let W be a stochastically mixable excess loss random variable taking values in $[-1, 1]$ and satisfying $\mathbf{E}[W] = \Gamma(\eta_S) > 0$, and let $S = -W$ be the corresponding negative excess loss random variable.

Let $k_S \in \mathbb{R}^2$ be the moments vector of S, defined as

$$k_S := \begin{pmatrix} \mathbf{E}[S] \\ \mathbf{E}[e^{\eta_S S}] \end{pmatrix} = \begin{pmatrix} -\Gamma(\eta_s) \\ 1 \end{pmatrix}.$$

Because $-\mathbf{E}[S] = \Gamma(\eta_S)$, from Lemma 7.2 the point k_S is extremal with respect to $\mathrm{co}(g([-1, 1]))$. Recall that the goal of this proof is to establish that Theorem 7.3 holds even for the extremal random variable S.

Since $\mathbf{E}[S] < 0$, there exists $A \subset \{x \in \mathbb{R} \colon x < 0\}$ for which we have $\mathbf{Pr}(S \in A) =: p > 0$. Now, consider the following two perturbed versions of S, which we call (I) and (II). In both perturbations, we deflate $\mathbf{Pr}(S \in A)$ by the same (multiplicative) factor $\varepsilon > 0$ uniformly over A so that the overall loss in probability mass over A is ε; this is always possible for small enough ε since $p > 0$, and throughout the rest of the proof we keep implicit that ε is suitably small. The perturbations differ in where they allocate the mass taken from A:

(I) Allocate ε additional mass to $\frac{3}{4}$.

(II) Allocate $\frac{\varepsilon}{2}$ additional mass to $\frac{1}{2}$ and $\frac{\varepsilon}{2}$ additional mass to 1.

We refer to these new random variables as S_I and S_{II}. Observe that

$$\mathbf{E}[S_I] = \mathbf{E}[S_{II}] \geq \mathbf{E}[S] + \frac{3}{4}\varepsilon.$$

Because $\mathbf{E}[S_I] = \mathbf{E}[S_{II}]$, it follows that if we can show that $\eta_{S_I} \neq \eta_{S_{II}}$, then k_{S_I} and $k_{S_{II}}$ cannot both are extremal since Γ is strictly increasing.

Now, by definition, $\mathbf{E} \exp(\eta_{S_I} S_I) = 1$. But observe that by strict convexity, for any $\eta > 0$, we have

$$e^{3\eta/4} < \frac{1}{2}\left(e^{\eta/2} + e^{\eta}\right).$$

Therefore, $\mathbf{E}[\exp(\eta_{S_I} S_I)] > 1$, and so $\eta_{S_{II}} < \eta_{S_I}$. Therefore, k_{S_I} cannot be extremal, and Theorem 7.3 can be applied to the excess loss random variable $-S_I$.

Now, for each (suitably small) ε, we refer to the corresponding S_I more precisely via the notation S_ε, and we define $\eta_\varepsilon := \eta_{S_\varepsilon}$. Since for all $\varepsilon > 0$,

$$\left|\exp\left(\frac{\eta_\varepsilon}{2} S_\varepsilon\right)\right| \leq \exp\left(\frac{\eta_S}{2}\right),$$

and since for each S_ε we have

$$\mathbf{E}\left[\exp\left(\frac{\eta_\varepsilon}{2} S_\varepsilon\right)\right] \leq 1 - 0.21(\eta_\varepsilon \wedge 1)\,\mathbf{E}[-S_\varepsilon],$$

from the dominated convergence theorem it follows that

$$\mathbf{E}\left[\exp\left(\frac{\eta_S}{2} S\right)\right] \leq 1 - 0.21(\eta_S \wedge 1)\,\mathbf{E}[-S],$$

i.e. using the familiar notation $\eta^* = \eta_S$:

$$\mathbf{E}\left[\exp\left(-\frac{\eta^*}{2}W\right)\right] \leq 1 - 0.21(\eta^* \wedge 1)\,\mathbf{E}[W].$$

■

Proof (of Corollary 7.5) Let X be a random variable taking values in $[-V, V]$ with mean $-\frac{a}{n}$ and $\mathbf{E}[e^{\eta X}] = 1$, and let Y be a random variable taking values in $[-1, 1]$ with mean $-\frac{a/V}{n}$ and $\mathbf{E}[e^{(V\eta)Y}] = 1$. Consider a random variable $\tilde{X}$ that is a $\frac{1}{V}$-scaled independent copy of X; observe that $\mathbf{E}[\tilde{X}] = -\frac{a/V}{n}$ and $\mathbf{E}[e^{(V\eta)\tilde{X}}] = 1$. Let the maximal possible value of $\mathbf{E}[e^{(\eta/2)X}]$ be b_X, and let the maximal possible value of $\mathbf{E}[e^{(V\eta/2)Y}]$ be b_Y. We claim that $b_X = b_Y$. Let X be a random variable with a distribution that maximizes $\mathbf{E}[e^{(\eta/2)X}]$ subject to the previously stated constraints on X. Since $\tilde{X}$ satisfies $\mathbf{E}[e^{(V\eta/2)\tilde{X}}] = b_X$, setting $Y = \tilde{X}$ shows that in fact $b_Y \geq b_X$. A symmetric argument (starting with Y and passing to some $\tilde{Y} = VY$) implies that $b_X \geq b_Y$. ■

Proof (of Theorem 7.6) Let $\gamma_n = \frac{a}{n}$ for a constant a to be fixed later. For each $\eta > 0$, let $\mathcal{F}_{\gamma_n}^{(\eta)} \subset \mathcal{F}_{\gamma_n}$ correspond to those functions in $\mathcal{F}_{\gamma_n}$ for which η is the largest constant such that $\mathbf{E}[\exp(-\eta W_f)] = 1$. Let $\mathcal{F}_{\gamma_n}^{\text{hyper}} \subset \mathcal{F}_{\gamma_n}$ correspond to functions f in $\mathcal{F}_{\gamma_n}$ for which $\lim_{\eta\to\infty} \mathbf{E}[\exp(-\eta W_f)] < 1$. Clearly, $\mathcal{F}_{\gamma_n} = \left(\bigcup_{\eta\in[\eta^*,\infty)} \mathcal{F}_{\gamma_n}^{(\eta)}\right) \cup \mathcal{F}_{\gamma_n}^{\text{hyper}}$. The excess loss random variables corresponding to elements $f \in \mathcal{F}_{\gamma_n}^{\text{hyper}}$ are 'hyper-concentrated' in the sense that they are infinitely stochastically mixable. However, Lemma A.1 above shows that for each hyper-concentrated W_f, there exists another excess loss random variable W'_f with mean arbitrarily close to that of W_f, with $\mathbf{E}[\exp(-\eta W'_f)] = 1$ for some arbitrarily large but finite η, and with $W'_f \leq W_f$ with probability 1. The last property implies that the empirical risk of W'_f is no greater than that of W_f; hence for each hyper-concentrated W_f it is sufficient (from the perspective of ERM) to study a corresponding W'_f. From now on, we implicitly make this replacement in $\mathcal{F}_{\gamma_n}$ itself, so that we now have $\mathcal{F}_{\gamma_n} = \bigcup_{\eta\in[\eta^*,\infty)} \mathcal{F}_{\gamma_n}^{(\eta)}$.

Consider an arbitrary $a > 0$. For some fixed $\eta \in [\eta^*, \infty)$ for which $|\mathcal{F}_{\gamma_n}^{(\eta)}| > 0$, consider the subclass $\mathcal{F}_{\gamma_n}^{(\eta)}$. Individually for each such function, we will apply Lemma 7.1 as follows. From Lemma 7.5, we have $\Lambda_{-W_f}(\eta/2) = \Lambda_{-\frac{1}{V}W_f}(V\eta/2)$. From Corollary 7.4, the latter is at most $-\frac{0.21(V\eta\wedge 1)(a/V)}{n} = -\frac{0.21\eta a}{(V\eta\vee 1)n}$. Hence, Lemma 7.1 with $t = 0$ and the η from the lemma taken to be $\eta/2$ implies that the probability of the event $P_n\,\ell(\cdot, f) \leq P_n\,\ell(\cdot, f^*)$ is at most $\exp\left(-0.21\frac{\eta}{V\eta\vee 1}a\right)$. Applying the union bound over all of $\mathcal{F}_{\gamma_n}$, we conclude that

$$\mathbf{Pr}\left\{\exists f \in \mathcal{F}_{\gamma_n} : P_n\,\ell_f \leq P_n\,\ell_{f^*}\right\} \leq N\exp\left(-\eta^*\left(\frac{0.21a}{V\eta^*\vee 1}\right)\right).$$

Since ERM selects hypotheses on their empirical risk, from inversion it holds that with probability at least $1-\delta$ ERM will not select any hypothesis with excess risk at least $\frac{5\max\left\{V, \frac{1}{\eta^*}\right\}\left(\log\frac{1}{\delta} + \log N\right)}{n}$. ■

References

Misha Alekhnovich, Mark Braverman, Vitaly Feldman, Adam R Klivans, and Toniann Pitassi. Learnability and automatizability. In *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on*, pages 621–630. IEEE, 2004.

Sylvain Arlot and Peter L. Bartlett. Margin-adaptive model selection in statistical learning. *Bernoulli*, 17(2):687–713, 2011.

Jean-Yves Audibert. *PAC-Bayesian statistical learning theory*. PhD thesis, Université Paris 6, 2004.

Jean-Yves Audibert. Progressive mixture rules are deviation suboptimal. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 41–48, 2007.

Jean-Yves Audibert. Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37(4):1591–1646, 2009.

Andrew R. Barron. Are Bayes rules consistent in information? *Open Problems in Communication and Computation*, pages 85–91, 1987.

Andrew R. Barron. Personal Communication, 2001.

Andrew R. Barron and Thomas M. Cover. Minimum complexity density estimation. *Information Theory, IEEE Transactions on*, 37(4):1034–1054, 1991.

Peter L. Bartlett and Shahar Mendelson. Empirical minimization. *Probability Theory and Related Fields*, 135(3):311–334, 2006.

Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Patrick Billingsley. *Convergence of Probability Measures*. Wiley, 1968.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Olivier Catoni. *Pac-Bayesian Supervised Classification: The Thermodynamics of Statistical Learning*. IMS, 2007.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Alexey Chernov, Yuri Kalnishkan, Fedor Zhdanov, and Vladimir Vovk. Supermartingales in prediction with expert advice. *Theoretical Computer Science*, 411:2647–2669, 2010.

Arnak S. Dalalyan and Alexandre B. Tsybakov. Mirror averaging with sparsity priors. *Bernoulli*, 18(3):914–944, 2012.

Pierpaolo De Blasi and Stephen G Walker. Bayesian asymptotics with misspecified models. *Statistica Sinica*, 23:169–187, 2013.

Steven de Rooij, Tim van Erven, Peter D. Grünwald, and Wouter M. Koolen. Follow the leader if you can, hedge if you must. *Journal of Machine Learning Research*, 15: 1281–1316, 2014.

Joseph L. Doob. Application of the theory of martingales. *Actes du Colloque International Le Calcul des Probabilités et ses Applications*, pages 23–27, 1949.

Thomas S. Ferguson. *Mathematical Statistics: A Decision Theoretic Approach*. Academic Press, 1967.

David Freedman. On tail probabilities for martingales. *Annals of Probability*, 3:100–118, 1975.

Subhashis Ghosal, Jayanta K. Ghosh, and Aad W. van der Vaart. Convergence rates of posterior distributions. *Annals of Statistics*, 28(2):500–531, 2000.

P. D. Grünwald. Viewing all models as "probabilistic". In *Proceedings of the Twelfth ACM Conference on Computational Learning Theory (COLT' 99)*, pages 171–182, 1999.

Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007.

Peter D. Grünwald. That simple device already used by Gauss. In P.D. Grünwald, P. Myllymäki, I. Tabus, M. Weinberger, and B. Yu, editors, *Festschrift in Honor of Jorma Rissanen on the Occasion of his 75th Birthday*, pages 293–304. Tampere University Press, Tampere, Finland, 2008.

Peter D. Grünwald. Safe learning: Bridging the gap between Bayes, MDL and statistical learning theory via empirical convexity. In *Proceedings of the 24th Conference on Learning Theory*, pages 397–419, 2011.

Peter D. Grünwald. The safe Bayesian. In *Proceedings of the 23rd International Conference on Algorithmic Learning Theory (ALT 2012)*, pages 169–183. Springer, 2012.

Peter D. Grünwald and A. Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.

Peter D. Grünwald and John Langford. Suboptimality of MDL and Bayes in classification under misspecification. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT 2004)*, New York, 2004. Springer-Verlag.

Peter D. Grünwald and Thijs van Ommen. Inconsistency of Bayesian inference for misspecified linear models, and a proposal for repairing it. *arXiv preprint arXiv:1412.3730*, 2014.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2–3):169–192, 2007.

Elad Hazan, Alexander Rakhlin, and Peter L. Bartlett. Adaptive online gradient descent. In J.C. Platt, D. Koller, Y. Singer, and S.T. Roweis, editors, *Advances in Neural Information Processing Systems (NIPS) 20*, pages 65–72, 2008.

Anatoli Juditsky, Philippe Rigollet, and Alexandre B. Tsybakov. Learning by mirror averaging. *The Annals of Statistics*, 36(5):2183–2206, 2008.

Parmeswaran Kamalaruban, Robert C. Williamson, and Xinhua Zhang. Exp-Concavity of Proper Composite Losses. In *JMLR Workshop and Conference Proceedings (Proceedings COLT 2015)*, volume 40, 2015.

Samuel Karlin and William J. Studden. *Tchebycheff Systems: With Applications in Analysis and Statistics*. Interscience Publishers, 1966.

Johannes H. B. Kemperman. The general moment problem, a geometric approach. *The Annals of Mathematical Statistics*, 39(1):93–122, 1968.

Johannes H. B. Kemperman. Geometry of the moment problem. In *Proceedings of Synmposia in Applied Mathematics*, volume 37, pages 16–53, 1987.

Jyrki Kivinen and Manfred Warmuth. Averaging expert predictions. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, pages 153–167, 1999.

Bas J. K. Kleijn and Aad W. van der Vaart. Misspecification in infinite-dimensional Bayesian statistics. *The Annals of Statistics*, 34(2):837–877, 2006.

Vladimir Koltchinskii. Local Rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 2006.

Wouter M. Koolen, Tim van Erven, and Peter D. Grünwald. Learning the learning rate for prediction with expert advice. In Z. Ghahramani, M. Welling, C. Cortes, N.D. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2294–2302, 2014.

Guillaume Lecué. *Interplay between concentration, complexity and geometry in learning theory with applications to high dimensional data analysis*. Habilitation à diriger des recherches, Université Paris-Est, 2011.

Wee Sun Lee, Peter L Bartlett, and Robert C Williamson. Efficient agnostic learning of neural networks with bounded fan-in. *IEEE Transactions on Information Theory*, 42(6): 2118–2132, 1996.

Wee Sun Lee, Peter L. Bartlett, and Robert C. Williamson. The importance of convexity in learning with squared loss. *IEEE Transactions on Information Theory*, 44(5):1974–1980, 1998. Correction 54(9), 4395 (2008).

Jonathan Qiang Li. *Estimation of mixture models*. PhD thesis, Yale University, 1999.

Nishant A. Mehta and Robert C. Williamson. From stochastic mixability to fast rates. In *Advances in Neural Information Processing Systems*, pages 1197–1205, 2014.

Shahar Mendelson. Lower bounds for the empirical minimization algorithm. *IEEE Transactions on Information Theory*, 54(8):3797–3803, 2008a.

Shahar Mendelson. Obtaining fast error rates in nonconvex situations. *Journal of Complexity*, 24(3):380–397, 2008b.

Shahar Mendelson. Learning without concentration. In *Proceedings of The 27th Conference on Learning Theory*, pages 25–39, 2014.

Shahar Mendelson and Robert C. Williamson. Agnostic learning of nonconvex function classes. In *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 2002)*, pages 1–13. Springer, 2002.

Yuri V. Prokhorov. Convergence of random processes and limit theorems in probability theory. *Theory of Probability and Its Applications*, I(2):157–214, 1956.

R.V. Ramamoorthi, Karthik Sriram, and Ryan Martin. On posterior concentration in misspecified models. *arXiv preprint arXiv:1312.4620*, 2013.

Hans Richter. Parameterfreie abschätzung und realisierung von erwartungswerten. *Blätter der DGVFM*, 3(2):147–162, 1957.

Shai Shalev-Shwartz and Yoram Singer. Logarithmic regret algorithms for strongly convex repeated games. Technical report, The Hebrew University, 2007.

Claude E. Shannon. Bounds on the tails of martingales and related questions (seminar notes on information theory, Massachusetts Institute of Technology). In Neil J.A. Sloane and Aaron D. Wyner, editors, *Claude Elwood Shannon Miscellaneous Writings*, pages 621–639. Mathematical Sciences Research Centre, AT&T Bell Laboratories, 1956. URL `https://archive.org/details/ShannonMiscellaneousWritings`.

Albert N. Shiryaev. *Probability*. Springer-Verlag, New York, 1996.

Alexander B. Tsybakov. Optimal aggregation of classifiers in statistical learning. *The Annals of Statistics*, 32(1):135–166, 2004.

Ruth Urner and Shai Ben-David. The sample complexity of agnostic learning under deterministic labels. In *Proceedings of the 27th Annual Conference on Learning Theory (COLT 2014)*, 2014.

Aad W. Van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer, 1996.

Tim van Erven. From exp-concavity to mixability. *Tim van Erven's Blog*, 2012.

Tim van Erven, Peter D. Grünwald, Mark D. Reid, and Robert C. Williamson. Mixability in statistical learning. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 1700–1708, 2012a.

Tim van Erven, Mark D. Reid, and Robert C. Williamson. Mixability is Bayes risk curvature relative to log loss. *Journal of Machine Learning Research*, 13:1639–1663, 2012b.

Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, 1998.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16 (2):264–280, 1971.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. *Theory of Pattern Recognition (in Russian)*. Nauka, Moscow, 1974. German translation: *Theorie der Zeichenerkennung*, Akademie Verlag, Berlin, 1979.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26(3):532–553, 1981.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. The necessary and sufficient conditions for consistency of the method of empirical risk minimization. *Pattern Recognition and Image Analysis*, 1(3):284–305, 1991.

Elodie Vernet, Mark D. Reid, and Robert C. Williamson. Composite multiclass losses. In *Advances in Neural Information Processing Systems*, pages 1224–1232, 2011.

Mathukumalli Vidyasagar. *Learning and Generalization with Applications to Neural Networks*. Springer, 2002.

Vladimir Vovk. Aggregating strategies. In *Proceedings of the third annual workshop on Computational learning theory*, pages 371–383. Morgan Kaufmann Publishers Inc., 1990.

Vladimir Vovk. A game of prediction with expert advice. *Journal of Computer and System Sciences*, 56(2):153–173, 1998.

Vladimir Vovk. Competitive on-line statistics. *International Statistical Review*, 69(2):213–248, 2001.

Vladimir Vovk and Fedor Zhdanov. Prediction with expert advice for the Brier game. *Journal of Machine Learning Research*, 10:2445–2471, 2009.

Yuhong Yang and Andrew R. Barron. Information-theoretic determination of minimax rates of convergence. *The Annals of Statistics*, 27(5):1564–1599, 1999.

Tong Zhang. From ε-entropy to KL-entropy: Analysis of minimum information complexity density estimation. *The Annals of Statistics*, 34(5):2180–2210, 2006a.

Tong Zhang. Information-theoretic upper and lower bounds for statistical estimation. *IEEE Transactions on Information Theory*, 52(4):1307–1321, 2006b.

Journal of Machine Learning Research 16 (2015) 1863-1877 Submitted 6/15; Published 9/15

On the Asymptotic Normality of an Estimate of a Regression Functional

László Györfi * GYORFI@CS.BME.HU
Department of Computer Science and Information Theory
Budapest University of Technology and Economics
Magyar Tudósok körútja 2., H-1117 Budapest, Hungary

Harro Walk WALK@MATHEMATIK.UNI-STUTTGART.DE
Department of Mathematics
University of Stuttgart
Pfaffenwaldring 57, D-70569 Stuttgart, Germany

Editor: Alex Gammerman and Vladimir Vovk

Abstract

An estimate of the second moment of the regression function is introduced. Its asymptotic normality is proved such that the asymptotic variance depends neither on the dimension of the observation vector, nor on the smoothness properties of the regression function. The asymptotic variance is given explicitly.

Keywords: nonparametric estimation, regression functional, central limit theorem, partitioning estimate

1. Introduction

This paper considers a histogram-based estimate of second moment of the regression function in multivariate problems. The interest in the second moment is motivated by the fact that by estimating it one obtains an estimate of the best possible achievable mean squared error, a quantity of obvious statistical interest. It is shown that the estimate is asymptotically normally distributed. It is remarkable that the asymptotic variance only depends on moments of the regression function but neither on its smoothness, nor on the dimension of the space. The proof relies on a Poissonization technique that has been used successfully in related problems.

Let Y be a real valued random variable with $\mathbb{E}\{Y^2\} < \infty$ and let $X = (X^{(1)}, \ldots, X^{(d)})$ be a d-dimensional random observational vector. In regression analysis one wishes to estimate Y given X, i.e., one wants to find a function g defined on the range of X so that $g(X)$ is "close" to Y. Assume that the main aim of the analysis is to minimize the mean squared error :

$$\min_g \mathbb{E}\{(g(X) - Y)^2\}.$$

∗. This research has been partially supported by the European Union and Hungary and co-financed by the European Social Fund through the project TMOP-4.2.2.C-11/1/KONV-2012-0004 - National Research Center for Development and Market Introduction of Advanced Information and Communication Technologies.

As is well-known, this minimum is achieved by the regression function $m(x)$, which is defined by

$$m(x) = \mathbb{E}\{Y \mid X = x\}. \tag{1}$$

For each measurable function g one has

$$\begin{aligned}\mathbb{E}\{(g(X)-Y)^2\} &= \mathbb{E}\{(m(X)-Y)^2\} + \mathbb{E}\{(m(X)-g(X))^2\} \\ &= \mathbb{E}\{(m(X)-Y)^2\} + \int \mid m(x)-g(x) \mid^2 \mu(dx),\end{aligned}$$

where μ stands for the distribution of the observation X.

It is of great importance to be able to estimate the minimum mean squared error

$$L^* = \mathbb{E}\{(m(X)-Y)^2\}$$

accurately, even before a regression estimate is applied: in a standard nonparametric regression design process, one considers a finite number of real-valued features $X^{(i)}$, $i \in I$, and evaluates whether these suffice to explain Y. In case they suffice for the given explanatory task, an estimation method can be applied on the basis of the features already under consideration, if not, more or different features must be considered. The quality of a subvector $\{X^{(i)},\, i \in I\}$ of X is measured by the minimum mean squared error

$$L^*(I) := \mathbb{E}\left(Y - \mathbb{E}\{Y|X^{(i)}, i \in I\}\right)^2$$

that can be achieved using the features as explanatory variables. $L^*(I)$ depends upon the unknown distribution of $(Y, X^{(i)} : i \in I)$. The first phase of any regression estimation process therefore heavily relies on estimates of L^* (even *before* a regression estimate is picked).

Concerning dimension reduction the related testing problem is on the hypothesis

$$L^* = L^*(I).$$

This testing problem can be managed such that we estimate both L^* and $L^*(I)$, and accept the hypothesis if the two estimates are close to each other. (Cf. De Brabanter et al. (2014).)

Devroye et al. (2003), Evans and Jones (2008), Liitiäinen et al. (2008), Liitiäinen et al. (2009), Liitiäinen et al. (2010), and Ferrario and Walk (2012) introduced nearest neighbor based estimates of L^*, proved strong universal consistency and calculated the (fast) rate of convergence.

Because of

$$L^* = \mathbb{E}\{Y^2\} - \mathbb{E}\{m(X)^2\}$$

and $\mathbb{E}\{Y^2\} < \infty$, estimating L^* is equivalent to estimating the second moment S^* of the regression function:

$$S^* = \mathbb{E}\{m(X)^2\} = \int m(x)^2 \mu(dx).$$

In this paper we introduce a partitioning based estimator of S^*, and show its asymptotic normality. It turns out that the asymptotic variance depends neither on the dimension of the observation vector, nor on the smoothness properties of the regression function. The asymptotic variance is given explicitly.

2. A Splitting Estimate

We suppose that the regression estimation problem is based on a sequence

$$(X_1, Y_1), (X_2, Y_2), \ldots$$

of i.i.d. random vectors distributed as (X, Y). Let

$$\mathcal{P}_n = \{A_{n,j}, j = 1, 2, \ldots\}$$

be a cubic partition of $\mathbb{R}^d$ of size $h_n > 0$.

The partitioning estimator of the regression function m is defined as

$$m_n(x) = \frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})} \text{ if } x \in A_{n,j}, \tag{2}$$

(interpreting $0/0 = 0$) with

$$\nu_n(A) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\{X_i \in A\}} Y_i$$

and

$$\mu_n(A) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}_{\{X_i \in A\}}.$$

(Here $\mathbb{I}$ denotes the indicator function.)

If for cubic partition

$$nh_n^d \to \infty \quad \text{and} \quad h_n \to 0 \tag{3}$$

as $n \to \infty$, then the partitioning regression estimate (2) is weakly universally consistent, which means that

$$\lim_{n\to\infty} \mathbb{E}\left\{\int (m_n(x) - m(x))^2 \mu(dx)\right\} = 0 \tag{4}$$

for any distribution of (X, Y) with $\mathbb{E}\{Y^2\} < \infty$, and for bounded Y it holds

$$\lim_{n\to\infty} \int (m_n(x) - m(x))^2 \mu(dx) = 0 \tag{5}$$

a.s. (Cf. Theorems 4.2 and 23.1 in Györfi et al. (2002).)

Assume splitting data

$$Z_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

and

$$D'_n = \{(X'_1, Y'_1), \ldots, (X'_n, Y'_n)\}$$

such that $(X_1, Y_1), \ldots, (X_n, Y_n), (X'_1, Y'_1), \ldots, (X'_n, Y'_n)$ are i.i.d.

The splitting data estimate of S^* is defined as

$$S_n := \frac{1}{n} \sum_{i=1}^{n} Y'_i m_n(X'_i) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{\infty} \mathbb{I}_{\{X'_i \in A_{n,j}\}} Y'_i \frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})}.$$

Put

$$\nu'_n(A) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{\{X'_i \in A\}} Y'_i,$$

then S_n has the equivalent form

$$S_n = \sum_{j=1}^{\infty} \nu'_n(A_{n,j}) \frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})}. \tag{6}$$

Theorem 1 *Assume (3) and that μ is non-atomic and has bounded support. Suppose that there is a finite constant C such that*

$$\mathbb{E}\{|Y|^3 \mid X\} < C. \tag{7}$$

Then

$$\sqrt{n}\left(S_n - \mathbb{E}\{S_n\}\right)/\sigma \xrightarrow{\mathcal{D}} N(0,1),$$

where

$$\sigma^2 = 2\int M_2(x) m(x)^2 \mu(dx) - \left(\int m(x)^2 \mu(dx)\right)^2 - \int m(x)^4 \mu(dx),$$

with

$$M_2(X) = \mathbb{E}\{Y^2 \mid X\}.$$

The estimation problem is motivated by the above mentioned dimension reduction such that one estimates S^* for the original observation vector and for the observation vector where some components are left out. If the two estimates are "close" to each other, then we decide that the left out components are ineffective. Theorem 1 is on the random part of the estimates. Therefore there is a further need to study the difference of the biases of the estimates. Under (3) we have

$$\lim_{n\to\infty} \mathbb{E}\{S_n\} = S^*$$

and for Lipschitz continuous m the rate of convergence can be of order $n^{-1/d}$ for suitable choice of h_n. (Cf. Devroye et al. (2013).) Similarly to De Brabanter et al. (2014) we conjecture that this difference of the biases has universally a fast rate of convergence.

Obviously, there are several other possibilities for defining partitioning based estimates and proving their asymptotic normality, for example,

$$\frac{1}{n}\sum_{i=1}^{n} m_n(X'_i)^2$$

or

$$\sum_{j=1}^{\infty} \frac{\nu_n(A_{n,j})^2}{\mu_n(A_{n,j})}.$$

Notice that both estimates have larger bias and variance than our estimate (6) has.

The proof of Theorem 1 works without any major modification for consistent k_n nearest neighbor (k_n-NN) estimate m_n if $k_n \to \infty$ and $k_n/n \to 0$. A delicate and important research problem is the case of non-consistent 1-NN estimate m_n, because for 1-NN estimate m_n the bias is smaller. We conjecture that even in this case one has a CLT.

We prove Theorem 1 in the next section.

3. Proof of Theorem 1

Introduce the notations

$$U_n = \sqrt{n}\left(S_n - \mathbb{E}\{S_n \mid Z_n\}\right)$$

and

$$V_n = \sqrt{n}\left(\mathbb{E}\{S_n \mid Z_n\} - \mathbb{E}\{S_n\}\right),$$

then

$$\sqrt{n}\left(S_n - \mathbb{E}\{S_n\}\right) = U_n + V_n.$$

We prove Theorem 1 by showing that for any $u, v \in \mathbb{R}$

$$\mathbb{P}\{U_n \le u, V_n \le v\} \to \Phi\left(\frac{u}{\sigma_1}\right)\Phi\left(\frac{v}{\sigma_2}\right) \tag{8}$$

where Φ denotes the standard normal distribution function, and

$$\sigma_1^2 = \int M_2(x)m(x)^2\mu(dx) - \left(\int m(x)^2\mu(dx)\right)^2 \tag{9}$$

and

$$\sigma_2^2 = \int M_2(x)m(x)^2\mu(dx) - \int m(x)^4\mu(dx). \tag{10}$$

Notice that V_n is measurable with respect to Z_n, therefore

$$\begin{aligned}
&\left|\mathbb{P}\{U_n \le u, V_n \le v\} - \Phi\left(\frac{u}{\sigma_1}\right)\Phi\left(\frac{v}{\sigma_2}\right)\right| \\
&= \left|\mathbb{E}\{\mathbb{I}_{\{V_n \le v\}}\mathbb{P}\{U_n \le u \mid Z_n\}\} - \Phi\left(\frac{u}{\sigma_1}\right)\Phi\left(\frac{v}{\sigma_2}\right)\right| \\
&\le \left|\mathbb{E}\left\{\mathbb{I}_{\{V_n \le v\}}\left(\mathbb{P}\{U_n \le u \mid Z_n\} - \Phi\left(\frac{u}{\sigma_1}\right)\right)\right\}\right| \\
&+ \left|\left(\mathbb{P}\{V_n \le v\} - \Phi\left(\frac{v}{\sigma_2}\right)\right)\Phi\left(\frac{u}{\sigma_1}\right)\right| \\
&\le \mathbb{E}\left\{\left|\mathbb{P}\{U_n \le u \mid Z_n\} - \Phi\left(\frac{u}{\sigma_1}\right)\right|\right\} + \left|\mathbb{P}\{V_n \le v\} - \Phi\left(\frac{v}{\sigma_2}\right)\right|.
\end{aligned}$$

Thus, (8) is satisfied if

$$\mathbb{P}\{U_n \le u \mid Z_n\} \to \Phi\left(\frac{u}{\sigma_1}\right) \tag{11}$$

in probability and

$$\mathbb{P}\{V_n \le v\} \to \Phi\left(\frac{v}{\sigma_2}\right). \tag{12}$$

Proof of (11).
Let's start with the representation

$$\begin{aligned}
U_n &= \sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n}(Y_i'm_n(X_i') - \mathbb{E}\{Y_i'm_n(X_i') \mid Z_n\})\right) \\
&= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(Y_i'm_n(X_i') - \mathbb{E}\{Y_i'm_n(X_i') \mid Z_n\}).
\end{aligned}$$

Because of (7) and the Jensen inequality, for any $1 \le s \le 3$, we get

$$M_s(X) := \mathbb{E}\{|Y|^s \mid X\} = (\mathbb{E}\{|Y|^s \mid X\}^{1/s})^s \le (\mathbb{E}\{|Y|^3 \mid X\}^{1/3})^s \le C^{s/3}, \tag{13}$$

especially, for $s = 1$

$$M_1(X) = |m(X)| \le C^{1/3}$$

and

$$\mathbb{E}\{|Y|^3\} \le C.$$

Next we apply a Berry-Esseen type central limit theorem (see Theorem 14 in Petrov (1975)). It implies that

$$\left| \mathbb{P}\{U_n \le u \mid Z_n\} - \Phi\left(\frac{u}{\sqrt{\mathbb{V}ar(Y_1' m_n(X_1') \mid Z_n)}} \right) \right| \le \frac{c}{\sqrt{n}} \frac{\mathbb{E}\{|Y_1' m_n(X_1')|^3 \mid Z_n\}}{\sqrt{\mathbb{V}ar(Y_1' m_n(X_1') \mid Z_n)}^3}$$

with the universal constant $c > 0$. Because of

$$\mathbb{E}\{Y_1' m_n(X_1') \mid Z_n\} = \int m(x) m_n(x) \mu(dx),$$

we get that

$$\begin{aligned} \mathbb{V}ar(Y_1' m_n(X_1') \mid Z_n) &= \mathbb{E}\{{Y_1'}^2 m_n(X_1')^2 \mid Z_n\} - \mathbb{E}\{Y_1' m_n(X_1') \mid Z_n\}^2 \\ &= \int M_2(x) m_n(x)^2 \mu(dx) - \left(\int m(x) m_n(x) \mu(dx) \right)^2. \end{aligned}$$

Now (4), together with the boundedness of M_2 by (13), implies that

$$\mathbb{V}ar(Y_1' m_n(X_1') \mid Z_n) \to \sigma_1^2$$

in probability, where σ_1^2 is defined by (9). Further

$$\mathbb{E}\{|Y_1' m_n(X_1')|^3 \mid Z_n\} \le C \int |m_n(x)|^3 \mu(dx).$$

Put

$$A_n(x) = A_{n,j} \text{ if } x \in A_{n,j}.$$

Again, applying the Jensen inequality we get

$$|m_n(x)|^3 \le \left| \frac{\sum_{i=1}^n \mathbb{I}_{\{X_i \in A_n(x)\}} |Y_i|^{3/2}}{\sum_{i=1}^n \mathbb{I}_{\{X_i \in A_n(x)\}}} \right|^2,$$

the right hand side of which is the square of the regression estimate, where Y is replaced by $|Y|^{3/2}$. Thus, (4) together with $\mathbb{E}\{|Y|^3\} < \infty$ implies that

$$\int \left| \frac{\sum_{i=1}^n \mathbb{I}_{\{X_i \in A_n(x)\}} |Y_i|^{3/2}}{\sum_{i=1}^n \mathbb{I}_{\{X_i \in A_n(x)\}}} \right|^2 \mu(dx) \to \mathbb{E}\{\mathbb{E}\{|Y|^{3/2} \mid X\}^2\} < C$$

in probability. These limit relations imply (11).

Proof of (12).
Assuming that the support S of μ is bounded, let l_n be such that $S \subset \cup_{j=1}^{l_n} A_{n,j}$. Also we re-index the partition so that

$$\mu(A_{n,j}) \geq \mu(A_{n,j+1}),$$

with $\mu(A_{n,j}) > 0$ for $j \leq l_n$, and $\mu(A_{n,j}) = 0$ otherwise. Then,

$$S_n = \sum_{j=1}^{l_n} \nu'_n(A_{n,j}) \frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})}, \tag{14}$$

and

$$l_n \leq \frac{c}{h_n^d}.$$

The condition $nh_n^d \to \infty$ implies that

$$l_n/n \to 0.$$

Because of (14) we have that

$$\begin{aligned} V_n &= \sqrt{n} \sum_{j=1}^{l_n} \mathbb{E}\{\nu'_n(A_{n,j}) \mid Z_n\} \left(\frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})} - \mathbb{E}\left\{ \frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})} \right\} \right) \\ &= \sqrt{n} \sum_{j=1}^{l_n} \nu(A_{n,j}) \left(\frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})} - \mathbb{E}\left\{ \frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})} \right\} \right), \end{aligned}$$

where

$$\nu(A) = \mathbb{E}\{\nu_n(A)\}.$$

Observe that we have to show the asymptotic normality for a finite sum of dependent random variables. In order to prove (12), we follow the lines of the proof in Beirlant and Györfi (1998) and use a Poissonization argument. With this we introduce a modification M_n of V_n such that

$$\Delta_n := V_n - M_n \to 0,$$

the proof of which follows, starting from (23).

Now we proceed arguing for M_n. Introduce the notation N_n for a Poisson(n) random variable independent of $(X_1, Y_1), (X_2, Y_2), \ldots$. Moreover put

$$n\tilde{\nu}_n(A) = \sum_{i=1}^{N_n} I_{\{X_i \in A\}} Y_i$$

and

$$n\tilde{\mu}_n(A) = \sum_{i=1}^{N_n} I_{\{X_i \in A\}}.$$

The key result in this step is the following property:

Proposition 2 *(Beirlant and Mason (1995), Beirlant et al. (1994).) Put*

$$\tilde{M}_n = \sqrt{n}\sum_{j=1}^{l_n}\nu(A_{n,j})\left(\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})} - \mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right\}\right), \tag{15}$$

and

$$M_n = \sqrt{n}\sum_{j=1}^{l_n}\nu(A_{n,j})\left(\frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})} - \mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right\}\right). \tag{16}$$

Assume that

$$\Phi_n(t,v) = \mathbb{E}\left(\exp\left(it\tilde{M}_n + iv\frac{N_n - n}{\sqrt{n}}\right)\right) \to e^{-(t^2\rho^2+v^2)/2}$$

for a constant $\rho > 0$, where $i = \sqrt{-1}$. Then

$$M_n/\rho \xrightarrow{\mathcal{D}} N(0,1).$$

Put

$$T_n = t\tilde{M}_n + v\frac{N_n - n}{\sqrt{n}},$$

for which a central limit result is to hold:

$$T_n \xrightarrow{\mathcal{D}} N\left(0, t^2\rho^2 + v^2\right) \tag{17}$$

as $n \to \infty$. Remark that

$$\mathbb{V}ar(T_n) = t^2\mathbb{V}ar(\tilde{M}_n) + 2tv\mathbb{E}\left\{\tilde{M}_n\frac{N_n - n}{\sqrt{n}}\right\} + v^2.$$

For a cell $A = A_{n,j}$ from the partition with $\mu(A) > 0$, let $Y(A)$ be a random variable such that

$$\mathbb{P}\{Y(A) \in B\} = \mathbb{P}\{Y \in B | X \in A\},$$

where B is an arbitrary Borel set.

Introduce the notations

$$q_{n,k} = \mathbb{P}\{n\mu_n(A) = k\} = \binom{n}{k}\mu(A)^k(1-\mu(A))^{n-k}$$

and

$$\tilde{q}_{n,k} = \mathbb{P}\{n\tilde{\mu}_n(A) = k\} = \frac{(n\mu(A))^k}{k!}e^{-n\mu(A)}.$$

Concerning the expectation, with $(Y_1(A), Y_2(A), \ldots)$ an i.i.d. sequence of random variables distributed as $Y(A)$ we find that

$$\begin{aligned}
\mathbb{E}\left\{\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)}\right\} &= \sum_{k=0}^{\infty}\mathbb{E}\left\{\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)} \mid n\tilde{\mu}_n(A) = k\right\}\mathbb{P}\{n\tilde{\mu}_n(A) = k\} \\
&= \sum_{k=1}^{\infty}\mathbb{E}\left\{\frac{\sum_{i=1}^k Y_i(A)}{k}\right\}\tilde{q}_{n,k} \\
&= \mathbb{E}\left\{Y_1(A)\right\}(1 - \tilde{q}_{n,0}) \\
&= \frac{\nu(A)}{\mu(A)}(1 - \tilde{q}_{n,0}),
\end{aligned} \tag{18}$$

further, by (24)

$$\mathbb{E}\left\{\frac{\nu_n(A)}{\mu_n(A)}\right\} = n\mathbb{E}\left\{\frac{Y_n(A)}{1+(n-1)\mu_{n-1}(A)}\right\} = \frac{\nu(A)}{\mu(A)}(1-(1-\mu(A))^n)), \tag{19}$$

Moreover,

$$\begin{aligned}
\mathbb{E}\left\{\frac{\tilde{\nu}_n(A)^2}{\tilde{\mu}_n(A)^2}\right\} &= \sum_{k=0}^{\infty}\mathbb{E}\left\{\frac{\tilde{\nu}_n(A)^2}{\tilde{\mu}_n(A)^2} \mid n\tilde{\mu}_n(A)=k\right\}\mathbb{P}\{n\tilde{\mu}_n(A)=k\} \\
&= \sum_{k=1}^{\infty}\mathbb{E}\left\{\frac{\left(\sum_{i=1}^k Y_i(A)\right)^2}{k^2}\right\}\tilde{q}_{n,k} \\
&= \sum_{k=1}^{\infty}\frac{k\mathbb{E}\left\{Y_1(A)^2\right\}+k(k-1)\mathbb{E}\left\{Y_1(A)\right\}^2}{k^2}\tilde{q}_{n,k} \\
&= \mathbb{V}ar\left(Y_1(A)\right)\sum_{k=1}^{\infty}\frac{1}{k}\tilde{q}_{n,k}+\mathbb{E}\left\{Y_1(A)\right\}^2(1-\tilde{q}_{n,0}),
\end{aligned}$$

and

$$\begin{aligned}
\sum_{k=1}^{\infty}\frac{1}{k}\tilde{q}_{n,k} &= \sum_{k=1}^{\infty}\frac{1}{k}\frac{(n\mu(A))^k}{k!}e^{-n\mu(A)} \\
&= \sum_{k=1}^{\infty}\frac{1}{k+1}\frac{(n\mu(A))^k}{k!}e^{-n\mu(A)}+\sum_{k=1}^{\infty}\frac{1}{k(k+1)}\frac{(n\mu(A))^k}{k!}e^{-n\mu(A)} \\
&\leq \frac{1}{n\mu(A)}(1-\tilde{q}_{n,0})+\frac{3}{n^2\mu(A)^2}(1-\tilde{q}_{n,0}).
\end{aligned}$$

The independence of the Poisson masses over different cells leads to

$$\begin{aligned}
\mathbb{V}ar(\tilde{M}_n) &= n\sum_{j=1}^{l_n}\nu(A_{n,j})^2\mathbb{V}ar\left(\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right) \\
&\leq n\sum_{j=1}^{l_n}\nu(A_{n,j})^2\Big(\mathbb{V}ar\left(Y_1(A_{n,j})\right)\big(\frac{1}{n\mu(A_{n,j})}(1-e^{-n\mu(A_{n,j})}) \\
&\quad+\frac{3}{n^2\mu(A_{n,j})^2}(1-e^{-n\mu(A_{n,j})})\big) \\
&\quad+\mathbb{E}\left\{Y_1(A_{n,j})\right\}^2(1-e^{-n\mu(A_{n,j})})-\mathbb{E}\left\{Y_1(A_{n,j})\right\}^2(1-e^{-n\mu(A_{n,j})})^2\Big) \\
&\leq \sum_{j=1}^{l_n}\frac{\nu(A_{n,j})^2}{\mu(A_{n,j})^2}\mathbb{V}ar\left(Y_1(A_{n,j})\right)\mu(A_{n,j}) \\
&\quad+\sum_{j=1}^{l_n}\frac{3\mathbb{V}ar\left(Y_1(A_{n,j})\right)\nu(A_{n,j})^2}{n\mu(A_{n,j})^2} \\
&\quad+n\sum_{j=1}^{l_n}\nu(A_{n,j})^2\mathbb{E}\left\{Y_1(A_{n,j})\right\}^2e^{-n\mu(A_{n,j})}\Big)
\end{aligned}$$

such that the bounding error in these inequalities is of order $O(l_n/n)$. (4) together with the boundedness of M_2 and m implies that

$$\begin{aligned}
&\sum_{j=1}^{l_n} \frac{\nu(A_{n,j})^2}{\mu(A_{n,j})^2} \mathbb{V}ar\left(Y_1(A_{n,j})\right) \mu(A_{n,j}) \\
&= \int \frac{\int_{A_n(x)} M_2(z)\mu(dz)}{\mu(A_n(x))} \left(\frac{\int_{A_n(x)} m(z)\mu(dz)}{\mu(A_n(x))}\right)^2 \mu(dx) - \int \left(\frac{\int_{A_n(x)} m(z)\mu(dz)}{\mu(A_n(x))}\right)^4 \mu(dx) \\
&= \sigma_2^2 + o(1),
\end{aligned}$$

where σ_2^2 is defined by (10). Moreover,

$$\sum_{j=1}^{l_n} \frac{3\mathbb{V}ar\left(Y_1(A_{n,j})\right) \nu(A_{n,j})^2}{n\mu(A_{n,j})^2} \leq \frac{3C^{4/3} l_n}{n} \to 0.$$

Then

$$\begin{aligned}
&n \sum_{j=1}^{l_n} \nu(A_{n,j})^2 \mathbb{E}\left\{Y_1(A_{n,j})\right\}^2 e^{-n\mu(A_{n,j})} \\
&= \sum_{j=1}^{l_n} \frac{\nu(A_{n,j})^2}{\mu(A_{n,j})^2} \mathbb{E}\left\{Y_1(A_{n,j})\right\}^2 n\mu(A_{n,j}) e^{-n\mu(A_{n,j})} \mu(A_{n,j}) \\
&\leq C^{4/3} \sum_{j=1}^{l_n} n\mu(A_{n,j})^2 e^{-n\mu(A_{n,j})} \\
&\leq C^{4/3} (\max_{z>0} z^2 e^{-z}) l_n/n \to 0.
\end{aligned}$$

So we proved that

$$\mathbb{V}ar(\tilde{M}_n) \to \sigma_2^2.$$

To complete the asymptotics for $\mathbb{V}ar(T_n)$, it remains to show that

$$\mathbb{E}\left\{\tilde{M}_n \frac{N_n - n}{\sqrt{n}}\right\} \to 0 \text{ as } n \to \infty.$$

Because of

$$N_n = n \sum_{j=1}^{l_n} \tilde{\mu}_n(A_{n,j})$$

and

$$n = n \sum_{j=1}^{l_n} \mu(A_{n,j}),$$

we have that

$$\begin{aligned}
&\mathbb{E}\left\{\tilde{M}_n\frac{N_n-n}{\sqrt{n}}\right\}\\
&=n\sum_{j=1}^{l_n}\mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\nu(A_{n,j})(\tilde{\mu}_n(A_{n,j})-\mu(A_{n,j}))\right\}\\
&=n\sum_{j=1}^{l_n}\nu(A_{n,j})\left(\mathbb{E}\left\{\tilde{\nu}_n(A_{n,j})\right\}-\mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right\}\mu(A_{n,j}))\right)\\
&=n\sum_{j=1}^{l_n}\nu(A_{n,j})\left(\nu(A_{n,j})-\frac{\nu(A_{n,j})}{\mu(A_{n,j})}(1-e^{-n\mu(A_{n,j})})\mu(A_{n,j}))\right)\\
&=n\sum_{j=1}^{l_n}\nu(A_{n,j})^2e^{-n\mu(A_{n,j})}\\
&\le C^{2/3}(\max_{z>0}z^2e^{-z})l_n/n\to 0.
\end{aligned}$$

To finish the proof of (17) by Lyapunov's central limit theorem, it suffices to prove that

$$n^{3/2}\sum_{j=1}^{l_n}\mathbb{E}\Big\{\Big|t\left(\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}-\mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right\}\right)\nu(A_{n,j})+v\left(\tilde{\mu}_n(A_{n,j})-\mu(A_{n,j})\right)\Big|^3\Big\}\to 0$$

or, by invoking the c_3 inequality $|a+b|^3\le 4(|a|^3+|b|^3)$, that

$$n^{3/2}\sum_{j=1}^{l_n}\mathbb{E}\left\{\left|\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}-\mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right\}\right|^3\right\}\nu(A_{n,j})^3\to 0 \tag{20}$$

and

$$n^{3/2}\sum_{j=1}^{l_n}\mathbb{E}\left\{|\tilde{\mu}_n(A_{n,j})-\mu(A_{n,j})|^3\right\}\to 0. \tag{21}$$

In view of (20), because of (13) it suffices to prove

$$D_n:=n^{3/2}\sum_{j=1}^{l_n}\mathbb{E}\left\{\left|\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}-\mathbb{E}\left\{\frac{\tilde{\nu}_n(A_{n,j})}{\tilde{\mu}_n(A_{n,j})}\right\}\right|^3\right\}\mu(A_{n,j})^3\to 0 \tag{22}$$

For a cell A, (18) implies that

$$\begin{aligned}
\mathbb{E}\left\{\left|\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)}-\mathbb{E}\left\{\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)}\right\}\right|^3\right\}&\le 4\mathbb{E}\left\{\left|\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)}-\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\mathbb{I}_{\{\tilde{\mu}_n(A)>0\}}\right|^3\right\}\\
&\quad+4\mathbb{E}\left\{\left|\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\mathbb{I}_{\{\tilde{\mu}_n(A)>0\}}-\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\right|^3\right\}.
\end{aligned}$$

On the one hand, (18), (13) and (25) imply that, for a constant K,

$$\begin{aligned}
&\mathbb{E}\left\{\left|\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)}-\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\mathbb{I}_{\{\tilde{\mu}_n(A)>0\}}\right|^3\right\}\\
&=\sum_{k=0}^{\infty}\mathbb{E}\left\{\left|\frac{\tilde{\nu}_n(A)}{\tilde{\mu}_n(A)}-\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\mathbb{I}_{\{\tilde{\mu}_n(A)>0\}}\right|^3 \mid n\tilde{\mu}_n(A)=k\right\}\mathbb{P}\{n\tilde{\mu}_n(A)=k\}\\
&=\sum_{k=1}^{\infty}\mathbb{E}\left\{\frac{\left|\sum_{i=1}^{k}(Y_i(A)-\mathbb{E}\{Y_i(A)\})\right|^3}{k^3}\right\}\tilde{q}_{n,k}\\
&\le K\sum_{k=1}^{\infty}\frac{1}{k^{3/2}}\tilde{q}_{n,k}\\
&\le c_1\frac{1}{n^{3/2}\mu(A)^{3/2}},
\end{aligned}$$

where we applied the Marcinkiewicz and Zygmund (1937) inequality for absolute central moments of sums of i.i.d. random variables. On the other hand

$$\mathbb{E}\left\{\left|\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\mathbb{I}_{\{\tilde{\mu}_n(A)>0\}}-\frac{\nu(A)}{\mu(A)}(1-\tilde{q}_{n,0})\right|^3\right\}\le C\tilde{q}_{n,0}.$$

Therefore

$$\begin{aligned}
D_n&\le n^{3/2}c_2\sum_{j=1}^{l_n}\left(\frac{1}{n^{3/2}\mu(A_{n,j})^{3/2}}+e^{-n\mu(A_{n,j})}\right)\mu(A_{n,j})^3\\
&\le c_2\left(\sum_{j=1}^{l_n}\mu(A_{n,j})^{3/2}+\sum_{j=1}^{l_n}n^{3/2}e^{-n\mu(A_{n,j})}\mu(A_{n,j})^3\right)\\
&\le c_2\sum_{j=1}^{l_n}\mu(A_{n,j})^{3/2}\left(1+\max_{z>0}z^{3/2}e^{-z}\right)\\
&= c_3\int\mu(A_n(x))^{1/2}\mu(dx)\\
&\to 0,
\end{aligned}$$

where we used the assumption that μ is non-atomic. Thus, (20) is proved.

The proof of (21) is easier. Notice that (21) means

$$F_n:=n^{-3/2}\sum_{j=1}^{l_n}\mathbb{E}\left\{\left|\sum_{i=1}^{N_n}\mathbb{I}_{\{X_i\in A_{n,j}\}}-n\mu(A_{n,j})\right|^3\right\}\to 0.$$

One has

$$\begin{aligned}
&\mathbb{E}\left\{\left|\sum_{i=1}^{N_n}\mathbb{I}_{\{X_i\in A_{n,j}\}}-n\mu(A_{n,j})\right|^3\right\}\\
&\le 4\mathbb{E}\left\{\left|\sum_{i=1}^{N_n}(\mathbb{I}_{\{X_i\in A_{n,j}\}}-\mu(A_{n,j}))\right|^3\right\}+4\mathbb{E}\left\{|(N_n-n)\mu(A_{n,j})|^3\right\}\\
&\le c_4\left(\sum_{k=1}^{\infty}k^{3/2}\mu(A_{n,j})^{3/2}e^{-n}\frac{n^k}{k!}+\mathbb{E}\left\{|N_n-n|^3\right\}\mu(A_{n,j})^3\right)\\
&\le c_5\left(n^{3/2}\mu(A_{n,j})^{3/2}+n^{3/2}\mu(A_{n,j})^3\right).
\end{aligned}$$

Therefore

$$F_n\le 2c_5\sum_{j=1}^{l_n}\mu(A_{n,j})^{3/2}\to 0,$$

and so (21) is proved, too.

The remaining step in the proof of (12) is to show that

$$\Delta_n:=V_n-M_n=n^{1/2}\sum_{j=1}^{l_n}\left(\mathbb{E}\left\{\frac{\tilde\nu_n(A_{n,j})}{\tilde\mu_n(A_{n,j})}\right\}-\mathbb{E}\left\{\frac{\nu_n(A_{n,j})}{\mu_n(A_{n,j})}\right\}\right)\nu(A_{n,j})\to 0. \tag{23}$$

By (18) and (19) have that

$$\begin{aligned}
|\Delta_n|&=\left|n^{1/2}\sum_{j=1}^{l_n}\frac{\nu(A_{n,j})}{\mu(A_{n,j})}(e^{-n\mu(A_{n,j})}-(1-\mu(A_{n,j}))^n)\nu(A_{n,j})\right|\\
&=n^{1/2}\sum_{j=1}^{l_n}\frac{\nu(A_{n,j})^2}{\mu(A_{n,j})^2}(e^{-n\mu(A_{n,j})}-(1-\mu(A_{n,j}))^n)\mu(A_{n,j})\\
&\le C^{2/3}n^{1/2}\sum_{j=1}^{l_n}(e^{-n\mu(A_{n,j})}-(1-\mu(A_{n,j}))^n)\mu(A_{n,j}).
\end{aligned}$$

For $0\le z\le 1$, using the elementary inequalities

$$1-z\le e^{-z}\le 1-z+z^2$$

we have that

$$e^{-nz}-(1-z)^n=(e^{-z}-(1-z))\sum_{k=0}^{n-1}e^{-kz}(1-z)^{n-1-k}\le nz^2e^{-(n-1)z},$$

and thus we get that

$$
\begin{aligned}
|\Delta_n| &\le C^{2/3} n^{1/2} \sum_{j=1}^{l_n} (e^{-n\mu(A_{n,j})} - (1-\mu(A_{n,j}))^n)\mu(A_{n,j}) \\
&\le C^{2/3} n^{1/2} \sum_{j=1}^{l_n} n\mu(A_{n,j})^3 e^{-(n-1)\mu(A_{n,j})} \\
&\le \frac{C^{2/3}}{n^{1/2}} \sum_{j=1}^{l_n} \mu(A_{n,j}) \left([n\mu(A_{n,j})]^2 e^{-n\mu(A_{n,j})} \right) e \\
&\le \frac{C^{2/3}}{n^{1/2}} \sum_{j=1}^{l_n} \mu(A_{n,j}) \max_{z \ge 0} (z^2 e^{-z}) e \\
&\to 0.
\end{aligned}
$$

This ends the proof of (12) and so the proof of Theorem 1 is complete. ■

Next we give two lemmas, which are used above.

Lemma 3 *If $B(n,p)$ is a binomial random variable with parameters (n,p), then*

$$
\mathbb{E}\left\{ \frac{1}{1+B(n,p)} \right\} = \frac{1-(1-p)^{n+1}}{(n+1)p}. \tag{24}
$$

Lemma 4 *If $Po(\lambda)$ is a Poisson random variable with parameter λ, then*

$$
\mathbb{E}\left\{ \frac{1}{Po(\lambda)^3} \mathbb{I}_{\{Po(\lambda)>0\}} \right\} \le \frac{24}{\lambda^3}. \tag{25}
$$

References

J. Beirlant and L. Györfi. On the asymptotic L_2-error in partitioning regression estimation. *Journal of Statistical Planning and Inference*, 71:93–107, 1998.

J. Beirlant and D. Mason. On the asymptotic normality of l_p-norms of empirical functionals. *Mathematical Methods of Statistics*, 4:1–19, 1995.

J. Beirlant, L. Györfi, and G. Lugosi. On the asymptotic normality of the l_1- and l_2- errors in histogram density estimation. *Canadian J. Statistics*, 22:309–318, 1994.

K. De Brabanter, P. G. Ferrario, and L. Györfi. Detecting ineffective features for nonparametric regression. In J. A. K. Suykens, M. Signoretto, and A. Argyriou, editors, *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 177–194. Chapman & Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

L. Devroye, D. Schäfer, L. Györfi, and H. Walk. The estimation problem of minimum mean squared error. *Statistics and Decisions*, 21:15–28, 2003.

L. Devroye, P. Ferrario, L. Györfi, and H. Walk. Strong universal consistent estimate of the minimum mean squared error. In B. Schölkopf, Z. Luo, and V. Vovk, editors, *Empirical Inference - Festschrift in Honor of Vladimir N. Vapnik*, pages 143–160. Springer, Heidelberg, 2013.

D. Evans and A. J. Jones. Non-parametric estimation of residual moments and covariance. *Proceedings of the Royal Society*, A 464:2831–2846, 2008.

P. G. Ferrario and H. Walk. Nonparametric partitioning estimation of residual and local variance based on first and second nearest neighbors. *Journal of Nonparametric Statistics*, 24:1019–1039, 2012.

L. Györfi, M. Kohler, A. Krzyżak, and H. Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer–Verlag, New York, 2002.

E. Liitiäinen, F. Corona, and A. Lendasse. On nonparametric residual variance estimation. *Neural Processing Letters*, 28:155–167, 2008.

E. Liitiäinen, M. Verleysen, F. Corona, and A. Lendasse. Residual variance estimation in machine learning. *Neurocomputing*, 72:3692–3703, 2009.

E. Liitiäinen, F. Corona, and A. Lendasse. Residual variance estimation using a nearest neighbor statistic. *Journal of Multivariate Analysis*, 101:811–823, 2010.

J. Marcinkiewicz and A. Zygmund. Sur les fonctions indépendantes. *Fundamenta Mathematicae*, 29:60–90, 1937.

V. V. Petrov. *Sums of Independent Random Variables*. Springer-Verlag, Berlin, 1975.

Journal of Machine Learning Research 16 (2015) 1879-1892 Submitted 6/15; Published 9/15

Sharp Oracle Bounds for Monotone and Convex Regression Through Aggregation

Pierre C. Bellec PIERRE.BELLEC@ENSAE.FR
Alexandre B. Tsybakov ALEXANDRE.TSYBAKOV@ENSAE.FR
ENSAE,
3 avenue Pierre Larousse
92240 Malakoff, France

Editor: Alex Gammerman and Vladimir Vovk

Abstract

We derive oracle inequalities for the problems of isotonic and convex regression using the combination of Q-aggregation procedure and sparsity pattern aggregation. This improves upon the previous results including the oracle inequalities for the constrained least squares estimator. One of the improvements is that our oracle inequalities are sharp, i.e., with leading constant 1. It allows us to obtain bounds for the minimax regret thus accounting for model misspecification, which was not possible based on the previous results. Another improvement is that we obtain oracle inequalities both with high probability and in expectation.

Keywords: aggregation, shape constraints, isotonic regression, convex regression, minimax regret, sharp oracle inequalities, model misspecification

1. Introduction

Assume that we have the observations

$$Y_i = \mu_i + \xi_i, \qquad i = 1, ..., n, \tag{1}$$

where $\boldsymbol{\mu} = (\mu_1, ..., \mu_n)^T \in \mathbb{R}^n$ is unknown, $\boldsymbol{\xi} = (\xi_1, ..., \xi_n)^T$ is a noise vector with n-dimensional Gaussian distribution $\mathcal{N}(0, \sigma^2 I_{n\times n})$ where $\sigma > 0$. We observe $\mathbf{y} = (Y_1, ..., Y_n)^T$ and we want to estimate $\boldsymbol{\mu}$. We can interpret μ_i as the values $f(X_i)$ of an unknown regression function $f : \mathcal{X} \to \mathbb{R}$ at given non-random points $X_i \in \mathcal{X}$, $i = 1, \dots, n$, where $\mathcal{X}$ is an abstract set. Then, the equivalent setting is that we observe $\mathbf{y}$ along with $(X_1, \dots, X_n)$ but the values of X_i are of no interest and can be replaced by their indices if we measure the loss in a discrete norm. Namely, for any $\boldsymbol{u} \in \mathbb{R}^n$ we consider the scaled (or the empirical) norm $\|\cdot\|$ defined by

$$\|\boldsymbol{u}\|^2 = \frac{1}{n}\sum_{i=1}^{n} u_i^2. \tag{2}$$

We will measure the error of an estimator $\hat{\boldsymbol{\mu}}$ of $\boldsymbol{\mu}$ by the distance $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|$. Let $\mathcal{S}^{\uparrow}$ be the set of all non-decreasing sequences:

$$\mathcal{S}^{\uparrow} := \{\boldsymbol{u} = (u_1, ..., u_n) \in \mathbb{R}^n : u_i \le u_{i+1}, \quad i = 1, ..., n-1\}. \tag{3}$$

For a subset $\mathcal{S}$ of $\mathcal{S}^{\uparrow}$, and any $\boldsymbol{\mu} \in \mathbb{R}^n$ the quantity $\min_{\boldsymbol{u}\in\mathcal{S}} \|\boldsymbol{u} - \boldsymbol{\mu}\|$ is the smallest approximation error achievable by a sequence in the set $\mathcal{S}$. This quantity defines a benchmark

or oracle performance on $\mathcal{S}$. The accuracy of an estimator $\hat{\boldsymbol{\mu}}$ with respect to the oracle for any $\boldsymbol{\mu}$, not necessarily $\boldsymbol{\mu} \in \mathcal{S}$, can be characterized by the excess loss $\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| - \min_{\boldsymbol{u}\in\mathcal{S}} \|\boldsymbol{u} - \boldsymbol{\mu}\|$. This is a measure of performance of $\boldsymbol{\mu}$ under model misspecification. One can also consider the expected quantities $R_1(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \mathbb{E}_{\boldsymbol{\mu}}\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| - \min_{\boldsymbol{u}\in\mathcal{S}} \|\boldsymbol{u} - \boldsymbol{\mu}\|$ or $R_2(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = \mathbb{E}_{\boldsymbol{\mu}}\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 - \min_{\boldsymbol{u}\in\mathcal{S}} \|\boldsymbol{u} - \boldsymbol{\mu}\|^2$ known under the name of regret measures. Here, $\mathbb{E}_{\boldsymbol{\mu}}$ denotes the expectation with respect to the distribution of $\mathbf{y}$ satisfying (1). The minimax regret is defined as $\min_{\hat{\boldsymbol{\mu}}} \max_{\boldsymbol{\mu}\in\mathbb{R}^n} R_i(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu})$ for $i = 1, 2$, where $\min_{\hat{\boldsymbol{\mu}}}$ denotes the minimum over all estimators. We can characterize the performance of an estimator $\tilde{\boldsymbol{\mu}}$ by the closeness of its maximal regret $\max_{\boldsymbol{\mu}\in\mathbb{R}^n} R_i(\tilde{\boldsymbol{\mu}}, \boldsymbol{\mu})$ to the minimax regret. This approach to measure the performance of estimators under model misspecification was pioneered by Vapnik and Chervonenkis who called it the criterion of minimax of the loss (Vapnik and Chervonenkis, 1974, Chapter 6). In this paper, we follow this approach and establish non-asymptotic bounds for the maximal regret for some classes $\mathcal{S}$ of monotone and convex functions.

When the model is well-specified, i.e., the true function $\boldsymbol{\mu}$ belongs to the class $\mathcal{S}$, the approximation error vanishes and instead of the minimax regret it is natural to consider the minimax risk defined either as $\min_{\hat{\boldsymbol{\mu}}} \max_{\boldsymbol{\mu}\in\mathcal{S}} \mathbb{E}_{\boldsymbol{\mu}}\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|$ or as $\min_{\hat{\boldsymbol{\mu}}} \max_{\boldsymbol{\mu}\in\mathcal{S}} \mathbb{E}_{\boldsymbol{\mu}}\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2$ (the minimax squared risk). It is easy to see that the minimax risk is not greater than the minimax regret. A classical problem in nonparametric statistics is to study the behavior of minimax risks for different classes $\mathcal{S}$. In particular, there exist results concerning the minimax risks for classes of monotone and convex functions in our setting. We review some of them below. The behavior of the minimax regret is much less studied. For a recent overview and some general results we refer to Rakhlin et al. (2013) where it is shown that the rate of minimax regret can be different from that of the minimax risk. Note that Rakhlin et al. (2013) studies the prediction problem with i.i.d. observations, which is a setting different from ours.

A well-studied estimator under the monotonicity and convexity assumptions is the least squares estimator

$$\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}) \in \operatorname*{argmin}_{\boldsymbol{u}\in\mathcal{S}} \|\mathbf{y} - \boldsymbol{u}\|^2 . \tag{4}$$

In Nemirovski et al. (1985) it was shown that $\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S})$ attains, up to logarithmic factors, the rates $n^{-2/3}$ and $n^{-4/5}$ of the mean squared risk for classes $\mathcal{S}$ of monotone and convex functions respectively and that these rates are optimal up to logarithmic factors when the minimax squared risk is used as a criterion. Under monotonicity constraints, the rate $n^{-2/3}$ was later observed in different settings, see for instance Banerjee and Wellner (2001); Balabdaoui and Wellner (2007).

One class of monotone functions we will be interested in here is defined as

$$\mathcal{S}^{\uparrow}(V) = \{\boldsymbol{\mu} \in \mathcal{S}^{\uparrow} : \ V(\boldsymbol{\mu}) \le V\}$$

where $V(\boldsymbol{\mu}) = \mu_n - \mu_1$ for any $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n) \in \mathcal{S}^{\uparrow}$, and $V > 0$ is a given constant. In Meyer and Woodroofe (2000); Zhang (2002) it was shown that for any $\boldsymbol{\mu} \in \mathcal{S}^{\uparrow}$ we have

$$\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 \le c \max\left(\left(\frac{\sigma^2 V(\boldsymbol{\mu})}{n} \right)^{2/3}, \frac{\sigma^2 \log n}{n} \right) \tag{5}$$

for $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\uparrow})$ and some absolute constant $c > 0$. This immediately implies an upper bound on the minimax risk on $\mathcal{S}^{\uparrow}(V)$. A recent paper Chatterjee et al. (2015) establishes the oracle inequality

$$\mathbb{E}_{\boldsymbol{\mu}} \left\| \hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\uparrow}) - \boldsymbol{\mu} \right\|^2 \leq C_* \min_{\boldsymbol{u} \in \mathcal{S}^{\uparrow}} \left(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c_* \sigma^2 k(\boldsymbol{u})}{n} \log \frac{en}{k(\boldsymbol{u})} \right) \tag{6}$$

valid for all $\boldsymbol{\mu} \in \mathcal{S}^{\uparrow}$ where either $C_* = 6, c_* = 1$ (Chatterjee et al., 2015, inequality (18)) or $C_* = 4, c_* = 4$ (Chatterjee et al., 2015, inequality (30)). Here, $k(\boldsymbol{u}) \geq 1$ for $\boldsymbol{u} = (u_1, \dots, u_n) \in \mathcal{S}^{\uparrow}$ is the integer such that $k(\boldsymbol{u}) - 1$ is the number of inequalities $u_i \leq u_{i+1}$ that are strict for $i = 1, \dots, n-1$ (number of jumps of $\boldsymbol{u}$). Inequality (6) implies (up to a logarithmic factor) a bound as in (5) and also gives some more insight into the problem. For example, (6) shows that the fast rate $\frac{\log n}{n}$ is achieved if $\boldsymbol{\mu}$ has only one jump or a fixed, independent of n, number of jumps. This is not granted by (5).

Along with the least squares estimator, one may consider estimation of monotone functions via penalized least squares with total variation penalty. The corresponding estimator $\hat{\boldsymbol{\mu}}^{TV}$ is defined as

$$\hat{\boldsymbol{\mu}}^{TV} \in \operatorname*{argmin}_{\boldsymbol{u} \in \mathbb{R}^n} \left(\frac{1}{2} \|\boldsymbol{u} - \mathbf{y}\|^2 + \lambda \sum_{i=1}^{n-1} |u_{i+1} - u_i| \right), \tag{7}$$

where $\lambda > 0$ is a tuning parameter. Statistical properties of this estimator were first studied in Mammen and van de Geer (1997) where it was shown that $\|\hat{\boldsymbol{\mu}}^{TV} - \boldsymbol{\mu}\|$ attains the optimal rate $n^{-1/3}$ in probability on the class of functions of bounded variation (and thus on $\mathcal{S}^{\uparrow}(V)$). Recently, the performance of $\hat{\boldsymbol{\mu}}^{TV}$ was analyzed in Dalalyan et al. (2014) by considering $\hat{\boldsymbol{\mu}}^{TV}$ as a special instance of the Lasso estimator. If $\boldsymbol{\mu}^{\uparrow}$ is the projection of $\boldsymbol{\mu}$ onto $\mathcal{S}^{\uparrow}$, $\delta \in (0,1)$ is a constant, and the tuning parameter λ is given by

$$\lambda = \sigma \sqrt{\frac{\log(n/\delta)}{k^* n}} \qquad \text{where } k^* = \left(\frac{V(\boldsymbol{\mu}^{\uparrow})^2 n \log(n/\delta)}{\sigma^2} \right)^{1/3}, \tag{8}$$

the estimator $\hat{\boldsymbol{\mu}}^{TV}$ satisfies with probability greater than $1 - 2\delta$ the following oracle inequality (Dalalyan et al., 2014, Proposition 6):

$$\begin{aligned} \left\| \hat{\boldsymbol{\mu}}^{TV} - \boldsymbol{\mu} \right\|^2 \;\leq\; & \left\| \boldsymbol{\mu}^{\uparrow} - \boldsymbol{\mu} \right\|^2 + 6 \left(\frac{\sigma^2 V(\boldsymbol{\mu}^{\uparrow}) \sqrt{\log(n/\delta)}}{n} \right)^{2/3} \\ & + \frac{2\sigma^2 (1 + 2\log(1/\delta))}{n} \end{aligned} \tag{9}$$

for all $\boldsymbol{\mu} \in \mathbb{R}^n$. It follows from (9) that if the tuning parameter is chosen correctly, the estimator $\hat{\boldsymbol{\mu}}^{TV}$ achieves, up to a logarithmic factor, the minimax rate $n^{-2/3}$ in probability on the class $\mathcal{S}^{\uparrow}(V)$. Also, (9) implies a bound for the excess losses $\|\hat{\boldsymbol{\mu}}^{TV} - \boldsymbol{\mu}\|^i - \min_{\boldsymbol{u} \in \mathcal{S}^{\uparrow}(V)} \|\boldsymbol{u} - \boldsymbol{\mu}\|^i$, $i = 1, 2$, corresponding to the class $\mathcal{S}^{\uparrow}(V)$. However, (9) does not allow us to evaluate the expected regrets $R_i(\hat{\boldsymbol{\mu}}^{TV}, \boldsymbol{\mu})$ since $\hat{\boldsymbol{\mu}}^{TV}$ depends on δ. It is also shown in (Dalalyan et al., 2014, Proposition 4) that if $\lambda = 2\sigma\sqrt{(2/n)\log(n/\delta)}$, the estimator $\hat{\boldsymbol{\mu}}^{TV}$ satisfies

$$\left\| \hat{\boldsymbol{\mu}}^{TV} - \boldsymbol{\mu} \right\|^2 \leq \min_{\boldsymbol{u} \in \mathbb{R}^n} \left(\|\boldsymbol{u} - \boldsymbol{\mu}\|^2 + \frac{4\sigma^2 k(\boldsymbol{u}) \log(n/\delta)}{n} r_n(\boldsymbol{u}) \right) \tag{10}$$

with probability greater than $1-2\delta$, where $k(\boldsymbol{u})-1$ for $\boldsymbol{u} \in \mathbb{R}^n$ is the number of jumps of $\boldsymbol{u}$, i.e., the cardinality of the set $\{i \in \{1, ..., n-1\} : u_i \neq u_{i+1}\}$, $r_n(\boldsymbol{u}) = 3 + 256(\log(n) + (n/\Delta(\boldsymbol{u})))$ and $\Delta(\boldsymbol{u})$ is the minimum distance between two jumps in the sequence $\boldsymbol{u}$:

$$\Delta(\boldsymbol{u}) = \min\{d \geq 1 : \ \exists k \in \{1, ..., n\} \text{ with } u_{k+1} \neq u_k \text{ and } u_{k+d+1} \neq u_{k+d}\}.$$

The expressions on the right hand sides of (6) and (10) are small if the unknown sequence $\boldsymbol{\mu}$ is well approximated by a piecewise constant sequence with not too many pieces. In this regard, the two bounds have some similarity to sparsity oracle inequalities in high-dimensional linear regression (cf. Rigollet and Tsybakov, 2011, 2012; Tsybakov, 2014). This similarity can be easily explained as follows. Write (1) in the equivalent form

$$\mathbf{y} = \mathbb{X}\boldsymbol{\beta}^* + \boldsymbol{\xi},$$

with the matrix $\mathbb{X} = (X_{ij})_{i=1,...,n,\ j=1,...,n}$ where $X_{ij} = 1$ if $j \leq i$ and $X_{ij} = 0$ otherwise, and $\boldsymbol{\beta}^* = (\beta_1^*, \dots, \beta_n^*)$ where $\beta_1^* = \mu_1$ and $\beta_i^* = \mu_i - \mu_{i-1}$ for $i = 2, \dots, n$. With this notation, $k(\boldsymbol{\mu}) \in \{|\boldsymbol{\beta}^*|_0, 1 + |\boldsymbol{\beta}^*|_0\}$, where $|\boldsymbol{\beta}^*|_0$ denotes the number of non-zero components of $\boldsymbol{\beta}^*$. The value $k(\boldsymbol{\mu})$ is small when $\boldsymbol{\beta}^*$ is sparse. Thus, the problem of estimation of piecewise constant sequence $\boldsymbol{\mu}$ with small number of pieces can be considered as the problem of prediction in sparse linear regression with a specific design matrix $\mathbb{X}$. Similarly, we may write $\boldsymbol{u} = \mathbb{X}\boldsymbol{\beta}$, for $\boldsymbol{\beta}$ with components $\beta_1 = u_1$ and $\beta_i = u_i - u_{i-1}$ for $i = 2, \dots, n$. These remarks suggest that we can apply the theory of sparsity oracle inequalities, in particular, sparsity pattern aggregation (cf. Rigollet and Tsybakov, 2011, 2012; Tsybakov, 2014) in the context of monotone estimation described above. Similar observation is valid for estimation under convexity constraints (see Section 3 below). In the present paper, we develop this argument using as a building block the Q-aggregation procedures Rigollet (2012); Dai et al. (2012, 2014); Bellec (2014). In particular, we construct an estimator $\hat{\boldsymbol{\mu}}$ such that

$$\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 \leq \min_{\boldsymbol{u} \in \mathcal{S}^{\uparrow}} \left(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c\sigma^2 k(\boldsymbol{u})}{n} \log \frac{en}{k(\boldsymbol{u})} \right), \quad \forall\ \boldsymbol{\mu} \in \mathbb{R}^n, \tag{11}$$

for some absolute constant $c > 0$. Note that (11) is a sharp oracle inequality (i.e., an inequality with leading constant 1). It improves upon the oracle inequality (6) for the least squares estimator where the leading constant C_* is noticeably greater than 1 and the bound is valid only for $\boldsymbol{\mu} \in \mathcal{S}^{\uparrow}$. The advantage of having leading constant 1 and arbitrary $\boldsymbol{\mu}$ in (11) is that it allows us to derive bounds on the excess risk and on the minimax regret, which was not possible based on the previous results. We also obtain sharp oracle inequalities with high probability for the same estimator. In addition, we show that it satisfies stronger sharp inequalities with the minimum $\min_{\boldsymbol{u} \in \mathcal{S}^{\uparrow}}$ on the right hand side of (11) replaced by $\min_{\boldsymbol{u} \in \mathbb{R}^n}$. This implies that our results are invariant to the direction of monotonicity; they remain valid if we replace everywhere monotone increasing by monotone decreasing functions. Finally, we derive similar results for the problem of estimation under the convexity constraints improving an oracle inequality obtained in Guntuboyina and Sen (2013).

2. Sparsity Pattern Aggregation for Piecewise Constant Sequences

For any non-empty set $J \subseteq \{1, ..., n-1\}$, let $|J|$ denote the cardinality of J and define

$$\pi_J \coloneqq \frac{\exp(-|J|)}{H \binom{n-1}{|J|}}, \qquad H \coloneqq \sum_{i=0}^{n-1} \exp(-i). \tag{12}$$

Let $P_J \in \mathbb{R}^{n\times n}$ be the projector on the linear subspace V_J of $\mathbb{R}^n$ defined by

$$V_J \coloneqq \Big\{ \boldsymbol{u} \in \mathbb{R}^n : \ \forall i \in \{1, ..., n-1\} \setminus J, \ u_{i+1} = u_i \Big\}. \tag{13}$$

In words, V_J is the space of all piecewise constant sequences that have jumps only at points in J. Given a vector $\mathbf{y}$ of observations and $\boldsymbol{\theta} = (\theta_J)_{J\subseteq\{1,...,n-1\}}$ where each $\theta_J \in \mathbb{R}$, let

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} = \sum_{J\subseteq\{1,...,n-1\}} \theta_J P_J \mathbf{y}. \tag{14}$$

Finally, let

$$\hat{\boldsymbol{\mu}}^Q = \boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}} \tag{15}$$

where $\hat{\boldsymbol{\theta}}$ is the solution of the optimization problem

$$\min_{\boldsymbol{\theta}\in\Lambda} \quad \|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \mathbf{y}\|^2 + \sum_{J\subseteq\{1,...,n-1\}} \theta_J \left(\frac{2\sigma^2|J|}{n} + \frac{1}{2}\|\boldsymbol{\mu}_{\boldsymbol{\theta}} - P_J\mathbf{y}\|^2 + \frac{46\sigma^2}{n} \log\frac{1}{\pi_J} \right)$$

where

$$\Lambda = \left\{ \boldsymbol{\theta} : \ \theta_J \geq 0 \text{ for all } J \subseteq \{1, ..., n-1\}, \text{ and } \sum_{J\subseteq\{1,...,n-1\}} \theta_J = 1 \right\}.$$

This optimization problem is a convex quadratic program with a simplex constraint. It performs aggregation of the linear estimators $(P_J\mathbf{y})_{J\subseteq\{1,...,n-1\}}$ using the Q-aggregation procedure Dai et al. (2012, 2014); Bellec (2014) with the prior weights (12). As the size of this quadratic program is of order 2^n, it is a computationally hard problem. The estimator $\hat{\boldsymbol{\mu}}^Q$ satisfies the following sharp oracle inequalities.

Theorem 1 *Let $\boldsymbol{\mu} \in \mathbb{R}^n$, $n \geq 2$, and assume that the noise vector $\boldsymbol{\xi}$ has distribution $\mathcal{N}(0, \sigma^2 I_{n\times n})$. There exist absolute constants $c, c' > 0$ such that for all $\delta \in (0, 1/3)$, the estimator $\hat{\boldsymbol{\mu}}^Q$ satisfies with probability at least $1 - 3\delta$,*

$$\left\|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\right\|^2 \leq \min_{\boldsymbol{u}\in\mathbb{R}^n} \left(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c\sigma^2 k(\boldsymbol{u}))}{n} \log\frac{en}{k(\boldsymbol{u})} \right) + \frac{c\sigma^2 \log(1/\delta)}{n}, \tag{16}$$

and

$$\mathbb{E}_{\boldsymbol{\mu}} \left\|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\right\|^2 \leq \min_{\boldsymbol{u}\in\mathbb{R}^n} \left(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c'\sigma^2 k(\boldsymbol{u})}{n} \log\frac{en}{k(\boldsymbol{u})} \right). \tag{17}$$

Proof Let $J \subseteq \{1, ..., n-1\}$. Denote by $d = |J| + 1$ the dimension of the subspace V_J. Then, the projection estimator $P_J \mathbf{y}$ satisfies with probability at least $1 - \delta$ (see, for example, Hsu et al. (2012)):

$$\begin{aligned} \|P_J \mathbf{y} - \boldsymbol{\mu}\|^2 &\le \|P_J \boldsymbol{\mu} - \boldsymbol{\mu}\|^2 + \frac{d + 2\sqrt{d \log(1/\delta)} + 2\log(1/\delta)}{n} \\ &\le \min_{\boldsymbol{u} \in V_J} \|\boldsymbol{u} - \boldsymbol{\mu}\|^2 + \frac{2(|J|+1) + 3\log(1/\delta)}{n}. \end{aligned} \tag{18}$$

The sharp oracle inequality from Bellec (2014) yields that with probability at least $1 - 2\delta$ for all $J \subseteq \{1, ..., n-1\}$ we have

$$\left\| \hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu} \right\|^2 \le \|P_J \mathbf{y} - \boldsymbol{\mu}\|^2 + C\sigma^2 \log \frac{1}{\pi_J} + C\sigma^2 \log(1/\delta), \tag{19}$$

for some absolute constant $C > 0$. Combining (18) and (19) with the union bound and the inequality (cf. (Rigollet and Tsybakov, 2012, (5.4))) $\log(1/\pi_J) \le 2(|J|+1)\log(en/(|J|+1)) + 1/2$, we find that with probability at least $1 - 3\delta$,

$$\begin{aligned} \left\| \hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu} \right\|^2 \le \min_{J \subseteq \{1,...,n-1\}} \min_{\boldsymbol{u} \in V_J} \Bigg(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c\sigma^2(|J|+1)}{n} \log\left(\frac{en}{|J|+1}\right) \Bigg) \\ + c\sigma^2 \log(1/\delta) \end{aligned}$$

where $c > 0$ is an absolute constant. Since we have that $|J| + 1 = k(\boldsymbol{u})$ for all $\boldsymbol{u} \in V_J$ and also that $\min_{J \subseteq \{1,...,n-1\}} \min_{\boldsymbol{u} \in V_J} = \min_{\boldsymbol{u} \in \mathbb{R}^n}$, the bound (16) follows. Finally, (17) is obtained from (16) by integration. ■

We now discuss some corollaries of Theorem 1. First, it follows that (11) is satisfied for $\hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}^Q$, so the remarks after (11) apply. Next, in view of (17), for the class of monotone sequences with at most k jumps $\mathcal{S}_k^{\uparrow} = \{\boldsymbol{u} \in \mathcal{S}^{\uparrow} : k(\boldsymbol{u}) \le k\}$ we have the following bounds for the maximal expected regrets

$$\max_{\boldsymbol{\mu} \in \mathbb{R}^n} \left(\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\| - \min_{\boldsymbol{u} \in \mathcal{S}_k^{\uparrow}} \|\boldsymbol{u} - \boldsymbol{\mu}\| \right) \le c\sqrt{\frac{\sigma^2 k}{n} \log\left(\frac{en}{k}\right)}, \tag{20}$$

$$\max_{\boldsymbol{\mu} \in \mathbb{R}^n} \left(\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\|^2 - \min_{\boldsymbol{u} \in \mathcal{S}_k^{\uparrow}} \|\boldsymbol{u} - \boldsymbol{\mu}\|^2 \right) \le \frac{c\sigma^2 k}{n} \log\left(\frac{en}{k}\right), \tag{21}$$

where $c > 0$ is an absolute constant. The same bounds hold for the minimax risks over $\mathcal{S}_k^{\uparrow}$ since the minimax risk is smaller than the minimax regret. Theorem 4 below shows that the bounds (20) and (21) are optimal up to logarithmic factors.

Finally, consider the consequences of Theorem 1 for the class $\mathcal{S}^{\uparrow}(V)$. To this end, define the integer k^* such that

$$k^* = \min \left\{ m \in \mathbb{N} : m \ge \left(\frac{V(\boldsymbol{\mu})^2 n}{\sigma^2 \log(en)} \right)^{1/3} \right\}$$

if the set $\left\{m \in \mathbb{N} : m \geq \left(\frac{V(\boldsymbol{\mu})^2 n}{\sigma^2 \log(en)}\right)^{1/3}\right\}$ is non-empty, and $k^* = 1$ otherwise. We will need the following lemma.

Lemma 2 *Let $\boldsymbol{\mu} \in \mathcal{S}^{\uparrow}$ and let $1 \leq k \leq n$ be an integer. Then there exists a sequence $\bar{\boldsymbol{u}} \in \mathcal{S}_k^{\uparrow}$ such that*

$$\|\bar{\boldsymbol{u}} - \boldsymbol{\mu}\| \leq \frac{V(\boldsymbol{\mu})}{2k}. \tag{22}$$

Next, there exists a sequence $\bar{\boldsymbol{u}} \in \mathcal{S}_{k^*}^{\uparrow}$ such that

$$\|\bar{\boldsymbol{u}} - \boldsymbol{\mu}\|^2 \leq \frac{1}{4} \max\left(\left(\frac{\sigma^2 V(\boldsymbol{\mu}) \log(en)}{n}\right)^{2/3}, \frac{\sigma^2 \log(en)}{n}\right). \tag{23}$$

In addition,

$$\frac{\sigma^2 k^*}{n} \log \frac{en}{k^*} \leq 2 \max\left(\left(\frac{\sigma^2 V(\boldsymbol{\mu}) \log(en)}{n}\right)^{2/3}, \frac{\sigma^2 \log(en)}{n}\right). \tag{24}$$

Proof To construct the sequence $\bar{\boldsymbol{u}}$, consider the k intervals

$$I_j = \Big[\mu_1 + \frac{j-1}{k} V(\boldsymbol{\mu}), \mu_1 + \frac{j}{k} V(\boldsymbol{\mu})\Big), \qquad j = 1, ..., k-1, \tag{25}$$

and $I_k = [\mu_1 + \frac{k-1}{k} V(\boldsymbol{\mu}), \mu_n]$. For all $j = 1, ..., k$, let

$$J_j = \{i = 1, ..., n : \quad \mu_i \in I_j\}. \tag{26}$$

For any $i \in \{1, ..., n\}$ there exists a unique $j \in \{1, ..., k\}$ such that $i \in I_j$. Let $\bar{u}_i = \mu_1 + \frac{j-1/2}{k} V(\boldsymbol{\mu})$ for all $i \in I_j$. Then the sequence $\bar{\boldsymbol{u}} = (\bar{u}_1, \ldots, \bar{u}_n)$ is non-decreasing, it has at most k pieces, i.e., $k(\bar{\boldsymbol{u}}) \leq k$, and $|\bar{u}_i - \mu_i| \leq \frac{V(\boldsymbol{\mu})}{2k}$ for $i = 1, ..., n$. Thus (22) follows. Next, note that if $k^* = 1$, then $V(\boldsymbol{\mu})^2 \leq \sigma^2 \log(en)/n$. If $k^* > 1$, then by definition of k^*, $V(\boldsymbol{\mu})^2/(k^*)^2 \leq (\sigma^2 V(\boldsymbol{\mu}) \log(en)/n)^{2/3}$. Thus, (23) follows. The bound (24) is straightforward by studying the cases $k^* = 1$ and $k^* > 1$ separately. ■

We can now derive the following corollary of Theorem 1.

Corollary 3 *Under the assumptions of Theorem 1, there exists an absolute constant $c > 0$ such that, for any $\boldsymbol{\mu} \in \mathcal{S}^{\uparrow}$,*

$$\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\|^2 \leq c \max\left(\left(\frac{\sigma^2 V(\boldsymbol{\mu}) \log n}{n}\right)^{2/3}, \frac{\sigma^2 \log n}{n}\right). \tag{27}$$

In addition, for any $V > 0$ and any $\boldsymbol{\mu} \in \mathbb{R}^n$ the expected regret of $\hat{\boldsymbol{\mu}}^Q$ satisfies

$$\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\| - \min_{\boldsymbol{u} \in \mathcal{S}^{\uparrow}(V)} \|\boldsymbol{u} - \boldsymbol{\mu}\| \ \leq c \max\left(\left(\frac{\sigma^2 V \log n}{n}\right)^{1/3}, \sigma\sqrt{\frac{\log n}{n}}\right) \tag{28}$$

where $c > 0$ is an absolute constant.

Proof Inequality (27) is straightforward in view of (17), (23), and (24). To prove (28), fix any $\boldsymbol{\mu} \in \mathbb{R}^n$ and consider

$$\boldsymbol{\mu}^* \in \operatorname*{argmin}_{\boldsymbol{\mu}' \in \mathcal{S}^\uparrow(V)} \|\boldsymbol{\mu}' - \boldsymbol{\mu}\|.$$

From (17) and the fact that the function $x \mapsto x \log\left(\frac{en}{x}\right)$ is increasing for $1 \le x \le n$ we get

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\| &\le \min_{\boldsymbol{u} \in \mathcal{S}^\uparrow_{k^*}} \left(\|\boldsymbol{u} - \boldsymbol{\mu}\| + \sqrt{c' \frac{\sigma^2 k^*}{n} \log\left(\frac{en}{k^*}\right)} \right) \\
&\le \min_{\boldsymbol{u} \in \mathcal{S}^\uparrow_{k^*}} \|\boldsymbol{u} - \boldsymbol{\mu}^*\| + \|\boldsymbol{\mu}^* - \boldsymbol{\mu}\| + \sqrt{c' \frac{\sigma^2 k^*}{n} \log\left(\frac{en}{k^*}\right)} \\
&\le \|\boldsymbol{\mu}^* - \boldsymbol{\mu}\| + c'' \max\left(\left(\frac{\sigma^2 V \log n}{n} \right)^{1/3}, \sigma \sqrt{\frac{\log n}{n}} \right)
\end{aligned}$$

for an absolute constant $c'' > 0$ where the last inequality follows from (23) and (24). ■

The estimator $\hat{\boldsymbol{\mu}}^Q$ shown in Theorem 1 satisfies the sharp oracle inequalities both in expectation and with high probability. Previous results for the least squares estimator Chatterjee et al. (2015) were only obtained in expectation and the results on the ℓ_1-penalized estimator (7) are only obtained with high probability.

Finally, the following result shows that the upper bounds (20) and (21) are optimal up to logarithmic factors.

Proposition 4 *Let $n \ge 2, V > 0$ and $\sigma > 0$. There exist absolute constants $c, c' > 0$ such that for any positive integer $k \le n$ satisfying $k^3 \le 16nV^2/\sigma^2$ we have*

$$\inf_{\hat{\boldsymbol{\mu}}} \sup_{\boldsymbol{\mu} \in \mathcal{S}^\uparrow_k \cap \mathcal{S}^\uparrow(V)} \mathbb{P}_{\boldsymbol{\mu}} \left(\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 \ge \frac{c\sigma^2 k}{n} \right) > c', \tag{29}$$

where $\mathbb{P}_{\boldsymbol{\mu}}$ denotes the distribution of $\mathbf{y}$ satisfying (1) and $\inf_{\hat{\boldsymbol{\mu}}}$ is the infimum over all estimators.

For $k = 1, ..., n$, take any $V > 0$ large enough to satisfy $k^3 \le 16nV^2/\sigma^2$. Then, Theorem 4 and Markov's inequality yield the following lower bounds on the minimax risks over the class $\mathcal{S}^\uparrow_k$:

$$\inf_{\hat{\boldsymbol{\mu}}} \sup_{\boldsymbol{\mu} \in \mathcal{S}^\uparrow_k} \mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\| \ge c \sqrt{\frac{c'\sigma^2 k}{n}}, \quad \inf_{\hat{\boldsymbol{\mu}}} \sup_{\boldsymbol{\mu} \in \mathcal{S}^\uparrow_k} \mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 \ge \frac{cc'\sigma^2 k}{n}. \tag{30}$$

As the minimax risk is smaller than the minimax regret, (30) also provides lower bounds for the corresponding minimax regrets over $\mathcal{S}^\uparrow_k$. Combining this with (20) and (21) we find that the estimator $\hat{\boldsymbol{\mu}}^Q$ achieves up to logarithmic factors the optimal rate with respect to the minimax regret.

Next, Proposition 4 implies the following lower bound on the minimax deviation risk on $\mathcal{S}^\uparrow(V)$.

Corollary 5 *Let $n \geq 2, V > 0$ and $\sigma > 0$. There exist absolute constants $c, c' > 0$ such that*

$$\inf_{\hat{\boldsymbol{\mu}}} \sup_{\boldsymbol{\mu} \in \mathcal{S}^{\uparrow}(V)} \mathbb{P}_{\boldsymbol{\mu}} \left(\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 \geq c \max \left\{ \left(\frac{\sigma^2 V}{n} \right)^{2/3}, \frac{\sigma^2}{n} \right\} \right) > c'. \tag{31}$$

To prove this corollary it is enough to note that if $16nV^2/\sigma^2 \geq 1$, by choosing k in Proposition 4 as the integer part of $(16nV^2/\sigma^2)^{1/3}$, we obtain the lower bound corresponding to $\left(\frac{\sigma^2 V}{n}\right)^{2/3}$ under the maximum in (31). On the other hand, if $16nV^2/\sigma^2 < 1$ the term $\frac{\sigma^2}{n}$ is dominant, so that we need to have the lower bound of the order $\frac{\sigma^2}{n}$, which is trivial (it follows from a reduction to the bound for the class composed of two constant functions).

It follows from (31) and (27) that the estimator $\hat{\boldsymbol{\mu}}^Q$ achieves, up to logarithmic factors, the optimal rate with respect to the minimax risk on the class $\mathcal{S}^{\uparrow}(V)$. Using (28) and the fact that the minimax risk is smaller than the minimax regret, we conclude that it is also the optimal rate up to logarithmic factors for the minimax regret.

Proof [Proof of Theorem 4] We assume for simplicity that n is a multiple of k. The general case is treated analogously. For any $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \{0,1\}^k$, let $d_H(\boldsymbol{\omega}, \boldsymbol{\omega}') = |\{i = 1, ..., k : \omega_i \neq \omega_i'\}|$ be the Hamming distance between $\boldsymbol{\omega}$ and $\boldsymbol{\omega}'$. By the Varshamov-Gilbert bound (Tsybakov, 2009, Lemma 2.9), there exists a set $\Omega \subset \{0,1\}^k$ such that

$$\mathbf{0} = (0, ..., 0) \in \Omega, \quad \log(|\Omega| - 1) \geq k/8, \quad \text{and} \quad d_H(\boldsymbol{\omega}, \boldsymbol{\omega}') > k/8 \tag{32}$$

for any two distinct $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega$. For each $\boldsymbol{\omega} \in \Omega$, define a vector $\boldsymbol{u}^{\boldsymbol{\omega}} \in \mathbb{R}^n$ with components

$$u_i^{\boldsymbol{\omega}} = \frac{\lfloor (i-1)k/n \rfloor \, V}{2k} + \gamma \omega_{\lfloor (i-1)k/n \rfloor + 1}, \qquad i = 1, ..., n,$$

where $\gamma = (1/8)\sqrt{\sigma^2 k/n}$, and $\lfloor x \rfloor$ denotes the maximal integer smaller than x. For any $\boldsymbol{\omega} \in \Omega$, $\boldsymbol{u}^{\boldsymbol{\omega}}$ is a piecewise constant sequence with $k(\boldsymbol{u}^{\boldsymbol{\omega}}) \leq k$, $\boldsymbol{u}^{\boldsymbol{\omega}}$ is a non-decreasing sequence because $\gamma \leq V/(2k)$, and by construction $V(\boldsymbol{u}^{\boldsymbol{\omega}}) \leq V$. Thus, $\boldsymbol{u}^{\boldsymbol{\omega}} \in \mathcal{S}_k^{\uparrow} \cap \mathcal{S}^{\uparrow}(V)$ for all $\boldsymbol{\omega} \in \Omega$. Moreover, for any $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega$,

$$\|\boldsymbol{u}^{\omega} - \boldsymbol{u}^{\omega'}\|^2 = \frac{\gamma^2}{k} d_H(\boldsymbol{\omega}, \boldsymbol{\omega}') \geq \frac{\gamma^2}{8} = \frac{\sigma^2 k}{512 n}. \tag{33}$$

Set for brevity $P_{\boldsymbol{\omega}} = \mathbb{P}_{\boldsymbol{u}^{\omega}}$. The Kullback-Leibler divergence $K(P_{\boldsymbol{\omega}}, P_{\boldsymbol{\omega}'})$ between $P_{\boldsymbol{\omega}}$ and $P_{\boldsymbol{\omega}'}$ is equal to $\frac{n}{2\sigma^2}\|\boldsymbol{u}^{\omega} - \boldsymbol{u}^{\omega'}\|^2$ for all $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega$. Thus,

$$K(P_{\boldsymbol{\omega}}, P_{\mathbf{0}}) = \frac{\gamma^2 n d_H(\mathbf{0}, \boldsymbol{\omega})}{2k\sigma^2} \leq \frac{k}{128} \leq \frac{\log(|\Omega| - 1)}{16}. \tag{34}$$

Applying (Tsybakov, 2009, Theorem 2.7) with $\alpha = 1/16$ completes the proof. ■

3. Estimation of Convex Sequences by Aggregation

Assume that $n \geq 3$ and define the set of convex sequences $\mathcal{S}^{\mathrm{C}}$ as follows:

$$\mathcal{S}^{\mathrm{C}} = \{\boldsymbol{u} = (u_1, \ldots, u_n) \in \mathbb{R}^n : \; 2u_i \leq u_{i+1} + u_{i-1}, \; i = 2, \ldots, n-1\}. \tag{35}$$

For any $\boldsymbol{u} \in \mathbb{R}^n$, we introduce the integer $q(\boldsymbol{u}) \geq 1$ such that $q(\boldsymbol{u}) - 1$ is the cardinality of the set $\{i = 1, ..., n-1 : 2u_i \neq u_{i+1} + u_{i-1}\}$. If $\boldsymbol{u} \in \mathcal{S}^{\mathrm{C}}$, $q(\boldsymbol{u}) - 1$ is the number of inequalities $2u_i \leq u_{i+1} + u_{i-1}$ that are strict for $i = 2, ..., n-1$. The value $q(\boldsymbol{u})$ is small if $\boldsymbol{u}$ is a piecewise linear sequence with a small number of pieces.

The performance of the least squares estimator over convex sequences $\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\mathrm{C}})$ has been recently studied in Guntuboyina and Sen (2013). If the unknown vector $\boldsymbol{\mu}$ belongs to the set $\mathcal{S}^{\mathrm{C}}$, Guntuboyina and Sen (2013) shows that the estimator $\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\mathrm{C}})$ satisfies the risk bound

$$\mathbb{E}_{\boldsymbol{\mu}} \left\| \hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\mathrm{C}}) - \boldsymbol{\mu} \right\|^2 \leq c \log(en)^{5/4} \left(\frac{\sigma^2 \sqrt{R(\boldsymbol{\mu})}}{n} \right)^{4/5},$$

where $R(\boldsymbol{\mu}) = \max(1, \min\{\|\boldsymbol{\tau} - \boldsymbol{\mu}\|^2, \boldsymbol{\tau} \text{ is affine}\})$ and $c > 0$ is an absolute constant. It is proved in (Chatterjee et al., 2015, Example 2.3) that the least squares estimator $\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\mathrm{C}})$ satisfies the oracle inequality

$$\mathbb{E}_{\boldsymbol{\mu}} \left\| \hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\mathrm{C}}) - \boldsymbol{\mu} \right\|^2 \leq 6 \min_{\boldsymbol{u} \in \mathcal{S}^{\mathrm{C}}} \left(\|\boldsymbol{u} - \boldsymbol{\mu}\|^2 + \frac{c\sigma^2 q(\boldsymbol{u}) \log\left(\frac{en}{q(\boldsymbol{u})}\right)^{5/4}}{n} \right), \tag{36}$$

where $c > 0$ is an absolute constant. The right hand side of (36) is small if the unknown vector $\boldsymbol{\mu}$ can be well approximated by a piecewise linear sequence in $\mathcal{S}^{\mathrm{C}}$ with not too many pieces.

The leading constant in (36) is 6. We will show that sparsity pattern aggregation achieves a substantially better performance. We obtain the sharp oracle inequality (39) below, improving upon (36) not only in the fact that the leading constant is 1 but also in the rate of the remainder term; we will see that the exponent 5/4 of the logarithmic factor is reduced to 1.

For any set $J \subseteq \{2, ..., n-1\}$, define

$$\nu_J \coloneqq \frac{\exp(-|J|)}{H_C \binom{n-2}{|J|}}, \qquad H_C \coloneqq \sum_{i=0}^{n-2} \exp(-i). \tag{37}$$

Let $Q_J \in \mathbb{R}^{n \times n}$ be the projector on the linear subspace W_J of $\mathbb{R}^n$ given by

$$W_J \coloneqq \Big\{ \boldsymbol{u} \in \mathbb{R}^n : \ \forall i \in \{2, ..., n-1\} \setminus J, \ 2u_i = u_{i+1} + u_{i-1} \Big\}.$$

Given a vector $\mathbf{y}$ of observations and $\boldsymbol{\theta} = (\theta_J)_{J \subseteq \{2,...,n-1\}}$ where each θ_J belongs to $\mathbb{R}$, let

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} = \sum_{J \subseteq \{2,...,n-1\}} \theta_J Q_J \mathbf{y}.$$

Finally, let

$$\hat{\boldsymbol{\mu}}^{Q-conv} = \boldsymbol{\mu}_{\hat{\boldsymbol{\theta}}}$$

where $\hat{\boldsymbol{\theta}}$ is the solution of the optimization problem

$$\min_{\boldsymbol{\theta} \in \Lambda'} \ \|\boldsymbol{\mu}_{\boldsymbol{\theta}} - \mathbf{y}\|^2 + \sum_{J \subset \{2,...,n-1\}} \theta_J \left(\frac{2\sigma^2 |J|}{n} + \frac{1}{2} \|\boldsymbol{\mu}_{\boldsymbol{\theta}} - Q_J \mathbf{y}\|^2 + \frac{46\sigma^2}{n} \log \frac{1}{\nu_J} \right)$$

where

$$\Lambda' = \left\{ \boldsymbol{\theta} : \ \theta_J \geq 0 \text{ for all } J \subseteq \{2, ..., n-1\}, \text{ and } \sum_{J \subseteq \{2,...,n-1\}} \theta_J = 1 \right\}.$$

The structure of this minimization problem is the same as of its analog introduced in Section 2. This is a quadratic program that aggregates the linear estimators $(Q_J \mathbf{y})_{J \subseteq \{2,...,n-1\}}$ using the Q-aggregation procedure Dai et al. (2012, 2014); Bellec (2014) with the prior weights (37).

Theorem 6 *Let $\boldsymbol{\mu} \in \mathbb{R}^n$, $n \geq 3$, and assume that the noise vector $\boldsymbol{\xi}$ has distribution $\mathcal{N}(0, \sigma^2 I_{n \times n})$. There exist absolute constants $c, c' > 0$ such that for all $\delta \in (0, 1/3)$, the estimator $\hat{\boldsymbol{\mu}}^{Q-conv}$ satisfies with probability at least $1 - 3\delta$,*

$$\left\| \hat{\boldsymbol{\mu}}^{Q-conv} - \boldsymbol{\mu} \right\|^2 \leq \min_{\boldsymbol{u} \in \mathbb{R}^n} \left(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c\sigma^2 q(\boldsymbol{u})}{n} \log \frac{en}{q(\boldsymbol{u})} \right) + \frac{c\sigma^2 \log(1/\delta)}{n}, \tag{38}$$

and we have

$$\mathbb{E}_{\boldsymbol{\mu}} \left\| \hat{\boldsymbol{\mu}}^{Q-conv} - \boldsymbol{\mu} \right\|^2 \leq \min_{\boldsymbol{u} \in \mathbb{R}^n} \left(\|\boldsymbol{\mu} - \boldsymbol{u}\|^2 + \frac{c'\sigma^2 q(\boldsymbol{u})}{n} \log \frac{en}{q(\boldsymbol{u})} \right). \tag{39}$$

The proof of this theorem is the same as that of Theorem 1 with the only difference that J is now a subset of $\{2, ..., n-1\}$ rather than that of $\{1, ..., n-1\}$, and we replace the notation P_J and V_J by Q_J and W_J respectively.

The leading constant of the oracle inequality (39) is 1, and the remainder term is proportional to $q(\boldsymbol{u}) \log(en/q(\boldsymbol{u}))$. These are two improvements upon (36), where the leading constant is 6 and the remainder term is proportional to $q(\boldsymbol{u}) \log(en/q(\boldsymbol{u}))^{5/4}$.

In view of (39), for the class of piecewise linear convex sequences with at most q linear pieces, $\mathcal{S}_q^{\mathrm{C}} = \{\boldsymbol{u} \in \mathcal{S}^{\mathrm{C}} : q(\boldsymbol{u}) \leq q\}$ we have the following bounds for the maximal expected regrets

$$\max_{\boldsymbol{\mu} \in \mathbb{R}^n} \left(\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\| - \min_{\boldsymbol{u} \in \mathcal{S}_q^{\mathrm{C}}} \|\boldsymbol{u} - \boldsymbol{\mu}\| \right) \leq c \sqrt{\frac{\sigma^2 q}{n} \log\left(\frac{en}{q}\right)}, \tag{40}$$

$$\max_{\boldsymbol{\mu} \in \mathbb{R}^n} \left(\mathbb{E}_{\boldsymbol{\mu}} \|\hat{\boldsymbol{\mu}}^Q - \boldsymbol{\mu}\|^2 - \min_{\boldsymbol{u} \in \mathcal{S}_q^{\mathrm{C}}} \|\boldsymbol{u} - \boldsymbol{\mu}\|^2 \right) \leq \frac{c\sigma^2 q}{n} \log\left(\frac{en}{q}\right), \tag{41}$$

where $c > 0$ is an absolute constant. The same bounds hold for the minimax risks over $\mathcal{S}_q^{\mathrm{C}}$ since the minimax risk is smaller than the minimax regret.

The following proposition shows that the rates of convergence in (40) and (41) are optimal up to logarithmic factors. We omit the discussion since it is similar to that after Theorem 4.

Proposition 7 *Let $n \geq 3$. There exist absolute constants $c, c' > 0$ such that, for any positive integer $q \leq n$,*

$$\inf_{\hat{\boldsymbol{\mu}}} \sup_{\boldsymbol{\mu} \in \mathcal{S}_q^{\mathrm{C}}} \mathbb{P}_{\boldsymbol{\mu}} \left(\|\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\|^2 \geq \frac{c\sigma^2 q}{n} \right) > c', \tag{42}$$

where the infimum is taken over all estimators.

Proof Assume that $q \geq 2$ since for $q = 1$ the result is trivial. We also assume for simplicity that n is a multiple of q. Let $m = n/q$ and $\gamma = (1/8)\sqrt{\sigma^2 q/n}$. Set $\beta_0 = 0, \alpha_0 = 0$ and define, for all integers $j \geq 1$,

$$\beta_j = \beta_{j-1} + \gamma + m\alpha_{j-1}, \qquad \alpha_j = 2\gamma + \alpha_{j-1}. \tag{43}$$

By the Varshamov-Gilbert bound (Tsybakov, 2009, Lemma 2.9) there exists $\Omega \subset \{0,1\}^q$ such that (32) is satisfied, with k replaced by q. For each $\boldsymbol{\omega} \in \Omega$, define a vector $\boldsymbol{u}^{\boldsymbol{\omega}} \in \mathbb{R}^n$ with components

$$u^{\boldsymbol{\omega}}_{jm+i} = \omega_{j+1}\gamma + \alpha_j(i-1) + \beta_j, \qquad j = 0, ..., q-1, \quad i = 1, ..., m.$$

The sequence $\boldsymbol{u}^{\boldsymbol{\omega}}$ is piecewise linear. It is linear with slope α_j on the set $\{jm+1, ..., (j+1)m\}$ for any $j = 0, ..., q-1$. Thus, $q(\boldsymbol{u}^{\boldsymbol{\omega}}) = q$. Next, we prove that $\boldsymbol{u}^{\boldsymbol{\omega}} \in \mathcal{S}^{\mathrm{C}}$ for all $\boldsymbol{\omega} \in \Omega$. It is enough to check the convexity condition at the endpoints of the linear pieces:

$$2u^{\boldsymbol{\omega}}_{jm} \leq u^{\boldsymbol{\omega}}_{jm-1} + u^{\boldsymbol{\omega}}_{jm+1}, \qquad 2u^{\boldsymbol{\omega}}_{jm+1} \leq u^{\boldsymbol{\omega}}_{jm} + u^{\boldsymbol{\omega}}_{jm+2}, \tag{44}$$

for all $j = 1, ..., q-1$. Using (43) we get that, for all $j = 1, ..., q-1$,

$$\begin{aligned} u^{\boldsymbol{\omega}}_{jm+1} - u^{\boldsymbol{\omega}}_{jm} &= \omega_{j+1}\gamma + \beta_j - (\omega_j\gamma + \alpha_{j-1}(m-1) + \beta_{j-1}), \\ &= (\omega_{j+1} - \omega_j + 1)\gamma + \alpha_{j-1}, \\ &= (\omega_{j+1} - \omega_j - 1)\gamma + \alpha_j. \end{aligned}$$

Hence, $\alpha_{j-1} \leq u^{\boldsymbol{\omega}}_{jm+1} - u^{\boldsymbol{\omega}}_{jm} \leq \alpha_j$. Since also $\alpha_{j-1} = u^{\boldsymbol{\omega}}_{jm} - u^{\boldsymbol{\omega}}_{jm-1}$ and $\alpha_j = u^{\boldsymbol{\omega}}_{jm+2} - u^{\boldsymbol{\omega}}_{jm+1}$, it follows that the two inequalities (44) hold, for all $j = 1, ..., q-1$. Thus, $\boldsymbol{u}^{\boldsymbol{\omega}} \in \mathcal{S}^{\mathrm{C}}$. In summary, we have proved that $\boldsymbol{u}^{\boldsymbol{\omega}} \in \mathcal{S}^{\mathrm{C}}_q$ for all $\boldsymbol{\omega} \in \Omega$.

Now, from the Varshamov-Gilbert bound, cf. (32), for $\boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega$ we have

$$\|\boldsymbol{u}^{\omega} - \boldsymbol{u}^{\omega'}\|^2 = \frac{\gamma^2}{q} d_H(\boldsymbol{\omega}, \boldsymbol{\omega}') \geq \frac{\gamma^2}{8} = \frac{\sigma^2 q}{512n}, \tag{45}$$

where $d_H(\cdot,\cdot)$ is the Hamming distance. Finally, similarly to (34), the Kullback-Leibler divergence between $P_{\boldsymbol{\omega}}$ and $P_{\mathbf{0}}$ satisfies $K(P_{\boldsymbol{\omega}}, P_{\mathbf{0}}) \leq \frac{\log(|\Omega|-1)}{16}$. Applying (Tsybakov, 2009, Theorem 2.7) with $\alpha = 1/16$ completes the proof. ■

4. Concluding Remarks and Discussion

In this short note, we have shown that the estimators $\hat{\boldsymbol{\mu}}^{Q}$ and $\hat{\boldsymbol{\mu}}^{Q-conv}$ based on sparsity pattern aggregation (in its Q-aggregation version) achieve oracle inequalities that improve on some previous results for isotonic and convex regression.

One of the improvements is that oracle inequalities (17) and (39) are sharp, i.e., with leading constant 1 and they are valid for all $\boldsymbol{\mu} \in \mathbb{R}^n$. It allows us to obtain bounds for the minimax regret under arbitrary model misspecification, which was not possible based on the previous results. We show that these bounds are rate optimal up to logarithmic factors.

The question on whether the least squares estimators under monotonicity and convexity constraints can achieve sharp oracle inequalities with correct rates remains open.

Another improvement is that we obtain oracle inequalities both with high probability and in expectation, which was not the case in the previous work.

An advantage of the least squares estimator is that it requires no tuning parameters. In particular, the knowledge of σ^2 is not needed to construct the estimators $\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\uparrow})$ and $\hat{\boldsymbol{\mu}}^{LS}(\mathcal{S}^{\mathrm{C}})$. This is in contrast to the ℓ_1 penalized estimator (7) and the estimators $\hat{\boldsymbol{\mu}}^{Q}$ and $\hat{\boldsymbol{\mu}}^{Q-conv}$; their construction requires the knowledge of σ^2. For the ℓ_1 penalized estimator (7), the issue may be addressed by using a scale-free version of the Lasso Belloni et al. (2014); Sun and Zhang (2012). For the Q-aggregation estimators $\hat{\boldsymbol{\mu}}^{Q}$ and $\hat{\boldsymbol{\mu}}^{Q-conv}$, we can treat the issue of unknown σ as in Bellec (2014). Namely, it is shown in Bellec (2014) that the oracle inequalities for Q-aggregation procedures are essentially preserved after plugging in an estimator $\hat{\sigma}^2$ of σ^2 that satisfies $|\hat{\sigma}^2/\sigma^2 - 1| \leq 1/8$ with high probability, which is even weaker than consistency.

Finally, note that instead of Q-aggregation we could have used sparsity pattern aggregation by the Exponential Screening procedure of Rigollet and Tsybakov (2011). This would lead to sharp oracle inequalities in expectation of the form (17) and (39) but not to inequalities with high probability such as (16) and (38). This is the reason why we have opted for Q-aggregation rather than for Exponential Screening in this paper. On the other hand, Exponential Screening estimators are computationally more attractive than Q-aggregation since they can be successfully approximated by MCMC algorithms (see Rigollet and Tsybakov (2011, 2012) for details).

Acknowledgement. This work was supported by GENES and by the French National Research Agency (ANR) under the grants IPANEMA (ANR-13-BSH1-0004-02), and Labex ECODEC (ANR - 11-LABEX-0047). It was also supported by the "Chaire Economie et Gestion des Nouvelles Données", under the auspices of Institut Louis Bachelier, Havas-Media and Paris-Dauphine.

References

Fadoua Balabdaoui and Jon A. Wellner. Estimation of a k-monotone density: Limit distribution theory and the spline connection. *Annals of Statistics*, 35:2536–2564, 2007.

Moulinath Banerjee and Jon A. Wellner. Likelihood ratio tests for monotone functions. *Annals of Statistics*, 29:1699–1731, 2001.

Pierre C. Bellec. Optimal bounds for aggregation of affine estimators. *arXiv:1410.0346*, 2014.

A. Belloni, V. Chernozhukov, and L. Wang. Pivotal estimation via square-root lasso in nonparametric regression. *Annals of Statistics*, 42:757–788, 2014.

S. Chatterjee, A. Guntuboyina, and B. Sen. On risk bounds in isotonic and other shape restricted regression problems. *Annals of Statistics*, 43:1774–1800, 2015.

D. Dai, P. Rigollet, and T. Zhang. Deviation optimal learning using greedy Q-aggregation. *Annals of Statistics*, 40:1878–1905, 2012.

D. Dai, P. Rigollet, Xia L., and Zhang T. Aggregation of affine estimators. *Electronic J. Statist.*, 8:302–327, 2014.

A. S. Dalalyan, M. Hebiri, and J. Lederer. On the prediction performance of the lasso. *arXiv:1402.1700*, 2014.

A. Guntuboyina and B. Sen. Global risk bounds and adaptation in univariate convex regression. *arXiv:1305.1648*, 2013. To appear in Probability Theory and Related Fields.

D. Hsu, S. Kakade, and T. Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17:1–6, 2012.

E. Mammen and S. van de Geer. Locally adaptive regression splines. *Annals of Statistics*, 25:387–413, 1997.

M. Meyer and M. Woodroofe. On the degrees of freedom in shape-restricted regression. *Annals of Statistics*, 28:1083–1104, 2000.

A.M. Nemirovski, B.T. Polyak, and Tsybakov A.B. Rate of convergence of nonparametric estimators of maximum-likelihood type. *Problems of Information Transmission*, 21: 258–272, 1985.

A. Rakhlin, K. Sridharan, and A.B. Tsybakov. Empirical entropy, minimax regret and minimax risk. *arXiv:1308.1147*, 2013. To appear in Bernoulli.

P. Rigollet. Kullback–Leibler aggregation and misspecified generalized linear models. *Annals of Statistics*, 40:639–665, 2012.

P. Rigollet and A.B. Tsybakov. Exponential screening and optimal rates of sparse estimation. *Annals of Statistics*, 39:731–771, 2011.

P. Rigollet and A.B. Tsybakov. Sparse estimation by exponential weighting. *Statistical Science*, 27:558–575, 2012.

T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99:879–898, 2012.

A.B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

A.B. Tsybakov. Aggregation and minimax optimality in high dimensional estimation. *Proceedings of International Congress of Mathematicians (Seoul, 2014)*, 3:225–246, 2014.

V. N. Vapnik and A. Y. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974.

C.-H. Zhang. Risk bounds in isotonic regression. *Annals of Statistics*, 30:528–555, 2002.

Journal of Machine Learning Research 16 (2015) 1893-1922 Submitted 5/15; Published 9/15

Exceptional Rotations of Random Graphs: A VC Theory

Louigi Addario-Berry LOUIGI.ADDARIO@MCGILL.CA
Department of Mathematics and Statistics, McGill University
Burnside Hall, Room 1219, 805 Sherbrooke W.
Montreal, QC, H3A 0B9, Canada

Shankar Bhamidi BHAMIDI@EMAIL.UNC.EDU
Department of Statistics and Operations Research
304 Hanes Hall CB # 3260
University of North Carolina
Chapel Hill, NC 27599, USA

Sébastien Bubeck SEBUBECK@MICROSOFT.COM
Microsoft Research
Building 99, 2955, Redmond, WA 98052, USA

Luc Devroye LUCDEVROYE@GMAIL.COM
School of Computer Science, McGill University
3480 University Street
Montreal, Canada H3A 0E9

Gábor Lugosi GABOR.LUGOSI@UPF.EDU
ICREA and Pompeu Fabra University
Department of Economics and Business
Ramon Trias Fargas 25–27
08005 Barcelona, Spain

Roberto Imbuzeiro Oliveira RIMFO@IMPA.BR
IMPA
Estrada Da. Castorina, 110.
Rio de Janeiro, RJ, Brazil. 22460-320

Editor: Alex Gammerman and Vladimir Vovk

Abstract

In this paper we explore maximal deviations of large random structures from their typical behavior. We introduce a model for a high-dimensional random graph process and ask analogous questions to those of Vapnik and Chervonenkis for deviations of averages: how "rich" does the process have to be so that one sees atypical behavior.

In particular, we study a natural process of Erdős-Rényi random graphs indexed by unit vectors in $\mathbb{R}^d$. We investigate the deviations of the process with respect to three fundamental properties: clique number, chromatic number, and connectivity. In all cases we establish upper and lower bounds for the minimal dimension d that guarantees the existence of "exceptional directions" in which the random graph behaves atypically with respect to the property. For each of the three properties, four theorems are established, to describe upper and lower bounds for the threshold dimension in the subcritical and supercritical regimes.

Keywords: random graphs, VC theory, clique number, chromatic number, connectivity

1. Introduction

One of the principal problems in probability and statistics is the understanding of maximal deviations of averages from their means. The revolutionary work of Vapnik and Chervonenkis (1971, 1974, 1981) introduced a completely new combinatorial approach that opened many paths and helped us understand this fundamental phenomena. Today, the Vapnik-Chervonenkis theory has become the theoretical basis of statistical machine learning, empirical process theory, and has applications in a diverse array of fields.

The purpose of this paper is to initiate the exploration of maximal deviations of complex random structures from their typical behavior. We introduce a model for a high-dimensional random graph process and ask analogous questions to those of Vapnik and Chervonenkis for deviations of averages: how "rich" does the process have to be so that one sees atypical behavior. In particular, we study a process of Erdős-Rényi random graphs. In the $G(n,p)$ model introduced by Erdős and Rényi (1959, 1960), a graph on n vertices is obtained by connecting each pair of vertices with probability p, independently, at random. The $G(n,p)$ model has been thoroughly studied and many of its properties are well understood—see, e.g., the monographs of Bollobás (2001) and Janson et al. (2000).

In this paper we introduce a random graph process indexed by unit vectors in $\mathbb{R}^d$, defined as follows. For positive integer n, write $[n] = \{1, \ldots, n\}$. For $1 \le i < j \le n$, let $X_{i,j}$ be independent standard normal vectors in $\mathbb{R}^d$. Denote by $\boldsymbol{X}_n = (X_{i,j})_{1 \le i < j \le n}$ the collection of these random points. For each $s \in S^{d-1}$ (where S^{d-1} denotes the unit sphere in $\mathbb{R}^d$) and $t \in \mathbb{R}$ we define the random graph $\Gamma(\boldsymbol{X}_n, s, t)$ with vertex set $v(\Gamma(\boldsymbol{X}_n, s, t)) = [n]$ and edge set $e(\Gamma(\boldsymbol{X}_n, s, t)) = \{\{i,j\} : \langle X_{i,j}, s\rangle \ge t\}$, where $\langle \cdot, \cdot \rangle$ denotes the usual inner product in $\mathbb{R}^d$.

For any fixed $s \in S^{d-1}$ and $t \in \mathbb{R}$, $\Gamma(\boldsymbol{X}_n, s, t)$ is distributed as an Erdős-Rényi random graph $G(n,p)$, with $p = 1 - \Phi(t)$ where Φ is the distribution function of a standard normal random variable. In particular, $\Gamma(\boldsymbol{X}_n, s, 0)$ is a $G(n, 1/2)$ random graph. With a slight abuse of notation, we write $\Gamma(\boldsymbol{X}_n, s)$ for $\Gamma(\boldsymbol{X}_n, s, 0)$.

We study the *random graph process*

$$\mathbb{G}_{d,p}(\boldsymbol{X}_n) = \left\{ \Gamma(\boldsymbol{X}_n, s, \Phi^{-1}(1-p)) : s \in S^{d-1} \right\} .$$

$\mathbb{G}_{d,p}(\boldsymbol{X}_n)$ is a stationary process of $G(n,p)$ random graphs, indexed by d-dimensional unit vectors. For larger values of d, the process becomes "richer". Our aim is to explore how large the dimension d needs to be for there to exist random directions s for which $\Gamma(\boldsymbol{X}_n, s, \Phi^{-1}(1-p)) \in \mathbb{G}_{d,p}(\boldsymbol{X}_n)$ has different behavior from what is expected from a $G(n,p)$ random graph. Adapting terminology from dynamical percolation Steif (2009), we call such directions *exceptional rotations*. More precisely, in analogy with the Vapnik-Chervonenkis theory of studying atypical deviations of averages from their means, our aim is to develop a *VC theory* of random graphs. In particular, we study three fundamental properties of the graphs in the family $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$: the size of the largest clique, the chromatic number, and connectivity. In the first two cases we consider $p = 1/2$ while in the study of connectivity we focus on the case when $p = c \log n / n$ for some constant $c > 0$.

The graph properties we consider are all monotone, so have a critical probability p^* at which they are typically obtained by $G(n,p)$. For example, consider connectivity, and suppose we first place ourselves above the critical probability in $G(n,p)$, e.g., $p = c \log n / n$

for $c > 1$, so that $G(n,p)$ is with high probability connected. Then the question is how large should d be to ensure that for some member graph in the class, the property (connectivity) disappears. There is a threshold *dimension* d for this, and we develop upper and lower bounds for that dimension. Secondly, consider the regime below the critical probability for connectivity in $G(n,p)$, e.g., $p = c \log n/n$ for $c < 1$. In this case, with high probability $G(n,p)$ is not connected, and we ask how large d should be to ensure that for some member graph in the class, the property (connectivity) appears. Again, we develop upper and lower bounds for the threshold dimension d for this.

In all, for each of the three properties considered in this paper, clique number, chromatic number, and connectivity, four theorems are needed, to describe upper and lower bounds for the threshold dimension for exceptional behaviour in the subcritical regime (when the property typically does not obtain) and in the supercritical regime (when the property typically does obtain). In every case, our results reveal a remarkable asymmetry between "upper" and "lower" deviations relative to this threshold.

Our techniques combine some of the essential notions introduced by Vapnik and Chervonenkis (such as shattering, covering, packing, and symmetrization), with elements of high-dimensional random geometry, coupled with sharp estimates for certain random graph parameters.

The model considered in this paper uses subsets of the collection of halfspaces in $\mathbb{R}^d$ to define the random graphs in the collection. A natural variant would be one in which we associate with each edge $\{i,j\}$ a uniformly distributed random vector on the torus $[0,1]^d$, and consider a class parametrized by $s \in [0,1]^d$. Then define the edge set $e(\Gamma(\boldsymbol{X}_n, s, t)) = \{\{i,j\} : \|X_{i,j} - s\| \le t\}$. For general classes of sets of $\mathbb{R}^d$, the complexity of the classes will affect the behaviour of the collection of random graphs in a universal manner. We can define the complexity of a class of graphs indexed in terms of the threshold dimension needed to make certain graph properties appear or disappear in the subcritical and supercritical regimes, respectively. It will be interesting to explore the relationship between the combinatorial geometry of the class and these complexities.

Note that when $d = 1$, $\mathbb{G}_{1,p}(\boldsymbol{X}_n)$ only contains two graphs (when $p = 1/2$, one is the complement of the other), and therefore the class is trivial. On the other extreme, when $d \ge \binom{n}{2}$, with probability one, the collection $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ contains *all* $2^{\binom{n}{2}}$ graphs on n vertices. This follows from the following classical result on the "VC shatter coefficient" of linear half spaces (see, e.g., Schläffli 1950, Cover 1965) that determines the number of different graphs in $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ (with probability one).

Lemma 1 *Given $N \ge d$ points $x_1, \ldots, x_N \in \mathbb{R}^d$ in general position (i.e., every subset of d points is linearly independent), the number of binary vectors $b \in \{0,1\}^N$ of the form $b = \left(\mathbb{1}_{\{\langle x_i, s\rangle \ge 0\}}\right)_{i \le N}$ for some $s \in S^{d-1}$ equals*

$$C(N,d) = 2\sum_{k=0}^{d-1} \binom{N-1}{k} .$$

In particular, when $N = d$, all 2^N possible dichotomies of the N points are realizable by some linear half space with the origin on its boundary. In such a case we say that the N points are *shattered* by half spaces.

1.1 Notation and Overview

Throughout the paper, log denotes natural logarithm. For a sequence $\{A_n\}$ of events, we say that A_n holds *with high probability* if $\lim_{n\to\infty} \mathbb{P}\{A_n\} = 1$.

The paper is organized as follows. In Section 2 we study the clique number in the case $p = 1/2$. The four parts of Theorem 2 establish upper and lower bounds for the critical dimension above which, with high probability, there exist graphs in $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ whose largest clique is significantly larger/smaller than the typical value, which is $\approx 2\log_2 n - 2\log_2\log_2 n$. We show that the critical dimension for which some graphs in $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ have a clique number at least, say, $10\log_2 n$ is of the order of $\log^2 n/\log\log n$.

In sharp contrast to this, d needs to be at least $n^2/\,polylog\, n$ to find a graph in $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ with maximum clique size 3 less than the typical value. We study this functional in Section 3. Theorem 3 summarizes the four statements corresponding to upper and lower bounds in the sub-, and super-critical regime. Once again, the two regimes exhibit an important asymmetry. While no graphs in $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ have a chromatic number a constant factor larger than typical unless d is is of the order of $n^2/\,polylog\, n$, there exist graphs with a constant factor smaller chromatic number for d near n.

Finally, in Section 4, connectivity properties are examined. To this end, we place ourselves in the regime $p = c\log n/n$ for some constant c. When $c < 1$, a typical graph $G(n,p)$ is disconnected, with high probability, while for $c > 1$ it is connected. In Theorem 6 we address both cases. We show that for $c > 1$, the critical dimension above which one finds disconnected graphs among $\mathbb{G}_{d,c\log n/n}(\boldsymbol{X}_n)$ is of the order of $\log n/\log\log n$. (Our upper and lower bounds differ by a factor of 2.) We also show that when $c < 1$, d needs to be at least roughly n^{1-c} in order to find a connected graph $\mathbb{G}_{d,c\log n/n}(\boldsymbol{X}_n)$. While we conjecture this lower bound to be sharp, we do not have a matching upper bound in this case. However, we are able to show that when d is at least of the order of $n\sqrt{\log n}$, $\mathbb{G}_{d,c\log n/n}(\boldsymbol{X}_n)$ not only contains some connected graphs but with high probability, for any spanning tree, there exists $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}_n, s, t)$ contains that spanning tree. This property holds for even much smaller values of p.

In the Appendix we gather some technical estimates required for the proofs. Before diving into the proofs we make one final remark regarding the proof techniques. Fix an increasing graph property $\mathcal{P}$. One *natural* way to show that with high probability there exists a direction s for which $\Gamma(\boldsymbol{X}_n, s, t)$ has $\mathcal{P}$ is as follows. Fix p in $(0,1)$ such that $G(n,p)$ has property $\mathcal{P}$ with high probability; then show that with high probability there exists a direction s for which $\Gamma(\boldsymbol{X}_n, s, t)$ has at least $p\binom{n}{2}$ edges. This type of argument, and its obvious analogue for decreasing graph properties, maximally decouple geometric and graph theoretic considerations. For the lower tail of the clique number, our results, Theorem 2 (i) and (ii), leave open the possibility that such an argument could yield tight bounds for threshold dimensions. For the remaining properties we consider, our results rule this out – the dimensional thresholds cannot be explained by edge density alone.

2. Clique Number

In this section we consider $p = 1/2$ and investigate the extremes of the clique number amongst the graphs $\Gamma(\boldsymbol{X}_n, s)$, $s \in S^{d-1}$. Denote by $cl(\boldsymbol{X}_n, s)$ the size of the largest clique in $\Gamma(\boldsymbol{X}_n, s)$.

The typical behavior of the clique number of a $G(n, 1/2)$ random graph is quite accurately described by Matula's classical theorem (Matula, 1972) that states that for any fixed $s \in S^{d-1}$, for any $\epsilon > 0$,

$$cl(\boldsymbol{X}_n, s) \in \{\lfloor \omega - \epsilon \rfloor, \lfloor \omega + \epsilon \rfloor\}$$

with probability tending to 1, where $\omega = 2\log_2 n - 2\log_2 \log_2 n + 2\log_2 e - 1$.

Here we are interested in understanding the values of d for which graphs with atypical clique number appear. We prove below that while for moderately large values of d some graphs $\Gamma(\boldsymbol{X}_n, s)$ have a significantly larger clique number than ω, one does not find graphs with significantly smaller clique number unless d is nearly quadratic in n.

Observe first that by Lemma 1 for any k, if $d \geq \binom{k}{2}$, then, with probability one, $cl(\boldsymbol{X}_n, s) \geq k$ for some $s \in S^{d-1}$. (Just fix any set of k vertices; all 2^k graphs on these vertices is present for some s, including the complete graph.) For example, when $d \sim (9/2)(\log_2 n)^2$, $cl(\boldsymbol{X}_n, s) \geq 3\log_2 n$ for some $s \in S^{d-1}$, a quite atypical behavior. In fact, with a more careful argument we show below that when d is a sufficiently large constant multiple of $(\log n)^2/\log\log n$, then, with high probability, there exists $s \in S^{d-1}$ such that $cl(\boldsymbol{X}_n, s) \geq 3\log_2 n$. We also show that no such s exists for $d = o((\log n)^2/\log\log n)$. Perhaps more surprisingly, clique numbers significantly smaller than the typical value only appear for huge values of d. The next theorem shows the surprising fact that in order to have that for some $s \in S^{d-1}$, $cl(\boldsymbol{X}_n, s) < \omega - 3$, the dimension needs to be $n^{2-o(1)}$. (Recall that for $d = \binom{n}{2}$ the point set $\boldsymbol{X}_n$ is shattered and one even has $cl(\boldsymbol{X}_n, s) = 1$ for some s. Our findings on the clique number are summarized in the following theorem.

Theorem 2 (CLIQUE NUMBER.) *If $cl(\boldsymbol{X}_n, s)$ denotes the clique number of $\Gamma(\boldsymbol{X}_n, s)$, then, with high probability the following hold:*

(i) (SUBCRITICAL; NECESSARY.) *If $d = o(n^2/(\log n)^9)$, then for all $s \in S^{d-1}$, $cl(\boldsymbol{X}_n, s) > \lfloor \omega - 3 \rfloor$.*

(ii) (SUBCRITICAL; SUFFICIENT.) *If $d \geq \binom{n}{2}$, then there exists $s \in S^{d-1}$ such that $cl(\boldsymbol{X}_n, s) = 1$.*

(iii) (SUPERCRITICAL; NECESSARY.) *For any $c > 2$ there exists $c' > 0$ such that if $d \leq c' \log^2 n/\log\log n$, then for all $s \in S^{d-1}$, we have $cl(\boldsymbol{X}_n, s) \leq c\log_2 n$.*

(iv) (SUPERCRITICAL; SUFFICIENT.) *For any $c > 2$ and $c' > c^2/(2\log 2)$, if it is the case that $d \geq c' \log^2 n/\log\log n$, then there exists $s \in S^{d-1}$ such that $cl(\boldsymbol{X}_n, s) \geq c\log_2 n$.*

The event described in (ii) holds with probability one for all n.

Proof To prove part (i), let $k = \lfloor \omega - 3 \rfloor$ and let $N_k(s)$ denote the number of cliques of size k in $\Gamma(\boldsymbol{X}_n, s)$. Let $\eta \in (0, 1]$ and let $\mathcal{C}_\eta$ be a minimal η-cover of S^{d-1}. Then

$$\begin{aligned}
&\mathbb{P}\left\{\exists s \in S^{d-1} : N_k(s) = 0\right\} \\
&\quad = \quad \mathbb{P}\left\{\exists s' \in \mathcal{C}_\eta \text{ and } \exists s \in S^{d-1} : \|s - s'\| \leq \eta : N_k(s) = 0\right\} \\
&\quad \leq \quad |\mathcal{C}_\eta| \mathbb{P}\left\{\exists s \in S^{d-1} : \|s - s_0\| \leq \eta : N_k(s) = 0\right\}
\end{aligned}$$

where $s_0 = (1, 0, \ldots, 0)$ and the last inequality follows from the union bound. Consider the graph $\Gamma(\boldsymbol{X}_n, s_0, -\eta\sqrt{1-\eta^2/4})$ in which vertex i and vertex j are connected if and only if the first component of $X_{i,j}$ is at least $-\eta\sqrt{1-\eta^2/4}$

The proof of Lemma 12 implies that the event $\left\{\exists s \in S^{d-1} : \|s - s_0\| \le \eta : N_k(s) = 0\right\}$ is included in the event that $\Gamma(\boldsymbol{X}_n, s_0, -\eta\sqrt{1-\eta^2/4})$ does not have any clique of size k. By Lemma 12, the probability of this is bounded by the probability that an Erdős-Rényi random graph $G(n, 1/2 - \alpha_n)$ does not have any clique of size k where $\alpha_n = \frac{\eta\sqrt{d}}{\sqrt{2\pi}}$. If we choose (say) $\eta = 1/n^2$ then for $d \le n^2$ we have $\alpha_n \le 1/n$ and therefore, by Lemma 17 below,

$$\mathbb{P}\left\{\exists s \in S^{d-1} : \|s - s_0\| \le \eta : N_k(s) = 0\right\} \le \exp\left(\frac{-C'n^2}{(\log_2 n)^8}\right)$$

for some numerical constant C'. Thus, using Lemma 10,

$$\mathbb{P}\left\{\exists s \in S^{d-1} : N_k(s) = 0\right\} \le (4n^2)^d \exp\left(\frac{-C'n^2}{(\log_2 n)^8}\right) = o(1)$$

whenever $d = o(n^2/(\log n)^9)$.

Part (ii) follows from the simple fact that, by Lemma 1, with $d = \binom{n}{2}$ even the empty graph appears among the $\Gamma(\boldsymbol{X}_n, s)$.

The proof of part (iii) proceeds similarly to that of part (i). Let $k = c\log_2 n$. Then

$$\begin{aligned}
&\mathbb{P}\left\{\exists s \in S^{d-1} : N_k(s) \ge 1\right\} \\
&\quad\le \quad |\mathcal{C}_\eta| \mathbb{P}\left\{\exists s \in S^{d-1} : \|s - s_0\| \le \eta : N_k(s) \ge 1\right\} .
\end{aligned}$$

Similarly to the argument of (i), we note that the event $\left\{\exists s \in S^{d-1} : \|s - s_0\| \le \eta : N_k(s) \ge 1\right\}$ is included in the event that $\Gamma(\boldsymbol{X}_n, s_0, -\eta\sqrt{1-\eta^2/4})$ has a clique of size k, which is bounded by the probability that an Erdős-Rényi random graph $G(n, 1/2 + \alpha_n)$ has a clique of size k where $\alpha_n = \frac{\eta\sqrt{d}}{\sqrt{2\pi}}$. Denoting $p = 1/2 + \alpha_n$, this probability is bounded by $\binom{n}{k} p^{\binom{k}{2}} \le \left(np^{k/2}\right)^k$. We may choose $\eta = 4/d$. Then, for d sufficiently large, $\alpha_n \le (c/2 - 1)\log 2$ and, using Lemma 10, we have

$$\begin{aligned}
\mathbb{P}\left\{\exists s \in S^{d-1} : N_k(s) \ge 1\right\} &\le (4/\eta)^d \left(np^{(c/2)\log_2 n}\right)^{c\log_2 n} \\
&\le e^{d\log d}\left(n^{1+(c/2)\log_2(1/2+\alpha_n)}\right)^{c\log_2 n} \\
&\le e^{d\log d}\left(n^{1-c/2+c\alpha_n/\log 2}\right)^{c\log_2 n} \\
&\le e^{d\log d} n^{(1-c/2)c(\log_2 n)/2} \\
&= e^{d\log d - (c-2)c(\log_2 n)^2 \log 2/4} ,
\end{aligned}$$

and the statement follows.

It remains to prove part (iv). The proof relies on the second moment method. Let $c > 2$, $c' > c^2/(2\log 2)$, and assume that $d \ge c'\log^2 n/\log\log n$. Let K be a constant satisfying

$K > 2/\sqrt{c'}$ and define $\theta = K\sqrt{\log\log n}/\log n$. Let A be a subset of S^{d-1} of cardinality at least $(d/16)\theta^{-(d-1)}$ such that for all distinct pairs $s, s' \in A$, we have $\langle s, s'\rangle \geq \cos(\theta)$. Such a set exists by Lemma 11. Also, let $\mathcal{C}$ be the family of all subsets of $[n]$ of cardinality $k = \lfloor c\log_2 n\rfloor$. For $s \in A$ and $\gamma \in \mathcal{C}$, denote by $Z_{s,\gamma}$ the indicator that all edges between vertices in γ are present in the graph $\Gamma(\boldsymbol{X}_n, s)$. Our aim is to show that $\lim_{n\to\infty}\mathbb{P}\{Z > 0\} = 1$ where

$$Z = \sum_{s\in A}\sum_{\gamma\in\mathcal{C}} Z_{s,\gamma} \ .$$

To this end, by the second moment method (see, e.g., Alon and Spencer 1992), it suffices to prove that $\mathbb{E}Z \to \infty$ and that $\mathbb{E}[Z^2] = (\mathbb{E}Z)^2(1 + o(1)$.

To bound $\mathbb{E}Z$ note that

$$\begin{aligned}
\mathbb{E}Z &= |A|\binom{n}{k}\mathbb{E}Z_{s,\gamma} \\
&\geq (d/16)\theta^{-(d-1)}\binom{n}{k}2^{-\binom{k}{2}} \\
&= \exp\left((\log n)^2\left(c' - \frac{c^2}{2\log 2} + \frac{c}{\log 2} + o(1)\right)\right) \to \infty \ .
\end{aligned}$$

On the other hand,

$$\begin{aligned}
\mathbb{E}[Z^2] &= \sum_{s,s'\in A}\sum_{\gamma,\gamma'\in\mathcal{C}}\mathbb{E}[Z_{s,\gamma}Z_{s',\gamma'}] \\
&= \sum_{s,s':s\neq s'\in A}\sum_{\gamma,\gamma'\in\mathcal{C}:|\gamma\cap\gamma'|\leq 1}\mathbb{E}[Z_{s,\gamma}Z_{s',\gamma'}] + \sum_{s\in A}\sum_{\gamma,\gamma'\in\mathcal{C}}\mathbb{E}[Z_{s,\gamma}Z_{s,\gamma'}] \\
&\quad + \sum_{s,s':s\neq s'\in A}\sum_{\gamma,\gamma'\in\mathcal{C}:|\gamma\cap\gamma'|\geq 2}\mathbb{E}[Z_{s,\gamma}Z_{s',\gamma'}] \\
&\stackrel{\text{def}}{=} I + II + III \ .
\end{aligned}$$

For the first term note that if γ and γ' intersect in at most one vertex then $Z_{s,\gamma}$ and $Z_{s',\gamma'}$ are independent and therefore

$$I = \sum_{s,s':s\neq s'\in A}\sum_{\gamma,\gamma'\in\mathcal{C}:|\gamma\cap\gamma'|\leq 1}\mathbb{E}Z_{s,\gamma}\mathbb{E}Z_{s',\gamma'} \leq (\mathbb{E}Z)^2 \ .$$

Hence, it suffices to prove that $II + III = o((\mathbb{E}Z)^2)$. To deal with II, we have

$$\begin{aligned}
\frac{II}{(\mathbb{E}Z)^2} &= \frac{1}{|A|\cdot\binom{n}{k}}\sum_{\ell=0}^{k}2^{\binom{\ell}{2}}\binom{n-k}{k-\ell}\binom{k}{\ell} \\
&\leq \frac{1}{|A|}\sum_{\ell=0}^{k}2^{\binom{\ell}{2}}\frac{k^{2\ell}}{(n-2k)^\ell\ell!} \\
&\leq \frac{1}{|A|}2^{\binom{\ell}{2}}\sum_{\ell=0}^{\infty}\left(\frac{k^2}{n-2k}\right)^\ell\frac{1}{\ell!} \\
&= \exp\left(-(\log n)^2\left(c' + o(1) - c^2/(2\log 2)\right)\right) \to 0 \ .
\end{aligned}$$

We now take care of III. To this end, we bound

$$\max_{\substack{s,s' \in A: s \neq s' \\ \gamma,\gamma': |\gamma \cap \gamma'| = \ell}} \mathbb{E}[Z_{s,\gamma} Z_{s',\gamma'}]$$

by

$$2^{\binom{\ell}{2} - 2\binom{k}{2} + 1} \mathbb{P}\left\{ \left\langle \frac{N}{\|N\|}, s_0 \right\rangle \geq \sin(\theta/2) \right\}^{\binom{\ell}{2}} ,$$

where N is a standard normal vector in $\mathbb{R}^d$. To see this, note that $2\binom{k}{2} - \binom{\ell}{2}$ edges of the two cliques occur independently, each with probability $1/2$. The remaining $\binom{\ell}{2}$ edges must be in both $\Gamma(\boldsymbol{X}_n, s)$ and $\Gamma(\boldsymbol{X}_n, s')$. A moment of thought reveals that this probability is bounded by the probability that the angle between a random normal vector and a fixed unit vector (say s_0) is less than $\pi/2 - \theta/2$. This probability may be bounded as

$$\begin{aligned}
\mathbb{P}\left\{ \langle N/\|N\|, s_0 \rangle \geq \sin(\theta/2) \right\} &= \frac{1}{2} \mathbb{P}\left\{ B \geq \sin^2(\theta/2) \right\} \\
&\quad \text{(where } B \text{ is a Beta}(1/2, (d-1)/2) \text{ random variable)} \\
&\leq \frac{\mathbb{E}B}{2\sin^2(\theta/2)} \\
&= \frac{1}{2d\sin^2(\theta/2)} \\
&= \frac{2 + o(1)}{d\theta^2} = \frac{2 + o(1)}{c' K^2} .
\end{aligned}$$

Via the same counting argument used in handling II, we have

$$\frac{III}{(\mathbb{E}Z)^2} \leq \sum_{\ell=2}^{k} 2^{\binom{\ell}{2}} \left(\frac{2 + o(1)}{c' K^2} \right)^{\binom{\ell}{2}} \left(\frac{k^2}{n - 2k} \right)^{\ell} \frac{1}{\ell!} .$$

Since $c' K^2 > 4$, we have, for n large enough,

$$\frac{III}{(\mathbb{E}Z)^2} \leq \sum_{\ell=2}^{k} \left(\frac{k^2}{n - 2k} \right)^{\ell} \frac{1}{\ell!} = O\left(\frac{(\log n)^2}{n^2} \right)$$

as required. This concludes the proof of the theorem. ■

We conclude the section by remarking that the above proof extends straightforwardly to $G(n,p)$ for any constant $p \in (0,1)$.

3. Chromatic Number

A proper coloring of vertices of a graph assigns a color to each vertex such that no pair of vertices joined by an edge share the same color. The *chromatic number* $\chi(G)$ of a graph G is the smallest number of colors for which a proper coloring of the graph exists.

Here we study the fluctuations of the chromatic numbers $\chi(\Gamma(\boldsymbol{X}_n, s))$ from its typical behavior as $s \in S^{d-1}$. Once again, for simplicity of the presentation, we consider $p = 1/2$. The arguments extend easily to other (constant) values of p.

For a fixed s, a celebrated result of Bollobás (1988) implies that

$$\frac{n}{2\log_2 n} \leq \chi(\Gamma(\boldsymbol{X}_n, s)) \leq \frac{n}{2\log_2 n}(1+o(1))$$

with high probability.

In this section we derive estimates for the value of the dimension d for which there exist random graphs in the collection $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ whose chromatic number differs substantially (i.e., by a constant factor) from that of a typical $G(n,1/2)$ graph. Similar to the case of the clique number studied in Section 2, we find that upper and lower deviations exhibit a different behavior—though in a less dramatic way. With high probability, one does not see a graph with a clique number larger than $(1+\epsilon)n/(2\log_2 n)$ unless d is at least $n^2/\,polylog\, n$. On the other hand, when d is roughly linear in n, there are graphs is $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$ with chromatic number at most $(1-\epsilon)n/(2\log_2 n)$. Below we make these statements rigorous and also show that they are essentially tight.

Theorem 3 (chromatic number.) *Let $\epsilon \in (0,1/2)$. If $\chi(\Gamma(\boldsymbol{X}_n, s))$ denotes the chromatic number of $\Gamma(\boldsymbol{X}_n, s)$, then, with high probability the following hold:*

(i) (subcritical; necessary.) *If $d = o(n/(\log n)^3)$, then for all $s \in S^{d-1}$, $\chi(\Gamma(\boldsymbol{X}_n, s)) \geq (1-\epsilon)n/(2\log_2 n)$.*

(ii) (subcritical; sufficient.) *If $d \geq 2n\log_2 n/(1-2\epsilon)$, then there exists $s \in S^{d-1}$ such that $\chi(\Gamma(\boldsymbol{X}_n, s)) \leq (1-\epsilon)n/(2\log_2 n)$.*

(iii) (supercritical; necessary.) *If $d = o(n^2/(\log n)^6)$, then we have that for all $s \in S^{d-1}$, $\chi(\Gamma(\boldsymbol{X}_n, s)) \leq (1+\epsilon)n/(2\log_2 n)$.*

(iv) (supercritical; sufficient.) *If $d \geq .5\left[(1+\epsilon)n/(2\log_2 n)\right]^2$, then there exists $s \in S^{d-1}$ such that $\chi(\Gamma(\boldsymbol{X}_n, s)) \geq (1+\epsilon)n/(2\log_2 n)$.*

Part (i) of Theorem 3 follows from the following "uniform concentration" argument.

Proposition 4 *If $d = o(n/(\log n)^3)$, we have*

$$\sup_{s\in S^{d-1}} \left| \chi(\Gamma(\boldsymbol{X}_n, s)) - \frac{n}{2\log_2 n} \right| = o_p\left(\frac{n}{\log_2 n}\right) \ ,$$

Proof A classical result of Shamir and Spencer (1987) shows that for any fixed $s \in S^{d-1}$,

$$|\chi(\Gamma(\boldsymbol{X}_n, s)) - \mathbb{E}\left(\chi(\Gamma(\boldsymbol{X}_n, s))\right)| = O_p(n^{1/2}) \ .$$

In fact, one may easily combine the above-mentioned results of Bollobás and Shamir and Spencer to obtain that

$$\frac{\mathbb{E}\chi(\Gamma(\boldsymbol{X}_n, s))}{n/(2\log_2 n)} \to 1 \ .$$

The proof of the proposition is based on combining the Shamir-Spencer concentration argument with Vapnik-Chervonenkis-style symmetrization.

For each $s \in S^{d-1}$ and $i = 2, \ldots, n$, define $Y_{i,s} = (\mathbb{1}_{\{\langle X_{i,j}, s\rangle \geq 0\}})_{j=1,\ldots,i-1} \in \{0,1\}^{i-1}$ as the collection of indicators of edges connecting vertex i smaller-labeled vertices in $\Gamma(\boldsymbol{X}_n, s)$. As Shamir and Spencer, we consider the chromatic number $\Gamma(\boldsymbol{X}_n, s)$ as a function of these variables and define the function $f : \prod_{i=2}^{n} \{0,1\}^{i-1} \to \mathbb{N}$ by

$$f(Y_{2,s}, \ldots, Y_{n,s}) = \chi(\Gamma(\boldsymbol{X}_n, s)) \ .$$

By Markov's inequality, it suffices to show that

$$\mathbb{E}\left[\sup_{s \in S^{d-1}} |f(Y_{2,s}, \ldots, Y_{n,s}) - \mathbb{E} f(Y_{2,s}, \ldots, Y_{n,s})|\right] = o\left(\frac{n}{\log n}\right) \ .$$

Let $\boldsymbol{X}'_n = (X'_{i,j})_{1 \leq i < j \leq n}$ be an independent copy of $\boldsymbol{X}_n$. Denote by $\mathbb{E}'$ conditional expectation given $\boldsymbol{X}_n$. We write $Y'_{i,s} = (\mathbb{1}_{\{\langle X'_{i,j}, s\rangle \geq 0\}})_{j=1,\ldots,i-1} \in \{0,1\}^{i-1}$.

Also introduce random "swap operators" $\epsilon_2, \ldots, \epsilon_n$ defined by

$$\epsilon_i(Y_{i,s}, Y'_{i,s}) = \begin{cases} Y_{i,s} & \text{with probability } 1/2 \\ Y'_{i,s} & \text{with probability } 1/2 \end{cases}$$

where the ϵ_i are independent of each other and of everything else.

$$\begin{aligned}
&\mathbb{E}\left[\sup_{s \in S^{d-1}} |f(Y_{2,s}, \ldots, Y_{n,s}) - \mathbb{E} f(Y_{2,s}, \ldots, Y_{n,s})|\right] \\
&= \mathbb{E}\left[\sup_{s \in S^{d-1}} \left|\mathbb{E}'\left(f(Y_{2,s}, \ldots, Y_{n,s}) - f(Y'_{2,s}, \ldots, Y'_{n,s})\right)\right|\right] \\
&\leq \mathbb{E}\left[\sup_{s \in S^{d-1}} \left|f(Y_{2,s}, \ldots, Y_{n,s}) - f(Y'_{2,s}, \ldots, Y'_{n,s})\right|\right] \\
&= \mathbb{E}\left[\sup_{s \in S^{d-1}} \left|f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s})) - f(\epsilon_2(Y'_{2,s}, Y_{2,s}), \ldots, \epsilon_n(Y'_{n,s}, Y_{n,s}))\right|\right] \ .
\end{aligned}$$

Introduce now the expectation operator $\mathbb{E}_\epsilon$ that computes expectation with respect to the random swaps only. Then we can further bound the expectation above by

$$2\mathbb{E}\mathbb{E}_\epsilon\left[\sup_{s \in S^{d-1}} \left|f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s})) - \mathbb{E}_\epsilon f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s}))\right|\right] \ .$$

Next we bound the inner expectation. Note that for fixed $\boldsymbol{X}_n, \boldsymbol{X}'_n$, by Lemma 1, there are at most n^{2d} different dichotomies of the $2\binom{n}{2}$ points in $\boldsymbol{X}_n \cup \boldsymbol{X}'_n$ by hyperplanes including the origin and therefore there are not more than n^{2d} random variables of the form $f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s}))$ as s varies over S^{d-1}. On the other hand, for any fixed s, the value of $f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s}))$ can change by at most 1 if one flips the value of one of the $\epsilon_i(Y_{i,s}, Y'_{i,s})$ $(i = 2, \ldots, n)$, since such a flip amounts to changing the edges incident to vertex i and therefore can change the value of the chromatic number by at most

one. Thus, by the bounded differences inequality (see, e.g., Boucheron et al. 2013, Section 6.1), for all $s \in S^{d-1}$ and $\lambda > 0$,

$$\mathbb{E}_\epsilon \left[\exp\left(\lambda(f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s})) - \mathbb{E}_\epsilon f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s})) \right) \right) \right] \leq \exp\left(\frac{(n-1)\lambda^2}{2} \right) .$$

Therefore, by a standard maximal inequality for sub-Gaussian random variables (Boucheron et al., 2013, Section 2.5),

$$\mathbb{E}_\epsilon \left[\sup_{s \in S^{d-1}} \left| f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s})) - \mathbb{E}_\epsilon f(\epsilon_2(Y_{2,s}, Y'_{2,s}), \ldots, \epsilon_n(Y_{n,s}, Y'_{n,s})) \right| \right] \leq \sqrt{4(n-1)d \log n} .$$

Since the upper bound is $o(n/\log n)$ for $d = o(n/\log^3 n)$, the result follows. ■

Parts (ii) and (iv) of Theorem 3 follow from the next, straightforward proposition by setting $k = \lfloor (1-\epsilon)n/(2\log_2 n) \rfloor$ and $k' = \lceil (1+\epsilon)n/(2\log_2 n) \rceil$.

Proposition 5 *Let $k, k' \leq n$ be positive integers. If $d \geq k\binom{\lceil n/k \rceil}{2}$, then, with probability one, there exists $s \in S^{d-1}$ such that $\chi(\Gamma(\boldsymbol{X}_n, s)) \leq k$. On the other hand, if $d \geq \binom{k'}{2}$, then, with probability one, there exists $s \in S^{d-1}$ such that $\chi(\Gamma(\boldsymbol{X}_n, s)) \geq k'$.*

Proof Partition the vertex set $[n]$ into k disjoint sets of size at most $\lceil n/k \rceil$ each. If for some $s \in S^{d-1}$ each of these sets is an independent set (i.e., contain no edge joining two vertices within the set) in $\Gamma(\boldsymbol{X}_n, s)$, then the graph $\Gamma(\boldsymbol{X}_n, s)$ is clearly properly colorable with k colors. Let A be the set of pairs of vertices (i, j) such that i and j belong to the same set of the partition. By Lemma 1, if $d \geq k\binom{\lceil n/k \rceil}{2} \geq |A|$, the set of points $\{X_{i,j} : (i,j) \in A\}$ is shattered by half spaces. In particular, there exists an $s \in S^{d-1}$ such that $\langle X_{i,j}, s \rangle < 0$ for all $(i, j) \in A$ and therefore $\Gamma(\boldsymbol{X}_n, s)$ has no edge between any two vertices in the same set. The first statement follows.

To prove the second statement, simply notice that is a graph has a clique of size k then its chromatic number at least k. But if $d \geq \binom{k}{2}$, then, by Lemma 1, for some $s \in S^{d-1}$, the vertex set $\{1, \ldots, k\}$ forms a clique. ■

It remains to prove Part (iii) of Theorem 3. To this end, we combine the covering argument used in parts (i) and (iii) of Theorem 2 with a result of Alon and Sudakov (2010) (see Proposition 18 below) that bounds the "resilience" of the chromatic number of a random graph.

Let $\mathcal{C}_\eta$ be a minimal η-cover of S^{d-1} where we take $\eta = c\epsilon^2/(\sqrt{d}\log^2 n)$ for a sufficiently small positive constant c. Then

$$\begin{aligned} &\mathbb{P}\left\{ \exists s \in S^{d-1} : \chi(\Gamma(\boldsymbol{X}_n, s)) > (1+\epsilon)\frac{n}{2\log_2 n} \right\} \\ &\quad \leq \quad |\mathcal{C}_\eta| \mathbb{P}\left\{ \exists s \in S^{d-1} : \|s - s_0\| \leq \eta : \chi(\Gamma(\boldsymbol{X}_n, s)) > (1+\epsilon)\frac{n}{2\log_2 n} \right\} \end{aligned}$$

where $s_0 = (1, 0, \ldots, 0)$. By the argument used in the proof of parts (i) and (iii) of Theorem 2,

$$\bigcup_{s \in S^{d-1} : \|s - s_0\| \le \eta} \Gamma(\boldsymbol{X}_n, s) \subset \Gamma(\boldsymbol{X}_n, s_0) \cup E$$

where E is a set of $\mathrm{Bin}(\binom{n}{2}, \alpha_n)$ edges where, $\alpha_n = \frac{\eta\sqrt{d}}{\sqrt{2\pi}}$. By our choice of η, we have $\alpha_n \le c_2 \epsilon^2 n^2/(\log_2 n)^2$ where c_2 is the constant appearing in Proposition 18. Thus, by the Chernoff bound,

$$\mathbb{P}\left\{ |E| > \frac{c_2 \epsilon^2 n^2}{(\log_2 n)^2} \right\} \le \exp\left(- \frac{c_2 (\log 2 - 1/2) \epsilon^2 n^2}{(\log_2 n)^2} \right) .$$

Hence, by Proposition 18,

$$\begin{aligned} \mathbb{P}\left\{ \exists s \in S^{d-1} : \|s - s_0\| \le \eta : \chi(\Gamma(\boldsymbol{X}_n, s)) > (1+\epsilon) \frac{n}{2 \log_2 n} \right\} \\ \le \exp\left(- \frac{c_2 (\log 2 - 1/2) \epsilon^2 n^2}{(\log_2 n)^2} \right) + \exp\left(- \frac{c_1 n^2}{(\log_2 n)^4} \right) . \end{aligned}$$

Combining this bound with Lemma 10 implies the statement. □

4. Connectivity

In this section we study connectivity of the random graphs in $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$. It is well known since the pioneering work of Erdős and Rényi (1960) that the threshold for connectivity for a $G(n,p)$ random graph is when $p = c \log n/n$. For $c < 1$, the graph is disconnected and for $c > 1$ it is connected, with high probability. In this section we investigate both regimes. In particular, for $c > 1$ we establish lower and upper bounds for the smallest dimension d such that some graph in $\mathbb{G}_{d, c \log n/n}(\boldsymbol{X}_n)$ is disconnected. We prove that this value of d is of the order of $(c-1) \log n / \log\log n$. For the regime $c < 1$ we also establish lower and upper bounds for the smallest dimension d such that some graph in $\mathbb{G}_{d, c \log n/n}(\boldsymbol{X}_n)$ is connected. As in the case of the clique number and chromatic number, here as well we observe a large degree of asymmetry. In order to witness some connected graphs in $\mathbb{G}_{d, c \log n/n}(\boldsymbol{X}_n)$, the dimension d has to be at least of the order of n^{1-c}. While we suspect that this bound is essentially tight, we do not have a matching upper bound. However, we are able to show that when d is of the order of $n \log n$, the family $\mathbb{G}_{d, c \log n/n}(\boldsymbol{X}_n)$ not only contains connected graphs, but also, with high probability, for every spanning tree of the vertices $[n]$, there exists an $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}_n, s, t)$ contains the spanning tree. (Recall that t is such that $p = 1 - \Phi(t)$.)

Theorem 6 (connectivity.) *Assume $p = c \log n/n$ and let $t = \Phi^{-1}(1-p)$. Then with high probability the following hold:*

(i) (subcritical; necessary.) *If $c < 1$ then for any $\epsilon > 0$, if $d = O(n^{1-c-\epsilon})$, then for all $s \in S^{d-1}$, $\Gamma(\boldsymbol{X}_n, s, t)$ is disconnected.*

(ii) (subcritical; sufficient.) *There exists an absolute constant C such that if $d \ge Cn\sqrt{\log n}$, then there exists an $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}_n, s, t)$ is connected.*

(iii) (SUPERCRITICAL; NECESSARY.) *If $c > 1$ then for any $\epsilon > 0$, if $d \le (1-\epsilon)(c-1)\log n/\log\log n$, then for all $s \in S^{d-1}$, $\Gamma(\boldsymbol{X}_n, s, t)$ is connected.*

(iv) (SUPERCRITICAL; SUFFICIENT.) *If $c > 1$ then for any $\epsilon > 0$, if $d \ge (2+\epsilon)(c-1)\log n/\log\log n$, then for some $s \in S^{d-1}$, $\Gamma(\boldsymbol{X}_n, s, t)$ is disconnected.*

4.1 Proof of Theorem 6, Part (i)

To prove part (i), we show that when $d = O(n^{1-c-\epsilon})$, with high probability, all graphs $\Gamma(\boldsymbol{X}_n, s, t)$ contain at least one isolated point. The proof of this is based on a covering argument similar those used in parts of Theorems 2 and 3, combined with a sharp estimate for the probability that $G(n, c\log n/n)$ has no isolated vertex. This estimate, given in Lemma 19 below, is proved by an elegant argument of O'Connell (1998).

Let $\eta \in (0,1]$ to be specified below and let $\mathcal{C}_\eta$ be a minimal η-cover of S^{d-1}. If $N(s)$ denotes the number of isolated vertices (i.e., vertices of degree 0) in $\Gamma(\boldsymbol{X}_n, s, t)$, then

$$\begin{aligned}
&\mathbb{P}\left\{\exists s \in S^{d-1} : \Gamma(\boldsymbol{X}_n, s, t) \text{ is connected}\right\} \\
&\quad\le\ \mathbb{P}\left\{\exists s \in S^{d-1} : N(s) = 0\right\} \\
&\quad\le\ |\mathcal{C}_\eta|\mathbb{P}\left\{\exists s \in S^{d-1} : \|s - s_0\| \le \eta : N(s) = 0\right\}
\end{aligned}$$

where $s_0 = (1, 0, \ldots, 0)$. It follows by the first half of Lemma 13 that there exists a constant $\kappa > 0$ such that if $\eta = \kappa\epsilon/(t\sqrt{d})$, then

$$\mathbb{P}\left\{\exists s \in S^{d-1} : \|s - s_0\| \le \eta : N_k(s) = 0\right\} \le \mathbb{P}\{N = 0\}$$

where N is the number of isolated vertices in a $G(n, (c+\epsilon/2)\log n/n)$ random graph. By Lemma 19, for n sufficiently large, this is at most $\exp(-n^{-(1-c-\epsilon/2)}/3)$. Bounding $|\mathcal{C}_\eta|$ by Lemma 10 and substituting the chosen value of η proves part (i).

4.2 Proof of Theorem 6, Part (ii)

Part (ii) of Theorem 6 follows from a significantly more general statement. Based on a geometrical argument, we show that for any positive integer k, if d is at least a sufficiently large constant multiple of $k\Phi^{-1}(1-p)$, then with high probability, k independent standard normal vectors in $\mathbb{R}^d$ are shattered by half spaces of the form $\{x : \langle x, s\rangle \ge t\}$. In particular, by taking $k = n-1$ and considering the normal vectors $X_{i,j}$ corresponding to the edges of any fixed spanning tree, one finds an $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}_n, s, t)$ contains all edges of the spanning tree, making the graph connected. Note that if $d \ge Cn\sqrt{\alpha\log n}$ then the same statement holds whenever $p = n^{-\alpha}$ regardless of how large α is. Thus, for $d \gg n\sqrt{\log n}$, some $\Gamma(\boldsymbol{X}_n, s, t)$ are connected, even though for a typical s, the graph is empty with high probability.

Fix a set E of edges of the complete graph K_n. We say that $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$ shatters E if $\{e(G) : G \in \mathbb{G}_{d,p}(\boldsymbol{X}_n)\}$ shatters E (where $e(G)$ denotes the set of edges of a graph G). In other words, $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$ shatters E if for all $F \subset E$ there is $G \in \mathbb{G}_{d,p}(\boldsymbol{X}_n)$ such that $e(G) \cap E = F$.

Proposition 7 *Fix $n \in \mathbb{N}$, $k \in \{1, 2, \ldots, \binom{n}{2}\}$, and a set $E = \{e_1, \ldots, e_k\}$ of edges of the complete graph K_n. There exist universal constants $b, c > 0$ such that for $d \geq (4/c) \cdot k \cdot \Phi^{-1}(1-p)$ we have*

$$\mathbb{P}\left(\mathbb{G}_{d,p}(\boldsymbol{X}_n) \text{ shatters } E\right) \geq 1 - e^{-bd} \ .$$

Proof Given points $x_1, \ldots, x_k$ in $\mathbb{R}^d$, the *affine span* of $x_1, \ldots, x_k$ is the set $\{\sum_{i=1}^k c_i X_i : \sum_{i=1}^k c_i = 1\}$. Fix $E = \{e_1, \ldots, e_k\} \in \mathcal{S}_k$ and let P_E be the affine span of $X_{e_1}, \ldots, X_{e_k}$. Also, let $t = \Phi^{-1}(1-p)$.

First suppose that $\min\{\|y\| : y \in P_E\} > t$. Then we may shatter E as follows. First, almost surely, P_E is a $(k-1)$-dimensional affine subspace in $\mathbb{R}^d$. Assuming this occurs, then E is shattered by halfspaces in P_E: in other words, for any $F \subset E$ there is a $(k-2)$-dimensional subspace H contained within P_E such that F and $E \setminus F$ lie on opposite sides of H in P_E (i.e., in different connected components of $P_E \setminus H$).

Fix $F \subset E$ and $H \subset P_E$ as in the preceding paragraph. Then let K be a $(d-1)$-dimensional hyperplane tangent to $tS^{d-1} = \{x \in \mathbb{R}^d : \|x\| = t\}$, intersecting P_E only at H, and separating the origin from F. In other words, K is such that $K \cap P_E = H$ and $|K \cap tS^{d-1}| = 1$, and also such that 0 and F lie on opposite sides of K of $\mathbb{R}^d \setminus K$. Since P_E has dimension $k-1 < d-2$, such a hyperplane K exists. Since F and $E \setminus F$ lie on opposite sides of H, we also obtain that 0 and $E \setminus F$ lie on the *same* side of K.

Let $s \in S^{d-1}$ be such that $ts \in K$. Then for $e \in F$ we have $\langle X_e, s\rangle > t$, and for $e \in E \setminus F$ we have $\langle X_e, s\rangle < t$. It follows that $E \cap \Gamma(\boldsymbol{X}, s, t) = F$. Since $F \subset E$ was arbitrary, this implies that

$$\mathbb{P}(\mathbb{G}_{d,p}(\boldsymbol{X}_n) \text{ shatters } E) \geq \mathbb{P}(\min\{\|y\| : y \in P_E\} > \Phi^{-1}(1-p)) \, ,$$

In light of the assumption that $d \geq (4/c) \cdot k \cdot \Phi^{-1}(1-p)$, the proposition is then immediate from Lemma 8 below. ■

The key element of the proof of Proposition 7 is that the affine span of $k \leq 4d$ independent standard normal vectors in $\mathbb{R}^d$ is at least at distance of the order of d/k from the origin. This is made precise in the following lemma whose proof crucially uses a sharp estimate for the smallest singular value of a $d \times k$ Wishart matrix, due to Rudelson and Vershynin (2009).

Lemma 8 *There exist universal constants $b, c > 0$ such that the following holds. Let $N_1, \ldots, N_k$ be independent standard normal vectors in $\mathbb{R}^d$, let P be the affine span of $N_1, \ldots, N_k$, and let $D = \min\{\|y\| : y \in P\}$. Then whenever $d \geq 4k$, we have $\mathbb{P}(D \leq cd/4k) < 2e^{-bd}$.*

Proof We use the notation $\boldsymbol{y} = (y_1, \ldots, y_k)$. We have

$$\begin{aligned} D &= \min_{\boldsymbol{y}:\sum y_i=1} \left\| \sum_{i=1}^k y_i N_i \right\|^2 \\ &= \min_{\boldsymbol{y}:\sum y_i=1} |\boldsymbol{y}|^2 \left\| \sum_{i=1}^k \frac{y_i}{\|\boldsymbol{y}\|} N_i \right\|^2 \\ &\geq \min_{\boldsymbol{y}:\sum y_i=1} \frac{1}{k} \left\| \sum_{i=1}^k \frac{y_i}{\|\boldsymbol{y}\|} N_i \right\|^2 \\ &\geq \frac{1}{k} \min_{\boldsymbol{y}:|\boldsymbol{y}|^2=1} \left\| \sum_{i=1}^k y_i N_i \right\|^2, \end{aligned}$$

where the first inequality holds because if $\sum_{i=1}^k y_i = 1$ then $\|\boldsymbol{y}\|^2 \geq k^{-1}$ and the second by noting that the vector $(y_i/\|\boldsymbol{y}\|, 1 \leq i \leq k)$ has 2-norm 1.

Let $\boldsymbol{N}$ be the $d \times k$ matrix with columns $N_1^t, \ldots, N_k^t$, and write $\boldsymbol{N} = (N_{ij})_{ij \in [d] \times [k]}$. Then

$$\begin{aligned} \min_{\boldsymbol{y}:|\boldsymbol{y}|^2=1} \left\| \sum_{i=1}^k y_i N_i \right\|^2 &= \left(\min_{\boldsymbol{y}:|\boldsymbol{y}|^2=1} \left\| \sum_{i=1}^k y_i N_i \right\| \right)^2 \\ &= \left(\min_{\boldsymbol{y}:|\boldsymbol{y}|^2=1} \|\boldsymbol{X} y\| \right)^2. \end{aligned}$$

The final quantity is just the square of the least singular value of $\boldsymbol{X}$. Theorem 1.1 of Rudelson and Vershynin (2009) states the existence of absolute constants $b, B > 0$ such that for every $\varepsilon > 0$ we have

$$\mathbb{P}\left(\min_{\boldsymbol{y}:|\boldsymbol{y}|=1} \|\boldsymbol{X} y\| \leq \varepsilon(\sqrt{d} - \sqrt{k-1}) \right) \leq (B\varepsilon)^{(d-k+1)} + e^{-bd}.$$

If $d \geq 4(k-1)$ then $\sqrt{d} - \sqrt{k-1} \geq \sqrt{d}/2$ and $d - k + 1 > d$. Combining the preceding probability bound with the lower bound on D, if $\varepsilon \leq e^{-b}/B$ we then obtain

$$\mathbb{P}\left(D < \varepsilon^2 \frac{d}{4k} \right) < 2e^{-bd}.$$

Taking $c = (e^{-b}/B)^2$ completes the proof. ■

One may now easily use Proposition 7 to deduce part (ii) of Theorem 6:

Proposition 9 *There are absolute constants $b, C > 0$ such that the following holds. For all $p \leq 1/2$, if $d \geq Cn\sqrt{\log(1/p)}$ then with probability at least $1 - e^{-bd}$ there exists $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}, s, \Phi^{-1}(1-p))$ is connected.*

Proof Fix any tree T with vertices $[n]$, and write E for the edge set of T. By Proposition 7, if $d \geq (4/c) \cdot k \cdot \Phi^{-1}(1-p)$ then with probability at least $1 - e^{-bd}$ there is s such that $\Gamma(\boldsymbol{X}, s, \Phi^{-1}(1-p))$ contains T, so in particular is connected. Now simply observe that for $p \leq 1/2$ we have $\Phi^{-1}(1-p) \leq \sqrt{2\log(1/p)}$. ■

Observe that the exponentially small failure probability stipulated in Proposition 9 allows us to conclude that if d is at least a sufficiently large constant multiple of $n(\log n \vee \sqrt{\log(1/p)})$, then, with high probability, for *any* spanning tree of the complete graph K_n there exists $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}, s, \Phi^{-1}(1-p))$ contains that spanning tree.

4.3 Proof of Theorem 6, Part (iii)

Let $c > 1$, $\epsilon \in (0,1)$, and assume that $d \leq (1-\epsilon)(c-1)\log n/\log\log n$. Let E be the event that $\Gamma(\boldsymbol{X}_n, t, s)$ is disconnected for some $s \in S^{d-1}$. Let $\mathcal{C}_\eta$ be a minimal η-cover of S^{d-1} for $\eta \in (0,1]$ to be specified below. Then

$$E \subseteq \bigcup_{s \in \mathcal{C}_\eta} E_s \ ,$$

where E_s is the event that the graph $\bigcap_{s' : \|s - s'\| \leq \eta} \Gamma(\boldsymbol{X}_n, t, s')$ is disconnected. Let $c' = c - (c-1)\epsilon/2$. Note that $1 < c' < c$. It follows from the second half of Lemma 13 that there exists a constant $\kappa > 0$ such that if $\eta = \kappa(1 - c'/c)/(t\sqrt{d})$, then

$$\mathbb{P}\{E_s\} \leq \mathbb{P}\left\{G(n, c'\log n/n) \text{ is disconnected}\right\} \leq n^{1-c'}(1 + o(1)) \ ,$$

where the second inequality follows from standard estimates for the probability that a random graph is disconnected, see (Palmer, 1985, Section 4.3). Bounding $|\mathcal{C}_\eta|$ by Lemma 10, and using the fact that $t = \sqrt{2\log n}(1 + o(1))$, we obtain that

$$\begin{aligned} \mathbb{P}\{E\} &\leq |\mathcal{C}_\eta| n^{1-c'}(1 + o(1)) \\ &\leq \exp\left(\frac{d\log\log n}{2} + \frac{d\log d}{2} + O(d) + (1-c')\log n\right) \to 0 \ , \end{aligned}$$

as desired.

4.4 Proof of Theorem 6, Part (iv)

Recall that $p = c\log n/n$ for $c > 1$ fixed, and that $t = \Phi^{-1}(p)$. Let $0 < \epsilon < 1$, and assume that $d \geq (2+\epsilon)(c-1)\log n/\log\log n$. Define $\theta \in (0, \pi/2)$ by $\theta = (\log n)^{-1/(2+\varepsilon)}$, so that $\log(1/\theta) = \log\log n/(2+\epsilon)$. Let $\mathcal{P}$ be a maximal θ-packing of S^{d-1}, that is, $\mathcal{P} \subset S^{d-1}$ is a set of maximal cardinality such that for all distinct $s, s' \in \mathcal{P}$ we have $\langle s, s' \rangle \leq \cos\theta$. By Lemma 11 we have that

$$|\mathcal{P}| \geq \frac{d}{16}\theta^{-(d-1)} \ .$$

It suffices to prove that for some $s \in \mathcal{P}$, $\Gamma(\boldsymbol{X}_n, s, t)$ contains an isolated vertex.

For each $s \in \mathcal{P}$, we write the number of isolated vertices in $\Gamma(\boldsymbol{X}_n, s, t)$ as

$$N(s) = \sum_{i=1}^{n} \prod_{j : j \neq i} Z_{i,j}(s),$$

where $Z_{i,j}(s)$ equals 1 if $\{i,j\}$ is not an edge in $\Gamma(\boldsymbol{X}_n, s, t)$ and is 0 otherwise. We use the second moment method to prove that $N \stackrel{\text{def}}{=} \sum_{s\in\mathcal{P}} N(s) > 0$ with high probability. This will establish the assertion of part (iv) since if $N > 0$ then there is $s \in S^{d-1}$ such that $\Gamma(\boldsymbol{X}_n, s, t)$ contains an isolated vertex.

To show that $N > 0$ with high probability, by the second moment method it suffices to prove that $\mathbb{E}N \to \infty$ and that $\mathbb{E}[N^2] = (\mathbb{E}N)^2(1 + o(1)$. First,

$$\mathbb{E}N = |\mathcal{P}| \cdot n \cdot \mathbb{P}\{\text{vertex 1 is isolated in } G(n,p)\} = |\mathcal{P}| \cdot n(1-p)^{n-1}.$$

The lower bound on $|\mathcal{P}|$ and the inequality $1 - p \le e^{-p} = n^{-c/n}$ together imply

$$\mathbb{E}N \ge \frac{d}{16}\theta^{-(d-1)}n^{1-c},$$

which tends to infinity by our choice of θ. We now turn to the second moment.

$$\mathbb{E}[N^2] = \sum_{s,s'\in\mathcal{P}} \sum_{i,j\in[n]} \prod_{k:k\neq i,\ell:\ell\neq j} Z_{i,k}(s)Z_{j,\ell}(s') \ .$$

When $s = s'$, separating the inner sum into diagonal and off-diagonal terms yields the identity

$$\sum_{i,j} \prod_{k\neq i,\ell\neq j} Z_{i,k}(s)Z_{j,\ell}(s) = n(1-p)^{n-1}+n(n-1)(1-p)^{2n-3} = n(1-p)^{n-1}\cdot[1+(n-1)(1-p)^{n-2}] \, .$$

Let $q = \sup_{s\neq s',s,s'\in\mathcal{P}} \mathbb{P}\{Z_{i,j}(s)Z_{i,j}(s') = 1\}$ be the greatest probability that an edge is absent in both $\Gamma(\boldsymbol{X}_n, s, t)$ and $\Gamma(\boldsymbol{X}_n, s', t)$. Then when $s \neq s'$, the inner sum is bounded by

$$nq^{n-1} + n(n-1) \cdot q \cdot (1-p)^{2n-4}.$$

Combining these bounds, we obtain that

$$\mathbb{E}[N^2] \le |\mathcal{P}|n(1-p)^{n-1}\cdot[1+(n-1)(1-p)^{n-2}]+|\mathcal{P}|(|\mathcal{P}|-1)\cdot[nq^{n-1}+n(n-1)\cdot q\cdot(1-p)^{2n-4}] \, .$$

The first term on the right is at most $\mathbb{E}N(1 + \mathbb{E}N/[(1-p)|\mathcal{P}|])$. The second is at most

$$|\mathcal{P}|^2n^2(1-p)^{2(n-1)}\cdot\left(\frac{1}{n}\Big(\frac{q}{(1-p)^2}\Big)^{n-1} + \frac{q}{(1-p)^2}\right) = (\mathbb{E}N)^2\cdot\left(\frac{1}{n}\Big(\frac{q}{(1-p)^2}\Big)^{n-1} + \frac{q}{(1-p)^2}\right) \, .$$

We will show below that $q \le (1-p)^2 \cdot (1 + o(p))$. Assuming this, the upper bounds on the two terms on the right together give

$$\frac{\mathbb{E}[N^2]}{[\mathbb{E}N]^2} \le \frac{1}{\mathbb{E}N}\left(1 + \frac{\mathbb{E}N}{(1-p)|\mathcal{P}|}\right) + n^{o(1)-1} + \frac{(1-\epsilon)\log n}{n} \ \to 1 \, ,$$

as required.

To prove the bound on q, fix $s, s' \in \mathcal{P}$ such that $q = \mathbb{P}\{Z_{i,j}(s)Z_{i,j}(s') = 1\}$. Using the definition of $Z_{i,j}(s)$ and $Z_{i,j}(s')$, we have

$$q = \mathbb{P}\left\{\{i,j\} \notin \Gamma(\boldsymbol{X}_n, s, t), \{i,j\} \notin \Gamma(\boldsymbol{X}_n, s', t), \right\} \, .$$

We may apply Lemma 14 to this quantity, noting that in our case $\theta = (\ln n)^{1/(2+\varepsilon)}$, $t = O(\sqrt{\ln n})$ and $\ln(1/t\,p) = (1+o(1))\ln n \gg \theta^{-2}$. This means that the Remark after the statement of the Lemma applies, and this gives precisely that $q \le (1-p)^2\,(1+o(p))$, as desired. ■

Acknowledgments

Louigi Addario-Berry was supported during this research by a Leverhulme Visiting Professorship and by an NSERC Discovery Grant. Luc Devroye was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada. Shankar Bhamidi has been partially supported by NSF-DMS grants 1105581 and 1310002 and SES grant 1357622. Gábor Lugosi and Roberto Imbuzeiro Oliveira gratefully acknowledge support from CNPq, Brazil via the *Ciência sem Fronteiras* grant # 401572/2014-5. Roberto Imbuzeiro Oliveira was also supported by a *Bolsa de Produtividade em Pesquisa* # 301662/2012-6 from CNPq and by the FAPESP Center for Neuromathematics (grant# 2013/ 07699-0 , FAPESP - S.Paulo Research Foundation). Gábor Lugosi was supported by the Spanish Ministry of Science and Technology grant MTM2012-37195

We thank two anonymous referees for a number of helpful comments that improved the exposition.

Appendix A. Appendix

Here we gather some of the technical tools used in the paper. In the first section we summarize results involving covering and packing results of the unit sphere that are essential in dealing with the random graph process $\mathbb{G}_{d,1/2}(\boldsymbol{X}_n)$. In Section A.2 we describe analogous results needed for studying $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$ for small values of p. These lemmas play an important role in the proof of Theorem 6. Finally, in Section A.4 we collect some results on $G(n,p)$ random graphs needed in our proofs.

A.1 Covering and Packing

Let $B(a,b) = \int_0^1 t^{a-1}(1-t)^{b-1}\mathrm{d}t$ be the beta function, and let $I_x(a,b)$ be the incomplete beta function,

$$I_x(a,b) = \frac{\int_0^x t^{a-1}(1-t)^{b-1}\mathrm{d}t}{B(a,b)}\,.$$

For $\alpha \in [0,\pi]$ and $s \in \mathbb{S}^{d-1}$, let

$$C_\alpha(s) = \{s' \in S^{d-1} : \langle s, s'\rangle \ge \cos\alpha\}$$

be the cap in S^{d-1} consisting of points at angle at most α from s. For $\alpha \le \pi/2$ the area of this cap (see, e.g., Li 2011) is

$$|C_\alpha(s)| = \frac{|S^{d-1}|}{2}\cdot I_{\sin^2\theta}\left(\frac{d-1}{2},\frac{1}{2}\right)\,. \tag{1}$$

We use the following standard estimate of the covering numbers of the Euclidean sphere (see, e.g., Matoušek 2002, Lemma 13.1.1).

Lemma 10 *For any $\eta \in (0,1]$ there exists a subset $\mathcal{C}_\eta$ of S^{d-1} of size at most $(4/\eta)^d$ such that for all $s \in S^{d-1}$ there exists $s' \in \mathcal{C}_\eta$ with $\|s - s'\| \leq \eta$.*

We now provide a rough lower bound on the number of points that can be packed in S^{d-1} while keeping all pairwise angles large.

Lemma 11 *For any $\theta \in (0, \pi/2)$ there exists a subset $\mathcal{P}_\theta$ of S^{d-1} of size at least*

$$\frac{d}{16}\theta^{-(d-1)}$$

such that for all distinct $s, s' \in \mathcal{P}_\theta$ we have $\langle s, s' \rangle \leq \cos\theta$.

Proof

First note that it suffices to consider $\theta < 1/2$ because otherwise the first bound dominates. Consider N independent standard normal vectors $X_1, \ldots, X_N$. Then $U_i = X_i/\|X_i\|$ $(i = 1, \ldots, N)$ are independent, uniformly distributed on S^{d-1}. Let

$$Z = \sum_{i=1}^{N} \mathbb{1}_{\{\min_{j:j\neq i} |\langle U_i, U_j\rangle| \leq \cos(\theta)\}}.$$

Denoting $\mathbb{P}\{|\langle U_i, U_j\rangle| > \cos(\theta)\} = \phi$,

$$\mathbb{E}Z = N(1-\phi)^N \geq N(1-\phi N) \geq N/2$$

whenever $\phi N \leq 1/2$. Since $Z \leq N$, this implies that

$$\mathbb{P}\left\{Z \geq \frac{N}{4}\right\} \geq \frac{\mathbb{E}Z - N/4}{N - N/4} \geq \frac{1}{3}$$

and therefore there exists a packing set A of cardinality $|A| \geq N/4$ as long as $\phi N \leq 1/2$. To study ϕ, note that

$$\phi = \mathbb{P}\left\{\frac{\sum_{j=1}^{d} Y_j Y_j'}{\|Y\| \cdot \|Y'\|} > \cos(\theta)\right\}$$

where $Y = (Y_1, \ldots, Y_d), Y' = (Y_1', \ldots, Y_d')$ are independent standard normal vectors. By rotational invariance, we may replace Y' by $(\|Y'\|, 0, \ldots, 0)$, and therefore

$$\begin{aligned}
\phi &= \mathbb{P}\left\{\frac{Y_1^2}{\|Y\|} > \cos^2(\theta)\right\} \\
&= \mathbb{P}\left\{B \leq \cos^2(\theta)\right\} \\
&\quad \text{(where } B \text{ is a Beta}(1/2, (d-1)/2) \text{ random variable)} \\
&\geq \frac{2\theta^{d-1}}{d-1}\,.
\end{aligned}$$

The result follows. ∎

The next lemma is used repeatedly in the proof of Theorem 2 and 3.

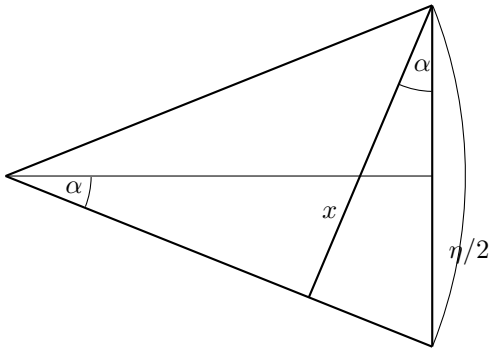


Figure 1: Since $\sin(\alpha) = \eta/2$, the height of the spherical cap that only includes points at distance at least η from the equator is $1 - \sin(2\alpha) = 1 - \eta\sqrt{1-(\eta/2)^2}$.

Lemma 12 *Fix $s' \in S^{d-1}$ and $\eta \in (0,1]$ and assume that $d \geq 12$. The probability that there exists $s \in S^{d-1}$ with $\|s - s'\| \leq \eta$ such that vertex 1 and vertex 2 are connected in $\Gamma(\boldsymbol{X}_n, s)$ but not in $\Gamma(\boldsymbol{X}_n, s')$ is at most*

$$\eta\sqrt{\frac{d}{2\pi}}\ .$$

Proof Without loss of generality, assume that $s' = (1, 0, \ldots, 0)$. Observe that the event that there exists $s' \in S^{d-1}$ with $\|s - s'\| \leq \eta$ such that vertex 1 and vertex 2 are connected in $\Gamma(\boldsymbol{X}_n, s)$ but not in $\Gamma(\boldsymbol{X}_n, s')$ is equivalent to $X_{1,2}/\|X_{1,2}\|$ having its first component between $-\eta\sqrt{1-\eta^2/4}$ and 0 (see Figure 1). Letting $Z = (Z_1, \ldots, Z_d)$ be a standard normal vector in $\mathbb{R}^d$, the probability of this is

$$\begin{aligned}
\mathbb{P}\left\{\frac{Z_1}{\|Z\|} \in \left(-\eta\sqrt{1-\eta^2/4}, 0\right)\right\} &\leq \mathbb{P}\left\{\frac{Z_1}{\|Z\|} \in (-\eta, 0)\right\} \\
&= \frac{1}{2}\mathbb{P}\left\{B \leq \eta^2\right\} \\
&\quad \text{(where } B \text{ is a Beta}(1/2, (d-1)/2) \text{ random variable)} \\
&= \frac{1}{2} I_{\eta^2}(1/2, (d-1)/2) \\
&\leq \frac{1}{2B(1/2,(d-1)/2)} \int_0^{\eta^2} x^{-1/2} dx \\
&= \frac{\eta}{2B(1/2,(d-1)/2)} \\
&\leq \eta\sqrt{\frac{d-1}{2\pi}}\ .
\end{aligned}$$

■

A.2 Auxiliary Results for $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$

In this section we develop some of the main tools for dealing with the random graph process $\mathbb{G}_{d,p}(\boldsymbol{X}_n)$. We assume throughout the section that

$$p := 1 - \Phi(t) \leq \frac{1}{2}. \tag{2}$$

Recall from the start of Section A.1 that $C_\alpha(s)$ denotes the spherical cap consisting of all unit vectors with an angle of $\leq \alpha$ with s. We will use the following expressions for $C_\alpha(s)$:

$$\begin{aligned} C_\alpha(s) &= \{s' \in S^{d-1} \,:\, \|s - s'\|^2 \leq 2\,(1 - \cos\alpha)\} \\ &= \{s\cos\theta + w\sin\theta \,:\, w \in S^{d-1} \cap \{v\}^\perp,\, 0 \leq \theta \leq \alpha\}. \end{aligned} \tag{3}$$

We are interested in studying the graphs $\Gamma(\boldsymbol{X}_n, s', t)$, for all $s' \in C_\alpha(s)$ simultaneously.

Lemma 13 *There exists a constant $c > 0$ such that, for all $\varepsilon \in (0, 1/2)$, if $t \geq 0$ and p are as in (2),*

$$0 \leq \alpha \leq \frac{\pi}{2},\, \tan\alpha \leq \frac{\varepsilon}{(t \vee 1)\sqrt{d-1}},$$

then, for some universal $c > 0$, if we define $\varepsilon' := \varepsilon + c\,(\varepsilon^2 + \varepsilon/(t^2 \vee 1))$,

1. *the union $\Gamma_+ := \bigcup_{s' \in C_\alpha(s)} \Gamma(\boldsymbol{X}_n, s', t)$ is stochastically dominated by $G(n, (1+\varepsilon')\,p)$;*
2. *the intersection $\Gamma_- := \bigcap_{s' \in C_\alpha(s)} \Gamma(\boldsymbol{X}_n, s', t)$ stochastically dominates by $G(n, (1-\varepsilon')\,p)$.*

Proof The *first step* in this argument is to note that the edges of both Γ_+ and Γ_- are independent. To see this, just notice that, for any $\{i,j\} \in \binom{[n]}{2}$, the event that $\{i,j\}$ is an edge in $\Gamma_\pm$ depends on $\boldsymbol{X}_n$ only through $X_{i,j}$. More specifically,

$$\begin{aligned} \{i,j\} \in \Gamma_+ &\;\Leftrightarrow\; \exists s' \in C_\alpha(s) \,:\, \langle X_{i,j}, s' \rangle \geq t; \\ \{i,j\} \in \Gamma_- &\;\Leftrightarrow\; \forall s' \in C_\alpha(s) \,:\, \langle X_{i,j}, s \rangle \geq t. \end{aligned}$$

The main consequence of independence is that we will be done once we show that

$$(1-\varepsilon')\,p \leq \mathbb{P}\{\{i,j\} \in \Gamma_-\} \leq \mathbb{P}\{\{i,j\} \in \Gamma_+\} \leq (1+\varepsilon')\,p. \tag{4}$$

As a *second step* in our proof, we analyze the inner product of $X_{i,j}$ with $s' = s\cos\theta + w\sin\theta \in C_\alpha(s)$ (with the same notation as in (3)). Note that

$$\left\langle s', X_{i,j} \right\rangle = N\cos\theta + \left\langle w, X_{i,j}^\perp \right\rangle \sin\theta = \cos\theta\,\left(N + \left\langle w, X_{i,j}^\perp \right\rangle \tan\theta\right),$$

where $N := \langle X_{i,j}, s \rangle$ and $X_{i,j}^\perp$ is the component of $X_{i,j}$ that is orthogonal to s. Crucially, the fact that $X_{i,j}$ is a standard Gaussian random vector implies that N is a standard normal random variable and $X_{i,j}^\perp$ is an independent standard normal random vector in $s^\perp$. Moreover,

$$\forall w \in S^{d-1} |\left\langle w, X_{i,j}^\perp \right\rangle| \leq \chi := \|X_{i,j}^\perp\|.$$

Since "$\theta \mapsto \tan\theta$" is increasing in $[0,\alpha]$, we conclude

$$\forall s' \in C_\alpha(s) \,:\, \left\langle s', X_{i,j}\right\rangle = \cos\theta\left(N + \Delta(s')\right), \text{ where } |\Delta(s')| \le (\tan\alpha)\,\chi. \tag{5}$$

Our *third step* is to relate the above to the events $\{\{i,j\} \in \Gamma_\pm\}$. On the one hand,

$$\begin{aligned}
\{i,j\} \in \Gamma_+ \quad &\Leftrightarrow \quad \max_{s' \in C_\alpha(s)} \left\langle s', X_{i,j}\right\rangle \ge t \\
&\Rightarrow \quad N + \max_{s' \in \mathcal{C}_\alpha(s)} \Delta(s') \ge t \quad \text{(use (5) and } 0 \le \cos\theta \le 1) \\
&\Rightarrow \quad N \ge t - (\tan\alpha)\,\chi,
\end{aligned}$$

and we conclude (using the independence of N and χ) that

$$\mathbb{P}\{\{i,j\} \in \Gamma_+\} \le 1 - \mathbb{E}[\Phi(t - (\tan\alpha)\,\chi)]. \tag{6}$$

Similarly,

$$\begin{aligned}
\{i,j\} \in \Gamma_- \quad &\Leftrightarrow \quad \min_{s' \in C_\alpha(s)} \left\langle s', X_{i,j}\right\rangle \ge t \\
&\Leftrightarrow \quad N + \min_{s' \in \mathcal{C}_\alpha(s)} \Delta(s') \ge \frac{t}{\cos\alpha} \quad \text{(by (5) and } \cos\theta \ge \cos\alpha > 0) \\
&\Leftarrow \quad N \ge \frac{t}{\cos\alpha} + (\tan\alpha)\,\chi,
\end{aligned}$$

and we conclude

$$\mathbb{P}\{\{i,j\} \in \Gamma_-\} \ge \mathbb{E}\left[1 - \Phi\left(\frac{t}{\cos\alpha} + (\tan\alpha)\,\chi\right)\right]. \tag{7}$$

The remainder of the proof splits into two cases, depending on whether or not

$$e^{\frac{5t^2}{8}}\,(1 - \Phi(t)) \ge 1 \tag{8}$$

Note that this condition holds if and only if $t \ge C$ for some $C > 0$, as $1 - \Phi(t) = e^{-(1+o(1))t^2/2}$ when $t \to +\infty$ and $e^{\frac{5t^2}{8}}\,(1 - \Phi(t)) = 1/2 < 1$ when $t = 0$.

Last step when (8) is violated. In this case t is bounded above, so $p > c_0$ for some positive constant $c_0 > 0$. We combine (6) and (7) with the fact that $\Phi(t)$ is $(2\pi)^{-1/2}$-Lipschitz. The upshot is that

$$|1 - \Phi(t) - \mathbb{P}\{\{i,j\} \in \Gamma_\pm\}| \le \frac{1}{\sqrt{\pi}}\left|1 - \frac{1}{\cos\alpha}\right| t + \mathbb{E}[\chi]\tan\alpha.$$

Now χ is the norm of a $d-1$ dimensional standard normal random vector, so $\mathbb{E}[\chi] \le \sqrt{\mathbb{E}[\chi^2]} = \sqrt{d-1}$. The choice of α implies:

$$\left|1 - \frac{1}{\cos\alpha}\right| = O(\sin\alpha) = O\left(\frac{\varepsilon^2}{d-1}\right), \text{ and } \tan\alpha \le \frac{\varepsilon}{\sqrt{d-1}}.$$

So

$$|1 - \Phi(t) - \mathbb{P}\{\{i,j\} \in \Gamma_\pm\}| \le \frac{1}{\sqrt{2\pi}} \left(c\,\varepsilon^2 + \varepsilon\right) \le \left[\varepsilon + c\,\left(\varepsilon^2 + \frac{\varepsilon}{t^2}\right)\right] p$$

for some universal $c > 0$.

Last step when (8) is satisfied. We start with (7) and note that we can apply Lemma 16 with $r := t$ and

$$h := \left(\frac{1}{\cos\alpha} - 1\right) t + (\tan\alpha)\,\chi \le O((\tan\alpha)^2)\,t + (\tan\alpha)\,\chi.$$

After simple calculations, this gives

$$\frac{\mathbb{P}\{\{i,j\} \in \Gamma_-\}}{1 - \Phi(t)} \ge \mathbb{E}\left[\exp\left(-X\right)\right],$$

where

$$X := O((\tan\alpha)^2)\,(t^2+1) - (t + t^{-1})\,(\tan\alpha)\chi - (\tan\alpha)^2\,\xi^2 - O((\tan\alpha)^2)t^2.$$

By Jensen's inequality, $\mathbb{E}[e^{-X}] \ge e^{-\mathbb{E}[X]}$. Since $\mathbb{E}[\chi]^2 \le \mathbb{E}[\chi^2] = d-1$ and $\tan\alpha = \varepsilon/t\,\sqrt{d-1}$ in this case,

$$\mathbb{E}[X] \le O\left(\frac{\varepsilon^2}{d-1}\right) + (1 + O(\varepsilon + t^{-2}))\,\varepsilon.$$

In other words, if we choose $c > 0$ in the statement of the theorem to be large enough, we can ensure that

$$\frac{\mathbb{P}\{\{i,j\} \in \Gamma_-\}}{1 - \Phi(t)} \ge (1 - \varepsilon').$$

We now turn to (6). Applying Lemma 16 below with $r := t - \chi\tan\alpha$ when $r \ge t/2$, we get

$$1 - \Phi(t - (\tan\alpha)\,\chi) \le e^{\left(t + \frac{2}{t}\right)(\tan\alpha)\,\chi + \frac{(\tan\alpha)^2\,\chi^2}{2}}\,(1 - \Phi(t)). \tag{9}$$

In fact, the same inequality holds when $r < t/2$, i.e., $(\tan\alpha)\,\chi > t/2$, for in that case the right-hand side is $\ge e^{\frac{5t^2}{8}}\,(1 - \Phi(t)) \ge 1$ (recall that we are under the assumption (8)). So (9) always holds, and integration over χ gives

$$\frac{\mathbb{P}\{\{i,j\} \in \Gamma_+\}}{1 - \Phi(t)} \le \mathbb{E}[e^{\left(t + \frac{2}{t}\right)(\tan\alpha)\,\chi + \frac{(\tan\alpha)^2\,\chi^2}{2}}]. \tag{10}$$

It remains to estimate the moment generating function on the right-hand side. The first step is to note that, since $\mathbb{E}[\xi]$ is the norm of a $d-1$ dimensional standard normal vector, $\mathbb{E}[\chi] \le \mathbb{E}[\chi^2]^{1/2} = \sqrt{d-1}$. So by Cauchy Schwartz,

$$\begin{aligned} & e^{-\left(t + \frac{2}{t}\right)(\tan\alpha)\sqrt{d-1}}\,\mathbb{E}[e^{\left(t + \frac{2}{t}\right)(\tan\alpha)\,\chi + \frac{(\tan\alpha)^2\,\chi^2}{2}}] \\ \le\ & \mathbb{E}[e^{\left(t + \frac{2}{t}\right)(\tan\alpha)\,(\chi - \mathbb{E}[\chi]) + \frac{(\tan\alpha)^2\,\chi^2}{2}}] \\ \le\ & \sqrt{\mathbb{E}[e^{\left(2t + \frac{4}{t}\right)(\tan\alpha)\,(\chi - \mathbb{E}[\chi])}]\,\mathbb{E}[e^{(\tan\alpha)^2\,\chi^2}]}. \end{aligned} \tag{11}$$

Next we estimate each of the two expectations on the right-hand side of the last line. In the first case we have the moment generating function of $\chi - \mathbb{E}[\chi]$, where χ is a 1-Lipschitz function of a standard Gaussian vector. A standard Gaussian concentration argument and our definition of α give

$$\mathbb{E}[e^{\left(2t+\frac{4}{t}\right)(\tan\alpha)(\chi-\mathbb{E}[\chi])}] \le e^{\frac{\left(2t+\frac{4}{t}\right)^2(\tan\alpha)^2}{2}} \le 1 + c_0\varepsilon^2$$

for some universal constant $c_0 > 0$. The second term in (11) is the moment generating function of χ^2, a chi-squared random variable with $d-1$ degrees of freedom. Since $(\tan\alpha)^2 \le \varepsilon^2/(d-1) \le 1/2$ under our assumptions, one can compute explicitly

$$\mathbb{E}[e^{(\tan\alpha)^2\chi^2}] = \left(\frac{1}{1-2(\tan\alpha)^2}\right)^{d/2} \le 1 + c_0\,\varepsilon^2$$

for a (potentially larger, but still universal $c_0 > 0$). Plugging the two estimates back into (11), we obtain

$$\mathbb{E}[e^{\left(t+\frac{2}{t}\right)(\tan\alpha)\chi+\frac{(\tan\alpha)^2\chi^2}{2}}] \le e^{\left(t+\frac{2}{t}\right)(\tan\alpha)\sqrt{d-1}}(1 + c_0\,\varepsilon^2),$$

and the fact that $t\,(\tan\alpha)\sqrt{d-1} = \varepsilon$ implies that the right-hand side is $\le 1+\varepsilon+c\,(t^{-2}\varepsilon+\varepsilon^2)$ for some universal $c > 0$. Going back to (10) we see that this finishes our upper bound for $\mathbb{P}\{\{i,j\} \in \Gamma_+\}$. ■

A.3 Correlations Between Edges and Non-Edges

In this case we consider $s, s' \in S^{d-1}$ and look at correlations of "edge events."

Lemma 14 *For any $t \ge 1$, $0 < \theta < \pi$, define*

$$\xi := 1 - \cos\theta,\ \gamma := \frac{(1-\cos\theta)^2}{\sin\theta}.$$

Then there exists a universal constant $C > 0$ such that for $s, s' \in S^{d-1}$ such that $\langle s, s'\rangle \le \cos\theta$, we have

$$\mathbb{P}\{\langle X_{ij}, s\rangle \ge t,\ \langle X_{ij}, s'\rangle \ge t\} \le p\,[(C\,p\,t)^{2\xi+\xi^2} + e^{\gamma(1-\gamma)t^2+\frac{\gamma}{1-\gamma}+\frac{\gamma^2t^2}{2}}\,p]. \tag{12}$$

$$\mathbb{P}\{\langle X_{ij}, s\rangle < t,\ \langle X_{ij}, s'\rangle < t\} \le 1 - 2p + p\,[(C\,p\,t)^{2\,\xi+\xi^2} + e^{\gamma(1-\gamma)t^2+\frac{\gamma}{1-\gamma}+\frac{\gamma^2t^2}{2}}\,p]$$

Remark 15 (nearly equal vectors.) *Suppose $p = o(1)$ and $\theta = o(1)$. One may check that $\gamma = (1+o(1))\,\theta^3/4$ and $\xi = (1+o(1))\,\theta^2/2$. This means that if $\theta^3\,t^2 = o(\ln(1/p))$ and $\theta^2\,\ln(1/t\,p) = \omega(1)$, then*

$$\mathbb{P}\{\langle X_{ij}, s\rangle < t,\ \langle X_{ij}, s'\rangle < t\} \le 1 - 2p + o(p) = (1-p)^2\,(1+o(p)).$$

This is used in the proof of Theorem 6, part (iv) above.

Proof We focus on the inequalities in (12), from which the other inequalities follow. For convenience, we write $\eta := \cos\theta$ and note that

$$\eta = 1 - \xi, \text{ so } \gamma = 1 - \frac{1-(1+\xi)\eta}{\sqrt{1-\eta^2}}. \tag{13}$$

Moreover, $0 < \gamma < 1$: the first inequality is obvious, and the second follows from the fact that

$$0 < \theta < \frac{\pi}{2} \Rightarrow 0 < \gamma = \frac{(1-\cos\theta)^2}{\sin\theta} < \frac{(1-\cos\theta)\,(1+\cos\theta)}{\sin\theta} = \frac{1-\cos^2\theta}{\sin\theta} = \sin\theta < 1.$$

Let E denote the event in (12). The properties of standard Gaussian vectors imply

$$\mathbb{P}\{E\} = \mathbb{P}(\{N_1 \geq t\} \cap \{\eta\, N_1 + \sqrt{1-\eta^2}\, N_2 \geq t\})$$

where N_1, N_2 are independent standard normal random variables. In particular, we can upper bound

$$\mathbb{P}\{E\} \leq \mathbb{P}\{N_1 \geq (1+\xi)\,t\} + \mathbb{P}\{N_1 \geq t\}\,\mathbb{P}\left\{N_2 \geq \left(\frac{1-(1+\xi)\eta}{\sqrt{1-\eta^2}}\right)\,t\right\}, \tag{14}$$

The first term in the right-hand side is $1-\Phi(t+\xi t) \leq e^{-\frac{\xi^2 t}{2} - \xi t^2}\,(1-\Phi(t)) = e^{-\frac{2\xi+\xi^2}{2}\,t^2}\,(1-\Phi(t))$ by Lemma 16. The fact that

$$\lim_{t\to+\infty} \frac{(1-\Phi(t))}{e^{-t^2/2}/(t\sqrt{2\pi})} = 1,$$

implies that, for $t > 1$, the ratio $e^{-t^2/2}/p$ is bounded by a $C\,t$, $C > 0$ a constant. We conclude

$$\mathbb{P}\{N_1 \geq (1+\xi)\,t\} \leq p\,(e^{-t^2/2})^{2\xi+\xi^2} \leq p\,(C\,t\,p)^{2\xi+\xi^2}. \tag{15}$$

As for the second term in the right-hand side of (14), we apply Lemma 16 with

$$r := \frac{t\,(1-(1+\xi)\eta)}{\sqrt{1-\eta^2}} = (1-\gamma)\,t \text{ and } h := \gamma\,t.$$

We deduce:

$$\mathbb{P}\left\{N_2 \geq \left(\frac{1-(1+\xi)\eta}{\sqrt{1-\eta^2}}\right)\,t\right\} = 1-\Phi(r) \leq e^{\gamma\,(1-\gamma)\,t^2 + \frac{\gamma}{1-\gamma} + \frac{\gamma^2\,t^2}{2}}\,(1-\Phi(t)),$$

The proof finishes by combining the estimates for the right-hand side of (14). ■

Lemma 16 *If $\varepsilon \in (0, 1/2)$, $r > 0$ and $h \geq 0$,*

$$e^{-h\,r - \frac{h}{r} - \frac{h^2}{2}} \leq \frac{1-\Phi(r+h)}{1-\Phi(r)} \leq e^{-h\,r - \frac{h^2}{2}}.$$

Proof We first show the upper bound, namely:

$$\forall r, h > 0 \, : \, 1 - \Phi(r+h) \leq e^{-r\,h - \frac{h^2}{2}} \, (1 - \Phi(r)). \tag{16}$$

To see this, we note that:

$$\begin{aligned} 1 - \Phi(r+h) &= \int_0^{+\infty} \frac{e^{-\frac{(x+r+h)^2}{2}}}{\sqrt{2\pi}} \, dx \\ &= \int_0^{+\infty} \frac{e^{-\frac{(x+r)^2}{2}}}{\sqrt{2\pi}} \, e^{-\left(x+r+\frac{h}{2}\right) h} \, dx \\ &\leq \int_0^{+\infty} \frac{e^{-\frac{(x+r)^2}{2}}}{\sqrt{2\pi}} \, e^{-r\,h - \frac{h^2}{2}} \, dx \\ &= [1 - \Phi(r)] \, e^{-r\,h - \frac{h^2}{2}}. \end{aligned}$$

To continue, we go back to the formula

$$1 - \Phi(r+h) = \left(\int_0^{+\infty} \frac{e^{-\frac{(x+r)^2}{2}} \, e^{-(x+r)\,h}}{\sqrt{2\pi}} \, dx \right) \, e^{-\frac{h^2}{2}},$$

which is clearly related to

$$1 - \Phi(r) = \int_0^{+\infty} \frac{e^{-\frac{(x+r)^2}{2}}}{\sqrt{2\pi}} \, dx.$$

In fact, inspection reveals that

$$\frac{1 - \Phi(r+h)}{1 - \Phi(r)} = e^{-\frac{h^2}{2}} \, \mathbb{E}[e^{-h\,N} \mid N \geq r].$$

Using Jensen's inequality, we have

$$\frac{1 - \Phi(r+h)}{1 - \Phi(r)} \geq e^{-\frac{h^2}{2}} \, e^{-h\,\mathbb{E}[N | N \geq r]},$$

and (16) means that $\mathbb{P}\{N - r \geq t \mid N \geq r\} \leq e^{-t\,r}$, so $\mathbb{E}[N \mid N \geq r] \leq r + \frac{1}{r}$. We deduce:

$$\frac{1 - \Phi(r+h)}{1 - \Phi(r)} \geq e^{-\frac{h^2}{2}} \, e^{-h\,r - \frac{h}{r}},$$

as desired. ■

A.4 Random Graph Lemmas

Here we collect some results on random graphs that we need in the arguments. In the proof of Theorem 2 we use the following lower tail estimate of the clique number of an Erdős-Rényi random graph that follows from a standard use of Janson's inequality.

Lemma 17 *Let N_k denote the number of cliques of size k of a $G(n, 1/2-\alpha_n)$ Erdős-Rényi random graph where $0 \le \alpha_n \le 1/n$ and let $\delta > 2$. Denote $\omega = 2\log_2 n - 2\log_2\log_2 n + 2\log_2 e - 1$. If $k = \lfloor \omega - \delta \rfloor$, then there exists a constant C' such that for all n,*

$$\mathbb{P}\{N_k = 0\} \le \exp\left(\frac{-C'n^2}{(\log_2 n)^8}\right) .$$

Proof Write $p = 1/2 - \alpha_n$ and define $\omega_p = 2\log_{1/p} n - 2\log_{1/p}\log_{1/p} n + 2\log_{1/p}(e/2) + 1$. We use Janson's inequality ((Janson et al., 2000, Theorem 2.18)) which implies that

$$\mathbb{P}\{N_k = 0\} \le \exp\left(\frac{-(\mathbb{E}N_k)^2}{\Delta}\right) ,$$

where $\mathbb{E}N_k = \binom{n}{k} p^{\binom{k}{2}}$ and

$$\Delta = \sum_{j=2}^{k} \binom{n}{k}\binom{k}{j}\binom{n-k}{k-j} p^{2\binom{k-j}{2} - \binom{j}{2} - 2j(k-j)} .$$

To bound the ratio $\Delta/(\mathbb{E}N_k)^2$, we may repeat the calculations of Matula's theorem on the 2-point concentration of the clique number (Matula (1972)), as in (Palmer, 1985, Section 5.3).

Let $\beta = \log_{1/p}(3\log_{1/p} n)/\log_{1/p} n$ and define $m = \lfloor \beta k \rfloor$ Then we split the sum

$$\frac{\Delta}{(\mathbb{E}N_k)^2} = \sum_{j=m}^{k} \frac{\binom{k}{j}\binom{n-k}{k-j}}{\binom{n}{k}} p^{-\binom{j}{2}} + \sum_{j=2}^{m-1} \frac{\binom{k}{j}\binom{n-k}{k-j}}{\binom{n}{k}} p^{-\binom{j}{2}} .$$

To bound the first term, we write

$$\sum_{j=m}^{k} \frac{\binom{k}{j}\binom{n-k}{k-j}}{\binom{n}{k}} p^{-\binom{j}{2}} = \frac{F(m)}{\mathbb{E}N_k} ,$$

where $F(m) = \sum_{j=m}^{k} \binom{k}{j}\binom{n-k}{k-j} p^{-\binom{j}{2}+\binom{k}{2}}$. Now if $k = \lfloor \omega_p - \delta \rfloor$ for some $\delta \in (0, \omega_p)$, then the computations in (Palmer, 1985, pp.77–78) show that

$$F(m) \le \sum_{j=0}^{\infty} \left(\frac{kn\sqrt{1/p}}{p^{-k(1+\beta)/2}}\right)^j ,$$

which is bounded whenever

$$\frac{kn\sqrt{(1/p)}}{p^{-k(1+\beta)/2}} = o(1) .$$

This is guaranteed by our choice of $\beta = \log_{1/p}(3\log_{1/p} n)/\log_{1/p} n$. Hence, the first term is bounded by

$$\frac{F(m)}{\mathbb{E}N_k} = O(1)\sqrt{k}p^{k\delta/2} .$$

For the second term, once again just like in Palmer (1985), note that

$$\begin{aligned}
\sum_{j=2}^{m-1} \frac{\binom{k}{j}\binom{n-k}{k-j}}{\binom{n}{k}} p^{-\binom{j}{2}} &\leq O(1) \sum_{j=2}^{m-1} \frac{k^{2j}}{n^j} p^{-\binom{j}{2}} \\
&\leq O(1) \sum_{j=2}^{m-1} \left(\frac{k p^{-m/2}}{n} \right)^j \\
&\leq O(1) \sum_{j=2}^{m-1} \left(\frac{2(\log_{1/p} n)^4}{n} \right)^j \\
&\leq O\left(\frac{(\log_{1/p} n)^8}{n^2} \right) .
\end{aligned}$$

Putting everything together, we have that there exist constants C, C' such that for $k = \lfloor \omega_p - \delta \rfloor$,

$$\mathbb{P}\{N_k = 0\} \leq \exp\left(-C \left(\frac{(\log_{1/p} n)^8}{n^2} + p^{k\delta/2}\sqrt{k} \right)^{-1} \right) \leq \exp\left(\frac{-C' n^2}{(\log_2 n)^8} \right) ,$$

whenever $\delta > 2$. Noting that $\omega_p = \omega + O(\alpha_n \log n)$ completes the proof. ■

Part (iii) of Theorem 3 crucially hinges on the following interesting result of Alon and Sudakov (2010) on the "resilience" of the chromatic number of a $G(n, 1/2)$ random graph. The form of the theorem cited here does not explicitly appear in Alon and Sudakov (2010) but the estimates for the probability of failure follow by a simple inspection of the proof of their Theorem 1.2.

Proposition 18 (Alon and Sudakov, 2010, Theorem 1.2). *There exist positive constants c_1, c_2 such that the following holds. Let $\epsilon > 0$ and let G be a $G(n, 1/2)$ random graph. With probability at least $1 - \exp(c_1 n^2/(\log n)^4)$, for every collection E of at most $c_2 \epsilon^2 n^2/(\log_2 n)^2$ edges, the chromatic number of $G \cup E$ is at most $(1+\epsilon)n/(2\log_2 n)$.*

The final lemma is used in proving part (i) of Theorem 6.

Lemma 19 *Fix $c \in (0,1)$. With $p = c \log n/n$, let N be the number of isolated vertices in $G(n,p)$. Then for n large, $\mathbb{P}(N = 0) \leq \exp(-n^{1-c}/3)$.*

Proof The following approach is borrowed from O'Connell (1998). Fix $q = 1 - \sqrt{1-p}$ and let $D(n,q)$ be the random directed graph with vertices $[n]$ in which each oriented edge ij appears independently with probability q. Write I for the number of vertices of $D(n,q)$ with no incoming edges, and M for the number of isolated vertices in $D(n,q)$, with no incoming or outgoing edges. Then M and N have the same distribution. Next, observe that I has law $\mathrm{Bin}\left(n, (1-q)^{n-1}\right) = \mathrm{Bin}\left(n, (1-p)^{(n-1)/2}\right)$. Furthermore, conditional on I,

$$M \stackrel{\mathrm{d}}{=} \mathrm{Bin}\left(I, (1-p)^{(n-I)/2}\right).$$

It follows that

$$\begin{aligned}\mathbb{P}(N=0) &= \mathbb{P}(M=0)\\ &\leq \mathbb{P}(|I-\mathbb{E}I| > \mathbb{E}I/2) + \sup_{k\in(1/2)\mathbb{E}I,(3/2)\mathbb{E}I} \mathbb{P}(\text{Bin}\left(k,(1-p)^{(n-k)/2}\right)=0). \qquad (17)\end{aligned}$$

For the first term, a Chernoff bound gives

$$\mathbb{P}(|I-\mathbb{E}I| > \mathbb{E}I/2) \leq 2e^{-\mathbb{E}I/10} = 2e^{-n(1-p)^{(n-1)/2}/10} = e^{-(1+o(1))n^{1-c/2}/10}, \qquad (18)$$

where the last inequality holds since $(1-p)^{(n-1)/2} = (1+o(1)n^{-c/2}$. Next, fix k as in the above supremum. For such k we have $p(n-k) = c\log n + O(\log n/n^{c/2})$. Using this fact and that $1-p \geq e^{-p-p^2}$ for p small yields

$$\begin{aligned}\mathbb{P}(\text{Bin}\left(k,(1-p)^{(n-k)/2}\right)=0) &= (1-(1-p)^{(n-k)/2})^k\\ &\leq \exp\left(-k(1-p)^{(n-k)/2}\right)\\ &= \exp\left(-ke^{-(p+p^2)(n-k)/2}\right)\\ &= \exp\left(-(1+o(1))kn^{-c/2}\right).\end{aligned}$$

Using that $1-p \geq e^{-p-p^2}$ a second time gives

$$k \geq \mathbb{E}I/2 = n(1-p)^{(n-1)/2}/2 \geq (1+o(1))ne^{-np/2}/2 = (1+o(1))n^{1-c/2}/2.$$

The two preceding inequalities together imply that

$$\mathbb{P}(\text{Bin}\left(k,(1-p)^{(n-k)/2}\right)=0) \leq \exp\left(-(1/2+o(1))\cdot n^{1-c}\right).$$

Using this bound and (18) in the inequality (17), the result follows easily. ■

References

N. Alon and J.H. Spencer. *The Probabilistic Method*. Wiley, New York, 1992.

N. Alon and B. Sudakov. Increasing the chromatic number of a random graph. *Journal of Combinatorics*, pages 345–356, 2010.

B. Bollobás. The chromatic number of random graphs. *Combinatorica*, 8:49–55, 1988.

B. Bollobás. *Random Graphs*. Cambridge University Press, Cambridge, UK, 2001.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

T.M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, 14: 326–334, 1965.

P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6: 290–297, 1959.

P. Erdős and A. Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.

S. Janson, T. Łuczak, and A. Ruciński. *Random Graphs*. John Wiley, New York, 2000.

S. Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian J. Math. Stat.*, 4(1):66–70, 2011. ISSN 1994-5418. doi: 10.3923/ajms.2011.66.70. URL `http://dx.doi.org/10.3923/ajms.2011.66.70`.

J. Matoušek. *Lectures on Discrete Geometry*. Springer, 2002.

D.W. Matula. Employee party problem. In *Notices of the American Mathematical Society*, volume 19, pages A382–A382, 1972.

N. O'Connell. Some large deviation results for sparse random graphs. *Probab. Theory Related Fields*, 110(3):277–285, 1998. ISSN 0178-8051. doi: 10.1007/s004400050149. URL `http://dx.doi.org/10.1007/s004400050149`.

E.M. Palmer. *Graphical Evolution*. John Wiley & Sons, New York, 1985.

M. Rudelson and R. Vershynin. Smallest singular value of a random rectangular matrix. *Communications in Pure and Applied Mathematics*, 62(12):1707–1739, 2009. URL `http://arxiv.org/abs/0802.3956`.

L. Schläffli. *Gesammelte Mathematische Abhandlungen*. Birkhäuser-Verlag, Basel, 1950.

E. Shamir and J. Spencer. Sharp concentration of the chromatic number on random graphs $G_{n,p}$. *Combinatorica*, 7:374–384, 1987.

J.E. Steif. A survey of dynamical percolation. In *Fractal Geometry and Stochastics IV*, pages 145–174. Springer, 2009.

V.N. Vapnik and A.Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

V.N. Vapnik and A.Ya. Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow, 1974. (in Russian); German translation: *Theorie der Zeichenerkennung*, Akademie Verlag, Berlin, 1979.

V.N. Vapnik and A.Ya. Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26:821–832, 1981.

Journal of Machine Learning Research 16 (2015) 1923-1948 Submitted 6/15; Published 9/15

Semi-Supervised Interpolation in an Anticausal Learning Scenario

Dominik Janzing DOMINIK.JANZING@TUEBINGEN.MPG.DE
Bernhard Schölkopf BS@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems
Spemannstr. 38
72076 Tübingen, Germany

Editor: Alex Gammerman and Vladimir Vovk

Abstract

According to a recently stated 'independence postulate', the distribution P_{cause} contains no information about the conditional $P_{\text{effect|cause}}$ while P_{effect} may contain information about $P_{\text{cause|effect}}$. Since semi-supervised learning (SSL) attempts to exploit information from P_X to assist in predicting Y from X, it should only work in anticausal direction, i.e., when Y is the cause and X is the effect. In causal direction, when X is the cause and Y the effect, unlabelled x-values should be useless. To shed light on this asymmetry, we study a deterministic causal relation $Y = f(X)$ as recently assayed in Information-Geometric Causal Inference (IGCI). Within this model, we discuss two options to formalize the independence of P_X and f as an orthogonality of vectors in appropriate inner product spaces. We prove that unlabelled data help for the problem of interpolating a monotonically increasing function if and only if the orthogonality conditions are violated – which we only expect for the anticausal direction. Here, performance of SSL and its supervised baseline analogue is measured in terms of two different loss functions: first, the mean squared error and second the surprise in a Bayesian prediction scenario.

Keywords: semi-supervised learning, anticausal learning, independence of cause and mechanism, information geometry, causality

1. Introduction

Semi-supervised learning (SSL) has received increasing attention during the past decade (Darnstädt et al., 2013; Ben-David et al., 2008; Yuanyuan et al., 2010; Chapelle et al., 2006). In contrast to supervised learning, where the prediction of a variable Y from another variable X is based on pairs $(x_1, y_1), \ldots, (x_n, y_n)$, semi-supervised learning uses additional x-values $x_{n+1}, \ldots, x_{n+m}$ to improve the prediction. Motivated by the fact that the y-values are often discrete variables, that is, 'labels', one often talks about the pairs as *labelled* instances and the unpaired x-values as *unlabelled* ones.

One can easily imagine scenarios where labelled instances are rare and unlabelled ones are easily available: consider, for example, the task of text classification, where labelling has to be done by humans while unlabelled instances can be retrieved from the internet automatically. Hence, SSL is useful provided that the unlabelled x-values indeed contain information about the relation between X and Y. Given the standard scenario where the pairs are i.i.d. drawn from P_{XY} and the unlabelled x-values from the corresponding marginal

distribution P_X, the essential question is the following. Predicting Y from X amounts to knowing properties of $P_{Y|X}$, while the unlabelled x-values only tell us something about P_X. Why should P_X contain information about $P_{Y|X}$?

Some recent approaches to distinguish cause and effect in causal structure learning (Janzing and Schölkopf, 2010; Daniusis et al., 2010; Janzing et al., 2012; Sgouritsa et al., 2015) were motivated by an informal 'independence' postulate stating that P_{cause} and $P_{\text{effect|cause}}$ contain no information about each other. On the other hand, P_{effect} and $P_{\text{cause|effect}}$ may contain information about each other. This has been shown by means of several toy examples (Janzing and Schölkopf, 2010; Daniusis et al., 2010; Janzing et al., 2012) using appropriate formalizations of the independence postulate. In the same spirit, Schölkopf et al. (2012, 2013) argue that under the independence postulate, SSL cannot work in the causal setting, that is, if X is the cause and Y the effect (provided that there is no common cause of both), while it may work in anticausal setting, i.e., when the cause is predicted from the effect. In a typical scenario of SSL that often appears in the literature (Chapelle et al., 2006), Y attains few values $\{1, \ldots, k\}$ only (Zhang and Oles, 2000) and $X \in \mathbb{R}^d$ is a high-dimensional vector. Then different labels j may correspond to different clusters in $\mathbb{R}^d$. If they are sufficiently apart, the modes of P_X tell us the centers of the clusters, which helps in learning $P_{Y|X}$ from fewer data. Distributions that satisfy this (loose) condition are said to follow the cluster assumption, a case for which SSL can plausibly be justified (Chapelle et al., 2006): as long as each cluster contains some labelled data points, we can propagate the labels to the other points in the same cluster, and thus convert the semi-supervised learning problem to a supervised one. In our terminology, this assumption implies that points in the same cluster have the same label, i.e., certain properties of P_X imply properties of $P_{Y|X}$. A related assumption states that the separating boundary should lie in a region of low density of P_X (Chapelle et al., 2006) – again, an assumption relating P_X and $P_{Y|X}$.

The goal of this paper is to provide a mathematical understanding of why the performance of SSL is related to the causal direction. Previous work remains vague regarding the question in what sense P_{effect} may contain information about $P_{\text{cause|effect}}$ and which mathematical postulates about asymmetries between cause and effect are needed for this claim. Here we present a model in which a well-defined independence assumption between P_{cause} and $P_{\text{effect|cause}}$ ensures that unlabelled data from the effect help in the sense of quantitatively improving the prediction of the cause from the effect with respect to a natural loss function, while it does not help in causal direction. To this end, we have chosen a model where X and Y have the same range. The more popular case where X is high-dimensional and Y of lower dimension or even a discrete label could be misleading for our purposes: different ranges define an asymmetry between X and Y that could erroneously be attributed to the fact that one is the cause and the other the effect.

We study the following simple **interpolation problem**: *Let X and Y be random variables attaining values in $[0, 1]$, deterministically related by $Y = f(X)$, where f is an unknown bijective strictly monotonically increasing map. We are given $n - 1$ points $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$. For some additional x-value x_n, we seek to infer the corresponding y-value $y_n = f(x_n)$.*

We will analyze why knowing P_X enables a better estimation (which implies that P_X and $P_{Y|X}$ are somehow dependent), given that a certain independence between P_Y and $P_{X|Y}$ holds.

The paper is structured as follows. Section 2 introduces a toy model of a bijective deterministic relation between X and Y and formalizes independence between P_{cause} and $P_{\text{effect|cause}}$ in two different ways. We explain why this independence implies dependence between P_{effect} and $P_{\text{cause|effect}}$ with respect to both formalizations. Section 3 describes the interpolation problem in the supervised scenario (i.e., with no unlabelled points) and presents a straightforward solution via linear interpolation, which will be the baseline our SSL method is later compared to.

Section 4 describes a semi-supervised modification and shows that the advantage can be quantified in terms of the dependence measures introduced in Section 2. The main contribution of this paper is to describe the relation between the performance of SSL to a mathematically well-defined notion of dependence between P_X and $P_{Y|X}$. Although our toy scenario is certainly an oversimplification compared to real SSL scenarios, the value of this work lies in providing the first link between causal direction and applicability of SSL that can be proven, subject to an assumption that links causality to statistics.

2. Asymmetries Between Cause and Effect for Deterministic Relations

Our restriction to monotonically increasing bijections of $[0,1]$ coincides with the typical toy scenario used by Daniusis et al. (2010); Janzing et al. (2012) to explain Information-Geometric Causal Inference (IGCI) although the formalism of IGCI introduced therein is actually more general.

We are given two random variables C, E ('cause' and 'effect') attaining values in $[0,1]$. We assume that their distributions P_C and P_E have strictly positive densities p_C and p_E with respect to Lebesgue measure. We will often use $p(c)$ as short hand for $p_C(c)$, for instance. Assume we observe that C and E are deterministically related by

$$E = g(C) \quad \text{and} \quad C = g^{-1}(E)\,,$$

for some strictly monotonically increasing diffeomorphism[1] g of $[0,1]$.

So far, the assumptions are symmetric with respect to C and E and there is no reason why observing the joint distribution of E and C should enable one to infer which variable is the cause and which the effect, assuming that exactly one of the alternatives is true. The problem of distinguishing cause and effect gets solvable only after introducing an assumption that links the causal direction to an observable implication. The essential idea is that g (which uniquely determines $P_{E|C}$) and p_C do not contain information about each other. Subsections 2.1 and 2.2 will describe two different formalizations of this idea which are the basis for two different SSL methods presented in Subsections 4.1 and 4.2, respectively.

2.1 Uncorrelatedness Between p_C and Slope

To formalize the idea of independence between g and p_C, Daniusis et al. (2010); Janzing et al. (2012, 2015) postulate uncorrelatedness between p_C and the logarithm of the derivative of g, which will be explained in Subsection 2.2. Here we state an assumption that simplifies the former by dropping the logarithm:

1. The 'diffeomorphism' assumption is convenient for the theory although it can be significantly weakened. The example in Figure 1(a) uses functions g and g^{-1} that are almost everywhere differentiable, which is also sufficient.

Independence Assumption 1 (with slope) *If C causes E with $E = g(C)$ then*

$$\mathrm{Cov}[g', p_C] = 0\,. \tag{1}$$

Here, both functions g' and p_C are considered as random variables on the probability space $[0,1]$ with Lebesgue measure. Their covariance, i.e., the left hand side of (1), equals

$$\int_0^1 g'(c)p(c)dc - \int_0^1 g'(c)dc \int_0^1 p(c)dc = \int_0^1 g'(c)p(c)dc - 1\,. \tag{2}$$

It turns out that Independence Assumption 1 implies that P_E contains information about g^{-1} (and thus about $P_{C|E}$):

Lemma 1 (p_E correlates with slope) *Let $g \neq id$ and (1) hold. Then the derivative of g^{-1}, denoted by $g^{-1'}$, is positively correlated with p_E:*

$$\mathrm{Cov}[g^{-1'}, p_E] > 0\,. \tag{3}$$

Proof By substitution of variables, (2) implies

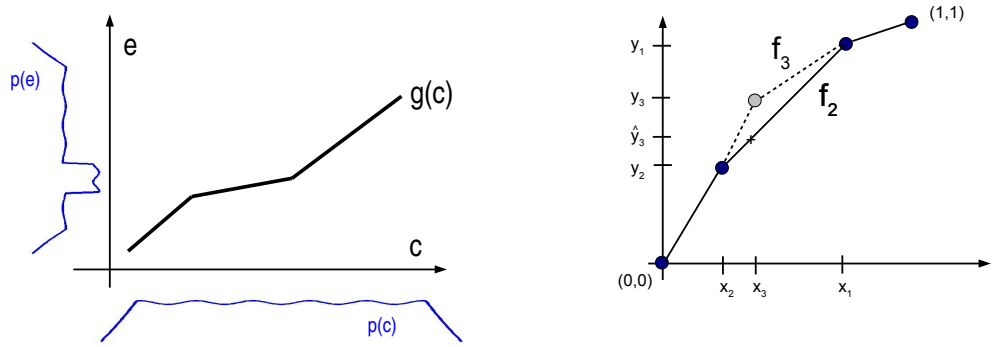
$$\int_0^1 p(e)\frac{1}{g^{-1'}(e)}de = 1\,. \tag{4}$$

We then conclude

$$\begin{aligned}
\int_0^1 p(e)g^{-1'}(e)de &= \int_0^1 p(e)g^{-1'}(e)de \cdot \int_0^1 p(e)\frac{1}{g^{-1'}(e)}de \\
&= \int_0^1 p(e)\left(\sqrt{g^{-1'}(e)}\right)^2 de \cdot \int_0^1 p(e)\left(\frac{1}{\sqrt{g^{-1'}(e)}}\right)^2 de \\
&\geq \left(\int_0^1 p(e)\sqrt{g^{-1'}(e)}\frac{1}{\sqrt{g^{-1'}(e)}}de\right)^2 = 1\,,
\end{aligned}$$

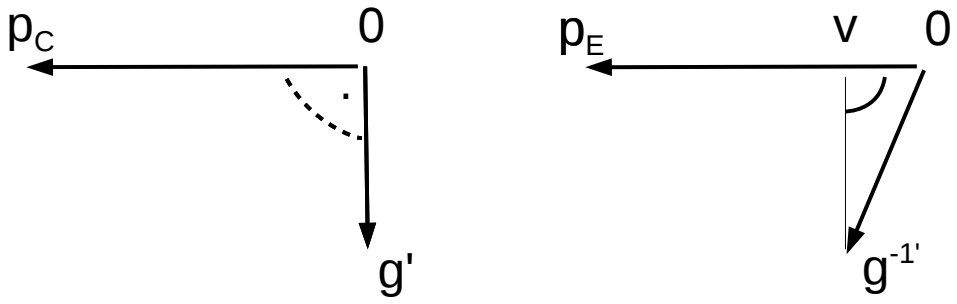
where we have applied the Cauchy-Schwarz inequality to the inner product $\langle \cdot, \cdot \rangle = \int p(e) \cdot \cdot de$ (note that it is strictly positive because p_E is strictly positive). Therefore we only have equality if $\sqrt{g^{-1'}}$ and $1/\sqrt{g^{-1'}}$ are linearly dependent, i.e., g' is constant and thus g is the identity due to $g(0) = 0$ and $g(1) = 1$. ■

Figure 1(a) provides a first intuition about Lemma 1: whenever the slope of g has been chosen independently of p_E, the density p_C tends to be high in regions where g is flat and g^{-1} is steep. Figures. 2(a) and 2(b) visualize the geometric content of Lemma 1 in the following sense. The covariance defines an inner product in the space of square integrable random variables if variables are identified up to constants. Then we have postulated orthogonality of g' and p_C and concluded non-orthogonality of g^{-1} and p_E. Therefore, the projection v of g^{-1} onto the line $(0, p_E)$ is closer to g^{-1} than 0. Within our setting, this point v will later play a crucial role for constructing the optimal prediction of g^{-1} that can be obtained from p_E.



(a) (b)

Figure 1: (a) If g has been designed independently of p_C, then the density p_E tends to be high in regions where g is flat. Source: Janzing et al. (2012). (b) The piecewise linear function f_2 interpolating the observations $(x_1, y_1), (x_2, y_2)$ is used for predicting y_3. f_3 accounts also for the point (x_3, y_3) and is later used to predict y_4 once x_4 is provided.



(a) (b)

Figure 2: Orthogonality of the random variables p_C and g' (in the sense of vanishing covariance) in Figure (a) implies non-orthogonality of p_E and $g^{-1'}$ in Figure (b). In Subsection 4.1, the squared distance of v and 0 will be the amount by which SSL can improve the performance of the interpolation.

2.2 Uncorrelatedness Between p_C and Logarithmic Slope

To phrase independence of g and p_C as uncorrelatedness of p_C and the derivative of g is certainly only one simple choice out of many options. Instead, Daniusis et al. (2010); Janzing et al. (2012) postulate uncorrelatedness between p_C and the *logarithm* of the derivative of g:

Independence Assumption 2 (with logarithmic slope) *If C causes E with $E = g(C)$ then*

$$\mathrm{Cov}[\log g', p_C] = 0\,. \tag{5}$$

Here, both functions $\log g'$ and p_C are considered as random variables on the probability space $[0,1]$. Again, their covariance is then computed with respect to the Lebesgue measure, i.e., the left hand side of (5) is short hand for

$$\int_0^1 \log g'(c)p(c)dc - \int_0^1 \log g'(c)dc \int_0^1 p(c)dc = \int_0^1 \log g'(c)p(c)dc - \int_0^1 \log g'(c)dc\,.$$

Assumption 2 admits several information theoretic interpretations (Daniusis et al., 2010; Janzing et al., 2012, 2015) of which we only explain the ones that are required for our analysis.

It turns out (Daniusis et al., 2010, Section 2) that Assumption 2 implies that P_E contains information about g^{-1} (and thus about $P_{C|E}$):

Lemma 2 (p_E correlates with logarithmic slope) *Let $g \neq id$ and (5) hold. Then the logarithm of the derivative of g^{-1}, denoted by $g^{-1'}$, is positively correlated with p_E:*

$$\mathrm{Cov}[\log g^{-1'}, p_E] > 0\,. \tag{6}$$

Our algorithm and the performance analysis will be based on the following information geometric rephrasing of the above.

Lemma 3 (covariance as difference of relative entropies) *Let*

$$D(q\|r) := \int_0^1 q(w) \log \frac{q(w)}{r(w)} dw$$

denote the relative entropy distance between the probability densities q and r. Then,

$$\mathrm{Cov}[\log g', p_C] = -D(p_C\|g') + D(p_C\|u) + D(u\|g')\,,$$

where u denotes the uniform density. Here we have interpreted g' as probability density which is possible due to $g' > 0$ and $\int g'(c)dc = 1$.

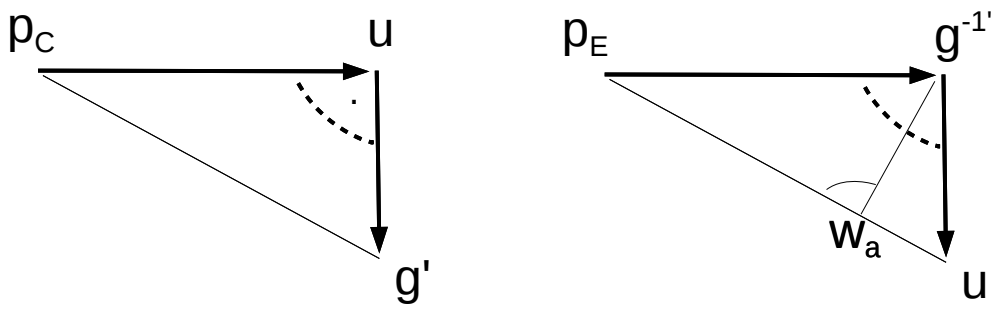
The following conclusion is immediate:

Corollary 1 (independence as orthogonality in information space) *(5) is equivalent to*

$$D(p_C\|g') = D(p_C\|u) + D(u\|g')\,. \tag{7}$$

Likewise, (6) is equivalent to

$$D(p_E\|g^{-1'}) < D(p_E\|u) + D(u\|g^{-1'})\,. \tag{8}$$



(a) (b)

Figure 3: (a) Independence Assumption 2 for P_C and $P_{E|C}$ implies that (p_C, u, g') is a Pythagorean triple, i.e., there is a rectangle at u. (b) Since bijections preserve relative entropy, the right angle for the backward direction occurs at $g^{-1'}$ instead of u, as would be required by the corresponding independence assumption for P_E and $P_{C|E}$. The point w_a obtained by projecting $g^{-1'}$ onto the line u, p_E will later play a crucial role for our SSL method and the distance $D(w_a\|u)$ will quantify the amount by which SSL improves the interpolation.

Without going to the details of information geometry Amari and Nagaoka (1993), we use some of its terminology and mention that due to (7), (p_C, u, g') is called a Pythagorean triple. This is visualized by drawing a right angle at u, see Figure 3(a). The idea is that square distance in Euclidean geometry is replaced with relative entropy in information geometry and therefore (7) replaces the usual Pythagorean theorem.[2] This way, Assumptions 1 and 2 both amount to orthogonality conditions in appropriate spaces.

Since relative entropy is preserved under bijections, we also have:

Lemma 4 (right angle at $g^{-1'}$) *Eq. (5) is equivalent to*

$$D(p_E\|u) = D(p_E\|g^{-1'}) + D(g^{-1'}\|u)\,. \tag{9}$$

Geometrically, this means that the right angle now occurs at $g^{-1'}$, as visualized by Figure 3(b), whereas independence between p_E and $g^{-1'}$ would require it to occur at u. In other words, by formalizing independence between input distribution and function as a certain orthogonality in information space, independence in causal direction implies dependence in anticausal direction. IGCI uses this asymmetry for inferring which of the two variables is the cause.

The goal of this paper is to answer the question why P_X is helpful for the interpolation problem stated in Section 1 when $X = E$ and $Y = C$, while it is useless when $X = C$

2. Then, the m-geodesic connecting p_C and u (given by the line $\lambda p_C + (1-\lambda)u$) is orthogonal to the e-geodesic connecting u and g' which is given by an affine combination on the logarithmic scale, that is, by $\lambda \log u + (1-\lambda)\log g'$.

and $Y = E$. Some thoughts on this can be found in Janzing et al. (2015, Section 4), but here we will describe a learning scenario where the information of P_X on f amounts to reducing the loss with respect to some natural loss function. To this end, we first describe a baseline method for the interpolation problem in Section 3 and analyze its performance with respect to two different loss functions. In Section 4 it will turn out that our two different formalizations of dependence vs. independence in Subsections 2.1 and 2.2 yield two different algorithms each of which improves the performance with respect to one of these loss functions.

3. Baseline Solutions of the Interpolation Problem

To analyze the performance of our interpolation methods (baseline and SSL) we consider a game consisting of infinitely many steps: In the nth step, we are given $(n-1)$ pairs

$$(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$$

obtained by i.i.d. sampling from P_{XY}. After observing the next x-value x_n, we are supposed to infer the corresponding value y_n. Having inferred it, we are told the true value y_n and the next x-value x_{n+1}. The reason why we define this game is that our theory will not provide a performance statement for any *specific* n. Instead, we will show that SSL outperforms the baseline method on average over all n until some $n_{\max}$ if $n_{\max}$ tends to infinity. Note, however, that the first step $n = 1$ would be usually called 'unsupervised learning', which we include as special case of SSL in our analysis.

Note, moreover, that 'inferring y_n' can mean two different things: either one infers one specific value $\hat{y}_n$. Then the performance is evaluated by some distance measure between the estimated value $\hat{y}_n$ and the true value y_n. The other sense of 'inferring' is to define some conditional probability density[3]

$$\mathrm{pr}(y_n | x_1, \ldots, x_n, y_1, \ldots, y_{n-1}) \tag{10}$$

expressing one's belief about y_n. Then it is natural to evaluate the performance of the prediction by the 'surprise' given by the negative logarithm of (10). Subsections 3.1 and 3.2 describe the supervised baseline scenarios for the two different settings.

3.1 Predicting One Specific Value by Linear Interpolation

As baseline method we consider interpolation by piecewise linear functions:

Definition 1 (linear interpolation) *For some $(n-1)$-tuple of points*

$$(x_1, y_1), \ldots, (x_{n-1}, y_{n-1}), \quad \text{with } n \geq 1,$$

let f_n denote the function that linearly interpolates between these points (see Figure 1(a), right). Explicitly, it is given by first ordering the x-values $x_1^\circ < \cdots < x_{n-1}^\circ$, which also

3. We use the notation pr to indicate that it is not connected to the probability densities p_X and p_Y. In a fully Bayesian scenario we would parameterize the set of distributions P_{cause} and the set of functions g and then define a prior on both parameter spaces. Here, pr expresses a belief on y_n that will later be based on some naive smoothness assumption formalized by the Dirichlet prior without accounting for any explicit generating model.

orders the y-values $y'_1 < \cdots < y'_{n-1}$. *Then* f_{n-1} *is the piecewise linear function that linearly connects* (x_j°, y_j°) *with* $(x_{j+1}^\circ, y_{j+1}^\circ)$ *for* $j = 0, \ldots, n-1$ *after we have set* $x_0^\circ = y_0^\circ = 0$ *and* $x_n^\circ = y_n^\circ = 1$. *Hence,* f_0 *is the identity.*

Although the interpolating function f_n depends on the whole $(n-1)$-tuple of points, it will be convenient to have only the index n since we refer to a fixed list of observations (obtained by i.i.d sampling from P_{XY}) of which we only know the first $n-1$. We set

$$\hat{y}_n := f_{n-1}(x_n)\,,$$

with f_{n-1} as in Definition 1, see also Figure 1(a), right. Here and throughout the paper i will denote the index for which x_n lies in the interval $(x_i^\circ, x_{i+1}^\circ)$ (see the notation of Definition 1). Then the estimated value is explicitly given by

$$\hat{y}_n = \frac{x_n - x_i^\circ}{x_{i+1}^\circ - x_i^\circ}(y_{i+1}^\circ - y_i^\circ) + y_i^\circ\,. \tag{11}$$

To analyze the performance of the SSL version versus standard liner interpolation, it would be natural to measure the deviation of $\hat{y}_n$ from y_n via the usual squared loss $(\hat{y}_n - y_n)^2$. Here we modify this term as follows:

Definition 2 (modified squared loss) *The deviation between the estimated value* $\hat{y}_n$ *and the true value* y_n *in step* n *is measured by the loss*

$$L_n(y_n, \hat{y}_n) := \left(\frac{1}{x_n - x_i^\circ} + \frac{1}{x_{i+1}^\circ - x_n}\right)(\hat{y}_n - y_n)^2\,, \tag{12}$$

where i *again denotes the index for which* $x_n \in (x_i^\circ, x_{i+1}^\circ)$.

The additional weighting factor amounts to stronger penalizing the deviation for those cases where x_n is close to the neighbors x_i° and x_{i+1}°. This can be justified by the idea that these errors should count stronger because one should actually be able to infer y_n more accurately when labelled points are close. The main reason, however, for the weighting factor is that it is necessary to link the performance of linear interpolation to Independence Assumption 1. The following reinterpretation will later be the reason why the loss (12) is convenient for our purposes:

Lemma 5 (squared loss as distance of derivatives) *Let* $\hat{f}_n$ *and* f_n *be the piecewise linear functions (linear on our* n *intervals) that interpolate the points* $(x_n, \hat{y}_n)$ *and* (x_n, y_n), *respectively, in addition to the points* (x_i, y_i) *for* $i = 1, \ldots, n-1$. *Then,*

$$L_n(y_n, \hat{y}_n) = \int_0^1 (f_n'(x) - \hat{f}_n'(x))^2 dx\,. \tag{13}$$

Proof:

$$\int_0^1 (f_n'(x) - \hat{f}_n'(x))^2 dx = \left(\frac{y_n - y_i^\circ}{x_n - x_i^\circ} - \frac{\hat{y}_n - y_i^\circ}{x_n - x_i^\circ}\right)^2 (x_n - x_i^\circ) +$$
$$\left(\frac{y_{i+1}^\circ - y_n}{x_{i+1}^\circ - x_n} - \frac{y_{i+1}^\circ - \hat{y}_n}{x_{i+1}^\circ - x_n}\right)^2 (x_{i+1}^\circ - x_n) = (y_n - \hat{y}_n)^2 \left(\frac{1}{x_n - x_i^\circ} + \frac{1}{x_{i+1}^\circ - x_n}\right) = L_n(y_n, \hat{y}_n)\,.$$

■

We now show that the loss until step $n_{\max}$ and the total loss over infinitely many steps can be given in a concise form. The proofs will be skipped because the corresponding results for the SSL scenario (Lemma 16 and Theorem 2) contain the statements below as special cases.

Lemma 6 (total loss until step $n_{\max}$) *The sum over all modified quadratic errors reads:*

$$\sum_{n=1}^{n_{\max}} L_n(y_n, \hat{y}_n) = \int_0^1 (f'_{n_{\max}}(x) - 1)^2 dx \,.$$

Therefore, the asymptotic loss reads:

Lemma 7 (total loss) *The sum over all modified quadratic errors reads:*

$$\sum_{n=1}^{\infty} L_n(y_n, \hat{y}_n) = \int_0^1 (f'(x) - 1)^2 dx = \mathrm{Var}(f') \,,$$

where we consider f' as random variable on the probability space $[0,1]$ with respect to the Lebesgue measure.

Recall that we have already considered derivatives of functions as random variables in Subsection 2.1. It is intuitively plausible that the complexity of the interpolation problem depends on the non-linearity of f, which can be quantified by the variance of f'. Note that this variance is also the squared length of the vectors g' and $g^{-1'}$ in Figure 2(a). Hence, we have linked the modified quadratic errors to Euclidean geometry in the space of random variables of Subsection 2.1. Accordingly, the non-orthogonality of p_E and $g^{-1'}$ in this space will be employed to construct an SSL algorithm that outperforms linear interpolation with respect to the modified quadratic errors.

3.2 Interpolation via a Dirichlet Process

To obtain a probability distribution that expresses our belief about y_n, given $x_1, \ldots, x_n$ and $y_1, \ldots, y_{n-1}$, we define a prior over the monotonically increasing functions. An arbitrary monotonic function f on $[0,1]$ with $f(0) = 0$ and $f(1) = 1$ can be interpreted as cumulative distribution function of a probability distribution on $[0,1]$. Since Dirichlet distributions can be used as priors for probability distributions, it is therefore also natural to use them as priors for increasing functions. We first introduce Dirichlet distributions of finite order (Balakrishnan and V., 2003):

Definition 3 (Dirichlet distribution) *The Dirichlet distribution* $\mathrm{Dir}(\alpha)$ *of order k and parameter vector $\alpha = (\alpha_1, \ldots, \alpha_k)$ with $\alpha_j > 0$ is defined as the density on the simplex*

$$\left\{ \theta \in \mathbb{R}^k \;\middle|\; \theta_j > 0, \sum_{j=1}^{k} \theta_j = 1 \right\} ,$$

given by

$$\mathrm{pr}(\theta) := \frac{1}{B(\alpha)} \prod_{j=1}^{k} \theta_j^{\alpha_j - 1}\,, \tag{14}$$

where $B(\alpha)$ is the normalization constant

$$B(\alpha) := \frac{\prod_{j=1}^{k} \Gamma(\alpha_j)}{\Gamma(\sum_{j=1}^{k} \alpha_j)}\,,$$

and Γ denotes the gamma function.

The following known result shows how the α_j control the expectations:

Lemma 8 (expectation of Dirichlet distribution) *The expectation of each θ_j is given by*

$$\mathbf{E}[\theta_j] = \frac{\alpha_j}{\sum_{j=1}^{k} \alpha_j}\,.$$

The sum over all α_j then controls to what extent the distribution is concentrated around its mean. The following well-known property will be crucial below:

Lemma 9 (aggregation property of Dirichlet) *If $(\theta_1, \ldots, \theta_k)$ is a random vector distributed according to $\mathrm{Dir}(\alpha_1, \ldots, \alpha_k)$ then $(\theta_1, \ldots, \theta_{k-2}, \theta_{k-1} + \theta_k)$ is distributed according to*
$\mathrm{Dir}(\alpha_1, \ldots, \alpha_{k-2}, \alpha_{k-1} + \alpha_k)$.

When a Dirichlet distribution is used to describe a distribution over distributions, then θ is the probability vector of k events. If we define Δ_j^Y with $j = 0, \ldots, n$ as the gaps obtained by ordering all values $y_1, \ldots, y_n$, then $\sum_{j=0}^{n} \Delta_j^Y = 1$ and thus the Dirichlet distribution of order $n+1$ defines a distribution over the set of possible difference vectors $\Delta^Y := (\Delta_0^Y, \ldots, \Delta_n^Y)$. However, we have to define distributions of order n for arbitrary n and need to ensure that the distributions defined for different n are consistent in the sense that marginalizing the distribution of $y_1, \ldots, y_n$ over y_n coincides with the distribution of $y_1, \ldots, y_{n-1}$ we define for $\tilde{n} = n - 1$. To this end, we use a Dirichlet process, which is the generalization of a Dirichlet distribution to infinite order:

Definition 4 (prediction via Dirichlet process) *Given the values $x_1, \ldots, x_n$, we define the probability density for the corresponding y-values by*

$$\mathrm{pr}(y_1, \ldots, y_n | x_1, \ldots, x_n) = \frac{1}{B(\alpha)} \prod_{j=0}^{n} (\Delta_j^Y)^{\alpha_j - 1}\,, \tag{15}$$

where the parameters are defined via the gaps of the corresponding x-values:

$$\alpha_j := \lambda \Delta_j^X \quad j = 0, \ldots, n\,, \tag{16}$$

where Δ_j^X are defined in analogy to Δ_j^Y. Here, $\lambda > 0$ is a parameter that controls to what extent we prefer linear function[4]

4. It should be noted that functions obtained by a Dirichlet process are almost surely discontinuous (Blackwell, 1973) although we have assumed the true function f to be differentiable. Yet, the process defines a reasonable prior for our 'naive' prediction scheme of finitely many y-values. Later, we will let λ go to infinity (which renders the discontinuities arbitrarily small) *before* we consider $n \to \infty$.

To understand this Definition, we need a few remarks. First note that actually Δ^Y is Dirichlet distributed, but the same probability density can be used for $\mathbf{y} := (y_1, \ldots, y_n)$ since the Jacobian of the transformation from Δ^Y to $\mathbf{y}$ is 1. This shows that the normalization of (14) still remains correct. To choose the parameters α_j proportional to the gaps in x-direction (16) amounts to taking the uniform distribution as 'base measure' according to standard terminology of Dirichlet processes. We will later see that changing the base measure provides a simple way to define an SSL version of the above prediction. Lemma 8 shows the implication of this choice: the expectation of each Δ_j^Y is given by the corresponding gap Δ_j^X. In this sense, the Dirichlet process a priori favors the linear function. For our further analysis it is also important to note that Lemma 9 implies

$$\mathrm{pr}(y_1, \ldots, y_{n-1}|x_1, \ldots, x_n) = \mathrm{pr}(y_1, \ldots, y_{n-1}|x_1, \ldots, x_{n-1})\,. \tag{17}$$

Hence, using (14) for n points and marginalizing over y_n is the same as applying it to $\tilde{n} := n - 1$ points only, which is the sense of consistency we have demanded above. In other words, the unlabelled value x_n is irrelevant for the prediction of the remaining $(n-1)$ y-values.

After having seen $(n-1)$ points, we interpolate via the prediction rule

$$\mathrm{pr}(y_n|x_1, \ldots, x_n, y_1, \ldots, y_{n-1}) = \frac{\mathrm{pr}(y_1, \ldots, y_{n-1}, y_n|x_1, \ldots, x_n)}{\mathrm{pr}(y_1, \ldots, y_{n-1}|x_1, \ldots, x_n)}\,. \tag{18}$$

Although our performance analysis does not require the explicit form of the left hand side of (18), the following result (which is shown in Appendix A) provides a better understanding about what it does:

Lemma 10 (interpolation by Dirichlet of order 2) *Eq. (15) yields*

$$\mathrm{pr}(y_n|x_1, \ldots, x_n, y_1, \ldots, y_{n-1}) = \frac{1}{(y_{i+1}^\circ - y_i^\circ)B(\alpha)} \prod_{l=1}^{2} (\theta_l)^{\alpha_l - 1}\,,$$

with $\theta_1 := (y_n - y_i^\circ)/(y_{i+1}^\circ - y_i^\circ)$ *and* $\theta_2 := 1 - \theta_1$. *The parameter vector reads*

$$\alpha := \lambda((x_n - x_i^\circ), (x_{i+1}^\circ - x_n))\,.$$

Note that we need the additional normalization factor $(y_{i+1}^\circ - y_i^\circ)$ compared to (14) because the Dirichlet distribution is actually a normalized probability density for $\theta_1 \in (0, 1)$ which we have transformed into a density for $y_n \in (y_{i+1}^\circ, y_i^\circ)$. Due to Lemma 8 the expectation of the ratio $\theta_1 = (y_n - y_i^\circ)/(y_{i+1}^\circ - y_i^\circ)$ is thus given by the corresponding ratio $(x_n - x_i^\circ)/(x_{i+1}^\circ - x_i^\circ)$. Hence, (18) favors piecewise linear interpolation as defined in Subsection 3.1. Note that the probability density of $\mathrm{Dir}(\alpha_1, \alpha_2)$ diverges at the boundaries $\theta_1 = 0, 1$ if $\alpha_j < 1$. To ensure that our interpolation uses a density that favours values y_n that are closer to the expectation instead of favouring those that are close to the bounds y_i° and y_{i+1}°, we choose $\lambda \gg n_{\max}$ because this yields $\lambda(x_j^\circ - x_{j+1}^\circ) > 1$ with high probability. Therefore, we will later consider the limit $\lambda \to \infty$.

We now define the loss in each step as the Bayesian surprise:

Definition 5 (Bayesian loss function) *The loss in step n is defined by*

$$L_n^\lambda(y_n) := -\log \mathrm{pr}(y_n|x_1,\dots,x_n,y_1,\dots,y_{n-1})\,,$$

where the superscript λ reminds us that pr *already depends on λ.*

Due to

$$\mathrm{pr}(y_1,\dots,y_n|x_1,\dots,x_n) = \prod_{j=1}^n \mathrm{pr}(y_j|x_1,\dots,x_j,y_1,\dots,y_{j-1})$$

(just apply (17) for each j) we obtain:

Lemma 11 (loss until step $n_{\max}$) *The total loss for steps $1,\dots,n_{\max}$ in the prediction game reads:*

$$\sum_{n=1}^{n_{\max}} L_n^\lambda(y_n) \quad = \quad -\log \mathrm{pr}(y_1,\dots,y_{n_{\max}}|x_1,\dots,x_{n_{\max}})$$

The asymptotic for large λ of the total loss can be nicely described in terms of relative entropies:

Theorem 1 (asymptotic total loss)

$$\lim_{\lambda\to\infty} \frac{1}{\lambda} \sum_{n=1}^{n_{\max}} L_n^\lambda(y_n) = D(u\|f'_{n_{\max}})\,. \tag{19}$$

Hence,

$$\lim_{n_{\max}\to\infty} \left[\lim_{\lambda\to\infty} \frac{1}{\lambda} \sum_{n=1}^{n_{\max}} L_n^\lambda(y_n) \right] = D(u\|f')\,. \tag{20}$$

Proof: To shorten notation, we write n and j for $n_{\max}$ and n, respectively. Taking the logarithm of (15) yields:

$$\log \mathrm{pr}(y_1,\dots,y_n|x_1,\dots,x_n) = \sum_{j=0}^n (\lambda\Delta_j^X - 1)\log\Delta_j^Y + \log\Gamma(\lambda) - \sum_{j=0}^n \log\Gamma(\lambda\Delta_j^X)\,. \tag{21}$$

We now use the Stirling approximation

$$\log\Gamma(z) = z\log z - z\log e + O(\log z)\,.$$

Thus,

$$-\sum_{j=0}^n \log\Gamma(\lambda\Delta_j^X) + \log\Gamma(\lambda) \quad = \quad -\lambda\sum_{j=0}^n \Delta_j^X \log\Delta_j^X - O(\log\lambda)\,.$$

Therefore,

$$\lim_{\lambda\to\infty} \frac{1}{\lambda} \log \mathrm{pr}(y_1,\dots,y_n|x_1,\dots,x_n) = \sum_{j=1}^n \Delta_j^X \log \Delta_j^Y/\Delta_j^X = -D(u\|f'_n)\,.$$

The second part of the statement holds because

$$\lim_{n\to\infty} \int_0^1 \log f_n'(x)dx = \int_0^1 \lim_{n\to\infty} \log f_n'(x)dx\,,$$

due to the bounded convergence theorem (the sequence $(\log f_n')_{n\in\mathbb{N}}$ is uniformly bounded because $\min_x\{f'(x)\} \leq f_n' \leq \max_x\{f'(x)\}$ by the mean value theorem). ■

We have seen that the complexity of the interpolation problem has turned out to depend on $D(u\|f')$ (for an appropriate limit, namely $\lambda \to \infty$). Since information geometry considers relative entropy as an analog of squared length in Euclidean geometry (Amari and Nagaoka, 1993), the total loss again depends on the squared length of the vector (u, g') or $(u, g^{-1'})$ in Figures 3(a) or 3(b), respectively, in analogy to Subsection 3.1 where it was given by $\mathrm{Var}(f')$ (i.e., the squared length of the vector g' or $g^{-1'}$ in Figures 2(a) or 2(b)).

4. Semi-Supervised Interpolation

In addition to the $n-1$ labelled points $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$ and the unlabelled value x_n, we are now given the density p_X. For the anticausal scenario, i.e., if $X = E$ and $Y = C$, Lemmas 1 and 2 state positive correlation between p_X and f' or $\log f'$, respectively. Hence, large density $p(x)$ tends to correspond to large slope. Qualitatively, this already provides a guideline on how to modify the linear interpolation: the value x_n defines a partition of $(x_i^\circ, x_{i+1}^\circ)$ into two intervals. We first compare the average probability density in the left interval with the one in the right one. Whenever it is larger in the left one than in the right one, we slightly increase $\hat{y}_n$ because we expect the slope of f to be larger on the left interval. This, however, is just a rough intuition. The precise method of employing our knowledge on p_X depends on whether we use the correlations between f' and p_X or between $\log f'$ and p_X. We start with the former because the performance analysis of the corresponding SSL method uses a loss function that is closer to standard loss functions in machine learning.

4.1 SSL Using Correlations Between Slope and Density

In our SSL version, the estimation reads:

Definition 6 (additive SSL interpolation) *Let F denote the cumulative distribution of X and $s > 0$ be a parameter that controls how strongly the interpolation accounts for the distribution p_X. Then additive SSL interpolation is given by*

$$\hat{y}_n^s := \hat{y}_n + s\frac{(x_{i+1}^\circ - x_n)(x_i^\circ - x_n)}{x_{i+1}^\circ - x_i^\circ}\left[\frac{F(x_n) - F(x_i^\circ)}{x_n - x_i^\circ} - \frac{F(x_{i+1}^\circ) - F(x_n)}{x_{i+1}^\circ - x_n}\right],$$

where $\hat{y}_n$ is defined as in (11). Note that s must be admissible in the sense that it is small enough to ensure that $\hat{y}_n^s$ remains inside the interval $(y_i^\circ, y_{i+1}^\circ)$.

To intuitively understand this interpolation, note that the term in the bracket is the difference between the average densities of the left and the right interval. Hence $\hat{y}_n^s$ is increased compared to the standard interpolation whenever the left interval contains higher density. Further understanding of why we define our SSL interpolation precisely in such a way will be provided below in the proof of Theorem 2. We first state our main result proved below:

Theorem 2 (total loss in terms of (co)-variances) *The total loss in the infinite interpolation game using $\hat{y}_n^s$ in Definition 6 reads:*

$$\sum_{n=1}^{\infty} L_n(y_n, \hat{y}_n^s) = \mathrm{Var}(f' - s\, p_X) = \mathrm{Var}(f') - 2s\,\mathrm{Cov}[f', p_X] + s^2\,\mathrm{Var}(p_X)\,.$$

In causal direction we have $\mathrm{Cov}[f', p_X] = 0$ and the additional term $s^2\,\mathrm{Var}(p_X)$ makes the performance worse than the baseline. In anticausal direction we have $\mathrm{Cov}[f', p_X] > 0$. Then standard linear regression tells us that the optimal improvement is reached for

$$s = \frac{\mathrm{Cov}[f', p_X]}{\mathrm{Var}(p_X)}\,,$$

if this value is admissible (otherwise one chooses a smaller one). Then the remaining loss reads:

$$\mathrm{Var}(f' - sp_X) \quad = \quad \mathrm{Var}(f') - \frac{(\mathrm{Cov}[f', p_X])^2}{\mathrm{Var}(p_X)}\,,$$

which is exactly the squared distance between v and g^{-1} in Figure 2(b). By Pythagoras, the squared length of $(v, 0)$ is the amount by which SSL improves the prediction for the optimal choice of s. We conclude:

Corollary 2 (Anticausal SSL works, causal SSL doesn't) *If $X = E$ and $Y = C$, SSL interpolation outperforms its supervised baseline version for sufficiently small s in the sense that*

$$\sum_{n=1}^{\infty} [L_n(y_n, \hat{y}_n^s) - L_n(y_n, \hat{y}_n)] < 0\,.$$

If $X = C$ and $Y = E$, SSL increases the total loss for all admissible s.

Finding the right value s needs to be a non-trivial problem for the following reason. p_E deviates from the uniform distribution for *two* reasons: first, because the function g is non-linear and second, because p_C is not uniform. In other words, we do not know which part of the structure of p_E is due to the structure of g and which part due to the structure of p_C. This is also shown by the two extreme cases (1) where g is the identity and p_C and p_E are identical densities and (2) p_C is uniform and $p_E = g^{-1'}$. The optimal way to use p_E for better predicting g^{-1} will typically be a compromise that neither assumes that p_C is uniform nor that g is linear. The two extreme cases nicely correspond to a degeneration of the triangles in Figures 3(a) and 2(a): For linear g, the derivative $g^{-1'}$ is constant and thus coincides with the trivial random variable 0 and the trivial density u. On the other hand, for uniform p_C, $g^{-1'}$ and p_E coincide. For the generic case, the projection of $g^{-1'}$ onto the line from p_E to u is an interior point. Finding the right balance between attributing the structure of p_E entirely to the structure of g or entirely to the structure of p_C amounts to finding the projection points v and w_a that correspond to an optimal performance of our SSL methods in Subsections 4.1 and in Subsection 4.2, respectively. Since we do not know g^{-1}, we do not know the projection points v and w_a beforehand. Therefore, we have to

work with the following heuristics: in step n, we choose the value s_{n-1} that minimizes the total loss until step $n-1$, which is easy to compute using Corollary 4 below.

The remainder of this subsection is devoted to the proof of Theorem 2 with some additional intuitive explanations at the end. To quantitatively analyze the loss, it is helpful to describe the estimation process as an estimation of the slope f'_n (which is equivalent) instead of an estimation of y_n. Let us define $\hat{f}_n$ as the function passing through $(x_n, \hat{y}_n)$ in addition to the points $(x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$. Standard linear interpolation obviously amounts to setting

$$\hat{f}'_n := f'_{n-1}\,.$$

Note that f'_{n-1} indicates the average slope for each open interval $(x_j^\circ, x_{j+1}^\circ)$ and is undefined for each x_j° with $j = 0, \ldots, n$. It is therefore convenient to consider f'_n as the following conditional expectation:

Lemma 12 (f'_n as conditional expectation of f') *Let $J_n : [0,1] \to \{0, \ldots, n\}$ be the random variable such that for each x the value $J_n(x)$ indicates the subinterval in which x lies (defined by the n observed x-values $x_1^\circ, x_i^\circ, x_n, x_{i+1}^\circ \ldots, x_{n-1}^\circ$). Then,*

$$f'_n = \mathbf{E}[f'|J_n]\,.$$

The proof is immediate via the mean value theorem. Similarly, we now introduce average densities:

Definition 7 (average density as conditional expectation) *Let J_n be defined as in Lemma 12. Then the average density (corresponding to the partition of $[0,1]$ defined by the first n x-values) is the function on $[0,1]$ given by*

$$p_n := \mathbf{E}[p_X|J_n]\,,$$

which is defined only in the interior of all $n+1$ intervals.

For $x \in (x_j^\circ, x_{j+1}^\circ)$ with $j \neq i$ we have, for instance:

$$p_n(x) = \frac{F(x_{j+1}^\circ) - F(x_j^\circ)}{x_{j+1}^\circ - x_j^\circ}\,. \tag{22}$$

Using these conditional expectations, our SSL interpolation can be written in a concise form:

Lemma 13 (additive SSL interpolation in terms of conditional expectations) *The interpolation in Definition 6 amounts to setting*

$$(\hat{f}^s)'_n = f'_{n-1} + s(p_n - p_{n-1})\,. \tag{23}$$

Proof: We only need to show that integrating (23) from x_i° to x_n yields the correct value for $\hat{y}_n^s$. On all intervals other than $(x_i^\circ, x_{i+1}^\circ)$ (23) is certainly true because $\hat{f}_n^s$ coincides with f_{n-1} and $p_n - p_{n-1}$ is zero. On the interval (x_i°, x_n) the average densities p_{n-1} and p_n are given by

$$p_{n-1} = \frac{F(x_{i+1}^\circ) - F(x_i^\circ)}{x_{i+1}^\circ - x_i^\circ} \quad \text{and} \quad p_n = \frac{F(x_n) - F(x_i^\circ)}{x_n - x_i^\circ}\,.$$

Inserting this into (23) and integrating it from x_i° to x_n yields:

$$\begin{aligned}
\hat{y}_n^s &= \hat{y}_n + s\left[F(x_n) - F(x_i^\circ) - \frac{F(x_{i+1}^\circ) - F(x_i^\circ)}{x_{i+1}^\circ - x_i^\circ}(x_n - x_i^\circ)\right] \\
&= \hat{y}_n + s\left[\frac{F(x_n) - F(x_i^\circ)}{x_{i+1}^\circ - x_i^\circ}(x_{i+1}^\circ - x_i^\circ) - \frac{F(x_{i+1}^\circ) - F(x_i^\circ)}{x_{i+1}^\circ - x_i^\circ}(x_n - x_i^\circ)\right] \\
&= \hat{y}_n + \frac{s}{x_{i+1}^\circ - x_i^\circ}\left[-(x_n - x_i^\circ)F(x_{i+1}^\circ) + (x_{i+1}^\circ - x_i^\circ)F(x_n) - (x_{i+1}^\circ - x_n)F(x_i^\circ)\right] \\
&= \hat{y}_n + s\frac{(x_{i+1}^\circ - x_n)(x_i^\circ - x_n)}{x_{i+1}^\circ - x_i^\circ}\left[\frac{F(x_n) - F(x_i^\circ)}{x_n - x_i^\circ} - \frac{F(x_{i+1}^\circ) - F(x_n)}{x_{i+1}^\circ - x_n}\right].
\end{aligned}$$

■

Using Lemma 5 we are now able to phrase the loss in step n using our conditional expectations:

Corollary 3 (difference between interpolating functions) *The loss of the SSL version in step n reads*

$$\begin{aligned}
L_n(y_n, \hat{y}_n^s) &= \int_0^1 \left[(f_n' - f_{n-1}') - s(p_n - p_{n-1})\right]^2 dx \qquad (24) \\
&= \int_0^1 [(f_n' - sp_n) - (f_{n-1}' - sp_{n-1})]^2 dx\,. \qquad (25)
\end{aligned}$$

To derive a closed form for the total loss until step n we observe that f_{n-1}' and p_{n-1} can also be seen as conditional expectations of f_n' and p_n, respectively:

Lemma 14 (concatenating conditional expectations)

$$\mathbf{E}[f'|J_{n-1}] = \mathbf{E}[f_n'|J_{n-1}] \quad \textit{and} \quad \mathbf{E}[p_X|J_{n-1}] = \mathbf{E}[p_n|J_{n-1}]\,.$$

Proof: Applying the law of total expectation $\mathbf{E}[\mathbf{E}[A|B]] = \mathbf{E}[A]$ to each value of J_{n-1} yields $\mathbf{E}[\mathbf{E}[f'|J_n]|J_{n-1} = j] = \mathbf{E}[f'|J_{n-1} = j]$. Hence, $\mathbf{E}[\mathbf{E}[f'|J_n]|J_{n-1}] = \mathbf{E}[f'|J_{n-1}]$. The proof for p_X is similar. ■

Since we want to show that the total loss until step n can be written as a variance, we first need to rewrite the loss in each step as variance:

Lemma 15 (loss as variance of conditional expectation)

$$L_n(y_n, \hat{y}_n^s) = \mathbf{E}[\mathrm{Var}(f_n' - sp_n|J_{n-1})]\,.$$

Proof: The right-hand side of (24) can be written as

$$\begin{aligned}
&\int_0^1 ((f_n' - sp_n) - (f_{n-1}' - sp_{n-1}))^2 dx = \int_0^1 (f_n' - sp_n - \mathbf{E}[f' - sp_X|J_{n-1}])^2 dx \\
= &\int_0^1 (f_n' - sp_n - \mathbf{E}[f_n' - sp_n|J_{n-1}])^2 dx = \mathbf{E}[\mathrm{Var}(f_n' - sp_n|J_{n-1})]\,.
\end{aligned}$$

■

We can now express the total loss after n steps as a variance:

Lemma 16 (total loss after n steps)

$$\sum_{j=1}^{n} L_j(y_j, \hat{y}_j^s) = \mathrm{Var}(f'_n - sp_n)\,, \tag{26}$$

where the variance is again meant with respect to the Lebesgue measure.

Proof: By induction over n. Let (26) hold for n. Using the law of total variance we have

$$\begin{aligned} \mathrm{Var}(f'_n - sp_n) &= \mathbf{E}[\mathrm{Var}(f'_n - sp_n|J_{n-1})] + \mathrm{Var}(\mathbf{E}[f'_n - sp_n|J_{n-1}]) \\ &= L_n(\hat{y}_n^s, y_n) + \mathrm{Var}(f'_{n-1} - sp_{n-1})\,, \end{aligned}$$

where we have used Lemma 15. ■

As a simple conclusion we find:

Corollary 4 (optimal value s_n) *The total loss $\sum_{j=1}^{n} L_j(y_j, \hat{y}_j^s)$ until step n is minimized for*

$$s_n := \frac{\mathrm{Cov}[f'_n, p_n]}{\mathrm{Var}(p_n)}\,.$$

Moreover, s_n converges to the value s optimizing the total loss for infinitely many steps.

The limit $n \to \infty$ now proves Theorem 2:

$$\begin{aligned} \lim_{n\to\infty} \mathrm{Var}(f'_n - sp_n) &= \lim_{n\to\infty} \int_0^1 (f'_n - sp_n) - (1-s))^2 dx \\ &= \int_0^1 ((f' - sp_X) - (1-s))^2 dx = \mathrm{Var}(f' - sp_X)\,. \end{aligned}$$

Theorem 2 only states an improvement of the total loss over the infinite number of steps without stating for which n we get an improvement. The following remarks provide an intuition about in which steps SSL is effective. The term $\mathrm{Cov}[f'_n, p_n]$ quantifies to what extent the covariance of f' and p_X is apparent on the level of coarse-graining defined by the observations available in step n. For $n \to \infty$, it converges to $\mathrm{Cov}[f', p_X]$, which is positive in the anticausal scenario. The difference

$$\mathrm{Cov}[f'_n, p_n] - \mathrm{Cov}[f'_{n-1}, p_{n-1}] \tag{27}$$

measures to what extent the correlations between f' and p_X get better visible when the coarse-graining is made finer by going from $n-1$ to n intervals. One can easily show that (27) can be rewritten as $\mathrm{Cov}[f'_n - f'_{n-1}, p_n - p_{n-1}]$, which is positive whenever either (1) y_n is greater than the value $\hat{y}_n$ obtained by linear interpolation $\hat{y}_n := f_{n-1}(x_n)$ and the average probability density is larger on the left interval (x_i°, x_n) than on the right interval (x_n, x_{i+1}°) or (2) y_n is smaller than $\hat{y}_n$ and the density is larger on the right interval. Hence, (27) is positive whenever our SSL method corrects $\hat{y}_n$ in the correct direction. In other words, SSL does the right thing in step n whenever n defines a level of coarse-graining for which the covariance of f' and p_X gets better visible than in the previous step.

4.2 SSL Using Correlations Between Log Slope and Density

As in Subsection 4.1 we modify the interpolation in a way that favors functions that have higher derivative in regions where p_X is large. To do so, we use the Dirichlet process in Definition 4 with respect to a coordinate system that makes p_X more uniform: if we reparameterize X such that the differences Δ_j^X get larger in regions with high density, the interpolation with respect to the new coordinates infers the corresponding Δ_j^Y to be larger. To analyze the total loss for such a 'deformed interpolation' does not require to redo the computations in Subsection 3.2. Instead, we observe that applying the transformation $x_j \mapsto \tilde{x}_j = b(x_j)$ with some diffeomorphism b and performing interpolation in the new coordinate system amounts to interpolating $\tilde{f} := f \circ b^{-1}$. We thus conclude that the term on the right hand side of (20) is replaced with $D(u\|(f \circ b^{-1})')$. As an aside, we should mention that interpolation in the new coordinates amounts to using a Dirichlet process with a different base measure, namely the density that is uniform in the *new* coordinates. We conclude:

Lemma 17 (loss of deformed interpolation) *The asymptotic of the loss with respect to the above 'b-deformed interpolation' (denoted by $\tilde{L}$) reads:*

$$\lim_{n_{\max}\to\infty} \left[\lim_{\lambda\to\infty} \sum_{n=1}^{n_{\max}} \tilde{L}_n^\lambda(y_n) \right] = D(\tilde{u}\|f'), \tag{28}$$

where $\tilde{u} := b'$ denotes the density that is the image of the uniform under b^{-1}.

Proof: Since the density f' is the image of the uniform distribution under f^{-1}, the density $(f\circ b^{-1})'$ is the image of the uniform distribution under $b\circ f^{-1}$. Relative entropy is preserved under bijections, we can thus apply b^{-1} to the left argument u of $D(.\|.)$ (which generates the density b') instead of applying b to the right one. ■

We can now easily compare the performance of interpolations with respect to different coordinate systems:

Lemma 18 (comparing Dirichlet interpolations)

$$\lim_{n_{\max}\to\infty} \left[\lim_{\lambda\to\infty} \sum_{n=1}^{n_{\max}} (L_n^\lambda(y_n) - \tilde{L}_n^\lambda(y_n)) \right] = D(u\|f') - D(\tilde{u}\|f')\,.$$

Given the relation between performance and the relative entropy stated by Lemma 18 we conclude:

Corollary 5 (benefit of changing the coordinate system) *The deformed interpolation with respect to a transformation that turns $\tilde{u}$ into the uniform distribution on $[0,1]$ asymptotically outperforms the standard interpolation for $n\to\infty$ if and only if*

$$D(\tilde{u}\|f') < D(u\|f')\,.$$

We now define the density that generates our SSL interpolation:

Definition 8 (SSL interpolation) *Let w_s be the mixture of p_X with the uniform distribution, i.e.,*

$$w_s = sp_X + (1-s)\,.$$

Apply the coordinate transformation that transforms w_s into the uniform distribution, i.e.,

$$W_s(x) := sF(x) + (1-s)x\,. \tag{29}$$

Then the deformed interpolation is our usual Dirichlet interpolation from Subsection 2.2 applied to the values $\tilde{x}_j := W_s(x_j)$.

We then state our main result regarding the performance of SSL by Dirichlet process in the modified coordinate system:

Theorem 3 (improvement of performace by SSL) *Predicting y_n via the Dirichlet process in the coordinate system W_s, as defined by (29), improves the performance by the amount $D(u\|w_s)$.*

To further understand our deformed interpolation one may wonder whether the expectation of y_n coincides with the value $\hat{y}_n^s$ in Subsection 4.1. Remarkably, this is not the case. Instead, it turns out that the SSL method in this subsection modifies the slope by a *multiplicative factor* that accounts for p_X while the SSL method in Subsection 4.1 corrects the slope by an *additive summand*. This nicely corresponds to the fact that Subsection 4.1 employs correlations between p_X and f' while this Subsection employs correlations between p_X and $\log f'$. This difference is made more explicit in Appendix D.

We now state our main result:

Theorem 4 (anticausal SSL works, causal SSL doesn't) *Let cause C and effect E satisfy Assumption 2. For $X := E$ and $Y := C$ and $f := g^{-1}$ there is an $s > 0$ for which the deformed interpolation outperforms standard linear interpolation. For $X := C$ and $Y := E$ and $f := g$, there is no such s.*

Proof In the terminology of information geometry (Amari and Nagaoka, 1993; Amari, 2001), $M := \{w_s\}_{s\in I}$ is an m-manifold. There is therefore a unique minimizer w_a of the distance $D(w_s\|f')$ (called the 'projection' of f' onto M) satisfying the orthogonality, see Eq. (60) in (Amari, 2001),

$$D(u\|f') = D(u\|w_a) + D(w_a\|f')\,. \tag{30}$$

For $X = C$ and $Y = E$, we have $w_a = u$. Therefore, M cannot contain any w_s for which $D(w_s\|f') < D(u\|f')$.

For the causal scenario $X = E$ and $Y = C$, we consider the function

$$h(s) := D(w_s\|f') = \int_0^1 (sp(x) - (1-s)) \log \frac{sp(x) - (1-s)}{f'(x)} dx\,. \tag{31}$$

Its derivative reads

$$h'(s) = \int_0^1 (p(x) - 1) \log \frac{w_s(x)}{f'(x)} dx\,. \tag{32}$$

We observe

$$h'(0) = \int_0^1 (p(x) - 1) \log \frac{1}{f'(x)} dx = -\mathrm{Cov}[p_X, \log f'] \,.$$

Using (6), we thus have $h'(0) < 0$. Therefore, the unique minimum a of h satisfies $a > 0$. ■

Remark 1 *We show in Appendix C that $a \leq 1$ holds in addition to $a > 0$ whenever one assumes the additional independence postulate*

$$\mathrm{Cov}[g', \log p_E] = 0 \,,$$

which has not been described in the literature yet.[5] *Then w_a is a* mixture *of u and p_X.*

In strong analogy to Subsection 4.1, the theory does not tell us how to find the optimal value $s = a$. We know that w_a is geometrically given by projecting f' onto the line connecting u and p_X, see Figure 3(b), but since we don't know f', it is not even clear how to find *any* w_s that is closer to f' than u is. In Subsection 4.1 we have provided intuitive arguments why this needs to be a non-trivial problem: The free parameter s defines a prior decision to what extent we attribute the non-uniformness of p_X to p_Y and to what extent to the non-linearity of f. Again, we propose the following heuristic procedure to iteratively adapt s during the SSL procedure: in each step n, we already know which value a_{n-1} minimizes $D(w_s \| f'_{n-1})$. In other words, among all possible deformations given by w_s, we can choose the one that yields the best prediction for the piecewise linear function f'_{n-1} interpolating the known values. Then, a_n converges to the optimal value a as shown by the following result which is proved in Appendix B:

Lemma 19 (continuity of projections) *Let f' be continuous and p_X be bounded from above. Define*

$$a_n := \mathrm{argmin}_{s \in I} D(w_s \| f'_n) \,.$$

Then we have

$$\lim_{n \to \infty} a_n = \mathrm{argmin}_{s \in I} D(w_s \| f') \,.$$

5. Conclusions

We have analyzed a semi-supervised interpolation for $Y = f(X)$ for an unknown strictly monotonically increasing function f. Whenever Y is the cause and X the effect the derivative of f tends to be high in regions where p_X is large – provided that one believes in the model assumptions of Information-Geometric Causal Inference. We have proposed two different SSL methods, one employs the fact that p_X is positively correlated with f', while the other one employs positive correlations between p_X and the logarithm of the slope. In both cases, the SSL method changes the value $\hat{y}_n$ inferred by standard linear interpolation by an amount that depends on the average probability densities of X in the intervals between x_n and the closest point to the left and to the right. It turns out that such a modified linear

5. It turns out to be equivalent to the dual version of (7) by replacing each relative entropy $D(p\|q)$ with $D(q\|p)$. It is known in information geometry (Amari, 2001) that many theorems have such a 'dual' counterpart.

interpolation outperforms standard linear interpolation with respect to two substantially different loss functions: the first one is a squared distance, the second one the Bayesian surprise.

To the best of our knowledge, this is the first theoretical result that links the performance of SSL to the causal direction, provided that one accepts the underlying independence assumption for P_{cause} and $P_{\text{effect}|\text{cause}}$. SSL-algorithms that employ P_X by changing the geometry of the input space accordingly have been described earlier Chapelle et al. (2006). For instance, P_X may define a notion of smoothness (e.g. via P_X-dependent kernels or graphs) and thus influence the regularization term. Here we have justified an appropriate change of the geometry based on a postulate that is linked to the causal direction.

Certainly, the notion of (in)dependence of P_X and $P_{Y|X}$ used throughout this article is a rather simplistic one. First, the deterministic scenario applies only to very specific causal relations in real life. Second, even for this case, one would not expect that independence between P_{cause} and $P_{\text{effect}|\text{cause}}$ always holds in the sense of vanishing correlations as discussed here. To find notions of (in)dependence that turn out to be related to the causal direction in realistic learning scenarios has to be left to the future.

Acknowledgments

The authors would like to thank Eleni Sgouritsa, Joris Mooij, and Jonas Peters for helpful remarks on the manuscript.

Appendix A. Proof of Lemma 10

Using (17) yields

$$\begin{aligned} & \log \text{pr}(y_n|x_1,\dots,x_n,y_1,\dots,y_{n-1}) \\ = \; & \text{pr}(y_1,\dots,y_n|x_1,\dots,x_n) - \log \text{pr}(y_1,\dots,y_{n-1}|x_1,\dots,x_{n-1})\,. \end{aligned}$$

We now compare the terms in (21) with those that occur in the same formula for $n+1$: the term

$$(\lambda(x^\circ_{i+1}-x^\circ_i)-1)\log(y^\circ_{i+1}-y^\circ_i) \tag{33}$$

is replaced with

$$(\lambda(x_{n+1}-x^\circ_i)-1)\log(y_{n+1}-y^\circ_i)+(\lambda(x^\circ_{i+1}-x_n)-1)\log(y^\circ_{i+1}-y_n)\,. \tag{34}$$

Splitting the term (33) into

$$(\lambda(x_n-x^\circ_i)-1)\log(y^\circ_{i+1}-y^\circ_i)+(\lambda(x_n-x^\circ_{i+1})-1)\log(y^\circ_{i+1}-y^\circ_i)+\log(y^\circ_{i+1}-y^\circ_i)\,,$$

the difference between (33) and (34) can be written as

$$\lambda(x_n-x^\circ_i)-1)\log\frac{y_n-y^\circ_i}{y^\circ_{i+1}-y^\circ_i}+\lambda(x^\circ_{i+1}-x_n)-1)\log\frac{y^\circ_{i+1}-y_n}{y^\circ_{i+1}-y^\circ_i}-\log(y^\circ_{i+1}-y^\circ_i)\,.$$

To understand how the normalization factors change from $n-1$ to n we observe that the term

$$\log\Gamma(\lambda(x^\circ_{i+1}-x^\circ_i))$$

is replaced with

$$\log\Gamma(\lambda(x_n - x_i^\circ)) + \log\Gamma(\lambda(x_{i+1}^\circ - x_n))\,.$$

Then the statement follows.

Appendix B. Proof of Lemma 19

We first consider the affine family of densities $q_\lambda := \lambda r_1 + (1-\lambda)r_2$, where r_1, r_2 are strictly positive densities with non-zero lower bound b. We then show that the value s minimizing $D(w_s\|q_\lambda)$ depends continuously on λ. To this end, we introduce the function

$$\ell(\lambda, s) := \frac{d}{ds} D(w_s\|q_\lambda) = \int (p(x)-1)\log\frac{w_s(x)}{q_\lambda}dx\,,$$

where the last equality is derived in analogy to (32) by replacing f' with q_λ in. Then

$$\begin{aligned}\frac{\partial}{\partial\lambda}\ell(\lambda,s) &= -\int_0^1 \frac{p(x)-1}{\lambda r_1(x) + (1-\lambda)r_2(x)}(r_1(x)-r_2(x))dx\\ \frac{\partial}{\partial s}\ell(\lambda,s) &= \int_0^1 \frac{(p(x)-1)^2}{w_s(x)}dx\,.\end{aligned}$$

Let $b > 0$ be a lower bound for r_1 and r_2 and $d > 0$ an upper bound for p_X. We then obtain

$$\left|\frac{\partial}{\partial\lambda}\ell(\lambda,s)\right| \le \frac{d}{b}\int |r_1(x) - r_2(x)|dx\,. \tag{35}$$

Moreover,

$$\left|\frac{\partial}{\partial s}\ell(\lambda,s)\right| \ge \frac{1}{1+d}\int (1-p(x))^2 dx\,. \tag{36}$$

Since (36) is non-zero because p_X is not the constant function 1 (otherwise it could not correlate with $\log f'$), the law of implicit functions states that we can locally find a function v (around some solution a) by

$$\ell(\lambda, v(\lambda)) = 0\,,$$

with

$$v'(\lambda) = \frac{\partial}{\partial\lambda}\ell(\lambda,a)\left(\frac{\partial}{\partial s}\ell(\lambda,a)\right)^{-1}\,.$$

The difference between the s-values s_1 and s_2 for r_1 and r_2, respectively, can be bounded from above by

$$|v(1) - v(0)| \le \sup_{\lambda\in[0,1]}|v'(\lambda)| \le \frac{d(d+1)}{b}\frac{\int|r_1(x)-r_2(x)|dx}{\int(1-p(x))^2dx}\,, \tag{37}$$

where the last inequality follows from combining (35) and (36). Since each f_n' is strictly positive and f_n' converges uniformly to f', which is strictly positive on the compact interval $[0,1]$, we can find a uniform lower bound b for the functions f_n'. Using (37) with $r_1 := f_n'$ and $r_2 := r_2$ yields

$$|a_n - a| \le \frac{d(d+1)}{b}\int |f_n'(x) - f'(x)|dx\,.$$

Then the right hand side converges to zero, again due to the uniform convergence of f_n' to f'.

Appendix C. Using the Dual Independence Postulate

Straightforward computation shows that the 'dual' independence postulate

$$\mathrm{Cov}[g', \log p_E] = 0$$

is equivalent to

$$D(g'\|p_C) = D(g'\|u) + (u\|p_C)\,.$$

Applying the function g to all distributions yields

$$D(u\|p_E) = D(u\|g^{-1'}) + D(g^{-1'}\|p_E)\,. \tag{38}$$

For the function h defined in (31) we observe that

$$\begin{aligned} h'(1) &= \int_0^1 (p(x) - u(x)) \log \frac{p(x)}{f'(x)} dx \\ &= D(p_X\|f') + D(u\|p_X) - D(u\|f')\,. \end{aligned}$$

Using

$$D(u\|p_X) = D(u\|f') + D(f'\|p_X)\,,$$

due to (38) yields

$$h'(1) = D(p_X\|f') + D(f'\|p_X) \geq 0\,,$$

with equality only for $f' = p_X$, i.e., if p_Y is uniform. Therefore the unique minimum a of h satisfies $s \leq 1$ with equality only for uniform input.

Appendix D. Comparing the Two Interpolation Schemes

We now explain why the SSL interpolation in Subsection 4.2 differs from the one in Subsection 4.1 not only by the fact that the former infers one specific value $\hat{y}_n^s$ while the latter provides a conditional distribution. We now see that the expectation of the conditional of the SSL version of the Dirichlet process does not coincide with $\hat{y}_n^s$ in Subsection 4.1. To this end, we recall that the expectation for the standard linear interpolation in Subsection 3.1 reads

$$\hat{y}_n = \frac{x_n - x_i^\circ}{x_{i+1}^\circ - x_i^\circ}(y_{i+1}^\circ - y_i^\circ) + y_i^\circ\,.$$

Now, we just have to replace each x-value by $W_s(x)$ and obtain:

Lemma 20 (expectation of deformed interpolation)

$$\hat{y}_n^s = \frac{W_s(x_n) - W_s(x_i^\circ)}{W_s(x_{i+1}^\circ) - W_s(x_i^\circ)}(y_{i+1}^\circ - y_i^\circ) + y_i^\circ\,. \tag{39}$$

To understand (39), we note that it amounts to multiplying the slope of f_n with some factor:

Lemma 21 (deformed interpolation in terms of derivatives)

$$(\hat{f}^s_n)' = f'_{n-1}\frac{(1-s)+sp_n}{(1-s)+sp_{n-1}} = f'_{n-1}\frac{w^s_n}{w^s_{n-1}}\,, \tag{40}$$

with $w^s_n := \mathbf{E}[w_s|J_n]$.

Proof: Rewrite (39) as

$$\frac{\hat{y}^s_n - y^\circ_i}{x_n - x^\circ_i} = \frac{y^\circ_{i+1} - y^\circ_i}{x^\circ_{i+1} - x^\circ_i}\frac{W_s(x_n) - W_s(x^\circ_i)}{x_n - x^\circ_i}\frac{x^\circ_{i+1} - x^\circ_i}{W_s(x^\circ_{i+1}) - W_s(x^\circ_i)}\,.$$

Then the right hand side can be written as $f'_n w^s_n/w^s_{n-1}$. ∎

To compare (40) to (23) we observe

$$s(p_n - p_{n-1}) = w^s_n - w^s_{n-1}\,.$$

Hence, (23) can also be written as

$$(\hat{f}^s_n)' = f'_{n-1} + w^s_n - w^s_{n-1},\,.$$

Therefore the additively deformed interpolation modifies f'_{n-1} by adding the difference $w^s_n - w^s_{n-1}$ as summand, while the SSL interpolation in Subsection 4.2 is multiplicative in the sense that it adds the quotient w^s_n/w^s_{n-1} as factor.

References

S. Amari. Information geometry on hierarchy of probability distributions. *IEEE Transactions on Information Theory*, 47(5):1701–1711, 2001.

S. Amari and H. Nagaoka. *Methods of Information Geometry*. Oxford University Press, 1993.

N. Balakrishnan and Nevzorov V. *A Primer on Statistical Distributions*. John Wileys, New Jersey, USA, 2003.

S. Ben-David, T. Lu, and D. Pl. Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning. In R. Servedio and T. Zhang, editors, *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 33–44. Omnipress, 2008.

D. Blackwell. Discreteness of Ferguson selections. *The Annals of Statistics*, 1(2):356–358, 1973.

O. Chapelle, B. Schölkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, USA, 2006.

P. Daniusis, D. Janzing, J. M. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schölkopf. Inferring deterministic causal relations. In *Proceedings of the 26th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 143–150. AUAI Press, 2010.

M. Darnstädt, H. Simon, and B. Szörényi. Unlabeled data does provably help. In N. Portier and T. Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 185–196, 2013.

D. Janzing and B. Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniušis, B. Steudel, and B. Schölkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 182–183:1–31, 2012.

D. Janzing, B. Steudel, N. Shajarisales, and B. Schölkopf. Justifying information-geometric causal inference. In V. Vovk, H. Papadopolous, and A. Gammerman, editors, *Measures of Complexity*, Festschrift for Alexey Chervonencis, pages 253–265. Springer Verlag, Heidelberg, 2015.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. On causal and anticausal learning. In Langford J. and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 1255–1262. ACM, 2012.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij. Semi-supervised learning in causal and anticausal settings. In B. Schölkopf, Z. Luo, and V. Vovk, editors, *Empirical Inference*, Festschrift in Honor of Vladimir Vapnik, pages 129–141. Springer, 2013.

E. Sgouritsa, D. Janzing, P. Hennig, and B. Schölkopf. Inference of cause and effect with unsupervised inverse regression. In G. Lebanon and S. Vishwanathan, editors, *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS)*, JMLR Workshop and Conference Proceedings, 2015.

G. Yuanyuan, X. Niu, and H. Zhang. An extensive empirical study on semi-supervised learning. In G. Webb, B. Liu, C. Zhang, D. Gunopulos, and X. Wu, editors, *10th IEEE International Conference on Data Mining (ICDM)*, pages 186–195. IEEE Computer Society, 2010.

T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. In P. Langley, editor, *17th International Conference on Machine Learning (ICML)*, pages 1191–1198. Morgan Kaufmann Publishers, 2000.

Journal of Machine Learning Research 16 (2015) 1949-2002 Submitted 6/15; Published 9/15

Towards an Axiomatic Approach to Hierarchical Clustering of Measures

Philipp Thomann PHILIPP.THOMANN@MATHEMATIK.UNI-STUTTGART.DE
Ingo Steinwart INGO.STEINWART@MATHEMATIK.UNI-STUTTGART.DE
Nico Schmid NICO.SCHMID@MATHEMATIK.UNI-STUTTGART.DE
Institute for Stochastics and Applications
University of Stuttgart, Germany

Editor: Alex Gammerman and Vladimir Vovk

Abstract

We propose some axioms for hierarchical clustering of probability measures and investigate their ramifications. The basic idea is to let the user stipulate the clusters for some elementary measures. This is done without the need of any notion of metric, similarity or dissimilarity. Our main results then show that for each suitable choice of user-defined clustering on elementary measures we obtain a unique notion of clustering on a large set of distributions satisfying a set of additivity and continuity axioms. We illustrate the developed theory by numerous examples including some with and some without a density.

Keywords: axiomatic clustering, hierarchical clustering, infinite samples clustering, density level set clustering, mixed Hausdorff-dimensions

1. Introduction

Clustering is one of the most basic tools to investigate unsupervised data: finding groups in data. Its applications reach from categorization of news articles over medical imaging to crime analysis. For this reason, a wealth of algorithms have been proposed, among the best-known being: k-means (MacQueen, 1967), linkage (Ward, 1963; Sibson, 1973; Defays, 1977), cluster tree (Stuetzle, 2003), DBSCAN (Ester et al., 1996), spectral clustering (Donath and Hoffman, 1973; von Luxburg, 2007), and expectation-maximization for generative models (Dempster et al., 1977). For more information and research on clustering we refer the reader to Jardine and Sibson (1971); Hartigan (1975); Kaufman and Rousseeuw (1990); Mirkin (2005); Gan et al. (2007); Kogan (2007); Ben-David (2015); Menardi (2015) and the references therein.

However, each *ansatz* has its own implicit or explicit definition of what clustering is. Indeed for k-means it is a particular Voronoi partition, for Hartigan (1975, Section 11.13) it is the collection of connected components of a density level set, and for generative models it is the decomposition of mixed measures into the parts. Stuetzle (2003) stipulates a grouping around the modes of a density, while Chacón (2014) uses gradient-flows. Thus, there is no universally accepted definition.

A good notion of clustering certainly needs to address the inherent random variability in data. This can be achieved by notions of clusterings for infinite sample regimes or complete knowledge scenarios—as von Luxburg and Ben-David (2005) put it. Such an approach has

various advantages: one can talk about ground-truth, can compare alternative clustering algorithms (empirically, theoretically, or in a combination of both by using artificial data), and can define and establish consistency and learning rates. Defining clusters as the connected components of density level sets satisfies all of these requirements. Yet it seems to be slightly *ad-hoc* and it will always be debatable, whether thin bridges should connect components, and whether close components should really be separated. Similar concerns may be raised for other infinite sample notions of clusterings such as Stuetzle (2003) and Chacón (2014).

In this work we address these and other issues by asking ourselves: *What does the set of clustering functions look like? What can defining properties—or axioms—of clustering functions be and what are their ramifications? Given such defining properties, are there functions fulfilling these? How many are there? Can a fruitful theory be developed? And finally, for which distributions do we obtain a clustering and for which not?*

These questions have led us to an axiomatic approach. The basic idea is to let the user stipulate the clusters for some elementary measures. Here, his choice does not need to rely on a metric or another pointwise notion of similarity though—only basic shapes for geometry and a separation relation have to be specified. Our main results then show that for each suitable choice we obtain a unique notion of clustering satisfying a set of additivity and continuity axioms on a large set of measures. These will be motivated in Section 1.2 and are defined in Axioms 1, 2, and 3. The major technical achievement of this work is Theorem 20: it establishes criteria (c.f. Definition 18) to ensure a unique limit structure, which in turn makes it possible to define a unique additive and continuous clustering in Theorem 21. Furthermore in Section 3.5 we explain how this framework is linked to density based clustering, and in the examples of Section 4.3 we investigate the consequences in the setting of mixed Hausdorff dimensions.

1.1 Related Work

Some axioms for clustering have been proposed and investigated, but to our knowledge, all approaches concern clustering of finite data. Jardine and Sibson (1971) were probably the first to consider axioms for hierarchical clusterings: these are maps of sets of dissimilarity matrices to sets of e.g. ultrametric matrices. Given such sets they obtain continuity and uniqueness of such a map using several axioms. This setting was used by Janowitz and Wille (1995) to classify clusterings that are equivariant for all monotone transformations of the values of the distance matrix. Later, Puzicha et al. (1999) investigate axioms for cost functions of data-partitionings and then obtain clustering functions as optimizers of such cost functions. They consider as well a hierarchical version, marking the last axiomatic treatment of that case until today. More recently, Kleinberg (2003) put forward an impossibility result. He gives three axioms and shows that any (non-hierarchical) clustering of distance matrices can fulfill at most two of them. Zadeh and Ben-David (2009) remedy the impossibility by restricting to k-partitions, and they use minimum spanning trees to characterize different clustering functions. A completely different setting is Meilă (2005) where an arsenal of axioms is given for distances of clustering partitions. They characterize some distances (variation of information, classification error metric) using different subsets of their axioms.

One of the reviewers brought clustering of discrete data to our attention. As far as we understand, consensus clustering (Mirkin, 1975; Day and McMorris, 2003) and additive

clustering (Shepard and Arabie, 1979; Mirkin, 1987) are popular in social studies clustering communities. What we call additive clustering in this work is something completely different though. Still, application of our notions to clustering of discrete structures warrants further research.

1.2 Spirit of Our Ansatz

Let us now give a brief description of our approach. To this end assume for simplicity that we wish to find a hierarchical clustering for certain distributions on $\mathbb{R}^d$. We denote the set of such distributions by $\mathcal{P}$. Then a clustering is simply a map c that assigns every $P \in \mathcal{P}$ to a collection $c(P)$ of non-empty events. Since we are interested in hierarchical clustering, $c(P)$ will always be a forest, i.e. we have

$$A, A' \in c(P) \implies A \perp A' \text{ or } A \subset A' \text{ or } A \supset A'. \tag{1}$$

Here $A \perp A'$ means *sufficiently distinct*, i.e. $A \cap A' = \emptyset$ or something stronger (cf. Definition 1. Following the idea that eventually one needs to store and process the clustering $c(P)$ on a computer, our first axiom assumes that $c(P)$ is *finite*. For a distribution with a continuous density the level set forest, i.e. the collection of all connected components of density level sets, will therefore *not* be viewed as a clustering. For densities with finitely many modes, however, this level set forest consists of long chains interrupted by finitely many branchings. In this case, the most relevant information for clustering is certainly represented at the branchings and not in the intermediate chains. Based on this observation, our second clustering axiom postulates that $c(P)$ does not contain chains. More precisely, if $s(F)$ denotes the forest that is obtained by replacing each chain in the forest F by the maximal element of the chain, our **structured forest axiom** demands that

$$s(c(P)) = c(P)\,. \tag{2}$$

To simplify notations we further extend the clustering to the cone defined by $\mathcal{P}$ by setting

$$c(\alpha P) := c(P) \tag{3}$$

for all $\alpha > 0$ and $P \in \mathcal{P}$. Equivalently we can view $\mathcal{P}$ as a collection of finite non-trivial measures and c as a map on $\mathcal{P}$ such that for $\alpha > 0$ and $P \in \mathcal{P}$ we have $\alpha P \in \mathcal{P}$ and $c(\alpha P) = c(P)$. It is needless to say that this extended view on clusterings does not change the nature of a clustering.

Our next two axioms are based on the observation that there do not only exist distributions for which the "right notion" of a clustering is debatable but there are also distributions for which everybody would agree about the clustering. For example, if P is the uniform distribution on a Euclidean ball B, then certainly everybody would set $c(P) = \{B\}$. Clearly, other such examples are possible, too, and therefore we view the determination of distributions with such simple clusterings as a *design decision*. More precisely, we assume that we have a collection $\mathcal{A}$ of closed sets, called **base sets** and a family $\mathcal{Q} = \{Q_A\}_{A \in \mathcal{A}} \subset \mathcal{P}$ called **base measures** with the property $A = \operatorname{supp} Q_A$ for all $A \in \mathcal{A}$. Now, our **base measure axiom** stipulates

$$c(Q_A) = \{A\}. \tag{4}$$

It is not surprising that different choices of $\mathcal{A}$, $\mathcal{Q}$, and $\perp$ may lead to different clusterings. In particular we will see that larger classes $\mathcal{A}$ usually result in more distributions for which we can construct a clustering satisfying all our clustering axioms. On the other hand, taking a larger class $\mathcal{A}$ means that more agreement needs to be sought about the distributions having a trivial clustering (4). For this reason the choice of $\mathcal{A}$ can be viewed as a trade-off.

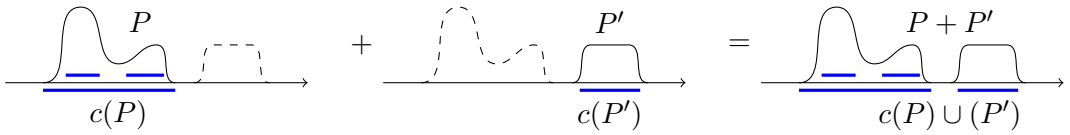


Figure 1: Example of disjoint additivity for two distributions having a density.

Axiom (4) only describes distributions that have a trivial clustering. However, there are also distributions for which everybody would agree on a non-trivial clustering. For example, if P is the uniform distribution on two well separated Euclidean balls B_1 and B_2, then the "natural" clustering would be $c(P) = \{B_1, B_2\}$. Our **disjoint additivity axiom** generalizes this observation by postulating

$$\operatorname{supp} P_1 \perp \operatorname{supp} P_2 \implies c(P_1 + P_2) = c(P_1) \cup c(P_2)\,. \tag{5}$$

In other words, if P consists of two spatially well separated sources P_1 and P_2, the clustering of P should reflect this spatial separation, see also Figure 1. Moreover note this axiom formalizes the vague term "spatially well separated" with the help of the relation $\perp$, which, like $\mathcal{A}$ and $\mathcal{Q}$ is a design parameter that usually influences the nature of the clustering.

The axioms (4) and (5) only described the horizontal behaviour of clusterings, i.e. the depth of the clustering forest is not affected by (4) and (5). Our second additivity axiom addresses this. To motivate it, assume that we have a $P \in \mathcal{P}$ and a base measure Q_A, e.g. a uniform distribution on A, such that $\operatorname{supp} P \subset A$. Then adding Q_A to P can be viewed as pouring uniform noise over P. Intuitively, this uniform noise should not affect the internal and possibly delicate clustering of P but only its roots, see also Figure 2. Our **base additivity axiom** formalizes this intuition by stipulating

$$\operatorname{supp} P \subset A \implies c(P + Q_A) = s\big(c(P) \cup \{A\}\big). \tag{6}$$

Here the structure operation $s(\,\cdot\,)$ is applied on the right-hand side to avoid a conflict with the structured forest axiom (2). Also note that it is this very axiom that directs our theory towards hierarchical clustering, since it controls the vertical growth of clusterings under a simple operation.

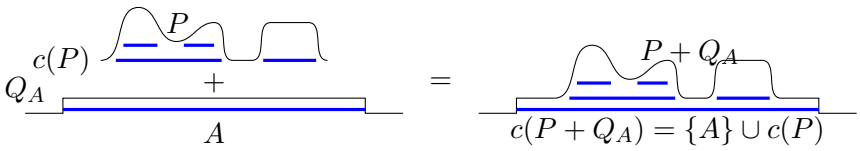


Figure 2: Example of base additivity.

Any clustering satisfying the axioms (1) to (6) will be called an **additive clustering**. Now the first, and rather simple part of our theory shows that under some mild technical assumptions there is a *unique* additive clustering on the set of **simple measures on forests**

$$\mathcal{S}(\mathcal{A}) := \left\{ \sum_{A\in F} \alpha_A Q_A \mid F \subset \mathcal{A} \text{ is a forest and } \alpha_A > 0 \text{ for all } A \in F \right\}.$$

Moreover, for $P \in \mathcal{S}(\mathcal{A})$ there is a unique representation $P = \sum_{A\in F} \alpha_A Q_A$ and the additive clustering is given by $c(P) = s(F)$.

Unfortunately, the set $\mathcal{S}(\mathcal{A})$ of simple measures, on which the uniqueness holds, is usually rather small. Consequently, additive clusterings on large collections $\mathcal{P}$ are far from being uniquely determined. Intuitively, we may hope to address this issue if we additionally impose some sort of continuity on the clusterings, i.e. an implication of the form

$$P_n \to P \implies c(P_n) \to c(P)\,. \tag{7}$$

Indeed, having an implication of the form (7), it is straightforward to show that the clustering is not only uniquely determined on $\mathcal{S}(\mathcal{A})$ but actually on the "closure" of $\mathcal{S}(\mathcal{A})$. To find a formalization of (7), we first note that from a user perspective, $c(P_n) \to c(P)$ usually describes a *desired* type of convergence. Following this idea, $P_n \to P$ then describes a *sufficient* condition for (7) to hold. In the remainder of this section we thus begin by presenting desirable properties $c(P_n) \to c(P)$ and resulting *necessary* conditions on $P_n \to P$.

Let us begin by assuming that all P_n are contained in $\mathcal{S}(\mathcal{A})$ and let us further denote the corresponding forests in the unique representation of P_n by F_n. Then we already know that $c(P_n) = s(F_n)$, so that the convergence on the right hand side of (7) becomes

$$s(F_n) \to c(P)\,. \tag{8}$$

Now, every $s(F_n)$, as well as $c(P)$, is a finite forest, and so a minimal requirement for (8) is that $s(F_n)$ and $c(P)$ are graph isomorphic, at least for all sufficiently large n. Moreover, we certainly also need to demand that every node in $s(F_n)$ converges to the corresponding node in $c(P)$. To describe the latter postulation more formally, we fix graph isomorphisms $\zeta_n : s(F_1) \to s(F_n)$ and $\zeta : s(F_1) \to c(P)$. Then our postulate reads as

$$\zeta_n(A) \to \zeta(A), \tag{9}$$

for all $A \in s(F_1)$. Of course, there do exist various notions for describing convergence of sets, e.g. in terms of the symmetric difference or the Hausdorff metric, so at this stage we need to make a decision. To motivate our choice, we first note that (9) actually contains two statements, namely, that $\zeta_n(A)$ converges for $n \to \infty$, and that its limit equals $\zeta(A)$. Now recall from various branches of mathematics that definitions of continuous extensions typically separate these two statements by considering approximating sequences that automatically converge. Based on this observation, we decided to consider monotone sequences in (9), i.e. we assume that $A \subset \zeta_1(A) \subset \zeta_2(A) \subset \dots$ for all $A \in s(F_1)$. Let us denote the resulting limit forest by F_∞, i.e.

$$F_\infty := \left\{ \bigcup_n \zeta_n(A) \mid A \in s(F_1) \right\},$$

which is indeed a forest under some mild assumptions on $\mathcal{A}$ and $\perp$. Moreover, $\zeta_\infty : s(F_1) \to F_\infty$ defined by $\zeta_\infty(A) := \bigcup_n \zeta_n(A)$ becomes a graph isomorphism, and hence (9) reduces to

$$\zeta_\infty(A) = \zeta(A) \qquad P\text{-almost surely for all } A \in s(F_1). \tag{10}$$

Summing up our considerations so far, we have seen that our demands on $c(P_n) \to c(P)$ imply some conditions on the forests associated to the sequence (P_n), namely $\zeta_n(A) \nearrow$ for all $A \in s(F_1)$. Without a formalization of $P_n \to P$, however, there is clearly no hope that this monotone convergence alone can guarantee (7). Like for (9), there are again various ways for formalizing a convergence of $P_n \to P$. To motivate our decision, we first note that a weak continuity axiom is certainly more desirable since this would potentially lead to more instances of clusterings. Furthermore, observe that (7) becomes weaker the stronger the notion of $P_n \to P$ is chosen. Now, if P_n and P had densities f_n and f, then one of the strongest notions of convergence would be $f_n \nearrow f$. In the absence of densities such a convergence can be expressed by $P_n \nearrow P$, i.e. by

$$P_n(B) \nearrow P(B) \qquad \text{for all measurable } B.$$

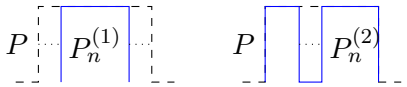
Combining these ideas we write $(P_n, F_n) \nearrow P$ iff $P_n \nearrow P$ and there are graph isomorphisms $\zeta_n : s(F_1) \to s(F_n)$ with $\zeta_n(A) \nearrow$ for all $A \in s(F_1)$. Our formalization of (7) then becomes

$$(P_n, F_n) \nearrow P \implies F_\infty = c(P) \text{ in the sense of (10)}, \tag{11}$$

which should hold for all $P_n \in \mathcal{S}(\mathcal{A})$ and their representing forests F_n.

While it seems tempting to stipulate such a continuity axiom it is unfortunately *inconsistent*. To illustrate this inconsistency, consider, for example, the uniform distribution P on $[0,1]$. Then P can be approximated by the following two sequences

$$P_n^{(1)} := \mathbf{1}_{[1/n,1-1/n]}P$$
$$P_n^{(2)} := \mathbf{1}_{[0,1/2-1/n]}P + \mathbf{1}_{[1/2,1]}P$$



By (11) the first approximation would then lead to the clustering $c(P) = \{[0,1]\}$ while the second approximation would give $c(P) = \{[0,1/2), [1/2,1]\}$.

Interestingly, this example not only shows that (11) is inconsistent but it also gives a hint how to resolve the inconsistency. Indeed the first sequence seems to be "adapted" to the limiting distribution, whereas the second sequence $(P_n^{(2)})$ is intuitively too complicated since its members have two clusters rather than the anticipated one cluster. Therefore, the idea to find a consistent alternative to (11) is to restrict the left-hand side of (11) to "adapted sequences", so that our **continuity axiom** becomes

$$(P_n, F_n) \nearrow P \text{ and } P_n \text{ is } P\text{-adapted for all } n \implies F_\infty = c(P) \text{ in the sense of (10)}.$$

In simple words, our main result then states that there exists exactly one such continuous clustering on the closure of $\mathcal{S}(\mathcal{A})$. The main message of this paper thus is:
Starting with very simple building blocks $\mathcal{Q} = (Q_A)_{A\in\mathcal{A}}$ for which we (need to) agree that they only have one trivial cluster $\{A\}$, we can construct a unique additive and continuous clustering on a rich set of distributions. Or, in other words, as soon as we have fixed $(\mathcal{A}, \mathcal{Q})$ and a separation relation $\perp$, there is no ambiguity left what a clustering is.

What is left is to explore how the choice of the **clustering base** $(\mathcal{A}, \mathcal{Q}, \perp)$ influences the resulting clustering. To this end, we first present various clustering bases, which, e.g. describe minimal thickness of clusters, their shape, and how far clusters need to be apart from each other. For distributions having a Lebesgue density we then illustrate how different clustering bases lead to different clusterings. Finally, we show that our approach goes beyond density-based clusterings by considering distributions consisting of several lower dimensional, overlapping parts.

2. Additive Clustering

In this section we introduce base sets, separation relations, and simple measures, as well as the corresponding axioms for clustering. Finally, we show that there exists a unique additive clustering on the set of simple measures.

Throughout this work let $\Omega = (\Omega, \mathcal{T})$ be a Hausdorff space and let $\mathcal{B} \supset \sigma(\mathcal{T})$ be a σ-algebra that contains the Borel sets. Furthermore we assume that $\mathcal{M} = \mathcal{M}_\Omega$ is the set of finite, non-zero, inner regular measures P on Ω. Similarly $\mathcal{M}_\Omega^\infty$ denotes the set of non-zero measures on Ω if Ω is a Radon space and else of non-zero, inner regular measures on Ω. In this respect, recall that any Polish space—i.e. a completely metrizable separable space—is Radon. In particular all open and closed subsets of $\mathbb{R}^d$ are Polish spaces and thus Radon. For inner regular measures the support is well-defined and satisfies the usual properties, see Appendix A for details. The set $\mathcal{M}_\Omega$ forms a cone: for all $P, P' \in \mathcal{M}_\Omega$ and all $\alpha > 0$ we have $P + P' \in \mathcal{M}_\Omega$ and $\alpha P \in \mathcal{M}_\Omega$.

2.1 Base Sets, Base Measures, and Separation Relations

Intuitively, any notion of a clustering should combine aspects of concentration and contiguousness. What is a possible core of this? On one hand clustering should be *local* in the sense of disjoint additivity, which was presented in the introduction: If a measure P is understood on two parts of its support and these parts are *nicely separated* then the clustering should be just a union of the two local ones. Observe that in this case $\operatorname{supp} P$ is not connected! On the other hand—in view of base clustering—base sets need to be impossible to partition into nicely separated components. Therefore they ought to be *nicely connected*. Of course, the meaning of *nicely connected* and *nicely separated* are interdependent, and highly disputable. For this reason, our notion of clustering assumes that both meanings are specified in front, e.g. by the user. Provided that both meanings satisfy certain technical criteria, we then show, that there exists exactly one clustering. To motivate how these technical criteria may look like, let us recall that for all connected sets A and all closed sets $B_1, \dots, B_k$ we have

$$A \subset B_1 \dot\cup \ldots \dot\cup B_k \implies \exists! i \leq k \colon A \subset B_i. \tag{12}$$

The left hand side here contains the condition that the $B_1, \dots, B_k$ are pairwise disjoint, for which we already introduced the following notation:

$$B \perp_\emptyset B' :\iff B \cap B' = \emptyset\,.$$

In order to transfer the notion of connectedness to other relations it is handy to generalize the notation $B_1 \dot\cup \ldots \dot\cup B_k$. To this end, let $\perp$ be a relation on subsets of Ω. Then we denote

the union $B_1 \cup \ldots \cup B_k$ of some $B_1, \ldots, B_k \subset \Omega$ by

$$B_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} B_k \,,$$

iff we have $B_i \perp B_j$ for all $i \neq j$. Now the key idea of the next definition is to generalize the notion of connectivity and separation by replacing $\perp_\emptyset$ in (12) by another suitable relation.

Definition 1 *Let $\mathcal{A} \subset \mathcal{B}$ be a collection of closed, non-empty sets. A symmetric relation $\perp$ defined on $\mathcal{B}$ is called a $\mathcal{A}$-**separation relation** iff the following holds:*

*(a) **Reflexivity**: For all $B \in \mathcal{B}$: $B \perp B \implies B = \emptyset$.*

*(b) **Monotonicity**: For all $A, A', B \in \mathcal{B}$:*

$$A \subset A' \text{ and } A' \perp B \implies A \perp B.$$

*(c) $\mathcal{A}$-**Connectedness**: For all $A \in \mathcal{A}$ and all closed $B_1, \ldots, B_k \in \mathcal{B}$:*

$$A \subset B_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} B_k \implies \exists i \leq k \colon A \subset B_i.$$

*Moreover, an $\mathcal{A}$-separation relation $\perp$ is called **stable**, iff for all $A_1 \subset A_2 \subset \ldots$ with $A_n \in \mathcal{A}$, all $n \geq 1$, and all $B \in \mathcal{B}$:*

$$A_n \perp B \quad \text{for all } n \geq 1 \implies \bigcup_{n \geq 1} A_n \perp B. \tag{13}$$

*Finally, given a separation relation $\perp$ then we say that B, B' are $\perp$-**separated**, if $B \perp B'$. We write $B \infty B'$ iff not $B \perp B'$, and say in this case that B, B' are $\perp$-**connected**.*

It is not hard to check that the disjointness relation $\perp_\emptyset$ is a stable $\mathcal{A}$-separation relation, whenever all $A \in \mathcal{A}$ are topologically connected. To present another example of a separation relation, we fix a metric d on Ω and some $\tau > 0$. Moreover, for $B, B' \subset \Omega$ we write

$$B \perp_\tau B' \;:\Longleftrightarrow\; d(B, B') \geq \tau \,.$$

In addition, recall that a $B \subset \Omega$ is τ-connected, if, for all $x, x' \in B$, there exists $x_0, \ldots, x_n \in B$ with $x_0 = x$, $x_n = x'$, and $d(x_{i-1}, x_i) < \tau$ for all $i = 1, \ldots, n$. Then it is easy to show that $\perp_\tau$ is an stable $\mathcal{A}$-separation relation if all $A \in \mathcal{A}$ are τ-connected. For more examples of separation relations we refer to Section 4.1.

It can be shown that $\perp_\emptyset$ is the weakest separation relation, i.e. for every $\mathcal{A}$-separation relation $\perp$ we have $A \perp A' \implies A \perp_\emptyset A'$ for all $A, A' \in \mathcal{A}$. We refer to Lemma 30, also showing that $\perp$-unions are unique, i.e., for all $A_1, \ldots, A_k$ and all $A'_1, \ldots, A'_{k'}$ in $\mathcal{A}$ we have

$$A_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} A_k = A'_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} A'_{k'} \implies \{A_1, \ldots, A_k\} = \{A'_1, \ldots, A'_{k'}\}.$$

Finally, the stability implication (13) is trivially satisfied for *finite* sequences $A_1 \subset \cdots \subset A_m$ in $\mathcal{A}$, since in this case we have $A_1 \cup \cdots \cup A_m = A_m$. For this reason stability will only become important when we will consider limits in Section 3.

We can now describe the properties a clustering base should satisfy.

Definition 2 *A (stable)* ***clustering base*** *is a triple $(\mathcal{A}, \mathcal{Q}, \perp)$ where $\mathcal{A} \subset \mathcal{B} \setminus \{\emptyset\}$ is a class of non-empty sets, $\perp$ is a (stable) $\mathcal{A}$-separation relation, and $\mathcal{Q} = \{Q_A\}_{A\in\mathcal{A}} \subset \mathcal{M}$ is a family of probability measures on Ω with the following properties:*

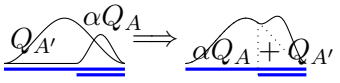
(a) ***Flatness****: For all $A, A' \in \mathcal{A}$ with $A \subset A'$ we either have $Q_{A'}(A) = 0$ or*

$$Q_A(\,\cdot\,) = \frac{Q_{A'}(\,\cdot \cap A)}{Q_{A'}(A)}\,.$$

(b) ***Fittedness****: For all $A \in \mathcal{A}$ we have $A = \operatorname{supp} Q_A$.*

We call a set A a ***base set*** *iff $A \in \mathcal{A}$ and a measure $\mathfrak{a} \in \mathcal{M}$ a* ***base measure on A*** *iff $A \in \mathcal{A}$ and there is an $\alpha > 0$ with $\mathfrak{a} = \alpha Q_A$.*

Let us motivate the two conditions of clustering bases. Flatness concerns nesting of base sets: Let $A \subset A'$ be base sets and consider the sum of their base measures $Q_A + Q_{A'}$. If the clustering base is not flat, weird things can happen—see the right. The way we defined flatness excludes such cases without taking densities into account. As a result we will be able to handle aggregations of measures of different Hausdorff-dimension in Section 4.3. Fittedness, on the other hand, establishes a link between the sets $A \in \mathcal{A}$ and their associated base measures.



Probably, the easiest example of a clustering base has measures of the form

$$Q_A(\,\cdot\,) = \frac{\mu(\,\cdot \cap A)}{\mu(A)} = \frac{1_A\, d\mu}{\mu(A)}\,, \tag{14}$$

where μ is some reference measure independent of Q_A. The next proposition shows that under mild technical assumptions such distributions do indeed provide a clustering base.

Proposition 3 *Let $\mu \in \mathcal{M}_\Omega^\infty$ and $\perp$ be a (stable) $\mathcal{A}$-separation relation for some $\mathcal{A} \subset \mathcal{K}(\mu)$, where*

$$\mathcal{K}(\mu) := \big\{\, C \in \mathcal{B} \mid 0 < \mu(C) < \infty \text{ and } C = \operatorname{supp}\ \mu(\cdot \cap C) \,\big\}$$

denotes the set of ***μ-support sets****. We write $\mathcal{Q}^{\mu,\mathcal{A}} := \big\{\, Q_A \mid A \in \mathcal{A} \,\big\}$, where Q_A is defined by (14). Then $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ is a (stable) clustering base.*

Interestingly, distributions of the form (14) are not the only examples for clustering bases. For further details we refer to Section 4.3, where we discuss distributions supported by sets of different Hausdorff dimension.

2.2 Forests, Structure, and Clustering

As outlined in the introduction we are interested in hierarchical clusterings, i.e. in clustering that map a finite measure to a forest of sets. In this section we therefore recall some fundamental definitions and notations for such forests.

Definition 4 *Let $\mathcal{A}$ be a class of closed, non-empty sets, $\perp$ be an $\mathcal{A}$-separation relation, and $\mathcal{C}$ be a class with $\mathcal{A} \subset \mathcal{C} \subset \mathcal{B} \setminus \{\emptyset\}$. We say that a non-empty $F \subset \mathcal{C}$ is a ($\mathcal{C}$-valued) $\perp$-**forest** iff*

$$A, A' \in F \implies A \perp A' \text{ or } A \subset A' \text{ or } A' \subset A.$$

We denote the set of all such finite forests by $\mathcal{F}_{\mathcal{C}}$ and write $\mathcal{F} := \mathcal{F}_{\mathcal{B}\setminus\{\emptyset\}}$.

A finite $\perp$-forest $F \in \mathcal{F}$ is partially ordered by the inclusion relation. The maximal elements $\max F := \{A \in F : \nexists A' \in F \text{ s.t. } A \subsetneq A'\}$ are called **roots** and the minimal elements $\min F := \{A \in F : \nexists A' \in F \text{ s.t. } A' \subsetneq A\}$ are called **leaves**. It is not hard to see that $A \perp A'$, whenever $A, A' \in F$ is a pair of roots or leaves. Moreover, the **ground** of F is

$$\mathbb{G}(F) := \bigcup_{A \in F} A\,,$$

that is, $\mathbb{G}(F)$ equals the union over the roots of F. Finally, F is a **tree**, iff it has only a single root, or equivalently, $\mathbb{G}(F) \in F$, and F is a **chain** iff it has a single leaf, or equivalently, iff it is totally ordered.

In addition to these standard notions, we often need a notation for describing certain sub-forests. Namely, for a finite forest $F \in \mathcal{F}$ with $A \in F$ we write

$$F\big|_{\supsetneq A} := \{A' \in F \mid A' \supsetneq A\}$$

for the chain of strict ancestors of A. Analogously, we will use the notations $F\big|_{\supset A}$, $F\big|_{\subset A}$, and $F\big|_{\subsetneq A}$ for the chain of ancestors of A (including A), the tree of descendants of A (including A), and the finite forest of strict descendants of A, respectively. We refer to Figure 3 for an example of these notations.

Definition 5 *Let F be a finite forest. Then we call $A_1, A_2 \in F$ **direct siblings** iff $A_1 \neq A_2$ and they have the same strict ancestors, i.e. $F\big|_{\supsetneq A_1} = F\big|_{\supsetneq A_2}$. In this case, any element*

$$A' \in \min F\big|_{\supsetneq A_1} = \min F\big|_{\supsetneq A_2}$$

*is called a **direct parent** of A_1 and A_2. On the other hand for $A, A' \in F$ we denote A' as a **direct child** of A, iff*

$$A' \in \max F\big|_{\subsetneq A}.$$

*Moreover, the **structure** of F is defined by*

$$s(F) := \Big\{ A \in F \,\Big|\, A \text{ is a root or it has a direct sibling } A' \in F \Big\}$$

*and F is a **structured forest** iff $F = s(F)$.*

For later use we note that direct siblings A_1, A_2 in a $\perp$-forest F always satisfy $A_1 \perp A_2$. Moreover, the structure of a forest is obtained by pruning all sub-chains in F, see Figure 3. We further note that $s(s(F)) = s(F)$ for all forests, and if F, F' are structured $\perp$-forests with $\mathbb{G}(F) \perp \mathbb{G}(F')$ then we have $s(F \cup F') = F \cup F'$.

Let us now present our first set of axioms for (hierarchical) clustering.

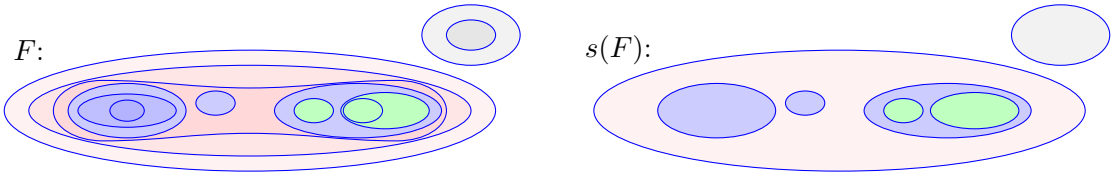


Figure 3: Illustrations of a forest F and of its structure $s(F)$.

Axiom 1 (Clustering) *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $\mathcal{P} \subset \mathcal{M}_\Omega$ be a set of measures with $\mathcal{Q} \subset \mathcal{P}$. A map $c\colon \mathcal{P} \to \mathcal{F}$ is called an **$\mathcal{A}$-clustering** if it satisfies:*

*(a) **Structured**: For all $P \in \mathcal{P}$ the forest $c(P)$ is structured, i.e. $c(P) = s(c(P))$.*

*(b) **ScaleInvariance**: For all $P \in \mathcal{P}$ and $\alpha > 0$ we have $\alpha P \in \mathcal{P}$ and $c(\alpha P) = c(P)$.*

*(c) **BaseMeasureClustering**: For all $A \in \mathcal{A}$ we have $c(Q_A) = \{A\}$.*

Note that the scale invariance is solely for notational convenience. Indeed, we could have defined clusterings for distributions only, in which case the scale invariance would have been obsolete. Moreover, assuming that a clustering produces structured forests essentially means that the clustering is only interested in the skeleton of the cluster forest. Finally, the axiom of base measure clustering means that we have a set of elementary measures, namely the base measures, for which we already agreed upon that they can only be clustered in a trivial way. In Section 4 we will present a couple of examples of $(\mathcal{A}, \mathcal{Q}, \bot)$ for which such an agreement is possible. Finally note that these axioms guarantee that if $c\colon \mathcal{P} \to \mathcal{F}$ is a clustering and $\mathfrak{a}$ is a base measure on A then $\mathfrak{a} \in \mathcal{P}$ and $c(\mathfrak{a}) = \{A\}$.

2.3 Additive Clustering

So far our axioms only determine the clusterings for base measures. Therefore, the goal of this subsection is to describe the behaviour of clusterings on certain combinations of measures. Furthermore, we will show that the axioms describing this behaviour are consistent and uniquely determine a hierarchical clustering on a certain set of measures induced by $\mathcal{Q}$.

Let us begin by introducing the axioms of additivity which we have already described and motivated in the introduction.

Axiom 2 (Additive Clustering) *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $\mathcal{P} \subset \mathcal{M}_\Omega$ be a set of measures with $\mathcal{Q} \subset \mathcal{P}$. A clustering $c\colon \mathcal{P} \to \mathcal{F}$ is called **additive** iff the following conditions are satisfied:*

*(a) **DisjointAdditivity**: For all $P_1, \dots, P_k \in \mathcal{P}$ with mutually $\bot$-separated supports, i.e. $\operatorname{supp} P_i \;\bot\; \operatorname{supp} P_j$ for all $i \neq j$, we have $P_1 + \ldots + P_k \in \mathcal{P}$ and*

$$c(P_1 + \ldots + P_n) = c(P_1) \cup \cdots \cup c(P_n)\,.$$

*(b) **BaseAdditivity**: For all $P \in \mathcal{P}$ and all base measures $\mathfrak{a}$ with $\operatorname{supp} P \subset \operatorname{supp} \mathfrak{a}$ we have $\mathfrak{a} + P \in \mathcal{P}$ and*

$$c(\mathfrak{a} + P) = s\big(\{\operatorname{supp} \mathfrak{a}\} \cup c(P)\big).$$

Our next goal is to show that there exist additive clusterings and that these are uniquely on a set $\mathcal{S}$ of measures that, in some sense, is spanned by $\mathcal{Q}$. The following definition introduces this set.

Definition 6 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $F \in \mathcal{F}_\mathcal{A}$ be an $\mathcal{A}$-valued finite $\bot$-forest. A measure Q is **simple on** F iff there exist base measures $\mathfrak{a}_A$ on $A \in F$ such that*

$$Q = \sum_{A \in F} \mathfrak{a}_A. \tag{15}$$

We denote the set of all simple measures with respect to $(\mathcal{A}, \mathcal{Q}, \bot)$ by $\mathcal{S} := \mathcal{S}(\mathcal{A})$.

Figure 4 provides an example of a simple measure. The next lemma shows that the representation 15 of simple measures is actually unique.

Lemma 7 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $Q \in \mathcal{S}(\mathcal{A})$. Then there exists exactly one $F_Q \in \mathcal{F}_\mathcal{A}$ such that Q is simple on F_Q. Moreover, the representing base measures $\mathfrak{a}_A$ in (15) are also unique and we have $\operatorname{supp} Q = \mathbb{G}F_Q$.*

Using Lemma 7 we can now define certain restrictions of simple measures $Q \in \mathcal{S}(\mathcal{A})$ with representation (15). Namely, any subset $F' \subset F$ gives a measure

$$Q\big|_{F'} := \sum_{A \in F'} \mathfrak{a}_A \,.$$

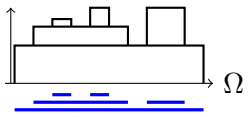


Figure 4: Simple measure.

We write $Q\big|_{\supset A} := Q\big|_{F\big|_{\supset A}}$ and similarly $Q\big|_{\supsetneq A}, Q\big|_{\subset A}, Q\big|_{\subsetneq A}$.

With the help of Lemma 7 it is now easy to explain how a possible additive clustering could look like on $\mathcal{S}(\mathcal{A})$. Indeed, for a $Q \in \mathcal{S}(\mathcal{A})$, Lemma 7 provides a unique finite forest $F_Q \in \mathcal{F}_\mathcal{A}$ that represents Q and therefore the structure $s(F_Q)$ is a natural candidate for a clustering of Q. The next theorem shows that this idea indeed leads to an additive clustering and that every additive clustering on $\mathcal{S}(\mathcal{A})$ retrieves the structure of the underlying forest of a simple measure.

Theorem 8 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $\mathcal{S}(\mathcal{A})$ the set of simple measures. Then we can define an additive $\mathcal{A}$-clustering $c : \mathcal{S}(\mathcal{A}) \to \mathcal{F}_\mathcal{A}$ by*

$$c(Q) := s(F_Q)\,, \qquad\qquad Q \in \mathcal{S}(\mathcal{A})\,. \tag{16}$$

Moreover, every additive $\mathcal{A}$-clustering $c : \mathcal{P} \to \mathcal{F}$ satisfies both $\mathcal{S}(\mathcal{A}) \subset \mathcal{P}$ and (16).

3. Continuous Clustering

As described in the introduction, we typically need, besides additivity, also some notion of continuity for clusterings. The goal of this section is to introduce such a notion and to show that, similarly to Theorem 8, this continuity uniquely defines a clustering on a suitably defined extension of $\mathcal{S}(\mathcal{A})$.

To this end, we first introduce a notion of monotone convergence for sequences of simple measures that does not change the graph structure of the corresponding clusterings given by Theorem 8. We then discuss a richness property of the clustering base, which essentially ensures that we can approximate the non-disjoint union of two base sets by another base set. In the next step we describe monotone sequences of simple measures that are in some sense adapted to the limiting distribution. In the final part of this section we then axiomatically describe continuous clusterings and show both their existence and their uniqueness.

3.1 Isomonotone Limits

The goal of this section is to introduce a notion of monotone convergence for simple measures that preserves the graph structure of the corresponding clusterings.

Our first step in this direction is done in the following definition that introduces a sort of monotonicity for set-valued isomorphic forests.

Definition 9 *Let $F, F' \in \mathcal{F}$ be two finite forests. Then F and F' are **isomorphic**, denoted by $F \cong F'$, iff there is a bijection $\zeta\colon F \to F'$ such that for all $A, A' \in F$ we have:*

$$A \subset A' \iff \zeta(A) \subset \zeta(A'). \tag{17}$$

Moreover, we write $F \leq F'$ iff $F \cong F'$ and there is a map $\zeta : F \to F'$ satisfying 17 and

$$A \subset \zeta(A)\,, \qquad\qquad A \in F. \tag{18}$$

*In this case, the map ζ, which is uniquely determined by (17), (18) and the fact that F and F' are finite, is called the **forest relating map** (FRM) between F and F'.*

Forests can be viewed as directed acyclic graphs: There is an edge between A and A' in F iff $A \subset A'$ and no other node is in between. Then $F \cong F'$ holds iff F and F' are isomorphic as directed graphs. From this it becomes clear that $\cong$ is an equivalence relation. Moreover, the relation $F \leq F'$ means that each node A of F can be graph isomorphically mapped to a node of F' that contains A, see Figure 5 for an illustration. Note that $\leq$ is a partial order on $\mathcal{F}$ and in particular it is transitive. Consequently, if we have finite forests $F_1 \leq \cdots \leq F_k$ then $F_1 \leq F_k$ and there is an FRM $\zeta_k\colon F_1 \to F_k$. This observation is used in the following definition, which introduces monotone sequences of forests and their limit.

Definition 10
*An **isomonotone sequence of forests** is a sequence of finite forests $(F_n)_n \subset \mathcal{F}$ such that $s(F_n) \leq s(F_{n+1})$ for all $n \geq 1$. If this is the case, we define the limit by*

$$F_\infty := \lim_{n\to\infty} s(F_n) := \left\{ \bigcup_{n\geq 1} \zeta_n(A) \mid A \in s(F_1) \right\},$$

where $\zeta_n\colon s(F_1) \to s(F_n)$ is the FRM obtained from $s(F_1) \leq s(F_n)$.

It is easy to see that in general, the limit forest F_∞ of an isomonotone sequence of $\mathcal{A}$-valued forests is not $\mathcal{A}$-valued. To describe the values of F_∞ we define the **monotone closure** of an $\mathcal{A} \subset \mathcal{B}$ by

$$\bar{\mathcal{A}} := \Big\{ \bigcup_{n\geq 1} A_n \mid A_n \in \mathcal{A} \text{ and } A_1 \subset A_2 \subset \dots \Big\}.$$

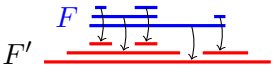


Figure 5: $F \leq F'$ and the arrows indicate ζ.

The next lemma states some useful properties of $\bar{\mathcal{A}}$ and F_∞.

Lemma 11 *Let $\perp$ be an $\mathcal{A}$-separation relation. Then $\perp$ is actually an $\bar{\mathcal{A}}$-separation relation. Moreover, if $\perp$ is stable and $(F_n) \subset \mathcal{F}_\mathcal{A}$ is an isomonotone sequence then $F_\infty := \lim_n s(F_n)$ is an $\bar{\mathcal{A}}$-valued $\perp$-forest and we have $s(F_n) \leq F_\infty$ for all $n \geq 1$.*

Unlike forests, it is straightforward to compare two measures Q_1 and Q_2 on $\mathcal{B}$. Indeed, we say that Q_2 **majorizes** Q_1, in symbols $Q_1 \leq Q_2$, iff

$$Q_1(B) \leq Q_2(B), \qquad \text{for all } B \in \mathcal{B}.$$

For $(Q_n) \subset \mathcal{M}$ and $P \in \mathcal{M}$, we similarly speak of **monotone convergence** $Q_n \uparrow P$ iff $Q_1 \leq Q_2 \leq \dots \leq P$ and

$$\lim_{n\to\infty} Q_n(B) = P(B), \qquad \text{for all } B \in \mathcal{B}.$$

Clearly, $Q \leq Q'$ implies $\operatorname{supp} Q \subset \operatorname{supp} Q'$ and it is easy to show, that $Q_n \uparrow P$ implies

$$P\big(\operatorname{supp} P \setminus \bigcup_{n\geq 1} \operatorname{supp} Q_n\big) = 0.$$

We will use such arguments throughout this section. For example, if $\mathfrak{a}, \mathfrak{a}'$ are base measures on A, A' with $\mathfrak{a} \leq \mathfrak{a}'$ then $A \subset A'$. With the help of these preparations we can now define isomonotone convergence of simple measures.

Definition 12 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and $(Q_n) \subset \mathcal{S}(\mathcal{A})$ be a sequence of simple measures on finite forests $(F_n) \subset \mathcal{F}_\mathcal{A}$. Then **isomonotone convergence**, denoted by $(Q_n, F_n) \uparrow P$, means that both*

$$Q_n \uparrow P \quad \text{and} \quad s(F_1) \leq s(F_2) \leq \dots .$$

In addition, $\bar{\mathcal{S}} := \bar{\mathcal{S}}(\mathcal{A})$ denotes the set of all isomonotone limits, i.e.

$$\bar{\mathcal{S}}(\mathcal{A}) = \big\{ P \in \mathcal{M} \mid (Q_n, F_n) \uparrow P \text{ for some } (Q_n) \subset \mathcal{S}(\mathcal{A}) \text{ on } (F_n) \subset \mathcal{F}_\mathcal{A} \big\}.$$

For a measure $P \in \bar{\mathcal{S}}(\mathcal{A})$ it is probably tempting to define its clustering by $c(P) := \lim_n s(F_n)$, where $(Q_n, F_n) \uparrow P$ is some isomonotone sequence. Unfortunately, such an approach does not yield a well-defined clustering as we have discussed in the introduction. For this reason, we need to develop some tools that help us to distinguish between "good" and "bad" isomonotone approximations. This is the goal of the following two subsections.

3.2 Kinship and Subadditivity

In this subsection we present and discuss a technical assumption on a clustering base that will make it possible to obtain unique continuous clusterings.

Let us begin by introducing a notation that will be frequently used in the following. To this end, we fix a clustering base $(\mathcal{A}, \mathcal{Q}, \perp)$ and a $P \in \mathcal{M}$. For $B \in \mathcal{B}$ we then define

$$\mathcal{Q}_P(B) := \{\alpha Q_A \mid \alpha > 0, A \in \mathcal{A}, B \subset A, \alpha Q_A \leq P\},$$

i.e. $\mathcal{Q}_P(B)$ denotes the set of all basic measures below P whose support contains B. Now, our first definition describes events that can be combined in a base set:

Definition 13 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and $P \in \mathcal{M}$. Two non-empty $B, B' \in \mathcal{B}$ are called **kin below** P, denoted as $B \sim_P B'$, iff $\mathcal{Q}_P(B \cup B') \neq \emptyset$, i.e., iff there is a base measure $\mathfrak{a} \in \mathcal{Q}$ such that the following holds:*

$$(a)\ B \cup B' \subset \operatorname{supp} \mathfrak{a} \qquad\qquad (b)\ \mathfrak{a} \leq P.$$

*Moreover, we say that every such $\mathfrak{a} \in \mathcal{Q}_P(B \cup B')$ **supports** B and B' below P.*

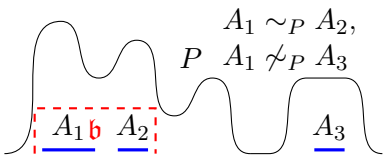


Figure 6: Kinship.

Kinship of two events can be used to test whether they belong to the same root in the cluster forest. To illustrate this we consider two events B and B' with $B \not\sim_P B'$. Moreover, assume that there is an $A \in \mathcal{A}$ with $B \cup B' \subset A$. Then $B \not\sim_P B'$ implies that for all such A there is no $\alpha > 0$ with $\alpha Q_A \leq P$. This situation is displayed on the right-hand side of Figure 6. Now assume that we have two base measures $\mathfrak{a}, \mathfrak{a}' \leq P$ on $A, A' \in \mathcal{A}$ that satisfy $A \sim_P A'$ and $P(A \cap A') > 0$. If $\mathcal{A}$ is rich in the sense of $A \cup A' \in \mathcal{A}$, then we can find a base measure $\mathfrak{b}$ on $B := A \cup A'$ with $\mathfrak{a} \leq \mathfrak{b} \leq P$ or $\mathfrak{a}' \leq \mathfrak{b} \leq P$. The next definition relaxes the requirement $A \cup A' \in \mathcal{A}$, see also Figure 7 for an illustration.

Definition 14 *Let $P \in \mathcal{M}_\Omega^\infty$ be a measure. For $B, B' \in \mathcal{B}$ we write*

$$B \perp\!\!\!\perp_P B' \ :\Longleftrightarrow\ P(B \cap B') = 0 \text{ and}$$
$$B \between_P B' \ :\Longleftrightarrow\ P(B \cap B') > 0\,.$$

*Moreover, a clustering base $(\mathcal{A}, \mathcal{Q}, \perp)$ is called **P-subadditive** iff for all base measures $\mathfrak{a}, \mathfrak{a}' \leq P$ on $A, A' \in \mathcal{A}$ we have*

$$A \between_P A' \implies \exists \mathfrak{b} \in \mathcal{Q}_P(A \cup A') \colon \mathfrak{b} \geq \mathfrak{a} \text{ or } \mathfrak{b} \geq \mathfrak{a}'. \tag{19}$$

Note that the implication (19) in particular ensures $\mathcal{Q}_P(A \cup A') \neq \emptyset$, i.e. $A \sim_P A'$. Moreover, the relation $\perp\!\!\!\perp_P$ is weaker than any separation relation $\perp$ since we obviously have $A \between_P A' \implies A \between_\emptyset A' \implies A \between A'$, where the second implication is shown in Lemma 30. The following definition introduces a stronger notion of additivity.

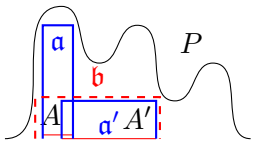


Figure 7: P-subadditivity.

Definition 15 *Let ∞ be a relation on $\mathcal{B}$. An $\mathcal{A} \subset \mathcal{B}$ is ∞-**additive** iff for all $A, A' \in \mathcal{A}$*

$$A \infty A' \implies A \cup A' \in \mathcal{A}.$$

The next proposition compares the several notions of (sub)-additivity. In particular it implies that if $\mathcal{A}$ is $\infty_\emptyset$-additive then $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ is P-subadditive for all $P \in \mathcal{M}$.

Proposition 16 *Let $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ be a clustering base as in Proposition 3. If $\mathcal{A}$ is ∞_P-additive for some $P \in \mathcal{M}$, then $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ is P-subadditive. Conversely, if $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ is P-subadditive for all $P \ll \mu$ then $\mathcal{A}$ is ∞_μ-additive and thus also ∞_P-additive.*

3.3 Adapted Simple Measures

We have already seen that isomonotone approximations by simple measures are not structurally unique. In this subsection we will therefore identify the most economical structure needed to approximate a distribution by simple measures. Such most parsimonious structures will then be used to define continuous clusterings.

Let us begin by introducing a different view on simple measures.

Definition 17 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and Q be a simple measure on $F \in \mathcal{F}_\mathcal{A}$ with the unique representation $Q = \sum_{A \in F} \alpha_A Q_A$. We define the map $\lambda_Q \colon F \to \mathcal{Q}$ by*

$$\lambda_Q(A) := \left(\sum_{A' \in F \colon A' \supset A} \alpha_{A'} Q_{A'}(A) \right) \cdot Q_A \,, \qquad A \in F.$$

*Moreover, we call the base measure $\lambda_Q(A) \in \mathcal{Q}$ the **level of A in Q**.*

In some sense, the level of an A in Q combines all ancestor measures including Q_A and then restricts this combination to A, see Figure 8 for an illustration of the level of a node. With the help of levels we can now describe structurally economical approximations of measures by simple measures.

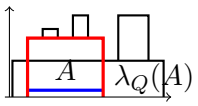


Figure 8: Level.

Definition 18 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and $P \in \mathcal{M}_\Omega$ a finite measure. Then a simple measure Q on a forest $F \in \mathcal{F}_\mathcal{A}$ is P-**adapted** iff all direct siblings A_1, A_2 in F are:*

*(a) P-**grounded**: if they are kin below P, i.e. $\mathcal{Q}_P(A_1 \cup A_2) \neq \emptyset$, then there is a parent around them in F.*

*(b) P-**fine**: every $\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup A_2)$ can be majorized by a base measure $\tilde{\mathfrak{b}}$ that supports all direct siblings $A_1, \ldots, A_k$ of A_1 and A_2, i.e.*

$$\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup A_2) \implies \exists \tilde{\mathfrak{b}} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_k) \text{ with } \tilde{\mathfrak{b}} \geq \mathfrak{b}.$$

*(c) **strictly motivated**: for their levels $\mathfrak{a}_1 := \lambda_Q(A_1)$ and $\mathfrak{a}_2 := \lambda_Q(A_2)$ in Q there is an $\alpha \in (0,1)$ such that every base measure $\mathfrak{b}$ that supports them below P is not larger than $\alpha\mathfrak{a}_1$ or $\alpha\mathfrak{a}_2$, i.e.*

$$\forall \mathfrak{b} \in \mathcal{Q} : \mathfrak{b} \geq \alpha\mathfrak{a}_1 \text{ or } \mathfrak{b} \geq \alpha\mathfrak{a}_2 \implies \mathfrak{b} \notin \mathcal{Q}_P(A_1 \cup A_2). \tag{20}$$

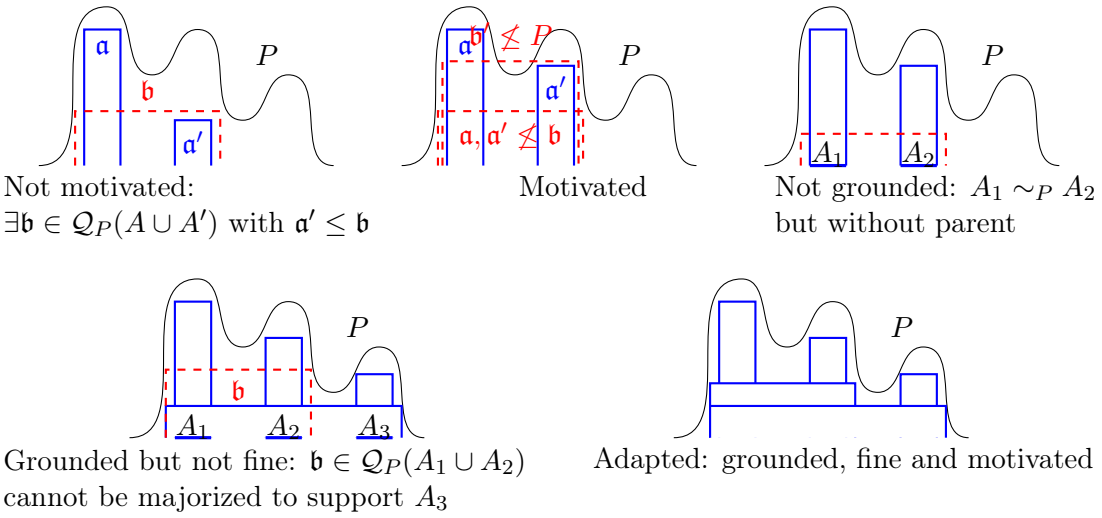


Figure 9: Illustrations for motivated, grounded, fine, and therefore adapted.

Finally, an isomonotone sequence $(Q_n, F_n) \uparrow P$ is adapted, if Q_n is P-adapted for all $n \geq 1$.

Since siblings are $\perp$-separated, they are $\perp\!\!\!\perp_P$-separated, so strict motivation is no contradiction to P-subadditivity. Levels are called **motivated** iff they satisfy condition (20) for $\alpha = 1$. Figure 9 illustrates the three conditions describing adapted measures. It can be shown that if $\mathcal{A}$ is ∞-additive, then any isomonotone sequence can be made adapted.

The following self-consistency result shows that every simple measure is adapted to itself. This result will guarantee that the extension of the clustering from $\mathcal{S}$ to $\bar{\mathcal{S}}$ is indeed an extension.

Proposition 19 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base. Then every $Q \in \mathcal{S}(\mathcal{A})$ is Q-adapted.*

3.4 Continuous Clustering

In this subsection we finally introduce continuous clusterings with the help of adapted, isomonotone sequences. Furthermore, we will show the existence and uniqueness of such clusterings.

Let us begin by introducing a notation that will be used to identify two clusterings as identical. To this end let $F_1, F_2 \in \mathcal{F}$ be two forests and $P \in \mathcal{M}_\Omega$ be finite measure. Then we write

$$F_1 =_P F_2\,,$$

if there exists a graph isomorphism $\zeta : F_1 \to F_2$ such that $P(A \triangle \zeta(A)) = 0$ for all $A \in F_1$. Now our first result shows that adapted isomonotone limits of two different sequences coincide in this sense.

Theorem 20 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a stable clustering base and $P \in \mathcal{M}_\Omega$ be a finite measure such that $\mathcal{A}$ is P-subadditive. If $(Q_n, F_n), (Q'_n, F'_n) \uparrow P$ are adapted isomonotone sequences then we have*

$$\lim_n s(F_\infty) =_P \lim_n s(F'_\infty)\,.$$

Theorem 20 shows that different adapted sequences approximating a measure P necessarily have isomorphic forests and that the corresponding limit nodes of the forests coincide up to P-null sets. This result makes the following axiom possible.

Axiom 3 (Continuous Clustering) *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base, $\mathcal{P} \subset \mathcal{M}_\Omega$ be a set of measures. We say that $c\colon \mathcal{P} \to \mathcal{F}$ is a* ***continuous clustering****, if it is an additive clustering and for all $P \in \mathcal{P}$ and all adapted isomonotone sequences $(Q_n, F_n) \uparrow P$ we have*

$$c(P) =_P \lim_n s(F_\infty)\,.$$

The following, main result of this section shows that there exist continuous clusterings and that they are uniquely determined on a large subset of $\bar{\mathcal{S}}(\mathcal{A})$.

Theorem 21 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a stable clustering base and set*

$$\mathcal{P}_\mathcal{A} := \big\{\, P \in \bar{\mathcal{S}}(\mathcal{A}) \mid \mathcal{A} \text{ is } P\text{-subadditive and there is } (Q_n, F_n) \nearrow P \text{ adapted} \big\}.$$

Then there exists a continuous clustering $c_\mathcal{A}\colon \mathcal{P}_\mathcal{A} \to \mathcal{F}_{\bar{\mathcal{A}}}$. Moreover, $c_\mathcal{A}$ is unique on $\mathcal{P}_\mathcal{A}$, that is, for all continuous clusterings $c\colon \tilde{\mathcal{P}} \to \mathcal{F}$ we have

$$c_\mathcal{A}(P) =_P c(P)\,, \qquad\qquad P \in \mathcal{P}_\mathcal{A}\,.$$

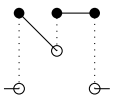
Recall from Proposition 16 that $\mathcal{A}$ is P-subadditive for all $P \in \mathcal{M}_\Omega$ if $\mathcal{A}$ is $\infty_\emptyset$-additive. It can be shown that if $\mathcal{A}$ is $\infty_\mathcal{A}$-additive, then any isomonotone sequence can be made adapted. In this case we thus have $\mathcal{P}_\mathcal{A} = \bar{\mathcal{S}}(\mathcal{A})$ and Theorem 21 shows that there exists a unique continuous clustering on $\bar{\mathcal{S}}(\mathcal{A})$.

3.5 Density Based Clustering

Let us recall from Proposition 3 that a simple way to define a set of base measures $\mathcal{Q}$ was with the help of a reference measure μ. Given a stable separation relation $\perp$, we denoted the resulting stable clustering base by $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$. Now observe that for this clustering base every $Q \in \mathcal{S}(\mathcal{A})$ is μ-absolutely continuous and its unique representation yields the μ-density $f = \sum_{A\in F} \alpha_A 1_A$ for suitable coefficients $\alpha_A > 0$. Consequently, each level set $\{f > \lambda\}$ consists of some elements $A \in F$, and if all elements in $\mathcal{A}$ are connected, the additive clustering $c(Q)$ of Q thus coincides with the "classical" cluster tree obtained from the level sets. It is therefore natural to ask, whether such a relation still holds for continuous clusterings on distributions $P \in \mathcal{P}_\mathcal{A}$.

Clearly, the first answer to this question needs to be negative, since in general the cluster tree is an infinite forest whereas our clusterings are always finite. To illustrate this, let us consider the **Factory** density on $[0,1]$, which is defined by

$$f(x) := \begin{cases} 1-x, & \text{if } x \in [0, \frac{1}{2}) \\ 1, & \text{if } x \in [\frac{1}{2}, 1] \end{cases}$$



Clearly, this gives the following $\perp_\emptyset$-decomposition of the level sets:

$$\{f > \lambda\} = \begin{cases} [0,1], & \text{if } \lambda < \frac{1}{2}, \\ [0, 1-\lambda) \overset{\perp_\emptyset}{\cup} [\frac{1}{2}, 1], & \text{if } \frac{1}{2} \le \lambda < 1, \end{cases}$$

which leads to the clustering forest $F_f = \{ [0,1], [\frac{1}{2},1] \} \cup \{ [0,1-\lambda) \mid \frac{1}{2} \leq \lambda < 1 \}$. Now observe that even though F_f is infinite, it is as graph somehow simple: there is a root $[0,1]$, a node $[\frac{1}{2},1]$, and an infinite chain $[0,1-\lambda)$, $\frac{1}{2} \leq \lambda < 1$. Replacing this chain by its supremum $[0,\frac{1}{2})$ we obtain the structured forest

$$\{[0,1],\, [0,\tfrac{1}{2}), [\tfrac{1}{2},1]\},$$

for which we can then ask whether it coincides with the continuous clustering obtained from $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp_\emptyset)$ if $\mathcal{A}$ consists of all closed intervals in $[0,1]$ and μ is the Lebesgue measure.

To answer this question we first need to formalize the operation that assigns a structured to an infinite forest. To this end, let F be an arbitrary $\perp$-forest. We say that $\mathcal{C} \subset F$ is a pure chain, iff for all $C, C' \in \mathcal{C}$ and $A \in F \setminus \mathcal{C}$ the following two implications hold:

$$A \subset C \implies A \subset C',$$
$$C \subset A \implies C' \subset A.$$

Roughly speaking, the first implication ensures that no node above a bifurcation is contained in the chain, while the second implication ensures that no node below a bifurcation is contained in the chain. With this interpretation in mind it is not surprising that we can define the structure of the forest F with the help of the maximal pure chains by setting

$$s(F) := \left\{ \bigcup \mathcal{C} \mid \mathcal{C} \subset F \text{ is a maximal pure chain} \right\}.$$

Note that for infinite forests the structure $s(F)$ may or may not be finite. For example, for the factory density it is finite as we have already seen above.

We have seen in Lemma 11 that the nodes of a continuous clustering are $\perp$-separated elements of $\bar{\mathcal{A}}$. Consequently, it only makes sense to compare continuous clustering with the structure of a level set forest, if this forest shares this property. This is ensured in the following definition.

Definition 22 *Let $f\colon \Omega \to [0,\infty]$ be a measurable function and $(\mathcal{A}, \mathcal{Q}, \perp)$ be a stable clustering base. Then f is of $(\mathcal{A}, \mathcal{Q}, \perp)$-**type** iff there is a dense subset $\Lambda \subset [0, \sup f)$ such that for all $\lambda \in \Lambda$ the level set $\{f > \lambda\}$ is a finite union of pairwise $\perp$-separated events $B_1(\lambda), \ldots, B_{k(\lambda)}(\lambda) \in \bar{\mathcal{A}}$. If this is the case the **level set $\perp$-forest** is given by*

$$F_{f,\Lambda} := \{B_i(\lambda) \mid i \leq k(\lambda) \text{ and } \lambda \in \Lambda\}.$$

Note that for given f and Λ the forest $F_{f,\Lambda}$ is indeed well-defined since $\perp$ is an $\bar{\mathcal{A}}$-separation relation by Lemma 11 and therefore the decomposition of $\{f > \lambda\}$ into the sets $B_1(\lambda), \ldots, B_{k(\lambda)}(\lambda) \in \bar{\mathcal{A}}$ is unique by Lemma 30.

With the help of these preparations we can now formulate the main result of this subsection, which compares continuous clusterings with the structure of level set $\perp$-forests:

Theorem 23 *Let $\mu \in \mathcal{M}_\Omega$, $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ the stable clustering based described in Proposition 3, and $P \in \mathcal{M}_\Omega$ such that $\mathcal{A}$ is P-subadditive. Assume that P has a μ-density f that is of $(\mathcal{A}, \mathcal{Q}, \perp)$-type with a dense subset Λ such that $s(F_{f,\Lambda})$ is finite and for all $\lambda \in \Lambda$ and all $i < j \leq k(\lambda)$ we have $\overline{B}_i(\lambda) \perp \overline{B}_j(\lambda)$. Then we have $P \in \bar{\mathcal{S}}(\mathcal{A})$ and*

$$c(P) =_\mu s(F_{f,\Lambda}).$$

On the other hand, it is not difficult to show that if $P \in \bar{\mathcal{S}}(\mathcal{A})$ then P has a density of $(\mathcal{A}, \mathcal{Q}, \perp)$-type. We do not know though whether there has to a density of $(\mathcal{A}, \mathcal{Q}, \perp)$-type for that even the closure of siblings are separated.

If $\operatorname{supp}\mu \neq \Omega$ one might think that this is not true since on the complement of the support anything goes. To be more precise—if μ is not inner regular and hence no support is defined—assume there is an open set $O \subset \Omega$ with $\mu(O) = 0$. This then means that there is no base set $A \subset O$, because base sets are support sets. Hence anything that would happen on O is determined by what happens in $\operatorname{supp} P$!

In the literature density based clustering is only considered for continuous densities since they may serve as a canonical version of the density. The following result investigates such densities.

Proposition 24 *For a compact $\Omega \subset \mathbb{R}^d$ and a measure $\mu \in \mathcal{M}_\Omega$ we consider the stable clustering base $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp_\emptyset)$. We assume that all open, connected sets are contained in $\bar{\mathcal{A}}$ and that $P \in \mathcal{M}_\Omega$ is a finite measure such that $\mathcal{A}$ is P-subadditive. If P has a continuous density f that has only finitely many local maxima $x_1^*, \ldots, x_k^*$ then $P \in \mathcal{P}_\mathcal{A}$ and there a bijection $\psi\colon \{x_1^*, \ldots, x_k^*\} \to \min c(P)$ such that*

$$x_i^* \in \psi(x_i^*)\,.$$

In this case $c(P) =_\mu \left\{ B_i\lambda \mid i \leq k(\lambda) \text{ and } \lambda \in \Lambda_0 \right\}$ where $\Lambda_0 = \left\{ 0 = \lambda_0 < \ldots < \lambda_m < \sup f \right\}$ is the finite set of levels at which the splits occur.

4. Examples

After having given the skeleton of this theory we now give more examples of how to use it. This should as well motivate some of the design decisions. It will also become clear in what way the choice of a clustering base $(\mathcal{A}, \mathcal{Q}, \perp)$ influences the clustering.

4.1 Base Sets and Separation Relations

In this subsection we present several examples of clustering bases. Our first three examples consider different separation relations.

Example 1 (Separation relations) *The following define stable $\mathcal{A}$-separation relations:*

(a) ***Disjointness****: If $\mathcal{A} \subset \mathcal{B}$ is a collection of non-empty, closed, and topologically connected sets then*

$$B \perp_\emptyset B' \iff B \cap B' = \emptyset.$$

(b) ***τ-separation****: Let (Ω, d) be a metric space, $\tau > 0$, and $\mathcal{A} \subset \mathcal{B}$ be a collection of non-empty, closed, and τ-connected sets then*

$$B \perp_\tau B' :\iff d(B, B') \geq \tau.$$

(c) ***Linear separation****: Let H be a Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$ and $\Omega \subset H$. Then non-empty events $A, B \subset \Omega$ are* ***linearly separated****, $A \perp_\ell B$, iff $A \perp_\emptyset B$ and*

$$\exists v \in H \setminus \{0\}, \alpha \in \mathbb{R}\, \forall a \in A, b \in B\colon \langle a \mid v \rangle \leq \alpha \text{ and } \langle b \mid v \rangle \geq \alpha.$$

The latter means there is an affine hyperplane $U \subset \Omega$ such that A and B are on different sides. Then $\perp_\ell$ is a $\mathcal{A}$ separation relation if no base set $A \in \mathcal{A}$ can be written as a finite union of pairwise $\perp_\ell$-disjoint closed sets. It is stable if H is finite-dimensional.

Our next goal is to present some examples of base set collections $\mathcal{A}$. Since these describe the sets we need to agree upon that their can only be trivially clustered, smaller collections $\mathcal{A}$ are generally preferred. Let μ be the Lebesgue measure on $\mathbb{R}^d$. To define possible collections $\mathcal{A}$ we will consider the following building blocks in $\mathbb{R}^d$:

$$\begin{aligned}\mathcal{C}_{\text{Dyad}} &:= \{\text{axis-parallel boxes with dyadic coordinates}\},\\ \mathcal{C}_p &:= \{\text{closed } \ell_p^d\text{-balls }\}, \quad p \in [1,\infty],\\ \mathcal{C}_{\text{Conv}} &:= \{\text{convex and compact } \mu\text{-support sets}\}.\end{aligned}$$

$\mathcal{C}_{\text{Dyad}}$ corresponds to the cells of a histogram whereas $\mathcal{C}_p$ has connections to moving-window density estimation. When combined with $\perp_\emptyset$ or $\perp_\tau$ and base measures of the form (14) these collections may already serve as clustering bases. However, $\bar{\mathcal{C}}_\bullet$ and $\bar{\mathcal{S}}_{\mathcal{C}}$ are not very rich since monotone increasing sequences in $\mathcal{C}_\bullet$ converge to sets of the same shape, and hence the sets in $\bar{\mathcal{C}}_\bullet$ have the same shape constraint as those in $\mathcal{C}_\bullet$. As a result the sets of measures $\bar{\mathcal{S}}_{\mathcal{C}_\bullet}$ for which we can determine the unique continuous clustering are rather small. However, more interesting collections can be obtained by considering finite, connected unions built of such sets. To describe such unions in general we need the following definition.

Definition 25 *Let $\perp\!\!\!\perp$ be a relation on $\mathcal{B}$, $\between$ be its negation, and $\mathcal{C} \subset \mathcal{B}$ be a class of non-empty events. The $\perp\!\!\!\perp$-**intersection graph** on $\mathcal{C}$, $\mathcal{G}_{\perp\!\!\!\perp}(\mathcal{C})$, has $\mathcal{C}$ as nodes and there is an edge between $A, B \in \mathcal{C}$ iff $A \between B$. We define:*

$$\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C}) := \{\, C_1 \cup \ldots \cup C_k \mid C_1, \ldots, C_k \in \mathcal{C} \text{ and the graph } \mathcal{G}_{\perp\!\!\!\perp}(\{C_1, \ldots, C_k\}) \text{ is connected}\}.$$

Obviously any separation relation can be used. But one can also consider weaker relations like $\perp\!\!\!\perp_P$, or e.g. $A \perp\!\!\!\perp A'$ if $A \cap A'$ has empty interior, or if it contains no ball of size τ. Such examples yield smaller $\mathcal{A}$ and indeed in these cases $\bar{\mathcal{S}}$ is much smaller.

The following example provides stable clustering bases.

Example 2 (Clustering bases) *The following examples are $\between_\emptyset$-additive:*

$$\begin{aligned}\mathcal{A}_{Dyad} := \mathbb{C}_{\perp_\emptyset}(\mathcal{C}_{Dyad}) &= \{\textit{finite connected unions of boxes with dyadic coordinates}\},\\ \mathcal{A}_p := \mathbb{C}_{\perp_\emptyset}(\mathcal{C}_p) &= \{\textit{finite connected unions of closed } L^p\textit{-balls}\},\\ \mathcal{A}_{Conv} := \mathbb{C}_{\perp_\emptyset}(\mathcal{C}_{Conv}) &= \{\textit{finite connected unions of convex } \mu\textit{-support sets}\}.\end{aligned}$$

Then $\mathcal{A}_{Dyad}, \mathcal{A}_p, \mathcal{A}_{Conv} \subset \mathcal{K}(\mu)$. Furthermore the following examples are $\between_\tau$-additive:

$$\mathcal{A}^\tau_{Dyad} := \mathbb{C}_{\perp_\tau}(\mathcal{C}_{Dyad}), \qquad \mathcal{A}^\tau_p := \mathbb{C}_{\perp_\tau}(\mathcal{C}_p), \qquad \mathcal{A}^\tau_{Conv} := \mathbb{C}_{\perp_\tau}(\mathcal{C}_{Conv}).$$

This leads to the following examples of stable clustering bases:

$$\begin{aligned}&(\mathcal{A}_{Dyad}, \mathcal{Q}^{\mu,\mathcal{A}_{Dyad}}, \perp_\emptyset), \quad (\mathcal{A}_p, \mathcal{Q}^{\mu,\mathcal{A}_p}, \perp_\emptyset), \quad (\mathcal{A}_{Conv}, \mathcal{Q}^{\mu,\mathcal{A}_{Conv}}, \perp_\emptyset),\\ &(\mathcal{A}^\tau_{Dyad}, \mathcal{Q}^{\mu,\mathcal{A}^\tau_{Dyad}}, \perp_\tau), \quad (\mathcal{A}^\tau_p, \mathcal{Q}^{\mu,\mathcal{A}^\tau_p}, \perp_\tau), \quad (\mathcal{A}^\tau_{Conv}, \mathcal{Q}^{\mu,\mathcal{A}^\tau_{Conv}}, \perp_\tau),\\ &(\mathcal{A}_{Dyad}, \mathcal{Q}^{\mu,\mathcal{A}_{Dyad}}, \perp_\tau), \quad (\mathcal{A}_p, \mathcal{Q}^{\mu,\mathcal{A}_p}, \perp_\tau), \quad (\mathcal{A}_{Conv}, \mathcal{Q}^{\mu,\mathcal{A}_{Conv}}, \perp_\tau).\end{aligned}$$

Figure 10: Some examples of sets in $\mathcal{A}_{\text{Box}}$, $\mathcal{A}_{\text{Conv}}$ and their closure.

The first row is the most common case, using connected sets and their natural separation relation. The second row is the τ-connected case. The third row shows how the fine tuning can be handled: We consider connected base sets, but siblings need to be τ-separated, hence e.g. saddle points cannot be approximated.

The larger the extended class $\bar{\mathcal{A}}$ is, the more measures we can cluster. The following proposition provides a sufficient condition for $\bar{\mathcal{A}}$ being rich.

Proposition 26 *Assume all $A \in \mathcal{A}$ are path-connected. Then all $B \in \bar{\mathcal{A}}$ are path-connected. Furthermore assume that $\mathcal{A}$ is intersection-additive and that it contains a countable neighbourhood base. Then $\bar{\mathcal{A}}$ contains all open, path-connected sets.*

One can show that the first statement also holds for topological connectedness. Furthermore note that $\mathcal{C}_{\text{Dyad}}$ is a countable neighbourhood base, and therefore $\mathcal{A}_{\text{Dyad}}$, $\mathcal{A}_p$, and $\mathcal{A}_{\text{Conv}}$ fulfill the conditions of Proposition 26.

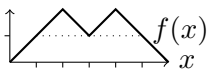
4.2 Clustering of Densities

Following the manual to cluster densities given in Theorem 23 by decomposing the density level sets into $\perp$-disjoint components, one first needs to understand the $\perp$-disjoint components of general events. In this subsection we investigate such decompositions and the resulting clusterings. We assume μ to be the Lebesgue measure on some suitable $\Omega \subset \mathbb{R}^d$ and let the base measures be the ones considered in Proposition 3. For visualization purposes we further restrict our considerations to the one- and two-dimensional case, only.

4.2.1 Dimension $d = 1$

In the one-dimensional case, in which Ω is an interval, the examples $\mathcal{A}_p = \mathcal{A}_{\text{Conv}}$ simply consist of compact intervals, and their monotone closures consist of all intervals. To understand the resulting clusters let us first consider the **twin peaks** density:

$$f(x) := \tfrac{1}{3} - \min\left\{ |x - \tfrac{1}{3}|, |x - \tfrac{2}{3}| \right\}.$$



Clearly, this gives the following $\perp_\emptyset$-decomposition of the level sets:

$$H_f(\lambda) = (\lambda, 1-\lambda) \text{ for } \lambda < \tfrac{1}{6}, \qquad H_f(\lambda) = (\lambda, \tfrac{1}{2} - \lambda) \overset{\perp_\emptyset}{\cup} (\tfrac{1}{2} + \lambda, \lambda) \text{ for } \tfrac{1}{6} \leq \lambda < \tfrac{1}{3}$$

and hence the $\perp_\emptyset$-clustering forest is $\left\{ (0,1), (\tfrac{1}{6}, \tfrac{1}{2}), (\tfrac{1}{2}, \tfrac{5}{6}) \right\}$. Since, none of the boundary points can be reached, any isomonotone, adapted sequence yields this result. However, the clustering changes, if the separation relation $\perp_\tau$ is considered. We obtain

$$H_f(\lambda) = (\lambda, 1-\lambda), \text{ for } \lambda < \tfrac{1}{6} + \tfrac{\tau}{2}, \quad H_f(\lambda) = (\lambda, \tfrac{1}{2} - \lambda) \overset{\perp_\tau}{\cup} (\tfrac{1}{2} + \lambda, \lambda), \text{ for } \tfrac{1}{6} + \tfrac{\tau}{2} \leq \lambda < \tfrac{1}{3}$$

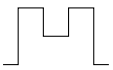
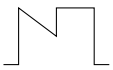
Name	Merlon	Camel	M	Factory
Density				
$(\mathcal{A}_p, \perp_\emptyset)$				
$(\mathcal{A}_p, \perp_\tau)$ with τ small				
$(\mathcal{A}_p, \perp_\tau)$ with τ large				

Table 1: Examples of clustering in dimension $d = 1$ using $\mathcal{A}_p$ and three separation relations.

if $\tau \in (0, \frac{1}{3})$ and the resulting $\perp_\tau$-clustering is $\{ (0,1), (\frac{1}{6}+\frac{\tau}{2}, \frac{1}{2}-\frac{\tau}{2}), (\frac{1}{2}+\frac{\tau}{2}, \frac{5}{6}-\frac{\tau}{2}) \}$. Finally, if $\tau \geq \frac{1}{3}$ then all level sets are τ-connected and the forest is simply $\{(0,1)\}$. In Table 1 more examples of clustering of densities can be found.

4.2.2 Dimension $d = 2$

Our goal in this subsection is to understand the $\perp$-separated decomposition of closed events. We further present the resulting clusterings for some densities that are indicator functions and illustrate clusterings for continuous densities having a saddle point.

Let us begin by assuming that P has a Lebesgue density of the form 1_B, where B is some μ-support set. Then one can show, see Lemma 50 for details, that adapted, isomonotone sequences (F_n) of forests $F_n \uparrow B$ are of the form $F_n = \{A_1^n, \ldots, A_k^n\}$, where the elements of each forest F_n are mutually disjoint and can be ordered in such a way that $A_i^1 \subset A_i^2 \subset \ldots$. The limit forest F_∞ then consists of the k pairwise $\perp$-separated sets:

$$B_i := \bigcup_{n \geq 1} A_i^n \,,$$

and there is a μ-null set $N \in \mathcal{B}$ with

$$B = B_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} B_k \overset{\perp}{\cup} N. \tag{21}$$

Let us now consider the base sets $\mathcal{A}_p$ in Example 2. By Proposition 26 we know that $\bar{\mathcal{A}}_p$ contains all open, path-connected sets and therefore all open L^q-balls. Moreover, all closed L^q-balls B are μ-support sets with $\mu(\partial B) = 0$. Our initial consideration shows that 1_B can be approximated by an adapted, isomonotone sequence (F_n) of forests of the form $F_n = \{A^n\}$ with $A^n \in \mathcal{A}_p$. However, depending on p and q the μ-null set N in (21) may differ.

Now that we have an understanding of $\bar{\mathcal{A}}_p$ and adapted, isomonotone approximations we can investigate some more interesting cases and appreciate the influence of the choice of $\mathcal{A}$ on the outcome of the clustering in the following example.

Example 3 (Clustering of indicators) *We consider 6 examples of μ-support sets $B \in \mathbb{R}^2$. The first 4 have two parts that only intersect at one point, the second to last has two*

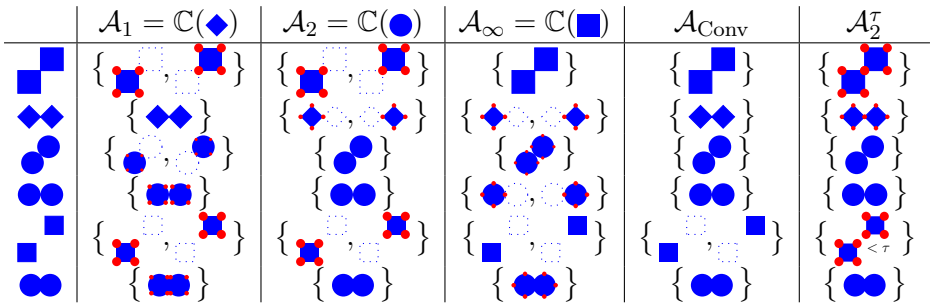


Table 2: Clusterings of indicators.

topological components, and the last one is connected in a fat way. By natural approximations we get the clusterings of Table 2. The red dots indicate points which never are achieved by any approximation. Observe how the geometry encoded in $\mathcal{A}$ shapes the clustering. Since $\mathcal{A}_{Conv}$ and $\mathcal{A}_2$ are invariant under rotation, they yield the same structure of clustering for rotated sets. The classes $\mathcal{A}_1$ and $\mathcal{A}_\infty$ on the other hand are not rotation-invariant and therefore the clustering depends on the orientation of B.

After having familiarized ourselves with the clustering of indicator functions we finally consider a continuous density that has a saddle point.

Example 4 *On $\Omega := [-1,1]^2$ consider the density $f : \Omega \to [0,2]$ given by $f(x,y) := x \cdot y + 1$. Then we have the following $\bot_\emptyset$-decomposition of the level sets $H_f(\lambda)$ of f:*

$$H_f(\lambda) = \begin{cases} \{(x,y) : xy > \lambda - 1\} & \text{if } \lambda \in [0,1), \\ [-1,0)^2 \,\dot\cup\, (0,1]^2 & \text{if } \lambda = 1, \\ \{(x,y) : x < 0 \text{ and } xy > \lambda - 1\} \,\dot\cup\, \{(x,y) : x > 0 \text{ and } xy > \lambda - 1\} & \text{if } \lambda \in (1,2). \end{cases}$$

For $(\mathcal{A}_p, \mathcal{Q}^{\mu,\mathcal{A}_p}, \bot_\emptyset)$ the clustering forest is therefore given by:

$$\{\, [-1,1]^2, [-1,0)^2, (0,1]^2 \,\} = \{ \blacksquare, \blacksquare, \blacksquare \}.$$

Moreover, for $(\mathcal{A}_2^\tau, \mathcal{Q}^{\mu,\mathcal{A}_2^\tau}, \bot_\tau)$ the clustering forest looks like $\{ \blacksquare, \blacksquare, \blacksquare \}$.

4.3 Hausdorff Measures

So far we have only considered clusterings of Lebesgue absolutely continuous distributions. In this subsection we provide some examples indicating that the developed theory goes far beyond this standard example. At first, lower dimensional base sets and their resulting clusterings are investigated. Afterwards we discuss collections of base sets of different dimensions and provide clusterings for some measures that are not absolutely continuous to any Hausdorff measure. For the sake of simplicity we will restrict our considerations to $\bot_\emptyset$-clusterings, but generalizations along the lines of the previous subsections are straightforward.

4.3.1 Lower Dimensional Base Sets

Let us begin by recalling that the s-dimensional Hausdorff-measure on $\mathcal{B}$ is defined by

$$\mathcal{H}^s(B) := \lim_{\varepsilon \to 0} \inf \Big\{ \sum_{i=1}^{\infty} (\operatorname{diam}(B_i))^s \mid B \subset \bigcup_i B_i \text{ and } \forall i \in \mathbb{N}\colon \operatorname{diam}(B_i) \leq \varepsilon \Big\}.$$

Moreover, the Hausdorff-dimension of a $B \in \mathcal{B}$ is the value $s \in [0, d]$ at which $s \mapsto \mathcal{H}^s(B)$ jumps from ∞ to 0. If B has Hausdorff-dimension s, then $\mathcal{H}^s(B)$ can be either zero, finite, or infinite. Hausdorff-measures are inner regular (Federer, 1969, Cor. 2.10.23) and $\mathcal{H}^d$ equals the Lebesgue-measure up to a normalization factor. For a reference on Hausdorff-dimensions and -measures we refer to Falconer (1993) and Federer (1969). Recall that given a Borel set $C \subset \mathbb{R}^s$ a map $\varphi\colon C \to \Omega$ is **bi-Lipschitz** iff there are constants $0 < c_1, c_2 < \infty$ s.t.

$$c_1 d(x, y) \leq d(\varphi(x), \varphi(y)) \leq c_2 d(x, y).$$

Lemma 27 *If C is a Lebesgue-support set in $\mathbb{R}^s$ and $\varphi\colon C \to \Omega$ is bi-Lipschitz then $C' := \varphi(C)$ has Hausdorff-dimension s and it is an $\mathcal{H}^s$-support set in Ω.*

Motivated by Lemma 27, consider the following collection of s-dimensional base sets in Ω:

$$\mathcal{C}_{p,s} := \big\{ \varphi(C) \subset \Omega \mid C \text{ is the closed unit } p\text{-ball in } \mathbb{R}^s \text{ and } \varphi\colon C \to \Omega \text{ is bi-Lipschitz} \big\}.$$

Using the notation of Definition 25 and Proposition 3 we further write

$$\mathcal{A}_{p,s} := \mathbb{C}_{\perp_\emptyset}(\mathcal{C}_{p,s}) \qquad \text{and} \qquad \mathcal{Q}^{p,s} := \mathcal{Q}^{\mathcal{H}^s, \mathcal{A}_{p,s}}.$$

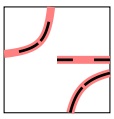
By $\mathcal{A}_0 := \big\{ \{x\} \mid x \in \Omega \big\}$ we denote the singletons and $\mathcal{Q}_0$ the collection of Dirac measures. Since continuous mappings of connected sets are connected, $(\mathcal{A}_{p,s}, \mathcal{Q}^{p,s}, \perp_\emptyset)$ is a stable $\perp_\emptyset$-additive clustering base. Remark that we take the union after embedding into $\mathbb{R}^d$ and therefore also crossings do happen, e.g. the cross $[-1,1] \times \{0\} \cup \{0\} \times [-1,1] \in \mathcal{A}_{p,1}$. Another possibility would be to embed $\mathcal{A}_p$ via a set of transformations into $\mathbb{R}^d$. Finally we confine the examples here only to integer Hausdorff-dimensions—it would be interesting though to consider e.g. the Cantor set or the Sierpinski triangle. The following example presents a resulting clustering of an $\mathcal{H}^1$-absolutely continuous measure on $\mathbb{R}^2$.

Example 5 (Measures supported on curves in the plane)
On $\Omega := [-1,1]^2$ consider the measure $P_1 := f\, d\mathcal{H}^1$ whose density is given by

$$f(x,y) := \begin{cases} f_{Merlon}(x) & \text{if } x \geq 0 \text{ and } y = 0, \\ f_{Camel}(t) & \text{if } x = -3^{2t-2} \text{ and } y = 3^{-2t}, \\ f_M(t) & \text{if } x = 2^{2t-2} \text{ and } y = -2^{-2t}. \end{cases}$$

Here the densities and clusterings for the Merlon, the Camel and the M can be seen in Table 1. So for $(\mathcal{A}_{p,1}, \mathcal{Q}^{p,1}, \perp_\emptyset)$ with any fixed $p \geq 1$ the clustering forest of P_1 is given by:

$$c(P_1) = \left\{ \begin{matrix} [0,1] \times \{0\}, [0, \frac{1}{3}] \times \{0\}, [\frac{2}{3}, 1] \times \{0\}, \\ g_1\big((0,1)\big), g_1\big((0.2, 0.5)\big), g_1\big((0.5, 0.8)\big), \\ g_2\big([0,1]\big), g_2\big([0, 0.5)\big), g_2\big((0.5, 1]\big) \end{matrix} \right\}$$



where $g_i\colon [0,1] \to \Omega$ are given by $g_1(t) = (-3^{2t-2}, 3^{-2t})$ and $g_2(t) = (2^{2t-2}, -2^{-2t})$.

4.3.2 Heterogeneous Hausdorff-Dimensions

In this subsection we consider measures that can be decomposed into measures that are absolutely continuous with respect to Hausdorff measures of different dimensions. To this end, we write $\mu \prec \mu'$ for two measures μ and μ' on $\mathcal{B}$, iff for all $B \in \mathcal{B}$ with $B \subset \operatorname{supp}\mu \cap \operatorname{supp}\mu'$ we have

$$\mu(B) < \infty \implies \mu'(B) = 0.$$

For $\mathcal{Q}, \mathcal{Q}' \subset \mathcal{M}_\Omega$ we further write $\mathcal{Q} \prec \mathcal{Q}'$ if $\mu \prec \mu'$ for all $\mu \in \mathcal{Q}$ and $\mu' \in \mathcal{Q}'$. Clearly, the relation $\prec$ is transitive. Moreover, we have $\mathcal{H}^s \prec \mathcal{H}^t$ whenever $s < t$. The next proposition shows that clustering bases whose base measures dominate each other in the sense of $\prec$ can be merged.

Proposition 28 *Let $(\mathcal{A}^1, \mathcal{Q}^1, \perp), \ldots, (\mathcal{A}^m, \mathcal{Q}^m, \perp)$ be stable clustering bases sharing the same separation relation $\perp$ and assume $\mathcal{Q}^1 \prec \cdots \prec \mathcal{Q}^m$. We define*

$$\mathcal{A} := \bigcup_i \mathcal{A}^i \qquad \text{and} \qquad \mathcal{Q} := \bigcup_i \mathcal{Q}^i.$$

Then $(\mathcal{A}, \mathcal{Q}, \perp)$ is a stable clustering base.

Proposition 28 shows that the $\perp_\emptyset$-additive, stable bases $(\mathcal{A}_{p,s}, \mathcal{Q}^{p,s}, \perp_\emptyset)$ on $\mathbb{R}^d$ can be merged. Unfortunately, however, its union is no longer $\perp_\emptyset$-additive, and therefore we need to investigate P-subadditivity in order to describe distributions for which our theory provides a clustering. This is done in the next proposition.

Proposition 29 *Let $(\mathcal{A}^1, \mathcal{Q}^1, \perp)$ and $(\mathcal{A}^2, \mathcal{Q}^2, \perp)$ be clustering bases with $\mathcal{Q}^1 \prec \mathcal{Q}^2$ and P_1 and P_2 be finite measures with $P_1 \prec \mathcal{A}^2$ and $\mathcal{A}^1 \prec P_2$. Furthermore, assume that $\mathcal{A}^i$ is P_i-subadditive for both $i = 1, 2$ and let $P := P_1 + P_2$. Then we have*

(a) For $i = 1, 2$ and all base measures $\mathfrak{a} \in \mathcal{Q}^i_P$ we have $\mathfrak{a} \le P_i$,

(b) If for all base measures $\mathfrak{a} \in \mathcal{Q}^2_{P_2}$ and $\operatorname{supp} P_1 \not\propto \operatorname{supp}\mathfrak{a}$ there exists a base measure $\tilde{\mathfrak{a}} \in \mathcal{Q}^2_{P_2}(\operatorname{supp} P_1)$ with $\mathfrak{a} \le \tilde{\mathfrak{a}}$ then $\mathcal{A}^1 \cup \mathcal{A}^2$ is P-subadditive.

To illustrate condition *(b)* consider clustering bases $(\mathcal{A}_{p,s}, \mathcal{Q}^{p,s}, \perp_\emptyset)$ and $(\mathcal{A}_{p,t}, \mathcal{Q}^{p,t}, \perp_\emptyset)$ for some $s < t$. The condition specifies that any such base measure $\mathfrak{a}$ intersecting $\operatorname{supp} P_1$ can be majorized by one which supports $\operatorname{supp} P_1$. Then all parts of $\operatorname{supp} P_1$ intersecting at least one component of $\operatorname{supp} P_2$ have to be on the same niveau line of P_2. Note that this is trivially satisfied if the $\operatorname{supp} P_1 \cap \operatorname{supp} P_1 = \emptyset$. Recall that mixtures of the latter form have already been clustered in Rinaldo and Wasserman (2010) by a kernel smoothing approach. Clearly, our axiomatic approach makes it possible to define clusterings for significantly more involved distributions as the following two examples demonstrate.

Example 6 (Mixture of atoms and Lebesgue measure)
Consider $\Omega = \mathbb{R}$. Let $(\mathcal{A}_0, \mathcal{Q}_0, \perp_\emptyset)$ be the singletons with Dirac measures and consider for

any fixed $p \geq 1$ the clustering base $(\mathcal{A}_{p,s}, \mathcal{Q}_{p,s}, \perp_\emptyset)$. Both are $\infty_\emptyset$-additive and stable and we have $\mathcal{Q}_0 \prec \mathcal{Q}_{p,1}$. Now consider the measures

$$P_0 := \delta_0 + 2\delta_1 + \delta_2 \qquad \text{and} \qquad P_1(dx) := \sin^2(\tfrac{2x}{\pi})\, \mathcal{H}_1(dx).$$

Then the assumptions of Proposition 29 are satisfied and the clustering of $P := P_0 + P_1$ is given by

$$c(P) = c(P_0) \cup c(P_1) = \left\{ \{0\}, (0, \tfrac{1}{2}), (\tfrac{1}{2}, 1), \{1\}, \{2\} \right\}.$$

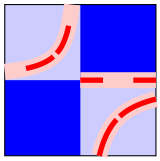
Our last example combines Examples 4 and 5.

Example 7 (Mixtures in dimension 2) *Consider $\Omega := [-1,1]^2$ and the densities f_1 and f_2 introduced in Examples 5 and 4, respectively. Furthermore, consider the measures*

$$P_2 := f_2\, d\mathcal{H}^2, \qquad\qquad P_1 := f_1\, d\mathcal{H}^1$$

and the clustering bases $(\mathcal{A}_{p,1}, \mathcal{Q}^{p,1}, \perp_\emptyset)$ and $(\mathcal{A}_{p',2}, \mathcal{Q}^{p',2}, \perp_\emptyset)$ for some fixed $p, p' \geq 1$. As above $\mathcal{Q}^{p,1} \prec \mathcal{Q}^{p',2}$. And by Proposition 29 the clustering forest of $P = P_1 + P_2$ is given by

$$c_1(P_1) \cup c_2(P_2) = \left\{ \begin{array}{c} [0,1] \times \{0\}, [0, \frac{1}{3}] \times \{0\}, [\frac{2}{3}, 1] \times \{0\}, \\ g_1\big((0,1)\big), g_1\big((0.2, 0.5)\big), g_1\big((0.5, 0.8)\big), \\ g_2\big([0,1]\big), g_2\big([0, 0.5)\big), g_2\big((0.5, 1]\big), \\ [-1,1]^2, [-1,0)^2, (0,1]^2 \end{array} \right\}$$



where $g_i \colon [0,1] \to \Omega$ are given by $g_1(t) = (-3^{2t-2}, 3^{-2t})$ and $g_2(t) = (2^{2t-2}, -2^{-2t})$. Observe that g_1 and g_2 lie on niveau lines of f_2.

5. Proofs

5.1 Proofs for Section 2

We begin with some simple properties of separation relations.

Lemma 30 *Let $\perp$ be an $\mathcal{A}$-separation relation. Then the following statements are true:*

(a) For all $B, B' \in \mathcal{B}$ with $B \perp B'$ we have $B \cap B' = \emptyset$.

(b) Suppose that $\perp$ is stable and $(A_i)_{i \geq 1} \subset \mathcal{A}$ is increasing. For $A := \bigcup_n A_n$ and all $B \in \mathcal{B}$ we then have

$$A_n \perp B \quad \text{for all } n \geq 1 \iff A \perp B$$

(c) Let $A \in \mathcal{A}$ and $B_1, \ldots, B_k \in \mathcal{B}$ be closed. Then:

$$A \subset B_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} B_k \implies \exists! i \leq k \colon A \subset B_i$$

(d) For all $A_1, \ldots, A_k \in \mathcal{A}$ and all $A'_1, \ldots, A'_{k'} \in \mathcal{A}$, we have

$$A_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} A_k = A'_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} A'_{k'} \implies \{A_1, \ldots, A_k\} = \{A'_1, \ldots, A'_{k'}\}.$$

Proof of Lemma 30: *(a).* Let us write $B_0 := B \cap B'$. Monotonicity and $B \perp B'$ implies $B_0 \perp B'$ and thus $B' \perp B_0$ by symmetry. Another application of the monotonicity gives $B_0 \perp B_0$ and the reflexivity thus shows $B \cap B' = B_0 = \emptyset$.
(b). "$\Rightarrow$" is stability and "$\Leftarrow$" follows from monotonicity.
(c). Existence of such an i is $\mathcal{A}$-connectedness. Now assume that there is an $j \neq i$ with $A \subset B_j$. Then $\emptyset \neq A \subset B_i \cap B_j$ contradicting $B_i \perp B_j$ by (a).
(d). We write $F := \{A_1, \ldots, A_k\}$ and $F' := \{A'_1, \ldots, A'_{k'}\}$. By (c) we find an injection $I \colon F \to F'$ such that $A \subset I(A)$ and hence $k \leq k'$. Analogously, we find an injection $J \colon F' \to F$ such that $A \subset J(A)$, and we get $k = k'$. Consequently, I and J are bijections. Let us now fix an $A_i \in F$. For $A_j := J \circ I(A_i) \in F$ we then find $A_i \subset I(A_i) \subset J(I(A_i)) = A_j$. This implies $i = j$, since otherwise $A_i \subset A_j$ would contradict $A_i \perp A_j$ by (a). Therefore we find $A_i = I(A_i)$ and the bijectivity of I thus yields the assertion. ∎

Proof of Proposition 3: We first need to check that the support is defined for all restrictions $\mu_{|C} := \mu(\cdot \cap C)$ to sets $C \in \mathcal{B}$ that satisfy $0 < \mu(C) < \infty$. To this end, we check that $\mu_{|C}$ is inner regular: If Ω is a Radon space then there is nothing to prove since $\mu_{|C}$ is a finite measure. If Ω is not a Radon space, then the definition of $\mathcal{M}_\Omega^\infty$ guarantees that μ is inner regular and hence $\mu_{|C}$ is inner regular by Lemma 51.

Let us now verify that $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ is a (stable) clustering base. To this end, we first observe that each $Q_A \in \mathcal{Q}^{\mu,\mathcal{A}}$ is a probability measure by construction and since we have already seen that $\mu_{|C}$ is inner regular for all $C \in \mathcal{K}(\mu)$ we conclude that $\mathcal{Q}^{\mu,\mathcal{A}} \subset \mathcal{M}$. Moreover, fittedness follows from $\mathcal{A} \subset \mathcal{K}(\mu)$. For flatness let $A, A' \in \mathcal{A}$ with $A \subset A'$ and $Q_{A'}(A) \neq 0$. Then for all $B \in \mathcal{B}$ we have

$$Q_A(B) = \frac{\mu(B \cap A)}{\mu(A)} = \frac{\mu(B \cap A \cap A')}{\mu(A \mid A') \cdot \mu(A')} = \frac{\mu(B \cap A \mid A')}{\mu(A \mid A')} = \frac{Q_{A'}(B \cap A)}{Q_{A'}(A)}. \qquad \blacksquare$$

Proof of Lemma 7: Let $Q = \sum_{A \in F} \alpha_A Q_A$ and $Q = \sum_{A' \in F'} \alpha'_{A'} Q_{A'}$ be two representations of $Q \in \mathcal{Q}$. By part (d) of Lemma 51 we then obtain

$$\operatorname{supp} Q = \operatorname{supp}\left(\sum_{A \in F} \alpha_A Q_A\right) = \bigcup_{A \in F} \operatorname{supp} Q_A = \bigcup_{A \in F} A\,, = \mathbb{G}F$$

and since we analogously find $\operatorname{supp} Q = \mathbb{G}F'$, we conclude that $\mathbb{G}F = \mathbb{G}F'$. The latter together with Lemma 30 gives $\max F = \max F'$. To show that $\alpha_A = \alpha'_A$ for all roots $A \in \max\ F = \max F'$, we pick a root $A \in \max\ F$ and assume that $\alpha_A < \alpha'_A$. Now, if A has no direct child, we set $B := A$. Otherwise we define $B := A \setminus (A_1 \cup \ldots \cup A_k)$, where the A_k are the direct children of A in F. Because of the definition of a direct child and part (d) of Lemma 30 we find $A_1 \cup \ldots \cup A_k \subsetneqq A$ in the second case. In both cases we conclude that B is non-empty and relatively open in $A = \operatorname{supp} Q_A$ and by Lemma 51 we obtain $Q_A(B) > 0$. Consequently, our assumption $\alpha_A < \alpha'_A$ yields $\alpha_A Q_A(B) < \alpha'_A Q_A(B) \leq Q(B)$. However, our construction also gives

$$Q(B) = \sum_{A'' \in F} \alpha_{A''} Q_{A''}(B) = \alpha_A Q_A(B) + \sum_{A'' \subsetneqq A} \alpha_{A''} Q_{A''}(B) + \sum_{A'' \perp A} \alpha_{A''} Q_{A''}(B) = \alpha_A Q_A(B)\,,$$

i.e. we have found a contradiction. Summing up, we already know that $\max F = \max F'$ and $\alpha_A = \alpha'_A$ for all $A \in \max F$. This yields

$$\sum_{A\in\max F} \alpha_A Q_A = \sum_{A'\in\max F'} \alpha'_{A'} Q_{A'} \,.$$

Eliminating the roots gives the forests $F_1 := F \setminus \max F$ and $F'_1 := F' \setminus \max F'$ and

$$Q_1 := \sum_{A\in F_1} \alpha_A Q_A = Q - \sum_{A\in\max F} \alpha_A Q_A = Q - \sum_{A'\in\max F'} \alpha'_{A'} Q_{A'} = \sum_{A'\in F'_1} \alpha'_{A'} Q_{A'} \,,$$

i.e. Q_1 has two representations based upon the reduced forests F_1 and F'_1. Applying the argument above recursively thus yields $F = F'$ and $\alpha'_A = \alpha'_A$ for all $A \in F$. ∎

Proof of Theorem 8: We first show that (16) defines an additive clustering. Since Axiom 1 is obviously satisfied, it suffices to check the two additivity axioms for $\mathcal{P} := \mathcal{S}(\mathcal{A})$. We begin by establishing DisjointAdditivity. To this end, we pick $Q_1, \ldots, Q_k \in \mathcal{S}(\mathcal{A})$ with representing $\perp$-forests F_i such that $\operatorname{supp} Q_i = \mathbb{G}F_i$ are mutually $\perp$-separated. For $A \in \max F_i$ and $A' \in \max F_j$ with $i \neq j$, we then have $A \perp A'$, and therefore

$$F := F_1 \cup \ldots \cup F_k$$

is the representing $\perp$-forest of $Q := Q_1 + \ldots + Q_k$. This gives $Q \in \mathcal{S}(\mathcal{A})$ and

$$c(Q) = s(F) = s(F_1) \cup \cdots \cup s(F_k) = c(Q_1) \cup \cdots \cup c(Q_k).$$

To check BaseAdditivity we fix a $Q \in \mathcal{S}(\mathcal{A})$ with representing $\perp$-forest F and a base measure $\mathfrak{a} = \alpha Q_A$ with $\operatorname{supp} Q \subset \operatorname{supp} \mathfrak{a}$. For all $A' \in F$ we then have $A' \subset \mathbb{G}F = \operatorname{supp} Q \subset A$ and therefore $F' := \{A\} \cup F$ is the representing $\perp$-forest of $\mathfrak{a} + Q$. This yields $\mathfrak{a} + Q \in \mathcal{S}(\mathcal{A})$ and

$$c(\mathfrak{a} + Q) = s(F') = s\big(\{A\} \cup F\big) = s\big(\operatorname{supp} \mathfrak{a} \cup c(Q)\big).$$

Let us now show that every additive $\mathcal{A}$-clustering $c : \mathcal{P} \to \mathcal{F}$ satisfies both $\mathcal{S}(\mathcal{A}) \subset \mathcal{P}$ and (16). To this end we pick a $Q \in \mathcal{S}(\mathcal{A})$ with representing forest F and show by induction over $|F| = n$ that both $Q \in \mathcal{P}$ and $c(Q) = s(F)$. Clearly, for $n = 1$ this immediately follows from Axiom 1. For the induction step we assume that for some $n \geq 2$ we have already proved $Q' \in \mathcal{P}$ and $c(Q') = s(F')$ for all $Q' \in \mathcal{S}(\mathcal{A})$ with representing forest F' of size $|F'| < n$.

Let us first consider the case in which F is a tree. Let A be its root and α_A be corresponding coefficient in the representation of Q. Then $Q' := Q - \alpha_A Q_A$ is a simple measure with representing forest $F' := F \setminus A$ and since $|F'| = n-1$ we know $Q' \in \mathcal{P}$ and $c(Q') = s(F')$ by the induction assumption. By the axiom of BaseAdditivity we conclude that

$$c(Q) = c(\alpha_A Q_A + Q') = s(\{A\} \cup c(Q')) = s(\{A\} \cup F') = s(F) \,,$$

where the last equality follows from the assumption that F is a tree with root A.

Now consider the case where F is a forest with $k \geq 2$ roots $A_1, \ldots, A_k$. For $i \leq k$ we define $Q_i := Q\big|_{\subset A_i}$. Then all Q_i are simple measures with representing forests $F_i := F\big|_{\subset A_i}$ and we have $Q = Q_1 + \cdots + Q_k$. Therefore, the induction assumption guarantees $Q_i \in \mathcal{P}$ and $c(Q_i) = s(F_i)$. Since $\operatorname{supp} Q_i = A_i$ and $A_i \perp A_j$ whenever $i \neq j$, the axiom of DisjointAdditivity then shows $Q \in \mathcal{P}$ and

$$c(Q) = c(Q_1) \cup \cdots \cup c(Q_k) = s(F_1) \cup \cdots \cup s(F_k) = s(F) \,.$$ ∎

5.2 Proofs for Section 3

Proof of Lemma 11: For the first assertion it suffices to check $\bar{\mathcal{A}}$-connectedness. To this end, we fix an $A \in \bar{\mathcal{A}}$ and closed sets $B_1, \ldots, B_k$ with $A \subset B_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} B_k$. Let $(A_n) \subset \mathcal{A}$ with $A_n \nearrow A$. For all $n \geq 1$ part (c) of Lemma 30 then gives exactly one $i(n)$ with $A_n \subset B_{i(n)}$. This uniqueness together with $A_n \subset A_{n+1}$ yields $i(1) = i(2) = \ldots$ and hence $A_n \subset B_{i(1)}$ for all n. We conclude that $A \subset B_{i(1)}$ by part (b) of Lemma 30.

For the second assertion we pick an isomonotone sequence $(F_n) \subset \mathcal{F}_{\mathcal{A}}$ and define $F_\infty := \lim_n s(F_n)$. Let us first show that F_∞ is a $\perp$-forest. To this end, we pick $A, A' \in F_\infty$. By the construction of F_∞ there then exist $A_1, A_1' \in s(F_1)$ such that for $A_n := \zeta_n(A_1)$ and $A_n' := \zeta_n(A_1')$ we have $A_n \nearrow A$ and $A_n' \nearrow A'$ Now, if $A_1 \perp A_1'$ then $A_n \perp A_n'$ and thus $A_m \perp A_n'$ for all m, n by isomonotonicity. Using the stability of $\perp$ twice we first obtain $A \perp A_n'$ for all n and then $A \perp A'$. If $A_1 \not\perp A_1'$, we may assume $A_1 \subset A_1'$ since $s(F_1)$ is a $\perp$-forest. Isomonotonicity implies $A_n \subset A_n' \subset A'$ for all n and hence $A \subset A'$. Finally, $s(F_n) \leq F_\infty$ is trivial. ∎

Proof of Proposition 16: We first show that $\mathcal{A}$ is P-subadditive if $\mathcal{A}$ is ∞_P-additive. To this end we fix $A, A' \in \mathcal{A}$ with $A \infty_P A'$. Since $\mathcal{A}$ is ∞_P-additive we find $B := A \cup A' \in \mathcal{A}$. This yields

$$Q_B(A) = \frac{\mu(A \cap B)}{\mu(B)} = \frac{\mu(A)}{\mu(B)} > 0$$

and analogously we obtain $Q_B(A') > 0$. For $\alpha Q_A, \alpha' Q_{A'} \leq P$ we can therefore assume that $\beta := \frac{\alpha}{Q_B(A)} < \frac{\alpha'}{Q_B(A')}$. Setting $\mathfrak{b} := \beta Q_B$ we now obtain by the flatness assumption

$$\alpha Q_A(\cdot) = \alpha \cdot \frac{Q_B(\cdot \cap A)}{Q_B(A)} = \mathfrak{b}(\cdot \cap A) \leq \mathfrak{b}(\cdot).$$

Now assume that $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$ is P-subadditive for all $P \ll \mu$. Let $A, A' \in \mathcal{A}$ with $A \infty_\mu A'$. Then we have $P := Q_A + Q_A' \ll \mu$ and $Q_A, Q_{A'} \leq P$. Since $\mathcal{A}$ is P-subadditive there is a base measure $\mathfrak{b} \leq P$ with $A \cup A' \subset \operatorname{supp} \mathfrak{b} \subset \operatorname{supp} P = A \cup A'$ by Lemma 51. Consequently we obtain $A \cup A' = \operatorname{supp} \mathfrak{b} \in \mathcal{A}$. ∎

Lemma 31 *Let $P \in \mathcal{M}$ and $(\mathcal{A}, \mathcal{Q}, \perp)$ be a P-subadditive clustering base. Then the kinship relation $\sim_P$ is a symmetric and transitive relation on $\{B \in \mathcal{B} \,|\, P(B) > 0\}$ and an equivalence relation on the set $\{A \in \mathcal{A} \,|\, \exists \alpha > 0$ such that $\alpha Q_A \leq P\}$. Finally, for all finite sequences $A_1, \ldots, A_k \in \mathcal{A}$ of sets that are pairwise kin below P there is $\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_k)$.*

Proof of Lemma 31: Symmetry is clear. Let $B_1 \sim_P B_2$ and $B_2 \sim_P B_3$ be events with $P(B_i) > 0$. Then there are base measures $\mathfrak{c} = \gamma Q_C \in \mathcal{Q}_P(B_1 \cup B_2)$ and $\mathfrak{c}' = \gamma' Q_{C'} \in \mathcal{Q}_P(B_2 \cup B_3)$ supporting them. This yields $B_2 \subset C \cap C'$ and thus $P(C \cap C') \geq P(B_2) > 0$. In other words, we have $C \infty_P C'$, and by subadditivity we conclude that there is a $\mathfrak{b} \in \mathcal{Q}_P(C \cup C')$. This gives $B_1 \cup B_3 \subset C \cup C' \subset \operatorname{supp} \mathfrak{b}$, and therefore $B_1 \sim_P B_3$ at $\mathfrak{b}$. To show reflexivity on the specified subset of $\mathcal{A}$, we fix an $A \in \mathcal{A}$ and an $\alpha > 0$ such that $\mathfrak{a} := \alpha Q_A \leq P$. Then we have $\mathfrak{a} \in \alpha Q_P(\mathcal{A})$ and hence we obtain $A \sim_P A$.

The last statement follows by induction over k, where the initial step $k = 2$ is simply the definition of kinship. Let us therefore assume the statement is true for some $k \geq 2$. Let

$A_1, \dots, A_{k+1} \in \mathcal{A}$ be pairwise kin. By assumption there is a $\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_k)$. Since this latter yields $A_1 \subset \operatorname{supp} \mathfrak{b}$ we find $A_1 \sim_P \operatorname{supp} \mathfrak{b}$ and by transitivity of $\sim_P$ we hence have $A_{k+1} \sim_P \operatorname{supp} \mathfrak{b}$. By definition there is thus a $\tilde{\mathfrak{b}} \in \mathcal{Q}_P(A_{k+1} \cup \operatorname{supp} \mathfrak{b})$ and since this gives $A_1 \cup \ldots \cup A_{k+1} \subset A_{k+1} \cup \operatorname{supp} \mathfrak{b} \subset \operatorname{supp} \tilde{\mathfrak{b}}$ we find $\tilde{\mathfrak{b}} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_{k+1})$. ■

Lemma 32 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $Q \in \mathcal{S}(\mathcal{A})$ with representing forest $F \in \mathcal{F}_{\mathcal{A}}$. Then for all $A \in F$ we have*

$$Q(\,\cdot \cap A) = \lambda_Q(A) + Q\big|_{\subsetneq A}\,.$$

Proof of Lemma 32: Let $A_0 \in \max F$ be the root with $A \subset A_0$. Then we can decompose F into $F = \{A' \in F : A' \supset A\} \dot\cup \{A' \in F : A' \subsetneq A\} \dot\cup \{A' \in F : A' \bot A\}$. Moreover, flatness of $\mathcal{Q}$ gives $Q_{A'}(\cdot \cap A) = Q_{A'}(A) \cdot Q_A(\cdot)$ for all $A' \in \mathcal{A}$ with $A \subset A'$ while fittedness gives $Q_{A'}(A) = 0$ for all $A' \in \mathcal{A}$ with $A' \bot A_0$ by the monotonicity of $\bot$, part (a) of Lemma 30, and part (b) of Lemma 51. For $B \in \mathcal{B}$ we thus have

$$\begin{aligned} Q(B \cap A) &= \sum_{A' \supset A} \alpha_{A'} Q_{A'}(B \cap A) + \sum_{A' \subsetneq A} \alpha_{A'} Q_{A'}(B \cap A) + \sum_{A' \bot A_0} \alpha_{A'} Q_{A'}(B \cap A) \\ &= \sum_{A' \supset A} \alpha_{A'} Q_{A'}(A) Q_A(B) + \sum_{A' \subsetneq A} \alpha_{A'} Q_{A'}(B \cap A) \\ &= \lambda_Q(A)(B) + Q\big|_{\subsetneq A}(B)\,, \end{aligned}$$

where the last step uses $Q_{A'}(B \cap A) = Q_{A'}(B)$ for $A' \subset A$, which follows from fittedness. ■

Lemma 33 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $\mathfrak{a}, \mathfrak{b}$ be base measures on $A, B \in \mathcal{A}$ with $A \subset B$. Then for all $C_0 \in \mathcal{B}$ with $\mathfrak{a}(C_0 \cap A) > 0$ we have*

$$\mathfrak{b}(\,\cdot \cap A) = \frac{\mathfrak{b}(C_0 \cap A)}{\mathfrak{a}(C_0 \cap A)} \cdot \mathfrak{a}(\,\cdot \cap A)\,.$$

Proof of Lemma 33: By assumption there are $\alpha, \beta > 0$ with $\mathfrak{a} = \alpha Q_A$ and $\mathfrak{b} = \beta Q_B$. Moreover, flatness guarantees $Q_B(\cdot \cap A) = Q_B(A) \cdot Q_A(\cdot)$. For all $C \in \mathcal{B}$ we thus obtain

$$\mathfrak{b}(C \cap A) = \beta Q_B(C \cap A) = \beta Q_B(A) \cdot Q_A(C) = \beta Q_B(A) \cdot Q_A(C \cap A) = \frac{\beta Q_B(A)}{\alpha} \mathfrak{a}(C \cap A)\,.$$

where in the second to last step we used $Q_A(\cdot) = Q_A(\cdot \cap A)$, which follows from $A = \operatorname{supp} Q_A$. For $C_0 \in \mathcal{B}$ with $\mathfrak{a}(C_0 \cap A) > 0$ we thus find $\frac{\beta Q_B(A)}{\alpha} = \frac{\mathfrak{b}(C_0 \cap A)}{\mathfrak{a}(C_0 \cap A)}$ and inserting this in the previous formula gives the assertion. ■

Lemma 34 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base and $Q \in \mathcal{S}(\mathcal{A})$ be a simple measure, $\mathfrak{a}$ be a base measures on some $A \in \mathcal{A}$, and $C \in \mathcal{B}$. Then the following statements are true:*

(a) If $\mathfrak{a} \le Q$ then there is a level $\mathfrak{b}$ in Q with $\mathfrak{a} \le \mathfrak{b}$.

(b) If $\mathfrak{b}(\cdot \cap C) \le \mathfrak{a}(\cdot \cap C)$ for all levels $\mathfrak{b}$ of Q then $Q(C) \le \mathfrak{a}(C)$.

(c) For all $P \in \mathcal{M}$ we have $Q \leq P$ if and only if $\mathfrak{b} \leq P$ for all levels $\mathfrak{b}$ in Q.

Proof of Lemma 34: In the following we denote the representing forest of Q by F.
(a). By $\mathfrak{a} \leq Q$ we find $A \subset \operatorname{supp} Q = \mathbb{G}F$. Since the roots $\max F$ form a finite $\perp$-disjoint union of closed sets of $\mathbb{G}F$, the $\mathcal{A}$-connectedness shows that A is already contained in one of the roots, say $A_0 \in \max F$. For $F' := \{A' \in F \mid A \subset A'\}$ we thus have $A_0 \in F'$. Moreover, F' is a chain, since if there were $\perp$-disjoint $A', A'' \in F'$ then A would only be contained in one of them by Lemma 30. Therefore there is a unique leaf $B := \min F' \in F$ and thus $A \subset B$. We denote the level of B in Q by $\mathfrak{b}$. Then it suffices to show $\mathfrak{a} \leq \mathfrak{b}$. To this end, let $\{C_1, \ldots, C_k\} = \max F\big|_{\subsetneq B}$ be the direct children of B in F. By construction we know $A \not\subset C_i$ for all $i = 1, \ldots, k$ and hence $\mathcal{A}$-connectedness yields $A \not\subset C_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} C_k$. Therefore $C_0 := A \setminus \bigcup_i C_i$ is non-empty and relatively open in $A = \operatorname{supp} Q_A$. This gives $\mathfrak{a}(C_0 \cap A) > 0$ by Lemma 51. Let us write $\mathfrak{b} := \lambda_Q(B)$ for the level of B in Q. Lemma 32 applied to the node $B \in F$ then gives

$$Q(C_0) = \mathfrak{b}(C_0) + Q\big|_{\subsetneq B}(C) = \mathfrak{b}(C_0) + \sum_{A' \in F : A' \subsetneq B} \alpha_{A'} Q_{A'}(C_0) = \mathfrak{b}(C_0)$$

since for $A' \in F$ with $A' \subsetneq B$ we have $A' \subset \bigcup_i C_i$ and thus $\operatorname{supp} Q_{A'} \cap C_0 = A' \cap C_0 = \emptyset$. Therefore, we find $\mathfrak{a}(C_0 \cap A) = \mathfrak{a}(C_0) \leq Q(C_0) = \mathfrak{b}(C_0) = \mathfrak{b}(C_0 \cap B)$. By Lemma 33 we conclude that $\mathfrak{b}(\cdot \cap A) \geq \mathfrak{a}(\cdot \cap A)$. For $B' \in \mathcal{B}$ the decomposition $B' = (B' \setminus A) \dot\cup (B' \cap A)$ and the fact that $A = \operatorname{supp} \mathfrak{a} \subset \operatorname{supp} \mathfrak{b}$ then yields the assertion.
(b). For $A \in F$ we define

$$B_A := A \setminus \bigcup_{A' \in F :\, A' \subsetneq A} A' \,,$$

i.e. B_A is obtained by removing the strict descendants from A. From this description it is easy to see that $\{B_A : A \in F\}$ is a partition of $\mathbb{G}(F) = \operatorname{supp} Q$. Hence we obtain

$$\begin{aligned} Q(C) = \sum_{A \in F} Q(C \cap B_A) &= \sum_{A \in F} \sum_{A' \in F} \alpha_{A'} Q_{A'}(C \cap B_A) \\ &= \sum_{A \in F} \sum_{A' \supset A} \alpha_{A'} Q_{A'}(C \cap B_A) + \sum_{A \in F} \sum_{A' \subsetneq A} \alpha_{A'} Q_{A'}(C \cap B_A) \\ &= \sum_{A \in F} \lambda_Q(A)(C \cap B_A) \,, \end{aligned} \tag{22}$$

where we used $Q_{A'}(C \cap B_A) = Q_{A'}(C \cap B_A \cap A)$ together with flatness applied to pairs $A \subset A'$ as well as $A' \cap B_A = \emptyset$ applied to pairs $A' \subsetneq A$. Our assumption now yields

$$Q(C) \leq \sum_{A \in F} \mathfrak{a}(C \cap B_A) = \mathfrak{a}(C \cap \operatorname{supp} Q) \leq \mathfrak{a}(C).$$

(c). Let $\mathfrak{b} := \lambda_Q(B)$ be a level of B in Q with $\mathfrak{b} \not\leq P$. Then there is a $B' \in \mathcal{B}$ with $\mathfrak{b}(B') > P(B)$ and for $B'' := B' \cap \operatorname{supp} \mathfrak{b} = B' \cap B$ we find $Q(B'') \geq \mathfrak{a}(B'') = \mathfrak{a}(B') > P(B') \geq P(B'')$. Conversely, assume $\mathfrak{b} \leq P$ for all levels $\mathfrak{b}$ in Q. By the decomposition (22) we then obtain

$$Q(C) = \sum_{A \in F} \lambda_Q(A)(C \cap B_A) \leq \sum_{A \in F} P(C \cap B_A) = P(C \cap \operatorname{supp} Q) \leq P(C). \qquad \blacksquare$$

Corollary 35 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base, $Q \in \mathcal{S}(\mathcal{A})$ a simple measure with representing forest F and $A_1, A_2 \in F$. Then for all $\mathfrak{a} \in \mathcal{Q}_Q(A_1 \cup A_2)$ there exists a level $\mathfrak{b}$ in Q such that $A_1 \cup A_2 \subset B$ and $\mathfrak{a} \leq \mathfrak{b}$.*

Proof of Corollary 35: Let us fix an $\mathfrak{a} \in \mathcal{Q}_Q(A_1 \cup A_2)$. Since $\mathfrak{a} \leq Q$, Lemma 34 gives a level $\mathfrak{b}$ in Q with $\mathfrak{a} \leq \mathfrak{b}$. Setting $B := \operatorname{supp} \mathfrak{b} \in F$ then gives $A_1 \cup A_2 \subset \operatorname{supp} \mathfrak{a} \subset B$. ■

Proof of Proposition 19: Let Q be a simple measure and $Q = \sum_{A \in F} \alpha_A Q_A$ be its unique representation. Moreover, let A_1, A_2 be direct siblings in F and $\mathfrak{a}_1, \mathfrak{a}_2$ be the corresponding levels in Q. Then Q-groundedness follows directly from Corollary 35. To show that A_1, A_2 are Q-motivated and Q-fine, we fix an $\mathfrak{a} \in \mathcal{Q}_Q(A_1 \cup A_2)$. Furthermore, let $\mathfrak{b}$ be the level in Q found by Corollary 35, i.e. we have $A_1 \cup A_2 \subset \operatorname{supp} \mathfrak{b} =: B$ and $\mathfrak{a} \leq \mathfrak{b} \leq Q$. Now let $A_3, \ldots, A_k \in F$ be the remaining direct siblings of A_1 and A_2. Since B is an ancestor of A_1 and A_2 it is also an ancestor of $A_3, \ldots, A_k$ and hence $A_1 \cup \cdots \cup A_k \subset B$. This immediately gives $\mathfrak{b} \in \mathcal{Q}_Q(A_1 \cup \cdots \cup A_k)$ and we already know $\mathfrak{b} \geq \mathfrak{a}$. In other words, A_1, A_2 are Q-fine. Finally, observe that for $B \subset A'$ flatness gives $Q_{A'}(B) Q_B(\cdot) = Q_{A'}(\cdot \cap B)$. Since $A_1 \subset B$ we hence obtain

$$\mathfrak{a}(A_1) \leq \mathfrak{b}(A_1) = \sum_{A' \supset B} \alpha_{A'} Q_{A'}(B) Q_B(A_1) = \sum_{A' \supset B} \alpha_{A'} Q_{A'}(A_1)$$

and since $Q_{A_1}(A_1) = 1$ we also find

$$\mathfrak{a}_1(A_1) = \sum_{A' \supset A_1} \alpha_{A'} Q_{A'}(A_1) Q_{A_1}(A_1) = \sum_{A' \supset A_1} \alpha_{A'} Q_{A'}(A_1) = \sum_{A' \supset B} \alpha_{A'} Q_{A'}(A_1) + \alpha_{A_1} .$$

Since $\alpha_{A_1} > 0$ we conclude that $\mathfrak{a}(A_1) < (1 - \varepsilon_1)\mathfrak{a}_1(A_1)$ for a suitable $\varepsilon_1 > 0$. Analogously, we find an $\varepsilon_2 > 0$ with $\mathfrak{a}(A_2) < (1 - \varepsilon_2)\mathfrak{a}_2(A_2)$ and taking $\alpha := 1 - \min\{\varepsilon_1, \varepsilon_2\}$ thus yields Q-motivation. ■

5.2.1 Proof of Theorem 20

Lemma 36 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a clustering base, $P \in \mathcal{M}_\Omega$, and $Q, Q' \leq P$ be simple measures on finite forests F and F'. If all roots in both F and F' are P-grounded, then any root in one tree can only be kin below P to at most one root in the other tree.*

Proof of Lemma 36: Let us assume the converse, i.e. we have an $A \in \max F$ and $B, B' \in \max F'$ such that $A \sim_P B$ and $A \sim_P B'$. Let $\mathfrak{a}, \mathfrak{b}, \mathfrak{b}'$ be the respective summands in the simple measures Q and Q'. Then $0 < \mathfrak{a}(A) \leq Q(A) \leq P(A)$ and analogously $P(B), P(B') > 0$. Then by transitivity of $\sim_P$ established in Lemma 31 we have $B \sim_P B'$ and by groundedness there has to be a parent for both in F', so they would not be roots. ■

Proposition 37 *Let $(\mathcal{A}, \mathcal{Q}, \bot)$ be a stable clustering base and $P \in \mathcal{M}$ such that $\mathcal{A}$ is P-subadditive. Let $(Q_n, F_n) \uparrow P$, where all forests F_n have k roots $A_n^1, \ldots, A_n^k$, which, in addition, are assumed to be P-grounded. Then $A^i := \bigcup_n A_n^i$ are unique under all such approximations up to a P-null set.*

Proof of Proposition 37: The $A^1, \ldots, A^k$ are pairwise $\perp$-disjoint by Lemma 11 and by Lemma 53 they partition $\operatorname{supp} P$ up to a P-null set, i.e. $P(\operatorname{supp} P \setminus \bigcup_i A^i) = 0$. Therefore any $B \in \mathcal{B}$ with $P(B) > 0$ intersects at least one of the A_i. Moreover, we have $0 < Q_1(A_1^i) \leq P(A_1^i) \leq P(A^i)$, i.e. $P(A^i) > 0$. Now let $(Q_n', F_n') \uparrow P$ be another approximation of the assumed type with roots B_n^i and limit roots $B^1, \ldots, B^{k'}$. Clearly, our preliminary considerations also hold for these limit roots. Now consider the binary relation $i \sim j$, which is defined to hold iff $A^i \circledcirc_P B^j$.

Since $P(A^i) > 0$ there has to be a B^j with $P(A^i \cap B^j) > 0$, so for all $i \leq k$ there is a $j \leq k'$ with $i \sim j$. Then, since $A_n^i \cap B_n^j \uparrow A^i \cap B^j$, there is an $n \geq 1$ with $P(A_n^i \cap B_n^j) > 0$. By P-subadditivity of $\mathcal{A}$ we conclude that A_n^i and B_n^j are kin below P, and Lemma 36 shows that this can only happen for at most one $j \leq k'$. Consequently, we have $k \leq k'$ and $\sim$ defines an injection $i \mapsto j(i)$. The same argument also holds in the other direction and we see that $k = k'$ and that $i \sim j$ defines a bijection. Clearly, we may assume that $i \sim j$ iff $i = j$. Then $P(A^i \cap B^j) > 0$ if and only if $i = j$, and since both sets of roots partition $\operatorname{supp} P$ up to a P-null set, we conclude that $P(A^i \triangle B^i) = 0$. ∎

Lemma 38 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and $P \in \mathcal{M}$ such that $\mathcal{A}$ is P-subadditive. Moreover, let $\mathfrak{a}_1, \ldots, \mathfrak{a}_k \leq P$ be base measures on $A_1, \ldots, A_k \in \mathcal{A}$ such that $A_1 \circledcirc_P A_i$ for all $2 \leq i \leq k$. Then there is $\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_k)$ and an $\mathfrak{a}_i$ such that $\mathfrak{b} \geq \mathfrak{a}_i$, and if $k \geq 3$ and the $\mathfrak{a}_2, \ldots, \mathfrak{a}_k$ satisfy the motivation implication (20) pairwise, then $\mathfrak{b} \geq \mathfrak{a}_1$.*

Proof of Lemma 38: The proof of the first assertion is based on induction. For $k = 2$ the assertion is P-subadditivity. Now assume that the statement is true for k. Then there is a $\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_k)$ and an $i_0 \leq k$ with $\mathfrak{b} \geq \mathfrak{a}_{i_0}$. The assumed $A_1 \circledcirc_P A_{k+1}$ thus yields

$$P(A_{k+1} \cap \operatorname{supp} \mathfrak{b}) \geq P(A_{k+1} \cap A_1) > 0\,,$$

and hence P-subadditivity gives a $\tilde{b} \in \mathcal{Q}_P(A_{k+1} \cup \operatorname{supp} \mathfrak{b})$ with $\tilde{\mathfrak{b}} \geq \mathfrak{a}_{k+1}$ or $\tilde{\mathfrak{b}} \geq \mathfrak{b} \geq \mathfrak{a}_{i_0}$. For the second assertion observe that $\mathfrak{b} \in \mathcal{Q}_P(A_i \cap A_j)$ for all i, j and hence (20) implies $\mathfrak{b} \not\geq \mathfrak{a}_i$ for $i \geq 2$. ∎

Lemma 39 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and $Q \leq P$ be a simple and P-adapted measure with representing forest F. Let $C^1, \ldots, C^k \in F$ be direct siblings for some $k \geq 2$. Then there exists an $\varepsilon > 0$ such that:*

(a) For all $\mathfrak{a} \in \mathcal{Q}_P(C^1 \cup \ldots \cup C^k)$ and $i \leq k$ we have $\mathfrak{a}(C^i) \leq (1 - \varepsilon) \cdot Q(C^i)$.

(b) Assume that $\mathcal{A}$ is P-subadditive and that $\mathfrak{a} \leq P$ is a simple measure with $\operatorname{supp} \mathfrak{a} \circledcirc_P C^i$ for at least two $i \leq k$. Then for all $i \leq k$ we have $\mathfrak{a}(C^i) \leq (1 - \varepsilon) \cdot Q(C^i)$.

(c) If $\mathcal{A}$ is P-subadditive and $Q' \leq P$ is a simple measure with representing forest F' such that there is an $i \leq k$ with the property that for all $B \in F'$ we have

$$B \circledcirc_P C^i \implies \exists j \neq i \colon B \circledcirc_P C^j.$$

Then $Q'(\cdot \cap C^i) \leq (1 - \varepsilon)\, Q(\cdot \cap C^i)$ holds true.

Proof of Lemma 39: Let $\mathfrak{c}_1, \dots, \mathfrak{c}_k$ be the levels of $C^1, \dots, C^k$ in Q. Since Q is adapted, (20) holds for some $\alpha \in (0,1)$. We define $\varepsilon := 1 - \alpha$.
(a). We fix an $\mathfrak{a} \in \mathcal{Q}_P(C^1 \cup \ldots \cup C^k)$, an $i \leq k$, and a $j \leq k$ with $j \neq i$. Let $\mathfrak{c}_i, \mathfrak{c}_j$ be the levels of C^i and C^j in Q. Since $\alpha\mathfrak{c}_i$ and $\alpha\mathfrak{c}_j$ are motivated, we have $\mathfrak{a} \not\geq \alpha\mathfrak{c}_i$ and $\mathfrak{a} \not\geq \alpha\mathfrak{c}_j$. Hence, there is a $C_0 \in \mathcal{B}$ with $\mathfrak{a}(C_0) < \alpha\mathfrak{c}_i(C_0)$ and thus also $\mathfrak{a}(C_0 \cap C^i) < \alpha\mathfrak{c}_i(C_0 \cap C^i)$. Lemma 33 then yields $\mathfrak{a}(\cdot \cap C^i) \leq \alpha\mathfrak{c}_i(\cdot \cap C^i)$ and the definition of levels gives

$$\mathfrak{a}(C^i) \leq \alpha\mathfrak{c}_i(C^i) = \alpha Q(C^i) = (1-\varepsilon)Q(C^i).$$

(b). We may assume $\operatorname{supp}\mathfrak{a} \not\perp_P C^1$ and $\operatorname{supp}\mathfrak{a} \not\perp_P C^2$. By the second part of Lemma 38 applied to $\operatorname{supp}\mathfrak{a}, C^1, C^2$ there is an $\mathfrak{a}' \in \mathcal{Q}_P(\operatorname{supp}\mathfrak{a} \cup C^1 \cup C^2) \subset \mathcal{Q}_P(C^1 \cup C^2)$ with $\mathfrak{a}' \geq \mathfrak{a}$, and since Q is P-fine, we may actually assume that $\mathfrak{a}' \in \mathcal{Q}_P(C^1 \cup \ldots \cup C^k)$. Now part (a) yields $\mathfrak{a}'(C^i) \leq (1-\varepsilon) \cdot Q(C^i)$ for all $i = 1, \dots, k$.
(c). We may assume $i = 1$. Our first goal is to show

$$\mathfrak{b}(\cdot \cap C^1) \leq (1-\varepsilon)\mathfrak{c}_1(\cdot \cap C^1) \tag{23}$$

for all levels $\mathfrak{b}$ in Q', To this end, we fix a level $\mathfrak{b}$ in Q' and write $B := \operatorname{supp}\mathfrak{b}$. If $P(B \cap C^1) = 0$, then (23) follows from

$$\mathfrak{b}(C^1) = \mathfrak{b}(B \cap C^1) \leq P(B \cap C^1) = 0\,.$$

In the other case we have $B \not\perp_P C^1$ and our assumption gives a $j \neq 1$ with $B \not\perp_P C^j$. By the second part of Lemma 38 we find an $\mathfrak{a} \in \mathcal{Q}_P(B \cup C^1 \cup C^j) \subset \mathcal{Q}_P(C^1 \cup C^j)$ with $\mathfrak{a} \geq \mathfrak{b}$, and by (a) we thus obtain $\mathfrak{a}(C^1) \leq (1-\varepsilon)\,Q(C^1) = (1-\varepsilon)\mathfrak{c}_1(C^1)$. Now, Lemma 33 gives $\mathfrak{a}(\cdot \cap C^1) \leq (1-\varepsilon)\mathfrak{c}_1(\cdot \cap C^1)$ and hence (23) follows.

With the help of (23) we now conclude by part (b) of Lemma 34 that $Q'(\cdot \cap C^1) \leq (1-\varepsilon)\mathfrak{c}_1(\cdot \cap C^1)$ and using $\mathfrak{c}_1(\cdot \cap C^1) \leq Q(\cdot \cap C^1)$ we thus obtain the assertion. ∎

Lemma 40 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base and $P \in \mathcal{M}$ such that $\mathcal{A}$ is P-subadditive. Moreover, let $Q, Q' \leq P$ be simple P-adapted measures on F, F', and $S \in s(F)$ and $S' \in s(F')$ be two nodes that have children in $s(F)$ and $s(F')$, respectively. Let*

$$\{C^1, \dots, C^k\} = \max s(F)\big|_{\subsetneq S} \qquad \text{and} \qquad \{D^1, \dots, D^{k'}\} = \max s(F')\big|_{\subsetneq S'}$$

be their direct children and consider the relation $i \sim j :\Leftrightarrow C^i \not\perp_P D^j$. Then we have $k, k' \geq 2$ and if $\sim$ is left-total, i.e. for every $i \leq k$ there is a $j \leq k'$ with $i \sim j$, then it is right-unique, i.e. for every $i \leq k$ there is at most one $j \leq k'$ with $i \sim j$.

Proof of Lemma 40: The definition of the structure of a forest gives $k, k' \geq 2$. Moreover, we note that $P(A) \geq Q(A) > 0$ for all $A \in F$ and $P(A) \geq Q'(A) > 0$ for all $A \in F'$. Now assume that $\sim$ is not right-unique, say $1 \sim j$ and $1 \sim j'$ for some $j \neq j'$. Applying P-subadditivity twice we then find a $\mathfrak{b} \in \mathcal{Q}_P(C^1 \cup D^j \cup D^{j'})$ with $\mathfrak{b} \geq \mathfrak{c}_1$ or $\mathfrak{b} \geq \mathfrak{d}_j$ or $\mathfrak{b} \geq \mathfrak{d}_{j'}$, where $\mathfrak{c}_1$, $\mathfrak{d}_j$, and $\mathfrak{d}_{j'}$ are the corresponding levels. Since $\mathfrak{d}^j, \mathfrak{d}^{j'}$ are motivated we conclude that $\mathfrak{b} \geq \mathfrak{c}_1$. Now, because of $\mathcal{Q}_P(C^1 \cup D^j \cup D^{j'}) \subset \mathcal{Q}_P(D^j \cup D^{j'})$ and P-fineness of Q' there is a $\mathfrak{b}' \in \mathcal{Q}_P(D_1 \cup \ldots \cup D_{k'})$ with $\mathfrak{b}' \geq \mathfrak{b}$. Now pick a direct sibling of C^1, say C^2. Then there is a j'' with $2 \sim j''$, and since $B' := \operatorname{supp}\mathfrak{b}' \supset D_1 \cup \ldots \cup D_{k'}$ this implies

$P(B' \cap C^2) \geq P(D^{j''} \cap C^i) > 0$. By P-subadditivity we hence find a $\mathfrak{b}'' \in \mathcal{Q}_P(B' \cup C^2) \subset \mathcal{Q}_P(C^1 \cup C^2)$ with $\mathfrak{b}'' \geq \mathfrak{b}'$ or $\mathfrak{b}'' \geq \mathfrak{c}_2$. Clearly, $\mathfrak{b}'' \geq \mathfrak{c}_2$ violates the fact that C^1, C^2 are motivated, and thus $\mathfrak{b}'' \geq \mathfrak{b}'$. However, we have shown $\mathfrak{b}' \geq \mathfrak{b} \geq \mathfrak{c}_1$, and thus $\mathfrak{b}'' \geq \mathfrak{c}_1$. Since this again violates the fact that C^1, C^2 are motivated, we have found a contradiction. ■

Proof of Theorem 20: We prove the theorem by induction over the generations in the forests. For a finite forest F, we define $s_0(F) := \max F$ and

$$s_{N+1}(F) := s_N(F) \cup \big\{ A \in s(F) \mid A \text{ is a direct child of a leaf in } s_N(F) \big\}.$$

We will now show by induction over N that there is a graph-isomorphism $\zeta_N \colon s_N(F_\infty) \to s_N(F'_\infty)$ with $P(A \triangle \zeta_N(A)) = 0$ for all $A \in s_N(F_\infty)$. For $N = 0$ this has already been shown in Proposition 37. Let us therefore assume that the statement is true for some $N \geq 0$. Let us fix an $S \in \min s_N(F_\infty)$ and let $S' := \zeta_N(S) \in \min s_N(F'_\infty)$ be the corresponding node. We have to show that both have the same number of direct children in $s_{N+1}(\cdot)$ and that these children are equal up to P-null sets. By induction this then finishes the proof.

Since $S \in s_N(F_\infty) \subset s(F_\infty)$, the node S has either no children or at least 2. Now, if both S and S' have no direct children then we are finished. Hence we can assume that S has direct children $C^1, \ldots, C^k$ for some $k \geq 2$, i.e.

$$\max(F_\infty|_{\subsetneq S}) = \{C^1, \ldots, C^k\}.$$

Let $S_n, C^1_n, \ldots, C^k_n \in s(F_n)$ and $S'_n \in s(F'_n)$ be the nodes that correspond to $S, C^1, \ldots, C^k$, and S', respectively. Since $P(S \triangle S') = 0$ we then obtain for all $i \leq k$

$$P(S' \cap C^i) = P(S \cap C^i) = P(C^i) \geq Q_1(C^i) \geq Q_1(C^i_1) > 0\,,$$

that is $S' \circledast_P C^i$ for all $i \leq k$. Since $S' = \bigcup_n S'_n$ and $C^i = \bigcup_n C^i_n$ this can only happen if $S'_n \circledast_P C^i_n$ for all sufficiently large n. We therefore may assume without loss of generality that

$$S'_1 \circledast_P C^i_n \qquad \text{for all } i \leq k \text{ and all } n \geq 1. \tag{24}$$

Let us now investigate the structure of $F'_n|_{\subset S'_n}$. To this end, we will seek a kind of anchor $B'_n \in F'_n|_{\subset S'_n}$, which will turn out later to be the direct parent of the yet to find $\zeta_{N+1}(C^i) \in F'_\infty$. We define this anchor by

$$B'_n := \min\{B \in F'_n \mid B \circledast_P C^i_1 \text{ for all } i = 1, \ldots, k\}.$$

This minimum is unique. Indeed, let $\tilde{B}'_n$ be any other minimum with $\tilde{B}'_n \circledast_P C^i_1$ for all $i \leq k$. Since both are minima, none is contained in the other and because F'_n is a forest this means $B'_n \perp \tilde{B}'_n$. Let $\mathfrak{b}'_n$ and $\tilde{\mathfrak{b}}'_n$ be their levels in Q'_n. Since Q'_n is P-adapted, these two levels are motivated. This means that there can be no base measure majorizing one of them and supporting $B'_n \cup \tilde{B}'_n$. On the other hand, by the second part of Lemma 38 there exists a $\mathfrak{b}''_n \in \mathcal{Q}_p(B'_n \cup C^1_1 \cup \cdots \cup C^k_1)$ with $\mathfrak{b}''_n \geq \mathfrak{b}'_n$. Now because of $P(\tilde{B}'_n \cap \operatorname{supp} \mathfrak{b}''_n) \geq P(\tilde{B}'_n \cap C^1_1) > 0$ and P-subadditivity there exists a base measure majorizing $\tilde{\mathfrak{b}}'_n \geq \mathfrak{b}'_n$ or $\mathfrak{b}''_n$ and supporting $\tilde{B}'_n \cap \operatorname{supp} \mathfrak{b}''_n$. This contradicts the motivatedness of $\mathfrak{b}'_n$ and $\tilde{\mathfrak{b}}'_n$ and hence the minimum B'_n is unique.

Since B'_n is the unique minimum among all $B \in F'_n$ with $B \Subset_P C_1^i$ for all i, we also have $B'_n \subset B$ for all such B and hence $B'_n \subset S'_n$ by (24). The major difficulty in handling B'_n though is that it may jump around as a function of n: Indeed we may have $B'_n \in F'_n \setminus s(F'_n)$ and therefore the monotonicity $s(F'_n) \leq s(F'_{n+1})$ says nothing about B'_n. In particular, we have in general $B'_n \not\subset B'_{n+1}$.

Let us now enumerate the set $\min F'_n\big|_{\subsetneq B'_n}$ of direct children of B'_n by $D_n^1, \ldots, D_n^{k_n}$, where $k_n \geq 0$. Again these D_n^i can jump around as a function of n. The number k_n specifies different cases: we have $B'_n \in \min F'_n$, i.e. B'_n is a leaf, iff $k_n = 0$; on the other hand $D_n^i \in s(F'_n)$ iff $k_n \geq 2$. Next we show that for all $i \leq k$ and all sufficiently large n there is an index $j(i,n) \in \{1, \ldots, k_n\}$ with

$$C_1^i \Subset_P D_n^{j(i,n)} \,. \tag{25}$$

Note that this in particular implies $k_n \geq 1$ for sufficiently large n. To this end we fix an $i \leq k$. Suppose that $C_1^i \perp\!\!\!\perp_P (D_{n_m}^1 \cup \cdots \cup D_{n_m}^{k_{n_m}})$ for infinitely many $n_1, n_2, \ldots$. By construction B'_{n_m} is the smallest element of F'_{n_m} that $\perp\!\!\!\perp_P$-intersects C_1^i. More precisely, for any $A \in F'_{n_m}$ with $A \Subset_P C_1^i$ we have $A \supset B'_{n_m}$ and therefore $A \Subset_P C_1^{i'}$ for all such A and all $i' \leq k$. Hence, all Q'_{n_m} in this subsequence fulfill the conditions of the last statement in Lemma 39 and we get an $\varepsilon > 0$ such that for all such n_m

$$Q'_{n_m}(C_1^i) \leq (1-\varepsilon) Q_1(C_1^i) \leq (1-\varepsilon) P(C_1^i) \tag{26}$$

which contradicts $Q'_{n_m}(C_1^i) \uparrow P(C_1^i)$ since $P(C_1^i) > 0$.

Therefore for all $i \leq k$ and all sufficiently large n there is an index $j(i,n)$ such that (25) holds. Clearly, we may thus assume that there is such an $j(i,n)$ for all $n \geq 1$. Since $j(i,n) \in \{1, \ldots, k_n\}$ we conclude that $k_n \geq 1$ for all $n \geq 1$. Moreover, $k_n = 1$ is impossible, since $k_n = 1$ yields $j(i,n) = 1$, and this would mean, that $C_1^i \Subset_P D_n^1$ for all $i \leq k$ contradicting that B'_n is the minimal set in F'_n having this property. Consequently B'_n has the direct children $D_n^1, \ldots, D_n^{k_n}$ where $k_n \geq 2$ for all $n \geq 1$.

So far we have seen that $D_n^1, \ldots, D_n^{k_n} \in s(F'_n)$ are inside S'_n. Therefore S'_n is not a leaf, and hence $S' \notin \min F'_\infty$ as well. But still for infinitely many n these D_n^j might not be the direct children of S'_n. Let us therefore denote the direct children of $S'_n \in s(F'_n)$ by $E_n^1, \ldots, E_n^{k'} \in s(F'_n)$, where we pick a numbering such that $E_n^i \subset E_{n+1}^i$ and by the definition of the structure of a forest we have $k' \geq 2$.

For an arbitrary but fixed n we now show $\{D_n^1, \ldots, D_n^{k_n}\} = \{E_n^1, \ldots, E_n^{k'}\}$. To this let us assume the converse. Since the E_n^j are the direct children of S'_n in the structure $s(F'_n)$ there is a $j_n \leq k'$ with $D_n^j \subset E_n^{j_n}$ for all j, and since B'_n is the direct parent of the D_n^j we conclude that $B'_n \subset E_n^{j_n}$. Therefore we have $C_1^i \Subset_P E_n^{j_n}$ for all $i \leq k$. Since Q_1 and Q'_n are adapted we can use Lemma 40 to see that for all $i \leq k$ we have $C_1^i \perp\!\!\!\perp_P E_n^j$ for all $j \neq j_n$. Let us fix a $j \neq j_n$. Our goal is to show

$$Q_m(E_n^j) < (1-\varepsilon) Q'_n(E_n^j) \,,$$

for all sufficiently large $m \geq n$, since this inequality contradicts the assumed convergence of $Q_m(E_n^j)$ to $P(E_n^j) \geq Q'_n(E_n^j) > 0$. By part (c) of Lemma 39 with Q'_n as Q and Q_m as Q' it suffices to show that for all $A \in F_m$ and all sufficiently large $m \geq n$ we have

$$A \Subset_P E_n^j \implies A \Subset_P E_n^{j_n} \,. \tag{27}$$

To this end, we fix an $A \in F_m$ with $A \not\perp_P E_n^j$. Then we first observe that for all $m \geq n$ we have $P(A \cap S'_m) \geq P(A \cap S'_n) \geq P(A \cap E_n^j) > 0$. Moreover, the induction assumption ensures $P(S \triangle S') = 0$ and since $S_m \nearrow S$ and $S'_m \nearrow S'$, we conclude that $P(A \cap S_m) > 0$ for all sufficiently large m. Now, $C_m^1 \cup \cdots \cup C_m^k$ are direct siblings and hence we either have $C_m^1 \cup \cdots \cup C_m^k \subset A$ or $A \subset C_m^{i_0}$ for exactly one $i_0 \leq k$. In the first case we get

$$P(A \cap E_n^{j_n}) \geq P(C_m^1 \cap E_n^{j_n}) \geq P(C_1^1 \cap E_n^{j_n}) > 0$$

by the already established $C_1^i \not\perp_P E_n^{j_n}$ for all $i \leq k$. The second case is impossible, since it contradicts adaptedness. Indeed, $A \subset C_m^{i_0}$ implies $C_m^{i_0} \not\perp_P E_n^j$ and by the already established $C_1^i \not\perp_P E_n^{j_n}$ for all $i \leq k$, we also know $C_m^{i_0} \not\perp_P E_n^{j_n}$. By the second part of Lemma 38 we therefore find a $\tilde{\mathfrak{c}} \in \mathcal{Q}_P(C_m^{i_0} \cup E_n^j \cup E_n^{j_n})$ with $\tilde{\mathfrak{c}} \geq \mathfrak{c}_m^{i_0}$, where $\mathfrak{c}_m^{i_0}$ is the level of $C_m^{i_0}$ in Q_m. Now fix any $i \leq k$ with $i \neq i_0$ and observe that we have $P(C_m^i \cap \operatorname{supp} \tilde{\mathfrak{c}}) \geq P(C_m^i \cap E_n^{j_n}) \geq P(C_1^i \cap E_n^{j_n}) > 0$, and hence P-subadditivity yields a $\mathfrak{c}'' \in \mathcal{Q}_P(C_m^i \cup \operatorname{supp} \tilde{\mathfrak{c}})$ with $\mathfrak{c}'' \geq \mathfrak{c}_m^i$ or $\mathfrak{c}'' \geq \tilde{\mathfrak{c}} \geq \mathfrak{c}_m^{i_0}$, where $\mathfrak{c}_m^i$ is the level of C_m^i in Q_m. Since $\mathfrak{c}'' \in \mathcal{Q}_P(C_m^i \cup \operatorname{supp} \tilde{\mathfrak{c}}) \subset \mathcal{Q}_P(C_m^i \cup C_m^{i_0})$, we have thus found a contradiction to the fact that the direct siblings C_m^i and $C_m^{i_0}$ are P-motivated.

So far we have shown $\{D_n^1, \ldots, D_n^{k_n}\} = \{E_n^1, \ldots, E_n^{k'}\}$ and $k_n = k'$ for all n. Without loss of generality we may thus assume that $D_n^j = E_n^j$ for all n and all $j \leq k'$. In particular, this means that the direct children of S'_n in $s(F'_n)$ equal the direct children of B'_n in F'_n. Let us write

$$D^j := \bigcup_{n \geq 1} D_n^j, \qquad j = 1, \ldots, k'$$

and $i \sim j$ iff $C_1^i \not\perp_P D_1^j$. We have seen around (25) that for all $i \leq k$ there is at least one $j \leq k_1 = k'$ with $i \sim j$, namely $j(i,1)$. By Lemma 40 we then conclude that $j(i,1)$ is the only index $j \leq k'$ satisfying $i \sim j$. By reversing the roles of C_1^i and D_1^j, which is possible since $D_1^j = E_1^j$ is a direct children of S'_n in $s(F'_n)$, we can further see that for all j there is an index i with $i \sim j$ and again by Lemma 40 we conclude that there is at most one i with $i \sim j$. Consequently, $i \sim j$ defines a bijection between $\{C_1^1, \ldots, C_1^k\}$ and $\{D_1^1, \ldots, D_1^{k'}\}$ and hence we have $k = k'$. Moreover, we may assume without loss of generality that $i \sim j$ iff $i = j$. From the latter we obtain $C_1^i \not\perp_P D_1^j$ iff $i = j$.

To generalize the latter, we fix $n, m \geq 1$ and write $i \sim j$ iff $C_n^i \not\perp_P D_m^j$. Since we have $P(C_n^i \cap D_m^i) \geq P(C_1^i \cap D_1^i) > 0$, we conclude that $i \sim i$, and by Lemma 40 we again see that $i \sim j$ is false for $i \neq j$. This yields $C_n^i \not\perp_P D_m^j$ iff $i = j$ and by taking the limits, we find $C^i \not\perp_P D^j$ iff $i = j$.

Next we show that $P(C^i \triangle D^i) = 0$ for all $i \leq k$. Clearly, it suffices to consider the case $i = 1$. To this end assume that $R := C^1 \setminus D^1$ satisfies $P(R) > 0$. For $R_n := R \cap C_n^1 = C_n^1 \setminus D^1$, we then have $R_n \uparrow R$ since $C_n^1 \uparrow C^1$ and $R \subset C^1$. Consequently, $0 < P(R) = P(R \cap C^1)$ implies $P(R_n) > 0$ for all sufficiently large n. On the other hand, we have $P(R \cap D^1) = 0$ by the definition of R and $P(R \cap D^j) \leq P(C^1 \cap D^j) = 0$ for all $j \neq 1$ as we have shown above.

We next show that $Q'_m(R_n) = Q'_m\big|_{\supset B'_m}(R_n)$. To this end it suffices to show that for any $A \in F'_m$ with $A \notin F'_m\big|_{\supset B'_m}$ we have $Q'_m(A \cap R_n) \leq P(A \cap R_n) = 0$. Let us thus fix an $A \in F'_m$ with $A \notin F'_m\big|_{\supset B'_m}$. Then we either have $A \subsetneqq B'_m$ or $A \perp B'_m$. In the first case there

is $j \leq k$ with $A \subset D_m^j$ which means, as shown above, that $P(A \cap R_n) \leq P(D_m^j \cap R_n) = 0$. In the second case, by definition of structure, we even have $A \perp S'_m$. So there is a $A'_m \in s(F'_m)$ with $A \subset A'_m$ and $A'_m \perp S'_m$ and by isomonotonicity of the structure there is $A' \in F'_\infty$ with $A'_m \subset A'$ and $A' \perp S'$. Hence by induction assumption $P(A \cap R_n) \leq P(A \cap S_n) \leq P(A \cap S) \leq P(A' \cap S) = P(A' \cap S') = 0$.

Using $P(C^i \cap D^i) > 0$ we now observe that $Q'_m\big|_{\supset B'_m}$ fulfills the conditions of part (c) of Lemma 39 for C^1 and C^2 and by $R_n \subset C_n^1$ we thus obtain

$$Q'_m(R_n) = Q'_m\big|_{\supset B'_m}(R_n) \leq (1-\varepsilon)Q_n(R_n) \leq (1-\varepsilon)P(R_n).$$

This contradicts $0 < P(R_n) = \lim_{m\to\infty} Q'_m(R_n)$. So we can assume $P(R_n) = 0$ for all n and therefore $P(R) = \lim_{n\to\infty} P(R_n) = 0$. By reversing roles we thus find $P(D^1 \triangle C^1) = P(C^1 \setminus D^1) + P(D^1 \setminus C^1) = 0$ and therefore the children are indeed the same up to P-null sets.

Finally, we are able to finish the induction: To this end we extend ζ_N to the map $\zeta_{N+1}\colon s_{N+1}(F_\infty) \to s_{N+1}(F'_\infty)$ by setting, for every leaf $S \in \min s_N(F_\infty)$,

$$\zeta_{N+1}(C^i) := D^i$$

where $C^1, \ldots, C^k \in s_{N+1}(F_\infty)$ are the direct children of S and $D^1, \ldots, D^k \in s_{N+1}(F'_\infty)$ are the nodes we have found during our above construction. Clearly, our construction shows that ζ_{N+1} is a graph isomorphism satisfying $P(A \triangle \zeta_{N+1}(A)) = 0$ for all $A \in s_{N+1}(F_\infty)$. ■

5.2.2 Proof of Theorem 21

Lemma 41 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base, $P_1, \ldots, P_k \in \mathcal{M}$ with $\operatorname{supp} P_i \perp \operatorname{supp} P_j$ for all $i \neq j$, and $Q_i \leq P_i$ be simple measures with representing forests F_i. We define $P := P_1 + \ldots + P_k$, $Q := Q_1 + \ldots + Q_k$, and $F := F_1 \cup \cdots \cup F_k$. Then we have:*

(a) The measure Q is simple and F is its representing $\perp$-forest.

(b) For all base measures $\mathfrak{a} \leq P$ there exists exactly one i with $\mathfrak{a} \leq P_i$.

(c) If $\mathcal{A}$ is P_i-subadditive for all $i \leq k$, then $\mathcal{A}$ is P-subadditive.

(d) if Q_i is P_i-adapted for all $i \leq k$, then Q is adapted to P.

Proof of Lemma 41: *(a).* Since $Q_i \leq P_i \leq P$ we have $\mathbb{G}F_i = \operatorname{supp} Q_i \subset \operatorname{supp} P_i$. By the monotonicity of $\perp$ we then obtain $\mathbb{G}F_i \perp \mathbb{G}F_j$ for $i \neq j$. From this we obtain the assertion.
(b). Let $\mathfrak{a} \leq P$ be a base measure on $A \in \mathcal{A}$. Then we have $A = \operatorname{supp} \mathfrak{a} \subset \operatorname{supp} P = \bigcup_i \operatorname{supp} P_i$. By $\mathcal{A}$-connectedness there thus exists a i with $A \subset \operatorname{supp} P_i$. For $B \in \mathcal{B}$ we then find $\mathfrak{a}(B) = \mathfrak{a}(B \cap \operatorname{supp} P_i) \leq P(B \cap \operatorname{supp} P_i) = P_i(B \cap \operatorname{supp} P_i) = P_i(B)$. Moreover, for $j \neq i$ we have $\mathfrak{a}(A) > 0$ and $P_j(A) = 0$ and thus i is unique.
(c). Let $\mathfrak{a}, \mathfrak{a}' \leq P$ be base measures on base sets A, A' with $A \infty_P A'$. Since $A \perp A'$ implies $A \perp_\emptyset A'$, we have $A \infty A'$. By (b) we find unique indices i, i' with $\mathfrak{a} \leq P_i$ and $\mathfrak{a}' \leq P_{i'}$. This implies $A \subset \operatorname{supp} P_i$ and $A' \subset \operatorname{supp} P_j$, and hence we have $\operatorname{supp} P_i \infty \operatorname{supp} P_{i'}$ by monotonicity. This gives $i = i'$, i.e. $\mathfrak{a}, \mathfrak{a}' \leq P_i$. Since $\mathcal{A}$ is P_i-subadditive there now is an $\tilde{\mathfrak{a}} \in \mathcal{Q}_{P_i}(A \cup A')$ with $\tilde{\mathfrak{a}} \geq \mathfrak{a}$ or $\tilde{\mathfrak{a}} \geq \mathfrak{a}'$, and since $\tilde{\mathfrak{a}} \leq P_i \leq P$ we obtain the assertion.

(d). From (b) we conclude $\mathcal{Q}_P(A_1 \cup A_2) = \emptyset$ for all roots $A_1 \in F_i$ and $A_2 \in F_j$ and all $i \neq j$. This can be used to infer the groundedness and fineness of Q from the groundedness and fineness of the Q_i. Now let $\mathfrak{a}, \mathfrak{a}' \leq P$ be the levels of some direct siblings $A, A' \in F$ in Q and $\mathfrak{b} \in \mathcal{Q}_P(A \cup A')$ be any base measure. By (b) there is a unique i with $\mathfrak{b} \leq P_i$, and hence $\mathfrak{a}, \mathfrak{a}' \leq P_i$ as well. Therefore Q inherits strict motivation from Q_i. ∎

Lemma 42 *Let $(\mathcal{A}, \mathcal{Q}, \perp)$ be a clustering base, $P \in \mathcal{M}$, $\mathfrak{a}$ be a base measure on $A \in \mathcal{A}$ with $\operatorname{supp} P \subset A$, and $Q \leq P$ be a simple measure with representing forest F. We define $P' := \mathfrak{a} + P$, $Q' := \mathfrak{a} + Q$, and $F' := \{A\} \cup F$. Then the following statements hold:*

(a) The measure Q' is simple and F' is its representing $\perp$-forest.

(b) Let $\mathfrak{a}' \leq P'$ be a base measure on A'. Then either $\mathfrak{a}' \leq \mathfrak{a}$ or there is an $\alpha \in (0,1)$ such that $\mathfrak{a}'(\cdot \cap A') = \mathfrak{a}(\cdot \cap A') + \alpha\mathfrak{a}'(\cdot \cap A')$.

(c) If $\mathcal{A}$ is P-subadditive then $\mathcal{A}$ is P'-subadditive.

(d) If Q is P-adapted, then Q' is P'-adapted.

Proof of Lemma 42: *(a).* We have $\mathbb{G}F = \operatorname{supp} Q \subset \operatorname{supp} P \subset A$ and hence F' is a $\perp$-forest, which is obviously representing Q.
(b). Let us assume that $\mathfrak{a}' \not\leq \mathfrak{a}$, i.e. there is a $C_0 \in \mathcal{B}$ with $\mathfrak{a}'(C_0) > \mathfrak{a}(C_0)$ and thus we find $\mathfrak{a}'(C_0 \cap A') = \mathfrak{a}'(C_0) > \mathfrak{a}(C_0) \geq \mathfrak{a}(C_0 \cap A')$. In addition, we have $A' = \operatorname{supp} \mathfrak{a}' \subset \operatorname{supp} \mathfrak{a} = A$, and therefore Lemma 33 shows $\mathfrak{a}(\cdot \cap A') = \gamma\mathfrak{a}'(\cdot \cap A')$, where $\gamma := \frac{\mathfrak{a}(C_0 \cap A')}{\mathfrak{a}'(C_0 \cap A')} < 1$. Setting $\alpha := 1 - \gamma$ yields the assertion.
(c). Let $\mathfrak{a}_1, \mathfrak{a}_2 \leq P'$ be base measures on sets $A_1, A_2 \in \mathcal{A}$ with $A_1 \not\perp_{P'} A_2$. Since $\operatorname{supp} P' = A$, we have $A_1 \cup A_2 \subset A$, and thus $\mathfrak{a} \in \mathcal{Q}_{P'}(A_1 \cup A_2)$. Clearly, if $\mathfrak{a} \geq \mathfrak{a}_1$ or $\mathfrak{a} \geq \mathfrak{a}_2$, there is nothing left to prove, and hence we assume $\mathfrak{a}_1 \not\leq \mathfrak{a}$ and $\mathfrak{a}_2 \not\leq \mathfrak{a}$. Then (b) gives $\alpha_i \in (0,1)$ with $\mathfrak{a}_i(\cdot \cap A_i) = \mathfrak{a}(\cdot \cap A_i) + \alpha_i\mathfrak{a}_i(\cdot \cap A_i)$. We conclude that $\mathfrak{a}(\cdot \cap A_i) + \alpha_i\mathfrak{a}_i(\cdot \cap A_i) = \mathfrak{a}_i(\cdot \cap A_i) \leq P'(\cdot \cap A_i) = \mathfrak{a}(\cdot \cap A_i) + P(\cdot \cap A_i)$, and thus $\alpha_i\mathfrak{a}_i = \alpha_i\mathfrak{a}_i(\cdot \cap A_i) \leq P(\cdot \cap A_i) \leq P$. Since $\mathcal{A}$ is P-subadditive, we thus find an $\tilde{\mathfrak{a}} \in \mathcal{Q}_P(A_1 \cup A_2)$ with say $\tilde{\mathfrak{a}} \geq \alpha_1\mathfrak{a}_1$. For $\tilde{A} := \operatorname{supp} \tilde{\mathfrak{a}}$ we then have

$$\tilde{\mathfrak{a}}' := \mathfrak{a}(\cdot \cap \tilde{A}) + \tilde{\mathfrak{a}}(\cdot \cap \tilde{A}) \geq \mathfrak{a}(\cdot \cap \tilde{A}) + \alpha_1\mathfrak{a}_1(\cdot \cap \tilde{A}) \geq \mathfrak{a}(\cdot \cap A_1) + \alpha_1\mathfrak{a}_1(\cdot \cap A_1) = \mathfrak{a}_1\,,$$

where we used $\operatorname{supp} \mathfrak{a}_1 = A_1 \subset \tilde{A}$. Moreover $\tilde{A} = \operatorname{supp} \tilde{\mathfrak{a}} \subset \operatorname{supp} P \subset A$, together with flatness of $\mathcal{Q}$ shows that $\tilde{\mathfrak{a}}'$ is a base measure, and we also have $\tilde{\mathfrak{a}}' \leq \mathfrak{a} + \tilde{\mathfrak{a}} \leq \mathfrak{a} + P = P'$. Finally we observe that $A_1 \cup A_2 \subset \tilde{A} = \operatorname{supp} \tilde{\mathfrak{a}}'$, and hence $\tilde{\mathfrak{a}}' \in \mathcal{Q}_{P'}(A_1 \cup A_2)$.
(d). Clearly, F' is grounded because it is a tree. Now let $A_1, \ldots, A_k \in F'$, $k \geq 2$ be direct siblings and $\mathfrak{a}'_i$ be their levels in Q'. Since A is the only root it has no siblings, so for all i we have $A_i \in F$. Moreover, the levels $\mathfrak{a}_i$ of A_i in Q are P-motivated and P-fine since Q is P-adapted. Now let $\mathfrak{b} \in \mathcal{Q}_{P'}(A_1 \cup A_2)$ and $B := \operatorname{supp} \mathfrak{b}$.

To check that Q' is P'-fine, we first observe that in the case $\mathfrak{b} \leq \mathfrak{a}$ there is nothing to prove since $\mathfrak{a} \in \mathcal{Q}_{P'}(A_1 \cup \ldots \cup A_k)$ by construction. In the remaining case $\mathfrak{b} \not\leq \mathfrak{a}$ we find a $\beta > 0$ with $\mathfrak{b}(\cdot \cap B) = \mathfrak{a}(\cdot \cap B) + \beta\mathfrak{b}(\cdot \cap B)$ by (b), and by P-fineness of Q, there exists a $\tilde{\mathfrak{b}} \in \mathcal{Q}_P(A_1 \cup \ldots \cup A_k)$ with $\tilde{\mathfrak{b}} \geq \beta\mathfrak{b}$. Since $\operatorname{supp} \tilde{\mathfrak{b}} \subset \operatorname{supp} P \subset \operatorname{supp} \mathfrak{a}$ we see that

$\mathfrak{a}+\tilde{\mathfrak{b}}$ is a simple measure, and hence we can consider the level $\tilde{\mathfrak{b}}'$ of $\operatorname{supp}\tilde{\mathfrak{b}}$ in $\mathfrak{a}+\tilde{\mathfrak{b}}$. Since $\tilde{\mathfrak{b}}' \leq \mathfrak{a}+\tilde{\mathfrak{b}} \leq \mathfrak{a}+P \leq P'$, we then obtain $\tilde{\mathfrak{b}}' \in \mathcal{Q}_{P'}(A_1 \cup \ldots \cup A_k)$ and for $C \in \mathcal{B}$ we also have

$$\mathfrak{b}(C) = \mathfrak{b}(C \cap B) = \mathfrak{a}(C \cap B) + \beta\mathfrak{b}(C \cap B) \leq \mathfrak{a}(C \cap B) + \tilde{\mathfrak{b}}(C \cap B) = \tilde{\mathfrak{b}}'(C \cap B) \leq \tilde{\mathfrak{b}}'(C).$$

To check that Q' is strictly P'-motivated we fix the constant $\alpha \in (0,1)$ appearing in the strict P-motivation of Q. Then there are $\tilde{\alpha}_i \in (0,1)$ such that $\mathfrak{a}(\cdot \cap A_i) + \alpha\mathfrak{a}_i = \tilde{\alpha}_i\mathfrak{a}'_i$. We set $\tilde{\alpha} := \max\{\tilde{\alpha}_1, \tilde{\alpha}_2\} \in (0,1)$ and obtain $\mathfrak{a}(\cdot \cap A_i) + \alpha\mathfrak{a}_i \leq \tilde{\alpha}\mathfrak{a}'_i$ for both $i = 1, 2$. Let us first consider the case $\mathfrak{b} \leq \mathfrak{a}$. Since our construction yields $\mathfrak{a}'_i = \mathfrak{a}(\cdot \cap A_i) + \tilde{\mathfrak{a}}_i \not\leq \mathfrak{a}$, there is a $C_0 \in \mathcal{B}$ with $\mathfrak{a}'_i(C_0) > \mathfrak{a}(C_0)$. This implies $\tilde{\alpha}\mathfrak{a}'_i(C_0) \geq \mathfrak{a}(C_0 \cap A_i) + \alpha\mathfrak{a}_i(C_0) > \mathfrak{a}(C_0 \cap A_i) \geq \mathfrak{b}(C_0 \cap A_i)$, i.e. $\mathfrak{b} \not\geq \tilde{\alpha}\mathfrak{a}'_i$. Consequently, it remains to consider the case $\mathfrak{b} \not\leq \mathfrak{a}$. By (b) and $\operatorname{supp}\mathfrak{b} \subset \operatorname{supp} P' = A$ there is a $\beta \in (0,1]$ with $\mathfrak{b}(\cdot \cap B) = \mathfrak{a}(\cdot \cap B) + \beta\mathfrak{b}(\cdot \cap B)$. Then

$$\beta\mathfrak{b} = \beta\mathfrak{b}(\cdot \cap B) = \mathfrak{b}(\cdot \cap B) - \mathfrak{a}(\cdot \cap B) \leq P'(\cdot \cap B) - \mathfrak{a}(\cdot \cap B) = P(\cdot \cap B) \leq P\,,$$

and since $\beta\mathfrak{b} \in \mathcal{Q}_P(A_1 \cup A_2)$ we obtain $\beta\mathfrak{b} \not\geq \alpha\mathfrak{a}_i$ for $i = 1, 2$. Hence there is an event $C_0 \subset \operatorname{supp}\mathfrak{b}$ with $\beta\mathfrak{b}(C_0) < \alpha\mathfrak{a}_i(C_0)$, which yields $\mathfrak{b}(C_0 \cap A_i) = \mathfrak{a}(C_0 \cap A_i \cap B) + \beta\mathfrak{b}(C_0 \cap A_i) < \mathfrak{a}(C_0 \cap A) + \alpha\mathfrak{a}_i(C_0 \cap A_i) \leq \tilde{\alpha}\mathfrak{a}'_i(C_0 \cap A_i)$, i.e. $\mathfrak{b} \not\geq \tilde{\alpha}\mathfrak{a}'_i$. ■

Proof of Theorem 21: For a $P \in \bar{\mathcal{S}}(\mathcal{A})$ and a P-adapted isomonotone sequence $(Q_n, F_n) \nearrow P$ we define $c_{\mathcal{A}}(P) :=_P \lim_{n\to\infty} s(F_n)$, which is possible by Theorem 20. By Proposition 19 we then now that $c_{\mathcal{A}}(Q) = c(Q)$ for all $Q \in \mathcal{Q}$, and hence $c_{\mathcal{A}}$ satisfies the Axiom of BaseMeasureClustering. Furthermore, $c_{\mathcal{A}}$ is obviously structured and scale-invariant, and continuity follows from Theorem 20.

To check that $c_{\mathcal{A}}$ is disjoint-additive, we fix $P_1, \ldots, P_k \in \mathcal{P}_{\mathcal{A}}$ with pairwise $\perp$-disjoint supports and let $(Q_n^i, F_n^i) \nearrow P_i$ be P_i-adapted isomonotone sequences of simple measures. We set $Q_n := Q_n^1 + \cdots + Q_n^k$ and $P := P_1 + \ldots + P_k$. By Lemma 41 Q_n is simple on $F_n := F_n^1 \cup \cdots \cup F_n^k$ and P-adapted, and $\mathcal{A}$ is P-subadditive. Moreover, we have $Q_n \nearrow P$ and $s(F_n) = \bigcup_i s(F_n^i)$ inherits monotonicity as well. Therefore $(Q_n, F_n) \nearrow P$ is P-adapted and $\lim s(F_n) = \bigcup_i \lim s(F_n^i)$ implies disjoint-additive.

To check BaseAdditivity we fix a $P \in \mathcal{P}_{\mathcal{A}}$ and a base measure $\mathfrak{a}$ with $\operatorname{supp} P \subset \mathfrak{a}$. Moreover, let $(Q_n, F_n) \nearrow P$ be a P-adapted sequence. Let $Q'_n := \mathfrak{a} + Q_n$ and $P' := \mathfrak{a} + P$. Then by Lemma 42 Q'_n is simple on $F'_n := \{A\} \cup F_n$ and P'-adapted, and $\mathcal{A}$ is P'-subadditive. Furthermore we have $(Q'_n, F'_n) \nearrow P'$ and therefore we find $P' \in \mathcal{P}_{\mathcal{A}}$ and $\lim s(F'_n) = s(\{A\} \cup \lim s(F_n))$.

For the uniqueness we finally observe that Theorem 8 together with the Axioms of Additivity shows equality on $\mathcal{S}(\mathcal{A})$ and the Axiom of Continuity in combination with Theorem 20 extends this equality to $\mathcal{P}_{\mathcal{A}}$. ■

5.2.3 Proof of Theorem 23

Lemma 43 *Let $\mu \in \mathcal{M}_\Omega^\infty$, and consider $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp)$.*

(a) If $A, A' \in \mathcal{A}$ with $A \subset A'$ μ-a.s. then $A \subset A'$.

(b) Let $P \in \mathcal{M}_\Omega$ such that $\mathcal{A}$ is P-subadditive and P has a μ-density f that is of $(\mathcal{A}, \mathcal{Q}, \perp)$-type with a dense subset Λ such that $s(F_{f,\Lambda})$ is finite. For all $\lambda \in \Lambda$ and all $A_1, \ldots, A_k \in \mathcal{A}$ with $A_1 \cup \ldots \cup A_k \subset \{f > \lambda\}$ μ-a.s. there is $B \in \mathcal{A}$ with $A_1 \cup \ldots \cup A_k \subset B$ pointwise and $B \subset \{f > \lambda\}$ μ-a.s.

Proof of Lemma 43: *(a).* Let $A, A' \in \mathcal{A}$ with $A \subset A'$ μ-a.s. and let $x \in A$. Now $B := A \setminus A'$ is relative open in A and if it is non-empty then $\mu(B) > 0$ since A is a support set. Since by assumption $\mu(B) = 0$ we have $B = \emptyset$.
(b). Since $H := \{f > \lambda\} \in \bar{\mathcal{A}}$ there is an increasing sequence $C_n \nearrow H$ of base sets. Let $d\mathfrak{b}_n := \lambda 1_{B_n} d\,\mu \in \mathcal{Q}_P$. For all $i \leq k$ eventually $B_n \not\perp_\mu A_i$, so there is a n s.t. B_n is connected to all of them. By P-subadditivity between $\mathfrak{b}_n$ and $\lambda 1_{A_1}\, d\mu, \ldots, \lambda 1_{A_k}\, d\mu$ there is $d\mathfrak{c} = \lambda' 1_C\, d\mu \in \mathcal{Q}_P$ that supports all of them and majorizes at least one of them. Hence $\lambda \leq \lambda'$ and thus $A_1 \cup \ldots \cup A_k \subset C \subset \{f > \lambda'\} \subset \{f > \lambda\}$ μ-a.s. By (a) we are finished. ■

Lemma 44 *Let f be a density of $(\mathcal{A}, \mathcal{Q}, \perp)$-type, set $P := f\,d\mu$ and assume $\mathcal{A}$ is P-subadditive and $F_{f,\Lambda}$ is a chain. For all $k \geq 0$ and all $n \in \mathbb{N}$ let $B_n = C_1 \cup \ldots \cup C_k$ be a (possibly empty) union of base sets $C_1, \ldots, C_k \in \mathcal{A}$ with $B_n \subset \{f > \lambda\}$ for all $\lambda \in \Lambda$. Then $P := f\,d\mu \in \mathcal{P}$ and there is $(Q_n, F_n) \nearrow P$ adapted where for all n F_n is a chain and $B_n \subset \min F_n$.*

Proof of Lemma 44: Let $(\lambda_n)_n \subset \Lambda$ be a dense countable subset with $\lambda_n < \rho$ and set $\Lambda_n := \{\lambda_1, \ldots, \lambda_n\}$, $\Lambda_\infty := \bigcup_n \Lambda_n$. Remark that $\max \Lambda_n < \rho$ for all n, $|\Lambda_n| = n$ and $\Lambda_1 \subset \Lambda_2 \subset \ldots$. For very n we enumerate the n elements of Λ_n by $\lambda(1,n) < \ldots < \lambda(n,n)$. For every $\lambda \in \Lambda_\infty$ we let $n_\lambda := \min\{n \mid \lambda \in \Lambda_n\} \in \mathbb{N}$.

Since f is of $(\mathcal{A}, \mathcal{Q}, \perp)$-type, $H(\lambda) := \{f > \lambda\} \in \bar{\mathcal{A}}$ for $\lambda \in \Lambda$. Therefore there is $A_{\lambda,n} \in \mathcal{A}$ s.t. $A_{\lambda,n} \uparrow H(\lambda)$, where $n \geq 0$. We would like to use these $A_{\lambda,n}$ to construct Q_n, but they need to be made compatible in order that $(Q_n, F_n)_n$ becomes isomonotone. Hence we construct by induction a family of sets $A(\lambda, n) \in \mathcal{A}$, $\lambda \in \Lambda_n$, $n \in \mathbb{N}$ with the following properties:

$$A_{\lambda,n} \cup A(\lambda(i+1,n),n) \cup A(\lambda, n-1) \cup B_n \subset A(\lambda,n) \subset H(\lambda) \dot\cup N(\lambda,n), \qquad \mu(N(\lambda,n)) = 0.$$

Here $A(\lambda(i+1,n),n)$ is thought as empty if $i = n$ and similarly $A(\lambda, n-1) = \emptyset$ if $n = 1$ or $\lambda \notin \Lambda_{n-1}$. All of these involved sets C are base sets with $C \subset H(\lambda)$ and hence by Lemma 43 there is such an $A(n,\lambda)$. Since $A_{\lambda,n} \nearrow_n H(\lambda)$ we then also have $A(\lambda, n_\lambda + n) \uparrow H(\lambda)$.

Now for all n consider the chain $F_n := \{A(\lambda,n) \mid \lambda \in \Lambda_n\} \subset \mathcal{A}$ and the simple measure Q_n on F_n given by:

$$h_n := \sum_{i=1}^{n} \big(\lambda(i,n) - \lambda(i-1,n)\big) \cdot 1_{A(\lambda(i,n),n)} = \sum_{\lambda \in \Lambda_n} \lambda \cdot 1_{A(\lambda,n) \setminus \bigcup_{\lambda' > \lambda} A(\lambda',n)} \qquad (\lambda(0,n) := 0)$$

Let $x \in B$. Let

$$\Lambda_n(x) := \{\lambda \in \Lambda_n \mid x \in A(\lambda,n)\}$$

Then $h_n(x) = \max \Lambda_n(x)$. And if $x \in A(\lambda,n)$ then $x \in A(\lambda, n+1)$ so $\Lambda_n(x) \subset \Lambda_{n+1}(x)$ and we have:

$$h_n(x) = \max \Lambda_n(x) \leq \max \Lambda_{n+1}(x) = h_{n+1}(x)$$

Furthermore if $\lambda \in \Lambda_n(x)$ then $x \in A(\lambda,n) \subset H(\lambda)$ implying $h(x) > \lambda$. Therefore $h_1 \leq h_2 \leq \cdots \leq h$.

On the other hand for all $\varepsilon > 0$, since Λ_∞ is dense, there is a n and $\lambda \in \Lambda_n$ with $h(x) - \varepsilon \leq \lambda < h(x))$. Then $x \in H(\lambda)$ and therefore for n big enough $x \in A(\lambda, n)$ and then:

$$h(x) \geq h_n(x) \geq \lambda \geq h(x) - \varepsilon.$$

This means $h_n(x) \uparrow h(x)$ for all $x \in B$ so we have $h_n \uparrow h$ pointwise and by monotone convergence $(Q_n, F_n) \uparrow P_0$. ∎

Proof of Theorem 23: Let f be a density as supposed and set $F := s(F_{f,\Lambda})$. By assumption F is finite. If $|F| = 1$ then $F_{f,\Lambda}$ is a chain and the Theorem follows from Lemma 44 using $B_n = \emptyset$, $n \in \mathbb{N}$, in the notation of the lemma. Hence we can now assume $|F| > 1$. We prove by induction over $|F|$ that $f\,d\mu \in \bar{\mathcal{S}}(\mathcal{A})$ and $c(f\,d\mu) =_\mu s(F_{f,\Lambda})$ and assume that this is true for all f' with level forests $|s(F_{f',\Lambda'}| < |F|$. For readability we first handle the case that F is not a tree.

Assume that F has two or more roots $A_1, \ldots, A_k$ with $k = k(0)$. Denote by $f_i := f\big|_{A_i}$ the corresponding densities, hence $f = f_1 + \ldots + f_k$, and set $F_i := s(F_{f_i,\Lambda}) = F\big|_{\subset A_i}$ and $P_i := f_i\,d\mu$. We cannot use DisjointAdditivity, because separation of the A_i does not imply separation of the supports. Hence we have to construct a P-adapted isomonotone sequence $(Q_n, F_n) \nearrow P$. Since $F = F_1 \dot\cup \ldots \dot\cup F_k$ we have $|F_i| < |F|$ and hence by induction assumption for all $i \leq k$ we have $c(P_i) = F_i$, and there is an isomonotone P_i-adapted sequence $(Q_{i,n}, F_{i,n}) \nearrow P_i$. For $Q_n := Q_{1,n} + \ldots + Q_{k,n}$ and $F_n := F_{1,n} \cup \ldots \cup F_{k,n}$ it is clear that $(Q_n, F_n) \nearrow P$ is isomonotone. Let $\mathfrak{b} \in \mathcal{Q}_P$ and $B := \operatorname{supp}\mathfrak{b}$. We show that this is ∞_μ-connected to exactly one A_i. There is $\beta > 0$ s.t. $d\mathfrak{b} = \beta 1_B d\mu$ and $\beta 1_B \leq f$ μ-a.s. Now let $\lambda \in \Lambda$ with $\lambda < \beta$ and $\lambda < \inf\left\{\, \lambda' \in \Lambda \mid k(\lambda') \neq k(0) \,\right\}$. Because for all $\lambda \in \Lambda$ also the closures of clusters are $\perp$-separated we have

$$B \subset \overline{H_f(\lambda)} = \overline{B_1(\lambda))} \overset{\perp}{\dot\cup} \ldots \overset{\perp}{\dot\cup} \overline{B_k(\lambda))}.$$

By connectedness there is a unique $i \leq k$ with $B \subset \overline{B_i(\lambda)}$ and by monotonicity $B \perp \overline{B_j(\lambda)}$ for all $i \neq j$. Since this holds for all $\lambda \in \Lambda$ small enough and Λ is dense, this means that B is ∞_μ-connected to exactly i. Using this, P-adaptedness of Q_n is inherited from P_i-adaptedness of $Q_{i,n}$. Therefore $P = \lim_n Q_n \in \mathcal{P}$ and $c(P) = F$.

Now assume that F is a tree. Since $|F| > 1$ there are direct children $A_1, \ldots, A_k$ of the root in the structured forest F with $k \geq 2$. Let $\rho := \inf\{\lambda \in \Lambda \mid k(\lambda) \neq 1\}$. Since F is a tree, $\rho > 0$. Let $f_0(\omega) := \min\{\rho, f(\omega)\}$ and $f'(\omega) := \max\{0, f(\omega) - \rho\}$ for all $\omega \in \Omega$, and set $dP_0 := f_0\,d\mu$ and $dP' := f'\,d\mu$. Then $P = P_0 + P'$ is split into a *podest* corresponding to the root and its chain and the density corresponding to the children. We set $\Lambda' := \left\{\, \lambda - \rho \mid \lambda \in \Lambda, \lambda > \rho \,\right\}$. Then $|F_{f',\Lambda'}| = |F| - 1$ and by induction assumption there is $(Q'_n, F'_n) \uparrow P'$ adapted. Set $B_n := \mathbb{G}F'_n$ and $B := \bigcup B_n$. Then by Lemma 44 there is $(Q_n, F_n) \nearrow P_0$ adapted, which is given by a density h_n.

Now there might be a gap $\varepsilon_n := \rho - \sup h_n > 0$. By construction $\varepsilon_n \to 0$ but to be precise we let

$$\tilde{Q}_n := Q'_n + \sum_{A \in \max F'_n} \varepsilon_n \cdot 1_A\,d\mu.$$

This is still a simple measure on F'_n and therefore $(Q_n + \tilde{Q}_n, F_n \cup F'_n) \nearrow P$. We have to show P-adapted:

Grounded: Is fulfilled, since we consider trees at the moment.

Fine: Let $C_1, \dots, C_k \in F_n \cup F_n'$ be direct siblings. Then $C_1, \dots, C_k \in F_n'$ because F_n is a chain. If they are contained in one of the roots of F_n' fineness is inherited from adaptedness of Q_n'. Else they are the roots of F_n'. Let $\mathfrak{a} = \alpha 1_A d\mu \in \mathcal{Q}_P$ be a basic measure that $\perp\!\!\!\perp_P$-intersects say C_1 and C_2. Then is clear that $\alpha \leq \rho$ and by P-subadditivity fineness is granted.

Motivated Let $C, C' \in F_n \cup F_n'$ be direct siblings. Then again $C, C' \in F_n'$. If they are contained in one of the roots of F_n' motivatedness is inherited from adaptedness of Q_n'. Else they are the roots of F_n'. Let $\mathfrak{a} = \alpha 1_A d\mu \in \mathcal{Q}_P$ be a base measure that supports $C_1 \cup C_2$. Again it is clear that $\alpha \leq \rho$ and hence it cannot majorize neither the level of C nor the one of C'. ■

Proof of Proposition 24: Since f is continuous, all $H_f(\lambda)$ are open and it is the disjoint union of its open connected components. We show any connected component contains at least one of the $\hat{x}_1, \dots, \hat{x}_k$. To this end let $\lambda_0 \geq 0$ and B_0 be a connected component of $H_f(\lambda_0)$ (then $B_0 \neq \emptyset$). Because Ω is compact, so is the closure $\bar{B}_0$, and hence the maximum of f on $\bar{B}_0$ is attained at some $y_0 \in \bar{B}_0$. Since there is $y_1 \in B_0$ we have $f(y_0) \geq f(y_1) > \lambda$ we have $y_0 \in H_f(\lambda)$. Now $H_f(\lambda)$ is an open set, so y_0 is an inner point of this open set, and we know $y_0 \in \bar{B}_0$, therefore $y_0 \in B_0$. Therefore $y_0 \in B_0$ is a local maximum.

Hence for all λ there are at most k components and f is of $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp_\emptyset)$-type. The generalized structure $\tilde{s}(F_f)$ is finite, since there are only k leaves.

Now, fix for the moment a local maximum $\hat{x}_i$. Since $\hat{x}_i$ is a local maximum, there is ε_0 s.t. $f(y) \leq f(\hat{x}_i)$ for all y with $d(y, \hat{x}_i) < \varepsilon_0$. For all $\varepsilon \in (0, \varepsilon_0)$ consider the sphere

$$S_\varepsilon(\lambda) := \big\{\, y \in \Omega \colon f(y) \geq \lambda \text{ and } d(y, \hat{x}_i) = \varepsilon \,\big\}.$$

Since Ω is compact and $S_\varepsilon(\lambda)$ is closed, it is also compact. So as $\lambda \uparrow f(\hat{x}_i)$ the $S_\varepsilon(\lambda)$ is a monotone decreasing sequence of compact sets. Assume that all $S_\varepsilon(\lambda)$ were non-empty: Let $y_n \in S_{\varepsilon_0/(n+1)}(\lambda)$ then $(y_n)_n$ is a sequence in the compact set $S_{\varepsilon_0/2}(\lambda)$, hence there would be a subsequence converging to some y_ε. This subsequence eventually is in every $S_{\varepsilon_0/(n+1)}$ and hence $y_\varepsilon \in \bigcap_{\lambda < f(\lambda)} S_\varepsilon(\lambda)$, so this would be non-empty. This means that $f(y_\varepsilon) \geq f(\hat{x}_i)$. On the other hand, since $\varepsilon < \varepsilon_0$ we have $f(y_\varepsilon) \geq f(\hat{x}_i)$. Therefore all y_ε are local maxima, yielding a contradiction to the assumption that there are only finitely many. Hence for all ε, $S_\varepsilon(\lambda) = \emptyset$ for all $\lambda \in (\lambda_\varepsilon, f(\hat{x}_i)$. From this follows, that all local maxima have from some point on their own leaf in F_f. Therefore there is a bijection $\psi \colon \{\hat{x}_1, \dots, \hat{x}_k\} \to \min c(P)$ s.t. $\hat{x}_i \in \psi(\hat{x}_i)$.

Lastly, we need to show that also the closures of the connected components are separated, to verify the conditions of Theorem 23. We are allowed to exclude a finite set of levels, in this case the levels $\lambda_1, \dots, \lambda_m$ at which $\lambda \mapsto k(\lambda) \in \mathbb{N}$ changes. Consider $0 \leq \lambda_0 < \lambda_1$ s.t. for all $\lambda \in (\lambda_0, \lambda_1)$ $k(\lambda)$ stays constant. Set $\tilde{\lambda} := \frac{\lambda_0 + \lambda}{2} \in (\lambda_0, \lambda)$. Now let A, A' be connected components of $H_f(\lambda)$ and let B, B' be the connected components of $H_f(\tilde{\lambda})$ with $A \subset B$ and $A' \subset B'$. First we show $\bar{A} \subset B$: let $y_0 \in \bar{A}$. Then there is $(y_n) \subset A$ with $y_n \to y_0$. Because f is continuous we have

$$\lambda < f(y_n) \to f(y_0) \geq \lambda > \tilde{\lambda}$$

and hence $y_0 \in B$. Similarly we have $\bar{A}' \subset B'$ and $B \perp_\emptyset B'$ implies $\bar{A} \perp_\emptyset \bar{A}'$. ■

5.3 Proofs for Section 4

Lemma 45 *Let A, A' be closed, non-empty, and (path-)connected. Then:*

$$A \cup A' \text{ is (path-)connected} \iff A \infty_\emptyset A'.$$

Therefore any finite or countable union $A_1 \cup \ldots \cup A_k$, $k \leq \infty$ of such sets is connected iff the graph induced by the intersection relation is connected.

Proof of Lemma 45: Topological connectivity means that $A \cup A'$ cannot be written as disjoint union of closed non-empty sets. Hence, if $A \cup A'$ is connected, then this union cannot be disjoint. On the other hand if $x \in A \cap A' \neq \emptyset$ and $A \cup A' = B \cup B'$ with non-empty closed sets then $x \in B$ or $x \in B'$. Say $x \in B$, then still B' has to intersect A or A', say $B' \cap A \neq \emptyset$. Then both B, B' intersect A and both $C := B \cap A$ and $C' := B' \cap A$ are closed and non-empty. But since $A = C \cup C'$ is connected there is $y \in C \cap C' \subset B \cap B'$ and therefore $B \cup B'$ is not a disjoint union.

For path-connectivity: If $x \in A \cap A' \neq \emptyset$ then for all $y \in A \cup A'$ there is a path connecting x to y, so $A \cup A'$ is path-connected. On the other hand, if $A \cup A'$ is path connected then for any $x \in A$ and $x' \in A'$ there is a continuous path $f\colon [0,1] \to A \cup A'$ connecting x to x'. Then $B := f^{-1}(A)$ and $B' := f^{-1}(A')$ are closed and non-empty, and $B \cup B' = [0,1]$. Since $[0,1]$ is topologically connected there is $y \in B \cap B'$ and so $f(y) \in A \cap A'$. ■

Proof of Example 1: Reflexivity and monotonicity are trivial for all the three relations. *Disjointness*: Stability is trivial and connectedness follows from Lemma 45 and from the observation:

$$A \subset B_1 \overset{\perp_\emptyset}{\cup} \ldots \overset{\perp_\emptyset}{\cup} B_k \qquad \Rightarrow \qquad A = (A \cap B_1) \overset{\perp_\emptyset}{\cup} \ldots \overset{\perp_\emptyset}{\cup} (A \cap B_k)$$

τ-separation: Connectedness follows from the definition of τ-connectedness. For stability let $A_n \uparrow_n A$ and $A_n \perp_\tau B$ for $n \in \mathbb{N}$ and observe

$$d(A,B) = \sup_{x \in A} d(x,B) = \sup_{n \in \mathbb{N}} \sup_{x \in A_n} d(x,B) = \sup_{n \in \mathbb{N}} d(A_n, B) \geq \tau.$$

Linear Separation: Connectedness follows from the condition on $\mathcal{A}$ since $A \subset B_1 \overset{\perp_\ell}{\cup} \ldots \overset{\perp_\ell}{\cup} B_k$ implies $A = A \cap B_1 \overset{\perp_\ell}{\cup} \ldots \overset{\perp_\ell}{\cup} A \cap B_k$. To prove stability let $A_n \uparrow_n A$ and $A_n \perp_\ell B$ for $n \in \mathbb{N}$. Observe that

$$v \mapsto \sup\{\alpha \in \mathbb{R} \mid \langle\, v \mid a \,\rangle \leq \alpha \forall a \in A\}$$

is continuous and the same holds for the upper bound for the α. Hence for each n and any vector $v \in H$ with $\langle\, v \mid v \,\rangle = 1$ there is a compact, possibly empty interval $I_n(v)$ of α fulfilling the separation along v. Since by assumption the unit sphere is compact so is the semi-direct product $I_n := \{(v,\alpha) \mid \alpha \in I_n(v)\}$. Since $I_n \neq \emptyset$ and $I_n \supset I_{n+1}$ is a monotone limit of non-empty compact sets, the limit $\bigcap_n I_n$ is non-empty. ■

Lemma 46 *Let $\mu \in \mathcal{M}_\Omega^\infty$. If $\mathcal{C} \subset \mathcal{K}(\mu)$ then $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C}) \subset \mathcal{K}(\mu)$.*

Proof of Lemma 46: Let $A = C_1 \cup \ldots \cup C_k \in \mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})$ then:

$$\operatorname{supp} 1_A \, d\mu = \operatorname{supp}(1_{C_1} + \cdots + 1_{C_k}) \, d\mu = C_1 \cup \ldots \cup C_k = A. \qquad \blacksquare$$

Lemma 47 *Let $\mathcal{C} \subset \mathcal{B}$ be a class of non-empty closed sets. We assume the following generalized stability: If $B \in \mathcal{B}$ and $A_1, \ldots, A_k \in \mathcal{C}$ form a connected subgraph of $\mathcal{G}_{\perp\!\!\!\perp}(\mathcal{C})$:*

$$A_i \perp\!\!\!\perp B \quad \forall i \leq k \implies A \perp\!\!\!\perp B.$$

Then $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})$ is $\perp\!\!\!\perp$-intersection additive. Furthermore the monotone closure $\overline{\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})}$ is

$$\bar{\mathbb{C}}_{\perp\!\!\!\perp}(\mathcal{C}) := \{\, C_1 \cup C_2 \cup \ldots \mid C_1, C_2, \ldots \in \mathcal{C} \text{ and the graph } \mathcal{G}_{\perp\!\!\!\perp}(\{C_1, C_2, \ldots\}) \text{ is connected} \}$$

Proof of Lemma 47: Let $A = C_1 \cup \ldots \cup C_n, A' = C'_1 \cup \ldots \cup C'_{n'} \in \mathbb{C}(\mathcal{C})$ with $A \circledcirc A'$. If for all $j \leq n'$ we had $C'_j \perp\!\!\!\perp A$ then by assumption $A' \perp\!\!\!\perp A$ and therefore there has to be $j \leq n'$ with $C'_j \circledcirc A$. By the same argument there then is $i \leq n$ with $C_i \circledcirc C_j$. Therefore the intersection graph on $C_1, \ldots, C_n, C'_1, \ldots, C'_{n'}$ is connected and

$$A \cup A' = C_1 \cup \ldots \cup C_n \cup C'_1 \cup \ldots \cup C'_{n'} \in \mathbb{C}(\mathcal{C}).$$

Let $B \in \overline{\mathbb{C}(\mathcal{C})}$ and $A_1, A_2, \ldots \in \mathbb{C}(\mathcal{C})$ with $A_n \uparrow B$. Then for all n we have $A_n = C_{n1} \cup \ldots \cup C_{nk(n)}$ with $C_{nj} \in \mathcal{C}$ and their intersection graph is connected. Since $A_n \subset A_{n+1}$ for all C_{nj} there is j' with $Cnj \subset C_{(n+1),j'}$ which even gives $C_{nj} \circledcirc C_{(n+1)j'}$. Hence, the family $\{C_{nj}\}_{n,j}$ being countable can be enumerated $\tilde{C}_1, \tilde{C}_2, \ldots$ s.t. for all m there is $i(m) < m$ with $C_m \circledcirc C_{i(m)}$. Therefore for all m, the intersection graph on $\tilde{C}_1, \ldots, \tilde{C}_m$ is connected and hence

$$\tilde{A}_m := \tilde{C}_1 \cup \ldots \cup \tilde{C}_m \in \mathbb{C}(\mathcal{C}).$$

And we see that $\bigcup_m \tilde{A}_m \in \bar{\mathbb{C}}(\mathcal{C})$ and therefore

$$B = \bigcup_n A_n = \bigcup_{nj} C_{nj} = \bigcup_m \tilde{C}_m \in \bar{\mathbb{C}}(\mathcal{C}).$$

Now let $B \in \bar{\mathbb{C}}(\mathcal{C})$ and $B = \bigcup_n C_n$ with $C_n \in \mathcal{C}$ and s.t. the intersection graph on $C_1, C_2, \ldots$ is connected. By Zorn's Lemma it has a spanning tree. Since there are at most countable many nodes, one can assume that this tree is locally countable and therefore there is an enumeration of the nodes $C_{n(1)}, C_{n(2)}, \ldots$ s.t. they form a connected subgraph for all m. Then the intersection graph on $C_{n(1)}, \ldots, C_{n(m)}$ is connected for all m and therefore $A_m := C_{n(1)} \cup \ldots \cup C_{n(m)} \in \mathbb{C}(\mathcal{C})$. $A_m \in \mathbb{C}_i(\mathcal{C}) \uparrow B$ is monotone and we have $B = \bigcup A_m \in \overline{\mathbb{C}_i(\mathcal{C})}$. $\blacksquare$

Proposition 48 *Let $\mathcal{C} \subset \mathcal{B}$ be a class of non-empty, closed events and $\perp$ a $\mathcal{C}$-separation relation. We assume the following generalized countable stability: If $B \in \mathcal{B}$ and $A_1, A_2, \ldots \in \mathcal{C}$ form a connected subgraph of $\mathcal{G}_\perp(\mathcal{C})$:*

$$A_n \perp B \quad \forall n \implies \bigcup_n A_n \perp B.$$

Then $\perp$ is a $\mathbb{C}_\perp(\mathcal{C})$-separation relation.

Proof of Proposition 48: Set $\tilde{\mathcal{A}} := \mathbb{C}_\perp$. The assumption assures $\tilde{\mathcal{A}}$-stability. We have to show $\tilde{\mathcal{A}}$-connectedness. So let $A \in \tilde{\mathcal{A}}$ and $B_1, \dots, B_k \in \mathcal{B}$ closed with:

$$A \subset B_1 \overset{\perp}{\cup} \dots \overset{\perp}{\cup} B_k.$$

By definition of $\mathbb{C}$ there are $C_1, \dots, C_n \in \mathcal{C}$ with $A = C_1 \cup \dots \cup C_n$ and s.t. the $\perp$-intersection graph on $\{C_1, \dots, C_n\}$ is connected. For all $j \leq n$ we have $C_j \subset A \subset B_1 \cup \dots \cup B_k$ and by $\mathcal{C}$-connectedness there is $i(j) \leq k$ with $C_j \subset B_{i(j)}$. Now, whenever $i(j) \neq i(j')$ since $B_{i(j)} \infty B_{i(j')}$ we have by monotonicity $C_j \infty C_{j'}$. So whenever there is an edge between C_j and $C_{j'}$ then $i(j) = i(j')$. This means that $i(\cdot)$ is constant on connected components of the graph, and hence on the whole graph. ■

Proposition 49 *Let $\mathcal{C} \subset \mathcal{B}$ be a class of non-empty, closed events and $\perp$ a $\mathcal{C}$-separation relation with the following alternative $\mathbb{C}_\perp(\mathcal{C})$-stability: For all $A_1, A_2, \dots \in \mathcal{C}$ and $B \in \mathcal{B}$:*

$$\mathcal{G}_{\perp\!\!\!\perp}(\{A_1, A_2, \dots\}) \text{ is connected and for all } n\colon A_n \perp B \implies \bigcup_n A_n \perp B. \tag{28}$$

Then $\perp$ is a $\mathbb{C}_\perp(\mathcal{C})$-separation relation and $\mathbb{C}_\perp(\mathcal{C})$ is $\perp$-intersection additive.

Assume furthermore $\perp\!\!\!\perp$ is a weaker relation ($B \perp A' \implies B \perp\!\!\!\perp A'$). Then $\perp$ is a $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})$-separation relation and $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})$ is $\perp\!\!\!\perp$-intersection additive.

Proof of Proposition 49: The first part is a corollary of Lemma 47 and Proposition 48. For the second part observe $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C}) \subset \mathbb{C}_\perp(\mathcal{C})$. hence $\perp$ is also a $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})$-separation relation. But now $\mathbb{C}_{\perp\!\!\!\perp}(\mathcal{C})$ is only $\perp\!\!\!\perp$-intersection additive. ■

Proof of Proposition 26: First if $A_n \uparrow B \in \bar{\mathcal{A}}$ then for all $x, x' \in B$ there is n with $x, x' \in A_n$ and since A_n is path-connected there is a path connecting x and x' in $A_n \subset B$, so they are connected also in B.

Let O be open and path-connected. Let $(A_n)_n \subset \mathcal{A}'$ be the subsequence of all $A \in \mathcal{A}'$ with $A \subset O$. Since O is open and $\mathcal{A}'$ a neighborhood base $O = \bigcup_n A_n$. Consider the graph on the $(A_n)_n$ given by the intersection relation. Then by Zorn's Lemma there is a spanning tree, and we can assume that it is locally at most countable. Therefore there is an enumeration $A'_1, A'_2, \dots$ such that $\{A'_1, \dots, A'_n\}$ is a connected sub-graph for all n. By intersection-additivity hence $\tilde{A}_n := A'_1 \cup \dots \cup A'_n \in \mathcal{A}$ and $\tilde{A}_n \uparrow O$. ■

Lemma 50 *Let $\mu \in \mathcal{M}_\Omega^\infty$ and assume there is a $B \in \mathcal{K}(\mu)$ with $dP = 1_B \, d\mu$. Assume that $(\mathcal{A}, \mathcal{Q}^{\mu,\mathcal{A}}, \perp_\mathcal{A})$ is a P-subadditive stable clustering base and $(Q_n, F_n) \uparrow P$ is adapted. Then $s(F_n) = \{A_1^n, \dots, A_k^n\}$ consists only of roots and can be ordered in such a way that $A_i^1 \subset A_i^2 \subset \dots$. The limit forest F_∞ then consists of the k pairwise $\perp_\mathcal{A}$-separated sets:*

$$B_i := \bigcup_{n \geq 1} A_i^n \,,$$

there is a μ-null set $N \in \mathcal{B}$ with

$$B = B_1 \overset{\perp_\mathcal{A}}{\cup} \dots \overset{\perp_\mathcal{A}}{\cup} B_k \overset{\perp_\emptyset}{\cup} N. \tag{29}$$

Proof of Lemma 50: Once we have shown that all $s(F_n)$ only consists of their roots, the rest is a direct consequence of the isomonotonicity, and the fact that there is a μ-null set N s.t.:

$$B = \operatorname{supp} P = N \overset{\perp_\emptyset}{\cup} \bigcup_n \operatorname{supp} Q_n = B_1 \overset{\perp_\mathcal{A}}{\cup} \ldots \overset{\perp_\mathcal{A}}{\cup} B_k \overset{\perp}{\cup} N.$$

Now let $A, A' \in F_n$ be direct siblings and denote by $\mathfrak{a} =, \mathfrak{a}' \leq P$ their levels in Q_n. Then there are $\alpha, \alpha' > 0$ with $\mathfrak{a} = \alpha 1_A \, d\mu$ and $\mathfrak{a}' = \alpha' 1_{A'} \, d\mu$. Now, $\mathfrak{a}, \mathfrak{a}' \leq P$ implies $\alpha 1_A, \alpha' 1_{A'} \leq 1_B$ (μ-a.s.) and hence $\alpha, \alpha' \leq 1$. Assume they have a common root $A_0 \in \max F_n$, i.e. $A \cup A' \subset A_0 \subset B$. Then $\alpha 1_A, \alpha' 1_{a'} \leq 1_{A_0} \leq 1_B$ (μ-a.s.) and hence they cannot be motivated. ■

Proof of Lemma 27: The Hausdorff-dimension is calculated in (Falconer, 1993, Corollary 2.4). Proposition 2.2 therein gives for all events $B \subset C$ and $B' \subset C'$:

$$\mathcal{H}^s(\varphi(B)) \leq c_2^s \mathcal{H}^s(B) \text{ and } \mathcal{H}^s(\varphi^{-1}(B')) \leq c_1^s \mathcal{H}^s(B').$$

We show that C' is a $\mathcal{H}^s$-support set. Let $B' \subset C'$ be any relatively open set and set $B := \varphi^{-1}(B') \subset C$. Then $B \subset C$ is open because φ is a homeomorphism. And since C is a support set we have $0 < \mathcal{H}^s(B) < \infty$. This gives

$$0 < \mathcal{H}^s(B) = \mathcal{H}^s(\varphi^{-1}(B')) \leq c_1^s \mathcal{H}^s(B') \text{ and } \mathcal{H}^s(B') = \mathcal{H}^s(\varphi(B)) \leq c_2^s \mathcal{H}^s(B) < \infty.$$

Therefore C' is a $\mathcal{H}^s$-support set. ■

Proof of Proposition 28: The proof is split into four steps: *(a).* We first show that for all $A \in \mathcal{A}$ there is a unique index $i(A)$ with $A \in \mathcal{A}^{i(A)}$. To this end, we fix an $A \in \mathcal{A}$. Then there is $i \leq m$ with $A \in \mathcal{A}^i$. Let $\mu \in \mathcal{Q}^i$ be the corresponding base measure with $\operatorname{supp} \mu = A$. Let $j \leq m$ and $\mu' \in \mathcal{Q}^j$ be another measure with $\operatorname{supp} \mu' = A$. Then $\mu(A) = 1$ and $\mu'(A) = 1$. If $j > i$ then by assumption $\mu \prec \mu'$ and this would give $\mu'(A) = 0$. If $j < i$ we have $\mu' \prec \mu$ and this would give $\mu(A) = 0$. So $i = j$.
(b). Next we show that for all $A, A' \in \mathcal{A}$ with $A \subset A'$ we have $i(A) \leq i(A')$. To this end we first observe that $A = A \cap A' = \operatorname{supp} Q_A \cap \operatorname{supp} Q_{A'}$. If we had $i > j$ then $Q_{A'} \in \mathcal{Q}^j \prec \mathcal{Q}^i \ni Q_A$ and since $Q_{A'}(A) \leq Q_{A'}(A') = 1 < \infty$ we would have $Q_A(A) = 0$. Therefore $i \leq j$.
(c). Now we show that $\perp$ is a stable $\mathcal{A}$-separation relation. Clearly, it suffices to show $\mathcal{A}$-stability and $\mathcal{A}$-connectedness. The former follows since $i(A_n)$ is monotone if $A_1 \subset A_2 \subset \ldots$ by (b) and hence eventually is constant. For the latter let $A \in \mathcal{A}^i$ and $B_1, \ldots, B_k \in \mathcal{B}$ closed with $A \subset B_1 \overset{\perp}{\cup} \ldots \overset{\perp}{\cup} B_k$. Then since $\perp$ is an $\mathcal{A}^i$-separation relation there is $j \leq k$ with $A \subset B_j$.
(d). Finally, we show that $(\mathcal{A}, \mathcal{Q}, \perp)$ is a stable clustering base. To this end observe that fittedness is inherited from the individual clustering bases. Let $A \in \mathcal{A}^i$ and $A' \in \mathcal{A}^j$ with $A \subset A'$. Then $i \leq j$ by (b). If $i = j$ then flatness follows from flatness of $\mathcal{A}^i$. If $i < j$ then by assumption $Q_A \prec Q_{A'}$ and because $Q_A(A) = 1 < \infty$ we have $Q_{A'}(A) = 0$. ■

Proof of Proposition 29: *(a).* Let $\mathfrak{a} \leq P$ be a base measure on $A \in \mathcal{A}^i$. If $i = 1$ then $Q_A(A \cap \operatorname{supp} P_2) \leq Q_A(A) = 1$ and by $\mathcal{A}^1 \prec P_2$ we have $Q_A \prec P_2$ and hence $P_2(A \cap \operatorname{supp} P_2) = P_2(A) = 0$. Now for all events $C \in A^c$ therefore $\mathfrak{a}(C) = 0 \leq P_1(C)$ and for all $C \subset A$:

$$\mathfrak{a}(C) \leq P(C) = \alpha_1 P_1(C) + \alpha_2 P_2(C) = \alpha_1 P_1(C).$$

Now if $i = 2$ then by assumption $P_1 \prec \mathfrak{a}$ and since $0 < P_1(A \cap \operatorname{supp} P_1) < \infty$ we therefore have $\mathfrak{a}(A \cap \operatorname{supp} P_1) \leq \mathfrak{a}(\operatorname{supp} P_1) = 0$ and for all events $C \subset \Omega \setminus \operatorname{supp} P_1$ we have $\mathfrak{a}(C) \leq P(C) = \alpha_2 P_2(C)$ and for all events $C \subset \operatorname{supp} P_1$:

$$\mathfrak{a}(C) \leq \mathfrak{a}(\operatorname{supp} P_1) = 0 \leq P_1(C).$$

(b). Let $\mathfrak{a}, \mathfrak{a}' \leq P$ be base measures on $A \in \mathcal{A}^i$ and $A \in \mathcal{A}^j$ with $A \infty_{\mathcal{A}} A'$. By the previous statement we then already have $\mathfrak{a} \leq \alpha_i P_i$ and $\mathfrak{a}' \leq \alpha_j P_j$. Now, if $i = j$ then by P_i-subadditivity of $\mathcal{A}^i$ there is a base measure $\mathfrak{b} \leq P_i \leq P$ on $B \in \mathcal{A}^i$ with $B \supset A \cup A'$.

Now if $i \neq j$ consider say $i = 2$ and $j = 1$. Since $A \cap \operatorname{supp} P_2 \supset A \cap A' \neq \emptyset$ by assumption $\mathfrak{a}$ can be majorized by a base measure $\tilde{\mathfrak{a}} \leq P_2$ on $\tilde{A} \in \mathcal{A}^2$ with $\operatorname{supp} P_1 \subset \tilde{A}$ and $\tilde{\mathfrak{a}} \geq \mathfrak{a}$. The latter also gives $A \subset \tilde{A}$ and hence $\tilde{a}$ supports A and $\operatorname{supp} P_1 \supset \operatorname{supp} \mathfrak{a}'$ and $\tilde{\mathfrak{a}} \geq \mathfrak{a}$. ■

Acknowledgments

This work has been supported by DFG Grant STE 1074/2-1. We thank the reviewers and editors for their helpful comments.

Appendix A. Appendix: Measure and Integration Theoretic Tools

Throughout this subsection, Ω is a Hausdorff space and $\mathcal{B}$ is its Borel σ-algebra. Recall that a measure μ on $\mathcal{B}$ is inner regular iff for all $A \in \mathcal{B}$ we have

$$\mu(A) = \sup \left\{ \mu(K) \mid K \subset A \text{ is compact} \right\}.$$

A Radon space is a topological space such that all finite measures are inner regular. Cohn (2013, Theorem 8.6.14) gives several examples of such spaces such as *a)* Polish spaces, i.e. separable spaces whose topology can be described by a complete metric, *b)* open and closed subsets of Polish spaces, and *c)* Banach spaces equipped with their weak topology. In particular all separable Banach spaces equipped with their norm topology are Polish spaces and infinite dimensional spaces equipped with the weak topology are not Polish spaces but still they are Radon spaces. Furthermore Hausdorff measures, which are considered in Section 4.3, are inner regular (Federer, 1969, Cor. 2.10.23). For any inner regular measure μ we define the **support** by

$$\operatorname{supp} \mu := \Omega \setminus \bigcup \left\{ O \subset \Omega \mid O \text{ is open and } \mu(O) = 0 \right\}.$$

By definition the support is closed and hence measurable. The following lemma collects some more basic facts about the support that are used throughout this paper.

Lemma 51 *Let μ be an inner regular measure and $A \in \mathcal{B}$. Then we have:*

(a) If $A \perp_{\emptyset} \operatorname{supp} \mu$, then we have $\mu(A) = 0$.

(b) If $\emptyset \neq A \subset \operatorname{supp} \mu$ is relatively open in $\operatorname{supp} \mu$, then $\mu(A) > 0$.

(c) If μ' is another inner regular measures and $\alpha, \alpha' > 0$ then

$$\operatorname{supp}(\alpha\mu + \alpha'\mu') = \operatorname{supp}(\mu) \cup \operatorname{supp}(\mu')$$

(d) The restriction $\mu_{|A}$ of μ to A defined by $\mu_{|A}(B) = \mu(B \cap A)$ is an inner regular measure and $\operatorname{supp} \mu_{|A} \subset \overline{A \cap \operatorname{supp} \mu}$.

If μ is not inner regular, (d) also holds provided that Ω is a Radon space and $\mu(A) < \infty$.

Proof of Lemma 51: *(a).* We show that $A := \Omega \setminus \operatorname{supp} \mu$ is a μ-null set. Let $K \subset A$ be any compact set. By definition A is the union of all open sets $O \subset \Omega$ with $\mu(O) = 0$. So those sets form an open cover of A and therefore of K. Since K is compact there exists a finite sub-cover $\{O_1, \ldots, O_n\}$ of K. By σ-subadditivity of μ we find

$$\mu(K) \leq \sum_{i=1}^{n} \mu(O_i) = 0,$$

and since this holds for all such compact $K \subset A$ we have by inner regularity

$$\mu(A) = \sup_{K \subset A} \mu(K) = 0.$$

(b). By assumption there an open $O \subset \Omega$ with $\emptyset \neq A = O \cap \operatorname{supp} \mu$. Now $O \cap \operatorname{supp} \mu \neq \emptyset$ implies $\mu(O) > 0$. Moreover, we have the partition $O = A \cup (O \setminus \operatorname{supp} \mu)$ and since $O \setminus \operatorname{supp} \mu$ is open, we know $\mu(O \setminus \operatorname{supp} \mu) = 0$, and hence we conclude that $\mu(O) = \mu(A)$.
(c). This follows from the fact that for all open $O \subset \Omega$ we have

$$(\alpha\mu + \alpha'\mu')(O) = \alpha\mu(O) + \alpha'\mu'(O) = 0 \iff \mu(O) = 0 \text{ and } \mu'(O) = 0.$$

(d). The measure $\mu_{|A}$ is inner regular since for $B \in \mathcal{B}$ we have

$$\begin{aligned} \mu'(B) = \sup\left\{ \mu(K') \mid K' \subset B \cap A \text{ is compact} \right\} &\leq \sup\left\{ \mu'(K') \mid K' \subset B \text{ is compact} \right\} \\ &\leq \mu'(B). \end{aligned}$$

Now observe that $X \setminus \overline{A \cap \operatorname{supp} \mu} \subset X \setminus (A \cap \operatorname{supp} \mu) = (X \setminus A) \cup (X \setminus \operatorname{supp} \mu)$. For the open set $O := X \setminus \overline{A \cap \operatorname{supp} \mu}$ we thus find

$$\mu_{|A}(O) \leq \mu_{|A}(X \setminus A) + \mu_{|A}(X \setminus \operatorname{supp} \mu) \leq \mu(X \setminus \operatorname{supp} \mu) = 0. \qquad \blacksquare$$

Lemma 52 *Let Q, Q' be σ-finite measures.*

(a) If Q and Q' have densities h, h' with respect to some measure μ then

$$Q \leq Q' \iff h \leq h' \quad \mu\text{-a.s.}$$

(b) If $Q \leq Q'$ then Q is absolutely continuous with respect to Q', i.e. Q has a density function h with respect to Q', $dQ = h\,dQ'$ such that:

$$h(x) = \begin{cases} \in [0,1] & \text{if } x \in \operatorname{supp} Q' \\ 0 & \text{else} \end{cases}$$

Proof of Lemma 52: *(a).* "'$\Leftarrow$"' a direct calculation gives

$$Q(B) = \int_B h\,d\mu \leq \int_B h'\,d\mu = Q'(B).$$

and monotonicity of the integral.
For "'$\Rightarrow$"' assume $\mu(\{x : h(x) > h'(x)\}) > 0$, then

$$\int_{h>h'} h d\mu = Q(\{h > h'\}) \leq Q'(\{h > h'\}) = \int_{h>h'} h' d\mu < \int_{h>h'} h d\mu,$$

where the last inequality holds since we assume $\mu(\{x : h(x) > h'(x)\} > 0$ and again the monotonicity of the integral. Through this contradiction implies the statement.
(b). $Q \leq Q'$ means every Q'-null set is a Q-null set. Furthermore since Q' is σ-finite Q is σ-finite as well. So we can use Radon-Nikodym theorem and there is a $h \geq 0$ s.t. $dQ = h\,dQ'$. Since the complement of $\operatorname{supp} Q'$ is a Q'-null set, we can assume $h(x) = 0$ on this complement.

We have to show that $h \leq 1$ a.s. Let

$$E_n := \{h \geq 1 + \tfrac{1}{n}\} \qquad \text{and} \qquad E := \{h > 1\}.$$

Then $E_n \uparrow E$ and we have

$$Q'(E_n) \geq Q(E_n) = \int_{E_n} h\,dQ' \geq (1 + \tfrac{1}{n}) \cdot Q'(E_n),$$

which implies $Q'(E_n) = 0$ for all n. Therefore $Q'(E) = \lim_n Q'(E_n) = 0$. ■

Lemma 53 *(a) Let $Q_n \uparrow P$, $A := \operatorname{supp} P$ and $B := \bigcup_n \operatorname{supp} Q_n$. Then $B \subset A$ and $P(B \setminus A) = 0$.*

(b) Assume Q is a finite measure and $Q_1 \leq Q_2 \leq \ldots \leq Q$ and let the densities $h_n := \frac{dQ_n}{dQ}$. Then $h_1 \leq h_2 \leq \ldots \leq 1$ Q-a.s. Furthermore, the following are equivalent:

(i) $Q_n \uparrow Q$
(ii) $h_n \uparrow 1$ Q-a.s.
(iii) $h_n \uparrow 1$ in L^1.

Proof of Lemma 53:

(a) Since $Q_n \leq P$ we have $\operatorname{supp} Q_n \subset A$ and therefore $B \subset A$. Because of $(A \setminus B) \cap \operatorname{supp} Q_n = \emptyset$ and the convergence we have for all n

$$P(A \setminus B) = \lim_{n\to\infty} Q_n(A \setminus B) = 0.$$

(b) By the previous lemma we have $h_1 \leq h_2 \leq \cdot \leq 1$ Q-a.s.

(i) $\Rightarrow$ (ii): Since $(h_n)_n$ is monotone Q-a.s. it converges Q-a.s. to a limit $h \leq 1$. Let

$$E_n := \{h \leq 1 - \tfrac{1}{n}\} \qquad \text{and} \qquad E := \{h < 1\}.$$

Then $E_n \uparrow E$ and we have by the monotone convergence theorem:

$$Q_m(E_n) = \int_{E_n} h_m \, dQ \xrightarrow[m\to\infty]{} \int_{E_n} h \, dQ \leq (1 - \tfrac{1}{n}) \, Q(E_n)$$

But since $Q_m(E_n) \uparrow_m Q(E_n)$ this means that $Q(E_n) = 0$ for all n and therefore $Q(E) = \lim_n Q(E_n) = 0$.

(ii) $\Rightarrow$ (iii): This follows from monotone convergence, because $1 \in L^1(Q)$.

(iii) $\Rightarrow$ (i): For all $B \in \mathcal{B}$:

$$Q(B) - Q_n(B) = \int_B |1 - h_n| \, dQ \leq \int |1 - h_n| \, dQ \to 0$$

because of $h_n \to 1$ in L^1. ■

References

S. Ben-David. Computational Feasibility of Clustering under Clusterability Assumptions. *ArXiv e-prints*, January 2015.

J. E. Chacón. A population background for nonparametric density-based clustering. *ArXiv e-prints*, August 2014. URL `http://arxiv.org/abs/1408.1381`.

D. L. Cohn. *Measure Theory*. Birkhäuser, 2nd ed. edition, 2013.

W. Day and F. McMorris. *Axiomatic Consensus Theory in Group Choice and Biomathematics*. Society for Industrial and Applied Mathematics, 2003.

D. Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20 (4):364–366, 1977.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

W.E. Donath and A.J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, Sept 1973.

M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *International Conference on Knowledge Discovery and Data Mining*, pages 226–231. AAAI Press, 1996.

K. Falconer. *Fractal Geometry: Mathematical Foundations and Applications*. Wiley, 1993.

H. Federer. *Geometric Measure Theory*. Springer, 1969.

G. Gan, C. Ma, and J. Wu. *Data clustering. Theory, algorithms, and applications*. SIAM, 2007.

John A. Hartigan. *Clustering Algorithms*. Wiley, 1975.

M. F. Janowitz and R. Wille. On the classification of monotone-equivariant cluster methods. In Cox, Hansen, and Julesz, editors, *Partitioning Data Sets: DIMACS Workshop 1993*, pages 117–142. AMS, 1995.

N. Jardine and R. Sibson. *Mathematical Taxonomy*. Wiley, 1971.

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, 1990.

J. M. Kleinberg. An impossibility theorem for clustering. In Becker, Thrun, and Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 463–470. MIT Press, 2003.

J. Kogan. *Introduction to Clustering Large and High-Dimensional Data*. Cambridge University Press, 2007.

J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press, 1967.

M. Meilă. Comparing clusterings: An axiomatic view. In *International Conference on Machine Learning*, ICML '05, pages 577–584. ACM, 2005.

G. Menardi. A review on modal clustering. *International Statistical Review*, 2015.

B. Mirkin. *Clustering for Data Mining. A Data Recovery Approach*. Chapman & Hall/CRC, 2005.

B.G. Mirkin. On the problem of reconciling partitions. *Quantitative Sociology, International Perspectives on Mathematical and Statistical Modelling*, pages 441–449, 1975.

B.G. Mirkin. Additive clustering and qualitative factor analysis methods for similarity matrices. *Psychological Review*, 4(1):7–31, 1987.

J. Puzicha, T. Hofmann, and J. M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33:617–634, 1999.

A. Rinaldo and L. Wasserman. Generalized density clustering. *Ann. of Stat.*, 38(5):2678–2722, 2010.

R. N. Shepard and P. Arabie. Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, 86(2):87–123, 1979.

R. Sibson. SLINK: An optimally efficient algorithm for the single-link cluster method. *The Computer Journal*, 16(1):30–34, 1973.

W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification*, 20(1):025–047, 2003.

U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

U. von Luxburg and S. Ben-David. Towards a statistical theory of clustering. In *PASCAL workshop on Statistics and Optimization of Clustering*, 2005.

J. H. Ward, Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.

R. B. Zadeh and S. Ben-David. A uniqueness theorem for clustering. In Bilmes and Ng, editors, *Conference on Uncertainty in Artificial Intelligence, 2009*. AUAI Press, 2009.

Journal of Machine Learning Research 16 (2015) 2003-2022 Submitted 7/15; Published 9/15

Predicting a Switching Sequence of Graph Labelings

Mark Herbster M.HERBSTER@CS.UCL.AC.UK
Stephen Pasteris S.PASTERIS@CS.UCL.AC.UK
Massimiliano Pontil M.PONTIL@CS.UCL.AC.UK
Department of Computer Science
University College London
Gower Street, London WC1E 6BT, UK

Editor: Alex Gammerman and Vladimir Vovk

Abstract

We study the problem of predicting online the labeling of a graph. We consider a novel setting for this problem in which, in addition to observing vertices and labels on the graph, we also observe a sequence of just vertices on a second graph. A latent labeling of the second graph selects one of K labelings to be active on the first graph. We propose a polynomial time algorithm for online prediction in this setting and derive a mistake bound for the algorithm. The bound is controlled by the geometric cut of the observed and latent labelings, as well as the resistance diameters of the graphs. When specialized to multitask prediction and online switching problems the bound gives new and sharper results under certain conditions.

Keywords: online learning over graphs, kernel methods, matrix winnow, switching

1. Introduction

We consider the problem of learning online a set of K binary labelings of a graph. In a simple scenario this set of labelings corresponds to a switching sequence of labelings. Initially we focus on this setting before introducing our more general model. Consider the following game for predicting the labeling of a graph: *Nature* presents a graph; *nature* queries a vertex i_1; the *learner* predicts $\hat{y}_1 \in \{-1, 1\}$ as the label of the vertex; *nature* presents a label y_1; *nature* queries a vertex i_2; the *learner* predicts $\hat{y}_2$; and so forth. The learner's goal is to minimize the total number of mistakes $M = |\{t : \hat{y}_t \neq y_t\}|$. If nature is adversarial, the learner will always mispredict, but if nature is regular or simple, there is hope that a learner may make only a few mispredictions. Thus, a central goal of online learning is to design algorithms whose total mispredictions can be bounded relative to the complexity of nature's labeling.

To predict a *single* labeling of a graph, one may employ a kernel perceptron algorithm based on the graph Laplacian (Herbster and Pontil, 2006). This method achieves a bound of $M \leq \mathcal{O}(R\phi)$, where ϕ is the *cut* (the number of edges joining disagreeing labels) and R is the (resistance) diameter of the graph. Thus ϕ measures the complexity of the labeling and R is a structural parameter of the graph independent of the labeling. Such a bound is particularly appealing when the parameters are mildly dependent or independent of the number of vertices in the graph (see Herbster and Pontil, 2006, for a discussion).

In the switching setting, we now consider a sequence *colored* by K graph labelings with $S \geq K$ switches. We illustrate a switching sequence in Figure 1. In this color illustration

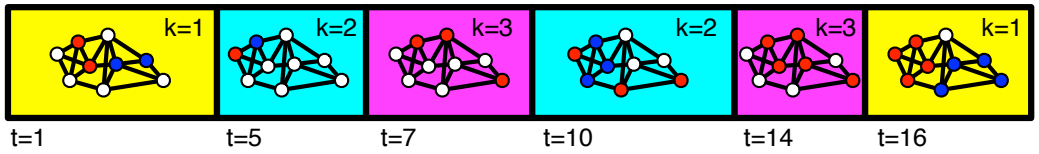


Figure 1: A switching sequence over 20 trials with K=3 graph labelings and S=5 switches.

there are $S = 5$ switches between $K = 3$ graph labelings. At each trial, a vertex of the graph is labeled according to one of the K binary functions. There are at most S trials at which the binary function currently in use is changed. In the specific example, the labeling 1 is used in trials 1–4 and 16–20, labeling 2 is used in trials 5–6 and 10–13, and labeling 3 is used in trials 7–9 and 14–15.

We will give an algorithm that achieves

$$M \leq \tilde{\mathcal{O}}\left(\left(S + R\sum_{k=1}^{K}\phi_k\right)K\log(n)\right), \tag{1}$$

where ϕ_k is the cut of the k-th binary labeling, n is the number of vertices in the graph, and the $\tilde{\mathcal{O}}(x)$ notation absorbs a polylogarithmic factor in x. Note that the term $R\sum_{k=1}^{K}\phi_k$ is the cost of learning the K binary labelings, given the information of which labeling is active on each trial. Since this information is not available to the learner, we pay a multiplicative term $K\log(n)$ and an additive term for the number of switches S. The particularly salient feature of this bound is that we pay the cost $R\sum_{k=1}^{K}\phi_k$ of learning all the binary labelings only once. This, and the fact that $S \geq K$, implies that the algorithm is maintaining an implicit *memory* of past graph labelings learned.

In the more general setting, the learner is given two graphs: an *observed* n-vertex graph $\mathcal{G}$ and a p-vertex *latent graph* $\mathcal{H}$. Hidden from the learner is a set $\{\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K\}$ of K binary labelings of $\mathcal{G}$. On each trial one of these labelings is *active*, the learner receives a pair of vertices, $i \in \mathcal{G}$ and $j \in \mathcal{H}$, and the learner's aim is to predict the currently active binary label of vertex i. It is the unknown K-ary label of j that determines the active labeling of $\mathcal{G}$ and hence the current label of i. After making its prediction the learner receives only the current label of i. The learner never receives the label of j. Note that if the learner did in fact receive the label of j, the learning problem would separate into K independent graph labeling tasks. Thus the graph $\mathcal{H}$ is called latent because the vertex labels of this graph are never observed, although it controls which of the K labelings of $\mathcal{G}$ is active at each given trial.

We propose a polynomial time algorithm for predicting the labelings of the observed graph and we derive a mistake bound for this algorithm. The bound involves two additive terms, which measure the complexity of the K binary labelings, and the complexity of the latent labeling, respectively; as well as a multiplicative term of the order of $K\log(K(n+p))$. Returning to the switching example, the latent graph can be thought of as a "line" graph,

where the sequence of vertices corresponds to the sequence of trials (although as we shall see in Section 6, for technical reasons we will need instead a binary support tree). The latent K-labeling function will then have a cut equal to S, the number of switches; and the bound (1) will be obtained as a special case of the general result described in this paper.

The paper is organized in the following manner. In Section 2, we comment about related work. In Section 3, we introduce the learning problem. In Section 4, we discuss the proposed learning algorithm. Section 5 presents our main result and details its proof. In Section 6, we illustrate our result in two specific examples and make final remarks.

2. Related Work

The problem of learning a labeling of a graph is a natural one in the online learning setting (Herbster et al., 2005; Herbster and Pontil, 2006), as well as a foundational technique for a variety of semi-supervised learning methods (Blum and Chawla, 2001; Kondor and Lafferty, 2002; Zhu et al., 2003). In the online setting, fast algorithms have been developed that operate on trees and path graphs (Herbster et al., 2008, 2009; Cesa-Bianchi et al., 2009, 2010; Vitale et al., 2011).

Our main application is to learning a switching sequence of graph labelings. Switching has been studied extensively in the online learning literature. The results divide largely into two directions: switching in the "experts" model (Herbster and Warmuth, 1998; Vovk, 1999; Bousquet and Warmuth, 2003; Gyorfi et al., 2005; Koolen and Rooij, 2008; Hazan and Seshadhri, 2009; Adamskiy et al., 2012; Cesa-Bianchi et al., 2012); and switching in online linear prediction model, see e.g. (Herbster and Warmuth, 2001; Kivinen et al., 2004; Cesa-Bianchi and Gentile, 2006). As we may view learning a graph labeling as learning a linear classifier based on a Laplacian kernel, our algorithm is directly comparable to these previous results. The implicit assumption of those switching techniques is that they learn a sequence of linear classifiers $w_1, w_2, \ldots$ and that this sequence is slowly changing over time, i.e, they are interested in predicting well when a *drifting* cost $\mathcal{O}(\sum_t \|w_t - w_{t+1}\|)$ is small. Our assumption is different: we consider that there exists a *small* set of K distinct classifiers, and we switch repeatedly between classifiers within this set. This setting is analogous to the setting proposed in an open problem by Freund (2000). Freund's challenge was to give an efficient algorithm in the expert advice model for the problem of switching repeatedly between a small set of experts within a larger set of experts. The problem was solved by Bousquet and Warmuth (2003) (see also Adamskiy et al., 2012). Those results, however, do not directly transfer to the graph labeling setting as the number of needed experts is 2^n for an n-vertex graph, and computing the marginal probabilities with a natural prior (i.e., an Ising distribution) on a graph even without switching is a well-known #P-complete problem (Goldberg and Jerrum, 2007).

An example of predicting in our more general setting applies to online multitask learning and is inspired by Cavallanti et al. (2010, Corollary 3). We adapt their model to our graph labeling set-up. Further related work includes (Dekel et al., 2007), which considered learning multiple tasks related through a joint loss function; and (Avishek et al., 2011), which generalized the usual setting to include negatively correlated tasks as well as positively correlated tasks. Rather than learning a group of interelated linear classifiers it is also

natural to consider multi-task learning with expert advice. Two prominent results include those of Abernethy et al. (2007) and Adamskiy et al. (2012).

Our main technical debt is to the following four papers. Firstly, the mistake bound analysis of matrix winnow (Warmuth, 2007), which strongly informs the proof of our main result. Secondly, our analysis of using matrix winnow on graphs is inspired by the graph Laplacian construction in (Gentile et al., 2013). Thirdly, our first two techniques require a modification of the Laplacian to ensure strict positive definiteness, and here we used the simple construction from (Herbster and Pontil, 2006). Finally we use the *binary support tree* construction (Herbster et al., 2008) to model the trial sequence in the switching setting.

3. Problem

In this section, we present the problem under study. We begin by introducing some graph terminology.

We are given two undirected graphs, an n-vertex graph $\mathcal{G}$ and a p-vertex graph $\mathcal{H}$. We let $\mathcal{V}(\mathcal{G})$ and $\mathcal{V}(\mathcal{H})$ be the set of vertices in $\mathcal{G}$ and $\mathcal{H}$, respectively, and let $\mathbf{L}_{\mathcal{G}}$ and $\mathbf{L}_{\mathcal{H}}$ be the corresponding graph Laplacians. For every positive integer d, we define $\mathbb{N}_d = \{1, \ldots, d\}$, the set of integers from 1 and up to including d. Unless confusion arises, for simplicity we identify vertices by their indices. Indices $i, i', i_t \in \mathbb{N}_n$ will always be associated with vertices in $\mathcal{G}$, and indices $j, j', j_t \in \mathbb{N}_p$ will be associated with vertices in $\mathcal{H}$.

A *labeling* of a graph is a function which maps vertices on the graph to a set of labels. We define the *cut* induced by a labeling of a graph as the number of edges whose end vertices have different labels. Note that this definition is independent of the number of labels used. We will use the notation $\mathrm{cut}_{\mathcal{G}}(\mathbf{u})$ to denote the cut associated with the labeling $\mathbf{u}$ of graph $\mathcal{G}$. In particular if $\mathbf{u}$ is a binary labelling with label set $\{-1, 1\}$ then $\mathrm{cut}_{\mathcal{G}}(\mathbf{u}) = \frac{1}{4}\mathbf{u}^{\top}\mathbf{L}_{\mathcal{G}}\mathbf{u}$.

In the paper we refer to $\mathcal{G}$ as the *observed graph* since during the learning process we will observe *both* a vertex of $\mathcal{G}$ and a corresponding label, whereas we refer to $\mathcal{H}$ as the *latent graph* because we will *only* observe a vertex of $\mathcal{H}$ but *never* observe the corresponding label. As we will see, the latent graph provides side information which can guide the prediction tasks on the observed graph. The goal is to predict well the binary labels associated to vertices in $\mathcal{G}$ using sequential information of the form $(i_t, j_t, y_t) \in \mathcal{V}(\mathcal{G}) \times \mathcal{V}(\mathcal{H}) \times \{-1, 1\}$ for $t = 1, 2, \ldots, T$; the true label y_t is determined by using one of the K binary classifiers, and which of these is active at each trial is determined by a K-class classifier which acts on the latent graph $\mathcal{H}$. Specifically, we let $\boldsymbol{\omega}_1, \ldots \boldsymbol{\omega}_K$ be the binary classifiers (labelings) on graph $\mathcal{G}$. Each $\boldsymbol{\omega}_k$ is a function from $\mathcal{V}(\mathcal{G})$ to $\{-1, 1\}$. The latent labeling controls which of the K labelings of $\mathcal{G}$ is currently active and it is given by a function $\boldsymbol{\mu} : \mathcal{V}(\mathcal{H}) \to \mathbb{N}_K$. In the paper, when confusion does not arise, we simply regard the functions $\boldsymbol{\omega}_k$ as vectors in $\{-1, 1\}^n$ and $\boldsymbol{\mu}$ as a vector in $\{1, \ldots, K\}^p$.

We consider the following online learning game between nature and learner. The learner knows the graphs $\mathcal{G}$ and $\mathcal{H}$ from the outset but does not initially know the labelings $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_K$, and as we already noted never observes the latent labeling $\boldsymbol{\mu}$. On trial t nature presents the learner with vertices $(i_t, j_t) \in \mathbb{N}_n \times \mathbb{N}_p$, the learner predicts a value $\hat{y}_t \in \{-1, 1\}$ and then the true label y_t is revealed to the learner. This label is computed by nature as $y_t = \omega_{\mu_{j_t}, i_t}$, that is the i_t-th component of the binary vector $\omega_{\mu_{j_t}}$. We define

Algorithm 1

Input: An n-vertex graph $\mathcal{G}$ and p-vertex graph $\mathcal{H}$.

Parameters: K, $\hat{\theta}$, η.

Initialization: $\boldsymbol{W}_0 \leftarrow \frac{\boldsymbol{I}}{K(n+p)}$, where $\boldsymbol{I}$ is the $(n+p)\times(n+p)$ identity matrix.

For $t = 1, \ldots, T$

- Get pair of vertices $i_t, j_t \in \mathcal{V}(\mathcal{G}) \times \mathcal{V}(\mathcal{H})$.
- Define the matrix $\boldsymbol{X}_t := \frac{1}{2}\boldsymbol{x}_t\boldsymbol{x}_t^T$, with $\boldsymbol{x}_t$ given by Equation (4).
- Predict
$$\hat{y}_t = \begin{cases} 1 & \text{if } \operatorname{Tr}(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) \geq \frac{K+1}{2K\hat{\theta}}, \\ -1 & \text{otherwise.} \end{cases}$$
- Receive label $y_t \in \{-1, 1\}$ and if $\hat{y}_t \neq y_t$ update
$$\boldsymbol{W}_t \leftarrow \exp\left(\log\left(\boldsymbol{W}_{t-1}\right) + \eta(y_t - \hat{y}_t)\boldsymbol{X}_t\right). \tag{2}$$

the set of mistakes as $\mathcal{M} := \{t : \hat{y}_t \neq y_t\}$ and the number of mistakes $M := |\mathcal{M}|$. The aim of the learner is for M to be small.

Before presenting the learning algorithm we require some more notation. Given a matrix $\boldsymbol{A}$ we define $\boldsymbol{A}^+$, $\boldsymbol{A}^T$ and $\operatorname{Tr}(\boldsymbol{A})$ to be its pseudoinverse, transpose and trace respectively. We let $\mathbf{S}^d$ be the set of $d\times d$ symmetric matrices and let $\mathbf{S}^d_+$ and $\mathbf{S}^d_{++}$ be the subset of positive semidefinite and strictly positive definite matrices. Recall that the set of symmetric matrices $\mathbf{S}^d_+$ has the following partial ordering: for every $\boldsymbol{A}, \boldsymbol{B} \in \mathbf{S}^d_+$ we say that $\boldsymbol{A} \preceq \boldsymbol{B}$ if and only if $\boldsymbol{B} - \boldsymbol{A} \in \mathbf{S}^d_+$. Every real valued function f induces a spectral function $f : \mathbf{S}^d \to \mathbf{S}^d$ which is obtained by applying f to the eigenvalues of A. Specifically, if $\{\lambda_i, \boldsymbol{u}_i\}_{i=1}^d$ is an eigensystem of A, that is, $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d$ are orthonormal vectors and λ_i are real numbers such that $\boldsymbol{A} = \sum_{i=1}^d \lambda_i \boldsymbol{u}_i\boldsymbol{u}_i^T$, then we define $f(A) = \sum_{i=1}^d f(\lambda_i)\boldsymbol{u}_i\boldsymbol{u}_i^T$. Examples of spectral functions which we will use are $\exp(t)$, $\log(t)$ and $t\log t$. Note that the last two functions are well defined only on $\mathbf{S}^d_{++}$ and the last function can be extended to $\mathbf{S}^d_+$ as a limiting process. Finally, for vectors $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{\beta} \in \mathbb{R}^p$ we define $[\boldsymbol{\alpha}, \boldsymbol{\beta}] \in \mathbb{R}^{n+p}$ to be the concatenation of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, which we regard as a column vector. Hence $[\boldsymbol{\alpha}, \boldsymbol{\beta}]^T \left[\bar{\boldsymbol{\alpha}}, \bar{\boldsymbol{\beta}}\right] = \boldsymbol{\alpha}^T\bar{\boldsymbol{\alpha}} + \boldsymbol{\beta}^T\bar{\boldsymbol{\beta}}$.

4. The Algorithm

The learning algorithm we propose fits into the broad category of online matrix learning. At the core of the algorithm is an implicit spectral regularization, and we use a modification of matrix winnow (Warmuth, 2007) as our base algorithm.

As input the algorithm is given the graphs $\mathcal{G}$ and $\mathcal{H}$. The algorithm then depends on two input parameters, $K > 1$ and $\hat{\theta}$. The first parameter is the number labelings of the observed graph, which then determines the learning rate η. The second parameter $\hat{\theta}$ is a scaled threshold for the linear classifier. The parameter $\hat{\theta}$ is an upper bound on a measure of the complexity of the underlying learning problem, which is denoted by θ (cf. (6)).

We map a pair of vertices on the observed and latent graphs to a rank one positive semidefinite matrix, and use a linear classifier in the embedded space. Specifically, we map $(i_t, j_t) \in \mathcal{V}(\mathcal{G}) \times \mathcal{V}(\mathcal{H})$ to $\boldsymbol{X}_t \in \mathbf{S}_+^{n+p}$ given by the equation

$$\boldsymbol{X}_t := \frac{1}{2}\boldsymbol{x}_t\boldsymbol{x}_t^T \tag{3}$$

where

$$\boldsymbol{x}_t := \left[\frac{1}{\sqrt{\rho(\mathbf{G})}}(\boldsymbol{G}^{\frac{1}{2}})_{i_t}, \frac{1}{\sqrt{\rho(\mathbf{H})}}(\boldsymbol{H}^{\frac{1}{2}})_{j_t}\right], \tag{4}$$

matrices $\boldsymbol{G} \in \mathbf{S}_{++}^n$ and $\boldsymbol{H} \in \mathbf{S}_{++}^p$ are prescribed and we defined $\rho(\mathbf{G}) := \max_{i=1}^n \mathbf{G}_{ii}$ and $\rho(\mathbf{H}) := \max_{j=1}^p \mathbf{H}_{jj}$. The algorithm works for any such embeddings but the mistake bound presented in Theorem 1 below is obtained by choosing

$$\mathbf{G} = \mathbf{L}_{\mathcal{G}}^+ + R_{\mathcal{G}}\mathbf{1}\mathbf{1}^T \quad \text{and} \quad \mathbf{H} = \mathbf{L}_{\mathcal{H}}^+ + R_{\mathcal{H}}\mathbf{1}\mathbf{1}^T \tag{5}$$

where $\mathbf{1}$ denotes the vector $(1, \ldots, 1)^T$ and $R_{\mathcal{G}} = \max_{i=1}^n (\mathbf{L}_{\mathcal{G}}^+)_{ii}$ and $R_{\mathcal{H}} = \max_{j=1}^p (\mathbf{L}_{\mathcal{H}}^+)_{jj}$ are (essentially) the resistance diameters[1] of $\mathcal{G}$ and $\mathcal{H}$, respectively.

At each trial we predict by a linear threshold function in the embedded space, namely we predict positive if $\mathrm{Tr}\,(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) > \frac{K+1}{2K\hat{\theta}}$ and negative otherwise, where $\boldsymbol{W}_t \in \mathbf{S}_+^{n+p}$ is a parameter matrix which is updated by the algorithm after each trial and initially set to a positive multiple of the identity matrix. Specifically, $\boldsymbol{W}_t$ is updated via Equation (2) only when a mistake is made. The worst case cost of an update is in the order of $(n+p)^3$ since this requires computing an eigensystem of an $(n+p) \times (n+p)$ matrix. However if the number of mistakes is much smaller than $n+p$ then the computation per trial can be substantially reduced because the weight matrix can be decomposed as the sum of a multiple of the identity matrix plus a low rank matrix (specifically the rank at trial t is equal to the current number of mistakes plus one). In this paper we are primarily concerned with the mistake bound and postpone further discussions on large scale implementations of the algorithm to a future occasion.

5. Main Result

In this section, we present our main result and give a detailed proof.

Theorem 1 *Let*

$$\theta = 8R_{\mathcal{G}}\sum_{k=1}^K \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k) + 4R_{\mathcal{H}}\mathrm{cut}_{\mathcal{H}}(\boldsymbol{\mu}) + 2\sum_{k=1}^K \left(\frac{1}{n}\sum_{i=1}^n \omega_{k,i}\right)^2 + 2\sum_{k=1}^K \frac{1}{p}\sum_{j=1}^p \mathcal{I}(\mu_j = k)\,,$$

and let $\bar{c} := (5\log(5/3) - 2)^{-1} \leq 1.81$. *The number of mistakes made by Algorithm 1 with* $\theta \leq \hat{\theta}$ *and learning rate* $\eta := \frac{1}{2}\log\left(\frac{K+3}{K+1}\right)$ *is upper bounded by*

$$4K\bar{c}\left(2R_{\mathcal{G}}\sum_{k=1}^K \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k) + R_{\mathcal{H}}\mathrm{cut}_{\mathcal{H}}(\boldsymbol{\mu}) + K\right)\left(\log(K(n+p)) + \frac{\hat{\theta}}{\theta} - 1\right).$$

1. Specifically, $\max_{i=1}^n (\mathbf{L}_{\mathcal{G}}^+)_{ii}$ is a lower bound on the resistance diameter of $\mathcal{G}$, see (Herbster and Pontil, 2006, Eq. (9)).

To prepare for the proof we introduce some notation. The K-class labeling $\boldsymbol{\mu}$ induces K boolean labelings on $\mathcal{H}$, denoted by $\boldsymbol{\mu}_k \in \{0,1\}^p$, $k \in \mathbb{N}_K$, and is defined componentwise as $\mu_{k,j} = 1$ if $\mu_j = k$ and $\mu_{k,j} = 0$ otherwise. We also define, for every $k \in \mathbb{N}_K$,

$$\Phi_k := \boldsymbol{\mu}_k^T \boldsymbol{H}^{-1} \boldsymbol{\mu}_k, \text{ and } \Phi'_k := \boldsymbol{\omega}_k^T \boldsymbol{G}^{-1} \boldsymbol{\omega}_k.$$

For $i \in \mathbb{N}_n$, we let $\boldsymbol{e}_i$ be the i-th unit basis vector, that is, $e_{i,i'} = 0$ if $i \neq i'$ and $e_{i,i} = 1$.

We let

$$\boldsymbol{z}_k := \left[\sqrt{\rho(\mathbf{G})} \boldsymbol{G}^{-\frac{1}{2}} \boldsymbol{\omega}_k, \sqrt{\rho(\mathbf{H})} \boldsymbol{H}^{-\frac{1}{2}} \boldsymbol{\mu}_k \right]$$

and define the k-th embedded classifier associated with the k-th labelings as

$$\boldsymbol{Z}_k := \frac{\boldsymbol{z}_k \boldsymbol{z}_k^T}{\hat{\theta}},$$

with $\hat{\theta} \geq \theta$ where

$$\theta := \sum_{k=1}^{K} \|\boldsymbol{z}_k\|^2 = \rho(\mathbf{G}) \sum_{k=1}^{K} \boldsymbol{\omega}_k^T \boldsymbol{G}^{-1} \boldsymbol{\omega}_k + \rho(\mathbf{H}) \sum_{k=1}^{K} \boldsymbol{\mu}_k^T \boldsymbol{H}^{-1} \boldsymbol{\mu}_k \,. \tag{6}$$

Note that the representation of the k-th embedded classifier depends on the k-th labeling of the observed graph and the k-th "one versus all" labeling of the latent graph.

We have the following proposition.

Proposition 2 *For all $k \in \mathbb{N}_K$ and trials t it holds that*

$$\text{(i)} \qquad \operatorname{Tr}\left(\boldsymbol{Z}_k^T \boldsymbol{X}_t\right) = \frac{(\omega_{k,i_t} + \mu_{k,j_t})^2}{2\hat{\theta}}$$

$$\text{(ii)} \qquad \sum_{k=1}^{K} \operatorname{Tr}\left(\boldsymbol{Z}_k^T \boldsymbol{X}_t\right) = \frac{(K + 1 + 2y_t)}{2\hat{\theta}}$$

$$\text{(iii)} \qquad \boldsymbol{X}_t \text{ has eigenvalues in } [0,1]$$

$$\text{(iv)} \qquad \|\boldsymbol{z}_k\|^2 = \rho(\mathbf{H})\Phi_k + \rho(\mathbf{G})\Phi'_k$$

$$\text{(v)} \qquad \operatorname{Tr}\left(\boldsymbol{Z}_k \log\left(\boldsymbol{Z}_k\right)\right) < 0.$$

Proof (i): Note that $\operatorname{Tr}\left(\boldsymbol{Z}_k^T \boldsymbol{X}_t\right) = \frac{\operatorname{Tr}\left(\boldsymbol{z}_k \boldsymbol{z}_k^T (\boldsymbol{x}_t \boldsymbol{x}_t^T)\right)}{2\hat{\theta}} = \frac{(\boldsymbol{x}_t^T \boldsymbol{z}_k)^2}{2\hat{\theta}}$. The result then follows since $\boldsymbol{x}_t^T \boldsymbol{z}_k = \boldsymbol{e}_{i_t}^T \boldsymbol{\omega}_k + \boldsymbol{e}_{j_t}^T \boldsymbol{\mu}_k = \omega_{k,i_t} + \mu_{k,j_t}$.

(ii): If $k \neq \mu_{j_t}$ then $\mu_{k,j_t} = 0$ and by part (i) we have

$$\operatorname{Tr}\left(\boldsymbol{Z}_k^T \boldsymbol{X}_t\right) = \frac{1}{2\hat{\theta}}(\omega_{k,i_t} + \mu_{k,j_t})^2 = \frac{1}{2\hat{\theta}}(\omega_{k,i_t})^2 = \frac{1}{2\hat{\theta}}.$$

Suppose now that $k = \mu_{j_t}$. By the definition of y_t we have $y_t = \omega_{\mu_{j_t},i_t} = \omega_{k,i_t}$ so since $\mu_{k,j_t} = 1$ when $k = \mu_{j_t}$ we have, by part (i) that

$$\operatorname{Tr}\left(\boldsymbol{Z}_k^T \boldsymbol{X}_t\right) = \frac{1}{2\hat{\theta}}(\omega_{k,i_t} + \mu_{k,j_t})^2 = \frac{1}{2\hat{\theta}}(1 + y_t)^2 = \frac{1}{2\hat{\theta}}(1 + 2y_t + y_t^2) = \frac{(1 + y_t)}{\hat{\theta}}$$

as $y_t^2 = 1$. By summing the above over k we get the result.

(iii): Note that

$$\boldsymbol{X}_t := \frac{1}{2}\boldsymbol{x}_t\boldsymbol{x}_t^T = \frac{\|\boldsymbol{x}_t\|^2}{2}\frac{\boldsymbol{x}_t}{\|\boldsymbol{x}_t\|}\frac{\boldsymbol{x}_t^T}{\|\boldsymbol{x}_t\|}.$$

Thus $\boldsymbol{X}_t$ is a rank one positive semidefinite matrix and its only nonzero eigenvalue is $\|\boldsymbol{x}_t\|^2/2$. A direct computation then gives that $\|\boldsymbol{x}_t\|^2 \leq 2$. The result follows.

(iv): $\|\boldsymbol{z}_k\|^2 = \boldsymbol{z}_k^T\boldsymbol{z}_k = \rho(\mathbf{G})\boldsymbol{\omega}_k^T\boldsymbol{G}^{-1}\boldsymbol{\omega}_k + \rho(\mathbf{H})\boldsymbol{\mu}_k^T\boldsymbol{H}^{-1}\boldsymbol{\mu}_k = \rho(\mathbf{G})\Phi'_k + \rho(\mathbf{H})\Phi_k$.

(v): Note that $\boldsymbol{Z}_k$ is a positive semidefinite rank one matrix. Hence denoting with λ the non-trivial eigenvalue we have $\operatorname{Tr}(\boldsymbol{Z}_k \log(\boldsymbol{Z}_k)) = \lambda \log \lambda$. The result then follows if we show that $\lambda \in (0,1)$. To see this we write

$$\boldsymbol{Z}_k = \frac{\boldsymbol{z}_k\boldsymbol{z}_k^T}{\hat{\theta}} = \frac{\|\boldsymbol{z}_k\|^2}{\hat{\theta}}\frac{\boldsymbol{z}_k}{\|\boldsymbol{z}_k\|}\frac{\boldsymbol{z}_k^T}{\|\boldsymbol{z}_k\|}.$$

By definition $\theta = \sum_{k=1}^K \|\boldsymbol{z}_k\|^2$ so since $\theta \leq \hat{\theta}$ we have $\frac{\|\boldsymbol{z}_k\|^2}{\hat{\theta}} \leq 1$ as required. ■

We now define the quantum relative entropy, which plays a central role in the amortized analysis of the algorithm. We note that this technique was previously employed in online learning by Tsuda et al. (2005).

Definition 3 *The quantum relative entropy of symmetric positive semidefinite square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ is*

$$\Delta(\boldsymbol{A}, \boldsymbol{B}) := \operatorname{Tr}(\boldsymbol{A}\log(\boldsymbol{A}) - \boldsymbol{A}\log(\boldsymbol{B}) + \boldsymbol{B} - \boldsymbol{A}).$$

We will utilize the following lemmas.

Lemma 4 *For $t \in \mathcal{M}$ we have that*

$$\sum_{k=1}^K (\Delta(\boldsymbol{Z}_k, \boldsymbol{W}_{t-1}) - \Delta(\boldsymbol{Z}_k, \boldsymbol{W}_t)) \geq \frac{c}{K\hat{\theta}}.$$

where $c := 5\log(5/3) - 2$.

Proof When $t \in \mathcal{M}$ we have, for all $k \in \mathbb{N}_K$, that

$$\begin{aligned}
&\Delta(\boldsymbol{Z}_k, \boldsymbol{W}_{t-1}) - \Delta(\boldsymbol{Z}_k, \boldsymbol{W}_t) \\
&\quad = \operatorname{Tr}(\boldsymbol{Z}_k \log(\boldsymbol{W}_t) - \boldsymbol{Z}_k \log(\boldsymbol{W}_{t-1})) + \operatorname{Tr}(\boldsymbol{W}_{t-1}) - \operatorname{Tr}(\boldsymbol{W}_t) \\
&\quad = \eta(y_t - \hat{y}_t)\operatorname{Tr}(\boldsymbol{Z}_k\boldsymbol{X}_t) + \operatorname{Tr}(\boldsymbol{W}_{t-1}) - \operatorname{Tr}(\exp(\log(\boldsymbol{W}_{t-1}) + \eta(y_t - \hat{y}_t)\boldsymbol{X}_t)) \qquad (7) \\
&\quad \geq \eta(y_t - \hat{y}_t)\operatorname{Tr}(\boldsymbol{Z}_k\boldsymbol{X}_t) + \operatorname{Tr}(\boldsymbol{W}_{t-1}) - \operatorname{Tr}(\exp(\log(\boldsymbol{W}_{t-1}))\exp(\eta(y_t - \hat{y}_t)\boldsymbol{X}_t)) \qquad (8) \\
&\quad = \eta(y_t - \hat{y}_t)\operatorname{Tr}(\boldsymbol{Z}_k\boldsymbol{X}_t) + \operatorname{Tr}(\boldsymbol{W}_{t-1}(\boldsymbol{I} - \exp(\eta(y_t - \hat{y}_t)\boldsymbol{X}_t))) \\
&\quad \geq \eta(y_t - \hat{y}_t)\operatorname{Tr}(\boldsymbol{Z}_k\boldsymbol{X}_t) + (1 - e^{\eta(y_t - \hat{y}_t)})\operatorname{Tr}(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) \qquad (9)
\end{aligned}$$

where Equation (7) comes from the algorithm's update of $\boldsymbol{W}_{t-1}$ (see Equation (2)), Equation (8) comes from Lemma A.8 with $\boldsymbol{A} := \log(\boldsymbol{W}_{t-1})$ and $\boldsymbol{B} := \eta(y_t - \hat{y}_t)\boldsymbol{X}_t$, and Equation (9) comes by first applying Lemma A.9 with $a := \eta(y_t - \hat{y}_t)$ and $\boldsymbol{A} := \boldsymbol{X}_t$ (using Proposition 2-(iii)), and then applying Lemma A.10 with $\boldsymbol{A} := \boldsymbol{W}_{t-1}$.
We hence have

$$\begin{aligned}
&\sum_{k=1}^{K}(\Delta(\boldsymbol{Z}_k, \boldsymbol{W}_{t-1}) - \Delta(\boldsymbol{Z}_k, \boldsymbol{W}_t)) \\
\geq &\eta(y_t - \hat{y}_t)\sum_{k=1}^{K}\operatorname{Tr}(\boldsymbol{Z}_k\boldsymbol{X}_t) + K(1 - e^{\eta(y_t-\hat{y}_t)})\operatorname{Tr}(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) \\
= &\eta(y_t - \hat{y}_t)\frac{(K + 1 + 2y_t)}{2\hat{\theta}} + K(1 - e^{\eta(y_t-\hat{y}_t)})\operatorname{Tr}(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) \qquad (10)
\end{aligned}$$

where Equation (10) comes from Proposition 2-(ii).
Let ρ be the right hand side of Equation (10). Noting that $\eta := \frac{1}{2}\log\left(\frac{K+3}{K+1}\right)$ we have the following. When $y_t = 1$ and $\hat{y}_t = -1$ then $(1-e^{\eta(y_t-\hat{y}_t)})$ is negative and $\operatorname{Tr}(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) < \frac{K+1}{2K\hat{\theta}}$ and thus

$$\begin{aligned}
\rho &\geq \eta(K+3)(\hat{\theta})^{-1} + \frac{K+1}{2}(1 - e^{2\eta})(\hat{\theta})^{-1} \\
&= \frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K+3)(\hat{\theta})^{-1} + \frac{K+1}{2}\left(1 - \frac{K+3}{K+1}\right)(\hat{\theta})^{-1} \\
&= \left(\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K+3) - 1\right)(\hat{\theta})^{-1} \\
&\geq \frac{c}{K\hat{\theta}}. \qquad (11)
\end{aligned}$$

Alternately, when $y_t = -1$ and $\hat{y}_t = 1$ then $(1 - e^{\eta(y_t-\hat{y}_t)})$ is positive and $\operatorname{Tr}(\boldsymbol{W}_{t-1}\boldsymbol{X}_t) \geq \frac{K+1}{2K\hat{\theta}}$ and thus

$$\begin{aligned}
\rho &\geq -\eta(K-1)(\hat{\theta})^{-1} + \frac{K+1}{2}(1 - e^{-2\eta})(\hat{\theta})^{-1} \\
&= -\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K-1)(\hat{\theta})^{-1} + \frac{K+1}{2}\left(1 - \frac{K+1}{K+3}\right)(\hat{\theta})^{-1} \\
&= \left(\frac{K+1}{K+3} - \frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K-1)\right)(\hat{\theta})^{-1} \\
&\geq \frac{c}{K\hat{\theta}}. \qquad (12)
\end{aligned}$$

The constant c in Equations (11) and (12) is derived from the following argument. For $K \geq 2$ the functions

$$K\left(\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K+3) - 1\right) \qquad (13)$$

and

$$K\left(\frac{K+1}{K+3} - \frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K-1)\right) \qquad (14)$$

are monotonic increasing (see Lemmas A.12 and A.13) so

$$\begin{aligned} K\left[\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K+3)-1\right] &\geq 2\left[\frac{1}{2}\log\left(\frac{2+3}{2+1}\right)(2+3)-1\right] \\ &= 5\log\left(\frac{5}{3}\right)-2=c \end{aligned}$$

and

$$\begin{aligned} K\left[\frac{K+1}{K+3}-\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K-1)\right] &\geq 2\left[\frac{2+1}{2+3}-\frac{1}{2}\log\left(\frac{2+3}{2+1}\right)(2-1)\right] \\ &= \frac{6}{5}-\log\left(\frac{5}{3}\right)>c. \end{aligned}$$

Hence $\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K+3)-1 \geq \frac{c}{K}$ and $\frac{K+1}{K+3}-\frac{1}{2}\log\left(\frac{K+3}{K+1}\right)(K-1) \geq \frac{c}{K}$. ■

Lemma 5 *It holds that* $\sum_{k=1}^{K}\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_0) \geq |\mathcal{M}|\frac{c}{K\hat{\theta}}$.

Proof We have

$$\begin{aligned} \sum_{k=1}^{K}\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_0) &\geq \sum_{k=1}^{K}(\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_0)-\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_T)) \\ &= \sum_{k=1}^{K}\sum_{t=1}^{T}(\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_{t-1})-\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_t)) \\ &= \sum_{t=1}^{T}\sum_{k=1}^{K}(\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_{t-1})-\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_t)) \\ &= \sum_{t\in\mathcal{M}}\sum_{k=1}^{K}(\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_{t-1})-\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_t))+\sum_{t\notin\mathcal{M}}\sum_{k=1}^{K}(\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_{t-1})-\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_t)) \\ &\geq |\mathcal{M}|\frac{c}{K\hat{\theta}} \qquad (15) \end{aligned}$$

where Equation (15) comes from Lemma 4 and the fact that on a trial $t \notin \mathcal{M}$ we have $\boldsymbol{W}_t = \boldsymbol{W}_{t-1}$ and hence $\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_{t-1}) = \Delta(\boldsymbol{Z}_k,\boldsymbol{W}_t)$ for all $k \in \mathbb{N}_K$. ■

Lemma 6 *It holds that* $\sum_{k=1}^{K}\Delta(\boldsymbol{Z}_k,\boldsymbol{W}_0) \leq \frac{\theta}{\hat{\theta}}\log\left(K(n+p)\right)+\left(1-\frac{\theta}{\hat{\theta}}\right)$.

Proof of Lemma 6 Recall that $\boldsymbol{W}_0 = \frac{\boldsymbol{I}}{K(n+p)}$, where $\boldsymbol{I}$ is the $(n+p) \times (n+p)$ identity matrix. We observe that

$$\begin{aligned}
\sum_{k=1}^{K} \Delta(\boldsymbol{Z}_k, \boldsymbol{W}_0) &= \sum_{k=1}^{K} (\operatorname{Tr}(\boldsymbol{Z}_k \log(\boldsymbol{Z}_k)) - \operatorname{Tr}(\boldsymbol{Z}_k \log(\boldsymbol{W}_0)) + \operatorname{Tr}(\boldsymbol{W}_0) - \operatorname{Tr}(\boldsymbol{Z}_k)) \\
&\leq -\sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k \log(\boldsymbol{W}_0)) + \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{W}_0) - \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) \qquad (16) \\
&= -\sum_{k=1}^{K} \operatorname{Tr}\left(\boldsymbol{Z}_k \log\left(\frac{\boldsymbol{I}}{K(n+p)}\right)\right) + \sum_{k=1}^{K} \operatorname{Tr}\left(\frac{\boldsymbol{I}}{K(n+p)}\right) - \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) \\
&= \log(K(n+p)) \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) + \frac{1}{K(n+p)} \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{I}) - \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) \\
&= \log(K(n+p)) \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) + 1 - \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) \\
&= 1 + (\log(K(n+p) - 1)) \sum_{k=1}^{K} \operatorname{Tr}(\boldsymbol{Z}_k) \\
&= 1 + (\log(K(n+p) - 1)) \sum_{k=1}^{K} \operatorname{Tr}\left(\frac{\boldsymbol{z}_k \boldsymbol{z}_k^T}{\hat{\theta}}\right) \\
&= 1 + (\log(K(n+p) - 1)) \frac{1}{\hat{\theta}} \sum_{k=1}^{K} \boldsymbol{z}_k^T \boldsymbol{z}_k \\
&= 1 + \left(\log(K(n+p) - 1) \frac{\theta}{\hat{\theta}}\right) \qquad (17) \\
&= \frac{\theta}{\hat{\theta}} \log(K(n+p)) + \left(1 - \frac{\theta}{\hat{\theta}}\right)
\end{aligned}$$

where Equation (16) comes from Proposition 2-(v) and Equation (17) comes from the definition of θ. ■

We are now ready to prove our main result.

Proof of Theorem 1 Combining Lemmas 6 and 5 we have

$$|\mathcal{M}| \frac{c}{K\hat{\theta}} \leq \sum_{k=1}^{K} \Delta(\boldsymbol{Z}_k, \boldsymbol{W}_0) \leq \frac{\theta}{\hat{\theta}} \log(K(n+p)) + \left(1 - \frac{\theta}{\hat{\theta}}\right)$$

which gives

$$\begin{aligned}|\mathcal{M}| &\leq \frac{K\hat{\theta}}{c}\frac{\theta}{\hat{\theta}}\log\left(K(n+p)\right) + \frac{K\hat{\theta}}{c}\left(1-\frac{\theta}{\hat{\theta}}\right)\\ &= \frac{K\theta}{c}\log\left(K(n+p)\right) + \frac{K\theta}{c}\frac{\hat{\theta}}{\theta}\left(1-\frac{\theta}{\hat{\theta}}\right)\\ &= \frac{K\theta}{c}\left(\log\left(K(n+p)\right) + \left(\frac{\hat{\theta}}{\theta}-1\right)\right).\end{aligned}$$

Finally we compute θ by choosing the matrices $\mathbf{H}$ and $\mathbf{G}$ as per Equation (5). A direct computation gives, for any vector $\boldsymbol{\omega} \in \mathbb{R}^n$, that

$$\boldsymbol{\omega}^T\boldsymbol{G}^{-1}\boldsymbol{\omega} = \boldsymbol{\omega}^T(\mathbf{L}_{\mathcal{G}}^{+} + R_{\mathcal{G}}\mathbf{1}\mathbf{1}^T)^{-1}\boldsymbol{\omega} = \boldsymbol{\omega}^T\mathbf{L}_{\mathcal{G}}\boldsymbol{\omega} + \frac{1}{R_{\mathcal{G}}}\left(\frac{1}{n}\sum_{i=1}^{n}\omega_i\right)^2$$

and, likewise, for any vector $\boldsymbol{\mu} \in \mathbb{R}^p$

$$\boldsymbol{\mu}^T\boldsymbol{H}^{-1}\boldsymbol{\mu} = \boldsymbol{\mu}^T(\mathbf{L}_{\mathcal{H}}^{+} + R_{\mathcal{H}}\mathbf{1}\mathbf{1}^T)^{-1}\boldsymbol{\mu} = \boldsymbol{\mu}^T\mathbf{L}_{\mathcal{H}}\boldsymbol{\mu} + \frac{1}{R_{\mathcal{H}}}\left(\frac{1}{p}\sum_{j=1}^{p}\mu_j\right)^2.$$

For the observed labelings we have $\boldsymbol{\omega}_k^T\mathbf{L}_{\mathcal{G}}\boldsymbol{\omega}_k = 4\mathrm{cut}(\boldsymbol{\omega}_k)$. Using this and $\rho(\mathbf{G}) = 2R_{\mathcal{G}}$, a direct computation gives

$$\begin{aligned}\rho(\mathbf{G})\sum_{k=1}^{K}\boldsymbol{\omega}_k^T\boldsymbol{G}^{-1}\boldsymbol{\omega}_k &= 2R_{\mathcal{G}}\left(4\sum_{k=1}^{K}\mathrm{cut}(\boldsymbol{\omega}_k) + \frac{1}{R_{\mathcal{G}}}\sum_{k=1}^{K}\left(\frac{1}{n}\sum_{i=1}^{n}\omega_{k,i}\right)^2\right)\\ &\leq 8R_{\mathcal{G}}\sum_{k=1}^{K}\mathrm{cut}(\boldsymbol{\omega}_k) + 2K.\end{aligned}$$

For the latent labeling we have $\sum_{k=1}^{K}\boldsymbol{\mu}_k^T\mathbf{L}_{\mathcal{H}}\boldsymbol{\mu}_k = 2\mathrm{cut}(\boldsymbol{\mu})$. Using this and $\rho(\mathbf{H}) = 2R_{\mathcal{H}}$, we obtain

$$\begin{aligned}\rho(\mathbf{H})\sum_{k=1}^{K}\boldsymbol{\mu}_k^T\boldsymbol{H}^{-1}\boldsymbol{\mu}_k &= 2R_{\mathcal{H}}\left(2\mathrm{cut}(\boldsymbol{\mu}) + \frac{1}{R_{\mathcal{H}}}\sum_{k=1}^{K}\left(\frac{1}{n}\sum_{i=1}^{n}\mu_{k,i}\right)^2\right)\\ &\leq 4R_{\mathcal{H}}\mathrm{cut}(\boldsymbol{\mu}) + 2K.\end{aligned}$$

We conclude that

$$\begin{aligned}\theta = \sum_{k=1}^{K}\|\boldsymbol{z}_k\|^2 &= \rho(\mathbf{G})\sum_{k=1}^{K}\boldsymbol{\omega}_k^T\boldsymbol{G}^{-1}\boldsymbol{\omega}_k + \rho(\mathbf{H})\sum_{k=1}^{K}\boldsymbol{\mu}_k^T\boldsymbol{H}^{-1}\boldsymbol{\mu}_k\\ &\leq 4\left(2R_{\mathcal{G}}\sum_{k=1}^{K}\mathrm{cut}(\boldsymbol{\omega}_k) + R_{\mathcal{H}}\mathrm{cut}(\boldsymbol{\mu}) + K\right).\end{aligned}$$

The result now follows by substituting the last inequality in the mistake bound. ■

6. Discussion

In this section, we consider two special cases of the problem studied in this paper and make final remarks. We tailor Theorem 1 to these cases and then compare to similar mistake bounds available in the literature.

6.1 Uniform Multitask Prediction

In the uniform multitask problem we suppose that we have p tasks corresponding to predicting the binary labeling of a graph. We assume that the tasks are interrelated so that only $K \ll p$ graph labelings are needed. To solve this problem we assume each task is given a number in $\{1, \ldots, p\}$. Each task number denotes a unique vertex in the latent graph which is a p-vertex clique. Applying the bound of Theorem 1 gives

$$M \leq \mathcal{O}\left(\left(\sum_{k=1}^{K} \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k) R_{\mathcal{G}} + p\right) K \log(K(n+p))\right).$$

This follows immediately from the fact that the clique has resistance diameter $\mathcal{O}(\frac{1}{p})$ and the cut of a K-"coloring" is $\mathcal{O}(p^2)$.

In (Cavallanti et al., 2010), a broad range of results are given for online multi-task learning in generic reproducing kernel Hilbert spaces. We apply their Corollary 3 to our problem with the kernel $\mathbf{G}^{-1} := \mathbf{L}_{\mathcal{G}}^{+} + R_{\mathcal{G}} \mathbf{1}\mathbf{1}^T$. In their setting there is no parameter K and instead they learn p distinct graph labelings, and thus obtain

$$M \leq \mathcal{O}\left(\frac{1}{p}\left(\sum_{k=1}^{p} \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k) + \sum_{i<j}^{p} (\boldsymbol{\omega}_i - \boldsymbol{\omega}_j)^T \mathbf{G} (\boldsymbol{\omega}_i - \boldsymbol{\omega}_j)\right) R_{\mathcal{G}}\right).$$

This is small when each of the p binary labelings are near one another in the norm induced by the Laplacian. This is distinct from our bound where we pay a fixed price for each of the p tasks of $K \log(K(n+p))$. Thus our bound is stronger when $K \ll p$ and the averaged squared norm between labelings of the p tasks is larger than $K \log(K(n+p))$.

6.2 Switching

We now consider the case where we have a switching sequence of graph labelings with S switches between K labelings. We sketch a proof of the bound announced in the introduction (cf. Equation (1)), namely

$$M \leq \tilde{\mathcal{O}}\left(\left(S + R_{\mathcal{G}} \sum_{k=1}^{K} \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k)\right) K \log(n)\right),$$

where the $\tilde{\mathcal{O}}(x)$ notation absorbs a polylogarithmic factor in x. Notice that the apparently natural structure for the proof would be to choose a latent graph which is a "line" with T vertices, where the linear ordering of the vertices reflects the linear trial sequence. Unfortunately, the resistance diameter of this line graph would then be equal to T which would make the bound vacuous. We overcome this difficulty by borrowing a trick from (Herbster

et al., 2008) and we instead use a binary tree with T leaves and thus a resistance diameter of $2\log_2 T$. We assume that for each trial we receive a label of a leaf along the natural linear ordering of the leaves. If ϕ is the cut along the leaves such a labeling may be extended to a labeling of the complete binary tree in a way that the cut increases by no more than a factor of $\log_2 T$. This extension works by choosing the label of the parent of each vertex to be consistent with the label of either of its children. The result follows since the labeling on each successive "level" of the tree down to the root is now a subsequence of the previous labeling, and the cut of a subsequence can only decrease. Hence with $\log_2 T$ levels the cut increases by no more than a logarithmic factor. A second insight is that we do not actually need a tree with T leaves, we in fact only need M leaves corresponding to when the algorithm incurs a mistake, hence,

$$M \leq \mathcal{O}\left(\left(S(\log(M))^2 + R_{\mathcal{G}} \sum_{k=1}^{K} \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k)\right) K \log(K(n+p))\right)$$

which we upper bound by

$$M \leq \mathcal{O}\left(\log(M)^3 \left(S + R_{\mathcal{G}} \sum_{k=1}^{K} \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_k)\right) K \log(Kn)\right).$$

Then the following technical lemma gives the result (proof in the Appendix).

Lemma 7 *Given a function $M : \mathbb{R} \to \mathbb{R}$, a constant $e > 0$ such that $M(x) \leq ex\log(M(x))^3$, then there exist constants $a, b > 0$ such that $M(x) \leq ax\log(x)^3$ for all $x > b$.*

We may also apply the technique of Herbster and Warmuth (1998) to the switching problem. Here, the underlying learning algorithm would be the perceptron with the kernel $\mathbf{G}^{-1} := \mathbf{L}_{\mathcal{G}}^{+} + R_{\mathcal{G}}\mathbf{1}\mathbf{1}^T$. As in (Cavallanti et al., 2010) the implicit assumption is that the underlying switching process is smooth and thus there is no parameter K, just a sequence of $S+1$ graph labelings. The other ingredient needed for a tracking kernel perceptron is an upper bound $\hat{\phi} := \max_{k\in\{1,\ldots,S+1\}} \boldsymbol{\omega}_k^T \mathbf{G} \boldsymbol{\omega}_k$. The upper bound is then used to define an additional update to the perceptron, which maintains the hypothesis vector in a ball of squared norm equal to $\hat{\phi}$. This perceptron update and projection step then lead to the bound (Herbster and Warmuth, 1998, Theorem 10),

$$M \leq \mathcal{O}\left(\left(\sum_{k=1}^{S} \sqrt{\hat{\phi}(\boldsymbol{\omega}_k - \boldsymbol{\omega}_{k+1})^T \mathbf{G}(\boldsymbol{\omega}_k - \boldsymbol{\omega}_{k+1})} + \mathrm{cut}_{\mathcal{G}}(\boldsymbol{\omega}_{S+1})\right) R_{\mathcal{G}}\right).$$

Thus we observe with the projection kernel perceptron we pay a cost of

$$\sqrt{\hat{\phi}(\boldsymbol{\omega}_k - \boldsymbol{\omega}_{k+1})^T \mathbf{G}(\boldsymbol{\omega}_k - \boldsymbol{\omega}_{k+1})} R_{\mathcal{G}}$$

for each switch $k \in \{1, \ldots, S\}$. Whereas when $K \ll S$ the dominant non poly-logarithmic term we pay per switch is $\mathcal{O}(K \log n)$.

6.3 Final Remarks

In this paper we presented a novel setting for online prediction over a graph. Our model is governed by K binary labelings and a latent K-labeling (defined on a second graph) which determines which one of the binary labelings is active at each trial.

We proposed an efficient algorithm for online prediction in this setting and derived a bound on the number of mistakes made by the algorithm. An interesting feature of this bound is that it mimics the bound one would obtain having *a-priori* information about which binary labeling is active at each trial. A shortcoming of the bound is that it requires knowledge of the number of binary labelings K and the threshold θ. In practice these parameters are not known in advance and techniques based on the "doubling trick" could be employed to tune the parameters.

Finally, we note that the problem considered in this paper could also be applied to the batch learning setting and our bound may be converted to a batch bound using techniques from (Cesa-Bianchi et al., 2004). In the batch setting a natural algorithm is given by empirical error minimization (Vapnik, 1998) over a hypothesis space of binary classifiers defined on the graph. This space is obtained by a certain function composition involving the binary labelings and the latent labeling. We conjecture that the problem of performing empirical error minimization over such a hypothesis space is NP-hard. Therefore in future work our algorithm could be employed to obtain an efficient sub-optimal solution to empirical error minimization in this challenging setting.

Acknowledgments

The third author acknowledges support from the EPSRC and GENES.

Appendix A. Appendix

In this appendix, we state some auxiliary results which are used in the main body of the paper.

The first result is the famous Golden-Thompson Inequality, whose proof can be found, for example, in (Bhatia, 1997).

Lemma A.8 *For any symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ we have that*

$$\operatorname{Tr}\left(\exp\left(\boldsymbol{A}+\boldsymbol{B}\right)\right) \leq \operatorname{Tr}\left(\exp\left(\boldsymbol{A}\right)\exp\left(\boldsymbol{B}\right)\right).$$

The next two results are taken from (Tsuda et al., 2005).

Lemma A.9 *If $\boldsymbol{A} \in \mathbf{S}_+^d$ with eigenvalues in $[0,1]$ and $a \in \mathbb{R}$, then*

$$(1-e^a)\boldsymbol{A} \preceq \boldsymbol{I} - \exp\left(a\boldsymbol{A}\right).$$

Lemma A.10 *If $\boldsymbol{A} \in \mathbf{S}_+^d$ and $\boldsymbol{B}, \boldsymbol{C}$ are symmetric matrices such that $\boldsymbol{B} \preceq \boldsymbol{C}$, then*

$$\operatorname{Tr}\left(\boldsymbol{A}\boldsymbol{B}\right) \leq \operatorname{Tr}\left(\boldsymbol{A}\boldsymbol{C}\right).$$

Next we show that the functions (13) and (14) are monotonic increasing. We will use the following lemma.

Lemma A.11 *For every $x > 0$ it holds that $\frac{2x}{2+x} < \log(1+x) < \frac{x}{\sqrt{x+1}}$.*

Proof To prove the right inequality, we let

$$f(x) = \frac{x}{\sqrt{x+1}} - \log(x+1).$$

Since $f(x) = 0$ as $x \to 0$, the result follows if we show that $f'(x) > 0$ for $x > 0$. We have that

$$f'(x) = \frac{x - 2\sqrt{x+1} + 2}{2(x+1)^{3/2}}\,.$$

With a change of variable $x \to z^2 - 1$, we have

$$\frac{x - 2\sqrt{x+1} + 2}{2(x+1)^{3/2}} = \frac{(1-z)^2}{2z^3}\,,$$

which is positive for $z \in (1, \infty]$ and hence $x \in (0, \infty)$.

The proof of the left inequality follows a similar pattern. ■

Lemma A.12 *The following function*

$$f(k) = k\left(\frac{1}{2}(k+3)\log\left(\frac{k+3}{k+1}\right) - 1\right)$$

is increasing for $k \geq 2$.

Proof Differentiating, we have

$$f'(k) = \frac{\left(2k^2+5k+3\right)\log\left(\frac{k+3}{k+1}\right) - 4k - 2}{2(k+1)}\,.$$

We will check to see if the numerator of the above expression is positive. Using the left inequality in Lemma A.11 we have that

$$(2k^2+5k+3)\log\left(\frac{k+3}{k+1}\right) - 4k - 2 \geq \frac{2(2k^2+5k+3)}{2+k} - 4k - 2 = \frac{2}{2+k} > 0.$$

■

Lemma A.13 *The following function*

$$g(k) = k\left(\frac{k+1}{k+3} - \frac{1}{2}(k-1)\log\left(\frac{k+3}{k+1}\right)\right)$$

is increasing for $k \geq 2$.

Proof Differentiating, we have

$$g'(k) = \frac{2\left(2k^3+9k^2+6k+3\right) - (k+3)^2\left(2k^2+k-1\right)\log\left(\frac{k+3}{k+1}\right)}{2(k+1)(k+3)^2}\,. \tag{18}$$

We will show that the numerator of the above expression is positive. The right inequality in Lemma A.11 gives that

$$\log\left(\frac{k+3}{k+1}\right) < \frac{2\sqrt{\frac{k+3}{k+1}}}{k+3}.$$

Using this, we lower bound the numerator in the r.h.s. of equation (18) by

$$2\left(-(k+3)\sqrt{\frac{k+3}{k+1}}\left(2k^2+k-1\right) + k(k(2k+9)+6)+3\right)\,.$$

With a change of variable $k \to \frac{3-y^2}{y^2-1}$, we have

$$\frac{8\left(y^6-7y^3+12y^2+3y^4\,(y-2)-3\right)}{\left(y^2-1\right)^3}\,.$$

Note $k \in [2,\infty)$ implies $y \in (1, \sqrt{\frac{5}{3}}]$. Since we are checking for positivity we strike the term $\frac{8}{(y^2-1)^3}$ which gives

$$y^6+3(y-2)y^4-7y^3+12y^2-3\,.$$

Factoring the above gives

$$(-1+y)^3(3+y(3+y)^2)\,,$$

which is positive for $y \in (1, \sqrt{\frac{5}{3}}]$. ■

Proof of Lemma 7. Without loss of generality let $e = 1$ (else consider the function M', defined by $M'(x) := M(x/e)$, instead of M (noting that $\log(ex)^3 \in O(\log(x)^3)$)).

Note first that we have some d such that for all $y > d$ we have that the function $y \to \frac{y}{\log(y)^3}$ is increasing.

Since $\exp(x) \in \omega(x^6)$ we have $\exp\left(x^{\frac{1}{3}}\right) \in \omega(x^2)$ so $\frac{1}{x}\exp\left(x^{\frac{1}{3}}\right) \in \omega(x)$. There hence exists a c such that for all $x > c$ we have $\frac{1}{x}\exp\left(x^{\frac{1}{3}}\right) > x$.

Let $b := \max\{c, \log(d)^3\}$. Now suppose we have some $x > b$. We then prove the inequality $\log(M(x))^3 \leq x$. To show this consider the converse, that $\log(M(x))^3 > x$. Then $M(x) > \exp\left(x^{\frac{1}{3}}\right)$. Since the function $y \to y/\log(y)^3$ is increasing for $y > d$ and $\exp\left(x^{\frac{1}{3}}\right) > d$, we then have that $M(x)/\log(M(x))^3 > \frac{1}{x}\exp\left(x^{\frac{1}{3}}\right)$, which is greater than x since $x > c$. But this contradicts the fact that $M(x) \leq x\log(M(x))^3$. So we have shown

that $\log(M(x))^3 \leq x$.

If we have $M(x) > 8x\log(x)^3$ then we have $8x\log(x)^3 < M(x) \leq x\log(M(x))^3$ so we must have $2\log(x) < \log(M(x))$ so we have $x^2 < M(x)$. But, by above, $\log(M(x))^3 \leq x$, and hence $M(x) \leq x\log(M(x))^3 \leq x^2$ which is a contradiction. Hence we have that $M(x) \leq 8x\log(x)^3$ as required. ■

References

J. Abernethy, P. Bartlett, and A. Rakhlin. Multitask learning with expert advice. In *Proceedings 20th Annual Conference on Learning Theory*, pages 484–498, 2007.

D. Adamskiy, M.K. Warmuth, and W.M. Koolen. Putting Bayes to sleep. In *Advances in Neural Information Processing Systems 25*, pages 135–143, 2012.

S. Avishek, R. Piyush, H. Daumé III, and S. Venkatasubramanian. Online learning of multiple tasks and their relationships. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, pages 643–651, 2011.

R. Bhatia. *Matrix Analysis*. Springer, 1997.

A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pages 19–26, 2001.

O. Bousquet and M.K. Warmuth. Tracking a small set of experts by mixing past posteriors. *The Journal of Machine Learning Research*, 3:363–396, 2003.

G. Cavallanti, N. Cesa-Bianchi, and C. Gentile. Linear algorithms for online multitask classification. *Journal of Machine Learning Research*, 1:2901–2934, 2010.

N. Cesa-Bianchi and C. Gentile. Tracking the best hyperplane with a simple budget perceptron. In *Proceedings of the 18th Conference on Learning Theory*, pages 483–498, 2006.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

N. Cesa-Bianchi, C. Gentile, and F. Vitale. Fast and optimal prediction on a labeled tree. In *Proceedings of the 22nd Annual Conference on Learning*, 2009.

N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella. Random spanning trees and the prediction of weighted graphs. In *Proceedings of the 27th International Conference on Machine Learning*, pages 175–182, 2010.

N. Cesa-Bianchi, P. Gaillard, G. Lugosi, and G. Stoltz. Mirror descent meets fixed share (and feels no regret). In *Advances in Neural Information Processing Systems 24*, pages 989–997, 2012.

O. Dekel, P.M. Long, and Y. Singer. Online learning of multiple tasks with a shared loss. *Journal of Machine Learning Research*, 8(10):2233–2264, 2007.

Y. Freund. Private communication, 2000. Also posted on http://www.learning-theory.org.

C. Gentile, M. Herbster, and S. Pasteris. Online similarity prediction of networked data from known and unknown graphs. In *Proceedings of the 26th Annual Conference on Learning Theory*, 2013.

L.A. Goldberg and M. Jerrum. The complexity of ferromagnetic Ising with local fields. *Combinatorics, Probability & Computing*, 16(1):43–61, 2007.

A. Gyorfi, T. Linder, and G. Lugosi. Tracking the best of many experts. In *Proceedings 18th Annual Conference on Learning Theory*, pages 204–216, 2005.

E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th International Conference on Machine Learning*, pages 393–400, 2009.

M. Herbster and M. Pontil. Prediction on a graph with a perceptron. In *Advances in Neural Information Processing Systems 19*, pages 577–584, 2006.

M. Herbster and M.K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998.

M. Herbster and M.K. Warmuth. Tracking the best linear predictor. *The Journal of Machine Learning Research*, 1:281–309, 2001.

M. Herbster, M. Pontil, and L. Wainer. Online learning over graphs. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 305–312, 2005.

M. Herbster, G. Lever, and M. Pontil. Online prediction on large diameter graphs. In *Advances in Neural Information Processing Systems 21*, pages 649–656, 2008.

M. Herbster, M. Pontil, and S. Rojas-Galeano. Fast prediction on a tree. In *Advances in Neural Information Processing Systems*, pages 657–664, 2009.

J. Kivinen, A. J. Smola, and R. C. Williamson. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52:2165–2176, 2004.

R.I. Kondor and J.D. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th International Conference on Machine Learning*, pages 315–322, 2002.

W. M. Koolen and S. Rooij. Combining expert advice efficiently. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 275–286, 2008.

K. Tsuda, G. Rätsch, and M.K. Warmuth. Matrix exponentiated gradient updates for online learning and Bregman projection. *Journal of Machine Learning Research*, 6:995–1018, 2005.

V. Vapnik. *Statistical Learning Theory*. Wiley-Blackwell, 1998.

F. Vitale, N. Cesa-Bianchi, C. Gentile, and G. Zappella. See the tree through the lines: The shazoo algorithm. In *Advances in Neural Information Processing Systems 23*, pages 1584–1592, 2011.

V. Vovk. Derandomizing stochastic prediction strategies. *Machine Learning*, 35(3):247–282, 1999.

M.K. Warmuth. Winnowing subspaces. In *Proceedings of the 24th International Conference on Machine Learning*, pages 999–1006, 2007.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 912–919, 2003.

Journal of Machine Learning Research 16 (2015) 2023-2049 Submitted 7/15; Published 9/15

In memory of Alexey Chervonenkis

Learning Using Privileged Information: Similarity Control and Knowledge Transfer

Vladimir Vapnik VLADIMIR.VAPNIK@GMAIL.COM
Columbia University
New York, NY 10027, USA
Facebook AI Research
New York, NY 10017, USA

Rauf Izmailov RIZMAILOV@APPCOMSCI.COM
Applied Communication Sciences
Basking Ridge, NJ 07920-2021, USA

Editor: Alex Gammerman and Vladimir Vovk

Abstract

This paper describes a new paradigm of machine learning, in which Intelligent Teacher is involved. During training stage, Intelligent Teacher provides Student with information that contains, along with classification of each example, additional privileged information (for example, explanation) of this example. The paper describes two mechanisms that can be used for significantly accelerating the speed of Student's learning using privileged information: (1) correction of Student's concepts of similarity between examples, and (2) direct Teacher-Student knowledge transfer.

Keywords: intelligent teacher, privileged information, similarity control, knowledge transfer, knowledge representation, frames, support vector machines, SVM+, classification, learning theory, kernel functions, similarity functions, regression

1. Introduction

During the last fifty years, a strong machine learning theory has been developed. This theory (see Vapnik and Chervonenkis, 1974, Vapnik, 1995, Vapnik, 1998, Chervonenkis, 2013) includes:

- The necessary and sufficient conditions for consistency of learning processes.
- The bounds on the rate of convergence, which, in general, cannot be improved.
- The new inductive principle called Structural Risk Minimization (SRM), which always converges to the best possible approximation in the given set of functions[1].

1. Let a set S of functions $f(x, \alpha), \alpha \in \Lambda$ be given. We introduce a structure $S_1 \subset S_2 \subset ... \subset S$ on this set, where S_k is the subset of functions with VC dimension k. Consider training set $(x_1, y_1), \ldots, (x_\ell, y_\ell)$. In the SRM framework, by choosing an element S_k and a function in this element to minimize the CV bound for samples of size ℓ, one chooses functions $f(x, \alpha_\ell) \in S_k$ such that the sequence $\{f(x, \alpha_\ell\}, \ell \to \infty$,

- The effective algorithms, such as Support Vector Machines (SVM), that realize the consistency property of SRM principle[2].

The general learning theory appeared to be completed: it addressed almost all standard questions of the statistical theory of inference. However, as always, the devil is in the detail: it is a common belief that human students require far fewer training examples than any learning machine. Why?

We are trying to answer this question by noting that a human Student has an Intelligent Teacher[3] and that Teacher-Student interactions are based not only on brute force methods of function estimation. In this paper, we show that Teacher-Student interactions can include special learning mechanisms that can significantly accelerate the learning process. In order for a learning machine to use fewer observations, it can use these mechanisms as well.

This paper considers a model of learning with the so-called Intelligent Teacher, who supplies Student with intelligent (privileged) information during training session. This is in contrast to the classical model, where Teacher supplies Student only with outcome y for event x.

Privileged information exists for almost any learning problem and this information can significantly accelerate the learning process.

2. Learning with Intelligent Teacher: Privileged Information

The existing machine learning paradigm considers a simple scheme: given a set of training examples, find, in a given set of functions, the one that approximates the unknown decision rule in the best possible way. In such a paradigm, Teacher does not play an important role.

In human learning, however, the role of Teacher is important: along with examples, Teacher provides students with explanations, comments, comparisons, metaphors, and so on. In the paper, we include elements of human learning into classical machine learning paradigm. We consider a learning paradigm called *Learning Using Privileged Information (LUPI)*, where, at the training stage, Teacher provides additional information x^* about training example x.

The crucial point in this paradigm is that the privileged information is available only at the training stage (when Teacher interacts with Student) and is not available at the test stage (when Student operates without supervision of Teacher).

In this paper, we consider two mechanisms of Teacher–Student interactions in the framework of the LUPI paradigm:

1. *The mechanism to control Student's concept of similarity between training examples.*

---

strongly uniformly converges to the function $f(x, \alpha_0)$ that minimizes the error rate on the closure of $\cup_{k=1}^{\infty} S_k$ (Vapnik and Chervonenkis, 1974), (Vapnik, 1982), (Devroye et al., 1996), (Vapnik, 1998).

2. Solutions of SVM belong to Reproducing Kernel Hilbert Space (RKHS). Any subset of functions in RKHS with bounded norm has a finite VC dimension. Therefore, SRM with respect to the value of norm of functions satisfies the general SRM model of strong uniform convergence. In SVM, the element of SRM structure is defined by parameter C of SVM algorithm.
3. This is how a Japanese proverb assesses teacher's influence: "Better than a thousand days of diligent study is one day with a great teacher."

2. *The mechanism to transfer knowledge from the space of privileged information (space of Teacher's explanations) to the space where decision rule is constructed.*

The first mechanism (Vapnik, 2006) was introduced in 2006 using SVM+ method. Here we reinforce SVM+ by constructing a parametric family of methods $\text{SVM}_\Delta+$; for $\Delta = \infty$, the method $\text{SVM}_\Delta+$ is equivalent to SVM+. The first experiments with privileged information using SVM+ method were described in Vapnik and Vashist (2009); later, the method was applied to a number of other examples (Sharmanska et al., 2013; Ribeiro et al., 2012; Liang and Cherkassky, 2008).

The second mechanism was introduced recently (Vapnik and Izmailov, 2015b).

2.1 Classical Model of Learning

Formally, the classical paradigm of machine learning is described as follows: given a set of iid pairs (training data)

$$(x_1, y_1), ..., (x_\ell, y_\ell), \quad x_i \in X, \quad y_i \in \{-1, +1\}, \tag{1}$$

generated according to a fixed but unknown probability measure $P(x, y)$, find, in a given set of indicator functions $f(x, \alpha), \alpha \in \Lambda$, the function $y = f(x, \alpha_*)$ that minimizes the probability of incorrect classifications (incorrect values of $y \in \{-1, +1\}$). In this model, each vector $x_i \in X$ is a description of an example generated by Nature according to an unknown generator $P(x)$ of random vectors x_i, and $y_i \in \{-1, +1\}$ is its classification defined according to a conditional probability $P(y|x)$. The goal of Learning Machine is to find the function $y = f(x, \alpha_*)$ that guarantees the smallest probability of incorrect classifications. That is, the goal is to find the function which minimizes the risk functional

$$R(\alpha) = \frac{1}{2} \int |y - f(x, \alpha)| dP(x, y) \tag{2}$$

in the given set of indicator functions $f(x, \alpha),\ \alpha \in \Lambda$ when the probability measure $P(x, y) = P(y|x)P(x)$ is unknown but training data (1) are given.

2.2 LUPI Paradigm of Learning

The LUPI paradigm describes a more complex model: given a set of iid triplets

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell), \quad x_i \in X, \quad x_i^* \in X^*, \quad y_i \in \{-1, +1\}, \tag{3}$$

generated according to a fixed but unknown probability measure $P(x, x^*, y)$, find, in a given set of indicator functions $f(x, \alpha), \alpha \in \Lambda$, the function $y = f(x, \alpha_*)$ that guarantees the smallest probability of incorrect classifications (2).

In the LUPI paradigm, we have exactly the same goal of minimizing (2) as in the classical paradigm, i.e., to find the best classification function in the admissible set. However, during the training stage, we have more information, i.e., we have triplets (x, x^*, y) instead of pairs (x, y) as in the classical paradigm. The additional information $x^* \in X^*$ belongs to space X^*, which is, generally speaking, different from X. For any element (x_i, y_i) of training example generated by Nature, Intelligent Teacher generates the privileged information x_i^* using some (unknown) conditional probability function $P(x_i^*|x_i)$.

In this paper, we first illustrate the work of these mechanisms on SVM algorithms; after that, we describe their general nature.

Since the additional information is available only for the training set and *is not* available for the test set, it is called *privileged information* and the new machine learning paradigm is called *Learning Using Privileged Information*.

Next, we consider three examples of privileged information that could be generated by Intelligent Teacher.

Example 1. Suppose that our goal is to find a rule that predicts the outcome y of a surgery in three weeks after it, based on information x available before the surgery. In order to find the rule in the classical paradigm, we use pairs (x_i, y_i) from previous patients.

However, for previous patients, there is also additional information x^* about procedures and complications during surgery, development of symptoms in one or two weeks after surgery, and so on. Although this information is not available *before* surgery, it does exist in historical data and thus can be used as privileged information in order to construct a rule that is better than the one obtained without using that information. The issue is how large an improvement can be achieved.

Example 2. Let our goal be to find a rule $y = f(x)$ to classify biopsy images x into two categories y: cancer ($y = +1$) and non-cancer ($y = -1$). Here images are in a pixel space X, and the classification rule has to be in the same space. However, the standard diagnostic procedure also includes a pathologist's report x^* that describes his/her impression about the image in a high-level holistic language X^* (for example, "aggressive proliferation of cells of type A among cells of type B" etc.).

The problem is to use the pathologist's reports x^* as privileged information (along with images x) in order to make a better classification rule for images x just in pixel space X. (Classification by a pathologist is a time-consuming procedure, so fast decisions during surgery should be made without consulting him or her).

Example 3. Let our goal be to predict the direction of the exchange rate of a currency at the moment t. In this problem, we have observations about the exchange rates before t, and we would like to predict if the rate will go up or down at the moment $t + \Delta$. However, in the historical market data we also have observations about exchange rates *after* moment t. Can this future-in-the-past privileged information be used for construction of a better prediction rule?

To summarize, privileged information is ubiquitous: it usually exists for almost any machine learning problem.

Section 4 describes the first mechanism that allows one to take advantage of privileged information by controlling Student's concepts of similarity between training examples. Section 5 describes examples where LUPI model uses similarity control mechanism. Section 6 is devoted to mechanism of knowledge transfer from space of privileged information X^* into decision space X.

However, first in the next Section we describe statistical properties of machine learning that enable the use of privileged information.

3. Statistical Analysis of the Rate of Convergence

According to the bounds developed in the VC theory (Vapnik and Chervonenkis, 1974), (Vapnik, 1998), the rate of convergence depends on two factors: how well the classification rule separates the training data

$$(x_1, y_1), ..., (x_\ell, y_\ell), \quad x \in R^n, \ y \in \{-1, +1\}, \tag{4}$$

and the VC dimension of the set of functions in which the rule is selected.

The theory has two distinct cases:

1. **Separable case:** there exists a function $f(x, \alpha_\ell)$ in the set of functions $f(x, \alpha), \alpha \in \Lambda$ with finite VC dimension h that separates the training data (4) without errors:

   $$y_i f(x_i, \alpha_\ell) > 0 \ \ \forall i = 1, ..., \ell.$$

   In this case, for the function $f(x, \alpha_\ell)$ that minimizes (down to zero) the empirical risk (on training set (4)), the bound

   $$P(yf(x, \alpha_\ell) \leq 0) < O^* \left(\frac{h - \ln \eta}{\ell} \right)$$

   holds true with probability $1 - \eta$, where $P(yf(x, \alpha_\ell) \leq 0)$ is the probability of error for the function $f(x, \alpha_\ell)$ and h is the VC dimension of the admissible set of functions. Here O^* denotes order of magnitude up to logarithmic factor.

2. **Non-separable case:** there is no function in $f(x, \alpha), \ \alpha \in \Lambda$ finite VC dimension h that can separate data (4) without errors. Let $f(x, \alpha_\ell)$ be a function that minimizes the number of errors on (4). Let $\nu(\alpha_\ell)$ be its error rate on training data (4). Then, according to the VC theory, the following bound holds true with probability $1 - \eta$:

   $$P(yf(x, \alpha_\ell) \leq 0) < \nu(\alpha_\ell) + O^* \left(\sqrt{\frac{h - \ln \eta}{\ell}} \right).$$

In other words, in the separable case, the rate of convergence has the order of magnitude $1/\ell$; in the non-separable case, the order of magnitude is $1/\sqrt{\ell}$. The difference between these rates[4] is huge: the same order of bounds requires 320 training examples versus 100,000 examples. Why do we have such a large gap?

3.1 Key Observation: SVM with Oracle Teacher

Let us try to understand why convergence rates for SVMs differ so much for separable and non-separable cases. Consider two versions of the SVM method for these cases.

SVM method first maps vectors x of space X into vectors z of space Z and then constructs a separating hyperplane in space Z. If training data can be separated with no error (the so-called separable case), SVM constructs (in space Z that we, for simplicity,

4. The VC theory also gives a more accurate estimate of the rate of convergence; however, the scale of difference remains essentially the same.

consider as an N-dimensional vector space R^N) a maximum margin separating hyperplane. Specifically, in the separable case, SVM minimizes the functional

$$\mathcal{T}(w) = (w, w)$$

subject to the constraints

$$(y_i(w, z_i) + b) \geq 1, \quad \forall i = 1, ..., \ell;$$

whereas in the non-separable case, SVM minimizes the functional

$$\mathcal{T}(w) = (w, w) + C \sum_{i=1}^{\ell} \xi_i$$

subject to the constraints

$$(y_i(w, z_i) + b) \geq 1 - \xi_i, \quad \forall i = 1, ..., \ell,$$

where $\xi_i \geq 0$ are slack variables. That is, in the separable case, SVM uses ℓ observations for estimation of N coordinates of vector w, whereas in the nonseparable case, SVM uses ℓ observations for estimation of $N + \ell$ parameters: N coordinates of vector w and ℓ values of slacks ξ_i. Thus, in the non-separable case, the number $N + \ell$ of parameters to be estimated is always larger than the number ℓ of observations; it does not matter here that most of slacks will be equal to zero: SVM still has to estimate all ℓ of them. Our guess is that the difference between the corresponding convergence rates is due to the number of parameters SVM has to estimate.

To confirm this guess, consider the SVM with *Oracle Teacher* (Oracle SVM). Suppose that Teacher can supply Student with the values of slacks as privileged information: during training session, Student is supplied with triplets

$$(x_1, \xi_1^0, y_1), ..., (x_\ell, \xi_\ell^0, y_\ell),$$

where $\xi_i^0,\ i = 1, ..., \ell$ are the slacks for the Bayesian decision rule. Therefore, in order to construct the desired rule using these triplets, the SVM has to minimize the functional

$$\mathcal{T}(w) = (w, w)$$

subject to the constraints

$$(y_i(w, z_i) + b) \geq r_i, \quad \forall i = 1, ..., \ell,$$

where we have denoted

$$r_i = 1 - \xi_i^0, \quad \forall i = 1, ..., \ell.$$

One can show that the rate of convergence is equal to $O^*(1/\ell)$ for Oracle SVM. The following (slightly more general) proposition holds true (Vapnik and Vashist, 2009).

Proposition 1. *Let $f(x, \alpha_0)$ be a function from the set of indicator functions $f(x, \alpha)$, with $\alpha \in \Lambda$ with VC dimension h that minimizes the frequency of errors (on this set) and let*

$$\xi_i^0 = \max\{0, (1 - f(x_i, \alpha_0))\}, \quad \forall i = 1, ..., \ell.$$

Then the error probability $p(\alpha_\ell)$ for the function $f(x, \alpha_\ell)$ that satisfies the constraints

$$y_i f(x, \alpha) \geq 1 - \xi_i^0, \quad \forall i = 1, ..., \ell$$

is bounded, with probability $1 - \eta$, as follows:

$$p(\alpha_\ell) \leq P(1 - \xi_0 < 0) + O^* \left(\frac{h - \ln \eta}{\ell} \right).$$

3.2 From Ideal Oracle to Real Intelligent Teacher

Of course, real Intelligent Teacher cannot supply slacks: Teacher does not know them. Instead, Intelligent Teacher can do something else, namely:

1. define a space X^* of (correcting) slack functions (it can be different from the space X of decision functions);

2. define a set of real-valued slack functions $f^*(x^*, \alpha^*),\ x^* \in X^*,\ \alpha^* \in \Lambda^*$ with VC dimension h^*, where approximations

$$\xi_i = f^*(x, \alpha^*)$$

of the slack functions[5] are selected;

3. generate privileged information for training examples supplying Student, instead of pairs (4), with triplets

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell). \tag{5}$$

During training session, the algorithm has to simultaneously estimate two functions using triplets (5): the decision function $f(x, \alpha_\ell)$ and the slack function $f^*(x^*, \alpha_\ell^*)$. In other words, the method minimizes the functional

$$\mathcal{T}(\alpha^*) = \sum_{i=1}^{\ell} \max\{0, f^*(x_i^*, \alpha^*)\} \tag{6}$$

subject to the constraints

$$y_i f(x_i, \alpha) > -f^*(x_i^*, \alpha^*), \quad i = 1, ..., \ell. \tag{7}$$

Let $f(x, \alpha_\ell)$ and $f^*(x^*, \alpha_\ell^*)$ be functions that solve this optimization problem. For these functions, the following proposition holds true (Vapnik and Vashist, 2009).

5. Note that slacks ξ_i introduced for the SVM method can be considered as a realization of some function $\xi = \xi(x, \beta_0)$ from a large set of functions (with infinite VC dimension). Therefore, generally speaking, the classical SVM approach can be viewed as estimation of two functions: (1) the decision function, and (2) the slack function, where these functions are selected from two different sets, with finite and infinite VC dimensions, respectively. Here we consider both sets with finite VC dimensions.

Proposition 2. *The solution $f(x, \alpha_\ell)$ of optimization problem (6), (7) satisfies the bounds*

$$P(yf(x, \alpha_\ell) < 0) \leq P(f^*(x^*, \alpha_\ell^*) \geq 0) + O^* \left(\frac{h + h^* - \ln \eta}{\ell} \right)$$

with probability $1 - \eta$, where h and h^* are the VC dimensions of the set of decision functions $f(x, \alpha),\ \alpha \in \Lambda$, and the set of correcting functions $f^*(x^*, \alpha^*),\ \alpha^* \in \Lambda^*$, respectively.

According to Proposition 2, in order to estimate the rate of convergence to the best possible decision rule (in space X) one needs to estimate the rate of convergence of $P\{f^*(x^*, \alpha_\ell^*) \geq 0\}$ to $P\{f^*(x^*, \alpha_0^*) \geq 0\}$ for the best rule $f^*(x^*, \alpha_0^*)$ in space X^*. Note that both the space X^* and the set of functions $f^*(x^*, \alpha_\ell^*), \alpha^* \in \Lambda^*$ are suggested by Intelligent Teacher that tries to choose them in a way that facilitates a fast rate of convergence. The guess is that a really Intelligent Teacher can indeed do that.

As shown in the VC theory, in standard situations, the uniform convergence has the order $O^*(\sqrt{h^*/\ell})$, where h^* is the VC dimension of the admissible set of correcting functions $f^*(x^*, \alpha^*)$, $\alpha* \in \Lambda^*$. However, for special privileged space X^* and corresponding functions $f^*(x^*, \alpha^*), \alpha^* \in \Lambda^*$, the convergence can be faster (as $O^*([1/\ell]^\delta),\ \delta > 1/2$).

A well-selected privileged information space X^* and Teacher's explanation $P(x^*|x)$ along with sets $\{f(x, \alpha_\ell), \alpha \in \Lambda\}$ and $\{f^*(x^*, \alpha^*), \alpha^* \in \Lambda^*\}$ engender a convergence that is faster than the standard one. The skill of Intelligent Teacher is being able to select of the proper space X^*, generator $P(x^*|x)$, set of functions $f(x, \alpha_\ell), \alpha \in \Lambda$, and set of functions $f^*(x^*, \alpha^*), \alpha^* \in \Lambda^*$: that is what differentiates good teachers from poor ones.

4. Similarity Control in LUPI Paradigm

4.1 $\text{SVM}_\Delta+$ for Similarity Control in LUPI Paradigm

In this section, we extend SVM method of function estimation to the method called SVM+, which allows one to solve machine learning problems in the LUPI paradigm (Vapnik, 2006). The $\text{SVM}_\varepsilon+$ method presented below is a reinforced version of the one described in Vapnik (2006) and used in Vapnik and Vashist (2009).

Consider the model of learning with Intelligent Teacher: given triplets

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell),$$

find in the given set of functions the one that minimizes the probability of incorrect classifications in space X.

As in standard SVM, we map vectors $x_i \in X$ onto the elements z_i of the Hilbert space Z, and map vectors x_i^* onto elements z_i^* of another Hilbert space Z^* obtaining triples

$$(z_1, z_1^*, y_1), ..., (z_\ell, z_\ell^*, y_\ell).$$

Let the inner product in space Z be (z_i, z_j), and the inner product in space Z^* be (z_i^*, z_j^*).

Consider the set of decision functions in the form

$$f(x) = (w, z) + b,$$

where w is an element in Z, and consider the set of correcting functions in the form

$$\xi^*(x^*, y) = [y((w^*, z^*) + b^*)]_+,$$

where w^* is an element in Z^* and $[u]_+ = \max\{0, u\}$.

Our goal is to we minimize the functional

$$\mathcal{T}(w, w^*, b, b^*) = \frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{\ell}[y_i((w^*, z_i^*) + b^*)]_+$$

subject to the constraints

$$y_i[(w, z_i) + b] \geq 1 - [y_i((w^*, z_i^*) - b^*)]_+.$$

The structure of this problem mirrors the structure of the primal problem for standard SVM. However, due to the elements $[u_i]_+ = \max\{0, u_i\}$ that define both the objective function and the constraints here we faced non-linear optimization problem.

To find the solution of this optimization problem, we approximate this non-linear optimization problem with the following quadratic optimization problem: minimize the functional

$$\mathcal{T}(w, w^*, b, b^*) = \frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{\ell}[y_i((w^*, z_i^*) + b^*) + \zeta_i] + \Delta C\sum_{i=1}^{\ell}\zeta_i \qquad (8)$$

(here $\Delta > 0$ is the parameter of approximation[6]) subject to the constraints

$$y_i((w, z_i) + b) \geq 1 - y_i((w^*, z^*) + b^*) - \zeta_i, \quad i = 1, ..., \ell, \qquad (9)$$

the constraints

$$y_i((w^*, z_i^*) + b^*) + \zeta_i \geq 0, \quad \forall i = 1, ..., \ell, \qquad (10)$$

and the constraints

$$\zeta_i \geq 0, \quad \forall i = 1, ..., \ell. \qquad (11)$$

To minimize the functional (8) subject to the constraints (10), (11), we construct the Lagrangian

$$\mathcal{L}(w, b, w^*, b^*, \alpha, \beta) = \qquad (12)$$

$$\frac{1}{2}[(w, w) + \gamma(w^*, w^*)] + C\sum_{i=1}^{\ell}[y_i((w^*, z_i^*) + b^*) + (1 + \Delta)\zeta_i] - \sum_{i=1}^{\ell}\nu_i\zeta_i -$$

$$\sum_{i=1}^{\ell}\alpha_i\left[y_i[(w, z_i) + b] - 1 + [y_i((w^*, z_i^*) + b^*) + \zeta_i]\right] - \sum_{i=1}^{\ell}\beta_i[y_i((w^*, z_i^*) + b^*) + \zeta_i],$$

where $\alpha_i \geq 0,\ \beta_i \geq 0,\ \nu_i \geq 0,\ i = 1, ..., \ell$ are Lagrange multipliers.

To find the solution of our quadratic optimization problem, we have to find the saddle point of the Lagrangian (the minimum with respect to w, w^*, b, b^* and the maximum with respect to $\alpha_i, \beta_i, \nu_i,\ i = 1, ..., \ell$).

6. In Vapnik (2006), parameter Δ was set at a sufficiently large value.

The necessary conditions for minimum of (12) are

$$\frac{\partial \mathcal{L}(w, b, w^*, b^*, \alpha, \beta)}{\partial w} = 0 \implies w = \sum_{i=1}^{\ell} \alpha_i y_i z_i \tag{13}$$

$$\frac{\partial \mathcal{L}(w, b, w^*, b^*, \alpha, \beta)}{\partial w^*} = 0 \implies w^* = \frac{1}{\gamma} \sum_{i=1}^{\ell} y_i (\alpha_i + \beta_i - C) z_i^* \tag{14}$$

$$\frac{\partial \mathcal{L}(w, b, w^*, b^*, \alpha, \beta)}{\partial b} = 0 \implies \sum_{i=1}^{\ell} \alpha_i y_i = 0 \tag{15}$$

$$\frac{\partial \mathcal{L}(w, b, w^*, b^*, \alpha, \beta)}{\partial b^*} = 0 \implies \sum_{i=1}^{\ell} y_i (C - \alpha_i - \beta_i) = 0 \tag{16}$$

$$\frac{\partial \mathcal{L}(w, b, w^*, b^*, \alpha, \beta)}{\partial \zeta_i} = 0 \implies \alpha_i + \beta_i + \nu_i = (C + \Delta C) \tag{17}$$

Substituting the expressions (13) in (12) and, taking into account (14), (15), (16), and denoting $\delta_i = C - \beta_i$, we obtain the functional

$$\mathcal{L}(\alpha, \delta) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} (z_i, z_j) y_i y_j \alpha_i \alpha_j - \frac{1}{2\gamma} \sum_{i,j=1}^{\ell} (\delta_i - \alpha_i)(\delta_j - \alpha_j)(z_i^*, z_j^*) y_i y_j.$$

To find its saddle point, we have to maximize it subject to the constraints[7]

$$\sum_{i=1}^{\ell} y_i \alpha_i = 0 \tag{18}$$

$$\sum_{i=1}^{\ell} y_i \delta_i = 0 \tag{19}$$

$$0 \le \delta_i \le C, \quad i = 1, ..., \ell \tag{20}$$

$$0 \le \alpha_i \le \delta_i + \Delta C, \quad i = 1, ..., \ell \tag{21}$$

Let vectors α^0, δ^0 be a solution of this optimization problem. Then, according to (13) and (14), one can find the approximations to the desired decision function

$$f(x) = (w_0, z_i) + b = \sum_{i=1}^{\ell} \alpha_i^* y_i (z_i, z) + b$$

and to the slack function

$$\xi^*(x^*, y) = y_i((w_0^*, z_i^*) + b^*) + \zeta = \sum_{i=1}^{\ell} y_i (\alpha_i^0 - \delta_i^0)(z_i^*, z^*) + b^* + \zeta.$$

7. In SVM+, instead of constraints (21), the constraints $\alpha_i \ge 0$ were used.

The Karush-Kuhn-Tacker conditions for this problem are

$$\begin{cases} \alpha_i^0[y_i[(w_0, z_i) + b + (w_0^*, z_i^*) + b^*] + \zeta_i - 1] = 0 \\ (C - \delta_i^0)[(w_0^*, z_i^*) + b^* + \zeta_i] = 0 \\ \nu_i^0 \zeta_i = 0 \end{cases}$$

Using these conditions, one obtains the value of constant b as

$$b = 1 - y_k(w^0, z_k) = 1 - y_k \left[\sum_{i=1}^{\ell} \alpha_i^0 (z_i, z_k) \right],$$

where (z_k, z_k^*, y_k) is a triplet for which $\alpha_k^0 \neq 0$, $\delta_k^0 \neq C$, $z_i \neq 0$.

As in standard SVM, we use the inner product (z_i, z_j) in space Z in the form of Mercer kernel $K(x_i, x_j)$ and inner product (z_i^*, z_j^*) in space Z^* in the form of Mercer kernel $K^*(x_i^*, x_j^*)$. Using these notations, we can rewrite the SVM$_\Delta$+ method as follows: the decision rule in X space has the form

$$f(x) = \sum_{i=1}^{\ell} y_i \alpha_i^0 K(x_i, x) + b,$$

where $K(\cdot, \cdot)$ is the Mercer kernel that defines the inner product for the image space Z of space X (kernel $K^*(\cdot, \cdot)$ for the image space Z^* of space X^*) and α^0 is a solution of the following dual space quadratic optimization problem: maximize the functional

$$\mathcal{L}(\alpha, \delta) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \frac{1}{2\gamma} \sum_{i,j=1}^{\ell} y_i y_j (\alpha_i - \delta_i)(\alpha_j - \delta_j) K^*(x_i^*, x_j^*)$$

subject to constraints (18) – (21).

Remark. Note that if $\delta_i = \alpha_i$ or $\Delta = 0$, the solution of our optimization problem becomes equivalent to the solution of the standard SVM optimization problem, which maximizes the functional

$$\mathcal{L}(\alpha, \delta) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(x_i, x_j)$$

subject to constraints (18) – (21) where $\delta_i = \alpha_i$.

Therefore, the difference between SVM$_\Delta$+ and SVM solutions is defined by the last term in objective function (8). In SVM method, the solution depends only on the values of pairwise similarities between training vectors defined by the Gram matrix K of elements $K(x_i, x_j)$ (which defines similarity between vectors x_i and x_j). The SVM$_\Delta$+ solution is defined by objective function (8) that uses two expressions of similarities between observations: one ($K(x_i, x_j)$ for x_i and x_j) that comes from space X and another one ($K^*(x_i^*, x_j^*)$ for x_i^* and x_j^*) that comes from space of privileged information X^*. That is how Intelligent Teacher changes the optimal solution by correcting the concepts of similarity.

The last term in equation (8) defines the instrument for Intelligent Teacher to control the concept of similarity of Student.

Efficient computational implementation of this SVM+ algorithm for classification and its extension for regression can be found in Pechyony et al. (2010) and Vapnik and Vashist (2009), respectively.

4.1.1 Simplified Approach

The described method $SVM_\Delta+$ requires to minimize the quadratic form $\mathcal{L}(\alpha, \delta)$ subject to constraints (18) – (21). For large ℓ it can be a challenging computational problem. Consider the following approximation. Let

$$f^*(x^*, \alpha^*_\ell) = \sum_{i=1}^{\ell} \alpha^*_i K^*(x^*_i, x) + b^*$$

be be an SVM solution in space X^* and let

$$\xi^*_i = [1 - f^*(x^*, \alpha^*_\ell) - b^*]_+$$

be the corresponding slacks. Let us use the linear function

$$\xi_i = t\xi^*_i + \zeta_i, \quad \zeta_i \geq 0$$

as an approximation of slack function in space X. Now we minimize the functional

$$(w, w) + C \sum_{i=1}^{\ell} (t\xi^*_i + (1 + \Delta)\zeta_i), \quad \Delta \geq 0$$

subject to the constraints

$$y_i((w, z_i) + b) > 1 - t\xi^*_i + \zeta_i,$$
$$t > 0, \quad \zeta_i \geq 0, \ i = 1, ..., \ell$$

(here z_i is Mercer mapping of vectors x_i in RKHS).

The solution of this quadratic optimization problem defines the function

$$f(x, \alpha_\ell) = \sum_{i=1}^{\ell} \alpha_i K(x_i, x) + b,$$

where α is solution of the following dual problem: maximize the functional

$$R(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to the constraints

$$\sum_{i=1}^{\ell} y_i \alpha_i = 0$$

$$\sum_{i=1}^{\ell} \alpha_i \xi^*_i \leq C \sum_{i=1}^{\ell} \xi^*_i$$

$$0 \leq \alpha_i \leq (1 + \Delta)C, \quad i = 1, ..., \ell$$

4.2 General Form of Similarity Control in LUPI Paradigm

Consider the following two sets of functions: the set $f(x,\alpha), \alpha \in \Lambda$ defined in space X and the set $f^*(x^*,\alpha^*), \alpha^* \in \Lambda^*$, defined in space X^*. Let a non-negative convex functional $\Omega(f) \geq 0$ be defined on the set of functions $f(x,\alpha), \alpha \in \Lambda$, while a non-negative convex functional $\Omega^*(f^*) \geq 0$ be defined on the set of functions $f(x^*,\alpha^*), \alpha^* \in \Lambda^*$. Let the sets of functions $\theta(f(x,\alpha)), \alpha \in \Lambda$, and $\theta(f(x^*,\alpha^*)), \alpha^* \in \Lambda^*$, which satisfy the corresponding bounded functionals

$$\Omega(f) \leq C_k$$

$$\Omega^*(f^*) \leq C_k,$$

have finite VC dimensions h_k and h_k, respectively. Consider the structures

$$S_1 \subset ... \subset S_m ....$$

$$S_1^* \subset ... \subset S_m^* ...$$

defined on corresponding sets of functions.

Let iid observations of triplets

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell)$$

be given. Our goal is to find the function $f(x,\alpha_\ell)$ that minimizes the probability of the test error.

To solve this problem, we minimize the functional

$$\sum_{i=1}^{\ell} f^*(x_i^*, \alpha)$$

subject to constraints

$$y_i[f(x,\alpha) + f(x^*,\alpha^*)] > 1$$

and the constraint

$$\Omega(f) + \gamma\Omega(f^*) \leq C_m$$

(we assume that our sets of functions are such that solutions exist).

Then, for any fixed sets S_k and S_k^*, the VC bounds hold true, and minimization of these bounds with respect to both sets S_k and S_k^* of functions and the functions $f(x,\alpha_\ell)$ and $f^*(x^(,\alpha_\ell^*)$ in these sets is a realization of universally consistent SRM principle.

The sets of functions defined in previous section by the Reproducing Kernel Hilbert Space satisfy this model since any subset of functions from RKHS with bounded norm has finite VC dimension according to the theorem about VC dimension of linear bounded functions in Hilbert space[8].

8. This theorem was proven in mid-1970s (Vapnik and Chervonenkis, 1974) and generalized for Banach spaces in early 2000s (Gurvits, 2001; Vapnik, 1998).

5. Transfer of Knowledge Obtained in Privileged Information Space to Decision Space

In this section, we consider the second important mechanism of Teacher-Student interaction: using privileged information for knowledge transfer from Teacher to Student[9].

Suppose that Intelligent Teacher has some knowledge about the solution of a specific pattern recognition problem and would like to transfer this knowledge to Student. For example, Teacher can reliably recognize cancer in biopsy images (in a pixel space X) and would like to transfer this skill to Student.

Formally, this means that Teacher has some function $y = f_0(x)$ that distinguishes cancer ($f_0(x) = +1$ for cancer and $f_0(x) = -1$ for non-cancer) in the pixel space X. Unfortunately, Teacher does not know this function explicitly (it only exists as a neural net in Teacher's brain), so how can Teacher transfer this construction to Student? Below, we describe a possible mechanism for solving this problem; we call this mechanism *knowledge transfer*.

Suppose that Teacher believes in some theoretical model on which the knowledge of Teacher is based. For cancer model, he or she believes that it is a result of uncontrolled multiplication of the cancer cells (cells of type B) that replace normal cells (cells of type A). Looking at a biopsy image, Teacher tries to generate privileged information that reflects his or her belief in development of such process; Teacher may describe the image as:

Aggressive proliferation of cells of type B into cells of type A.

If there are no signs of cancer activity, Teacher may use the description

Absence of any dynamics in the of standard picture.

In uncertain cases, Teacher may write

There exist small clusters of abnormal cells of unclear origin.

In other words, Teacher has developed a special language that is appropriate for description x_i^* of cancer development based on the model he or she believes in. Using this language, Teacher supplies Student with privileged information x_i^* for the image x_i by generating training triplets

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell). \tag{22}$$

The first two elements of these triplets are descriptions of an image in two languages: in language X (vectors x_i in pixel space), and in language X^* (vectors x_i^* in the space of privileged information), developed for Teacher's understanding of cancer model.

Note that the language of pixel space is universal (it can be used for description of many different visual objects; for example, in the pixel space, one can distinguish between male and female faces), while the language used for describing privileged information is very specific: it reflects just a model of cancer development. This has an important consequence:

9. In machine learning, transfer learning refers to the framework, where experience obtained for solving one problem is used (with proper modifications) for solving another problem, related to the previous one; both problems are assumed to be in the same space, with only some parameters being changed. The knowledge transfer considered here is different: it denotes the transfer of knowledge obtained in one (privileged) space to another (decision) space.

the set of admissible functions in space X has to be rich (has a large VC dimension), while the set of admissible functions in space X^* may be not rich (has a small VC dimension).

One can consider two related pattern recognition problems using triplets (22):

1. The problem of constructing a rule $y = f(x)$ for classification of biopsy in the pixel space X using data
$$(x_1, y_1), ..., (x_\ell, y_\ell). \tag{23}$$

2. The problem of constructing a rule $y = f^*(x^*)$ for classification of biopsy in the space X^* using data
$$(x_1^*, y_1), ..., (x_\ell^*, y_\ell). \tag{24}$$

Suppose that language X^* is so good that it allows to create a rule $y = f_\ell^*(x^*)$ that classifies vectors x^* corresponding to vectors x with the same level of accuracy as the best rule $y = f_\ell(x)$ for classifying data in the pixel space[10].

In the considered example, the VC dimension of the admissible rules in a special space X^* is much smaller than the VC dimension of the admissible rules in the universal space X and, since the number of examples ℓ is the same in both cases, the bounds on the error rate for the rule $y = f_\ell^*(x^*)$ in X^* will be better[11] than those for the rule $y = f_\ell(x)$ in X. Generally speaking, the knowledge transfer approach can be applied if the classification rule $y = f_\ell^*(x^*)$ is more accurate than the classification rule $y = f_\ell(x)$ (the empirical error in privileged space is smaller than the empirical error in the decision space).

The following problem arises: how one can use the knowledge of the rule $y = f_\ell^*(x^*)$ in space X^* to improve the accuracy of the desired rule $y = f_\ell(x)$ in space X?

5.1 Knowledge Representation for SVMs

To answer this question, we formalize the concept of representation of the knowledge about the rule $y = f_\ell^*(x^*)$.

Suppose that we are looking for our rule in Reproducing Kernel Hilbert Space (RKHS) associated with kernel $K^*(x_i^*, x^*)$. According to Representer Theorem (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001), such rule has the form

$$f_\ell^*(x^*) = \sum_{i=1}^{\ell} \gamma_i K^*(x_i^*, x^*) + b, \tag{25}$$

where $\gamma_i,\ i = 1, ..., \ell$ and b are parameters.

Suppose that, using data (24), we found a good rule (25) with coefficients $\gamma_i = \gamma_i^*,\ i = 1, ..., \ell$ and $b = b^*$. This is now the knowledge about our classification problem. Let us formalize the description of this knowledge.

Consider three elements of knowledge representation used in Artificial Intelligence (Brachman and Levesque, 2004):

10. The rule constructed in space X^* cannot be better than the best possible rule in space X, since all information originates in space X.
11. According to VC theory, the guaranteed bound on accuracy of the chosen rule depends only on two factors: the frequency of errors on training set and the VC dimension of admissible set of functions.

1. Fundamental elements of knowledge.
2. Frames (fragments) of the knowledge.
3. Structural connections of the frames (fragments) in the knowledge.

We call the *fundamental elements of the knowledge* a limited number of vectors $u_1^*,...,u_m^*$ from space X^* that can approximate well the main part of rule (25). It could be the support vectors or the smallest number of vectors[12] $u_i \in X^*$:

$$f_\ell^*(x^*) - b = \sum_{i=1}^{\ell} \gamma_i^* K^*(x_i^*, x^*) \approx \sum_{k=1}^{m} \beta_k^* K^*(u_k^*, x^*). \tag{26}$$

Let us call the functions $K^*(u_k^*, x^*),\ k = 1, ..., m$ the *frames* (fragments) of knowledge. Our knowledge

$$f_\ell^*(x^*) = \sum_{k=1}^{m} \beta_k^* K^*(u_k^*, x^*) + b$$

is defined as a linear combination of the frames.

5.1.1 Scheme of Knowledge Transfer Between Spaces

In the described terms, knowledge transfer from X^* into X requires the following:

1. To find the fundamental elements of knowledge $u_1^*, ..., u_m^*$ in space X^*.
2. To find frames (m functions) $K^*(u_1^*, x^*), ..., K^*(u_m^*, x^*)$ in space X^*.
3. To find the functions $\phi_1(x), ..., \phi_m(x)$ in space X such that

$$\phi_k(x_i) \approx K^*(u_k^*, x_i^*) \tag{27}$$

holds true for almost all pairs (x_i, x_i^*) generated by Intelligent Teacher that uses some (unknown) generator $P(x^*, x) = P(x^*|x)P(x)$.

Note that the capacity of the set of functions from which $\phi_k(x)$ are to be chosen can be smaller than that of the capacity of the set of functions from which the classification function $y = f_\ell(x)$ is chosen (function $\phi_k(x)$ approximates just one fragment of knowledge, not the entire knowledge, as function $y = f_\ell^*(x^*)$, which is a linear combination (26) of frames). Also, as we will see in the next section, estimates of all the functions $\phi_1(x), ..., \phi_m(x)$ are done using different pairs as training sets of the same size ℓ. That is, we hope that transfer of m fragments of knowledge from space X^* into space X can be done with higher accuracy than estimating the function $y = f_\ell(x)$ from data (23).

After finding images of frames in space X, the knowledge about the rule obtained in space X^* can be approximated in space X as

$$f_\ell(x) \approx \sum_{k=1}^{m} \delta_k \phi_k(x) + b^*,$$

where coefficients $\delta_k = \gamma_k$ (taken from (25)) if approximations (27) are accurate. Otherwise, coefficients δ_k can be estimated from the training data, as shown in Section 6.3.

12. In machine learning, it is called the reduced number of support vectors (Burges, 1996).

5.1.2 Finding the Smallest Number of Fundamental Elements of Knowledge

Let our functions ϕ belong to RKHS associated with the kernel $K^*(x_i^*, x^*)$, and let our knowledge be defined by an SVM method in space X^* with support vector coefficients α_i. In order to find the smallest number of fundamental elements of knowledge, we have to minimize (over vectors $u_1^*, ..., u_m^*$ and values $\beta_1, ..., \beta_m$) the functional

$$R(u_1^*, ..., u_m^*; \beta_1, ..., \beta_m) = \tag{28}$$

$$\left\| \sum_{i=1}^{\ell} y_i \alpha_i K^*(x_i^*, x^*) - \sum_{s=1}^{m} \beta_s K^*(u_s^*, x^*) \right\|^2_{RKHS} =$$

$$\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K^*(x_i^*, x_j^*) - 2 \sum_{i=1}^{\ell} \sum_{s=1}^{m} y_i \alpha_i \beta_s K^*(x_i^*, u_s^*) + \sum_{s,t=1}^{m} \beta_s \beta_t K^*(u_s^*, u_t^*).$$

The last equality was derived from the following property of the inner product for functions in RKHS (Kimeldorf and Wahba, 1971; Schölkopf et al., 2001):

$$\big(K^*(x_i^*, x^*), K(x_j^*, x^*)\big)_{RKHS} = K^*(x_i^*, x_j^*).$$

5.1.3 Smallest Number of Fundamental Elements of Knowledge for Homogeneous Quadratic Kernel

For general kernel functions $K^*(\cdot, \cdot)$, minimization of (28) is a difficult computational problem. However, for the special homogeneous quadratic kernel

$$K^*(x_i^*, x_j^*) = (x_i^*, x_j^*)^2,$$

this problem has a simple exact solution (Burges, 1996). For this kernel, we have

$$R = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j (x_i^*, x_j^*)^2 - 2 \sum_{i=1}^{\ell} \sum_{s=1}^{m} y_i \alpha_i \beta_s (x_i^*, u_s^*)^2 + \sum_{s,t=1}^{m} \beta_s \beta_t (u_s^*, u_t^*)^2. \tag{29}$$

Let us look for solution in set of orthonormal vectors $u_i^*, ..., u_m^*$ for which we can rewrite (29) as follows

$$\hat{R} = \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j (x_i^*, x_j^*)^2 - 2 \sum_{i=1}^{\ell} \sum_{s=1}^{m} y_i \alpha_i \beta_s (x_i^*, u_s^*)^2 + \sum_{s=1}^{m} \beta_s^2 (u_s^*, u_s^*)^2. \tag{30}$$

Taking derivative of $\hat{R}$ with respect to u_k^*, we obtain that the solutions u_k^*, $k = 1, ..., m$ have to satisfy the equations

$$\frac{d\hat{R}}{du_k} = -2\beta_k \sum_{i=1}^{\ell} y_i \alpha_i x_i^* x_i^{*T} u_k^* + 2\beta_k^2 u_k^* = 0.$$

Introducing notation

$$S = \sum_{i=1}^{\ell} y_i \alpha_i x_i^* x_i^{*T}, \tag{31}$$

we conclude that the solutions satisfy the equation

$$Su_k^* = \beta_k u_k^*, \quad k = 1, ..., m.$$

Let us chose from the set $u_1^*, ..., u_m^*$ of eigenvectors of the matrix S the vectors corresponding to the largest in absolute values eigenvalues $\beta_1, \ldots, \beta_m$, which are coefficients of expansion of the classification rule on the frames $(u_k, x^*)^2, \quad k = 1, \ldots, m$.

Using (31), one can rewrite the functional (30) in the form

$$\hat{R} = \mathbf{1}^{\mathbf{T}} S_2 \mathbf{1} - \sum_{k=1}^{m} \beta_k^2, \tag{32}$$

where we have denoted by S_2 the matrix obtained from S with its elements $s_{i,j}$ replaced with $s_{i,j}^2$, and by $\mathbf{1}$ we have denoted the $(\ell \times 1)$-dimensional matrix of ones.

Therefore, in order to find the fundamental elements of knowledge, one has to solve the eigenvalue problem for $(n \times n)$-dimensional matrix S and then select an appropriate number of eigenvectors corresponding to eigenvalues with largest absolute values. One chooses such m eigenvectors for which functional (32) is small. The number m does not exceed n (the dimensionality of matrix S).

5.1.4 Finding Images of Frames in Space X

Let us call the conditional expectation function

$$\phi_k(x) = \int K^*(u_k^*, x^*) p(x^*|x)\, dx^*$$

the image of frame $K^*(u_k^*, x^*)$ in space X. To find m image functions $\phi_k(x)$ of the frames $K(u_k^*, x^*),\ k = 1, ..., m$ in space X, we solve the following m regression estimation problems: find the regression function $\phi_k(x)$ in X, $k = 1, \ldots, m$, using data

$$(x_1, K^*(u_k^*, x_1^*)), ..., (x_\ell, K^*(u_k^*, x_\ell^*)), \quad k = 1, \ldots, m, \tag{33}$$

where pairs (x_i, x_i^*) belong to elements of training triplets (22).

Therefore, using fundamental elements of knowledge $u_1^*, ...u_m^*$ in space X^*, the corresponding frames $K^*(u_1^*, x^*), ..., K^*(u_m^*, x^*)$ in space X^*, and the training data (33), one constructs the transformation of the space X into m-dimensional feature space[13]

$$\phi(x) = (\phi_1(x), ...\phi_m(x)),$$

where k-th coordinate of vector function $\phi(x)$ is defined as $\phi_k = \phi_k(x)$.

5.1.5 Algorithms for Knowledge Transfer

1. Suppose that our regression functions can be estimated accurately: for a sufficiently small $\varepsilon > 0$ the inequalities

$$|\phi_k(x_i) - K^*(u_k^*, x_i^*)| < \varepsilon, \quad \forall k = 1, ..., m \quad \text{and} \quad \forall i = 1, ..., \ell$$

13. One can choose any subset from $(m + n)$-dimensional space $(\phi_1(x), ...\phi_m(x)), x^1, ..., x^n)$.

hold true for almost all pairs (x_i, x_i^*) generated according to $P(x^*|y)$. Then the approximation of our knowledge in space X is

$$f(x) = \sum_{k=1}^{m} \beta_k^* \phi_k(x) + b^*,$$

where $\beta_k^*,\ k = 1, ..., m$ are eigenvalues corresponding to eigenvectors $u_1^*, ..., u_m^*$.

2. If, however, ε is not too small, one can use privileged information to employ both mechanisms of intelligent learning: controlling similarity between training examples and knowledge transfer.

In order to describe this method, we denote by vector ϕ_i the m-dimensional vector with coordinates

$$\phi_i = (\phi_1(x_i), ..., \phi_m(x_i))^T.$$

Consider the following problem of intelligent learning: given training triplets

$$(\phi_1, x_1^*, y_1), ..., (\phi_\ell, x_\ell^*, y_\ell),$$

find the decision rule

$$f(\phi(x)) = \sum_{i=1}^{\ell} y_i \hat{\alpha}_i \hat{K}(\phi_i, \phi) + b. \tag{34}$$

Using SVM$_\Delta$+ algorithm described in Section 4, we can find the coefficients of expansion $\hat{\alpha}_i$ in (34). They are defined by the maximum (over $\hat{\alpha}$ and δ) of the functional

$$R(\hat{\alpha}, \delta) = \sum_{i=1}^{\ell} \hat{\alpha}_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \hat{\alpha}_i \hat{\alpha}_j \hat{K}(\phi_i, \phi_j) - \frac{1}{2\gamma} \sum_{i,j=1}^{\ell} y_i y_j (\hat{\alpha}_i - \delta_i)(\hat{\alpha}_j - \delta_j) K^*(x_i^*, x_j^*)$$

subject to the equality constraints

$$\sum_{i=1}^{\ell} \hat{\alpha}_i y_i = 0, \quad \sum_{i=1}^{\ell} \hat{\alpha}_i = \sum_{i=1}^{\ell} \delta_i$$

and the inequality constraints

$$0 \le \hat{\alpha}_i \le \delta_i + \Delta C, \quad 0 \le \delta_i \le C, \quad i = 1, \ldots, \ell$$

(see Section 4).

5.2 General Form of Knowledge Transfer

One can use many different ideas to represent knowledge obtained in space X^*. The main factors of these representations are concepts of fundamental elements of the knowledge. They could be, for example, just the support vectors (if the number of support vectors is not too big) or coordinates (features) $x^{t*},\ t = 1, \ldots, d$ of d-dimensional privileged space X^* (if the number of these features not too big). In the latter case, the small number of fundamental elements of knowledge would be composed of features x^{*k} in the privileged space that can be then approximated by regression functions $\phi_k(x)$. In general, using

privileged information it is possible to try transfer set of useful features for rule in X^* space into their image in X space.

The space where depiction rule is constructed can contain both features of space X and new features defined by the regression functions. The example of knowledge transfer described further in subsection 5.5 is based on this approach.

In general, the idea is to specify small amount important feature in privileged space and then try to transfer them (say, using non-linear regression technique) in decision space to construct useful (additional) features in decision space.

Note that in SVM framework, with the quadratic kernel the minimal number m of fundamental elements (features) does not exceed the dimensionality of space X^* (often, m is much smaller than dimensionality. This was demonstrated in multiple experiments with digit recognition by Burges 1996): in order to generate the same level of accuracy of the solution, it was sufficient to use m elements, where the value of m was at least 20 times smaller than the corresponding number of support vectors.

5.3 Kernels Involved in Intelligent Learning

In this paper, among many possible Mercer kernels (positive semi-definite functions), we consider the following three types:

1. Radial Basis Function (RBF) kernel:

$$K_{RBF_\sigma}(x, y) = \exp\{-\sigma^2(x - y)^2\}.$$

2. INK-spline kernel. Kernel for spline of order zero with infinite number of knots is defined as

$$K_{INK_0}(x, y) = \prod_{k=1}^{d}(\min(x^k, y^k) + \delta)$$

(δ is a free parameter) and kernel of spline of order one with infinite number of knots is defined in the non-negative domain and has the form

$$K_{INK_1}(x, y) = \prod_{k=1}^{d}\left(\delta + x^k y^k + \frac{|x^k - y^k| \min\{x_k, y^k\}}{2} + \frac{(\min\{x^k, y^k\})^3}{3}\right)$$

where $x^k \geq 0$ and $y^k \geq 0$ are k coordinates of d-dimensional vector x.

3. Homogeneous quadratic kernel

$$K_{Pol_2} = (x, y)^2,$$

where (x, y) is the inner product of vectors x and y.

The RBF kernel has a free parameter $\sigma > 0$; two other kernels have no free parameters. That was achieved by fixing a parameter in more general sets of functions: the degree of polynomial was chosen to be 2, and the order of INK-splines was chosen to be 1.

It is easy to introduce kernels for any degree of polynomials and any order of INK-splines. Experiments show excellent properties of these three types of kernels for solving many machine learning problems. These kernels also can be recommended for methods that use both mechanisms of Teacher-Student interaction.

5.4 Knowledge Transfer for Statistical Inference Problems

The idea of privileged information and knowledge transfer can be also extended to Statistical Inference problems considered in Vapnik and Izmailov (2015a) and Vapnik et al. (2015).

For simplicity, consider the problem of estimation[14] of conditional probability $P(y|x)$ from iid data

$$(x_1, y_1), ..., (x_\ell, y_\ell), \quad x \in X,\ y \in \{0, 1\}, \tag{35}$$

where vector $x \in X$ is generated by a fixed but unknown distribution function $P(x)$ and binary value $y \in \{0, 1\}$ is generated by an unknown conditional probability function $P(y = 1|x)$ (similarly, $P(y = 0|x) = 1 - P(y = 1|x)$); this is the function we would like to estimate.

As shown in Vapnik and Izmailov (2015a) and Vapnik et al. (2015), this requires solving the Fredholm integral equation

$$\int \theta(x - t)P(y = 1|t)dP(t) = P(y = 1, x),$$

where probability functions $P(y = 1, x)$ and $P(x)$ are unknown but iid data (35) generated according to joint distribution $P(y, x)$ are given. Vapnik and Izmailov (2015a) and Vapnik et al. (2015) describe methods for solving this problem, producing the solution

$$P_\ell(y = 1|x) = P(y = 1|x; (x_1, y_1), ..., (x_\ell, y_\ell)).$$

In this section, we generalize classical Statistical Inference problem of conditional probability estimation to a new model of Statistical Inference with Privileged Information. In this model, along with information defined in the space X, one has the information defined in the space X^*.

Consider privileged space X^* along with space X . Suppose that any vector $x_i \in X$ has its image $x_i^* \in X^*$. Consider iid triplets

$$(x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell) \tag{36}$$

that are generated according to a fixed but unknown distribution function
$P(x, x^*, y)$. Suppose that, for any triplet (x_i, x_i^*, y_i), there exist conditional probabilities $P(y_i|x_i^*)$ and $P(y_i|x_i)$. Also, suppose that the conditional probability function $P(y|x^*)$, defined in the privileged space X^*, is *better* than the conditional probability function $P(y|x)$, defined in space X; here by "better" we mean that the *conditional entropy* for $P(y|x^*)$ is smaller than conditional entropy for $P(y|x)$:

$$-\int [\log_2 P(y = 1|x^*) + \log_2 P(y = 0|x^*)]\, dP(x^*) <$$

$$-\int [\log_2 P(y = 1|x) + \log_2 P(y = 0|x)]\, dP(x).$$

Our goal is to use triplets (36) for estimating the conditional probability
$P(y|x; (x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y_\ell))$ in space X better than it can be done with training pairs (35). That is, our goal is to find such a function

$$P_\ell(y = 1|x) = P(y = 1|x; (x_1, x_1^*, y_1), ..., (x_\ell, x_\ell^*, y))$$

14. The same method can be applied to all the problems described in Vapnik and Izmailov (2015a) and Vapnik et al. (2015).

that the following inequality holds:

$$-\int [\log_2 P(y=1|x;(x_i,x_i^*,y_i)_1^\ell) + \log_2 P(y=0|x;(x_i,x_i^*,y_i)_1^\ell)]dP(x) <$$

$$-\int [\log_2 P(y=1|x;(x_i,y_i)_1^\ell) + \log_2 P(y=0|x;(x_i,y_i)_1^\ell,)]dP(x).$$

Consider the following solution for this problem:

1. Using kernel $K(u^*, v^*)$, the training pairs (x_i^*, y_i) extracted from given training triplets (36) and the methods of solving our integral equation described in Vapnik and Izmailov (2015a) and Vapnik et al. (2015), find the solution of the problem in space of privileged information X^*:

$$P(y=1|x^*;(x_i^*,y_i)_1^\ell) = \sum_{i=1}^{\ell} \hat{\alpha}_i K(x_i^*, x^*) + b.$$

2. Find the fundamental elements of knowledge: vectors $u_1^*, ..., u_m^*$.

3. Using some universal kernels (say RBF or INK-Spline), find in the space X the approximations $\phi_k(x), k = 1, \ldots, m$ of the frames $(u_k^*, x^*)^2,\ k = 1, ..., m$.

4. Find the solution of the conditional probability estimation problem $P(y|\phi;(\phi_i,y_i)_1^\ell)$ in the space of pairs (ϕ, y), where $\phi = (\phi_1(x), \ldots, \phi_m(x))$.

5.5 Example of Knowledge Transfer Using Privileged Information

In this subsection, we describe an example where privileged information was used in the knowledge transfer framework. In this example, using set of of pre-processed video snapshots of a terrain, one has to separate pictures with specific targets on it (class $+1$) from pictures where there are no such targets (class -1).

The original videos were made using aerial cameras of different resolutions: a low resolution camera with wide view (capable to cover large areas quickly) and a high resolution camera with narrow view (covering smaller areas and thus unsuitable for fast coverage of terrain). The goal was to make judgments about presence or absence of targets using wide view camera that could quickly span large surface areas. The narrow view camera could be used during training phase for zooming in the areas where target presence was suspected, but it was not to be used during actual operation of the monitoring system, i.e., during test phase. Thus, the wide view camera with low resolution corresponds to standard information (space X), whereas the narrow view camera with high resolution corresponds to privileged information (space X^*).

The features for both standard and privileged information spaces were computed separately, using different specialized video processing algorithms, yielding 15 features for decision space X and 116 features for space of privileged information X^*.

The classification decision rules for presence or absence of targets were constructed using respectively,

- SVM with RBF kernel trained on 15 features of space X;

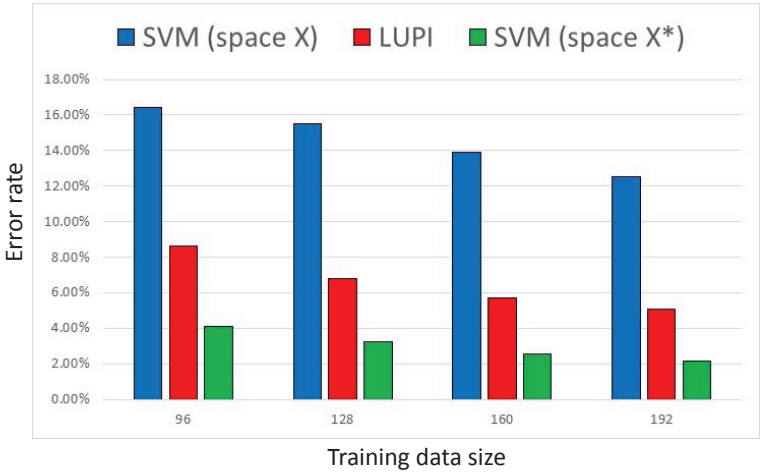


Figure 1: Comparison of SVM and knowledge transfer error rates: video snapshots example.

- SVM with RBF kernel trained on 116 features of space X^*;
- SVM with RBF kernel trained 15 original features of space X augmented with 116 knowledge transfer features, each constructed using regressions on the 15-dimensional decision space X (as outlined in subsection 5.2).

Parameters for SVMs with RBF kernel were selected using standard grid search with 6-fold cross validation.

Figure 1 illustrates performance (defined as an overage of error rate) of three algorithms each trained of 50 randomly selected subsets of sizes 64, 96, 128, 160, and 192: SVM in space X, SVM in space X^*, and SVM in space with transferred knowledge.

Figure 1 shows that, the larger is the training size, the better is the effect of knowledge transfer. For the largest training size considered in this experiment, the knowledge transfer was capable to recover almost 70% of the error rate gap between the error rates of SVM using only standard features and SVM using privileged features. In this Figure, one also can see that, even in the best case, the error rate using SVM in the space of privileged information is half of that of SVM in the space of transferred knowledge. This gap, probably, can be reduced even further by better selection of the fundamental concepts of knowledge in the space of privileged information and / or by constructing better regression.

5.6 General Remarks about Knowledge Transfer

5.6.1 What Knowledge Does Teacher Transfer?

In previous sections, we linked the knowledge of Intelligent Teacher about the problem of interest in X space to his knowledge about this problem in X^* space[15].

15. This two space learning paradigm with knowledge transfer for one space to another space reminds Plato's idea about *space of Ideas and space of Things* with transfer of knowledge from one space to another. This idea in different forms was explored by many philosophers.

One can give the following general mathematical justification for our model of knowledge transfer. Teacher knows that the goal of Student is to construct a good rule in space X with one of the functions from the set $f(x, \alpha),\ x \in X,\ \alpha \in \Lambda$ with capacity VC_X. Teacher also knows that there exists a rule of the same quality in space X^* – a rule that belongs to the set $f^*(x^*, \alpha^*),\ x^* \in X^*,\ \alpha^* \in \Lambda^*$ and that has a much smaller capacity VC_{X^*}. This knowledge can be defined by the ratio of the capacities

$$\kappa = \frac{VC_X}{VC_{X^*}}.$$

The larger is κ, the more knowledge Teacher can transfer to Student; also the larger is κ, the fewer examples will Student need to select a good classification rule.

5.6.2 Learning from Multiple Intelligent Teachers

Model of learning with Intelligent Teachers can be generalized for the situation when Student has $m > 1$ Intelligent Teachers that produce m training triplets

$$(x_{k_1}, x^{k*}_{k_1}, y_1), ..., (x_{k_\ell}, x^{k*}_{k_\ell}, y_\ell),$$

where $x_{k_t}, k = 1, ..., m,\ t = 1, ..., \ell$ are elements x of different training data generated by the same generator $P(x)$ and $x^{k*}_{k_t}, k = 1, ..., m,\ t = 1, ..., \ell$ are elements of the privileged information generated by kth Intelligent Teacher that uses generator $P_k(x^{k*}|x)$. In this situation, the method of knowledge transfer described above can be expanded in space X to include the knowledge delivered by all m Teachers.

5.6.3 Quadratic Kernel

In the method of knowledge transfer, the special role belongs to the quadratic kernel $(x_1, x_2)^2$. Formally, only two kernels are amenable for simple methods of finding the smallest number of fundamental elements of knowledge: the linear kernel (x_1, x_2) and the quadratic kernel $(x_1, x_2)^2$.

Indeed, if linear kernel is used, one constructs the separating hyperplane in the space of privileged information X^*

$$y = (w^*, x^*) + b^*,$$

where vector of coefficients w^* also belongs to the space X^*, so there is only one fundamental element of knowledge, i.e., the vector w^*. In this situation, the problem of constructing the regression function $y = \phi(x)$ from data

$$(x_1, (w^*, x^*_1)), ..., (x_\ell, (w^*, x^*_\ell)) \tag{37}$$

has, generally speaking, the same level of complexity as the standard problem of pattern recognition in space X using data (35). Therefore, one should not expect performance improvement when transferring the knowledge using (37).

With quadratic kernel, one obtains fewer than d fundamental elements of knowledge in d-dimensional space X^* (experiments show that the number of fundamental elements can be significantly smaller than d). According to the methods described above, one defines the knowledge in space X^* as a linear combination of m frames. That is, one splits the desired

function into m fragments (a linear combination of which defines the decision rule) and then estimates each of m functions $\phi_k(x)$ separately, using training sets of size ℓ. The idea is that, in order to estimate a fragment of the knowledge well, one can use a set of functions with a smaller capacity than is needed to estimate the entire function $y = f(x),\ x \in X$. Here privileged information can improve accuracy of estimation of the desired function.

To our knowledge, there exists only one nonlinear kernel (the quadratic kernel) that leads to an exact solution of the problem of finding the fundamental elements of knowledge. For all other nonlinear kernels, the problems of finding the minimal number of fundamental elements require difficult (heuristic) computational procedures.

6. Conclusions

In this paper, we tried to understand mechanisms of learning that go beyond brute force methods of function estimation. In order to accomplish this, we used the concept of Intelligent Teacher who generates privileged information during training session. We also described two mechanisms that can be used to accelerate the learning process:

1. The mechanism to control Student's concept of similarity between training examples.

2. The mechanism to transfer knowledge from the space of privileged information to the desired decision rule.

It is quite possible that there exist more mechanisms in Teacher-Student interactions and thus it is important to find them.

The idea of privileged information can be generalized to any statistical inference problem creating non-symmetric (two spaces) approach in statistics.

Teacher-Student interaction constitutes one of the key factors of intelligent behavior and it can be viewed as a basic element in understanding intelligence (for both machines and humans).

Acknowledgments

This material is based upon work partially supported by AFRL and DARPA under contract FA8750-14-C-0008. Any opinions, findings and / or conclusions in this material are those of the authors and do not necessarily reflect the views of AFRL and DARPA.

We thank Professor Cherkassky, Professor Gammerman, and Professor Vovk for their helpful comments on this paper.

References

R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufman Publishers, San Francisco, CA, 2004.

C. Burges. Simplified support vector decision rules. In *13th International Conference on Machine Learning, Proceedings*, pages 71–77, 1996.

A. Chervonenkis. *Computer Data Analysis (in Russian)*. Yandex, Moscow, 2013.

L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Applications of mathematics : stochastic modelling and applied probability. Springer, 1996.

L. Gurvits. A note on a scale-sensitive dimension of linear bounded functionals in banach spaces. *Theoretical Computer Science*, 261(1):81–90, 2001.

G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.

L. Liang and V. Cherkassky. Connection between SVM+ and multi-task learning. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2008, part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, June 1-6, 2008*, pages 2048–2054, 2008.

D. Pechyony, R. Izmailov, A. Vashist, and V. Vapnik. Smo-style algorithms for learning using privileged information. In *International Conference on Data Mining*, pages 235–241, 2010.

B. Ribeiro, C. Silva, N. Chen, A. Vieira, and J. das Neves. Enhanced default risk models with svm+. *Expert Systems with Applications*, 39(11):10140–10152, 2012.

B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In *Proceedings of the 14th Annual Conference on Computational Learning Theory and and 5th European Conference on Computational Learning Theory*, COLT '01/EuroCOLT '01, pages 416–426, London, UK, UK, 2001. Springer-Verlag.

V. Sharmanska, N. Quadrianto, and C. Lampert. Learning to rank using privileged information. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 825–832. IEEE, 2013.

V. Vapnik. *Estimation of Dependences Based on Empirical Data: Springer Series in Statistics (Springer Series in Statistics)*. Springer-Verlag New York, Inc., 1982.

V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

V. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer–Verlag, 2nd edition, 2006.

V. Vapnik and A. Chervonenkis. *Theory of Pattern Recognition (in Russian)*. Nauka, Moscow, 1974.

V. Vapnik and R. Izmailov. Statistical inference problems and their rigorous solutions. In Alexander Gammerman, Vladimir Vovk, and Harris Papadopoulos, editors, *Statistical Learning and Data Sciences*, volume 9047 of *Lecture Notes in Computer Science*, pages 33–71. Springer International Publishing, 2015a.

V. Vapnik and R. Izmailov. Learning with intelligent teacher: Similarity control and knowledge transfer. In A. Gammerman, V. Vovk, and H. Papadopoulos, editors, *Statistical Learning and Data Sciences*, volume 9047 of *Lecture Notes in Computer Science*, pages 3–32. Springer International Publishing, 2015b.

V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5-6):544–557, 2009.

V. Vapnik, I. Braga, and R. Izmailov. Constructive setting for problems of density ratio estimation. *Statistical Analysis and Data Mining*, 8(3):137–146, 2015.

Journal of Machine Learning Research 16 (2015) 2051-2066 Published 9/15

Alexey Chervonenkis's Bibliography: Introductory Comments

Alex Gammerman ALEX@CS.RHUL.AC.UK
Vladimir Vovk V.VOVK@RHUL.AC.UK
Computer Learning Research Centre, Department of Computer Science
Royal Holloway, University of London

This introduction to Alexey Chervonenkis's bibliography, which is published next in this issue, mainly consists of historical notes. The bibliography is doubtless incomplete, and it is just a first step in compiling more comprehensive ones. *En route* we also give some basic information about Alexey as a researcher and person; for further details, see, e.g., the short biography (Editors, 2015) in the Chervonenkis Festschrift. In this introduction, the numbers in square brackets refer to Chervonenkis's bibliography, and the author/year citations refer to the list of references at the end of this introduction.

Alexey Chervonenkis was born in Moscow in 1938. In 1955 he became a student at the MIPT, Moscow Institute of Physics and Technology (Faculty 1, Radio Engineering, nowadays Radio Engineering and Cybernetics). As part of his course of studies at the MIPT, he was attached to a laboratory at the ICS (the Institute of Control Sciences, called the Institute of Automation and Remote Control at the time), an institution in a huge system known as the Soviet Academy of Sciences.

In 1961 Alexey graduated from the MIPT and started his work for the ICS, where he stayed for the rest of his life. His first project at the ICS was very applied and devoted to designing a light organ for an exhibition in London (Russian Trade Fair, Earls Court, 1961). After completion of this project, Alexey was given an opportunity to concentrate on problems of cybernetics, namely pattern recognition; at that time cybernetics became extremely popular in the USSR, perhaps as a reaction to its earlier perception as a pseudo-science invented by the capitalist society and a "whore of imperialism" (Novoseltsev, 2015, p. 43).

In 1962 the joint work of Vapnik and Chervonenkis began. At that time they were members of the laboratory headed by Aleksandr Lerner, a leading cyberneticist. Lerner's laboratory was allowed to work on pattern recognition as a counterbalance to another laboratory, led by Mark Aizerman, which was the first to start work on this topic at the ICS: it was part of the strategy of Vadim Trapeznikov, the Institute director, to foster rivalry between different laboratories. Vapnik, a newly admitted PhD student, and Chervonenkis, hired a few months earlier as an engineer, were supposed to work as a pair. In hindsight, it appears that it was a perfect match; as Novoseltsev (2015) writes in his reminiscences, it is said that Vapnik was often inventing new things while Chervonenkis was proving them.

1. Foundations of Statistical Learning

Now Alexey Chervonenkis and Vladimir Vapnik are known, first of all, as the creators of the statistical theory of machine learning. However, their earliest joint work was devoted to non-statistical approaches to learning, as Alexey describes in [66]; it appears that this

work is not reflected at all in their joint publications. It was only in March 1963 that they brought statistics into their research.

When they started their joint work in Autumn 1962, they were interested in a problem that had more to do with the power of the teacher than the power of the learner. Suppose there are N decision rules, and one of them, say F, is used for classifying each point in a sequence $x_1, \dots, x_l$. The question is how small l can be so that there is only one decision rule (namely, F itself) compatible with the observed sequence $x_1, \dots, x_l$ and the classes $F(x_1), \dots, F(x_l)$. By choosing such a sequence the teacher can teach the learner to classify new points perfectly.

This problem is somewhat reminiscent of the problem of finding the counterfeit coin in a pile of N coins all but one of which are genuine. In the latter problem, we can take l of the order $\log N$, and the hope was that this remains true for the former. However, this is not the case. Consider N decision rules $F_1, \dots, F_N$ and $N-1$ points $x_1, \dots, x_{N-1}$ such that

$$F_i(x) := \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{if not,} \end{cases} \qquad i = 1, \dots, N-1,$$

and $F_N(x) = 0$ for all x. If F_N is used for classifying the points, seeing a new labelled point will allow the learner to discard at most one decision rule, and in order to discard all apart from the true one, we need $N-1$ observations. Therefore, the required value of l can be as large as $N-1$, which the two young researchers perceived as a failure of their non-statistical setting.

The statistical approach was first successfully used in [2] and [3]; the latter was prepared for a conference of young specialists in Spring 1963 (Editors, 2015). It was applicable to *algorithms with full memory*, introduced by Vapnik, Lyudmila Dronfort, and Chervonenkis in 1963, i.e., to algorithms that make no errors on the training set (the term "with full memory" was coined by Lyudmila). Suppose we are given two parameters, κ (the desired upper bound on the risk of the chosen decision rule) and η (the desired upper bound on the probability that the risk will in fact exceed κ). Let the true decision rule be F_i and the decision rule F_j chosen after l observations have a risk exceeding κ. The probability of getting all l labels right for F_j is at most $(1-\kappa)^l$, and the probability that at least one decision rule with risk exceeding κ will get all l labels right is at most $N(1-\kappa)^l$. To ensure that $N(1-\kappa)^l \le \eta$ it suffices to require

$$l \ge \frac{\log \eta - \log N}{\log(1-\kappa)}. \tag{1}$$

(This is Theorem 1 in [2].) As $-\log(1-\kappa) \ge \kappa$, the simpler inequality

$$l \ge \frac{\log N - \log \eta}{\kappa}$$

is also sufficient (and approximately equivalent to (1) for a small κ).

In might seem strange nowadays, but the presence of N in (1) was a real breakthrough. In [66] Alexey vividly describes discussions about the necessity of such an adjustment. The common reasoning among their colleagues (in related contexts) was that, since the probability of getting all labels right is at most $(1-\kappa)^l$ for **any** F_j, this is also true for the F_j

actually chosen by the algorithm, and so the fact that F_j is chosen *a posteriori* is irrelevant. Their colleagues, some of them very distinguished, could not be impressed by results like (1) believing that better results could be proved for a continuum of decision rules.

Alexey remembered a heated discussion with Yakov I. Khurgin (Lerner's friend and Professor of the Russian State University of Oil and Gas at the time) in Summer 1965. Khurgin's argument was, as usual, that a probabilistic statement that is true for all decision rules must be true for the one that was chosen by the algorithm. Alexey's counterargument was "The probability to meet randomly a syphilitic in Moscow is, say, 10^{-5}. But if you went to a venereal clinic, it is significantly greater, even though it is also in Moscow. Looking for the best decision rule is like a trip to a venereal clinic." In the context of infinitely many decision rules (e.g., linear), Khurgin argued that Vapnik and Chervonenkis were playing on the non-compactness of the Hilbert ball and, crucially, that they were demanding uniform convergence. Alexey agreed. This was the first time that the words "uniform convergence" were mentioned in this context. Later they became part of the titles of the fundamental papers [10,11,28,57].

Paper [2] applied the general performance guarantee (1) to the problem of classification of binary vectors of dimension n using perceptrons. For simplicity, in this introduction we will only discuss binary decision rules, in which case a decision rule can be identified with a set in the input space, and perceptrons then become half-spaces. The authors bounded above the number of ways in which an $n-1$-dimensional hyperplane can split the n-dimensional Boolean cube by $2^{n(n+1)}/(n+1)!$, which gives the sample size

$$l \geq \frac{\log \eta - n(n+1) + \log((n+1)!)}{\log(1-\kappa)} \approx \frac{\log \eta - n^2}{\log(1-\kappa)}$$

for the Boolean input space with n attributes, where log now stands for binary logarithm.

In [5] Vapnik and Chervonenkis introduced a new learning framework, which they called "extremal imitation teaching". Suppose we observe a sequence of random pairs $(X(k), Y(k))$, $k = 1, 2, \ldots$, generated independently from the same distribution P and also observe, at each step k, the value $g(X(k), Y(k))$ of a "reward function" g. For each k, we are allowed to replace $Y(k)$ by our chosen value $Y^*(k)$, in which case we observe $g(X(k), Y^*(k))$ instead of $g(X(k), Y(k))$. This can be a model of, e.g., a chemical process at a chemical plant: $X(k)$ describes the kth batch of raw materials, $Y(k)$ describes the parameters of the process (such as temperature, pressure, or reagents) chosen by an experienced plant operator, and $g(X(k), Y(k))$ is the quality of the choice (assumed observable and determined by $X(k)$ and $Y(k)$ alone). It is supposed that g belongs to a known finite set Q of functions, say of size N, and that it takes values in a known interval $[a, b]$. The learning problem involves three positive parameters, ϵ, κ, and η, and is to find $l = l(N, \epsilon, \kappa, \eta)$ (as small as possible) such that after l steps we can come up with a strategy of choosing Y^* as a function of X that satisfies, with a probability at least $1 - \eta$ over the training set,

$$P\left(g(X, Y) > g(X, Y^*) + \epsilon\right) < \kappa,$$

where P is the probability over the random choice of (X, Y). Vapnik and Chervonenkis's proposed strategy is, in their terminology, sequential: first it observes $(X(k), Y(k))$, $k = 1, 2, \ldots$, and then it "trains", replacing $Y(k)$ by its own $Y^*(k)$ from some k on. The strategy

requires

$$l = O\left(\frac{\log N}{\epsilon\kappa}\left(\log\log N - \log\epsilon - \log\eta\right)\right),$$

where the O notation refers to $N \to \infty$, $\eta \to 0$, $\kappa \to 0$, and $\epsilon \to 0$; a and b are regarded as fixed constants. Namely, the strategy first makes

$$l_1 = O\left(\frac{\log N - \log\eta}{\epsilon\kappa}\right) \tag{2}$$

passive observations, and then it starts $d = O((\log N)/\epsilon)$ active training periods of length

$$l_0 = O\left(\frac{\log\log N - \log\epsilon - \log\eta}{\kappa}\right)$$

each; in each training period it tests a new strategy of choosing Y^* until it fails, in some sense (if it never fails during the l_0 steps, training is complete and the learning procedure is stopped).

They derive a very interesting corollary in the spirit of prediction with expert advice (Cesa-Bianchi and Lugosi, 2006). Suppose, in the language of our example, we can observe n experienced plant operators instead of just one. Observing each of the operators for l_1 steps (see (2)) and then training as before, our resulting strategy is likely to be competitive with the best operator at each step: with a probability at least $1 - \eta$ over the training set,

$$P\left(\max_{i=1,\dots,n} g(X, Y^i) > g(X, Y^*) + \epsilon\right) < \kappa,$$

where Y^i is the ith operator's output (being competitive at each step is unusual from the point of view of prediction with expert advice, where the goal is to be competitive in the sense of cumulative rewards or losses). Now passively observing takes nl_1 steps, whereas active training still takes $l_2 := dl_0$ steps.

In [6] Vapnik and Chervonenkis extended the methods of [5] to limited infinite classes of reward functions, and in [7] they applied them to the problem of playing an unknown zero-sum game.

Until Summer 1966 Vapnik and Chervonenkis could prove performance guarantees only for a finite number of decision rules. At the level of mathematical rigour that they set for themselves, they could not accept their colleagues' argument (see above) although they shared their optimism about, say, the learnability of the linear decision rules in Euclidean space. The first breakthrough came in July 1966 [66], when they extended their learnability results to classes of decision rules with a slow growth function (see below), and the second in September 1966, when they characterized the classes with slow growth functions in terms of what is now known as VC dimension. The main definitions (very well known by now) that we will need to talk about these developments are:

- Given a class S of decision rules on (i.e., subsets of) an input space X, let $\Delta^S(x_1, \dots, x_l)$ be the number of different restrictions of those decision rules to the finite set $\{x_1, \dots, x_l\}$ in X; Vapnik and Chervonenkis called $\Delta(x_1, \dots, x_l)$ the *index* of S with respect to $x_1, \dots, x_l$.

- They called the maximum

$$m^S(l) := \max_{x_1,\dots,x_l} \Delta^S(x_1,\dots,x_l)$$

  (as function of l) the *growth function* of S.

- The value

$$\mathrm{VC}(S) := \max\{l \mid m^S(l) = 2^l\}$$

  is now known as the *VC-dimension* of the class S. See Dudley (2015b, Section 4.6) for a discussion of the origin of the expression "VC-dimension"; the earliest mention of it seems to be in the article by Blumer et al. (1986) (in the form "Vapnik–Chervonenkis dimension", which was abbreviated to "VC dimension" in the journal version), whereas the abbreviation VC was coined by Dudley himself (Bottou, 2013).

The key fact about the growth function is now known as *Sauer's lemma*: if $\mathrm{VC}(S) = \infty$, $m^S(l) = 2^l$ for all l; otherwise,

$$m^S(l) \le \sum_{j=0}^{\mathrm{VC}(S)} \binom{l}{j} = O(l^{\mathrm{VC}(S)}) \tag{3}$$

for all l (the binomial coefficient is defined to be 0 when $j > l$). Therefore, we have the *Vapnik–Chervonenkis dichotomy*: the rate of growth of m^S is either exponential or at most polynomial. Sauer's lemma is more precise and also gives the degree of the polynomial ($\mathrm{VC}(S)$, which Sauer referred to as the density of S).

The main papers in which the 1966 breakthrough was announced and described were published in 1968 and 1971:

- In the first 1968 paper [9] Vapnik and Chervonenkis showed that the class S is learnable, in the now standard sense, if the rate of growth of m^S is polynomial (or slower).

- The second 1968 paper [10] is the famous announcement of their main results obtained during this period; it was "the true beginnings of Statistical Learning Theory" according to Bottou (2013).

- In [11] (1971), they gave detailed proofs.

- In their other 1971 paper [12] they explained in detail how the results of [11] can be applied to machine learning.

The results of [9] were obtained in July 1966, as Alexey describes in [66]. At that time Vapnik and Chervonenkis started to suspect that there are only two kinds of growth functions: exponential and (at most) polynomial. Vapnik said that even if this were true, it would be very difficult to prove it, but Chervonenkis presented a proof two months later (Novoseltsev, 2015).

A footnote in [9] says that after the paper had been submitted, the authors discovered that either $m^S(l) = 2^l$ for all l or $m^S(l) \le l^{\mathrm{VC}(S)+1}$ for all l. (This should have said "for all $l > 1$".) The date of submission is given as 20 September 1966.

Announcement [10] was published in *Doklady AN SSSR* (usually translated as *Proceedings of the USSR Academy of Sciences*). This journal only publishes papers presented by full and corresponding members of the Academy. The announcement stated the Vapnik–Chervonenkis dichotomy, in the form of the footnote in [9], as Theorem 1. It was first submitted for publication in 1966. The authors wanted Academician Andrei N. Kolmogorov to present their note, but submitted it directly to the journal, which forwarded it to Kolmogorov, who gave it to Boris V. Gnedenko to read. The authors did not hear from the journal for a long time, and a chain of enquiries led them to Gnedenko. Gnedenko explained to the young authors that what they were doing was not statistics; statistics was what Kolmogorov, Gnedenko himself, and their students were doing, and there was no chance that Gnedenko or his students would work in this new area. In the end the note was presented by the ICS Director Trapeznikov and submitted for publication on the same date, 6 October 1967; as compared to the original 1966 submission the authors only changed 2 lines in their manuscript: "Presented by Academician A. N. Kolmogorov" became "Presented by Academician V. A. Trapeznikov". The topic to which the note was assigned in the journal changed from "Probability theory" to "Cybernetics".

In [11], the authors still have Sauer's lemma with $\mathrm{VC}(S)+1$ instead of $\mathrm{VC}(S)$ (for the first time they will give the optimal exponent $\mathrm{VC}(S)$ in (3) in their book [18]). That paper was written in 1966, at the same time as their *Doklady* announcement [10], as it was customary for such announcements to be submitted together with a full paper, so that proofs of their statements could be checked. The key results of [11] were the VC dichotomy (Theorem 1), the uniform convergence of frequencies to probabilities over classes with polynomial growth functions (Theorem 3 and its small-sample counterpart Theorem 2), and an elegant necessary and sufficient condition for the uniform convergence of frequencies to probabilities in terms of the *entropy* $H^S(l) := \mathbb{E} \log \Delta^S(x_1, \dots, x_l)$ (Theorem 4).

In the four papers [9–12], Vapnik and Chervonenkis made a great leap forward in mathematical rigour. However, the required assumptions of measurability were very subtle, and even Vapnik and Chervonenkis did not get them quite right. After a modest description of his own mistakes in related measurability conditions, Dudley (2015a) points out that their requirement (in the penultimate paragraph of the Introduction to [11]) of

$$(x_1, \dots, x_l) \mapsto \sup_{A \in S} \left| \nu_A^{(l)}(x_1, \dots, x_l) - P(A) \right| \tag{4}$$

(where $\nu_A^{(l)}(x_1, \dots, x_l) := n_A/l$ and n_A is the frequency of A in the sample $x_1, \dots, x_l$) being measurable is not sufficient, as shown in the introduction to Dudley (1999, Chap. 5). The condition that is actually needed in the proof is that

$$(x_1, \dots, x_{2l}) \mapsto \sup_{A \in S} \left| \nu_A^{(l)}(x_1, \dots, x_l) - \nu_A^{(l)}(x_{l+1}, \dots, x_{2l}) \right|$$

be measurable.

The notion of growth function introduced in [9,10] was innovative but had had several interesting precursors, as described by Dudley (2015b). Already by 1852 Schläfli (1814–1895) found the growth function for the class S of all half-spaces in $\mathbb{R}^d$ containing 0 on their

boundary,

$$m^S(l) = 2\sum_{j=0}^{d-1} \binom{l-1}{j} = O(l^{d-1}) \qquad (l \to \infty). \tag{5}$$

Schläfli's memoir containing this result was published only in 1901 despite being written in 1850–1852. Among other fundamental achievements of this memoir were the introduction of d-dimensional Euclidean geometry (mathematicians had only treated the case $d \le 3$ before) and the extension of the ancient Greeks' result that there are only five platonic solids, i.e., convex regular polytopes in $\mathbb{R}^3$, to the case of $\mathbb{R}^d$ with $d > 3$ (it turned out that for $d > 4$ there are only three trivial platonic solids, the generalizations of the tetrahedron, cube, and octahedron, whereas for $d = 4$ there are six). Cover (1965) pointed out that, using Schläfli's method, one can obtain

$$m^S(l) = 2\sum_{j=0}^{d} \binom{l-1}{j} = O(l^d) \qquad (l \to \infty) \tag{6}$$

for the class S of all half-spaces in $\mathbb{R}^d$; he also obtained similar results for other classes, such as the parts of $\mathbb{R}^d$ bounded by hyperspheres or hypercones.

Richard Dudley wrote enthusiastic reviews of both [10] and [11] for *Mathematical Reviews*; interestingly, his review of [10] was instrumental in obtaining the permission to publish [11] (Bottou, 2012, with a reference to Vapnik). These reviews attracted attention of some leading mathematicians, and it seems likely that they were the means through which the VC dichotomy, in the form of a conjecture, reached the attention of Sauer and another independent discoverer, Shelah (together with his PhD student Perles).

The first statement of convergence of frequencies of events to their probabilities was James Bernoulli's (1713) celebrated law of large numbers, stating that, for all $\epsilon > 0$, events A, and probability measures P,

$$P^l\left(\left|\nu_A^{(l)} - P(A)\right| > \epsilon\right) \to 0 \tag{7}$$

as $l \to \infty$ (using the notation $\nu_A^{(l)}$ introduced in (4) and under unnecessary but mild restrictions on $P(A)$ and ϵ). Now we know that the convergence (7) is uniform in P, but Bernoulli did not know that, which might have been one of his reasons for not completing his manuscript (published in 1713 posthumously by his nephew): if the convergence is uniform, we can easily invert (7) to obtain a confidence interval for $P(A)$ given the observed frequency $\nu_A^{(l)}$. (This is one of the two reasons put forward by Hald 2003, p. 263; other authors have come up with more.)

Uspensky (1937) gives a "modernized" version of James Bernoulli's proof that does give a uniform convergence in (7) (cf. Hald 2003, p. 268). Nowadays, the most standard proof is based on Chebyshev's inequality and immediately gives uniform convergence:

$$P^l\left(\left|\nu_A^{(l)} - P(A)\right| > \epsilon\right) \le \frac{P(A)(1-P(A))}{l\epsilon^2} \le \frac{1}{4l\epsilon^2} \to 0.$$

(Although large-deviation inequalities, such as Hoeffding's, often give better results.)

Vapnik and Chervonenkis came up with a much deeper, and entirely different, statement of uniformity: for a fixed P,

$$P^l\left(\sup_{A\in S}\left|\nu_A^{(l)}-P(A)\right|>\epsilon\right)\to 0$$

for many interesting classes S of events; the requirement of uniformity was again motivated by statistical applications. If we require uniformity in both A and P,

$$\sup_P P^l\left(\sup_{A\in S}\left|\nu_A^{(l)}-P(A)\right|>\epsilon\right)\to 0$$

(i.e., that S be a *uniformly Glivenko–Cantelli class*), the condition $\mathrm{VC}(S)<\infty$ becomes both necessary and sufficient, for any $\epsilon\in(0,1)$; Vapnik and Chervonenkis understood this well already in 1966 (Editors, 2015).

In 1974 Vapnik and Chervonenkis published their book [18] in which they gave a survey of their work so far on the foundations of statistical learning (in Part II) and the method of generalized portrait (see the next section). They introduced a name for $\mathrm{VC}(S)$ (Chapter V, Section 7), namely the *capacity* of S (емкость S). The authors sent a copy of the book to *Mathematical Reviews*, requesting that it be sent to Richard Dudley to review. Whereas the papers [10] and [11] were reviewed quickly, in 1969 and 1972, respectively, reviewing [18] took five years, and Dudley's review appeared only in 1979. Dudley (2015b) explains this by the Peter principle: as reviewer, he was promoted to reviewing more and more difficult publications by Vapnik and Chervonenkis until his knowledge of the Russian language and pattern recognition became insufficient for the task.

In their book Vapnik and Chervonenkis gave Sauer's (1972) form of their dichotomy, which is obviously sharp in general: it suffices to take as S the class of all sets of cardinality $\mathrm{VC}(S)$ in an infinite input space X. For specific classes, however, even very important ones, the bound can be far from being sharp: e.g., for Schläfli's and Cover's cases Sauer's lemma only gives

$$m^S(l)\le\sum_{j=0}^{d}\binom{l}{j}=\Omega(l^d)\quad\text{and}\quad m^S(l)\le\sum_{j=0}^{d+1}\binom{l}{j}=\Omega(l^{d+1})$$

in place of (5) and (6), respectively.

An important contribution of the book [18], alongside with the papers [16,17], was the introduction of the method of Structural Risk Minimization (in Chapter VI) and its application to various specific problems. An appendix to Chapter VI (Section 14) gives lower bounds for the performance guarantees of learning algorithms in terms of the VC dimension.

In [28] Vapnik and Chervonenkis extended their necessary and sufficient condition of uniform convergence for classes of events (Theorem 4 of [11]) to classes of functions, defining the functional analogue of the entropy function H^S using Kolmogorov and Tikhomirov's ϵ-entropy. In [38] they found necessary and sufficient conditions for one-sided uniform convergence (Theorem Б of [38], where Б is the second letter of the Russian alphabet), which is particularly important from the viewpoint of machine learning because of its equivalence to the consistency of the method of empirical risk minimization (Theorem A of [38], where A is the first letter of the Russian alphabet).

Alexey's last great mathematical achievement [57,58] was the definitive quantitative form of Michel Talagrand's result about the existence of "bad sets" in machine learning (Talagrand, 1987, Theorem 5, and Talagrand, 1996, Theorem 2.1), which he first discovered just a few year's after Talagrand discovered his (Talagrand, 2014) but could prove rigorously only in the last years of his life (see [67], Theorem 13.1). Typically, he did this without any knowledge of Talagrand's work (Novoseltsev, 2015). Alexey's result says that if $H^S(l)/l \to c$ as $l \to \infty$ (the limit always exists), there exists a set $E \subseteq X$ of probability c such that for any n almost all sequences in E^n are shattered by S (a sequence $x_1, \ldots, x_n$ is *shattered* by S if $\Delta^S(x_1, \ldots, x_n) = 2^n$). Talagrand's result only asserts, for $c > 0$, the existence of E of positive probability satisfying the last condition. A precursor of this result was stated in [38] as Theorem B (B being the third letter of the Russian alphabet), and the result found its way into Vapnik (1998, Theorem 3.6).

Chervonenkis and Talagrand met in Paris in May 2011 and discussed the former's quantitative form of the latter's result (which Talagrand was really proud of but which, as he says, would not have been even conceivable without Chervonenkis's previous contributions). Chervonenkis asked Talagrand whether the quantitative form should be published. Talagrand replied that the quantitative form did not seem to have much use and so discouraged Chervonenkis from its publication (Talagrand, 2014).

2. Generalized Portrait and Optimal Separating Hyperplane

Vapnik and Chervonenkis's first joint paper [1] introduced the method of generalized portrait, which is a linear precursor of support vector machines, in the case of supervised learning. The idea of the method itself was first published by Vapnik and Lerner (1963) a year earlier, and Vapnik, Lerner, and Chervonenkis started discussing the method already in 1962 (see [61], which is an excellent source for the early history of support vector machines).

Vapnik and Lerner (1963) work in the context of unsupervised learning. The starting point of the early versions of the method of generalized portrait was that patterns were represented by points on the unit sphere in a Hilbert space. (Vapnik and Lerner consider a family of mappings from the patterns to the unit sphere, but let us, for simplicity, fix such a mapping, assume that it is a bijection, and identify patterns with the corresponding points of the unit sphere.) A set F of patterns *divides into n images* $F_1, \ldots, F_n$ (these are disjoint subsets of F) if for each F_i there is a point ϕ_i on the sphere such that, for all i, $j \neq i$, $f_i \in F_i$, and $f_j \in F_j$, it is true that $(\phi_i, f_i) > (\phi_i, f_j)$. Under a further restriction (the images should be "definite"), ϕ_i is called a *generalized portrait* for F_i. In their definition, Vapnik and Lerner do not specify a precise optimization problem with a unique solution that generalized portraits are required to solve. Later in the paper they do give two ideas for such optimization problems:

- In Section 4, they say that, for a given F_i, ϕ_i can be defined to maximize the *recognition threshold* $\min_{f \in F_i}(\phi_i, f)$. (This is the optimization problem that Alexey describes in the section devoted to Vapnik and Lerner's 1963 paper in his historical contribution to the Vapnik Festschrift: see [61], Section 3.1.1.) The overall optimization problem (to be solved before dividing F into images), however, remains unspecified.

- In the concluding Section 5 of their paper, Vapnik and Lerner make the problem of "self-learning" (unsupervised learning in this context) more precise by requiring that the generalized portraits $\phi_1, \dots, \phi_n$ maximize the *order of distinguishability* $1 - \max_{i,j}(\phi_i, \phi_j)$. This optimization problem will rarely determine generalized portraits completely: e.g., in the case of two images, $n = 2$, this condition only restricts ϕ_1 and ϕ_2 to being anti-collinear. And only rarely will any of its solutions maximize the recognition thresholds.

In [1] Vapnik and Chervonenkis made several important steps in the development of the method of generalized portrait; in particular, they defined it in the case of supervised learning and expressed it as a precise optimization problem. Suppose we are interested in a class K_1 of patterns and K_2 is the union of the other classes; these are assumed to be subsets of the unit sphere in a Hilbert space. The generalized portrait of K_1 is defined in this paper as the unit vector ϕ solving the optimization problem

$$\begin{aligned} (\phi, X) &\geq c, \quad \forall X \in K_1, \\ (\phi, Y) &\leq c, \quad \forall Y \in K_2, \\ c &\to \max . \end{aligned} \tag{8}$$

When the solution $(\phi, c) = (\phi_0, C(\phi_0))$ exists (i.e., when the class K_1 is linearly separable from the rest of data), it is unique, and the vectors $X \in K_1$ and $Y \in K_2$ satisfying $(\phi_0, X) = C(\phi_0)$ or $(\phi_0, Y) = C(\phi_0)$, respectively, were called the *marginal vectors*; these are precursors of support vectors. It was shown that the generalized portrait is a linear combination of marginal vectors (with nonnegative coefficients if they belong to K_1 and nonpositive if not).

Another contribution of [1] was that the method was rewritten in terms of scalar products between input vectors, which was an important step towards support vector machines. As it often happens, necessity was the mother of invention ([61], Section 3.3; Editors, 2015). At that time the ICS only had analogue computers, and inputting data was difficult. The easiest way was to calculate the scalar products by hand or using calculators, and then input them into the analogue computers by adjusting corresponding resistors. In 1964 the first digital computers arrived, and the dual form of the method lost much of its appeal for a few dozen years.

Vapnik and Chervonenkis kept the name "method of generalized portrait" in [1]. This might have been the first application of their decision (Novoseltsev, 2015) not to coin a new name for each new modification of their main recognition method; Vladimir proposed to use the same name for all modifications, the method of generalized portrait, and Alexey agreed. (There might have been one exception: it appears that in print the method of optimal separating hyperplane has not been explicitly referred to as that of "generalized portrait". In particular, the methods of generalized portrait and optimal separating hyperplane are treated as different ones in [61].)

As Alexey discusses in his historical paper [61] (Sections 3.1–3.2), already in 1962 he and Vladimir considered a more general version of the method, with $(\phi, Y) \leq kc$ in place of

$(\phi, Y) \le c$ in (8), for a given constant $k < 1$:

$$(\phi, X) \ge c, \quad \forall X \in K_1, \tag{9}$$

$$(\phi, Y) \le kc, \quad \forall Y \in K_2, \tag{10}$$

$$c \to \max. \tag{11}$$

In the same year they obtained the possibility of decomposition of the generalized portrait via marginal vectors directly, without the use of the Kuhn–Tucker theorem.

The generalization (9)–(11) was first published in [9], where the assumption that the training patterns should belong to a unit hypersphere is no longer mentioned. The authors retained the name "generalized portrait" for this more general setting. Using the Kuhn–Tucker theorem, they showed that the generalized portrait can be found by minimizing a quadratic function over the positive quadrant and developed several algorithms for solving such problems.

Further important developments were made in the 1973 papers [14,15] published in the same book edited by Vapnik and describing a library of computer programs written by Zhuravel' and Glazkova and implementing the method of generalized portrait (improved versions are described in [18], Chapter XV). In [14], Vapnik, Chervonenkis, and their co-authors consider the method of generalized portrait (9)–(11), whereas in [15] they consider a new method, that of optimal separating hyperplane. Given two linearly separable sets of vectors, X and $\bar{X}$ (the notation they use for K_1 and K_2 in this paper), they define the optimal separating hyperplane as the hyperplane that separates the two sets and is as far as possible from their convex closures. They notice that the optimal separating hyperplane can be represented by the equation $(\psi, x) = (c_1 + c_2)/2$, where ψ is the shortest vector satisfying $(\psi, z) \ge 1$ for all z of the form $z = x - \bar{x}$, $x \in X$ and $\bar{x} \in \bar{X}$, and

$$c_1 = \min_X (\psi, x), \quad c_2 = \max_{\bar{X}} (\psi, \bar{x}).$$

Together with the fact that ψ can be represented as a linear combination of margin vectors, this serves as the basis of their algorithm GP-4 for finding the optimal separating hyperplane.

The fundamental 1974 book [18] consists of three parts, one of which, Part III, is devoted to the methods of generalized portrait and optimal separating hyperplane (Part I is introductory and Part II is called "Statistical foundations of the theory"). In this part (Chapter XIV, Section 12) the authors derive another kind of performance guarantees for the two methods, which, as they say, are much closer to the lower bounds of Section VI.14 (already mentioned in Section 1 above) and so demonstrate special statistical properties of the method. A simple performance guarantee of this kind is that the (unconditional) probability of error does not exceed $m/(l+1)$, where l is the length of the training sequence and m is the expectation of the number of essential support vectors (which they called informative marginal vectors at the time). Since, in their context, m does not exceed the dimension n (assumed finite) of the input space, the probability of error is also bounded by $n/(l+1)$. This result was obtained by Alexey in June 1966 [66], but Vladimir was reluctant to publish it as it was embarrassingly simple. Let us call this type of error bounds *VC74 bounds* and the type of bounds discussed in the previous section *VC68 bounds* (following Vovk 2015). There were hints of VC74 bounds in [9], Section 5.3, and [14], pp. 91–92; however, the first

precise statements were first published only in the 1974 book [18]. It is interesting that, as Alexey says at the end of Section 3.6 of [61], VC74 bounds led to the notions of the growth function and VC dimension and to conditions for uniform convergence; it can be concluded that VC74 bounds led to VC68 bounds.

At the beginning of Chapter XIV the authors emphasize that in many cases the optimal separating hyperplane should be constructed not in the original input space but in a feature space (спрямляемое пространство). They only discuss finite-dimensional feature spaces, but since they already have the dual form of the optimization problem, there is only one step to support vector machines: to combine their algorithms with the idea of kernels that was already used by their competitors in Aizerman's laboratory (Aizerman et al., 1964); but this step had to wait for another 20 years.

The book [18] treats the methods of generalized portrait and optimal separating hyperplane more or less on equal footing, and studies relations between them, such as the latter being a special case of the former corresponding to a certain value of k. In the historical paper [61] Alexey mentions that in his and Vladimir's experience the number of support vectors for the optimal separating hyperplane often turned out to be larger than that for the generalized portrait for other values of k. His suggestion is to return to the method of generalized portrait (surely in combination with kernel methods—Eds.) looking for k providing the fewest number of support vectors. His intuition was that in the case of two approximately equal classes the method of optimal separating hyperplane is preferable. However, in the case where a small class is being separated from a much larger one (such as separating the letter "a" from the other letters of the English alphabet) the method of generalized portrait with a constant k close to 1 is preferable.

3. Other Publications

Approximately one half of Alexey's publications are devoted to applications of machine learning in various fields, such as natural language systems, geology, and medicine. This work was mainly done in collaboration with colleagues at the Institute of Control Sciences, the University of London, and Yandex.

In 1975–1983 Alexey and his colleagues at the ICS published a series of papers [19,21–27,29] describing their interactive data-retrieval system using a subset of the Russian language to control a large sea port. Alexey's main co-author was Leonid Mikulich, who also worked in Lerner's laboratory starting from 1961. In the course of numerous conversations between them Alexey proposed a formal logical calculus for describing non-trivial linguistic structures [19]. They also often discussed modelling evolution, and much later they were surprised to discover that it had become popular under the name of evolutionary and genetic programming.

Alexey's next significant area of applied research was geology [30–33,35,37,39–41,43,44]. This work included designing mathematical models for geological processes and non-parametric alternatives to the popular method of Kriging for restoring conditional distributions from empirical data. On the practical side, Alexey created a system for optimal automatic delineation of ore bodies that has been in operation at the world's largest gold deposit Murun-Tau since 1986 (Novoseltsev, 2015). For the creation of this system he was awarded the State Prize of the USSR (formerly Stalin Prize) in 1987.

Alexey's first work in medicine was done in 1971 [13] jointly with Vapnik and Lerner, but most of his papers in this area [51,53,56,59,60,68] were written together with his colleagues at Royal Holloway, University of London (whose Professor he formally became in 2000). A closely related application area in which Alexey was active is bioinformatics: see [45–47]. In the course of his work on bioinformatics he independently (albeit significantly later) rediscovered Watkins's (2000) and Haussler's (1999) string kernels. In general, independent rediscoveries were a typical feature of his research, arising naturally when a creative mind does not follow current literature preferring instead to invent new directions for itself at the risk of "discovering" well-known results and concepts. (A good example of this is Werner Heisenberg's rediscovery of matrix algebra in developing his approach to quantum mechanics.) Another independent rediscovery was his combination of Bayes and maximum likelihood methods for regression [42], which he later found in the work of David MacKay and finally [42,52] traced to a 1970 paper (Turchin et al., 1971).

Among Alexey's other applied papers were those devoted to energy load forecasting [49] and aircraft engineering [55]. One of Alexey's last applied research areas was the problem of optimal placement of advertisements among the results of a web search [63–65], which is of great interest to Yandex, the Russian analogue of Google, with which he was affiliated (alongside the ICS and Royal Holloway, University of London) since 2011.

From 2007 Alexey lectured at the School of Data Analysis founded by Yandex, and it is due to this activity that we owe his excellent textbook [52]. We are also lucky to have historical papers and notes published or prepared for publication during the last years of his life, namely the two sets of reminiscences in Vladimir Vapnik's and his own Festschriften [61,66] and his review [67] (preceded by the abstract [62]).

A lot remains unwritten or unfinished. Alexey was active, physically and mentally, and full of ideas until the very moment of his tragic death in the early hours of 22 September 2014 in Elk Island just outside Moscow.

Acknowledgments

The editors were greatly helped in their work by the late Alexey Chervonenkis, who generously shared his recollections with them. Many thanks to Vladimir Vapnik, Leonid Mikulich, and Michel Talagrand, who were invaluable sources of further information. Anatolii Mikhalsky's help is also gratefully appreciated.

References

Mark A. Aizerman, Emmanuel M. Braverman, and Lev I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.

Jacob Bernoulli. *Ars Conjectandi*. Thurnisius, Basel, 1713.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred Warmuth. Classifying learnable geometric concepts with the Vapnik–Chervonenkis dimension. In *Proceedings of the 18th ACM Symposium on Theory of Computing*, pages 273–282, New York,

1986. ACM. Extended abstract. The full journal paper appeared as "Learnability and the Vapnik–Chervonenkis dimension" in *Journal of the Association for Computing Machinery*, 36:929–965, 1989.

Léon Bottou. On the Vapnik–Chervonenkis–Sauer lemma, 2012. URL `http://leon.bottou.org/news/vapnik-chervonenkis_sauer`. Accessed in September 2015.

Léon Bottou. In hindsight: Doklady Akademii Nauk SSSR, 181(4), 1968. In *Empirical Inference: A Festschrift in Honor of Vladimir N. Vapnik*, pages 13–20. Springer, Berlin, 2013.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, Cambridge, 2006.

Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14:326–334, 1965.

R. M. Dudley. *Uniform Central Limit Theorems*. Cambridge University Press, Cambridge, 1999. Second edition: 2014.

R. M. Dudley. A paper that created three new fields: Teoriya veroyatnosteĭ i ee primeneniya 16(2), 1971, pp. 264–279. In *Measures of Complexity: Festschrift for Alexey Chervonenkis*, chapter 2, pages 9–10. Springer, Berlin, 2015a.

R. M. Dudley. Sketched history: VC combinatorics, 1826 up to 1975. In *Measures of Complexity: Festschrift for Alexey Chervonenkis*, chapter 4, pages 31–42. Springer, Berlin, 2015b.

Editors. Short biography of Alexey Chervonenkis. In Vladimir Vovk, Harris Papadopoulos, and Alex Gammerman, editors, *Measures of Complexity: Festschrift for Alexey Chervonenkis*, pages ix–xvii. Springer, Berlin, 2015. This biography draws heavily on Chervonenkis's unpublished reminiscences.

Anders Hald. *History of Probability and Statistics and Their Applications before 1750*. Wiley, Hoboken, NJ, 2003.

David Haussler. Convolution kernels on discrete structures. Technical Report UCSC-CRL-99-10, University of California at Santa Cruz, Computer Science Department, July 1999.

Vasily N. Novoseltsev. Institute of Control Sciences through the lens of VC dimension. In *Measures of Complexity: Festschrift for Alexey Chervonenkis*, chapter 5, pages 43–53. Springer, Berlin, 2015.

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972. Submitted on 4 February 1970.

Ludwig Schläfli. *Theorie der vielfachen Kontinuität*, volume 38 (1st half) of *Denkschriften der Schweizerischen Naturforschenden Gesellschaft*. Zürcher & Furrer, Bern, 1901. Written in 1850–1852.

Michel Talagrand. The Glivenko–Cantelli problem. *Annals of Probability*, 15:837–870, 1987.

Michel Talagrand. The Glivenko–Cantelli problem, ten years later. *Journal of Theoretical Probability*, 9:371–384, 1996.

Michel Talagrand. An email to Alex Gammerman of 12 October, 2014.

V. F. Turchin, V. P. Kozlov, and M. S. Malkevich. The use of mathematical-statistics methods in the solution of incorrectly posed problems. *Soviet Physics Uspekhi*, 13: 681–703, 1971. Russian original: В. Ф. Турчин, В. П. Козлов, М. С. Малкевич. Использование методов математической статистики для решения некорректных задач. *Успехи физических наук*, 102(3):345–386, 1970.

J. V. Uspensky. *Introduction to Mathematical Probability.* McGraw-Hill, New York, 1937.

Vladimir N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.

Vladimir N. Vapnik and Aleksandr Ya. Lerner. Pattern recognition using generalized portraits. *Automation and Remote Control*, 24:709–715, 1963. Russian original: В. Н. Вапник, А. Я. Лернер. Узнавание образов при помощи обобщенных портретов. *Автоматика и телемеханика*, 24(6):774–780, 1964. The original article submitted on 26 December 1962.

Vladimir Vovk. Comment: The two styles of VC bounds. In *Measures of Complexity: Festschrift for Alexey Chervonenkis*, chapter 11, pages 161–164. Springer, Berlin, 2015.

Chris Watkins. Dynamic alignment kernels. In A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 39–50. MIT Press, Cambridge, MA, 2000.

Figure 1: Alexey Chervonenkis (1938–2014)

Journal of Machine Learning Research 16 (2015) 2067-2080 Published 9/15

Alexey Chervonenkis's Bibliography

Alex Gammerman ALEX@CS.RHUL.AC.UK
Vladimir Vovk V.VOVK@RHUL.AC.UK
Computer Learning Research Centre, Department of Computer Science
Royal Holloway, University of London

This bibliography does not contain Alexey's patents (he has at least two), technical reports, unpublished manuscripts, and collections edited by him. "NA" indicates that a journal paper was not assigned to a volume; e.g., it is common for Russian journals (such as *Проблемы управления* and, in some years, *Автоматика и телемеханика*) not to have volumes, and also to have pages numbered separately inside each issue. All papers published by Alexey before 2001 (and afterwards in the case of papers whose original language was Russian) have author lists ordered according to the Cyrillic alphabetic order; for other papers the order may reflect the authors' contributions (people who contributed most tend to be listed first) and administrative positions (bosses tend to be listed last).

The bibliography is given by the year of the original publication (which may be different from the year of the English translation, always given first when available).

1964

[1] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On a class of perceptrons. *Automation and Remote Control*, 25(1):103–109, 1964. Russian original: В. Н. Вапник, А. Я. Червоненкис. Об одном классе персептронов. *Автоматика и телемеханика*, 25(1):112–120, 1964; with English summary entitled "On a perceptron class". The original article submitted on 21 February 1963.

[2] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On a class of pattern-recognition learning algorithms. *Automation and Remote Control*, 25(6):838–845, 1964. Russian original: В. Н. Вапник, А. Я. Червоненкис. Об одном классе алгоритмов обучения распознаванию образов. *Автоматика и телемеханика*, 25(6):937–945, 1964; with English summary entitled "A class of algorithms for pattern recognition learning". The submission date is not given.

[3] Vladimir N. Vapnik, Lyudmila M. Dronfort, and Alexey Ya. Chervonenkis. Some questions of the self-organization of recognizing systems (in Russian). In *Theory and Application of Automatic Systems* (Russian), pages 172–177. Nauka, Moscow, 1964. In the original language: В. Н. Вапник, Л. М. (Людмила Михайловна) Дронфорт, А. Я. Червоненкис. Некоторые вопросы самоорганизации распознающих устройств. *Теория и применение автоматических систем*, сс. 172–177. Наука, Москва, 1964.

1965

[4] Vladimir N. Vapnik, Aleksandr Ya. Lerner, and Alexey Ya. Chervonenkis. The systems of learning in pattern recognition based on generalized portraits (in Russian). *Izvestiya Akademii Nauk SSSR. Tekhnicheskaya Kibernetika*, NA(1), 1965. English translation: Engineering Cybernetics (USSR). In the original language: В. Н. Вапник, А. Я. Лернер, А. Я. Червоненкис. Системы обучения распознаванию образов при помощи обобщенных портретов. *Известия АН СССР, Техническая кибернетика*, №1, 1965.

1966

[5] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Automaton extremal imitation teaching. I. *Automation and Remote Control*, 27(5), 1966. Russian original: В. Н. Вапник, А. Я. Червоненкис. Обучение автомата экстремальной имитации. II. *Автоматика и телемеханика*, №5, 125–135, 1966; with English summary entitled "Automaton extremal imitation teaching. I". The original article submitted on 24 June 1965.

[6] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Automaton extremal imitation teaching. II. *Automation and Remote Control*, 27(6), 1966. Russian original: В. Н. Вапник, А. Я. Червоненкис. Обучение автомата экстремальной имитации. II. *Автоматика и телемеханика*, №6, 120–132, 1966; with English summary entitled "Automaton extremal imitation teaching. II". The original article submitted on 24 June 1965.

1967

[7] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Automaton games with zero sum teaching. *Automation and Remote Control*, 27(6), 1966. Russian original: В. Н. Вапник, А. Я. Червоненкис. Обучение автомата играм с нулевой суммой. *Автоматика и телемеханика*, №7, 113–118, 1967; with English summary entitled "Automaton games with zero sum teaching". The original article submitted on 23 June 1966.

[8] Vladimir N. Vapnik, Aleksandr Ya. Lerner, and Alexey Ya. Chervonenkis. Learning machines and pattern recognition on the basis of generalized portraits. In *Automatic and Remote Control III. Proceedings of the 3rd International Congress of the International Federation of Automatic Control (IFAC),* London, 1966, volume 1, pages 29–27. Institution of Mechanical Engineers, London, 1967. Paper 14F.

1968

[9] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Algorithms with complete memory and recurrent algorithms in pattern recognition learning. *Automation and Remote Control*, 29(4):606–616, 1968. Russian original: В. Н. Вапник, А. Я. Червоненкис. Алгоритмы с полной памятью и рекуррентные алгоритмы в задаче об обучении распознаванию образов. *Автоматика и телемеханика*, №4, 95–106, 1968. The original article submitted on 20 September 1966.

[10] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Uniform convergence of frequencies of occurrence of events to their probabilities. *Soviet Mathematics Doklady*, 9 (4):915–918, 1968. Translated by Lisa Rosenblatt. Russian original: В. Н. Вапник, А. Я. Червоненкис. О равномерной сходимости частот появления событий к их вероятностям. *Доклады Академии Наук СССР*, 181(4):781–783, 1968. The English translation reprinted as Chapter 2 of *Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik* (ed. by Bernhard Schölkopf, Zhiyuan Luo, and Vladimir Vovk), pages 7–12. Springer, Berlin, 2013. The original article submitted on 6 October 1966.

1971

[11] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971. Translated by B. Seckler. Russian original: В. Н. Вапник, А. Я. Червоненкис. О равномерной сходимости частот появления событий к их вероятностям. *Теория вероятностей и ее применения*, 16(2):264–279, 1971; with English summary entitled "On uniform convergence of the frequencies of events to their probabilities". The English translation reprinted as Chapter 3 of *Measures of Complexity: Festschrift for Alexey Chervonenkis* (ed. by Vladimir Vovk, Harris Papadopoulos, and Alex Gammerman), pages 11–30. Springer, Berlin, 2015. The original article submitted on 7 May 1969.

[12] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Theory of uniform convergence of frequencies of events to their probabilities and problems of search for optimal solutions from empirical data. *Automation and Remote Control*, 32(2):207–217, 1971. Russian original: В. Н. Вапник, А. Я. Червоненкис. Теория равномерной сходимости частот появления событий к их вероятностям и задачи поиска оптимального решения по эмпирическим данным. *Автоматика и телемеханика*, №2, 42–53, 1971; with English summary entitled "Theory of uniform convergence of frequencies of appearance of attributes to their probabilities and problems of defining optimal solution by empiric data". The original article submitted on 10 February 1970.

[13] Vladimir N. Vapnik, Aleksandr Ya. Lerner, and Alexey Ya. Chervonenkis. Learning methods in problems of diagnostics (in Russian). In Yakov Z. Tsypkin, editor, *Pattern Recognition. Adaptive Systems. Proceedings of International Symposium on Technological and Biological Control Problems* (Russian), Yerevan, 24–28

September 1968. Nauka, Moscow, 1971. In the original language: В. Н. Вапник, А. Я. Лернер, А. Я. Червоненкис. Методы обучения в задачах диагностики. Сборник *Распознавание образов. Адаптивные системы. Труды Международного симпозиума по техническим и биологическим проблемам управления*, с. 260. Наука, Москва, 1971.

1973

[14] Vladimir N. Vapnik, A. A. Zhuravel', and Alexey Ya. Chervonenkis. Algorithms for learning pattern recognition using the method of generalized portraits. Algorithms GP-1, GP-2, GP-3 (in Russian). In *Algorithms for Learning Pattern Recognition* (Russian), pages 89–109. Soviet Radio (Russian), Moscow, 1973. In the original language: В. Н. Вапник, А. А. Журавель, А. Я. Червоненкис. Алгоритмы обучения распознаванию образов, использующие метод обобщенных портретов. Алгоритмы ОП-1, ОП-2, ОП-3. Сборник *Алгоритмы обучения распознаванию образов* (под ред. В. Н. Вапника), сс. 89–109. Советское радио, Москва, 1973. Sent to the printer on 23 March 1972.

[15] Vladimir N. Vapnik, Glazkova T. G., and Alexey Ya. Chervonenkis. Algorithms for learning pattern recognition using the method of generalized portraits. Algorithms GP-4, GP-5, GP-6, GP-7 (in Russian). In *Algorithms for Learning Pattern Recognition* (Russian), pages 136–150. Soviet Radio (Russian), Moscow, 1973. In the original language: В. Н. Вапник, Т. Г. Глазкова, А. Я. Червоненкис. Алгоритмы обучения распознаванию образов, использующие метод обобщенных портретов. Алгоритмы ОП-4, ОП-5, ОП-6, ОП-7. Сборник *Алгоритмы обучения распознаванию образов* (под ред. В. Н. Вапника), сс. 136–150. Советское радио, Москва, 1973. Sent to the printer on 23 March 1972.

1974

[16] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Ordered risk minimization I. *Automation and Remote Control*, 35(8):1226–1235, 1974. Russian original: В. Н. Вапник, А. Я. Червоненкис. О методе упорядоченной минимизации риска. I. *Автоматика и телемеханика*, №8, 21–30, 1974; with English summary entitled "On the method of ordered risk minimization. I".

[17] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Ordered risk minimization II. *Automation and Remote Control*, 35(9):1403–1412, 1975. Russian original: В. Н. Вапник, А. Я. Червоненкис. О методе упорядоченной минимизации риска. II. *Автоматика и телемеханика*, №9, 29–39, 1974; with English summary entitled "On the method of ordered risk minimization. II".

[18] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. *Theory of Pattern Recognition: Statistical Learning Problems* (in Russian). Nauka, Moscow, 1974. In the original language:

В. Н. Вапник, А. Я. Червоненкис. *Теория распознавания образов: статистические проблемы обучения*. Наука, Москва, 1974. Sent to the printer on 3 July 1973. German translation: W. N. Wapnik, A. J. Tscherwonenkis. *Theorie der Zeichenerkennung* (translated by Klaus-Günter Stöckel and Barbara Schneider, translation edited by Siegfried Unger and Klaus Fritzsch). Elektronisches Rechnen und Regeln, Sonderband (Electronic Computing and Control, Special Issue), vol. 28. Akademie-Verlag, Berlin, 1979.

1975

[19] Leonid I. Mikulich and Alexey Ya. Chervonenkis. On the use of formal calculi in conversational natural language systems (in Russian). In *Proceedings of the 4th Joint Conference on Artificial Intelligence (supplementary materials,* Russian*),* Tbilisi, Georgia, volume 12, pages 176–187. Research Council for Cybernetics, Moscow, 1975. In the original language: Л. И. Микулич, А. Я. Червоненкис. Об использовании формальных исчислений в диалоговых системах. *Труды IV Международной объединенной конференции по искусственному интеллекту (дополнительные материалы)*, Тбилиси, 1975, сс. 176–187. Издательство Научного совета "Кибернетика" (выпуск 12), Москва, 1975. English abstract: On the use of formal calculi in conversational natural language systems. *Firbush News*, No. 7, 1976.

[20] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Asymptotic properties of the method of ordered minimization. *Automation and Remote Control*, 36(12):1986–1999, 1975. Russian original: В. Н. Вапник, А. Я. Червоненкис. Об асимптотических свойствах метода упорядоченной минимизации. Автоматика и телемеханика, №12, 65–77, 1975; with English summary entitled "Asymptotic properties of the method of orderly minimization".

1978

[21] A. S. Belen'kiy, Leonid I. Mikulich, Elena Ya. Naydyonova, and Alexey Ya. Chervonenkis. An interactive data-retrieval system for solving some problems in industrial control system "Morflot" (in Russian). In *Abstracts of the All-Union Scientific and Technological Conference "Software for Industrial Control Systems"*, pages 84–87. Kalinin Technology House, 1978. In the original language: А. С. Беленький, Л. И. Микулич, Е. Я. Найденова, А. Я. Червоненкис. Диалоговая информационно-справочная система для решения некоторых задач в АСУ "Морфлот". *Всесоюзная научно-техническая конференция "Математическое обеспечение АСУ". Тезисы докладов*, сс. 84–87. Калининский дом техники, 1978.

[22] Leonid I. Mikulich and Alexey Ya. Chervonenkis. Linguistic processor for system DISPUT (in Russian). In *Proceedings of the Seminar "Software for Systems of Artificial Intelligence"* (Russian). Moscow House of Scientific and Technological Propaganda (Russian), Moscow, 1978. In the original language: Л. И. Микулич,

А. Я. Червоненкис. Лингвистический процессор системы ДИСПУТ. *Материалы семинара "Информационно-программное обеспечение систем искусственного интеллекта"*, сс. 51–56. Издательство Московского дома научно-технической пропаганды, Москва, 1978.

1979

[23] Leonid I. Mikulich and Alexey Ya. Chervonenkis. A specialized conversational system (in Russian). In *Questions of Design of Applied Systems* (Russian), pages 112–129. Computer Centre of the Siberia Division of the Academy of Sciences of the USSR, Novosibirsk, 1979. In the original language: Л. И. Микулич, А. Я. Червоненкис. Специализированная диалоговая система. Сборник *Вопросы разработки прикладных систем*, сс. 112–129. Издательство ВЦ СО АН СССР, Новосибирск, 1979.

1980

[24] Leonid I. Mikulich and Alexey Ya. Chervonenkis. A conversational question-answer system (DISPUT2) (in Russian). In *Proceedings of the 8th All-Union Conference on Control Problems* (Russian). Tallinn, 1980. In the original language: Л. И. Микулич, А. Я. Червоненкис. Диалоговая вопросно-ответная система (ДИСПУТ2). *Труды VIII Всесоюзного совещания по проблемам управления*. Таллин, 1980.

[25] Leonid I. Mikulich and Alexey Ya. Chervonenkis. Conversational system DISPUT: linguistic and pragmatic processors (in Russian). In *Proceedings of the 2nd International Conference on Artificial Intelligence* (Russian). Research Council for Cybernetics, Moscow, 1980. In the original language: Л. И. Микулич, А. Я. Червоненкис. Диалоговая система ДИСПУТ: лингвистический и прагматический процессоры. *Доклады II Международного совещания по искусственному интеллекту*, Репино. Научный совет "Кибернетика", Москва, 1980.

1981

[26] A. S. Belen'kiy, Leonid I. Mikulich, Elena Ya. Naydyonova, and Alexey Ya. Chervonenkis. DISPUT—An interactive data-retrieval system for planning and management in transportation. 1. Design and operating principles. *Automation and Remote Control*, 42(3):394–401, 1981. Russian original: А. С. Беленький, Л. И. Микулич, Е. Я. Найденова, А. Я. Червоненкис. Диалоговая информационно-справочная система для планирования и управления в транспортных системах (ДИСПУТ). I. Принципы построения и описание функционирования системы "ДИСПУТ". *Автоматика и телемеханика*, №3, 152–162, 1981; with English summary entitled "A dialog data retrieval system for scheduling and management of transportation systems (DISPUT). I. Design principles and functioning".

[27] A. S. Belen'kiy, Leonid I. Mikulich, Elena Ya. Naydyonova, and Alexey Ya. Chervonenkis. DISPUT—an interactive data-retrieval system for planning and management in transportation. 2. System implementation. *Automation and Remote Control*, 42(5):693–701, 1981. Russian original: А. С. Беленький, Л. И. Микулич, Е. Я. Найденова, А. Я. Червоненкис. Диалоговая информационно-справочная система для планирования и управления в транспортных системах (ДИСПУТ). II. Реализация системы "ДИСПУТ". *Автоматика и телемеханика*, №5, 169–180, 1981; with English summary entitled "A dialog data retrieval system for scheduling and management of transportation systems (DISPUT). II".

[28] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Necessary and sufficient conditions for the uniform convergence of means to their expectations. *Theory of Probability and its Applications*, 26(3):532–553, 1982. Translated by W. U. Sirk. Russian original: Необходимые и достаточные условия равномерной сходимости средних к математическим ожиданиям. *Теория вероятностей и ее применения*, 26(3):543–563, 1981; with English summary entitled "Necessary and sufficient conditions for the uniform convergence of empirical means to their true values". The original article submitted on 28 July 1978.

1983

[29] Viktor M. Bryabrin, Yuriy Ya. Lyubarskii, Leonid I. Mikulich, Elena Ya. Naydyonova, Dmitrii A. Pospelov, Aleksandr B. Preobrazhenskii, Vladimir F. Khoroshevskii, and Alexey Ya. Chervonenkis. *Conversational Systems for Industrial Control* (in Russian, ed. by D. A. Pospelov). Energoatomizdat, Moscow, 1983. In the original language: В. М. Брябрин, Ю. Я. Любарский, Л. И. Микулич, Е. Я. Найденова, Д. А. Поспелов, А. Б. Преображенский, В. Ф. Хорошевский, А. Я. Червоненкис. *Диалоговые системы в АСУ* (под ред. Д. А. Поспелова). Энергоатомиздат, Москва, 1983. Chapter 4 was written by Mikulich, Naydyonova, and Chervonenkis.

[30] Alexey V. Kantsel' and Alexey Ya. Chervonenkis. On the mechanism of rhythmic-zonal distribution of mineral formations in the process of evolution of hydrothermal systems (in Russian). *Geology of Ore Deposits* (Russian), 25(5):38–49, 1983. In the original language: А. В. Канцель, А. Я. Червоненкис. О механизме ритмически-зонального распределения минеральных образований в процессе эволюции гидротермальных систем. *Геология рудных месторождений*, 25(5):38–49, 1983.

1984

[31] V. L. Barsukov, A. A. Belyaev, V. S. Serebrennikov, and Alexey Ya. Chervonenkis. A model for periodic processes in seismogenic structures and a statistical method of earthquake forecasting. *Geochemistry International*, 21(1):87–93, 1984.

[32] V. L. Barsukov, A. A. Belyaev, V. S. Serebrennikov, and Alexey Ya. Chervonenkis. Analysis of the dynamics of seismic effects on an observable hydrothermal system.

Geochemistry International, 22(1):168–174, 1985. Russian original: Анализ динамики сейсмического воздействия на наблюдаемую геохимическую систему. *Геохимия*, №8, 1147–1154, 1984.

[33] A. A. Belyaev, Alexey V. Kantsel', V. I. Rekharsky, and Alexey Ya. Chervonenkis. A kinetic model of ore formation and the problem of rhythmic zonation of ore deposits. In T. V. Janelidze and A. G. Tvalchrelidze, editors, *Proceedings of the 6th IAGOD Symposium*, pages 477–480. E. Schweizerbart'sche Verlagsbuch-handlung, Stuttgart, 1984.

[34] Vladimir N. Vapnik, T. G. Glazkova, V. A. Koshcheev, Anatolii I. Mikhalsky, and Alexey Ya. Chervonenkis. *Algorithms and Programs for Reconstructing Dependencies* (in Russian). Nauka, Moscow, 1984. In the original language: В. Н. Вапник, Т. Г. Глазкова, В. А. Кощеев, А. И. Михальский, А. Я. Червоненкис. *Алгоритмы и программы восстановления зависимостей*. Наука, Москва, 1984.

1987

[35] Alexey V. Kantsel, V. I. Rekharsky, and Alexey Ya. Chervonenkis. Nonlinear effects and pulsation development of hydrothermal ore-forming systems (in Russian). *Doklady Akademii Nauk SSSR*, 294(6):1429–1432, 1987. In the original language: А. В. Канцель, В. И. Рехарский, А. Я. Червоненкис. Нелинейные эффекты и пульсационное развитие гидротермальных рудообразующих систем. *Доклады АН СССР*, 294(6):1429–1432, 1987.

[36] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Minimization of expected risk based on empirical data. In *Proceedings of the 1st World Congress of the Bernoulli Society,* Tashkent, 1986, volume 2, pages 821–832. VNU Science Press, Utrecht, 1987.

1989

[37] Alexey V. Kantsel, V. I. Rekharsky, and Alexey Ya. Chervonenkis. Nonlinear effects in hydrothermal ore-forming and rhythmic zoning of mineralization (in Russian). In *Ore-forming Processes and Systems* (Russian), pages 91–102. Nauka, Moscow, 1989. In the original language: А. В. Канцель, В. И. Рехарский, А. Я. Червоненкис. Нелинейные эффекты в гидротермальном рудообразовании и ритмическая зональность оруденения. Сборник *Рудообразующие процессы и системы*, сс. 91–102. Наука, Москва, 1989.

[38] Vladimir N. Vapnik and Alexey Ya. Chervonenkis. Necessary and sufficient conditions of the consistency of the method of empirical risk minimization (in Russian). In Yu. I. Zhuravlev, editor, *Pattern recognition. Classification. Prediction: Mathematical Techniques and their Application* (Russian, with English title), volume 2, pages 207–249. Nauka, Moscow, 1989. In the original language: В. Н. Вапник, А. Я. Червоненкис. Необходимые и достаточные условия состоятельности метода минимизации эмпирического риска. Сборник *Распознавание. Классификация.*

Прогноз: Математические методы и их применение, выпуск 2, сс. 207–249. Наука, Москва, 1989. English translation: V. N. Vapnik and A. Ya. Chervonenkis. The necessary and sufficient conditions for consistency in the empirical risk minimization method. *Pattern Recognition and Image Analysis* (Advances in Mathematical Theory and Applications). 1(3):284–305, 1991.

1990

[39] Alexey V. Kantsel' and Alexey Ya. Chervonenkis. Multistructural model of a hydrothermal geochemical field (in Russian). *Geology of Ore Deposits* (Russian), NA (1):9–20, 1990. In the original language: А. В. Канцель, А. Я. Червоненкис. Мультиструктурная модель гидротермального геохимического поля. *Геология рудных месторождений*, №1, сс. 9–20, 1990.

1998

[40] A. P. Mazurkevich, V. G. Zaikov, Alexey V. Kantsel', A. V. Danilov, V I. Kokushev, and Alexey Ya. Chervonenkis. Computer methods for construction of mathematical models of deposits and calculation of their reserves (in Russian). *Mining Journal*, NA(8):67–68, 1998. In the original language: А. П. Мазуркевич, В. Г. Зайков, А. В. Канцель, А. В. Данилов, В. И. Кокушев, А. Я. Червоненкис. Компьютерные методы построения математических моделей месторождения и подсчет запасов по ним. *Горный журнал*, №8, с. 67–68, 1998.

2000

[41] Yu. Yu. Bakhtin, A. V. Danilov, Alexey V. Kantsel', and Alexey Ya. Chervonenkis. A method of restoration of conditional distributions from empirical data. *Automation and Remote Control*, 61(12):2003–2012, 2000. Russian original: Ю. Ю. Бахтин, А. В. Данилов, А. В. Канцель, А. Я. Червоненкис. Метод восстановления поля условных распределений по эмпирическим данным. *Автоматика и телемеханика*, №12, 75–86, 2000; no English summary.

2001

[42] Alexey Ya. Chervonenkis, Alex Gammerman, and Mark Herbster. A combined Bayes–maximum likelihood method for regression. In Giacomo Della Riccia, Hans-Joachim Lenz, and Rudolf Kruse, editors, *Proceedings of the 5th Workshop on Data Fusion and Perception,* Udine, Italy, 5–7 October 2000, volume 431 of *Courses and Lectures—International Centre for Mechanical Sciences*, pages 25–49. Springer, Vienna, 2001.

2002

[43] Alexey Ya. Chervonenkis. Reconstruction of conditional distribution field based on empirical data. In Ajith Abraham, Javier Ruiz del Solar, and Mario Köppen, editors, *Soft Computing Systems (Design, Management and Application)*, volume 87, pages 462–469. IOS Press, Amsterdam, 2002. The paper presented at HIS'02, the Second International Conference on Hybrid Intelligent Systems, Santiago de Chile, 1–4 December 2002.

[44] Alexey Ya. Chervonenkis. The problem of a deposit resource estimate dependence on the size of elementary excavating volume (in Russian). In *Transactions of the Institute of Control Science (Russian Academy of Sciences)*, volume XV, pages 122–128. Moscow, 2002.

2003

[45] Leo Gordon, Alexey Ya. Chervonenkis, Alex J. Gammerman, Ilham A. Shahmuradov, and Victor V. Solovyev. Genome-wide prokaryotic promoter recognition based on sequence alignment kernel. In Michael R. Berthold, Hans-Joachim Lenz, Elizabeth Bradley, Rudolf Kruse, and Christian Borgelt, editors, *Advances in Intelligent Data Analysis V, Proceedings of the 5th International Symposium on Intelligent Data Analysis, IDA 2003,* Berlin, Germany, 28–30 August 2003, volume 2810 of *Lecture Notes in Computer Science*, pages 386–396. Springer, Berlin, 2003.

[46] Leo Gordon, Alexey Ya. Chervonenkis, Alex J. Gammerman, Ilham A. Shahmuradov, and Victor V. Solovyev. Sequence alignment kernel for recognition of promoter regions. *Bioinformatics*, 19(15):1964–1971, 2003.

2005

[47] Alexey Ya. Chervonenkis. Application of pattern recognition to molecular biology problems (in Russian). *Control Problems* (Russian), NA(4):41–46, 2005. In the original language: А. Я. Червоненкис. Применение методов распознавания образов в задачах молекулярной биологии *Проблемы управления*, №4, 41–46, 2005.

2007

[48] Alexey Ya. Chervonenkis. Discussion of "Hedging predictions in machine learning" by A. Gammerman and V. Vovk. *Computer Journal*, 50(2):164, 2007.

[49] Anatolii I. Mikhalsky and Alexey Ya. Chervonenkis. Application of methods of machine learning for short-term forecasts of the load in energy systems (in Russian). In *Managing the development of large-scale systems. MLSD 2007. Proceedings of the 1st International Conference* (Russian), page 131. Institute of Control

Problems named after V. A. Trapeznikov, Moscow, 2007. In the original language: А. И. Михальский, А. Я. Червоненкис. Применение методов машинного обучения для краткосрочного прогноза нагрузки в энергосистемах. Сборник *Управление развитием крупномасштабных систем. MLSD 2007. Тезисы докладов первой международной конференции*, с. 131. Институт проблем управления им. В. А. Трапезникова, Москва, 2007.

2008

[50] Alexey Ya. Chervonenkis. On reconstructing dependencies via experimental data (in Russian). In *Proceedings of the 7th International Conference "System Identification and Control Problems", SICPRO'08,* Moscow, 28–31 January 2008. Institute of Control Problems named after V. A. Trapeznikov, Moscow, 2008. In the original language: А. Я. Червоненкис. Проблемы восстановления зависимостей по эмпирическим данным. *Труды VII Международной конференции "Идентификация систем и задач управления", SICPRO'08*, Москва, 28–31 января 2008 г. Институт проблем управления им. В. А. Трапезникова РАН, Москва, 2008.

[51] Alex Gammerman, Ilia Nouretdinov, Brian Burford, Alexey Chervonenkis, Vladimir Vovk, and Zhiyuan Luo. Clinical mass spectrometry proteomic diagnosis by conformal predictors. *Statistical Applications in Genetics and Molecular Biology*, 7(2):Article 13 (12 pp.), 2008.

2009

[52] Alexey Ya. Chervonenkis. *Computer Data Analysis (in Russian).* Yandex, Moscow, 2009. In the original language: А. Я. Червоненкис. Компьютерный анализ данных. Яндекс, Москва, 2009. Lectures at the Yandex School of Data Analysis.

[53] Alex Gammerman, Vladimir Vovk, Brian Burford, Ilia Nouretdinov, Zhiyuan Luo, Alexey Ya. Chervonenkis, Mike Waterfield, Rainer Cramer, Paul Tempst, Josep Villanueva, Musarat Kabir, Stephane Camuzeaux, John Timms, Usha Menon, and Ian Jacobs. Serum proteomic abnormality predating screen detection of ovarian cancer. *Computer Journal*, 52(3):326–333, 2009.

2011

[54] Alexey Ya. Chervonenkis. Problems of machine learning. In Sergei O. Kuznetsov, Deba P. Mandal, Malay K. Kundu, and Sankar K. Pal, editors, *Pattern Recognition and Machine Intelligence: Proceedings of the 4th International Conference, PReMI 2011,* Moscow, Russia, June 27 – July 1, 2011, volume 6744, pages 21–23. Springer, Berlin, 2011.

[55] Alexey Ya. Chervonenkis, S. S. Chernova, and T. V. Zykova. Applications of kernel ridge estimation to the problem of computing the aerodynamical characteristics of a passenger plane (in comparison with results obtained with artificial neural networks). *Automation and Remote Control*, 72(5):1061–1067, 2011. Russian original: А. Я. Червоненкис, С. С. Чернова, Т. В. Зыкова. Применение ядерной гребневой оценки к задаче расчета аэродинамических характеристик пассажирского самолета (сравнение с результатами, полученными с использованием искусственных нейронных сетей). *Автоматика и телемеханика*, №5, 175–182, 2011.

[56] Ilia Nouretdinov, Sergi G. Costafreda, Alex Gammerman, Alexey Ya. Chervonenkis, Vladimir Vovk, Vladimir Vapnik, and Cynthia H. Y. Fu. Machine learning classification with confidence: Application of transductive conformal predictors to MRI-based diagnostic and prognostic markers in depression. *NeuroImage*, 56(2):809–813, 2011.

2012

[57] Alexey Ya. Chervonenkis. On some properties of classes of events for which the conditions for the uniform convergence of the relative frequencies to probabilities fail to hold. *Izvestiya Mathematics*, 76(6):1271–1285, 2012. Translated by I. Shtern. Russian original: О некоторых свойствах классов событий, для которых не выполнены условия равномерной сходимости частот к вероятностям. *Известия РАН, Серия математическая*, 76(6):207–221, 2012.

[58] Alexey Ya. Chervonenkis. Some properties of infinite VC-dimension systems. In Mireille Gettler Summa, Léon Bottou, Bernard Goldfarb, Fionn Murtagh, Catherine Pardoux, and Myriam Touati, editors, *Statistical Learning and Data Science*, chapter 5, pages 53–59. CRC Press, Boca Raton, FL, 2012.

[59] Dmitry Devetyarov, Ilia Nouretdinov, Brian Burford, Stephane Camuzeaux, Aleksandra Gentry-Maharaj, Ali Tiss, Celia J. Smith, Zhiyuan Luo, Alexey Ya. Chervonenkis, Rachel Hallett, Vladimir Vovk, Mike Waterfield, Rainer Cramer, John Timms, John Sinclair, Usha Menon, Ian Jacobs, and Alex Gammerman. Conformal predictors in early diagnostics of ovarian and breast cancers. *Progress in AI*, 1(3):245–257, 2012.

[60] Ilia Nouretdinov, Dmitry Devetyarov, Brian Burford, Stephane Camuzeaux, Aleksandra Gentry-Maharaj, Ali Tiss, Celia Smith, Zhiyuan Luo, Alexey Chervonenkis, Rachel Hallett, Vladimir Vovk, Mike Waterfield, Rainer Cramer, John F. Timms, Ian Jacobs, Usha Menon, and Alex Gammerman. Multiprobabilistic Venn predictors with logistic regression. In Lazaros Iliadis, Ilias Maglogiannis, Harris Papadopoulos, Kostas Karatzas, and Spyros Sioutas, editors, *Proceedings of the AIAI 2012 Workshop on Conformal Prediction and its Applications*, volume 382 of *IFIP Advances in Information and Communication Technology*, pages 224–233. Springer, Berlin, 2012.

2013

[61] Alexey Chervonenkis. Early history of support vector machines. In Bernhard Schölkopf, Zhiyuan Luo, and Vladimir Vovk, editors, *Empirical Inference: A Festschrift in Honor of Vladimir N. Vapnik*, pages 13–20. Springer, Berlin, 2013.

[62] Alexey Ya. Chervonenkis. Measures of complexity. In Harris Papadopoulos, Andreas S. Andreou, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Intelligence Applications and Innovations*, pages xvii–xviii. Springer, Heidelberg, 2013.

[63] Alexey Chervonenkis, Anna Sorokina, and Valery A. Topinsky. Optimization of ads allocation in sponsored search. In Leslie Carr, Alberto H. F. Laender, Bernadette Farias Lóscio, Irwin King, Marcus Fontoura, Denny Vrandecic, Lora Aroyo, José Palazzo M. de Oliveira, Fernanda Lima, and Erik Wilde, editors, *WWW 2013 Companion – Proceedings of the 22nd International Conference on World Wide Web 2013,* 13–17 May 2013, Rio de Janeiro, Brazil, pages 121–122. International World Wide Web Conferences Steering Committee / ACM, 2013.

[64] A. N. Kornetova and Alexey Ya. Chervonenkis. Optimization of advertising display in search systems (in Russian). *Control Problems* (Russian), NA(1):40–49, 2013. In the original language: А. Н. Корнетова, А. Я. Червоненкис. Оптимизация показов рекламы в поисковых системах. *Проблемы управления*, №1, 40–49, 2013.

2014

[65] Anna N. Sorokina and Alexey Ya. Chervonenkis. An improved optimization algorithm of ads' allocation in sponsored search and the results of experiments. *Automation and Remote Control*, 76(7):1315–1325, 2015. Russian original: А. Н. Сорокина, А. Я. Червоненкис. Усовершенствованный алгоритм оптимизации показов рекламы в поисковых системах и результаты экспериментов. *Проблемы управления*, №3, 57–63, 2014.

2015

[66] Alexey Chervonenkis. Chervonenkis's recollections. In Vladimir Vovk, Harris Papadopoulos, and Alex Gammerman, editors, *A Festschrift for Alexey Chervonenkis*, pages 3–8. Springer, Berlin, 2015.

[67] Alexey Chervonenkis. Measures of complexity in the theory of machine learning. In Vladimir Vovk, Harris Papadopoulos, and Alex Gammerman, editors, *A Festschrift for Alexey Chervonenkis*, pages 171–183. Springer, Berlin, 2015.

[68] Ilia Nouretdinov, Dmitry Devetyarov, Vladimir Vovk, Brian Burford, Stephane Camuzeaux, Aleksandra Gentry-Maharaj, Ali Tiss, Celia Smith, Zhiyuan Luo, Alexey

Chervonenkis, Rachel Hallett, Mike Waterfield, Rainer Cramer, John F. Timms, Ian Jacobs, Usha Menon, and Alex Gammerman. Multiprobabilistic prediction in early medical diagnoses. *Annals of Mathematics and Artificial Intelligence*, 74(1–2):203–222, 2015.

Journal of Machine Learning Research 16 (2015) 2081-2097 Submitted 8/14; Revised 11/14; Published 10/15

Photonic Delay Systems as Machine Learning Implementations

Michiel Hermans MICHIEL.HERMANS@ULB.AC.BE
OPERA - Photonics Group
Université Libre de Bruxelles
Avenue F. Roosevelt 50, 1050 Brussels, Belgium

Miguel C. Soriano MIGUEL@IFISC.UIB-CSIC.ES
Instituto de Física Interdisciplinar y Sistemas Complejos, IFISC (UIB-CSIC)
Campus Universitat de les Illes Balears
E-07122 Palma de Mallorca, Spain

Joni Dambre JONI.DAMBRE@UGENT.BE
ELIS departement
Ghent University
Sint Pietersnieuwstraat 41, 9000 Ghent, Belgium

Peter Bienstman PETER.BIENSTMAN@UGENT.BE
INTEC departement
Ghent University
Sint Pietersnieuwstraat 41, 9000 Ghent, Belgium

Ingo Fischer INGO@IFISC.UIB-CSIC.ES
Instituto de Física Interdisciplinar y Sistemas Complejos, IFISC (UIB-CSIC)
Campus Universitat de les Illes Balears
E-07122 Palma de Mallorca, Spain

Editor: Yoshua Bengio

Abstract

Nonlinear photonic delay systems present interesting implementation platforms for machine learning models. They can be extremely fast, offer great degrees of parallelism and potentially consume far less power than digital processors. So far they have been successfully employed for signal processing using the Reservoir Computing paradigm. In this paper we show that their range of applicability can be greatly extended if we use gradient descent with backpropagation through time on a model of the system to optimize the input encoding of such systems. We perform physical experiments that demonstrate that the obtained input encodings work well in reality, and we show that optimized systems perform significantly better than the common Reservoir Computing approach. The results presented here demonstrate that common gradient descent techniques from machine learning may well be applicable on physical neuro-inspired analog computers.

Keywords: recurrent neural networks, optical computing, machine learning models

1. Introduction

Applied research in neural networks is currently strongly influenced by available computer architectures. Most strikingly, the increasing availability of general-purpose graphical processing unit (GPGPU) programming has sped up the computations required for training (deep) neural networks by an order of magnitude. This development allowed researchers to dramatically scale up their models, in turn leading to the major improvements on state-of-the-art performances on tasks such as computer vision (Krizhevsky et al., 2012; Cireşan et al., 2010).

One class of neural models which has only seen limited effects of the boost in speed from GPUs are recurrent models. Recurrent neural networks (RNNs) are very interesting for processing time series, as they can take into account an arbitrarily long context of their input history. This has important implications in tasks such as natural language processing, where the desired output of the system may depend on context that has been presented to the network a relatively long time ago. In common feedforward networks such dependencies are very hard to include without scaling up the model to an impractically large size. Recurrent networks, however, can–at least in principle–carry along relevant context as they are being updated.

In practice, recurrent models suffer from two important drawbacks. First of all, where feedforward networks fully benefit from massively parallel architectures in terms of scalability, recurrent networks, with their inherently sequential nature do not fit so well into this framework. Even though GPUs have been used to speed up training RNNs (Sutskever et al., 2011; Hermans and Schrauwen, 2013), the total obtainable acceleration for a given GPU architecture will still be limited by the number of sequential operations required in an RNN, which is typically much higher than in common neural networks. The second issue is that training RNNs is a notoriously slow process due to problems associated with fading gradients, which is especially cumbersome if the network needs to learn long-term dependencies within the input time series. Recent attempts to solve this problem using the Hessian-free approach have proved promising (Martens and Sutskever, 2011). Other attempts using stochastic gradient descent combined with more heuristic ideas have been described in Bengio et al. (2013).

In this paper we will consider a radical alternative to common, digitally implemented RNNs. A steadily growing branch of research is concerned with *Reservoir Computing* (RC), a concept which employs high-dimensional, randomly initialized dynamical systems (termed the *reservoir*) to perform feature extraction on time series (Jaeger, 2001; Jaeger and Haas, 2004; Maass et al., 2002; Steil, 2004; Lukosevicius and Jaeger, 2009). Despite its simplicity, RC has several important advantages over traditional gradient descent training methods. First of all, the training process is extremely fast. Only output weights are trained, and this is performed by solving a single linear system of equations. Second, and of great importance, the RC concept is applicable to *any* non-linear dynamical system, as long as it exhibits consistent responses, a high-dimensional state space, and fading memory. This has opened lines of research that go beyond common digital implementations and into analog physical implementations. The RC concept has been demonstrated to work on a variety of physical implementation platforms, such as water ripples (Fernando and Sojakka, 2003), mechanical constructs and tensegrity structures (Caluwaerts et al., 2013; Hauser et al., 2011), electro-

optical devices (Larger et al., 2012; Paquot et al., 2012), fully optical devices (Brunner et al., 2013) and nanophotonic circuits (Vandoorne et al., 2008, 2014). As opposed to digital implementations, physical systems can offer great speed-ups, inherent massive parallelism, and great reductions in power consumption. In this sense, physical dynamical systems as machine learning implementation platforms may one day break important barriers in terms of scalability. In the near future, especially optical computing devices might find applications in several tasks where fast processing is essential, such as in optical header recognition, optical signal recovery, or fast control loops.

The RC paradigm, despite its notable successes, still suffers from an important drawback. Its inherently unoptimized nature makes it relatively inefficient for many important machine learning problems. When the dimensionality of the input time series is low, the expansion into a high-dimensional nonlinear space offered by the reservoir will provide a sufficiently diverse set of features to approximate the desired output. If the input dimensionality becomes larger, however, relying on random features becomes increasingly difficult as the space of possible features becomes so massive. Here, optimization with gradient descent still has an important edge over the RC concept: it can shape the necessary nonlinear features automatically from the data.

In this paper we aim to integrate the concept of gradient descent in neural networks with physically implemented analog machine learning models. Specifically, we will employ a physical dynamical system that has been studied extensively from the RC paradigm, a delayed feedback electro-optical system (Larger et al., 2012; Paquot et al., 2012; Soriano et al., 2013). In order to use such a system as a reservoir, an input time series is encoded into a continuous time signal and subsequently used to drive the dynamics of the physical setup. The response of the device is recorded and converted to a high-dimensional feature set, which in turn is used with linear regression in the common RC setup. In this particular case, the randomness of RC is incorporated in the input encoding. This encoding is performed offline on a computer, but is usually completely random. Even though efforts have been performed to improve this encoding in a generic way (by ensuring a high diversity in the network's response, discussed in Rodan and Tino (2011) and Appeltant et al. (2014)), a way to create task-specific input encodings is still lacking.

In Hermans et al. (2014b), the possibility to use *backpropagation through time* (BPTT) (Rumelhart et al., 1986) as a generic optimization tool for physical dynamical systems was addressed. It was found that BPTT can be used to find remarkably intricate solutions to complicated problems in dynamical system design. In Hermans et al. (2014a) simulated results of BPTT used as an optimization method for input encoding in the physical system described above were presented. In this paper we go beyond this work and show for the first time experimental evidence that model-based BPTT is a viable training strategy for physical dynamical systems. We choose two often-used high-dimensional data sets for validation, and we show that input encoding that is optimized using BPTT in a common machine learning approach, provides a significant boost in performance for these tasks when compared to random input encodings. This not only demonstrates that machine learning approaches are more broadly applicable than is generally assumed, but also that physical analog computers can in fact be considered as parametrizable machine learning models, and may play a significant role in the next generation of signal processing hardware.

This paper is structured as follows: first of all we discuss the physical system and its corre-

sponding model in detail. We explain how we convert the continuous-time dynamics of the system into a discrete-time update equation which we use as model in our simulation. Next, we present and analyze the results on the tasks we considered and compare experimental and simulated results.

2. Physical System

In this section we will explain the details of the physical system. We will start by formally introducing its delay dynamics operating in continuous time. Next, we will explain how the feedback delay can be used for realizing a high-dimensional state space encoded in time, and we demonstrate that–combined with special input and output encoding–the setup can be seen as a special case of RNN. Finally we explain how we discretize the system's input and output encoding, which enables us to approximate the dynamics of the system by a discrete-time update equation.

The physical system we employ in this paper is a delayed feedback system exhibiting Ikeda-type dynamics (Larger et al., 2004; Weicker et al., 2012). We provide a schematic depiction of the physical setup in Figure 1. It consists of a laser source, a Mach-Zehnder modulator, a long optical fiber (≈ 4 km) which acts as a physical delay line, and an electronic circuit which transforms the optical beam intensity in the fiber into a voltage. This voltage is amplified and low-pass filtered and can be measured to serve as the system output. Moreover, it is added to an external input voltage signal, and then serves as the driving signal for the Mach-Zehnder modulator. The measured output signal is well described by the following differential equation (Larger et al., 2012):

$$T\dot{a}(t) = -a(t) + \beta \left[\sin^2(a(t-D) + z(t) + \phi) - 1/2\right]. \tag{1}$$

Here, the signal $a(t)$ corresponds to a measured voltage signal (down to a constant scaling and bias factor). The factor T is the time scale of the low-pass filtering operation in the electronic circuit, equal to 0.241 µs, β is the total amplification in the loop, which in the experiments can be varied by changing the power of the laser source. D is the delay of the system, which has been chosen as 20.82 µs. $z(t)$ is the external input signal, and ϕ is a constant offset phase (which can be controlled by setting a bias voltage), which we set at $\pi/4$ for all results presented in this paper. For ease of notation we will call the system a *delay-coupled Mach-Zehnder*, which we abbreviate as DCMZ.

Note that the parameters β and ϕ, together with the global scaling of the input signal $z(t)$, control the global dynamical behavior of the system (Larger et al., 2012). Indeed, previous research in the RC context have identified the role of these parameters in connection with task performance. They found that good performance is usually found when the parameters put the system in an asymptotically stable regime. For instance, if we keep $\phi = \pi/4$, and $\beta < 1$, the system state will always fall back to zero in the absence of input. In the case of $\beta > 1$, the state of the system will spontaneously start to oscillate, which has a detrimental effect on task performance. In this paper we will simply use values for β and ϕ that were found to generally work well in the reservoir setup.

2.1 Input and Output Encoding

Delay-coupled systems have–in principle–an infinite-dimensional state space, as these systems directly depend on their full history covering an interval of one delay time. This property has been the initial motivation for using delay-coupled systems in the RC paradigm in the past years. Suppose we have a multivariate input time series, which we will denote by $\mathbf{s}_i$, for $i \in \{1, 2, \cdots, S\}$, S being the total number of instances (the length of the input sequence). Each $\mathbf{s}_i$ is a column vector of size $N_{\text{in}} \times 1$, with N_{in} the number of input dimensions. We wish to construct an accompanying output time series $\mathbf{y}_i$. We convert each data point $\mathbf{s}_i$ to a continuous-time segment $z_i(t)$ as follows:

$$z_i(t) = m_0(t) + \mathbf{m}^{\mathsf{T}}(t)\mathbf{s}_i,$$

where $m_0(t)$ and $\mathbf{m}(t)$ are *masking signals*, which are defined for $t \in [0 \cdots P]$, with P the masking period. The signal $m_0(t)$ is scalar, and constitutes a bias signal, and $\mathbf{m}(t)$ is a column vector of size $N_{\text{in}} \times 1$. The total input signal $z(t)$ is then constructed by time-concatenation of the segments $z_i(t)$:

$$z(t) = z_i(t \bmod P) \quad \text{for} \quad t \in \{(i-1)P \cdots iP\}.$$

Similarly, we define an output mask $\mathbf{u}(t)$. We divide the state variable time traces $a(t)$ in segments $a_i(t)$ of duration P such that

$$a(t) = a_i(t \bmod P) \quad \text{for} \quad t \in \{(i-1)P \cdots iP\}.$$

The output time series $\mathbf{y}_i$ is then defined as

$$\mathbf{y}_i = \mathbf{y}_0 + \int_0^P dt\ a_i(t)\mathbf{u}(t). \tag{2}$$

It is possible to see the delay-coupled dynamical system combined with the masking principle as a special case of an infinite-dimensional discrete-time recurrent neural network, as illustrated in Figure 2. The recurrent weights, connecting the hidden states over time, are fixed, and manifested by the delayed feedback connection. The input and output weights correspond to the input and output masks.

The role of the parameters D and P is important to consider. If they are equal to each other the recurrent network analogy, as shown in Figure 2b, reduces to a network where all nodes have self-connections, and interaction between different nodes between different tasking periods is due to a combination of the low-pass filtering effect and the self-connection. If the difference between D and P is small, there will be direct time-interaction between different nodes. In fact, using a difference of one masking step between D and P has been the basis for opto-electronic systems that do not have a low-pass filter (Paquot et al., 2012). If the difference between D and P becomes significant it is difficult to anticipate how performance will be affected. If $D \ll P$, most interactions will happen within a single masking period, such that there will be little useful interaction between the nodes at different time steps. If $D \gg P$, the nodes interact over connections that bridge several time steps. We found that, for small differences of D and P, there is little to no noticeable effect on performance, such that we kept $D = P$, as was used in previous publications.

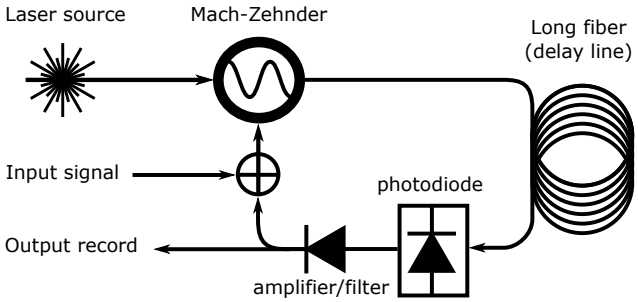


Figure 1: Schematic depiction of a delay-coupled Mach-Zehnder interferometer.

In practice, we cannot measure the state trajectory with infinite time resolution, nor can we produce signals with an arbitrary time dependency, as there will always be constraints that limit the maximum bandwidth of the generated signals. Therefore, we assume that $m_0(t)$, $\mathbf{m}(t)$ and $\mathbf{u}(t)$ all consist of piecewise constant signals[1], which are segmented in N_m parts, N_m being the number of *masking steps*:

$$\begin{aligned} m_0(t) &= m_{0k} \quad \text{for} \quad t \in \{(k-1)P_m \cdots kP_m\}, \\ \mathbf{m}(t) &= \mathbf{m}_k \quad \text{for} \quad t \in \{(k-1)P_m \cdots kP_m\}, \\ \mathbf{u}(t) &= \mathbf{u}_k \quad \text{for} \quad t \in \{(k-1)P_m \cdots kP_m\}, \end{aligned} \tag{3}$$

where the length of each step is given by $P_m = P/N_m$. This means that we now have a finite number of parameters that fully determine $m_0(t)$, $\mathbf{m}(t)$ and $\mathbf{u}(t)$. Note that, due to our choice of $P = D$, P_m will by definition be an integer number of times the delay length D. This is convenient for the next section, where we will make a discrete-time approximation of the system, but it is not a necessary requirement of the system to perform well.

2.2 Converting the System to a Trainable Machine Learning Model

In Hermans et al. (2014b) it was shown that BPTT can be applied to models of continuous-time dynamical systems. Indeed, it is perfectly possible to simulate the system using differential equation solvers and consequently compute parameter gradients. One issue, however, is the significant computational cost. Note that, in a common discrete-time RNN, a single state update corresponds to a single matrix-vector multiplication and the application of a nonlinearity. In our case it involves the sequential computation of the full time trace of $a_i(t)$. This is considerably more costly to compute, especially given the fact that–as in most gradient descent algorithms–we may need to compute it on large amounts of data and this for multiple thousands of iterations.
Due to the piecewise constant definition of $\mathbf{u}(t)$ we can make a good approximation of $a(t)$.

1. Note that with a finite frequency bandwidth we cannot produce immediate jumps from one constant level to the next. Therefore, we make sure that the duration of each constant part is much longer than the transient in between, and we can safely ignore it.

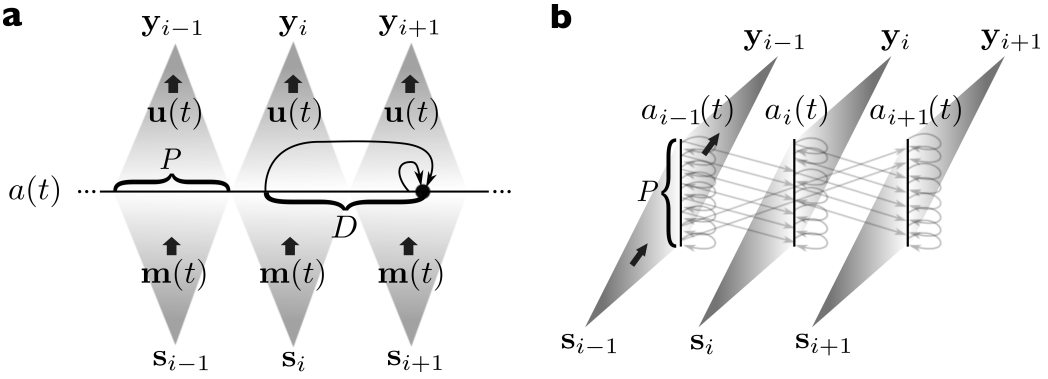


Figure 2: Schematic representation of the masking principle. **a**: Depiction of the input time series $\mathbf{s}_i$ and the way it is converted into a continuous-time signal by means of the input masking signals $\mathbf{m}(t)$. The horizontal line in the middle shows the time evolution of the system state $a(t)$. We have depicted two connection arrows at one point in time, which indicate that $a(t)$ depends on its immediately preceding value (due to the low-pass filtering operation), and its delayed value. The state trajectories are divided into segments each of which are projected to an output instance $\mathbf{y}_i$. **b**: The same picture as in panel **a**, but now represented as a time-unfolded RNN. We have shown the connections between the states as light grey arrows, but note that there are in principle infinitely many connections.)

First we combine Equations 2 and 3. This gives us:

$$\mathbf{y}_i = \sum_{k=1}^{N_m} \int_{(k-1)P_m}^{kP_m} dt\ \mathbf{u}_k a_i(t) = \sum_{k=1}^{N_m} \mathbf{u}_k \bar{a}_{ik},$$

where $\bar{a}_{ik} = \int_{(k-1)P_m}^{kP_m} dt\ a_i(t)$. This means that we can represent $a_i(t)$ by a finite set of variables $\bar{a}_{ik}$. To represent the full time trace of $a(t)$ we adopt a simplified notation as follows[2]: $\bar{a}_j = \bar{a}_{ik}$, where $j = (i-1)N_m + k$.

Now we make the following approximation: we assume that for the duration of a single masking step, we can replace the term $a(t-D)$ by $\bar{a}_{i-N_m}$, that is, we consider it to be constant. With this assumption, we can solve Equation 1 for the duration of one masking step:

$$a(t) = \gamma_i + (\hat{a}_i - \gamma_i) \exp\left(-\frac{t}{T}\right) \qquad \text{for} \qquad t \in \{0 \cdots P_m\}, \tag{4}$$

with

$$\gamma_i = \beta \left[\sin^2(\bar{a}_{i-N_m} + z(t) + \phi) - 1/2\right],$$

and $\hat{a}_i$ the value of $a(t)$ at the start of the interval. Integrating over the interval $t = \{0 \cdots P_m\}$ we find:

$$\bar{a}_i = (\hat{a}_i - \gamma_i)\kappa + P_m \gamma_i,$$

2. Please do not confuse with the index i in $a_i(t)$. Here the index indicates single masking steps, rather than full mask periods.

with $\kappa = 1 - \mathrm{e}^{-P_m/T}$. We can eliminate $\hat{a}_i$ as follows. First we derive from Equation 4 that $\hat{a}_{i+1} = (\hat{a}_i - \gamma_i)\mathrm{e}^{-P_m/T} + \gamma_i$. If we combine this expression with the following two:

$$\bar{a}_i = (\hat{a}_i - \gamma_i)\kappa + P_m\gamma_i,$$

$$\bar{a}_{i+1} = (\hat{a}_{i+1} - \gamma_{i+1})\kappa + P_m\gamma_{i+1},$$

we can eliminate $\hat{a}_i$, and we end up with the following update equation for $\bar{a}_i$:

$$\bar{a}_{i+1} = \rho_o\bar{a}_i + \rho_1\gamma_i + \rho_2\gamma_{i+1},$$

with $\rho_0 = \mathrm{e}^{-P_m/T}$, $\rho_1 = T\kappa - P_m\mathrm{e}^{-P_m/T}$, and $\rho_2 = P_m - T\kappa$. This leads to a relatively quick-to-compute update equation to simulate the system. BPTT can also be readily applied on this formula, as it is a simple update equation just like for a common RNN. This is the simulation model we used for training the input and output masks of the system.
We verified the accuracy of this approximation both on measured data of the DCMZ and on a highly accurate simulation of the system. For the parameters used in the DCMZ we got very good correspondence with the model (obtaining a correlation coefficient between simulated and measured signals of 99.6%).

2.3 Hybrid Training Approach

One challenge we faced when trying to match the model with the experimentally measured data was that we obtained a sufficiently good correspondence only when we very carefully fitted the values for β and ϕ. We can physically control these parameters, but exactly setting their numerical values turned out not to be trivial in the experiments, especially since they tend to show slight drifting behavior over longer periods of time (in the order of hours). As a consequence, it turned out to be a challenge to train parameters in simulation, and simply apply them directly on the DCMZ. Therefore, we applied a hybrid approach between gradient descent and the RC approach. We train both the input and output masks in simulations. Next, we only use the input masks for the physical setup. After recording all the data, we retrained the output weights using gradient descent, this time on the measured data itself. The idea is that the input encoding will produce highly useful features for the system even when it is trained on a model that may show small, systematic differences with the physical setup.

2.4 Input Limitations

One additional physical constraint is the fact that the voltages that can be generated by the electronic part of the system are limited within a range set by its supply voltage. The output voltage of the electronic part serves as the input of the Mach-Zehnder interferometer, and corresponds to the term $a(t-D)+z(t)$ in the argument of the squared sine in Equation 1 (the offset phase ϕ is controlled by a separate voltage source). The voltage range we were able to cover before the amplifiers started to saturate, roughly corresponded to a range of $[-\pi/2 \cdots \pi/2]$ in Equation 1: one full wavelength. Instead of accounting for the saturation of the amplifiers in our simulations, we made sure that when the input argument $z(t)$ went outside of this range, we mapped it back into this range by adding or subtracting π. Note that this has no effect on Equation 1 due to the periodicity of the squared sine. Due to the

addition of the input signal with the delayed feedback $a(t-D)$, there is still a chance that the total argument falls out of the range $[-\pi/2 \cdots \pi/2]$, but in practice such occurrences turned out to be rare, and could safely be ignored.

3. Experiments

We tested the use of BPTT for training the input masks both in simulation and in experiment on two benchmark tasks. First, we considered the often-used MNIST written digit recognition data set, where we use the dynamics of the system indirectly. Next, we applied it on the TIMIT phoneme data set. For the MNIST experiment we used $N_m = 400$ masking steps. For TIMIT we used $N_m = 600$.

3.1 MNIST

To classify static images using a dynamical system, we follow an approach similar to the one introduced in Rolfe and LeCun (2013). Essentially, we repeat the same input segment several times until the state vector $a_i(t)$ of the DCMZ no longer changes. Next we choose the final instance of $a_i(t)$ to classify the image. In practice we used 10 iterations for each image in the MNIST data set (i.e., each input digit is repeated for 10 masking periods). This sufficed for $a_i(t)$ to no longer depend on its initial conditions, and in practice this meant that we were able to present all digits to the network right after each other.
Input masks were trained using 10^6 training iterations, where for each iteration the gradient was determined on 500 randomly sampled digits. For training we used Nesterov momentum (Sutskever et al., 2013), with momentum coefficient 0.9, and a learning rate of 0.01 which linearly decayed to zero over the duration of the training. As regularization we only performed 1-pixel shifts for the digits. Note that these 1-pixel shifts were used for training the input masks, but we did not include them when retraining the output weights, as we only presented the DCMZ with the original 60,000 training examples.
After training the input weights, we gathered both physical and simulated data for the 4 experiments as described below, and retrained the output weights to obtain a final score. Output weights are trained using the cross-entropy loss function over 10^6 training iterations, where for each iteration the gradient was determined on 1000 randomly sampled digits. We again used Nesterov momentum, with momentum coefficient 0.9. The learning rate was chosen at 0.002 and linearly decayed to zero. Meta-parameter optimization was performed using 10,000 randomly selected examples from the training set.
We performed 4 tests on MNIST. First of all we directly compared performances between the simulated and experimental data. When we visualized the features that the trained input masks generated, we noticed that they seemed ordered (see Figure 3). Indeed, for each masking step, a single set of weights $\mathbf{m}_k$, which can be seen as a receptive field, is applied to the input image, and the resulting signals from the receptive fields are injected into the physical setup one after each other. Apparently, the trained input masks have similar features grouped together in time. To confirm that this ordering in time is a purposeful property, we shuffled the features $\mathbf{m}_k$ over a single masking period to obtain a new input mask without a natural ordering in the features. Next we tested (in simulation) how much the performance degraded when using these masks. Finally, we also tested classification employing masks with completely random elements, where only the scaling of the weights

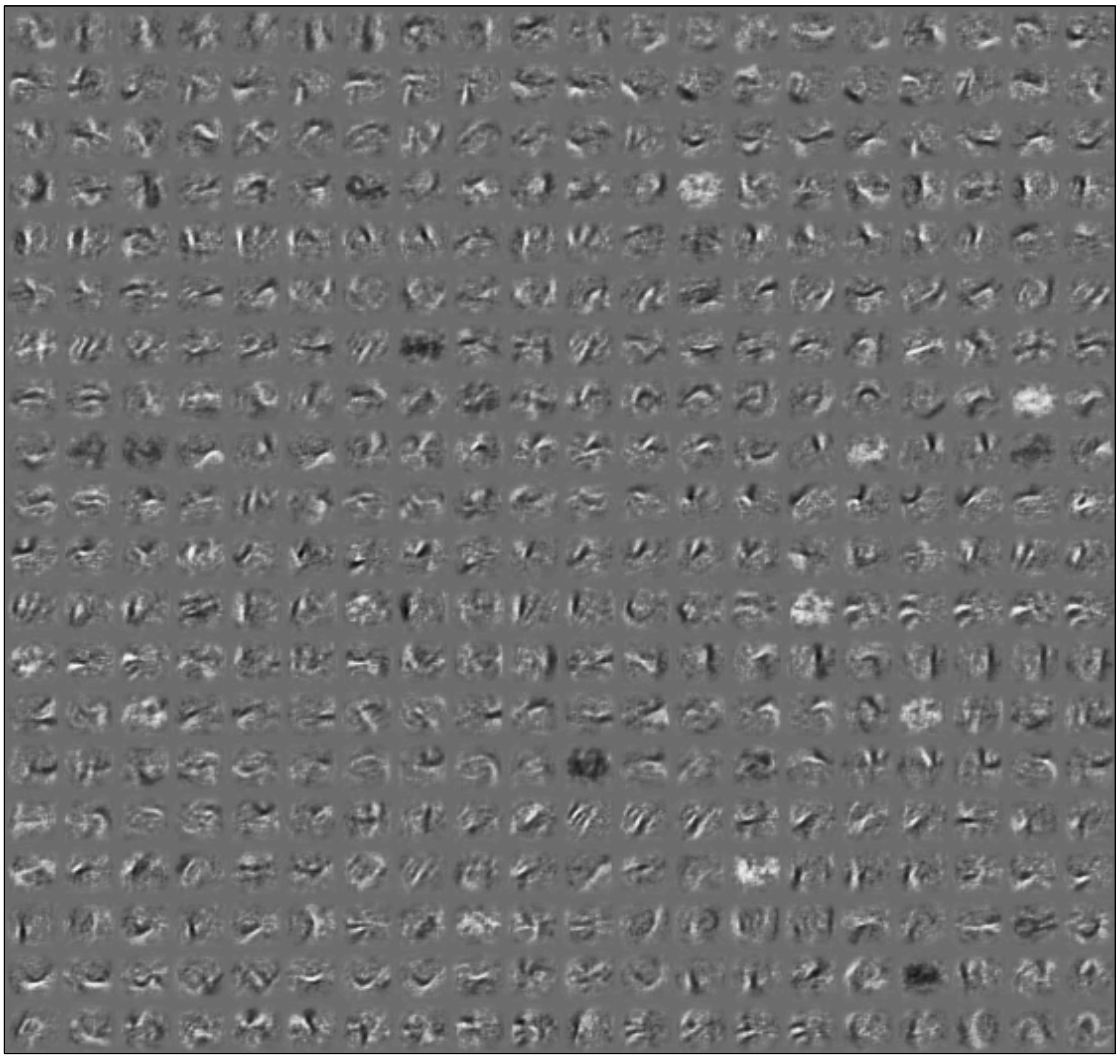


Figure 3: Depiction of the image features present in the input masks for the MNIST task. We have shown the input weights of the 400 masking steps, which we have reshaped into a 20×20 grid of 28×28 pixel representations, corresponding to the receptive fields of each masking step (which can be considered virtual "neurons"). Time (progression of the masking steps, and hence physical time) runs row by row. Notice that the order in which they occur is not random, but rather similar features are grouped in time.

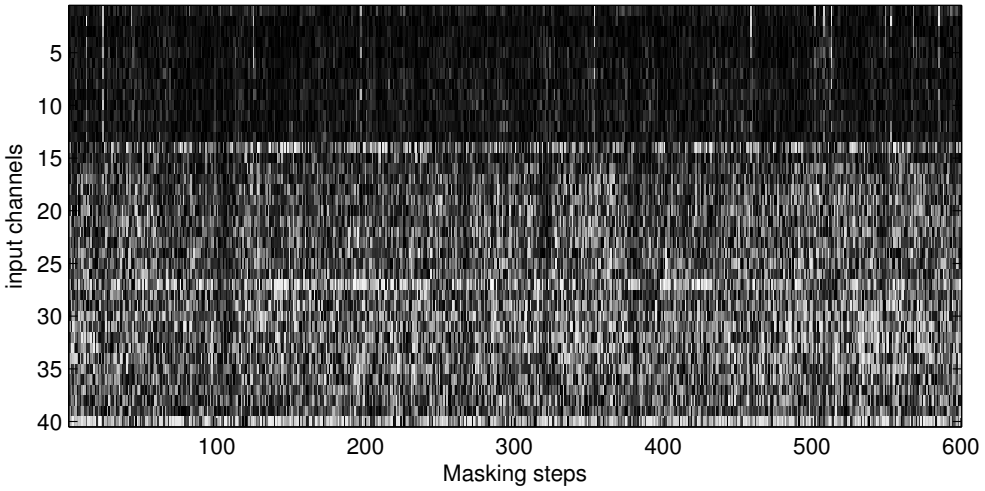


Figure 4: Depiction of the input mask trained on the TIMIT task. We have shown the input weights of the 600 masking steps (horizontal axis) for each channel (vertical axis). For the sake of visualization we have here depicted the natural logarithm of the absolute value of the mask plus 0.1. This enhances the difference in scaling for the different channels.

	MNIST test error	TIMIT frame error rate
Experimental data	1.16%	33.2%
Simulated data	1.08%	31.7%
Simulated data: time-shuffled	1.41%	32.8%
Simulated data: random	6.72%	40.5%
Best in literature	0.23% (Cireşan et al., 2012)	25.0% (Cheng et al., 2009)

Table 1: Benchmark performances for different experimental setups.

was optimized (which is the RC approach).
Results are presented in the middle column of Table 1. The difference between experimental and simulation results is very small. The time shuffled features do indeed cause a notable increase in the classification error rate, indicating that the optimized input masks actively make use of the internal dynamics of the system, and not just offer a generically good feature set.
For the sake of comparison we have added the current state-of-the-art result on MNIST. For a comprehensive overview of results on MNIST please consult `http://yann.lecun.com/exdb/mnist/`. Our result are comparable to the best results obtained using neural networks with a single hidden layer (denoted as a 2-layer NN on the previously mentioned website).

3.2 TIMIT

We applied frame-wise phoneme recognition to the TIMIT data set (Garofolo et al., 1993). The data was pre-processed to 39-dimensional feature vectors using Mel Frequency Cepstral Coefficients (MFCCs). The data consists of the log energy, and the first 12 MFCC coefficients, enhanced with their first and second derivative (the so-called delta and delta-delta features). The phonemes were clustered into a set of 39 classes, as is the common approach. Note that we did not include the full processing pipeline to include segmentation of the labels and arrive at a phoneme error rate. Here, we wish to illustrate the potential of our approach and demonstrate how realizations of physical computers can be extended to further concepts, rather than to claim state-of-the-art performance. Given that, in addition, the input masks are trained to perform frame-wise phoneme classification, including the whole processing pipeline would not be informative.

Input masks are trained using $50,000$ training iterations, where for each iteration the gradient was determined on 200 randomly sampled sequences of a length of 50 frames. For training we again used Nesterov momentum, with momentum coefficient 0.9, and a learning rate of 0.2 which linearly decayed to zero over the duration of the training. As we were in a regime far from overfitting, we simply chose the training error for meta-parameter optimization. We have depicted the optimized input mask in Figure 4. Note that the training process strongly rescaled the masking weights for different input channels, putting more emphasis on the delta and delta-delta features (respectively channels 14 to 26 and 27 to 39).

We repeated the four scenarios previously discussed: using optimized masks in simulation and experiment, using time-shuffled masks, and using random masks. The resulting frame error rates are presented in the right column of Table 1. The simulated and experimental data differ by 1.5%, a relatively small difference, indicating that input masks optimized in simulation are useful in practice, even in the presence of unavoidable discrepancies between the used model and the DCMZ. Results for random masks are significantly worse than those with optimized input masks.

Comparison to literature is not straightforward as most publications do not mention frame error rate, but rather the error rate after segmentation. We included the lowest frame error rate mentioned in literature to our knowledge, though it should be stated that other works may have even lower values, even when they are not explicitly mentioned. For an overview of other results on frame error rate please check Keshet et al. (2011).

The decrease in performance when using time-shuffled masks is quite modest, suggesting that in this case, most of the improvement over random masks is due directly from the features themselves, and the precise details of the dynamics of the system are less crucial than was the case in the MNIST task [3]. Although further testing is needed, we suggest two possible reasons for this. First of all, the TIMIT data set we used contained the first and second derivatives of the first thirteen channels, which already provides information on the preceding and future values and acts as an effective time window. Indeed as can be seen from Figure 4, the input features amplify these derivatives. Therefore, a lot of temporal

3. Note that, when the features are shuffled in time over a single masking period, this indirectly also affects the way information is passed on between different masking periods as the communication between specific nodes between masking periods is a combined effect of the low-pass filter and the self connection.

context is already embedded in a single input frame, reducing the need for recurrent connections. Secondly, the lack of control over the way information is mixed over time may still pose an important obstacle to effectively use the recurrence in the system. Currently, input features are trained to perform two tasks at once: provide a good representation of the current input, and at the same time design the features in such a way that they can make use of the (fixed) dynamics present within the system. It may prove the case that the input masks do not have enough modeling power to fulfill both tasks at once, or that the way temporal mixing occurs in the network cannot be effectively put to use for this particular task.

4. Discussion and Future Work

In this paper we presented an experimental survey of the use of backpropagation through time on a physical delay-coupled electro-optical dynamical system, in order to use it as a machine learning model. We have shown that such a physical setup can be approached as a special case of recurrent neural network, and consequently can be trained with gradient descent using backpropagation. Specifically, we have shown that both the input and output encodings (input and output *masks*) for such a system can be fully optimized in this way, and that the encodings can be successfully applied to the real physical setup.

Previous research in the usage of electro-optical dynamical systems for signal processing used random input encodings, which are quite inefficient in scenarios where the input dimensionality is high. We focused on two tasks with a relatively high input dimensionality: the MNIST written digit recognition data set and the TIMIT phoneme recognition data set. We showed that in both cases, optimizing the input encoding provides a significant performance boost over random masks. We also showed that the input encoding for the MNIST data set seems to directly utilize the inherent dynamics of the system, and hence does more than simply provide a useful feature set.

Note that the comparison with Reservoir Computing is based on the constraints by a given physical setup and a given set of resources. We note that the Reservoir Computing setup could give good results on the proposed tasks too, if we were greatly scaling up its effective dimensionality. This has been evidenced in, for example, Triefenbach et al. (2010), where good results on the TIMIT data set were achieved by using Echo State Networks (a particular kind of Reservoir Computing) of up to 20,000 nodes. In our setup this would be achieved by increasing the number of masking steps N_m within one masking period. In reality, however, we will face two practical problems. First of all, there are bandwidth limitations in signal generation and measurement. Parts of the signal that fluctuate rapidly would be lost when reducing the duration of a single masking step. If one would scale up by keeping the length of the mask steps fixed but use a longer physical delay, for instance a fiber of tens or hundreds of kilometers, the potential gain in performance comes at the cost of one of the systems important advantages: its speed. Also it is hard to foresee how other optical effects in such long fibers such as dispersion and attenuation, would affect performance. This would be an interesting research topic for future investigations.

At the current stage we did not quantify how much the results in this paper hinge on the ability to model the system mathematically. This particular system can be modeled rather precisely, but it is unclear how fast the usefulness of the presented approach would degrade

when the model becomes less acute.
Several directions for future improvements are apparent. The most obvious one is that we could greatly simplify the training process by putting the DCMZ measurements directly in the training loop: instead of optimizing input masks in simulations, we could just as well directly use real, measured data. The Jacobians required for the backpropagation phase can be computed from the measured data. A training iteration would then consist of the following steps: sample data, produce the corresponding input to the DCMZ with the current input mask, measure the output, perform backpropagation in simulation, and update the parameters. The benefit would be that we directly end up with functional input and output masks, without the need for retraining. On top of that, data collection would be much faster. The only additional requirement for this setup would be the need for a single computer controlling both signal generation and signal measurement.
The next direction for improvement would be to rethink the design of the system from a machine learning perspective. The current physical setup on which we applied backpropagation finds its origins in reservoir computing research. As we argue in Section 2, the system can be considered as a special case of recurrent network with a fixed, specific connection matrix between the hidden states at different time steps. In the reservoir computing paradigm, one always uses fixed dynamical systems that remain largely unoptimized, such that in the past this fact was not particularly restrictive. However, given the possibility of fully optimizing the system that was demonstrated in this paper, the question on how to redesign this system such that we can assert more control over the recurrent connection matrix, and hence the dynamics of the system itself, becomes far more relevant. Currently we have a fixed dynamical system of which we optimize the input signal to accommodate a certain signal processing task. As explained at the end of Section 3.2, it appears that backpropagation can currently only leverage the recurrence of the system to a limited degree, when using a single delay loop. Therefore it would be more desirable to optimize both the input signal and the internal dynamics of the system to accommodate a certain task. Alternatively, the configuration can be easily extended to multiple delay loops, allowing for a richer recurrent connectivity.
The most significant result of this paper is that we have shown experimentally that the backpropagation algorithm, a highly abstract machine learning algorithm, can be used as a tool in designing analog hardware to perform signal processing. This means that we may be able to vastly broaden the scope of research into physical and analog realizations of neural architectures. In the end this may result in systems that combine the best of both worlds: powerful processing capabilities at a tremendous speed and with a very low power consumption.

Acknowledgements

P.B., M.H. and J.D. acknowledge support by the interuniversity attraction pole (IAP) Photonics@be of the Belgian Science Policy Office, the ERC NaResCo Starting grant and the European Union Seventh Framework Programme under grant agreement no. 604102 (Human Brain Project). M.C.S. and I.F. acknowledge support by MINECO (Spain), Comunitat Autònoma de les Illes Balears, FEDER, and the European Commission under Projects TEC2012-36335 (TRIPHOP), and Grups Competitius. M.H. and I.F. acknowledge support

from the Universitat de les Illes Balears for an Invited Young Researcher Grant. In addition, we acknowledge Prof. L. Larger for developing the optoelectronic delay setup.

References

Lennert Appeltant, Guy Van der Sande, Jan Danckaert, and Ingo Fischer. Constructing optimized binary masks for reservoir computing with delay systems. *Scientific Reports*, 4:3629, 2014.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 8624–8628. IEEE, 2013.

Daniel Brunner, Miguel C Soriano, Claudio R Mirasso, and Ingo Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications*, 4:1364, 2013.

Ken Caluwaerts, Michiel D'Haene, David Verstraeten, and Benjamin Schrauwen. Locomotion without a brain: physical reservoir computing in tensegrity structures. *Neural Computation*, 19(1):35–66, 2013.

Chih-Chieh Cheng, Fei Sha, and Lawrence Saul. A fast online algorithm for large margin training of online continuous density hidden markov models. In *Interspeech 2009*, pages 668–671, 2009.

Dan Cireşan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep, big, simple neural nets for handwritten digit recognition. *Neural Computation*, 22(12):3207–3220, 2010.

Dan Cireşan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.

Chrisantha Fernando and Sampsa Sojakka. Pattern recognition in a bucket. In *Proceedings of the 7th European Conference on Artificial Life*, pages 588–597, 2003.

John Garofolo, National Institute of Standards, Technology (US, Linguistic Data Consortium, Information Science, Technology Office, United States, and Defense Advanced Research Projects Agency). *TIMIT Acoustic-phonetic Continuous Speech Corpus.* Linguistic Data Consortium, 1993.

Helmut Hauser, Auke J Ijspeert, Rudolf M Füchslin, Rolf Pfeifer, and Wolfgang Maass. Towards a theoretical foundation for morphological computation with compliant bodies. *Optics Express*, 105(5-6):355–370, 2011.

Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 190–198, 2013.

Michiel Hermans, Joni Dambre, and Peter Bienstman. Optoelectronic systems trained with backpropagation through time. *IEEE Transactions in Neural Networks and Learning Systems*, 2014a. in press.

Michiel Hermans, Benjamin Schrauwen, Peter Bienstman, and Joni Dambre. Automated design of complex dynamic systems. *PloS One*, 9(1):e86696, 2014b.

Herbert Jaeger. Short term memory in echo state networks. Technical Report GMD Report 152, German National Research Center for Information Technology, 2001.

Herbert Jaeger and Harald Haas. Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless telecommunication. *Science*, 308:78–80, April 2 2004.

Joseph Keshet, David McAllester, and Tamir Hazan. Pac-bayesian approach for minimization of phoneme error rate. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2224–2227, 2011.

Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

Laurent Larger, Jean-Pierre Goedgebuer, and Vladimir Udaltsov. Ikeda-based nonlinear delayed dynamics for application to secure optical transmission systems using chaos. *Comptes Rendus Physique*, 5(6):669–681, 2004.

Laurent Larger, Miguel C Soriano, Daniel Brunner, Lennert Appeltant, Jose M Gutiérrez, Luis Pesquera, Claudio R Mirasso, and Ingo Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Optics Express*, 3:20, 2012.

Mantas Lukosevicius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

Wolfgang Maass, Thomas Natschläger, and Henri Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, 2002.

James Martens and Ilya Sutskever. Learning recurrent neural networks with hessian-free optimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1033–1040, 2011.

Yvan Paquot, Francois Duport, Antoneo Smerieri, Joni Dambre, Benjamin Schrauwen, Marc Haelterman, and Serge Massar. Optoelectronic reservoir computing. *Scientific Reports*, 2:1–6, 2012.

Ali Rodan and Peter Tino. Minimum complexity echo state network. *Neural Networks, IEEE Transactions on*, 22(1):131–144, 2011.

Jason Tyler Rolfe and Yann LeCun. Discriminative recurrent sparse auto-encoders. In *International Conference on Learning Representations (ICLR)*, 2013.

David Rumelhart, Geoffrey Hinton, and Ronald Williams. *Learning internal representations by error propagation.* MIT Press, Cambridge, MA, 1986.

Miguel C Soriano, Silvia Ortín, Daniel Brunner, Laurent Larger, Claudio Mirasso, Ingo Fischer, and Luís Pesquera. Opto-electronic reservoir computing: tackling noise-induced performance degradation. *Optics Express*, 21(1):12–20, 2013.

Jochen Steil. Backpropagation-Decorrelation: Online recurrent learning with O(N) complexity. In *Proceedings of the International Joint Conference on Neural Networks*, volume 1, pages 843–848, 2004.

Ilya Sutskever, James Martens, and Geoffrey Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1017–1024, 2011.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 1139–1147, 2013.

Fabian Triefenbach, Azaraksh Jalalvand, Benjamin Schrauwen, and Jean-Pierre Martens. Phoneme recognition with large hierarchical reservoirs. In *Advances in Neural Information Processing Systems 23*, pages 2307–2315, 2010.

Kristof Vandoorne, Wouter Dierckx, Benjamin Schrauwen, David Verstraeten, Roel Baets, Peter Bienstman, and Jan van Campenhout. Toward optical signal processing using photonic reservoir computing. *Optics Express*, 16(15):11182–11192, 2008.

Kristof Vandoorne, Pauline Mechet, Thomas Van Vaerenbergh, Martin Fiers, Geert Morthier, David Verstraeten, Benjamin Schrauwen, Joni Dambre, and Peter Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Artificial Life*, 5, 2014.

Lionel Weicker, Thomas Erneux, Otti DHuys, Jan Danckaert, Maxime Jacquot, Yanne Chembo, and Laurent Larger. Strongly asymmetric square waves in a time-delayed system. *Physical Review E*, 86(5):055201, 2012.

Journal of Machine Learning Research 16 (2015) 2099-2145 Submitted 10/12; Published 12/15

On Linearly Constrained Minimum Variance Beamforming

Jian Zhang JZ79@KENT.AC.UK
Chao Liu CL304@KENT.AC.UK
School of Mathematics, Statistics and Actuarial Science
University of Kent
Canterbury, Kent CT2 7NF, UK

Editor: Xiaotong Shen

Abstract

Beamforming is a widely used technique for source localization in signal processing and neuroimaging. A number of vector-beamformers have been introduced to localize neuronal activity by using magnetoencephalography (MEG) data in the literature. However, the existing theoretical analyses on these beamformers have been limited to simple cases, where no more than two sources are allowed in the associated model and the theoretical sensor covariance is also assumed known. The information about the effects of the MEG spatial and temporal dimensions on the consistency of vector-beamforming is incomplete. In the present study, we consider a class of vector-beamformers defined by thresholding the sensor covariance matrix, which include the standard vector-beamformer as a special case. A general asymptotic theory is developed for these vector-beamformers, which shows the extent of effects to which the MEG spatial and temporal dimensions on estimating the neuronal activity index. The performances of the proposed beamformers are assessed by simulation studies. Superior performances of the proposed beamformers are obtained when the signal-to-noise ratio is low. We apply the proposed procedure to real MEG data sets derived from five sessions of a human face-perception experiment, finding several highly active areas in the brain. A good agreement between these findings and the known neurophysiology of the MEG response to human face perception is shown.

Keywords: MEG neuroimaging, vector-beamforming, sparse covariance estimation, source localization and reconstruction

1. Introduction

MEG is a non-invasive imaging technique that records brain activity with high temporal resolution. Postsynaptic current flow within the dendrites of active neurons generates a magnetic field that can be measured close to the scalp surface by use of sensors (Hämäläinen et al., 1993). The magnitude of these measured fields is directly related to neuronal current strength, and hence their measurement will reflect the amplitude of brain activity. The major challenge, however, is to localize active regions inside the head, given the measured magnetic fields outside the head (i.e., given MEG data). This is an ill-posed problem of source localization since the magnetic fields could be caused by an infinite number of neuronal regions. Mathematically, the problem can be stated as follows: one observes a vector of time-series $\mathbf{Y}(t) = (Y_1(t), ..., Y_n(t))^T \in \mathbb{R}^n$, $t = t_j, 1 \leq j \leq J$ from n sensors,

which are linked to candidate sources located at $r_k, 1 \leq k \leq p$ in the brain via the model

$$\mathbf{Y}(t) = \sum_{k=1}^{p} H_k \mathbf{m}_k(t) + \varepsilon(t), \tag{1.1}$$

where H_k is an $n \times 3$ lead field matrix at r_k (i.e., the unit output of the candidate source at location r_k, which is derived from Maxwell's equations), $\mathbf{m}_k(t)$ with covariance matrix Σ_k is a 3×1 moment (time-course) at time t and location r_k, $\varepsilon(t)$ with covariance matrix $\sigma_0^2 I_n$ represents white noises at the MEG channels, and I_n is the $n \times n$ identity matrix. See Mosher et al. (1999) for more details. In practice, when candidate source locations (i.e., voxels) are created by discretizing the source space in the brain, the number of these sources can be substantially larger than the number of available sensors. Moreover, unlike the traditional functional data, not only source time courses but also sensor readings are spatially correlated. Therefore, searching for a small set of latent sources of non-null powers from a large number of candidates poses a challenge to standard i.i.d. sample-based methods in functional data analysis (Ramsay, 2006). Here, the source power at location r_k is referred as the trace of the covariance matrix Σ_k.

Two types of approaches have been proposed for handling the above problem in the literature: global approach and local approach (e.g., Henson et al., 2011; Bolstad et al., 2009; Van Veen et al., 1997; Robinson, 1999; Huang et al., 2004; Quraan et al., 2011). In the global approach, one puts all candidate sources into the model and solves a sparse estimation problem. In the local approach, on other hand, one invokes a list of local models, each is tailored to a particular candidate region. The global approach often requires to specify parametric models, while the local approach is model-free. When the number of candidate sources p is small or moderate compared to the number of available sensors n, one may use a Bayesian method to infer latent sources, with helps of computationally intensive algorithms (e.g., Henson et al., 2011). To make an accurate inference, a large p should be chosen. However, when p is large, the global approach may be computationally intractable and the local approach is preferred. Here, we focus on the so-called linearly constrained minimum variance (LCMV) beamforming (also called vector-beamforming), a local method for solving the above large-p-small-n problem. It involves two steps as follows:

- **Projection step.** For location r_k in the source space, one searches for the optimal $n \times 3$ weighting-matrix W by minimizing the trace of the sample covariance of the projected data $W^T Y(t_j)$, $1 \leq j \leq J$, subject to $W^T H_k = I_3$, where I_3 is a 3×3 identity matrix. This gives the optimal trace

$$\hat{S}_k = \mathrm{tr}([H_k^T \hat{C}^{-1} H_k]^{-1}), \tag{1.2}$$

  where $\hat{C}$ is a sensor covariance estimator and for any invertible matrix A, A^{-1} denotes its inverse, and $\mathrm{tr}(\cdot)$ stands for the matrix trace operator. See Van Veen et al. (1997) for the details.

- **Mapping step.** For each location r_k, calculate the neuronal activity index $\hat{S}_k/(\sigma_0^2 \mathrm{tr}([H_k^T H_k]^{-1}))$, where σ_0^2 is estimated by certain baseline noise data such as the pre-stimulus data. Plot the index against the grid points, creating a neuronal activity map over a given temporal window.

In the **projection step**, the procedure aims at estimating the desired signal from each chosen location while minimizing the contributions of other unknown locations in the presence of noises by optimizing the variation of the projected data. This can be easily seen from the following decomposition of the projected covariance under the constrain $W^T H_k = I_3$:

$$\begin{aligned}\mathrm{tr}\left(\mathrm{cov}(W^T\mathbf{Y}(t))\right) &= \mathrm{tr}(\Sigma_k) + \mathrm{tr}(W^T\mathrm{cov}(\sum_{j\neq k} H_j m_j(t) + \varepsilon(t))W) \\ &\quad +2\mathrm{tr}(\mathrm{cov}(m_k(t), W^T(\sum_{j\neq k} H_j m_j(t) + \varepsilon(t)))),\end{aligned}$$

where the first term is the underlying signal strength at r_k and the last two terms are the contributions of other locations and background noises to the estimated strength of the signal at r_k. Therefore, minimizing the trace of the projected covariance of the data with respect to W is equivalent to minimizing the the contributions of other locations and background noises to estimating the true signal strength at r_k. The further mathematical details can be found in Sekihara and Nagarajan (2008). As pointed out before, in practice, we often have the baseline noise data. Performing the above projection procedure on the noise data under the assumption that the noise covariance matrix is approximately $\sigma_0^2 I_n$, we obtain the optimal trace of the covariance matrix of the projected noise at r_k, $\sigma_0^2\mathrm{tr}([H_k^T H_k]^{-1})$. This implies that the above neuronal activity index is a signal-to-noise ratio (SNR) at location r_k. Therefore, the map generated in the **mapping step** is a SNR map. A similar formula can be derived under a general model of the noise covariance. However, to avoid high-dimensional effects on estimating sensor covariance matrices, we often employed a diagonal noise covariance model even when the true one is not diagonal.

Both theoretical and empirical studies have suggested that the vector-beamforming can provide excellent performance given a sufficient number of observations (e.g., Sekihara et al., 2004; Brookes et al., 2008; Quraan et al., 2011). However, the existing theoretical analyses have been limited to simple cases, where no more than two sources are allowed in the model and the theoretical sensor covariance is assumed known. In limited data scenarios the estimated sensor covariance may possess considerable variation and thus deteriorate the performance of localization. Empirical studies have also demonstrated that the sampling window and rate are generally required to increase as the number of spatial sensors increases. For example, when using the sample covariance matrix to estimate the sensor covariance matrix, the number of statistically independent data records should be three or more times the number of sensors in order to obtain statistically stable source location estimates (e.g., Rodríguez-Rivera et al., 2006). Consequently, the potential advantages of having a large number of sensors are offset by the requirement for increased sampling window and rate. Therefore, it is important to develop a general framework for users to examine the extent of effects to which the spatial dimension (i.e., the lead field matrix) and the temporal dimension (i.e., the temporal correlations of sensor measurements) of MEG on the accuracy of source localization. Furthermore, most brain activities are conducted by neural networks which consist of multiple sources. For example, in the so-called evoked median-nerve MEG response study, scientists have found the relatively large number of neuronal sources activated in a relatively short period of time by the median-nerve stimulation with typical repetition rates, which challenges covariance-based analysis techniques such as beamformer due to source cancellations (Huang et al., 2004). We need to understand how the accuracy

of localization is affected by source cancellations both theoretically and empirically. In particular, we need to address the fundamental questions of whether the neuronal activity map can reveal the true sources when the number of sensors and the width of the sampling window are large enough and of how much multiple source cancellation effects are reduced by increasing spatial and temporal dimensions of MEG.

The goal of the present study is to demonstrate at both theoretical and empirical levels the behavior of a class of vector-beamforming techniques which includes the standard vector-beamformer as a special example. These beamformers are based on thresholding the sample sensor covariance matrix. By thresholding, we aim at reducing the noise level in the sample sensor covariance. We provide an asymptotic theory on these beamformers when the sensor covariance matrix is consistently estimated and when multiple sources exist. We show that the estimated source power is consistent when multiple sources are asymptotically separable in terms of a lead field distance. We further assess the performance of the proposed procedure by both simulations and real data analyses.

The paper is organized as follows. The details of the proposed procedures are given in Section 2. The asymptotic analysis is provided in Section 3. Other covariance estimator-based beamformers are introduced in Section 4. The simulation studies on these beamformers and an application to face-perception data are conducted in Section 5. The discussion and conclusion are made in Section 6. The proofs of the theorems and corollaries are deferred to Section 7. Throughout the paper, let $||A||$ denote the operator norm of matrix A. For a sequence of matrix A_n, we mean by $A_n = O(1)$ that $||A_n||$ is bounded and by $A_n = o(1)$ that $||A_n|| = o(1)$. Similarly, we define the notations O_p and o_p for a sequence of random matrices A_n. For non-negative matrices A and B, we say $A < B$ if $a^T Aa < a^T Ba$ for any a with $||a|| = 1$. We say that random matrix A_n is asymptotically larger than random matrix B_n in probability if $\min_{||a||=1} a^T(A_n - B_n)a$ is asymptotically bounded below from zero in probability.

2. Methodology

Suppose that the sensor measurements $(\mathbf{Y}(t_j) : 1 \le j \le J)$ are weakly stationary time-courses observed from n sensors. We want to identify a small set of non-null sources that underpin these observations. To this end, we introduce a family of vector-beamformers based on thresholding sensor covariance as follows.

2.1 Thresholding the sensor covariance matrix

The sensor covariance matrix of $\mathbf{Y}(t)$, C can be estimated by the sample covariance matrix

$$\hat{C} = (\hat{c}_{ij}) = \frac{1}{J}\sum_{j=1}^{J} \mathbf{Y}(t_j)\mathbf{Y}(t_j)^T - \bar{\mathbf{Y}}\bar{\mathbf{Y}}^T,$$

where $\bar{\mathbf{Y}}$ is the sample mean of $(\mathbf{Y}(t_j) : 1 \le j \le J)$. It is well-known that the sample covariance estimator can breakdown when the dimension n is large (Bickel and Levina, 2008). In the statistical literature (Bickel and Levina, 2008), various sparse estimation procedures have been proposed to fix the sample covariance, including the following thresholded

estimator:

$$\hat{C}(\tau_{nJ}) = (\hat{c}_{ij}(\tau_{nJ}))$$

with $\hat{c}_{ij}(\tau_{nJ}) = \hat{c}_{ij} I(|\hat{c}_{ij}| \geq \tau_{nJ})$, where τ_{nJ} is a varying constant in n and J.

As with the i.i.d. case (Bickel and Levina, 2008), the above thresholded estimator will be shown to converges to positive definite limit with probability tending to 1 in the Lemma 7.2 in Section 7 below. Although the thresholded estimator has good theoretical properties, it may not be always positive definite when the sample size is finite or when sensors are spatially too close to each other. To tackle the issue, we assume that $\hat{C}(\tau_{nJ})$ has the eigen-decomposition $\hat{C}(\tau_{nJ}) = \sum_{k=1}^{n} \hat{\lambda}_k v_k^T v_k$ and then a positive semidefinite estimator can be obtained by setting these negative eigenvalues to zeros. We further shrinkage the covariance matrix estimator by artificially adding $\epsilon_0 I_n$ to it in our implementation, where we choose ϵ_0 to be a tuning constant which is equal to or slightly larger than the maximum eigenvalue of the noise covariance matrix. We will show in the following sections that adding $\epsilon_0 I_n$ to the thresholded covariance matrix does not affect the consistency of the neuronal activity index.

2.2 Beamforming

As before, let Σ_k denote the covariance matrix of the moment $\mathbf{m}_k(t)$ at the location r_k. Based on the thresholded sensor covariance estimator $\hat{C}(\tau_{nJ})$, we estimate $\Sigma_k, 1 \leq k \leq p$ and create a neuronal activity map in the following two steps.

In the projection step, for $1 \leq k \leq p$, we search for an $n \times 3$ weight matrix $\hat{W}_k$ which attains the minimum trace of $W^T \hat{C}(\tau_{nJ}) W$ subject to $W^T H_k = I_3$. When $\hat{C}(\tau_{nJ})$ is invertible, it follows from Van Veen et al. (1997) that

$$\hat{W}_k = \hat{C}(\tau_{nJ})^{-1} H_k \left[H_k^T \hat{C}(\tau_{nJ})^{-1} H_k \right]^{-1}$$

with the resulting moment covariance matrix and trace estimators

$$\hat{\Sigma}_k = \left[H_k^T \hat{C}^{-1}(\tau_{nJ}) H_k \right]^{-1}, \quad \hat{S}_k = \mathrm{tr} \left\{ \left[H_k^T \hat{C}(\tau_{nJ})^{-1} H_k \right]^{-1} \right\}$$

respectively. In the mapping step, we calculate the so-called neuronal activity index

$$\mathrm{NAI}(r_k) = \hat{S}_k / \Big(\sigma_0^2 \mathrm{tr} \left(\left[H_k^T H_k \right]^{-1} \right) \Big),$$

creating a brain activity map, where σ_0^2 is estimated from baseline data (i.e., called pre-stimulus data in the next subsection). One of the underlying sources can be then estimated by the global peak on the map with the associated latent time-course estimated by projecting the data along the optimal weighting vector. The multiple sources can also be identified by grouping the local peaks on the transverse slices of the brain.

2.3 Choosing the thresholding level

In practice, the MEG imaging is often run on a subject first without stimulus and then with stimulus. This allows us to calculate the sample covariance $\hat{C}$ for the stimulus data

as well as the sample covariance $\hat{C}_0$ for the pre-stimulus data. The latter can provide an estimator of the background noise level. In the next section, we will show that the convergence rate of the thresholded sample covariance is $O(\sqrt{\log(n)/J})$. In light of this, we set $\tau_{nJ} = c_0\hat{\sigma}_0^2\sqrt{\log(n)/J}$ with a tuning constant c_0 and threshold $\hat{C}$ by τ_{nJ}, where $\hat{\sigma}_0^2$ is the minimum diagonal element in $\hat{C}_0$ and c_0 is a tuning constant. Note that, when $c_0 = 0$, the proposed procedure reduces to the standard vector-beamformer implemented in the software FieldTrip (Oostenveld et al., 2010). For each value of c_0, we apply the proposed procedure to the data and calculate the maximum neuronal activity index

$$\text{NAI}_{c_0} = \max\{\text{NAI}(r) : r \text{ is running over the grid}\}. \tag{2.3}$$

In simulations, we will show that $c_0 \in D_0 = \{0, 0.5, 1, 1.5, 2\}$ has covered its useful range. Our simulations also suggests that there is an optimal value of c_0, which depends on several factors including the strengths of signals and source interferences. To exploit these two factors, we choose c_0 in which NAI_{c_0} attains maximum or minimum, resulting in two procedures called **ma** and **mi** respectively. By choosing c_0, the procedure **ma** intends to increase the maximum SNR value, while the procedure **mi** tries to reduce source interferences. The simulation studies in Section 5 suggest that **mi** can perform better than **ma** when sources are correlated.

2.4 Two sets of stimuli

Suppose now that MEG measurements $(\mathbf{Y}^{(1)}(t))$ and $(\mathbf{Y}^{(2)}(t))$ are made under two different sets of stimuli and pre-stimuli with the associated neuronal activity indices denoted by $\text{NAI}^{(1)}(r_k)$ and $\text{NAI}^{(2)}(r_k)$ respectively. The previous strategy for selecting the tuning constant c_0 can be adopted here when we calculate these indices. To identify source locations that respond to the change of stimulus set, we calculate a log-contrast $\log(\text{NAI}^{(1)}(r_k)/\text{NAI}^{(2)}(r_k))$ between the two sets of stimuli at location r_k, $1 \le k \le p$, creating a log-contrast map. The resulting log-contrast map is equivalent to the map based on index ratio $\text{NAI}^{(1)}(r_k)/\text{NAI}^{(2)}(r_k)$, which was often seen in the literature (e.g., Hillebrand et al., 2005). We further take the global peak of the log-contrast as the maximum location estimator for a source location that contributes to the difference between the two sets of MEG measurements.

3. Theory

In this section, we develop a theory on the consistency as well as the convergence rate of the hard thresholding-based beamformer estimator under regularity conditions. In particular, we show that the consistency holds true under regularity conditions if we let the hard threshold $\tau_{nJ} = A\sqrt{\log(n)/J}$ with constant A. This provides a theoretical basis for using the proposed procedures **ma** and **mi**.

Without loss of generality, we assume that the first $q \le p$ moment vectors are of non-zero covariance matrices $\Sigma_k, 1 \le k \le q$, where q is unknown and often much smaller than p in practice. For the simplicity of mathematical derivations, we also assume that Σ_k does

not grow with the number of sensors n. Our task is to identify the unknown true model

$$\mathbf{Y}(t) = \sum_{k=1}^{q} H_k \mathbf{m}_k(t) + \varepsilon(t), \tag{3.4}$$

from the working model (1.1) by using the proposed procedure, where the unknown moments $\mathbf{m}_k(t), 1 \le k \le q$ are of non-zero powers $\mathrm{tr}(\Sigma_k), 1 \le k \le q$. To establish a theory for the proposed procedures, we assume that

(A1): Both the moment vectors $(\mathbf{m}_k(t) : 1 \le k \le q)$ and the white noise process $(\varepsilon(t))$ are stationary with zero means and temporally uncorrelated with each other. Also, $\mathbf{m}_k(t)$ is temporally uncorrelated with $\mathbf{m}_j(t)$ for $k \ne j$.

Under Condition (A1), the sensor covariance matrix of $\mathbf{Y}(t)$, C can be expressed in the form

$$C = \sum_{k=1}^{q} H_k \Sigma_k H_k^T + \sigma_0^2 I_n.$$

As pointed out by Sekihara and Nagarajan (2008, Chapter 9), Condition (A1) is one of fundamental assumptions in the vector-beamforming. However, source activities in the brain are inevitably correlated to some degree, and in strict sense, (A1) cannot be satisfied. The theoretical influence of temporally correlated sources has been investigated by Sekihara and Nagarajan (2008, Chapter 9). The equation (9.3) in Sekihara and Nagarajan (2008, Chapter 9) implies that the influence can be ignored if the partial correlations between sources are close to zeros in order of $o(1/n)$ when the number of sensors n is sufficiently large. Note that although in practice the number of sensors is limited to a few hundreds, we still ideally let n tend to infinity to identify potential spatial factors that affect the performance of a vector-beamformer. In the next section, by using simulations, we will demonstrate that the source correlations can mask some true sources.

To show the consistency of the estimators $\hat{\Sigma}_k$ and S_k, we need more notations and condition as follows. Let H_k denote the lead field matrix at the location r_k. For the simplicity of the technical derivations later, we further assume that the lead field matrices satisfy the condition that for any location r_k, $H_k^T H_k/n \to G$ in terms of the operator norm as n tends infinity, where G is a 3×3 positive definite matrix.

Under the above condition, we can find a positive definite matrix Q_k satisfying that $H_k^T H_k = n Q_k Q_k^T$ and $Q_k^{-1} H_k^T H_k Q_k^{-T} = n I_3$ when n is large enough, where I_3 is an identity matrix. Letting $H_k^* = H_k Q_k^{-T}$, $\mathbf{m}_k^*(t) = Q_k^T \mathbf{m}_k$ and $\Sigma_k^* = Q_k^T \Sigma_k Q_k$, we reparametrize the model (1.1) as follows:

$$\mathbf{Y}(t) = \sum_{k=1}^{p} H_k^* \mathbf{m}_k^* + \varepsilon(t)$$

with the covariance matrix $C = \sum_{k=1}^{p} H_k^* \Sigma_k^* H_k^{*T} + \sigma_0^2 I_n$. Then, under the reparametrized model, the estimators

$$\begin{aligned} \hat{\Sigma}_k^* &= \left[H_k^{*T} \hat{C}(\tau_{nJ})^{-1} H_k^* \right]^{-1} = \left[Q_k^{-1} H_k^T \hat{C}(\tau_{nJ})^{-1} H_k Q_k^{-T} \right]^{-1} = Q_k^T \hat{\Sigma}_k Q_k. \\ \hat{S}_k &= \mathrm{tr}(Q_k^{-T} \hat{\Sigma}_k^* Q_k^{-1}). \end{aligned}$$

Consequently, $\hat{\Sigma}_k^*$ is consistent with Σ_k^* if and only if $\hat{\Sigma}_k$ is consistent with Σ_k. Therefore, without loss of generality, hereinafter we assume that

(A2): $H_k^T H_k = nI_3$ for any location r_k.

We process the remaining analysis in two stages: In the first stage, we develop an asymptotic theory for the proposed vector-beamformers when the sensor covariance matrix C is known. The sensor covariance matrix can be assumed known if the width of the sampling window can be arbitrarily large. In the second stage, we extend the theory to the case where C is estimated by $\hat{C}(\tau_{nJ})$.

3.1 Beamformer analysis when C is known

We begin with introducing some more notations. For any locations r_x and r_y, let H_x and H_y denote their lead field matrices. Define the lead field coherent matrix by $\rho_{xy} = \rho(r_x, r_y) = H_x^T H_y/n$. Note that $\rho_{xy} + \rho_{yx} = I_3 - (H_x - H_y)^T(H_x - H_y)/(2n)$. Therefore, $I_3 - (\rho_{xy} + \rho_{yx})$ indicates how close r_x is to r_y. In general, the partial coherence factor matrices (or called partial correlation matrices) $a_{yx|k}$, $1 \le k \le q$ are defined iteratively by the so-called sweep operation (Goodnight, 1979) as follows:

$$\begin{aligned} a_{yx|1} &= \sigma_0^{-2}\rho(r_y, r_1, r_x) = \sigma_0^{-2}\left(\rho(r_y, r_x) - \rho(r_y, r_1)\rho(r_1, r_x)\right), \\ a_{yx|(k+1)} &= a_{yx|k} - a_{y(k+1)|k}\left[a_{(k+1)(k+1)|k}\right]^{-1} a_{(k+1)x|k}, \quad 1 \le k \le q-1. \end{aligned}$$

For example, we have

$$\begin{aligned} \sigma_0^2 a_{yx|1} &= \rho_{yx} - \rho_{y1}\rho_{1x}, \quad \sigma_0^2 a_{22|1} = I_3 - \rho_{12}^T\rho_{12}, \\ \sigma_0^2 a_{33|2} &= I_3 - \rho_{13}^T\rho_{13} - \left(\rho_{23} - \rho_{12}^T\rho_{13}\right)^T\left[I_3 - \rho_{12}^T\rho_{12}\right]^{-1}\left(\rho_{23} - \rho_{12}^T\rho_{13}\right). \end{aligned}$$

Note that $\sigma_0^2 a_{(k+1)(k+1)|k}$ gauges the partial variability of r_{k+1} given the previous $r_k's$ while $\sigma_0^2 a_{yx|(k+1)}$ shows the partial coherence between r_x and r_y given $\{r_1, ..., r_{k+1}\}$. We expect that $a_{yx|(k+1)}$ will be small if r_y and r_x are spatially far away from each other. We define $b_{yx|k}$, $1 \le k \le q$, by letting $b_{yx|1} = \rho_{y1}\Sigma_1^{-1}\rho_{1x}$ and

$$\begin{aligned} b_{yx|k} &= b_{yx|(k-1)} - b_{yk|(k-1)}\left[a_{kk|(k-1)}\right]^{-1} a_{kx|(k-1)} - a_{yk|(k-1)}\left[a_{kk|(k-1)}\right]^{-1} b_{kx|(k-1)} \\ &\quad + a_{yk|(k-1)}\left[a_{kk|(k-1)}\right]^{-1}\left\{\Sigma_k^{-1} + b_{kk|(k-1)}\right\}\left[a_{kk|(k-1)}\right]^{-1} a_{kx|(k-1)}. \end{aligned}$$

We also define $c_{jj|k}$, $1 \le j \le k \le q$ by

$$c_{jj|k} = \begin{cases} -\Sigma_k^{-1}\left[a_{kk|(k-1)}\right]^{-1}\Sigma_k^{-1}, & j = k \\ c_{jj|(k-1)} - b_{jk|(k-1)}\left[a_{kk|(k-1)}\right]^{-1} b_{jk|(k-1)}^T, & 1 \le j \le k-1. \end{cases}$$

Let $a_{nq} = n\min_{1\le k\le q-1}||a_{(k+1)(k+1)|k}||$, and let $k_m = 0$ if $a_{nq} \to \infty$ and $k_m = \min\{1 \le k \le q-1 : n||a_{(k+1)(k+1)|k}|| = O(1)\}$ if $a_{nq} = O(1)$. Let $d_{x|q} = \max_{2\le k\le q}||a_{kx|(k-1)}a_{kk|(k-1)}^{-1}||$, which measures the maximum absolute partial correlation among q sources by using their lead field matrix. As the lead field matrix measures the unit outputs of sources recorded by sensors, the maximum absolute partial correlation may increase when the number of sensors n increases. In the following theorem, for any location r_x of interest, the condition that

$d_{x|q} = O(1)$ (i.e., the maximum absolute partial correlation will be bounded) is imposed on the lead field matrix. The condition is used to ensure the coherence stability for the grid approximation to the lead field. Our numerical experience suggests that the condition roughly holds when the underlying sources are asymptotically not close to each other. See the discussion in Section 7. The following theorem shows when the source covariance estimator is consistent and when it is not.

Theorem 1 *Under Conditions (A1)∼(A2) and C is known, we have:*

(1) If $a_{nq} = O(1)$ and $\max_{1\le k\le q} d_{k|q} = O(1)$, then the estimated source covariance at r_{k_m+1} $\left[{H_{k_m+1}}^T C^{-1} H_{k_m+1}\right]^{-1}$ is asymptotically larger than Σ_{k_m+1}.

(2) If $a_{nq} \to \infty$, then for $1 \le j \le q$, the estimated source covariance at r_j admits

$$\left[H_j^T C^{-1} H_j\right]^{-1} = \Sigma_j + \frac{1}{n}\Sigma_j c_{jj|q}\Sigma_j + O(a_{nq}^{-2}),$$

provided $\max_{1\le k\le q} d_{k|q} = O(1)$, where $||\Sigma_j c_{jj|q}\Sigma_j/n|| = O(a_{nq}^{-1})$ as $n \to \infty$.

(3) If $a_{nq} \to \infty$, then for $r_x \notin \{r_1, ..., r_q\}$, the estimated source covariance at r_x admits

$$\left[H_x^T C^{-1} H_x\right]^{-1} = \frac{1}{n}a_{xx|q}^{-1} - \frac{1}{n^2}a_{xx|q}^{-1} b_{xx|q} a_{xx|q}^{-1} + O(a_{nq}^{-3}),$$

provided $\max_{1\le j\le q} d_{j|q} = O(1)$, $||n a_{xx|q}|| \to \infty$, and $d_{x|q} = O(1)$ as n tends to infinity, where $b_{xx|q} = O(1)$ as $n \to \infty$.

The following lemma shows when the source power estimator is consistent.

Corollary 2 *Under Condition (A1)∼(A2), we have:*

(1) If $a_{nq} = O(1)$ and $\max_{1\le k\le q} d_{k|q} = O(1)$, then the estimated source power at r_{k_m+1} $tr\left(\left[{H_{k_m+1}}^T C^{-1} H_{k_m+1}\right]^{-1}\right)$ is asymptotically larger than $tr(\Sigma_{k_m+1})$.

(2) If $a_{nq} \to \infty$, then for $1 \le j \le q$, the estimated source power at r_j admits

$$tr\left(\left[H_j^T C^{-1} H_j\right]^{-1}\right) = tr(\Sigma_j) + \frac{1}{n}tr(\Sigma_j c_{jj|q}\Sigma_j) + O(a_{nq}^{-2}),$$

provided $\max_{1\le k\le q} d_{k|q} = O(1)$, where $||\Sigma_j c_{jj|q}\Sigma_j/n|| = O(a_{nq}^{-1})$ as $n \to \infty$.

(3) If $a_{nq} \to \infty$, then for $r_x \notin \{r_1, ..., r_q\}$, the estimated source power at r_x admits

$$tr\left(\left[H_x^T C^{-1} H_x\right]^{-1}\right) = \frac{1}{n}tr(a_{xx|q}^{-1}) - \frac{1}{n^2}tr(a_{xx|q}^{-1} b_{xx|q} a_{xx|q}^{-1}) + O(a_{nq}^{-3}),$$

provided $\max_{1\le j\le q} d_{j|q} = O(1)$, $||n a_{xx|q}|| \to \infty$, and $d_{x|q} = O(1)$ as n tends to infinity, where $b_{xx|q} = O(1)$ as $n \to \infty$.

Remark 3 *It follows from the definition that $c_{jj|q}$ is proportional to σ_0^2, which implies the convergence rate of the neuronal activity index is of order $O(\sigma_0^2/(\sigma_0^2 a_{nq}))$, where $\sigma_0^2 a_{nq}$ is independent of σ_0^2. Therefore, the effect of adding $\epsilon_0 I_n$ to C on the above convergence rate is increasing or decreasing the rate by the amount of $O(\epsilon_0/((\sigma_0^2+\epsilon_0)a_{nq}))$. In particular, adding $\epsilon_0 I_n$ to C does not affect the consistency of the neuronal activity index if a_{nq} tends infinity.*

Remark 4 *From the proof in Section 7, we can see that if we relax the coherence stability condition $\max_{1\le k\le q} d_{k|q} = O(1)$ to $\max_{1\le k\le q} d_{k|q} = O(\log(n))$, then the convergence rates in the theorem will be reduced by a factor of $\log(n)$.*

Remark 5 *If there are MEG measurements made under two different sets of stimuli and pre-stimuli, we let $C^{(1)} = \sum_{k=1}^p H_k^T \Sigma_k^{(1)} H_k + \sigma_{01}^2 I_n$ and $C^{(2)} = \sum_{k=1}^p H_k^T \Sigma_k^{(2)} H_k + \sigma_{02}^2 I_n$ be the corresponding sensor covariance matrices. We perform the proposed beamformers on $C^{(1)}$ and $C^{(2)}$ respectively. Then, under certain conditions, Theorem 1 can be extended to this setting. When r_k is a source location for both sets of stimuli, the log-contrast tends to the true one as $n \to \infty$; when r_k is a source for stimulus set 1 but not for stimulus set 2, the log-contrast tends to infinite; when r_k is a source location for stimulus set 2 but not for stimulus set 1, the log-contrast tends to $-\infty$; when r_j is neither a source for stimulus set 1 nor a source for stimulus 2, the log-contrast tends to a finite value depending on the associated values of $a_{xx|q}$. The details are omitted here.*

3.2 Beamformer analysis when C is estimated

We now estimate the sensor covariance matrix by using the sensor observations over J time instants. Following Bickel and Levina (2008) and Fan et al. (2011), we establish the asymptotic theory for the resulting beamformer estimators when both n and J are tending to infinity.

In addition to Conditions (A1) and (A2), we need the following two conditions for conducting the asymptotic analysis above. The first one is imposed to regularize the tail behavior of the sensor processes.

(A3): There exist positive constants κ_1 and τ_1 such that for any $u > 0$ and all t,

$$\max_{1\le i\le n} P(||Y_i(t)|| > u) \le \exp(1 - \tau_1 u^{\kappa_1})$$

and $\max_{1\le i\le n} E||Y_i(t)||^2 < +\infty$, where the noise covariance matrix is $\sigma_0^2 I_n$ and $||\cdot||$ is the L_2 norm.

Note that Condition (A3) holds if $\mathbf{Y}(t)$ is a multivariate normal.

In the second additional condition, we assume that the sensor processes are strong mixing. Let $\mathcal{F}_{-\infty}^0$ and $\mathcal{F}_k^\infty$ denote the σ-algebras generated by $\{\mathbf{Y}(t) : -\infty \le t \le 0\}$ and $\{\mathbf{Y}(t) : t \ge k\}$ respectively. Define the mixing coefficient

$$\alpha(k) = \sup_{A\in\mathcal{F}_{-\infty}^0, B\in\mathcal{F}_k^\infty} |P(A)P(B) - P(AB)|.$$

The mixing coefficient $\alpha(k)$ quantifies the degree of the temporal dependence of the process $\{\mathbf{Y}(t)\}$ at lag k. We assume that $\alpha(k)$ is decreasing exponentially fast as lag k is increasing.

(A4): There exist positive constants κ_2 and τ_2 such that

$$\alpha(k) \leq \exp(-\tau_2 k^{\kappa_2}).$$

Condition (A4) is a commonly used assumption for studying asymptotic behavior of time series.

For a constant A, let $\tau_{nJ} = A\sqrt{\log(n)/J}$. As before, let $\bar{Y}_i$ be the sample mean of the i-th sensor and

$$\hat{c}_{ik} = \frac{1}{J}\sum_{j=1}^{J}(Y_i(t_j) - \bar{Y}_i)(Y_k(t_j) - \bar{Y}_k), \quad \hat{C}(\tau_{nJ}) = (\hat{c}_{ik} I(\hat{c}_{ik} \geq \tau_{nJ})),$$

where $I(\cdot)$ is the indicator.

We are now in position to generalize Theorem 1 to the case where the sensor covariance is estimated by the thresholded covariance estimator.

Theorem 6 *Under Conditions (A1)∼(A4) and assuming that* $n^2\sqrt{\log(n)/J} = o(1)$ *as* n *and* J *tend to infinity, we have:*

(1) If $a_{nq} = O(1)$ *and* $\max_{1\leq k\leq q} d_{k|q} = O(1)$*, then as* n *and* J *tend to infinity, the estimated source covariance at* r_{k_m+1} $\hat{\Sigma}_{k_m+1}$ *is asymptotically larger than* Σ_{k_m+1} *in probability.*

(2) If $a_{nq} \to \infty$*, then as* n *and* J *tend to infinity, for* $1 \leq j \leq q$*, the estimated source covariance at* r_j *admits*

$$\hat{\Sigma}_j = \Sigma_j + \frac{1}{n}\Sigma_j c_{jj|q}\Sigma_j + O_p(a_{nq}^{-2} + n^2\sqrt{\log(n)/J}),$$

provided $\max_{1\leq k\leq q} d_{k|q} = O(1)$*, where* $||\Sigma_j c_{jj|q}\Sigma_j/n|| = O(a_{nq}^{-1})$ *as* $n \to \infty$*.*

(3) If $a_{nq} \to \infty$*, then as* n *and* J *tend to infinity, for* $r_x \notin \{r_1, ..., r_q\}$*, the estimated source covariance at* r_x *admits*

$$\hat{\Sigma}_x \quad = \quad \frac{1}{n}a_{xx|q}^{-1} - \frac{1}{n^2}a_{xx|q}^{-1}b_{xx|q}a_{xx|q}^{-1} + O(a_{nq}^{-3} + n^2\sqrt{\log(n)/J}),$$

provided $\max_{1\leq j\leq q} d_{j|q} = O(1)$*,* $||na_{xx|q}|| \to \infty$*, and* $d_{x|q} = O(1)$ *as* n *tends to infinity, where* $b_{xx|q} = O(1)$ *as* $n \to \infty$*.*

Corollary 7 *Under Conditions (A1)∼(A4) and assuming that* $n^2\sqrt{\log(n)/J} = o(1)$ *as* n *and* J *tend to infinity, we have:*

(1) If $a_{nq} = O(1)$*,* $\max_{1\leq k\leq q} d_{k|q} = O(1)$*, as* n *and* J *tend to infinity, the estimated source power at* r_{k_m+1}*,* $\hat{S}_{k_m+1}$ *is asymptotically larger than* $tr(\Sigma_{k_m+1})$*.*

(2) If $a_{nq} \to \infty$*, then as* n *and* J *tend to infinity, for* $1 \leq j \leq q$*, the estimated source power at* r_j *admits*

$$\hat{S}_j = tr(\Sigma_j) + \frac{1}{n}tr(\Sigma_j c_{jj|q}\Sigma_j) + O(a_{nq}^{-2} + n^2\sqrt{\log(n)/J}),$$

provided $\max_{1\leq k\leq q} d_{k|q} = O(1)$*, where* $||\Sigma_j c_{jj|q}\Sigma_j/n|| = O(a_{nq}^{-1})$ *as* $n \to \infty$*.*

(3) If $a_{nq} \to \infty$, then as n and J tend to infinity, for $r_x \notin \{r_1, ..., r_q\}$, the estimated source power at r_x admits

$$\hat{S}_x = \frac{1}{n} tr(a^{-1}_{xx|q}) - \frac{1}{n^2} tr(a^{-1}_{xx|q} b_{xx|q} a^{-1}_{xx|q}) + O(a^{-3}_{nq} + n^2\sqrt{\log(n)/J}),$$

provided $\max_{1\le j\le q} d_{j|q} = O(1)$, $||na_{xx|q}|| \to \infty$, *and* $d_{x|q} = O(1)$ *as n tends to infinity, where* $b_{xx|q} = O(1)$ *as* $n \to \infty$.

Remark 8 *Theorem 6 indicates the convergence rate of the vector-beamformer estimation is much slower than the empirical rate suggested by Rodríguez-Rivera et al. (2006). However, the result is in agreement with an empirical result of Brookes et al. (2008). In fact, using their heuristic arguments, we can show that the error of the power estimation at location r_x is determined by the factor $H_x(\hat{C}(\tau_{nJ})^{-1} - C^{-1})H_x$, which has a rate of $n^2\sqrt{\log(n)/J}$.*

Theorem 6 can be also extended to the scenarios where MEG data are obtained under two different sets of stimuli.

Remark 9 *From the proof of Theorem 6, we can see that the thresholded covariance is still consistent with the true C even when the underlying sources are correlated.*

4. Other covariance estimator-based beamformers

There are various ways to estimate the sensor covariance matrix. Each can be used to construct a beamformer. These covariance estimators can be roughly divided into two categories, namely global shrinkage-based methods and elementwise thresholding-based methods. In shrinkage-based settings, the sample covariance is shrinking toward a target structure (for example, a diagonal matrix). The so-called optimal shrinkage estimator belongs to this category (Ledoit and Wolf, 2004). In thresholding-based settings, an elementwise thresholding is applied to the sample covariance estimator. Examples of these approaches include hard thresholding, generalized thresholding and adaptive thresholding (Bickel and Levina, 2008; Rothman et al., 2009; Cai and Liu, 2011). Here, we focus on the following three methods recommended by the above authors.

The optimal shrinkage covariance matrix is defined by

$$\hat{C}_{opt} = \frac{b_n^2}{d_n^2}\mu_n I_n + \frac{d_n^2 - b_n^2}{d_n^2}\hat{C},$$

where

$$\mu_n = \left\langle \hat{C}, I_n \right\rangle, \quad d_n^2 = \left\langle \hat{C} - \mu_n I_n, \hat{C} - \mu_n I_n \right\rangle,$$

$$\bar{b}_n^2 = \frac{1}{J^2}\sum_{j=1}^{J} \left\langle \mathbf{Y}_j\mathbf{Y}_j^T - \hat{C}, \mathbf{Y}_j\mathbf{Y}_j^T - \hat{C} \right\rangle, \quad b_n^2 = \min(\bar{b}_n^2, d_n^2),$$

and the operator $< A, B >= \mathrm{tr}(AB^T)/n$ for any $n \times n$ matrices A and B. The idea behind the above estimator is to find the optimal weighted average of the sample covariance matrix $\hat{C}$ and the identity matrix via minimizing the expected squared loss. Under certain

conditions $\hat{C}_{opt}$ converges to the true covariance C as n tends infinity, implying that $\hat{C}_{opt}$ can be degenerate if C is degenerate (Ledoit and Wolf, 2004). As before, we tackle the issue by adding $\epsilon_0 I_n$ to $\hat{C}_{opt}$, where ϵ_0 is determined by the maximum eigenvalue of the pre-stimulus sample covariance matrix. The beamformer based on the above covariance estimator is denoted as **sh**.

A family of generalized thresholding-based covariance estimators indexed by tuning constants $c_0 \geq 0$ and $\delta_0 > 0$ can be defined by replacing the hard thresholding in Subsection 2.1 with the generalized thresholding, i.e.,

$$\hat{C}_g = (g(\hat{c}_{ij}))$$

with $g(\hat{c}_{ij}) = \hat{c}_{ij}(1 - (\tau_{nJ}/|\hat{c}_{ij}|)^{\delta_0})$, where $\tau_{nJ} = c_0\hat{\sigma}_0^2\sqrt{\log(n)/J}$ and $\hat{\sigma}_0^2$ is estimated from a baseline sample. Following the suggestion of Rothman et al. (2009), we choose $\delta_0 = 4$. The same maximum/minimum strategy as in Subsection 2.3 can be adapted to choose the tuning constant c_0 when we use the above estimator to construct a beamformer. The corresponding beamformers are denoted by **gmax** and **gmin** respectively.

Similarly, an adaptive thresholding estimator can be introduced by replacing the above τ_{nJ} in the g function by $\lambda_{ij} = 2\sqrt{\hat{\theta}_{ij}\log(n)/J}$, where $\hat{\theta}_{ij}$ is the estimated variance of the (i,j)-th entry $\hat{c}_{ij}$ and is defined by

$$\hat{\theta}_{ij} = \frac{1}{J}\sum_{k=1}^{J}[(Y_{ik} - \bar{Y}_i)(Y_{jk} - \bar{Y}_j) - \hat{c}_{ij}]^2$$

and $\bar{Y}_i$ and $\bar{Y}_j$ are the sample means of the i-th and the j-th sensors. See Cai and Liu (2011). The corresponding beamformer is denoted by **adp**.

5. Numerical results

In this section, we compare the proposed procedures to the standard vector-beamformer (with the tuning $c_0 = 0$) and to the other covariance estimator-based beamformers in terms of localization bias by simulation studies and real data analyses. Here, for any estimator $\hat{r}$ of a source location r, the localization bias $|\hat{r} - r|$ is the L_1 distance between $\hat{r}$ and r. The spatial correlation $\rho_{\max}$ between locations r_1 and r_2 is measured by the maximum correlation between the projected lead field vectors at r_1 and r_2:

$$\rho_{\max}(r_1, r_2) = \left\{ \max_{||\eta_1||=1, ||\eta_2||=1} \frac{(l(r_1)\eta_1)^T l(r_1)\eta_1}{||l(r_1)\eta_1|| \cdot |l(r_2)\eta_2||} \right\}.$$

By simulations, we attempted to answer the following questions:

- Has the vector-beamformer been improved by using the thresholded covariance estimator?
- To what extent will the performance of the proposed beamformer procedure deteriorate by source interferences (or source cancellations) and source spatial correlations?
- Can the proposed beamformers **ma** and **mi** be superior to the other covariance estimator-based beamformers?

5.1 Simulated data

We started with specifying the following two head models (Sarvas, 1987). The simple head model that uses a homogeneous sphere in simulating the magnetic fields emanating from current electric dipole neuronal activity possesses the advantage that the lead field matrix can be calculated analytically. However, with more realistic head models, the numerical approximations such as a finite element method have to be used when we calculate the lead field matrix. Here, we considered both of them: the simple one is a spherical volume conductor with 10cm radius from the origin and with 91 sensors, created by using the software FieldTrip (Oostenveld et al., 2010), and the realistic one is a single shell head model by using the magnetic resonance imaging (MRI) scan of a human brain provided by Henson et al. (2011). We then discretized the inside brain space into a 3D-grid of resolution 1 cm. This yielded a grid with 2222 points for the simple model and 1487 points for the realistic model. The grids was further sliced into 10 and 14 transverse layers along the z-axis of the brain respectively. We put two non-null sources at r_1 and r_2 or three sources at r_1, r_2 and r_3 respectively, where two sources $\{r_1, r_2\}$ are equal to $\{(3,-1,4)^T, (-5,2,6)^T\}$ cm or $\{(-5,5,6)^T, (-6,-2,5)^T\}$ cm, and three sources $\{r_1, r_2, r_3\}$ are equal to $\{(3,-1,4)^T, (-5,2,6)^T, (5,5,6)^T\}$ in the Subject Coordinate System (SCS/CTF). Note that the second set of source locations was obtained in our real data analyses which will be presented later. These sources were located in the region of the parietal and occipital lobes, where visual, auditory and touch information is processed. We considered two types of sources in the brain: evoked responses that are phase-locked to the stimulus and induced responses that are not. The induced responses often have oscillatory patterns. Combining these sources with the two head models, we had the following four scenarios:

- **Scenario 1**: For the simple head model, we put two oscillatory sources at locations $r_1 = (3,-1,4)^T$ and $r_2 = (-5,2,6)^T$ with time-courses

$$\mathbf{m}_k(t) = \eta_k \cos(20t\pi), \quad k = 1, 2,$$

  respectively, where $\eta_1 = (10,1,1)^T$ and $\eta_2 = (8,0,0)^T$. We considered two values of the signal-to-noise-ratio (SNR): 0.04 and $1/0.64 = 1.5625$.

- **Scenario 2**: For the simple head model, we put the above oscillatory sources at locations $r_1 = (-5,5,6)^T$ and $r_2 = (-6,-2,5)^T$. We also considered two values of the SNR: 0.04 and $1/0.64 = 1.5625$.

- **Scenario 3**: For the realistic head model, we put the following evoked response sources at locations $r_1 = (3,-1,4)^T$ and $r_2 = (-5,2,6)^T$ with moments (i.e., time-courses)

$$\mathbf{m}_k(t) = \alpha_k \exp(-(t-\tau_{k1})^2/\omega_k^2) \sin(f_k 2\pi(t-\tau_{k2})), k = 1, 2,$$

  respectively, where $\alpha_1 = (5,0,0)^T$, $\alpha_2 = (20,0,0)^T$, $\tau_{11} = 0.239$, $\tau_{12} = 0.139$, $\tau_{21} = 0.199$, $\tau_{22} = 0.139$, $f_1 = 4.75$, $f_2 = 6.25$, and $\omega_1 = \omega_2 = 0.067$. We considered three values of the SNR: $1/0.35^2 = 8.16, 1/0.4^2 = 6.25, 1/0.5^2 = 4$.

- **Scenario 4**: For the realistic head model, we put the following evoked response sources at locations $r_1 = (-5, 5, 6)^T$ and $r_2 = (-6, -2, 5)^T$ with moments (i.e., time-courses)

$$\mathbf{m}_k(t) = \alpha_k \exp(-(t - \tau_{k1})^2/\omega_k^2) \sin(f_k 2\pi(t - \tau_{k2})), k = 1, 2,$$

  respectively, where $\alpha_1 = (2, 0, 0)^T$, $\alpha_2 = (18, 0, 0)^T$, $\tau_{11} = 0.439$, $\tau_{12} = 0.139$, $\tau_{21} = 0.399$, $\tau_{22} = 0.139$, $f_1 = 6$, $f_2 = 9$, and $\omega_1 = \omega_2 = 2$. We considered three values of the SNR: $1/0.7^2 = 2.04, 1/0.76^2 = 1.73, 1/0.78^2 = 1.64$.

- **Scenario 5**: We added another evoked response source at location $r_3 = (5, 5, 6)^T$ to the model in Scenario 3 with moment

$$\mathbf{m}_3(t) = \alpha_3 \exp(-(t - \tau_{31})^2/\omega_3^2) \sin(f_3 2\pi(t - \tau_{32})),$$

  where $\alpha_3 = (2.5, 0.25, 0.25)$, $\tau_{31} = 0.1$, $\tau_{32} = 0.139$, $f_3 = 1.25$, and $w_3 = 0.067$. The three source locations are highly spatially correlated with the pairwise spatial correlations $\rho(r_1, r_2) = 0.7289$, $\rho(r_1, r_3) = 0.7935$, and $\rho(r_2, r_3) = 0.5924$. We considered the same SNR values as in Scenario 3.

The pair sources $\mathbf{m}_k(t), k = 1, 2$ for the first four scenarios and the treble sources $\mathbf{m}_k(t), k = 1, 2, 3$ for Scenario 5 are plotted respectively in Figure 1. By Scenarios 1 and 2, we compared the proposed procedure to the standard vector-beamformer (with $c_0 = 0$) and to the other estimator-based beamformer, when there existed two highly correlated oscillatory sources (they have the same frequency and phase, but with slightly different amplitudes). By Scenarios 3 and 4, we tested these beamformers when there existed two unbalanced evoked response (or slightly dumped-oscillatory) sources. By Scenario 5, we assessed these beamformers when there were three spatially correlated source locations. In each scenario, with time-window width 1 and sample rate J, we sampled 30 data sets of $\mathbf{Y}(t)$ from the model

$$\mathbf{Y}(t) = \sum_{k=1}^{p} H_k \mathbf{m}_k(t) + \varepsilon(t), \tag{5.5}$$

where in Scenarios 1∼4, $\mathbf{m}_k(t)$, $k = 1, 2$ are non-null time-courses at the two locations and $\mathbf{m}_k(t), 3 \le k \le p$ are null time-courses at other grid points, while in Scenario 5, $\mathbf{m}_k(t)$, $k = 1, 2, 3$ are non-null time-courses at the three locations and $\mathbf{m}_k(t), 4 \le k \le p$ are null time-courses at other grid points. As before, $\{\varepsilon(t)\}$ is a white noise process with noise level σ_0^2. We considered various combinations of $(n, p) = (91, 2222)$ and $(102, 1487)$, and $J = 500, 1000, 2000$, and 3000. Note that p is substantially larger than n and that the sources are sparse in the sense that there are only two or three non-null sources among p candidates.

We first applied the proposed procedures **ma**, **mi** and **sh** to each data set. We calculated the maximum indices over the grids and the L_1-biases of the maximum location estimates to two sources respectively. For each combination of (n, p, J) and the SNR, we then summarized these values in the form of a box-whisker plot as in Figures 2, 3, 4, and 5 corresponding to Scenarios 1, 2, 3, and 4 respectively. The results demonstrate that the proposed hard thresholding-based procedure **mi** can outperform both the conventional vector-beamformer

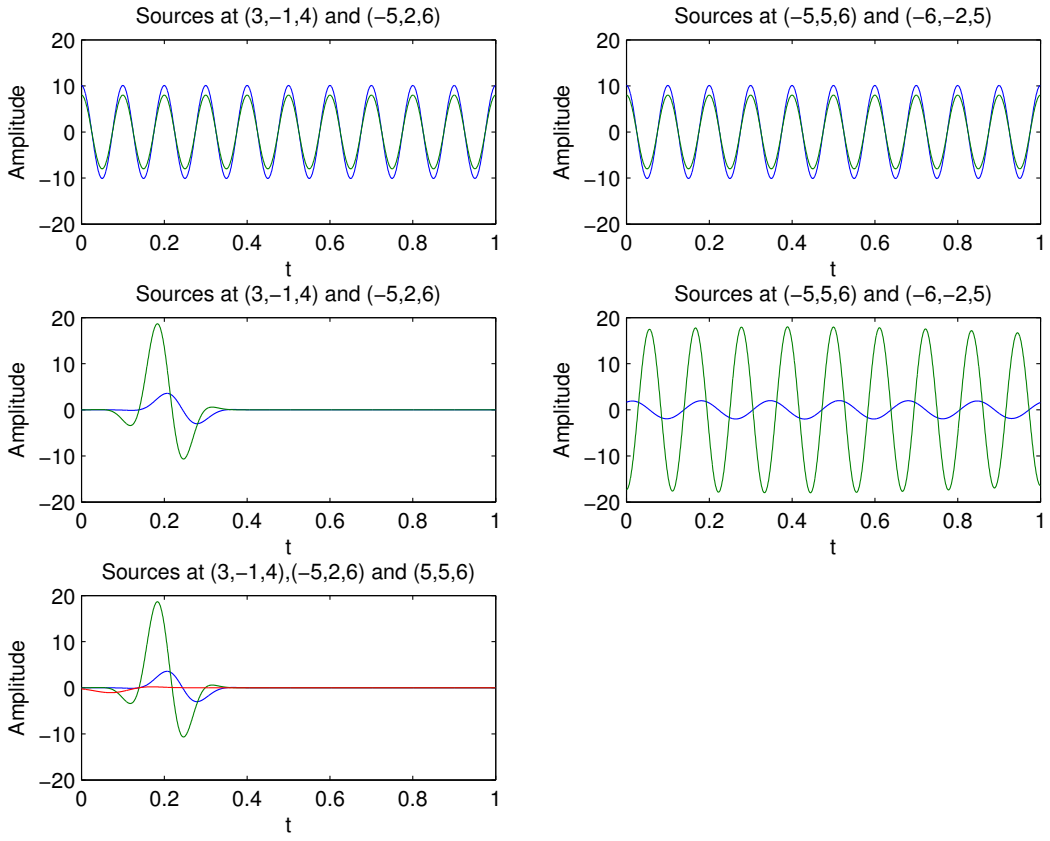


Figure 1: The amplitude plots of $\mathbf{m}_k(t), k = 1, 2$ for Scenarios 1 to 4 and the amplitude plots of $\mathbf{m}_k(t), k = 1, 2, 3$ for Scenario 5. In these plots, the blue, green and red colored curves are corresponding to the amplitudes of $\mathbf{m}_k(t)$, $k = 1, 2, 3$ respectively.

and the procedures **ma** and **sh** in all four scenarios, in particular when the SNR is low. We note that in several cases, the localization bias and the maximum index were degenerate to a single value with some outliers, indicating that random variations have not changed the global peak location although they have effects on local peaks on the map. The simulations also suggest that the proposed procedure may be unable to detect evoked response sources of low SNR values. The local peak box-whisker plots in these figures reveal that all the local peaks on the transverse slices are not close to the source location r_1, implying that the source at r_1 has been masked on the neuronal activity index-based map even when two sources have a similar power level. This may be due to source cancellations as the lead field vectors at these two locations were correlated and the sensor positions might favor the detection of r_2. Finally, we note that the results are robust to the choice of J in the sense that increasing sampling frequency has only slightly reduced both the mean and standard error of localization bias.

To compare the procedures **ma**, **mi** and **sh** with the procedures **gma**, **gmi** and **adp** based on the generalized and adaptive thresholding, we again generated 30 data sets from model (5.5) for each of the above four scenarios and for each combination of $(n, p) = (91, 2222)$ and $(102, 1487)$, and $J = 500, 1000, 2000$, and 3000. We applied these procedures to each data set and calculated their localization biases respectively. As before, we displayed these biases by multiple box-whisker plots in Figures 6, 7 and 8. From these figures, we can see a dramatic improvement in localization performance of the hard thresholding-based procedure **mi** over the other procedures in Scenarios 1 and 2 and a slightly better or similar performance to **ma**, **gma**, **gmi**, **adp** and **sh** in Scenarios 3 and 4. This is striking because the existing studies have already shown that the soft (or generalized) and adaptive thresholding-based covariance estimators can improve the hard thresholding-based covariance estimator in terms of estimation loss. The potential explanations for this phenomena are as follows: (1) The procedure **adp** may lose efficiency by not using the pre-stimulus data. (2) The existing covariance estimators were aimed to improve the estimation accuracy by reducing the estimation loss (the distance between the estimator and the true covariance matrix) or by increasing the sensitivity and specificity in recovering sparse entries in the true covariance matrix (Rothman et al., 2009; Cai and Liu, 2011). Unfortunately, the sparsity in MEG means a sparse signal distribution, which is quite different from the entry sparsity of the sensor covariance matrix. Therefore, these estimators may be not efficient for improving the accuracy of the beamformer estimation which is related to the signal sparsity. In fact, our simulation experience suggests that besides the covariance estimation, there are other factors that can affect the performance of a beamformer such as the lead field matrix and the spatial distribution of signals in the brain. Therefore, the covariance estimator with a smaller estimation loss may not give rise to a beamformer with a lower localization bias.

To assess the performances of the six procedures **ma**, **mi**, **gma**, **gmi**, **adp** and **sh** when there are more than two spatially correlated sources, we applied these procedures to the 30 data sets generated for Scenario 5. We calculated the average localization bias for each procedure and presented them in Figure 9. It can be seen from these plots that like in two-source scenarios, **mi** can have superior performance over the other procedures. However, compared the above result to those in Scenario 3, we can see that the source cancellation from r_3 has increased the average localization bias from zero to the value of three.

Note that although Theorem 2 suggests that in general the localization bias will be reduced as the sampling rate increases, it does not implies the localization bias is a monotone function of the sampling rate (or the number of time instances). In fact, from row 4 in Figure 2 and row one in Figure 9, it can be seen that the localization bias when $J = 500$ is smaller than when $J = 1000, 2000$ and 3000. A potential explanation is that in finite cases a higher sampling rate may cause a higher amount of leakage of background noises (in a neighborhood of the target location) into the neuronal activity index calculation.

Finally, we notice that we also carried out simulations with the soft thresholding ($\delta_0 = 1$). The result is very similar to the case with $\delta_0 = 4$. For reasons of space, we do not report it here.

5.2 Face-perception data

We applied the proposed methodology to human MEG data acquired in five sessions by Wakeman and Henson (Henson et al., 2011). In each session, 96 face trials and 50 scrambled face trials were performed on a healthy young adult subject. Each trial started with a central fixation cross (presented for a random duration of 400 to 600 ms), followed by a face or scrambled face (presented for a random duration of 800 to 1000 ms), and followed by a central circle for 1700 ms. The subject used either his/her left or right index finger to report whether he/she thought the stimulus was symmetrical or asymmetrical vertically through its center. The data were collected with a Neuromag VectorView system, containing a magnetometer and two orthogonal, planar gradiometers located at each of 102 positions within a hemispherical array situated in a light, magnetically shielded room. The sampling rate was 1100Hz. We focused our analysis on localizing non-null source positions, where neuronal activity increases for the face stimuli relative to the scrambled face stimuli.

For this purpose, we normalized the subject's MRI scan to a MRI template by using the FieldTrip, on which a grid CTF system of 1 cm resolution was created with 1487 points. For each session, we applied the neuroimaging software SPM8 to read and preprocess the recorded data, and to epoch and average the data generated from the face stimulus trials and the scrambled face stimulus trials respectively. This gives rise to five 306×771 data matrices: the first 220 columns for 200ms pre-stimuli and the later 551 columns for the stimuli. For each session, we calculated the sample covariance $\hat{C}$ and noise covariance $\hat{C}_0$ by using the stimulus data and the pre-stimulus data respectively. We estimated the baseline noise level by $\hat{\sigma}_0^2$, the minimum diagonal element in $\hat{C}_0$. We applied the beamforming procedures **ma**, **mi**, **gma**, **gmi**, **adp**, and **sh** to the face data set and the scrambled face data set respectively, obtaining the log-contrasts at each grid point. Here, if there exist the negative eigenvalues of the covariance estimators (used in **ma**, **mi**, **gma**, **gmi**, **adp** and **sh**), we set them to zeros and added ϵ_0 to them to make the resulting covariance estimators positive definite, where ϵ_0 was determined by the maximum eigenvalue of the noise matrix $\hat{C}_0$. For each procedure, we interpolated and overlaid its log-contrasts on the structural MRI of the subject, obtaining its index map. There were no visible differences among the maps derived from **ma**, **mi**, **gma**, **gmi** and **sh**. The map derived from the **adp** slightly differed from the rest. So, we reported only the **mi**-based and **adp**-based maps below.

For each session, we first identified the global peak location from each map, followed by slicing the maps through their global peak locations as shown in Figure 10. For sessions $1 \sim 4$, the global peaks derived from the mi and adp were the same, which were located at $(-4, 3, 8)$cm, $(-1, -6, 8)$cm, $(-6, -2, 5)$cm, and $(-4, -4, 6)$cm respectively. However, for session 5, the global peaks derived from the **mi** and the **adp** were located at two slightly different positions, $(-4, -4, 6)$cm and $(-7, -3, 6)$cm. We then projected the data along the associated optimal weight directions, obtaining estimated time-courses at these global peaks. For reasons of space, we presented only these time-courses derived from the procedure **mi**. See Figure 12. Finally, we made 20 transverse slices along the z-axis to identify the local peaks. There were some subtle differences between the **mi**-based and the **adp**-based local peaks. For example, in session 1, the **mi**-based local peaks were located at $(1, 5, 2)$ cm, $(0, -1, 11)$ cm, $(3, 2, 10)$ cm, $(3, 4, 9)$ cm, $(-5, -3, 3)$ cm, $(-4, -3, 4)$ cm, $(-2, 1, 1)$ cm,$(-4, -3, -1)$ cm,$(-2, 1, 0)$ cm,$(-4, -5, 5)$ cm,$(-4, 2, 6)$ cm,$(-5, 3, 7)$ cm and

$(-4, 3, 8)$ cm, whereas the **adp**-based local peaks were located at $(3, 2, 2)$cm, $(0, -1, 11)$cm, $(-4, 3, 9)$cm, $(-6, -2, 1)$cm, $(-4, -3, 4)$cm, $(2, 3, 10)$cm, $(-4, -3, -1)$ cm, $(-1, 1, 0)$ cm, $(-3, 6, 3)$ cm, $(-4, -4, 5)$ cm, $(-4, 2, 6)$ cm, $(-5, 3, 7)$ cm, and $(-4, 3, 8)$cm. They are not the same as shown in Figure 11. Note that the previous simulations demonstrated that the procedure **mi** was expected to give a more accurate localization result than did the procedure **adp**.

Although the areas highlighted in Figures 10 and 11 were varying over sessions, they did reveal the following known regions of face perception: the occipital face area (OFA), the inferior occipital gyrus (IOG), and the superior temporal sulcus (STS), and the precuneus (PCu). Interestingly, in each session, we identified a pair of nearly symmetric sources, of which one was strongly powered while the other was weakly powered. This phenomenon occurred due to source cancellations that prevented the second source from identification as we have demonstrated in our simulation studies. The time-courses plots in Figure 12 showed the response differences under face stimuli and scrambled face stimuli during the time period 100ms$\sim$300ms. The results are consistent with recent findings in face-perception studies by using an MEG-based multiple sparse prior approach (Friston et al., 2006; Henson et al., 2011) and by other empirical approaches (e.g., Pitcher et al., 2011; Kanwisher et al., 1997). However, in the first two papers, the authors made a parametric model assumption on source temporal correlation structures and imposed a limit on the number of candidate sources in the model, whereas in our approach, the model is non-parametric and allows for arbitrary number of candidate sources.

6. Discussion and Conclusion

In the present study, we have proposed a class of vector-beamformers by thresholding the sensor sample covariance matrix. The consistency and the convergence rate of the proposed vector-beamformer estimation have been proved in the presence of multiple sources. The theory has provided a basis for choosing the threshold $\tau_{nJ} = c_0\sigma_0^2\sqrt{\log(n)/J}$ in the beamformer construction. However, it requires a number of conditions. As pointed out in Section 3, conditions (A1)$\sim$(A4) are commonly used assumptions in literature for studying multiple time series (Sekihara and Nagarajan, 2008; Fan et al., 2011). We only need to validate the coherence stability condition which is new. Intuitively, the strength of correlations between sensors (therefore the absolute partial correlation) will increase when the number of sensors increases in general. Taking the face-perception data (session 1) as an example, we show how to validate it empirically by random sub-samples of the 306 sensors below. We take the first two peaks in Figure 8 as two true sources. They are located at CTF (-4,3,8) cm and (-4,-5,5) cm respectively. First, we reparametrize the lead field matrix as in Section 3. Then, for $k = 1, 2, ..., 306$, we randomly choose k sensors, obtaining a $k \times 4461$ sub lead field matrix for the 1487 voxels in the brain. We calculate the maximum absolute partial correlation $d_{12}(k) = \max\{d_{1|2}, d_{2|2}\}$ between the two sources and the maximum absolute correlation $d_{\max}(k) = \max d_{x|2}$ for all voxels, where x is running over these voxels. Finally, we plot $d_{12}(k)$, $d_{\max}(k)$, and $\log(\log(k))$ against $k = 1, 2, ..., 306$ respectively as displayed in Figure 13. As expected, the result shows that both $d_{12}(k)$ and $d_{\max}(k)$ change very slowly when the number of sensors k changes, with a rate much slower than $\log(\log(k))$. This implies that the coherence stability condition nearly holds.

In real world situations, the underlying number of true sources, q needs to be estimated. The influence of q on the beamformer estimators can be measured by the lead field partial correlation coefficient a_{nq} defined in Section 3. In this paper, local peaks on transverse slices have been used to reduce the search space of sources. We can cluster the local peak values into two groups, one of which is taken as a group of potential sources. The size of the selected group gives an estimate of q. In the face-perception data, we have only presented the first two sources which are ranked higher than the remaining local peaks, because these two are of clear neurological implications. Our approach is non-parametric in the sense that we have not made any parametric assumptions on the model (1.1). However, if we are willing to assume a family of parametric models for background noises, then we can determine q via model selection criteria such as Bayesian information criterion.

By theoretical and empirical studies, we have shown that due to source cancellations, the beamformer power estimator can be inconsistent if the underlying multiple sources are not well separated in terms of a lead field distance. Unlike the existing theories in the literature, the new theory is applicable to more general scenarios, where multiple sources exist and the sensor covariance matrix are estimated from the data. In the new theory, we do assume that the powers of the unknown no-null sources as well as the underlying number q are not growing with the number of sensors n. This assumption is natural to neurologists and has simplified mathematical derivations of the theory very much. However, the theory can be extended to the case where these quantities are growing with n. In the theory, we have not impose any constraint on p as we only consider local behavior of beamformers. If we want to investigate global properties of the neuronal activity map, then some constraints need to be imposed on the growth rate of p with respect to n.

The performances of the proposed beamformers have further been assessed by simulations and real data analyses. We have demonstrated that thresholding the sensor covariance matrix can help reduce the source localization bias when the data have a low SNR value. We have applied the vector-beamformer to an MEG data set for identifying the active regions related to human face perception. Some excellent agreements have been found between the current results and the existing neurological facts on human face perception. Finally, we note that there are other ways to measure the contrast between two source covariances such as the information-divergence. The theory can be easily extended to this case. The details will be presented elsewhere.

7. Proofs

In this section we prove the theorems and corollaries in Section 3.

To prove Theorem 1, we need the following lemma.

Lemma 10 *If $a_{nq} \to \infty$ as $n \to \infty$, then we have*

$$\begin{aligned}
H_j^T C_k^{-1} H_j &= b_{jj|k} + \frac{c_{jj|k}}{n} + O(a_{nk}^{-2}), \quad b_{jj|k} = \Sigma_j^{-1}, \quad \text{for } 1 \le j \le k \\
H_{j_1}^T C_k^{-1} H_{j_2} &= \frac{c_{j_1 j_2|k}}{n} + O(a_{nk}^{-2}), \quad \text{for } 1 \le j_1 \ne j_2 \le k \\
H_j^T C_k^{-1} H_x &= b_{jx|k} + \frac{c_{jx|k}}{n} + O(a_{nk}^{-2}), \quad \text{for } 1 \le j \le k,\ x \notin R_k \\
H_y^T C_k^{-1} H_x &= n a_{yx|k} + b_{yx|k} + O(a_{nk}^{-1}), \quad \text{for } x, y \notin R_k
\end{aligned}$$

where $a_{nk} = n \min_{1\le j\le k-1} tr\left(a_{(j+1)(j+1)|j}\right)$, $R_k = \{r_1, \ldots, r_k\}$, $C_k = \sum_{j=1}^{k} H_j^T \Sigma_j H_j + \sigma_0^2 I_n$, *and* $a_{yx|k}$, $b_{yx|k}$ *and* $c_{jj|k}$ *are defined before and the other* c'*s are defined iteratively as follows:*

$$c_{j_1 j_2|k} = \begin{cases} b_{j_1 k|(k-1)} \Sigma_k^{-1} a_{kk|(k-1)}^{-1}, & 1 \le j_1 \le k-1, j_2 = k \\ \Sigma_k^{-1} a_{kk|(k-1)}^{-1} b_{kj_2|(k-1)}, & 1 \le j_2 \le k-1, j_1 = k \\ c_{j_1 j_2|(k-1)} - b_{j_1 k|(k-1)} a_{kk|(k-1)}^{-1} b_{kj_2|(k-1)}, & 1 \le j_1 \ne j_2 \le k-1. \end{cases}$$

$$c_{jx|k} = \begin{cases} \left(a_{kk|(k-1)}\Sigma_k\right)^{-1} \left\{ b_{kx|(k-1)} \right. \\ \left. - \left(I_3 + b_{kk|(k-1)}\right) \left(a_{kk|(k-1)}\Sigma_k\right)^{-1} a_{kx|(k-1)} \right\}, & j = k \\ c_{jx|(k-1)} - c_{jk|(k-1)} a_{kk|(k-1)}^{-1} a_{kx|(k-1)} \\ -b_{jk|(k-1)} a_{kk|(k-1)}^{-1} b_{kx|(k-1)} & 1 \le j \le k-1 \\ +b_{jk|(k-1)} a_{kk|(k-1)}^{-1} \left[\Sigma_k^{-1} + b_{kk|(k-1)}\right] a_{kk|(k-1)}^{-1} a_{kx|(k-1)}. \end{cases}$$

Proof Note that under the stability condition and the assumption that $a_{nq} \to \infty$, we have $b_{yx|k} = O(1),\ 1 \le k \le q$. And for any r_x in the source space,

$$\frac{c_{1x|1}}{n} = O(n^{-1}), \quad \frac{c_{yx|k}}{n} = O(a_{n(k-1)}^{-1}), 1 \le y \le k, 2 \le k \le q.$$

We prove the lemma by induction. For $k = 1$, we have

$$\begin{aligned} C_1^{-1} &= \sigma_0^{-2} I_n - \sigma_0^{-4} H_1 (\Sigma_1^{-1} + n\sigma_0^{-2} I_3)^{-1} H_1^T, \\ H_1^T C_1^{-1} H_1 &= n\sigma_0^{-2} I_3 - n^2 \sigma_0^{-4} (\Sigma_1^{-1} + n\sigma_0^{-2} I_3)^{-1} \\ &= n\sigma_0^{-2} \left(I_3 - \left(I_3 + \Sigma_1^{-1} \frac{\sigma_0^2}{n} \right)^{-1} \right) \\ &= n\sigma_0^{-2} \left(I_3 + n\Sigma_1 \sigma_0^{-2} \right)^{-1} \\ &= \Sigma_1^{-1} \left(I_3 - \sigma_0^2 \Sigma_1^{-1} / n \right) + O(n^{-2}) \\ &= \Sigma_1^{-1} - \Sigma_1^{-1} \sigma_0^2 \Sigma_1^{-1} / n + O(n^{-2}) \\ &= b_{11|1} + \frac{c_{11|1}}{n} + O(n^{-2}), \end{aligned}$$

where

$$b_{11|1} = \Sigma_1^{-1},\ c_{11|1} = -\sigma_0^2 \Sigma_1^{-2}.$$

Analogously,

$$\begin{aligned} H_1^T C_1^{-1} H_x &= \sigma^{-2} H_1^T H_x - \sigma_0^{-4} n (\Sigma_1^{-1} + n\sigma_0^{-2} I_3)^{-1} H_1^T H_x \\ &= \left(I_3 - \left(I_3 + \Sigma_1^{-1} \sigma_0^2 / n \right)^{-1} \right) H_1^T H_x \\ &= \left(I_3 + \frac{n}{\sigma_0^2} \Sigma_1 \right)^{-1} H_1^T H_x \\ &= \Sigma_1^{-1} \left(I_3 + \frac{\sigma_0^2}{n} \Sigma_1^{-1} \right)^{-1} \rho_{1x} \\ &= \Sigma_1^{-1} \rho_{1x} - \Sigma_1^{-1} \sigma_0^2 \Sigma_1^{-1} \rho_{1x} / n + O(n^{-2}) \\ &= b_{1x|1} + \frac{c_{1x|1}}{n} + O(n^{-2}), \end{aligned}$$

where

$$b_{1x|1} = \Sigma_1^{-1}\rho_{1x},\ c_{11|1} = -\Sigma_1^{-2}\sigma_0^2\rho_{1x}.$$

And

$$\begin{aligned} H_y^T C_1^{-1} H_x &= \sigma_0^{-2} H_y^T H_x - \sigma_0^{-4} H_y^T H_1 (\Sigma_1^{-1} + n\sigma_0^{-2} I_3)^{-1} H_1^T H_x \\ &= n\sigma_0^{-2}\rho_{yx} - \sigma_0^{-4} H_y^T H_1 \frac{\sigma_0^2}{n} \left(I_3 + \sigma_0^2\Sigma_1^{-1}/n\right)^{-1} H_1^T H_x \\ &= n\sigma_0^{-2}\rho_{yx} - n\sigma_0^{-2}\rho_{y1}\left(I_3 - \sigma_0^2\Sigma_1^{-1}/n\right)\rho_{1x} + O(n^{-1}) \\ &= n\sigma_0^{-2}\rho_{y1x} + \rho_{y1}\Sigma_1^{-1}\rho_{1x} + O(n^{-1}) \\ &= n a_{yx|1} + b_{yx|1} + O(n^{-1}), \end{aligned}$$

where

$$\rho_{y1x} = \rho_{yx} - \rho_{y1}\rho_{1x},\ a_{yx|1} = \sigma_0^{-2}\rho_{y1x},\ b_{yx|1} = \rho_{y1}\Sigma_1^{-1}\rho_{1x}.$$

This implies the lemma holds for $k = 1$.

Assuming the lemma holds for the cases with less or equal to k sources, we show that it is also true for the case with $k+1$ sources by invoking the matrix inversion formulas

$$\begin{aligned} C_{k+1}^{-1} &= C_k^{-1} - C_k^{-1} H_{k+1}\left(\Sigma_{k+1}^{-1} + H_{k+1}^T C_k^{-1} H_{k+1}\right)^{-1} H_{k+1}^T C_k^{-1}, \\ C_k^{-1} &= C_{k+1}^{-1} + C_{k+1}^{-1} H_{k+1}\Sigma_{k+1} H_{k+1}^T C_k^{-1}. \end{aligned} \tag{7.6}$$

The details are as follows.

For $1 \le j \le k$,

$$\begin{aligned} H_j^T C_{k+1}^{-1} H_j &= H_j^T C_k^{-1} H_j - \left(H_j^T C_k^{-1} H_{k+1}\right) \times \left(\Sigma_{k+1}^{-1} + H_{k+1}^T C_k^{-1} H_{k+1}\right)^{-1} \left(H_{k+1}^T C_k^{-1} H_j\right) \\ &= b_{jj|k} + \frac{c_{jj|k}}{n} + O(a_{nk}^{-2}) - \left(b_{j(k+1)|k} + \frac{c_{j(k+1)|k}}{n} + O(a_{nk}^{-2})\right) \\ &\quad \times \left(\Sigma_{k+1}^{-1} + n a_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-1})\right)^{-1} \\ &\quad \times \left(b_{j(k+1)|k} + \frac{c_{j(k+1)|k}}{n} + O(a_{nk}^{-2})\right)^T \\ &= b_{jj|k} + \frac{c_{jj|k}}{n} + O(a_{nk}^{-2}) \\ &\quad - \left(b_{j(k+1)|k} + O(a_{nk}^{-1})\right)\left((n a_{(k+1)(k+1)|k})^{-1} - O(a_{n(k+1)}^{-2})\right) \\ &\quad \times \left(b_{j(k+1)|k} + O(a_{nk}^{-1})\right)^T \\ &= b_{jj|k} + \frac{c_{jj|k}}{n} - \frac{1}{n} b_{j(k+1)|k} a_{(k+1)(k+1)|k}^{-1} b_{j(k+1)|k}^T + O(a_{n(k+1)}^{-2}) \\ &= b_{jj|(k+1)} + \frac{c_{jj|(k+1)}}{n} + O(a_{n(k+1)}^{-2}). \end{aligned}$$

For $j = k+1$, we have

$$\begin{aligned} H_j^T C_{k+1}^{-1} H_j &= H_{k+1}^T C_{k+1}^{-1} H_{k+1} = H_{k+1}^T C_k^{-1} H_{k+1}\left(I_3 + \Sigma_{k+1} H_{k+1}^T C_k^{-1} H_{k+1}\right)^{-1} \\ &= \left(n a_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-1})\right) \\ &\quad \times \left(I_3 + \Sigma_{k+1}\left(n a_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-1})\right)\right)^{-1} \end{aligned}$$

$$\begin{aligned}
&= \left(na_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-1})\right)\left(na_{(k+1)(k+1)|k}\right)^{-1} \\
&\quad \times \left(I_3 + (na_{(k+1)(k+1)|k})^{-1}\Sigma_{k+1}^{-1} + O(a_{n(k+1)}^{-2})\right)^{-1}\Sigma_{k+1}^{-1} \\
&= \Sigma_{k+1}^{-1} - \frac{1}{n}\Sigma_{k+1}^{-1}a_{(k+1)(k+1)|k}^{-1}\Sigma_{k+1}^{-1} + O(a_{n(k+1)}^{-2}) \\
&= b_{(k+1)(k+1)|(k+1)} + \frac{c_{(k+1)(k+1)|(k+1)}}{n} + O(a_{n(k+1)}^{-2}).
\end{aligned}$$

This completes the proof of the first equation in the lemma.

To prove the second equation in the lemma, we let

$$\begin{aligned}
B &= \left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-1} + \left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-\frac{1}{2}} \\
&\quad \times\Sigma_{k+1}b_{(k+1)(k+1)|k}\left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-\frac{1}{2}}.
\end{aligned}$$

Then, when $1 \le j_1 \le k$, $j_2 = k+1$, we have

$$\begin{aligned}
H_{j_1}^T C_{k+1}^{-1}H_{j_2} &= H_{j_1}^T C_{k+1}^{-1}H_{k+1} = H_{j_1}^T C_k^{-1}H_{k+1}\left\{I_3 + \Sigma_{k+1}H_{k+1}^T C_k^{-1}H_{k+1}\right\}^{-1} \\
&= \left(b_{j_1(k+1)|k} + \frac{1}{n}c_{j_1(k+1)|k} + O(a_{nk}^{-2})\right) \\
&\quad \times\left(I_3 + \Sigma_{k+1}\left(na_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-1})\right)\right)^{-1} \\
&= \left(b_{j_1(k+1)|k} + \frac{1}{n}c_{j_1(k+1)|k} + O(a_{nk}^{-2})\right)\left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-\frac{1}{2}} \\
&\quad \times(I_3 + B + O(a_{n(k+1)}^{-2}))^{-1}\left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-\frac{1}{2}} \\
&= b_{j_1(k+1)|k}\left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-\frac{1}{2}} \\
&\quad \times\left(I_3 + O\left((na_{(k+1)(k+1)|k})^{-1}\right)\right)\left[\Sigma_{k+1}na_{(k+1)(k+1)|k}\right]^{-\frac{1}{2}} + O(a_{n(k+1)}^{-2}) \\
&= \frac{1}{n}b_{j_1(k+1)|k}\Sigma_{k+1}^{-1}a_{(k+1)(k+1)|k}^{-1} + O(a_{n(k+1)}^{-2}) \\
&= \frac{c_{j_1(k+1)|(k+1)}}{n} + O(a_{n(k+1)}^{-2}).
\end{aligned}$$

Similarly, when $1 \le j_1 \ne j_2 \le k$, we have

$$\begin{aligned}
H_{j_1}^T C_{k+1}^{-1}H_{j_2} &= H_{j_1}^T C_k^{-1}H_{j_2} - H_{j_1}^T C_k^{-1}H_{k+1}\left(\Sigma_{k+1}^{-1} + H_{k+1}^T C_k^{-1}H_{k+1}\right)^{-1}H_{k+1}^T C_k^{-1}H_{j_2} \\
&= \frac{1}{n}c_{j_1j_2|k} + O(a_{nk}^{-2}) - \frac{1}{n}b_{j_1(k+1|k)}a_{(k+1)(k+1)|k}^{-1}b_{(k+1)j_2|k} + O(a_{n(k+1)}^{-2}) \\
&= \frac{1}{n}c_{j_1j_2|(k+1)} + O(a_{n(k+1)}^{-2}).
\end{aligned}$$

We complete the proof of the second equation in the lemma.

To prove the third equation in the lemma, we let

$$\begin{aligned}
D &= \left(na_{(k+1)(k+1)|k}\Sigma_{k+1}\right)^{-1} + \left(na_{(k+1)(k+1)|k}\Sigma_{k+1}\right)^{-\frac{1}{2}} \\
&\quad \times b_{(k+1)(k+1k|k)}\Sigma_{k+1}\left(na_{(k+1)(k+1)|k}\Sigma_{k+1}\right)^{-\frac{1}{2}}, \\
F &= (na_{(k+1)(k+1)|k})^{-\frac{1}{2}}\left(\Sigma_{k+1}^{-1} + b_{(k+1)(k+1)|k}\right)(na_{(k+1)(k+1)|k})^{-\frac{1}{2}}.
\end{aligned}$$

Then, for $j = k+1$,

$$\begin{aligned}
H_j^T C_{k+1}^{-1} H_x &= \left[I_3 + H_{k+1}^T C_k^{-1} H_{k+1} \Sigma_{k+1}\right]^{-1} H_{k+1}^T C_k^{-1} H_x \\
&= \left[I_3 + n a_{(k+1)(k+1)|k} \Sigma_{k+1} + b_{(k+1)(k+1)|k} \Sigma_{k+1} + O(a_{nk}^{-1})\right]^{-1} \\
&\quad \times \left[n a_{(k+1)x|k} + b_{(k+1)x|k} + O(a_{nk}^{-1})\right] \\
&= \left(n a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-\frac{1}{2}} \left(I_3 + D + O(a_{n(k+1)}^{-2})\right)^{-1} \left(n a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-\frac{1}{2}} \\
&\quad \times \left[n a_{(k+1)x|k} + b_{(k+1)x|k} + O(a_{nk}^{-1})\right] \\
&= \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-\frac{1}{2}} \left\{I_3 - D + O(a_{n(k+1)}^{-2})\right\} \\
&\quad \times \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-\frac{1}{2}} \left(a_{(k+1)x|k} + b_{(k+1)x|k}/n + O(a_{nk}^{-1})/n\right) \\
&= \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1} a_{(k+1)x|k} \\
&\quad - \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1/2} D \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1/2} a_{(k+1)x|k} \\
&\quad + O(a_{n(k+1)}^{-3}) + \frac{1}{n} \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1} b_{(k+1)x|k} + O(a_{n(k+1)}^{-2}) \\
&= \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1} a_{(k+1)x|k} + \frac{1}{n} \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1} \\
&\quad \times \left\{b_{(k+1)x|k} - \left(I_3 + b_{(k+1)(k+1)|k} \Sigma_{k+1}\right) \left(a_{(k+1)(k+1)|k} \Sigma_{k+1}\right)^{-1} a_{(k+1)x|k}\right\} \\
&\quad + O(a_{n(k+1)}^{-2}) \\
&= b_{(k+1)x|(k+1)} + \frac{1}{n} c_{(k+1)x|(k+1)} + O(a_{n(k+1)}^{-2}).
\end{aligned}$$

For $1 \le j \le k$, we have

$$\begin{aligned}
H_j^T C_{k+1}^{-1} H_x &= H_j^T C_k^{-1} H_x - H_j^T C_k^{-1} H_{k+1} \left(\Sigma_{k+1}^{-1} + H_{k+1}^T C_k^{-1} H_{k+1}\right)^{-1} H_{k+1}^T C_k^{-1} H_x \\
&= b_{jx|k} + \frac{1}{n} c_{jx|k} + O(a_{nk}^{-2}) - \left(b_{j(k+1)|k} + \frac{1}{n} c_{j(k+1)|k} + O(a_{nk}^{-2})\right) \\
&\quad \times \left(\Sigma_{k+1}^{-1} + n a_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-2})\right)^{-1} \\
&\quad \times \left(n a_{(k+1)x|k} + b_{(k+1)x|k} + O(a_{nk}^{-1})\right) \\
&= b_{jx|k} + \frac{1}{n} c_{jx|k} + O(a_{nk}^{-2}) - \left(b_{j(k+1)|k} + \frac{1}{n} c_{j(k+1)|k} + O(a_{nk}^{-2})\right) \\
&\quad \times (n a_{(k+1)(k+1)|k})^{-1/2} \left[I_3 + F + O(a_{nk}^{-2})\right]^{-1} (n a_{(k+1)(k+1)|k})^{-1/2} \\
&\quad \times \left(n a_{(k+1)x|k} + b_{(k+1)x|k} + O(a_{nk}^{-1})\right) \\
&= b_{jx|k} + \frac{1}{n} c_{jx|k} + O(a_{nk}^{-2}) - \left(b_{j(k+1)|k} + \frac{1}{n} c_{j(k+1)|k} + O(a_{nk}^{-2})\right) \\
&\quad \times \Big(a_{(k+1)(k+1)|k}^{-1} a_{(k+1)x|k} - a_{(k+1)(k+1)|k}^{-1/2} F a_{(k+1)(k+1)|k}^{-1/2} a_{(k+1)x|k} + O(a_{nk}^{-2}) \\
&\quad (n a_{(k+1)(k+1)|k})^{-1} b_{(k+1)x|k} - a_{(k+1)(k+1)|k}^{-1/2} F a_{(k+1)(k+1)|k}^{-1/2} b_{(k+1)x|k}/n \\
&\quad + O(a_{n(k+1)}^{-2})\Big)
\end{aligned}$$

$$
\begin{aligned}
&= b_{jx|k} + \frac{1}{n}c_{jx|k} + O(a_{nk}^{-2}) - \left(b_{j(k+1)|k} + \frac{1}{n}c_{j(k+1)|k} + O(a_{nk}^{-2})\right) \\
&\quad \times \left(a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k} - \frac{1}{n}a^{-1}_{(k+1)(k+1)|k}(\Sigma_{k+1}^{-1} + b_{(k+1)(k+1)|k}) \right. \\
&\quad \left. \times a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k}\frac{1}{n}a^{-1}_{(k+1)(k+1)|k}b_{(k+1)x|k} + O(a^{-2}_{n(k+1)})\right) \\
&= b_{jx|k} + \frac{1}{n}c_{jx|k} - b_{j(k+1)|k}a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k} \\
&\quad -\frac{1}{n}\left\{c_{j(k+1)|k}a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k} + b_{j(k+1)|k}a^{-1}_{(k+1)(k+1)|k}b_{(k+1)x|k}\right. \\
&\quad \left. -b_{j(k+1)|k}a^{-1}_{(k+1)(k+1)|k}\left[\Sigma_{k+1}^{-1} + b_{(k+1)(k+1)|k}\right]a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k}\right\} \\
&\quad +O(a^{-2}_{n(k+1)}) \\
&= b_{jx|k} - b_{j(k+1)|k}a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k} + \frac{1}{n}c_{jx|(k+1)} + O(a^{-2}_{n(k+1)}). \qquad (7.7)
\end{aligned}
$$

Note that for $k = j$,

$$a_{jx|j} = a_{jx|(j-1)} - a_{jj|(j-1)}a^{-1}_{jj|(j-1)}a_{jx|(j-1)} = 0.$$

Assuming that for $k = j + m, m > 0$, the statement is true, i.e., $a_{jx|(k+m)} = 0$ for all x. Then,

$$
\begin{aligned}
a_{jx|(j+m+1)} &= a_{jx|(j+m)} - a_{j(j+m+1)|(j+m)}a^{-1}_{(j+m+1)(j+m+1)|(j+m)}a_{(j+m+1)x|(j+m)} \\
&= 0.
\end{aligned}
$$

By induction, we have that $a_{jx|k} = 0$ for all x, $j \le k$. This implies that and

$$b_{jx|(k+1)} = b_{jx|k} - b_{j(k+1)|k}a^{-1}_{(k+1)(k+1)|k}a_{(k+1)x|k}$$

by the definition of $b_{jx|(k+1)}$. Combining this with (7.7), we complete the proof of the third equation in the lemma.

Finally, we turn to the last equation in the lemma. Assume that the equation holds for the case k. We show that it also holds for $k + 1$ below. For $x, y \notin R_{k+1}$ (thus $x, y \notin R_k$), by the assumption, we have

$$H_y^T C_k^{-1} H_x = na_{yx|k} + b_{yx|k} + O(a_{nk}^{-1}).$$

This together with (7.6) yields

$$
\begin{aligned}
H_y^T C_{k+1}^{-1} H_x &= H_y^T C_k^{-1} H_x - H_y^T C_k^{-1} H_{k+1}\left(\Sigma_{k+1}^{-1} + H_{k+1}^T C_k^{-1} H_{k+1}\right)^{-1} H_{k+1}^T C_k^{-1} H_x \\
&= na_{yx|k} + b_{yx|k} + O(a_{nk}^{-1}) - \left(na_{y(k+1)|k} + b_{y(k+1)|k} + O(a_{nk}^{-1})\right) \\
&\quad \times \left(\Sigma_{k+1}^{-1} + na_{(k+1)(k+1)|k} + b_{(k+1)(k+1)|k} + O(a_{nk}^{-1})\right)^{-1} \\
&\quad \times \left(na_{(k+1)x|k} + b_{(k+1)x|k} + O(a_{nk}^{-1})\right) \\
&= na_{yx|k} + b_{yx|k} + O(a_{nk}^{-1}) - \left(na_{y(k+1)|k} + b_{y(k+1)|k} + O(a_{nk}^{-1})\right)
\end{aligned}
$$

$$
\begin{aligned}
&\times \left((na_{(k+1)(k+1)|k})^{-1} - \frac{1}{n^2} a^{-1}_{(k+1)(k+1)|k} \left(\Sigma^{-1}_{k+1} + b_{(k+1)(k+1)|k} \right) \right. \\
&\left. \times a^{-1}_{(k+1)(k+1)|k} + O(a^{-3}_{n(k+1)}) \right) \left(na_{y(k+1)|k} + b_{y(k+1)|k} + O(a^{-1}_{nk}) \right) \\
= \;& na_{yx|k} + b_{yx|k} + O(a^{-1}_{nk}) - \left\{ a_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} \right. \\
&- a_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} \left(\Sigma^{-1}_{k+1} + b_{(k+1)(k+1)|k} \right) a^{-1}_{(k+1)(k+1)|k} / n \\
&\left. + b_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} / n + O(a^{-2}_{n(k+1)}) \right\} \\
&\times \left(na_{(k+1)x|k} + b_{(k+1)x|k} + O(a^{-1}_{nk}) \right) \\
= \;& n \left[a_{yx|k} - a_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} a_{(k+1)x|k} \right] \\
&+ \left[b_{yx|k} - b_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} a_{(k+1)x|k} - a_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} b_{(k+1)x|k} \right. \\
&\left. + a_{y(k+1)|k} a^{-1}_{(k+1)(k+1)|k} \left(\Sigma^{-1}_{k+1} + b_{(k+1)(k+1)|k} \right) a^{-1}_{(k+1)(k+1)|k} a_{(k+1)x|k} \right] \\
&+ O(a^{-1}_{n(k+1)}) \\
= \;& na_{yx|(k+1)} + b_{yx|(k+1)} + O(a^{-1}_{n(k+1)}).
\end{aligned}
$$

The proof is completed. ■

Proof of Theorem 1. Note that $b_{yx|1} = \rho(r_y, r_1)\Sigma_1^{-1}\rho(x_1, x)$, $a_{yx|1} = \sigma_0^{-2}(\rho_{yx} - \rho_{y1}\rho_{1x})$, and both are bounded. By induction and the stability condition, it can be shown that $a_{yx|k}$ and $b_{yx|k}$ are bounded for $2 \le k \le q$. If a_{nq} is bounded, then there exists k_m such that $na_{(k_m+1)(k_m+1)|k_m} = O(1)$ and $a_{nk_m} = \min_{1 \le j \le k_m - 1} na_{(j+1)(j+1)|j} \to \infty$ as n tends to infinity. By Lemma 10, we have

$$
H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \;=\; na_{(k_m+1)(k_m+1)|k_m} + b_{(k_m+1)(k_m+1)|k_m} + O(a^{-1}_{nk_m}),
$$

which is bounded and non-negative definite. Furthermore, there exists an orthogonal matrix Q and a diagonal matrix $D = \mathrm{diag}(d_1, d_2, d_3)$ such that

$$
\Sigma^{1/2}_{k_m+1} H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \Sigma^{1/2}_{k_m+1} = QDQ^T.
$$

Therefore,

$$
\begin{aligned}
& H^T_{k_m+1} C^{-1}_{k_m+1} H_{k_m+1} \\
&= H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \left(I_3 - \left(\Sigma^{-1}_{k_m+1} + H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \right)^{-1} H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \right) \\
&= H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \left(\Sigma^{-1}_{k_m+1} + H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \right)^{-1} \Sigma^{-1}_{k_m+1} \\
&= H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \Sigma^{1/2}_{k_m+1} \left(I_3 + \Sigma^{1/2}_{k_m+1} H^T_{k_m+1} C^{-1}_{k_m} H_{k_m+1} \Sigma^{1/2}_{k_m+1} \right)^{-1} \Sigma^{-1/2}_{k_m+1} \\
&= \Sigma^{-1/2}_{k_m+1} QDQ^T \left(I_3 + QDQ^T \right)^{-1} \Sigma^{-1/2}_{k_m+1} \\
&= \Sigma^{-1/2}_{k_m+1} \left(I_3 + QD^{-1}Q^T \right)^{-1} \Sigma^{-1/2}_{k_m+1} \\
&= \Sigma^{-1/2}_{k_m+1} \left(Q(I_3 + D^{-1})Q^T \right)^{-1} \Sigma^{-1/2}_{k_m+1} \\
&= \Sigma^{-1/2}_{k_m+1} Q(I_3 + D^{-1})^{-1} Q^T \Sigma^{-1/2}_{k_m+1}. \qquad (7.8)
\end{aligned}
$$

Note that $\Sigma_{k_m+1}^{1/2} H_{k_m+1}^T C_{k_m}^{-1} H_{k_m+1} \Sigma_{k_m+1}^{1/2} = O(1)$, which implies that $d_k \geq 0, 1 \leq k \leq 3$ are bounded. We can find a positive constant ϵ_0 such that $\max_{1\leq k\leq 3}(1+d_k^{-1})^{-1} < (1+\epsilon_0)^{-1}$ when n is large enough. Consequently, for any vector $a \in \mathbb{R}^3$ with $||a|| = 1$, we have

$$a^T \Sigma_{k_m+1}^{1/2} Q(I_3 + D^{-1}) Q^T \Sigma_{k_m+1}^{1/2} a > (1+\epsilon_0) a^T \Sigma_{k_m+1}^{1/2} Q Q^T \Sigma_{k_m+1}^{1/2} a,$$

which shows that $\Sigma_{k_m+1}^{1/2} Q(I_3 + D^{-1}) Q^T \Sigma_{k_m+1}^{1/2}$ (thus $\left[H_{k_m+1}^T C_{k_m+1}^{-1} H_{k_m+1}\right]^{-1}$ due to (7.8)) is asymptotically larger than $\Sigma_{k_m+1}(1+\epsilon_0)$.

We now consider the case where $a_{nq} \to \infty$. For $j = q$, by Lemma 10, we have

$$\begin{aligned}
\frac{c_{qq|q}}{n} &= -\Sigma_q^{-1} \left[n a_{qq|(q-1)}\right]^{-1} \Sigma_l^{-1} = O(a_{nq}^{-1}), \\
\left[H_q^T C_q^{-1} H_q\right]^{-1} &= \left[\Sigma_q^{-1} + \frac{c_{qq|q}}{n} + O(a_{nq}^{-2})\right]^{-1}, \\
&= \Sigma_q^{1/2} \left[I_3 + \Sigma_q^{1/2} \frac{c_{qq|q}}{n} \Sigma_q^{1/2} + O(a_{nq}^{-2})\right]^{-1} \Sigma_q^{1/2} \\
&= \Sigma_q^{1/2} \left[I_3 - \Sigma_q^{1/2} \frac{c_{qq|q}}{n} \Sigma_q^{1/2} + O(a_{nq}^{-2})\right] \Sigma_q^{1/2} \\
&= \Sigma_q - \left[n a_{qq|(q-1)}\right]^{-1} + O(a_{nq}^{-2})
\end{aligned}$$

as $n \to \infty$. For $1 \leq j \leq q-1$, by Lemma 10, we have

$$H_j^T C_q^{-1} H_j = \Sigma_j^{-1} + \frac{c_{jj|q}}{n} + O(a_{nq}^{-2}),$$

where $\frac{c_{jj|q}}{n} = O(a_{nq}^{-1})$. This entails

$$\begin{aligned}
\left[H_j^T C_q^{-1} H_j\right]^{-1} &= \Sigma_j^{1/2} \left(I_3 + \frac{1}{n} \Sigma_j^{1/2} c_{jj|q} \Sigma_j^{1/2} + O(a_{nq}^{-2})\right)^{-1} \Sigma_j^{1/2} \\
&= \Sigma_j^{1/2} \left(I_3 - \frac{1}{n} \Sigma_j^{1/2} c_{jj|q} \Sigma_j^{1/2} + O(a_{nq}^{-2})\right) \Sigma_j^{1/2} \\
&= \Sigma_j - \frac{1}{n} \Sigma_j c_{jj|q} \Sigma_j + O(a_{nq}^{-2}).
\end{aligned}$$

For any location r_x, by Lemma 10, we have

$$\begin{aligned}
\left[H_x^T C_q^{-1} H_x\right]^{-1} &= \frac{1}{n} \left[I_3 + \frac{1}{n} a_{xx|q}^{-1} b_{xx|q} + O(a_{nq}^{-2})\right]^{-1} a_{xx|q}^{-1} \\
&= \frac{1}{n} \left[I_3 - \frac{1}{n} a_{xx|q}^{-1} b_{xx|q} + O(a_{nq}^{-2})\right] a_{xx|q}^{-1} \\
&= \frac{1}{n} a_{xx|q}^{-1} - \frac{1}{n^2} a_{xx|q}^{-1} b_{xx|q} a_{xx|q}^{-1} + O(a_{nq}^{-3}).
\end{aligned}$$

The proof is completed.

Proof of Corollary 2. First, let $A_n = \left[{H_{k_m+1}}^T C_l^{-1} H_{k_m+1}\right]^{-1}$. If $a_{nq} = O(1)$ and $\max_{1\leq k\leq q} d_{k|q} = O(1)$, then by Theorem (1), there exists a positive constant ϵ_0 such that

$\min_{||a||=1} a^T(A_n - \Sigma_{k_m+1})a > \epsilon_0$ for large n. Let $a_1 = (1,0,0)^T, a_2 = (0,1,0)^T$ and $a_3 = (0,0,1)^T$. Then, we have

$$\begin{aligned}
\mathrm{tr}(A_n) &= \mathrm{tr}(A_n \sum_{k=1}^{3} a_k a_k^T) = \sum_{k=1}^{3} \mathrm{tr}(A_n a_k a_k^T) \\
&= \sum_{k=1}^{3} a_k^T A_n a_k > 3\epsilon_0 + \sum_{k=1}^{3} a_k \Sigma_{k_m+1} a_k^T \\
&= 3\epsilon_0 + \sum_{k=1}^{3} \mathrm{tr}(\Sigma_{k_m+1} a_k a_k^T) = 3\epsilon_0 + \mathrm{tr}(\Sigma_{k_m+1} \sum_{k=1}^{3} a_k a_k^T) \\
&= 3\epsilon_0 + \mathrm{tr}(\Sigma_{k_m+1}),
\end{aligned}$$

which implies $\mathrm{tr}(A_n)$ is asymptotically larger than Σ_{k_m+1}.

To prove Theorem 2, we need two more lemmas as follows and the following condition

$$(A1') : \{\mathbf{Y}(t_j) : 1 \le j \le J\} \text{ is stationary and has a finite covariance matrix.}$$

Lemma 11 *Under Conditions (A1') and (A3)∼(A4), if* $\tau_{nJ} = O(\sqrt{\log(n)/J})$ *and* $n\tau_{nJ} = o(1)$ *as* $n \to \infty$ *and* $J \to \infty$, *then*

(i) $\max_{1\le i,j\le n} |\hat{c}_{ij} - c_{ij}| = O_p(\sqrt{\log(n)/J})$,

(ii) $||\hat{C}(\tau_{nJ}) - C|| = O_p(m_n\sqrt{\log(n)/J})$,

(iii) $||\hat{C}(0) - C|| \le (m_n + n)\tau_{nJ}$,

where $m_n = \max_{1\le i\le n} \sum_{j=1}^{n} I(c_{ij} \ne 0) \le n$.

Proof. Let $\kappa_3 = \max\{2(2/\kappa_1 + 1/\kappa_2) - 1, (4/3)(1/\kappa_1 + 1/\kappa_2) - 1/3, 1\}$. Then $n\sqrt{\log(n)/J} = o(1)$ yields $(\log(n))^{\kappa_3}/J = o(1)$. We adopted the techniques of Bickel and Levina (2008); Fan et al. (2011); Zhang et al. (2014) to prove it. To prove (i), we set up more notations. Let $\tau(t)$ be the so-called Dedecker-Prieur τ-mixing coefficients (Merlevède et al., 2011, see). Let

$$\Theta(u,t) = \infty\{v > 0 : P(|y_1(t)y_2(t)| > v) \le u\}, \quad \psi_y(M,t) = \max\{\min\{y_i(t)y_j(t), M\}, -M\}.$$

It follows from Lemma 7 in Dedecker and Prieur (2004) that

$$\sup_t \Theta(u,t) \le b_1(1 - \log(u))^{2/\kappa_1},$$

which, under Condition (A4), gives $\tau(t) \le b_2 \exp(-b_3 t^{\kappa_2})$. Similarly, it is derived from Remark 3 in Merlevède et al. (2011) that

$$\begin{aligned}
&\sup_{M>0}[\sup_t \mathrm{var}(\psi_y(M,t)) + 2\sum_{t_1>t_2} |\mathrm{cov}(\psi_y(M,t_1), \psi_y(M,t_2))] \\
&\quad \le \sup_{M>0}\sup_t \mathrm{var}(\psi_y(M,t)) \\
&+2\left(\sup_{M>0}\sup_t \mathrm{var}(\psi_y(M,t)) + 4\sum_{t>0}\int_0^{2\alpha(t)} (\sup_t \Theta(u))^2 du\right) < \infty.
\end{aligned}$$

Let $1/\kappa = 2/\kappa_1 + 1/\kappa_2$. By Theorem 1 in Merlevède et al. (2011), we can find positive constants $d_k, 1 \le k \le 5$ that only depend on $\tau_1, \kappa_2, b_2, b_3$ such that

$$\begin{aligned} P\left(|\frac{1}{J}\sum_{t=1} y_i(t)y_j(t) - c_{ij}| \ge u\right) &\le J\exp\left(-\frac{(Ju)^\kappa}{d_1}\right) + \exp\left(-\frac{(Ju)^2}{d_2(1+Jd_3)})\right) \\ &\quad + \exp\left(-\frac{(Ju)^2}{d_4 J}\exp\left(\frac{(Ju)^{\kappa(1-\kappa)}}{d_5(\log(Ju))^\kappa}\right)\right). \end{aligned}$$

Consequently,

$$\begin{aligned} &P\left(\max_{1\le i,j\le n}|\frac{1}{J}\sum_{t=1}^{J} y_i(t)y_j(t) - c_{ij}| > u\right) \\ &\le n^2 \max_{1\le i,j\le n} P\left(|\frac{1}{J}\sum_{t=1}^{J} y_i(t)y_j(t) - c_{ij}| > u\right) \\ &\le n^2 J\exp\left(-\frac{(Ju)^\kappa}{d_1}\right) + n^2\exp\left(-\frac{(Ju)^2}{d_2(1+Jd_3)}\right) \\ &+n^2\exp\left(-\frac{(Ju)^2}{d_4 J}\exp\left(\frac{(Ju)^{\kappa(1-\kappa)}}{d_5(\log(Ju))^\kappa}\right)\right). \end{aligned}$$

Let $u = A\sqrt{\log(n)/J}$. Then $Ju = \sqrt{J\log(n)}$. When both n and J tend to infinity, we have

$$\begin{aligned} n^2 J\exp\left(-\frac{(Ju)^\kappa}{d_1}\right) &= \exp\left(2\log(n) + \log(J) - \frac{(A\sqrt{J\log(n)})^\kappa}{d_1}\right) \\ &= \exp\left((2\frac{(\log(n))^{1-\kappa/2}}{J^{\kappa/2}} - \frac{A}{d_1})(J\log(n))^{\kappa/2} + \log(J)\right) \\ &= o(1), \end{aligned}$$

since $(\log(n))^{1-\kappa/2}/J^{\kappa/2} = o(1)$. Similarly, if we choose $A > \sqrt{2d_2(d_3+1)}$, we have

$$\begin{aligned} n^2\exp\left(-\frac{(Ju)^2}{d_2(1+Jd_3)}\right) &= n^2\exp\left(-\frac{A^2 J\log(n)}{d_2(1+Jd_3)}\right) \\ &= \exp\left(\left(2 - \frac{A^2}{d_2(d_3+1/J)}\right)\log(n)\right) = o(1). \end{aligned}$$

And

$$\begin{aligned} &n^2\exp\left(-\frac{(Ju)^2}{d_4 J}\exp\left(\frac{(Ju)^{\kappa(1-\kappa)}}{d_5(\log(Ju))^\kappa}\right)\right) \\ &= \exp\left(\log(n)\left(2 - \frac{A^2}{d_4}\exp\left(\frac{A^{\kappa(1-\kappa)}(J\log(n))^{\kappa(1-\kappa)/2}}{d_5(\log(A\sqrt{J\log(n)}))^\kappa}\right)\right)\right) \\ &= o(1). \end{aligned}$$

Therefore,

$$P\left(\max_{1\le i,j\le n}|\frac{1}{J}\sum_{t=1}^{J}y_i(t)y_j(t)-c_{ij}|>u\right)=o(1). \tag{7.9}$$

Note that for $u=A\sqrt{\log(n)/J}$, there exist positive constants $d_k, 1\le k\le 5$ so that

$$\begin{aligned}
P(\max_{1\le i,j\le n}|\bar{y}_i||\bar{y}_j|>u) &= P(\max_{1\le i\le n}|\bar{y}_i|>\sqrt{u})\\
&\le n\max_{1\le i\le n}P(|\bar{y}_i|>\sqrt{u})\\
&= nJ\exp\left(-\frac{(J\sqrt{u})^{\kappa_1}}{d_1}\right)+n\exp\left(-\frac{(J\sqrt{u})^2}{d_2(1+Jd_3)}\right)\\
&\quad +n\exp\left(-\frac{(J\sqrt{u})^2}{d_4J}\exp\left(\frac{(J\sqrt{u})^{\kappa_1(1-\kappa_1)}}{d_5(\log(Ju))^{\kappa_1}}\right)\right)\\
&= o(1),
\end{aligned}$$

since $(\log(n))^{4/(3\kappa_1)-1/3}/J=o(1)$ and $\log(n)/J=o(1)$. This together with (7.9) yields that for $u=O(\sqrt{\log(n)/J})$,

$$\begin{aligned}
P\left(\max_{1\le i,j\le n}|\hat{c}_{ij}-c_{ij}|>u\right) &\le P\left(\max_{1\le i,j\le n}|\frac{1}{J}\sum_{t=1}^{J}y_i(t)y_j(t)-c_{ij}|>u\right)\\
&\quad +P\left(\max_{1\le i,j\le n}|\bar{y}_i||\bar{y}_j|>u\right)=o(1),
\end{aligned}$$

which implies

$$\max_{1\le i,j\le n}|\hat{c}_{ij}-c_{ij}|=O_p\left(\sqrt{\log(n)/J}\right).$$

We turn to $\hat{C}(\tau_{nJ})$ in (ii). Let $T_1=||(\hat{c}_{ij}I(|\hat{c}_{ij}|>\tau_{nJ})-c_{ij}I(|c_{ij}|>\tau_{nJ}))||$. We have

$$\begin{aligned}
||\hat{C}(\tau_{nJ})-C|| &\le ||(\hat{c}_{ij}I(|\hat{c}_{ij}|>\tau_{nJ})-c_{ij}I(|c_{ij}|>\tau_{nJ}))||+||(c_{ij}I(|c_{ij}|\le\tau_{nJ}))||\\
&\le T_1+\max_i\sum_{j=1}^{n}|c_{ij}|I(|c_{ij}|\le\tau_{nJ})\\
&\le T_1+\tau_{nJ}m_n. \qquad (7.10)
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
||\hat{C}(0)-C|| &\le T_1+\tau_{nJ}m_n+\max_i\sum_{j=1}^{n}|\hat{c}_{ij}|I(|\hat{c}_{ij}|\le\tau_{nJ})\\
&\le T_1+(m_n+n)\tau_{nJ}.
\end{aligned}$$

Note that

$$T_1 \le \max_i\sum_{j=1}^{n}|\hat{c}_{ij}I(|\hat{c}_{ij}|>\tau_{nJ})-c_{ij}I(|c_{ij}|>\tau_{nJ})|$$

$$
\begin{aligned}
&= \max_i \sum_{j=1}^n |\hat{c}_{ij}\,(I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| \le \tau_{nJ}) + I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| > \tau_{nJ})) \\
&\quad -c_{ij}\,(I(|c_{ij}| > \tau_{nJ}, |\hat{c}_{ij}| > \tau_{nJ}) + I(|c_{ij}| > \tau_{nJ}, |\hat{c}_{ij}| \le \tau_{nJ}))\,| \\
&\le \text{I} + \text{II} + \text{III},
\end{aligned}
$$

where

$$
\begin{aligned}
\text{I} &= \max_i \sum_{j=1}^n |\hat{c}_{ij} - c_{ij}| I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| > \tau_{nJ}), \\
\text{II} &= \max_i \sum_{j=1}^n |\hat{c}_{ij}| I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| \le \tau_{nJ}), \\
\text{III} &= \max_i \sum_{J} |c_{ij}| I(|c_{ij}| > \tau_{nJ}, |\hat{c}_{ij}| \le \tau_{nJ}).
\end{aligned}
$$

We bound the above three terms as follows.

$$
\begin{aligned}
\text{I} &\le \max_{i,j} |\hat{c}_{ij} - c_{ij}| \max_i \sum_{j=1}^n I(|c_{ij}| > 0) \\
&= O_p\left(m_n \sqrt{\log(n)/J}\right). \qquad (7.11)
\end{aligned}
$$

For $\delta > 0$, using the equality in (i), we have

$$
\begin{aligned}
\text{II} &\le \max_i \sum_{j=1}^n |\hat{c}_{ij} - c_{ij}| I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| \le \tau_{nJ}) \\
&\quad + \max_i \sum_{j=1}^n |c_{ij}| I(|c_{ij}| \le \tau_{nJ}) \\
&\le \max_i \sum_{j=1}^n |\hat{c}_{ij} - c_{ij}| I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| \le \delta\tau_{nJ}) \\
&\quad + \max_i \sum_{j=1}^n |\hat{c}_{ij} - c_{ij}| I(|\hat{c}_{ij}| > \tau_{nJ}, \delta\tau_{nJ} < |c_{ij}| < \tau_{nJ}) + \tau_{nJ} m_n \\
&\le \max_{i,j} |\hat{c}_{ij} - c_{ij}| \left(\max_i \sum_{j=1}^n I(|\hat{c}_{ij}| > \tau_{nJ}, |c_{ij}| \le \delta\tau_{nJ}) + m_n \right) + \tau_{nJ} m_n \\
&\le O_p(\sqrt{\log(n)/J}) \left(\max_i \sum_{j=1}^n I(|\hat{c}_{ij} - c_{ij}| \ge (1-\delta)\tau_{nJ}) + m_n \right) + \tau_{nJ} m_n \\
&= O_p(\sqrt{\log(n)/J})(o_p(1) + m_n) + \tau_{nJ} m_n = O_p(\tau_{nJ} m_n), \qquad (7.12)
\end{aligned}
$$

since

$$
P\left(\max_i \sum_{j=1}^n I(|\hat{c}_{ij} - c_{ij}| \ge (1-\delta)\tau_{nJ}) > \epsilon \right) \le P\left(\max_{i,j} |\hat{c}_{ij} - c_{i,j}| \ge (1-\delta)\tau_{nJ} \right)
$$

$$= \quad o(1).$$

Similarly,

$$\begin{aligned}
\text{III} &\leq \max_i \sum_{j=1}^{n} \left(|\hat{c}_{ij} - c_{ij}| + |\hat{c}_{ij}||\right) I(|c_{ij}| > \tau_{nJ}, |\hat{c}_{ij}| \leq \tau_{nJ}) \\
&\leq \max_{i,j} |\hat{c}_{ij} - c_{ij}| \sum_{j=1}^{n} I(|c_{ij}| > \tau_{nJ}) + \tau_{nJ} \max_i \sum_{j=1}^{n} I(|c_{ij}| > \tau_{nJ}) \\
&\leq O_p(\tau_{nJ}) m_n + \tau_{nJ} m_n = O_p(\tau_{nJ} m_n).
\end{aligned}$$

Combining this with (7.11), (7.12) and (7.10), we obtain the desired result in (ii). The proof is completed.

Lemma 12 *Under Conditions (A1') and (A3)∼(A4), if* $\tau_{nJ} = O(\sqrt{\log(n)/J})$ *and* $n\tau_{nJ} = o(1)$ *as* $n \to \infty$ *and* $J \to \infty$, *then*

(i) $||\hat{C}(\tau_{nJ})^{-1} - C^{-1}|| = O_p(m_n \tau_{nJ})$ *and* $||\hat{C}(\tau_{nJ})^{-2} - C^{-2}|| = O_p(m_n \tau_{nJ})$,

(ii) $||\hat{C}(0)^{-1} - C^{-1}|| \leq O_p(\tau_{nJ}(m_n + n))$; $||\hat{C}(0)^{-2} - C^{-2}|| \leq O_p(\tau_{nJ}(m_n + n))$,

where $m_n = \max_{1 \leq i \leq n} \sum_{j=1}^{n} I(c_{ij} \neq 0) \leq n$.

Proof. Let $\kappa_3 = \max\{2(2/\kappa_1 + 1/\kappa_2) - 1, (4/3)(1/\kappa_1 + 1/\kappa_2) - 1/3, 1\}$. Then $n\sqrt{\log(n)/J}$ $= o(1)$ yields $(\log(n))^{\kappa_3}/J = o(1)$. If let $\lambda_{\min}(C)$ denote the minimum eigenvalue of C, then we have that $\lambda_{\min}(C) \geq \sigma_0^2$. If let $\lambda_{\min}(\hat{C}(\tau_{nJ}))$ denote the minimum eigenvalue of $\hat{C}(\tau_{nJ})$, then it follows from Lemma 11 that

$$\begin{aligned}
\lambda_{\min}(\hat{C}(\tau_{nJ})) &= \lambda_{\min}(C) + O_p(m_n \tau_{nJ}) \\
&\geq \sigma_0^2 + O_p(m_n \tau_{nJ}),
\end{aligned}$$

which is bounded below by $\sigma_0^2/2$ if $\tau_{nJ} m_n$ is small enough. Therefore, we have

$$\begin{aligned}
||\hat{C}(\tau_{nJ})^{-1} - C^{-1}|| &= ||\hat{C}(\tau_{nJ})^{-1}(C - \hat{C}(\tau_{nJ}))C^{-1}|| \\
&\leq ||\hat{C}(\tau_{nJ})^{-1}||(||C - \hat{C}(\tau_{nJ})||)||C^{-1}|| \\
&\leq \lambda_{\min}(\hat{C}(\tau_{nJ}))^{-1} \lambda_{\min}(C)^{-1} ||C - \hat{C}(\tau_{nJ})|| = O_p(\tau_{nJ} m_n). \\
||\hat{C}(\tau_{nJ})^{-2} - C^{-2}|| &\leq ||\hat{C}(\tau_{nJ})^{-1}(\hat{C}(\tau_{nJ})^{-1} - C^{-1})|| + ||(\hat{C}(\tau_{nJ})^{-1} - C^{-1})C^{-1}|| \\
&\leq ||\hat{C}(\tau_{nJ})^{-1}||||\hat{C}(\tau_{nJ})^{-1} - C^{-1}|| + ||\hat{C}(\tau_{nJ})^{-1} - C^{-1}||||C^{-1}|| \\
&= (\lambda_{\min}(\hat{C}(\tau_{nJ}))^{-1} + \lambda_{\min}(C)^{-1})||\hat{C}(\tau_{nJ})^{-1} - C^{-1}|| \\
&= O_p(\tau_{nJ} m_n). \\
||\hat{C}(0)^{-1} - C^{-1}|| &\leq O_p(\tau_{nJ}(m_n + n)), \\
||\hat{C}(0)^{-2} - C^{-2}|| &\leq O_p(\tau_{nJ}(m_n + n)).
\end{aligned}$$

Proof of Theorem 6. Note that for any x, $H_x^T H_x = n$. We have

$$\begin{aligned}
&|| \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right]^{-1} - \left[H_j^T C^{-1} H_j\right]^{-1} || \\
&= || \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right]^{-1} \left(H_j^T C^{-1} H_j - H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right) \left[H_j^T C^{-1} H_j\right]^{-1} || \\
&\leq \frac{1}{n} || \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j/n\right]^{-1} ||||H_j^T C^{-1} H_j/n - H_j^T \hat{C}(\tau_{nJ})^{-1} H_j/n|| \\
&\quad \times || \left[H_j^T C^{-1} H_j/n\right]^{-1} || \\
&\leq \frac{1}{n} || \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j/n\right]^{-1} ||||C^{-1} - \hat{C}(\tau_{nJ})^{-1}|||| \left[H_j^T C^{-1} H_j/n\right]^{-1} ||,
\end{aligned}$$

which combining with Lemma 11 yields

$$\begin{aligned}
&|| \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right]^{-1} - \left[H_j^T C^{-1} H_j\right]^{-1} || \\
&\leq O_p(n^2 \sqrt{\log(n)/J}) || \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right]^{-1} |||| \left[H_j^T C^{-1} H_j\right]^{-1} ||. \qquad (7.13)
\end{aligned}$$

Let λ_m and $\hat{\lambda}_m$ denote the smallest eigenvalues of $H_j^T C^{-1} H_j$ and $H_j^T \hat{C}(\tau_{nJ})^{-1} H_j$ respectively. Invoking Theorem 1, $(H_j^T C^{-1} H_j)^{-1} = \Sigma_j + o(1)$. There exists a positive constant ϵ_0 such that for large n, $\lambda_m \geq \epsilon_0$. By the definition, there exists $a_m \in \mathbb{R}^3$ with $||a_m|| = 1$, such that $\hat{\lambda}_m = a_m^T H_j^T \hat{C}(\tau_{nJ})^{-1} H_j a_m$. So

$$\begin{aligned}
|\hat{\lambda}_m - a_m^T H_j^T C^{-1} H_j a_m| &= |(H_i a_m)^T (\hat{C}^{-1} - C^{-1}) H_i a_m| \leq n||\hat{C}(\tau_{nJ})^{-1} - C^{-1}|| \\
&\leq O_p(n^2 \sqrt{\log(n)/J}),
\end{aligned}$$

which implies

$$\begin{aligned}
\hat{\lambda}_m &\geq a_m^T H_j^T C^{-1} H_j a_m - O_p(n^2 \sqrt{\log(n)/J}) \\
&\geq \lambda_m - O_p(n^2 \sqrt{\log(n)/J}) \geq \epsilon_0 - O_p(n^2 \sqrt{\log(n)/J}).
\end{aligned}$$

This shows that for large n, $\hat{\lambda}_m$ is bounded below from zero. Consequently, we have

$$|| \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right]^{-1} || = O(1), \quad || \left[H_j^T C^{-1} H_j\right]^{-1} || = O(1).$$

This together with (7.13) proves that

$$|| \left[H_j^T \hat{C}(\tau_{nJ})^{-1} H_j\right]^{-1} - \left[H_j^T C^{-1} H_j\right]^{-1} || = O_p(n^2 \sqrt{\log(n)/J}).$$

The proof is completed.

Proof of Corollary 7. It follows from Theorem 6 directly. The details are omitted.

Acknowledgments

We thank Professor Richard Henson from MRC Cognition and Brain Sciences Unit, Cambridge for sharing his MEG data with us. The software SPM8 is available at `http://www.fil.ion.ucl.ac.uk/spm/software/spm8/`. We also thank the Editor and three anonymous reviewers for their helpful comments.

References

P. J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, pages 2577–2604, 2008.

A. Bolstad, B. Van Veen, and R. Nowak. Space–time event sparse penalization for magneto-/electroencephalography. *NeuroImage*, 46(4):1066–1081, 2009.

M. J. Brookes, J. Vrba, S. E. Robinson, C. M. Stevenson, A. M. Peters, G. R. Barnes, A. Hillebrand, and P. G. Morris. Optimising experimental design for meg beamformer imaging. *Neuroimage*, 39(4):1788–1802, 2008.

T. Cai and W. Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.

J. Dedecker and C. Prieur. Coupling for τ-dependent sequences and applications. *Journal of Theoretical Probability*, 17(4):861–885, 2004.

J. Fan, Y. Liao, and M. Mincheva. High dimensional covariance matrix estimation in approximate factor models. *Annals of statistics*, 39(6):3320, 2011.

K. Friston, R. Henson, C. Phillips, and J. Mattout. Bayesian estimation of evoked and induced responses. *Human brain mapping*, 27(9):722–735, 2006.

J. H. Goodnight. A tutorial on the sweep operator. *The American Statistician*, 33(3): 149–158, 1979.

M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalographytheory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, 65(2):413, 1993.

R. N. Henson, D. G. Wakeman, V. Litvak, and K. J. Friston. A parametric empirical bayesian framework for the eeg/meg inverse problem: generative models for multi-subject and multi-modal integration. *Frontiers in human neuroscience*, 5, 2011.

A. Hillebrand, K. D. Singh, I. E. Holliday, P. L. Furlong, and G. R. Barnes. A new approach to neuroimaging with magnetoencephalography. *Human brain mapping*, 25(2):199–211, 2005.

M. X. Huang, J. J. Shih, R. R. Lee, D. L. Harrington, R. J. Thoma, M. P. Weisend, F. Hanlon, K. M. Paulson, T. Li, K. Martin, et al. Commonalities and differences among vectorized beamformers in electromagnetic source imaging. *Brain topography*, 16(3):139–158, 2004.

N. Kanwisher, J. McDermott, and M. M. Chun. The fusiform face area: a module in human extrastriate cortex specialized for face perception. *The Journal of Neuroscience*, 17(11): 4302–4311, 1997.

O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of multivariate analysis*, 88(2):365–411, 2004.

F. Merlevède, M. Peligrad, and E. Rio. A bernstein type inequality and moderate deviations for weakly dependent sequences. *Probability Theory and Related Fields*, 151(3-4):435–474, 2011.

J. C. Mosher, R. M. Leahy, and P. S. Lewis. Eeg and meg: forward solutions for inverse methods. *Biomedical Engineering, IEEE Transactions on*, 46(3):245–259, 1999.

R. Oostenveld, P. Fries, E. Maris, and J. Schoffelen. Fieldtrip: open source software for advanced analysis of meg, eeg, and invasive electrophysiological data. *Computational intelligence and neuroscience*, 2011, 2010.

D. Pitcher, D. D. Dilks, R. R. Saxe, C. Triantafyllou, and N. Kanwisher. Differential selectivity for dynamic versus static information in face-selective cortical regions. *Neuroimage*, 56(4):2356–2363, 2011.

M. A. Quraan, S. N. Moses, Y. Hung, T. Mills, and M. J. Taylor. Detection and localization of hippocampal activity using beamformers with meg: a detailed investigation using simulations and empirical data. *Human brain mapping*, 32(5):812–827, 2011.

J. O. Ramsay. *Functional data analysis*. Wiley Online Library, 2006.

S. E. Robinson. Functional neuroimaging by synthetic aperture magnetometry (sam). *Recent advances in biomagnetism*, pages 302–305, 1999.

A. Rodríguez-Rivera, B. V. Baryshnikov, B. D. Van Veen, and R. T. Wakai. Meg and eeg source localization in beamspace. *Biomedical Engineering, IEEE Transactions on*, 53(3): 430–441, 2006.

A. J. Rothman, E. Levina, and J. Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104(485):177–186, 2009.

J. Sarvas. Basic mathematical and electromagnetic concepts of the biomagnetic inverse problem. *Physics in medicine and biology*, 32(1):11, 1987.

K. Sekihara and S. S. Nagarajan. *Adaptive spatial filters for electromagnetic brain imaging*. Springer Science & Business Media, 2008.

K. Sekihara, S. S. Nagarajan, D. Poeppel, and A. Marantz. Asymptotic snr of scalar and vector minimum-variance beamformers for neuromagnetic source reconstruction. *Biomedical Engineering, IEEE Transactions on*, 51(10):1726–1734, 2004.

B. D. Van Veen, W. Van Drongelen, M. Yuchtman, and A. Suzuki. Localization of brain electrical activity via linearly constrained minimum variance spatial filtering. *Biomedical Engineering, IEEE Transactions on*, 44(9):867–880, 1997.

J. Zhang, C. Liu, and G. Green. Source localization with meg data: A beamforming approach based on covariance thresholding. *Biometrics*, 70(1):121–131, 2014.

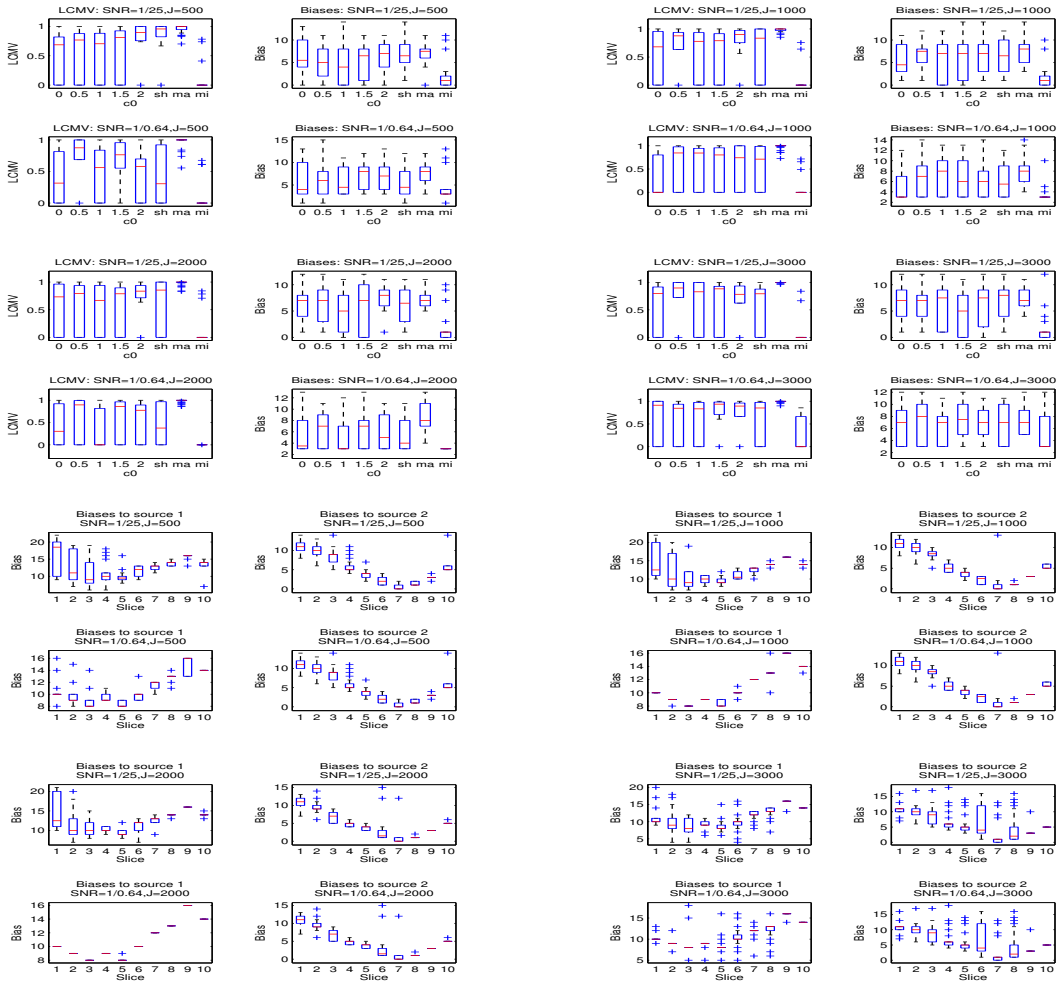


Figure 2: Scenario 1: Two sources located at CTF coordinates $(3, -1, 4)^T$ cm and $(-5, 2, 6)^T$ cm respectively. The first four rows display the box-and-whisker plots of the index values and the localization biases against the tuning constant $c_0 = 0, 0.5, 1, 1.5, 2$, **ma**, **mi** and **sh** for the combinations of $n = 91$, SNR$= 1/25, 1/0.64$, and $J = 500, 1000, 2000, 3000$ respectively. Here, **ma** and **mi** stand for the proposed hard-thresholded covariance based methods. **sh** stands for the optimal shrinkage-based method. With a slightly abuse of notation, $c_0 =$ **ma**, **mi**, **sh** refer to that **ma**, **mi**, and **sh** are used. The remaining rows present the box-and-whisker plots of the local localization bias to the sources r_1 and r_2 against the transverse slice indices from 0 to 10 when c_0 was selected by the minimum strategy for the above combinations respectively. The red colored lines in the boxes are the medians. Note that when the distribution of the localization biases are degenerate, the upper and lower quartiles and medians of localization biases will be equal. Consequently, the box in the plot will reduce to a red colored line. The plots in the last four rows show that all the local peaks on the transverse slices are not close to the source location r_1, implying that the source 1 has been masked on the neuronal activity map by source cancellations.

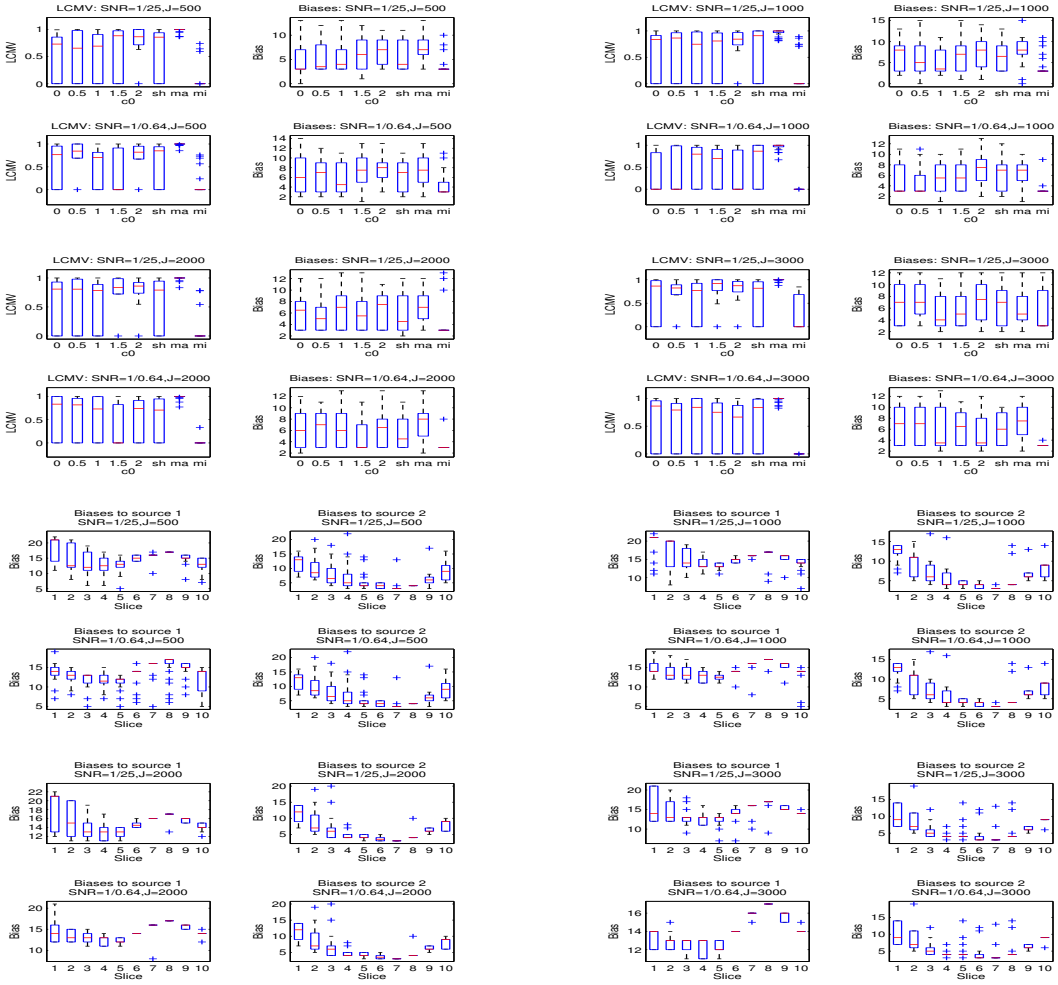


Figure 3: Scenario 2: Two sources located at CTF coordinates $(-5, 5, 6)^T$ cm and $(-6, -2, 5)^T$ cm respectively. The first four rows display the box-and-whisker plots of the index values and the localization biases against the tuning constant $c_0 = 0, 0.5, 1, 1.5, 2$, **ma**, **mi** and **sh** for the combinations of $n = 91$, SNR$= 1/25, 1/0.64$, and $J = 500, 1000, 2000, 3000$ respectively. The remaining rows present the box-and-whisker plots of the minimum local localization bias to the sources r_1 and r_2 against the transverse slice indices from 0 to 10 when c_0 is selected by the minimum strategy for the above combinations respectively. The red colored lines in the boxes are the medians. When the upper and lower quartiles and medians of localization biases have the same value, the box in the plot will reduce to a red colored line. The plots in the last four rows show that all the local peaks on the transverse slices are not close to the source location r_1, implying the source 1 has been masked on the neuronal activity map by source cancellations.

Figure 4: Scenario 3: Two sources located at CTF coordinates $(3, -1, 4)^T$ cm and $(-5, 2, 6)^T$ cm respectively. The first six rows show the box-and-whisker plots of the index values and the localization biases against the tuning constant $c_0 = 0, 0.5, 1, 1.5, 2$, **ma**, **mi** and **sh** for the combinations of $n = 102$ sensors, SNR$= 1/0.35^2, 1/0.4^2, 1/0.5^2$, and the sample rates $J = 500, 1000, 2000, 3000$ respectively. The last six rows give the box-and-whisker plots of the minimum local localization bias to the sources r_1 and r_2 against the transverse slice indices from 0 to 10 when c_0 is selected by the minimum strategy for these combinations respectively. The red colored lines in the boxes are the medians. When the upper and lower quartiles and medians of localization biases are equal, the box in the plot will reduce to a red colored line. The last six rows of the plots show all the local peaks on the transverse slices are not close to the source location r_1, implying the source 1 has been masked on the neuronal activity map by source cancellations.

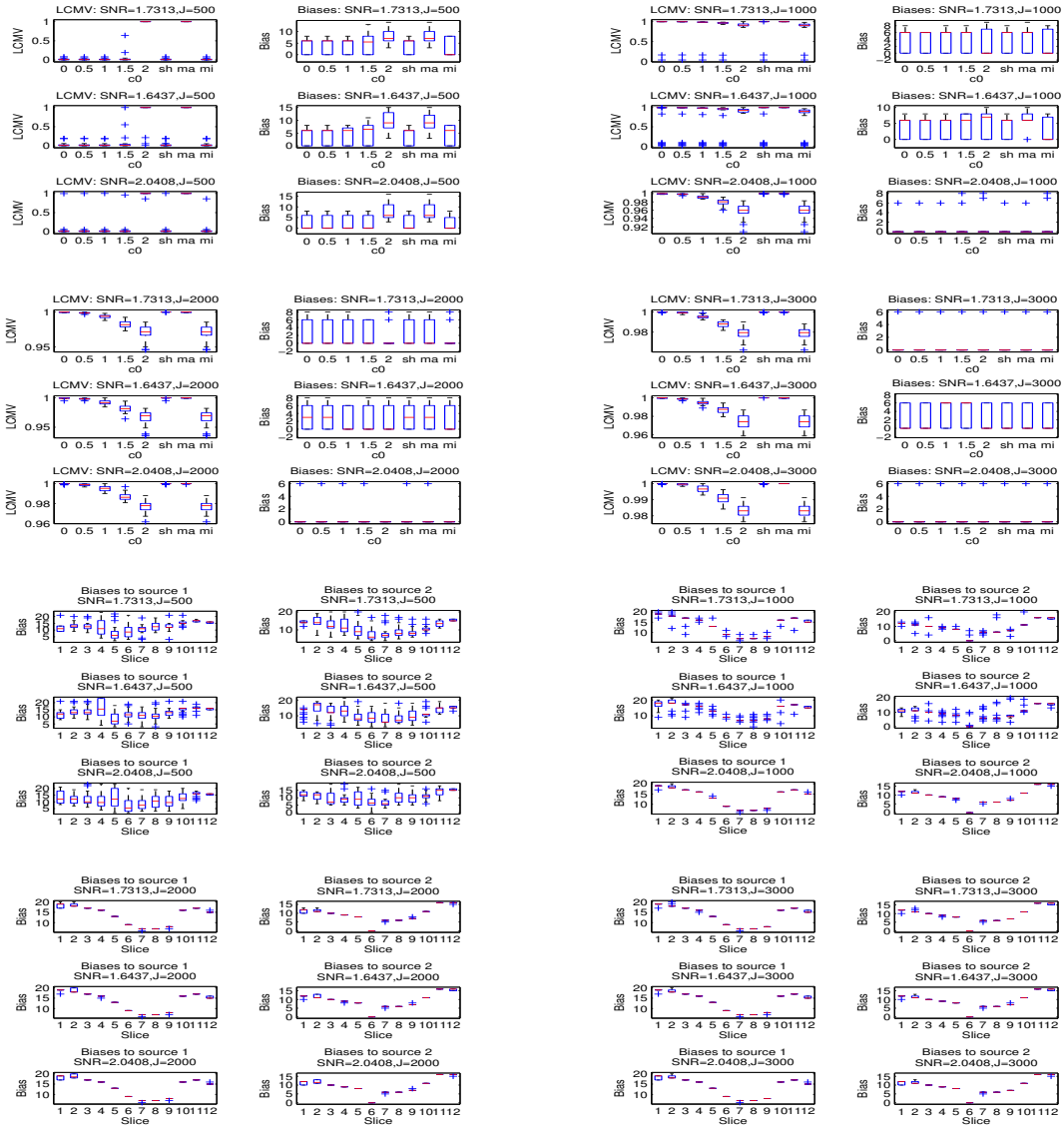


Figure 5: Scenario 4: Two sources located at CTF coordinates $(-5, 5, 6)^T$ cm and $(-6, -2, 5)^T$ cm respectively. The first six rows show the box-and-whisker plots of the index values and the localization biases against the tuning constant $c_0 = 0, 0.5, 1, 1.5, 2$, **ma**, **mi** and **sh** for the combinations of $n = 102$ sensors, SNR$= 1/0.35^2, 1/0.4^2, 1/0.5^2$, and the sample rates $J = 500, 1000, 2000, 3000$ respectively. The last six rows give the box-and-whisker plots of the minimum local localization bias to the sources r_1 and r_2 against the transverse slice indices from 0 to 10 when c_0 was selected by the minimum strategy for these combinations respectively. The red colored lines in the boxes are the medians. When the upper and lower quartiles and medians of localization biases are equal, the box in the plot will reduce to a red colored line. The last six rows of the plots show all the local peaks on the transverse slices are not close to the source location r_1, implying the source 1 has been masked on the neuronal activity map by source cancellations.

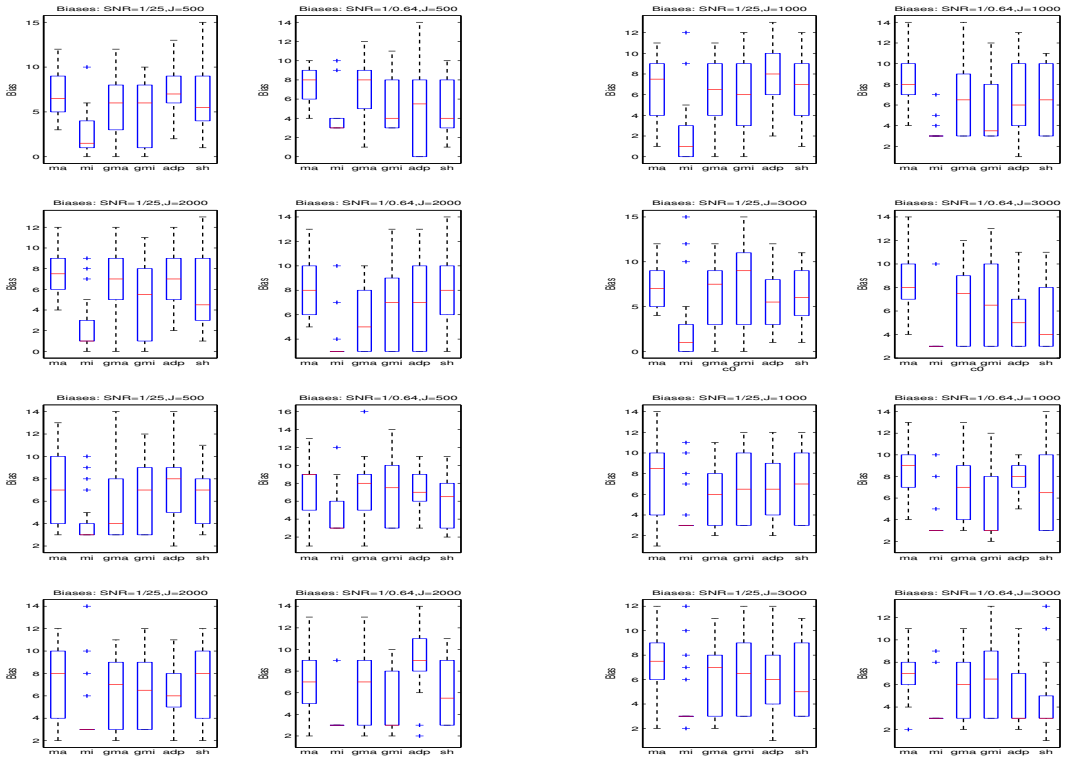


Figure 6: Performance comparison of the six different beamformers, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh** in Scenarios 1 and 2. Here, **ma** and **mi** stand for the hard-thresholded covariance based methods when the tuning constant c_0 is chosen by use of the maximum strategy and the minimum strategy respectively; **gma** and **gmi** stand for the generalized thresholded covariance based methods when the tuning constant c_0 is chosen by use of the maximum strategy and the minimum strategy respectively; **adp** and **sh** stand for the adaptive thresholding-based method and the optimal shrinkage-based method. The upper two rows of multiple box-whisker plots are for the combinations of $n = 91$, SNR$= 1/25, 1/0.64$, and $J = 500, 1000, 2000, 3000$ in Scenario 1, while the lower two rows are for the combinations of $n = 91$, SNR$= 1/25, 1/0.64$, and $J = 500, 1000, 2000, 3000$ in Scenario 2. Each panel shows the localization biases against the above six different beamformer methods. The red colored lines in the boxes are the medians. When the upper and lower quartiles and medians of localization biases are equal, the box in the plot will reduce to a red colored line.

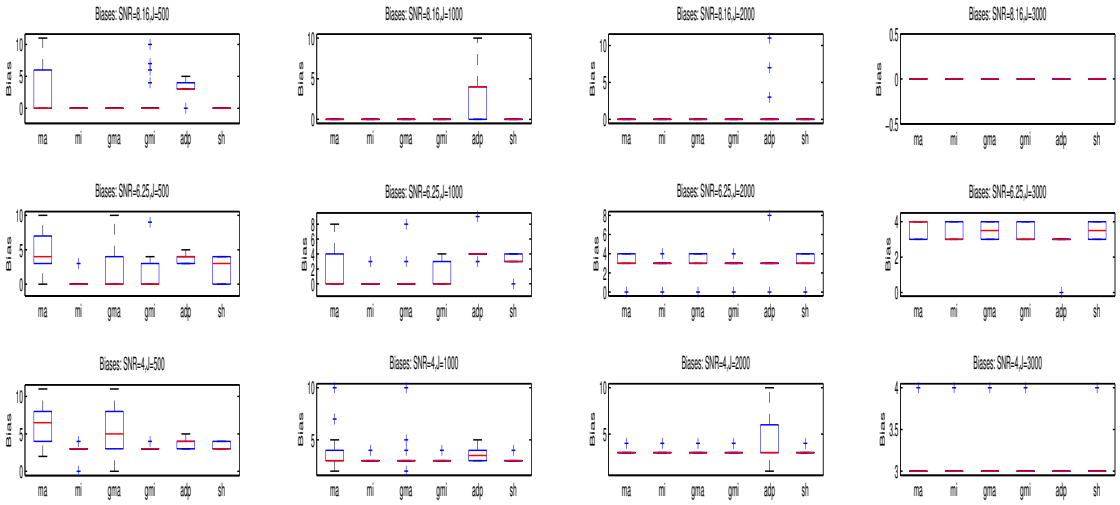


Figure 7: Performance comparison of the six different beamformers, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh** in Scenario 3. Multiple box-whisker plots of localization biases are displayed for the combinations of $n = 102$, SNR$= 1/0.35^2$, $1/0.4^2$, $1/0.5^2$, and $J = 500, 1000, 2000, 3000$. Each panel shows the localization biases against the six different beamformer methods, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh**. The red colored lines in the boxes are the medians. When the upper and lower quartiles and medians of localization biases are equal, the box in the plot will reduce to a red colored line.

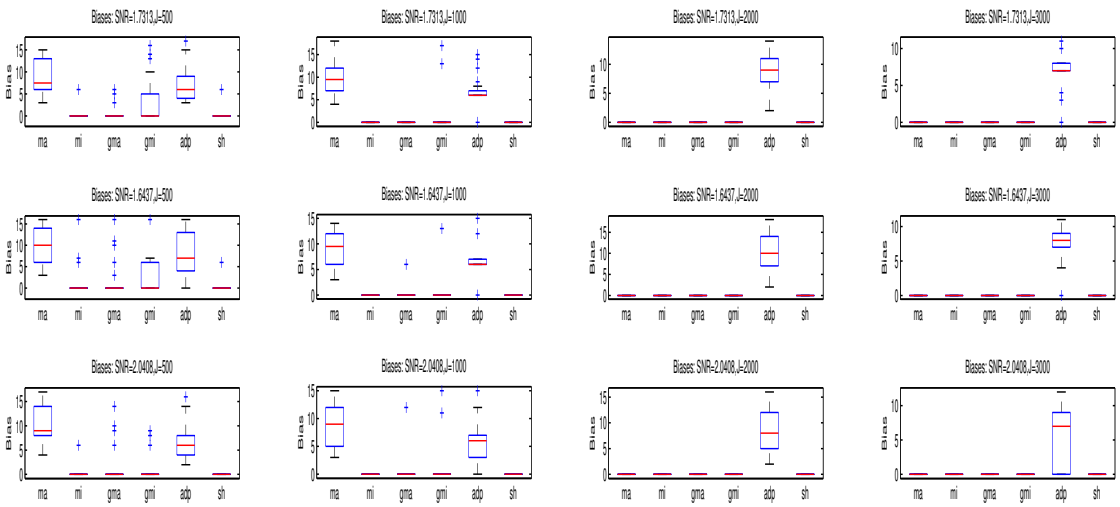


Figure 8: Performance comparison of the six different beamformers, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh** in Scenario 4. Multiple box-whisker plots of localization biases are displayed for the combinations of $n = 102$, SNR$= 1/0.35^2$, $1/0.4^2$, $1/0.5^2$, and $J = 500, 1000, 2000, 3000$ in Scenario 4. Each panel shows the localization biases against the six different beamformer methods, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh**. The red colored lines in the boxes are the medians. When the upper and lower quartiles and medians of localization biases are equal, the box in the plot will reduce to a red colored line.

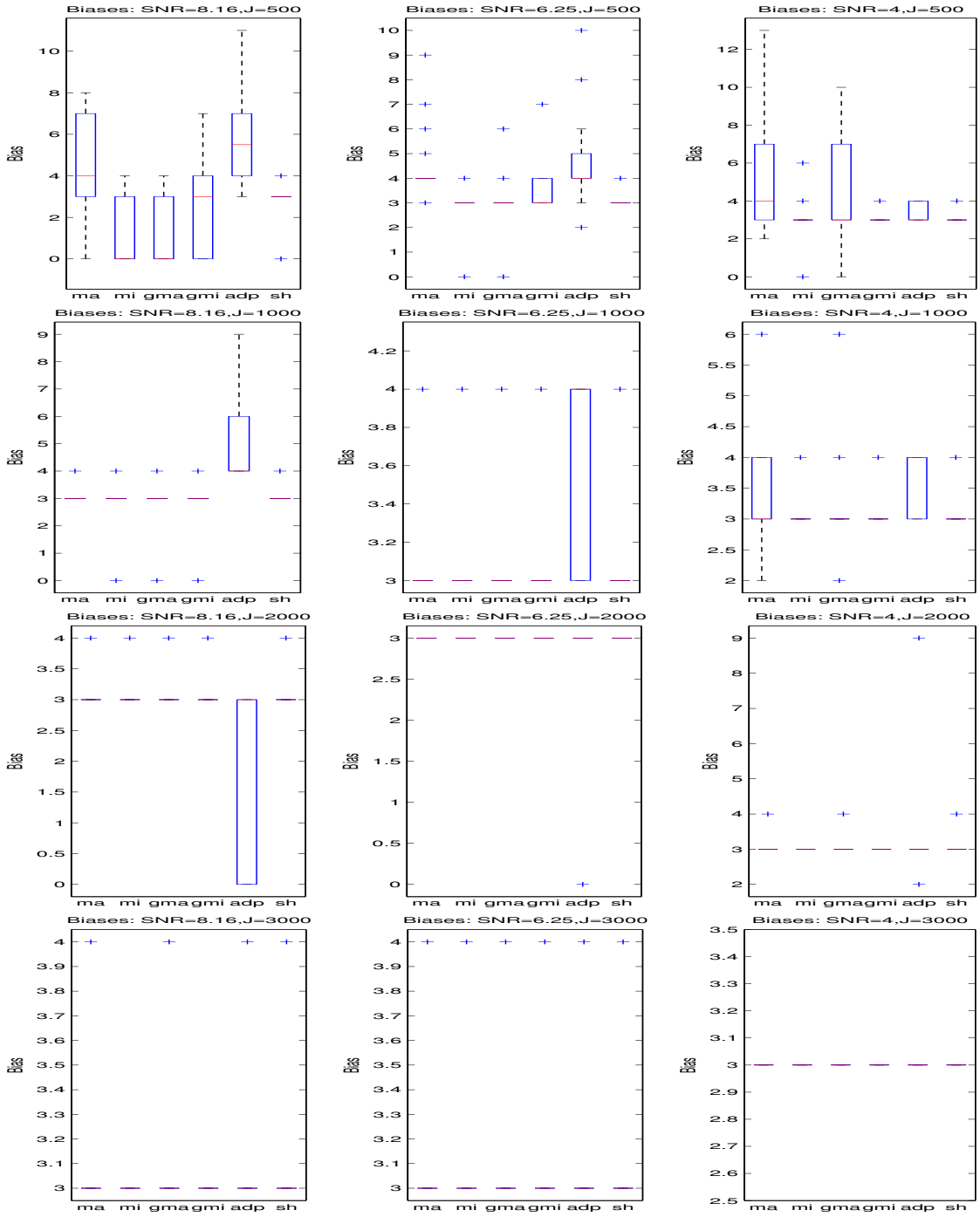


Figure 9: Performance comparison of the six different beamformers, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh** in Scenario 5. Multiple box-whisker plots of localization biases are displayed for the combinations of $n = 102$, SNR$= 1/0.35^2$, $1/0.4^2$, $1/0.5^2$, and $J = 500, 1000, 2000, 3000$ respectively. Each panel shows the localization biases against the six different beamformer methods, namely **ma**, **mi**, **gma**, **gmi**, **adp** and **sh**.

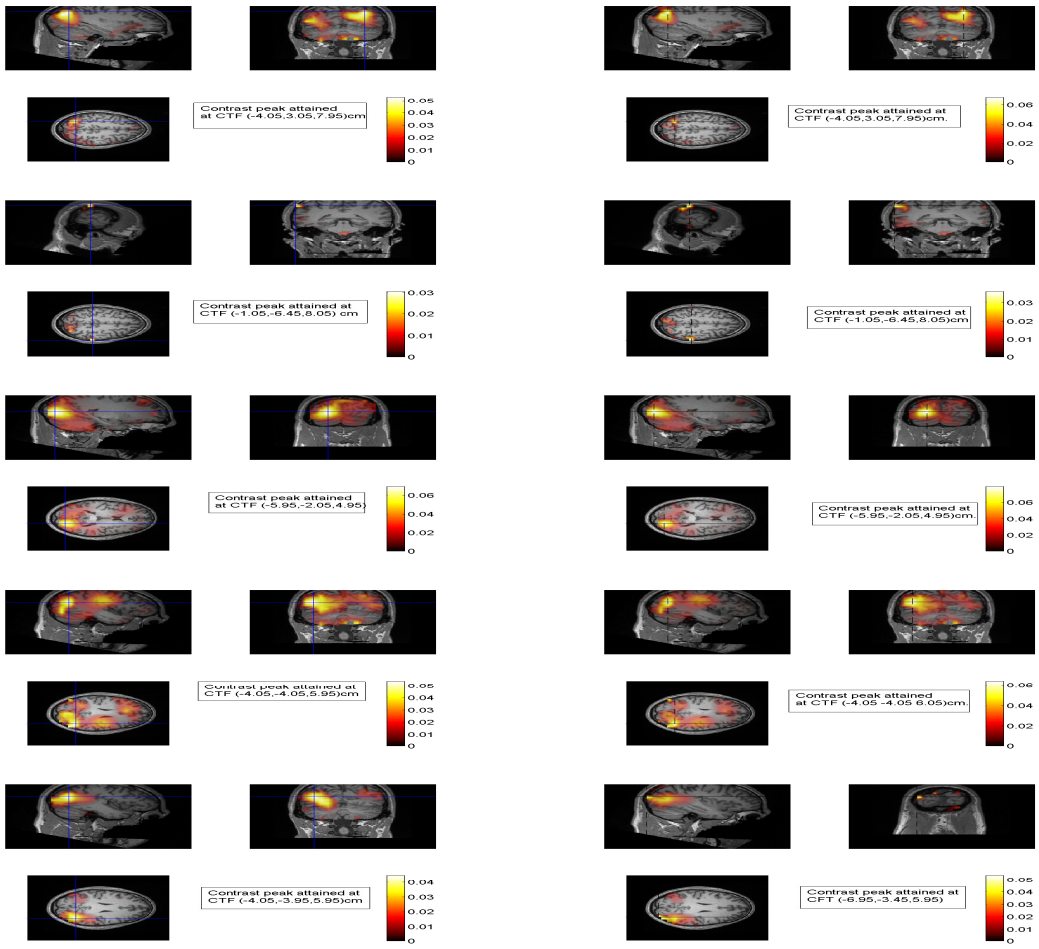


Figure 10: Plots of the log-contrasts between the faces and scrambled faces on three orthogonal slices through the peak locations for each of five sessions, which are overlaid on the subject's MRI scan. The plots in the left-hand two columns and the right-hand two columns are derived from the procedures **mi** and **adp** respectively. Rows 1 and 2, 3 and 4, 5 and 6, and 7 and 8 are for sessions $1 \sim 5$ respectively. The highlighted yellow colored areas revealed neuronal activity increases or decreases for the faces relative to the scrambled faces. The areas shown in the left-hand two columns are in or close to the IOG, STS, and PCu regions which are known to be related to the human face perception.

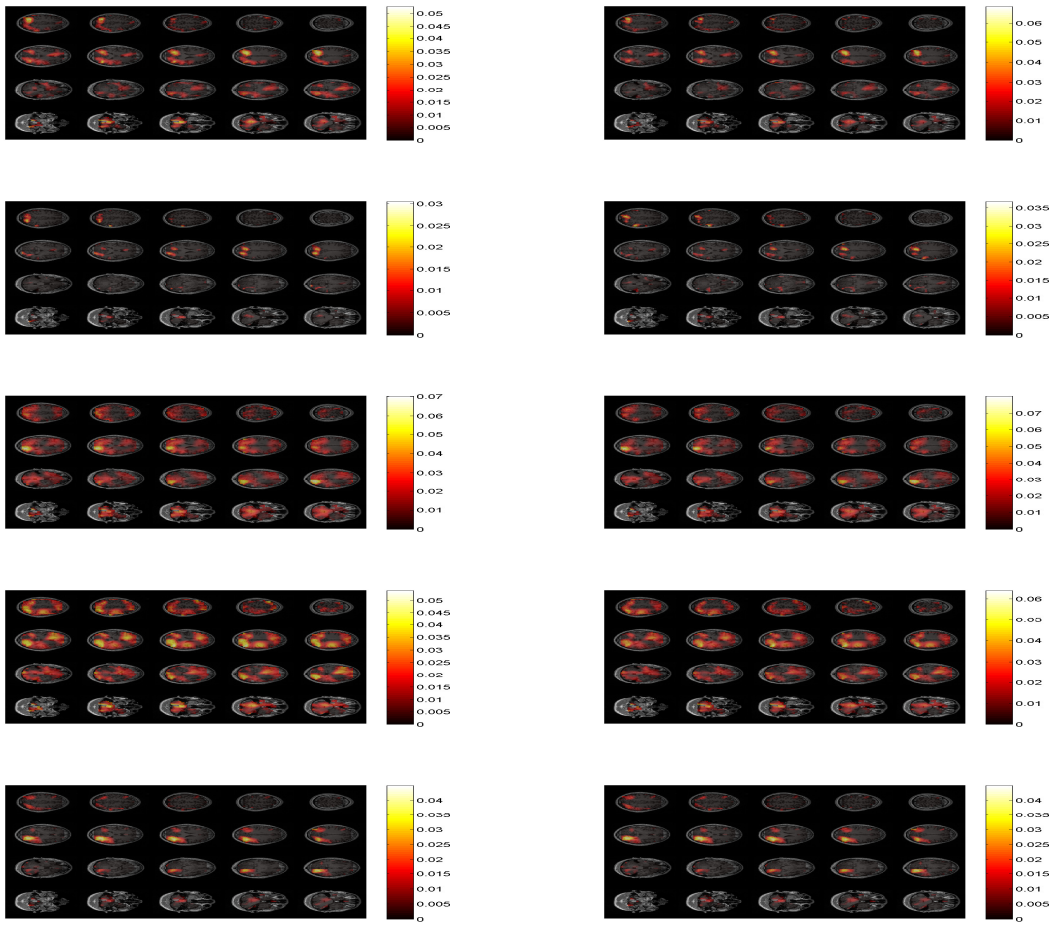


Figure 11: Plots of the log-contrasts between the faces and scrambled faces on 20 transverse slices for each of five sessions, which are overlaid on the subject's MRI scan. The plots in the left-hand column and the right-hand column are derived from the procedures **mi** and **adp** respectively. Rows $1 \sim 5$ are for sessions $1 \sim 5$ respectively. The highlighted yellow colored areas revealed neuronal activity increases for the faces relative to the scrambled faces. The areas highlighted in the first column are in or close to the OFA, IOG, STS, and PCu regions which are known to be related to the human face perception.

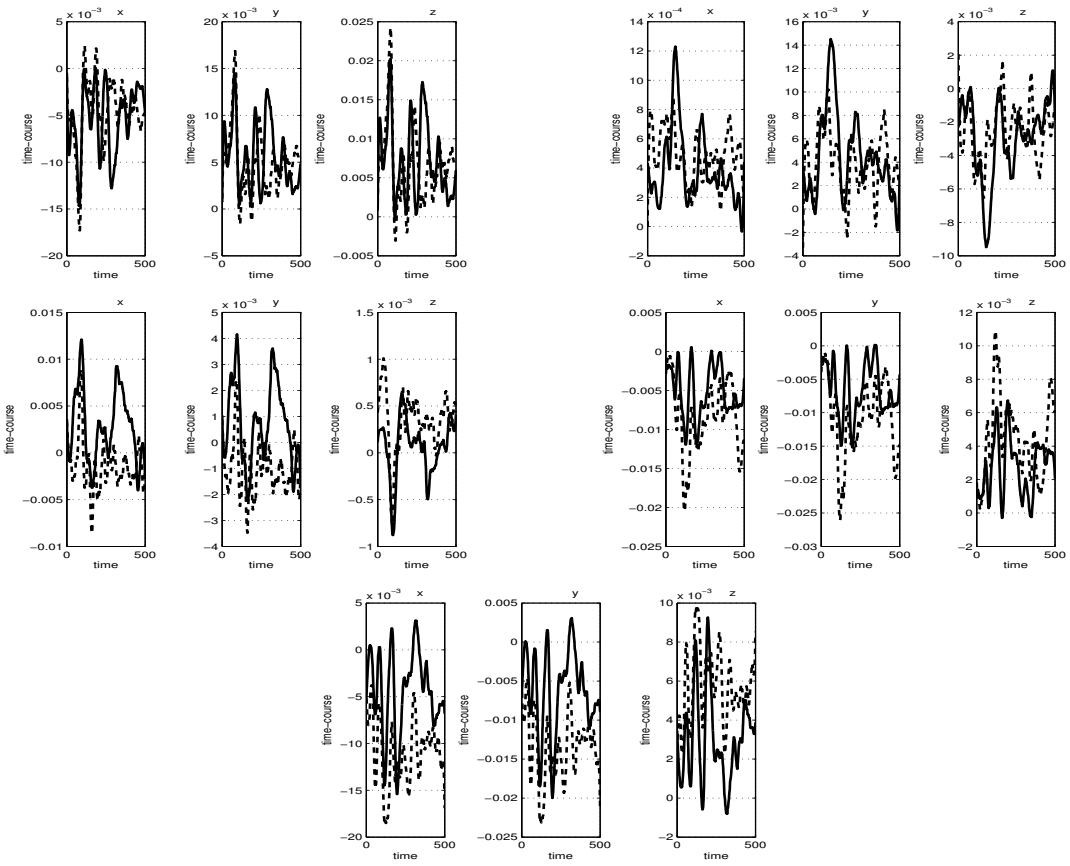


Figure 12: Plots of the estimated time-courses at the global peaks along x, y and z-axes respectively for each of five sessions. The solid curve and the dashed curve in each plot stand for the estimated time-courses under the faces and the scrambled faces respectively. The plots are ordered from the top left panel to the right panel to the bottom panel corresponding to sessions $1 \sim 5$.

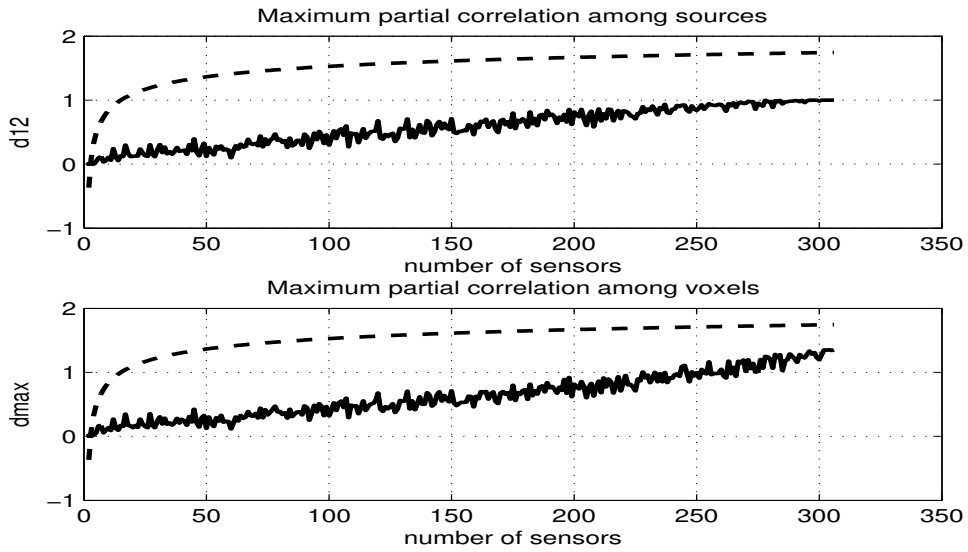


Figure 13: Plots of $d_{12}(k)$ and $d_{\max}(k)$ against $k = 1, 2, ..., 306$ respectively, where k stands for k randomly chosen sensors from the 306 sensors in the face-perception data, two sources are located at CTF (-4, 3,8)cm and (-4,-5,5) cm respectively, and the dashed curve in each plot is for the function $\log(\log(k))$.

Journal of Machine Learning Research 16 (2015) 2147-2205 Submitted 10/13; Revised 10/14; Published 10/15

Constraint-based Causal Discovery from Multiple Interventions over Overlapping Variable Sets

Sofia Triantafillou* STRIANT@ICS.FORTH.GR
Ioannis Tsamardinos* TSAMARD@ICS.FORTH.GR
Institute of Computer Science
Foundation for Research and Technology - Hellas (FORTH)
N. Plastira 100 Vassilika Vouton
GR-700 13 Heraklion, Crete, Greece

Editor: Christopher Meek

Abstract

Scientific practice typically involves repeatedly studying a system, each time trying to unravel a different perspective. In each study, the scientist may take measurements under different experimental conditions (interventions, manipulations, perturbations) and measure different sets of quantities (variables). The result is a collection of heterogeneous data sets coming from different data distributions. In this work, we present algorithm COmbINE, which accepts a collection of data sets over overlapping variable sets under different experimental conditions; COmbINE then outputs a summary of all causal models indicating the invariant and variant structural characteristics of all models that simultaneously fit all of the input data sets. COmbINE converts estimated dependencies and independencies in the data into path constraints on the data-generating causal model and encodes them as a SAT instance. The algorithm is sound and complete in the sample limit. To account for conflicting constraints arising from statistical errors, we introduce a general method for sorting constraints in order of confidence, computed as a function of their corresponding p-values. In our empirical evaluation, COmbINE outperforms in terms of efficiency the only pre-existing similar algorithm; the latter additionally admits feedback cycles, but does not admit conflicting constraints which hinders the applicability on real data. As a proof-of-concept, COmbINE is employed to co-analyze 4 real, mass-cytometry data sets measuring phosphorylated protein concentrations of overlapping protein sets under 3 different interventions.

Keywords: causality, causal discovery, graphical models, maximal ancestral graphs, semi-Markov causal models, randomized experiments, latent variables

1. Introduction

Causal discovery is an abiding goal in almost every scientific field. In order to discover the causal mechanisms of a system, scientists typically have to perform a series of experiments (interchangeably: manipulations, interventions, or perturbations). Each experiment adds to the existing knowledge of the system and sheds light to the sought-after mechanism from a different perspective. In addition, each measurement may include a different set of

∗. Also in Department of Computer Science, University of Crete.

quantities (variables), when for example the technology used allows only a limited number of measured quantities.

However, for the most part, machine learning and statistical methods focus on analyzing a single data set. They are unable to make joint inferences from the complete collection of available heterogeneous data sets, since each one is following a different data distribution (albeit stemming from the same system under study). Thus, data sets are often analyzed in isolation and independently of each other; the resulting knowledge is typically synthesized ad hoc in the researcher's mind.

The proposed work tries to automate the above inferences. We propose a general, constraint-based algorithm named COmbINE for learning causal structure characteristics from the integrative analysis of collections of data sets. The data sets can be heterogeneous in the following manners: they may be measuring different overlapping sets of variables $\mathbf{O}_i$ under different hard manipulations on a set of observed variables $\mathbf{I}_i$. A hard manipulation on a variable I, corresponds to a Randomized Controlled Trial (Fisher, 1935) where the experimentation procedure completely eliminates any other causal effect on I (e.g., randomizing mice to two groups having two different diets; the effect of all other factors on the diet is completely eliminated).

What connects together the available data sets and allows their co-analysis is the assumption that *there exists a single underlying causal mechanism that generates the data*, even though it is measured with a different experimental setting each time. A causal model is plausible as an explanation if it simultaneously fits all data sets when the effect of manipulations and selection of measured variables is taken into consideration.

COmbINE searches for the set of causal models that simultaneously fits all available data sets in the sense given above. The algorithm outputs a summary network that includes all the variant and invariant pairwise causal characteristics of the set of fitting models. For example, it indicates the causal relations upon which all fitting models agree, as well as the ones for which conflicting explanations are plausible. As our formalism of choice for causal modeling, we employ Semi-Markov Causal Models (**SMCMs**). SMCMs (Tian and Pearl, 2003) are extensions of Causal Bayesian Networks (**CBNs**) that can account for latent confounding variables, but do not admit feedback cycles. Internally, the algorithm also makes heavy use of the theory and learning algorithm for Maximal Ancestral Graphs (**MAGs**) (Richardson and Spirtes, 2002).

The algorithm builds upon the ideas in Triantafillou et al. (2010) to convert the observed statistical dependencies and independencies in the data to path constraints on the plausible data generating structures. The constraints are encoded as a SAT instance and solved with modern SAT engines, exploiting the efficiency of state-of-the-art solvers. However, due to statistical errors in the determination of dependencies and independencies, conflicting constraints may arise. In this case, the SAT instance is unsolvable and no useful information can be inferred. COmbINE includes a technique for sorting constraints according to confidence: The constraints are added to the SAT instance in decreasing order of confidence, and the ones that conflict with the set of higher-ranked constraints are discarded. The technique is general and the ranking score is a function of the p-values of the statistical tests of independence. It can therefore be applied to any type of data, provided an appropriate test exists.

COmbINE is empirically compared against a similar, recently developed algorithm by Hyttinen et al. (2013). The latter is also based on conversion to SAT and is able to additionally deal with cyclic structures, but assumes lack of statistical errors and corresponding conflicts. It can therefore not be directly applied to typical real problems that may generate such conflicts. COmbINE proves to be more efficient than Hyttinen et al. (2013) and scales to larger problem sizes, due to an inherently more compact representation of the path-constraints. The empirical evaluation also includes a quantification of the effect of sample size, number of data sets co-analyzed, and other factors on the quality and computational efficiency of learning. In addition, the proposed conflict resolution technique's superiority is demonstrated over several other alternative conflict resolution methods. Finally, we present a proof-of-concept computational experiment by applying the algorithm on 5 heterogeneous data sets from Bendall et al. (2011) and Bodenmiller et al. (2012) measuring overlapping variable sets under 3 different manipulations. The data sets measure protein concentrations in thousands of human cells of the autoimmune system using mass-cytometry technologies. Mass cytometers can perform single-cell measurements with a rate of about 10,000 cells per second, but can currently only measure up to circa 30 variables per run. Thus, they seem to form a suitable test-bed for integrative causal analysis algorithms.

The rest of this paper is organized as follows: Section 2 presents the related literature on learning causal models and combining multiple data sets. Section 3 reviews the necessary theory of MAGs and SMCMs and discusses the relation between the two and how hard manipulations are modeled in each. Section 4 is the core of this paper, and it is split in three subsections; presenting the conversion to SAT; introducing the algorithm and proving soundness and completeness with respect to the observed independence models; introducing the conflict resolution strategy. Section 5 is devoted to the experimental evaluation of the algorithm: testing the algorithm's performance in several settings and presenting an actual case study where the algorithm can be applied. Finally, Section 6 summarizes the conclusions and proposes some future directions of this work.

2. Related Work

Methods for causal discovery have been, for the most part, limited to the analysis of a single data set. However, the great advancement of intervention and data collection technology has led to a vast increase of available data sets, both observational and experimental. Therefore, over the last few years, there have been a number of works that focus on causal discovery from multiple sources. Algorithms in that area may differ in the formalism they use to model causality or in the type of heterogeneity in the studies they co-analyze. In any case, the goal is always to discover the single underlying data-generating causal mechanism.

One group of algorithms focuses on combining observational data that measure overlapping variables. Tillman et al. (2008) and Triantafillou et al. (2010) both provide sound and complete algorithms for learning the common characteristics of MAGs from data sets measuring overlapping variables. Tillman et al. (2008) handles conflicts by ignoring conflicting evidence, while the method presented in Triantafillou et al. (2010) only works with an oracle of conditional independence. Tillman and Spirtes (2011) present an algorithm for the same task that handles a limited type of conflicts (those concerning p-values for the same pair of variables stemming from different data sets) by combining the p-values for conditional

independencies that are testable in more than one data sets. Claassen and Heskes (2010b) present a sound, but not complete, algorithm for causal structure learning from multiple independence models over overlapping variables by transforming independencies into a set of causal ancestry rules.

Another line of work deals with learning causal models from multiple experiments. Cooper and Yoo (1999) use a Bayesian score to combine experimental and observational data in the context of causal Bayesian networks. Hauser and Bühlmann (2012) extend the notion of Markov equivalence for DAGs to the case of interventional distributions arising from multiple experiments, and propose a learning algorithm. Tong and Koller (2001) and Murphy (2001) use Bayesian network theory to propose experiments that are most informative for causal structure discovery. Eberhardt and Scheines (2007) and Eaton and Murphy (2007b) discuss how some other types of interventions can be modeled and used to learn Bayesian networks. Hyttinen et al. (2012a) provides an algorithm for learning linear cyclic models from a series of experiments, along with sufficient and necessary conditions for identifiability. This method admits latent confounders but uses linear structural equations to model causal relations and is therefore inherently limited to linear relations. Meganck et al. (2006) propose learning SMCMs by learning the Markov equivalence classes of MAGs from observational data and then designing the experiments necessary to convert it to a SMCM.

Finally, there is a limited number of methods that attempt to co-analyze data sets measuring overlapping variables under different experimental conditions. In Hyttinen et al. (2012b) the authors extend the methods of Hyttinen et al. (2012a) to handle overlapping variables, again under the assumption of linearity. Hyttinen et al. (2013) propose a constraint-based algorithm for learning causal structure from different manipulations of overlapping variable sets. The method works by transforming the observed m-connection and m-separation constraints into a SAT instance. The method uses a path analysis heuristic to reduce the number of tests translated into path constraints. Causal insufficiency is allowed, as well as feedback cycles. However, this method cannot handle conflicts and therefore relies on an oracle of conditional independence. Moreover, the method can only scale up to about 12 variables. Claassen and Heskes (2010a) present an algorithm for learning causal models from multiple experiments; the experiments here are not hard manipulations, but general experimental conditions, modeled like variables that have no parents in the graph but can cause other variables in some of the conditions.

To the best of our knowledge, COmbINE is the first algorithm to address both overlapping variables and multiple interventions for acyclic structures without relying on specific parametric assumptions or requiring an oracle of conditional independence. While the limits of COmbINE in terms of input size have not been exhaustively checked, the algorithm scales up to networks of up to 100 variables for relatively sparse networks (maximum number of parents equals 5).

3. Mixed Causal Models

Causally insufficient systems are often described using Semi-Markov causal models (SMCMs) (Tian and Pearl, 2003) or Maximal Ancestral Graphs (MAGs) (Richardson and Spirtes, 2002; Richardson, 2003). Both of them are **mixed graphs**, meaning they can contain both directed ($\longrightarrow$) and bi-directed ($\longleftrightarrow$) edges. We use the term **mixed causal**

graph to denote both. In this section, we will briefly present their common and unique properties. First, let us review the basic mixed graph notation:

In a mixed graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, a path is a sequence of distinct nodes $\langle V_0, V_1, \ldots, V_n \rangle$ s.t for $0 \leq i < n$, V_i and V_{i+1} are adjacent in $\mathcal{G}$. X is called a **parent** of Y and Y a **child** of X in $\mathcal{G}$ if $X \dashrightarrow Y$ in $\mathcal{G}$. A path from V_0 to V_n is **directed** if for $0 \leq i < n$, V_i is a parent V_{i+1}. X is called an **ancestor** of Y and Y is called a **descendant** of X in $\mathcal{G}$ if $X = Y$ in $\mathcal{G}$ or there exists a directed path from X to Y in $\mathcal{G}$. We use the notation $\mathbf{Pa}_{\mathcal{G}}(\mathbf{X}), \mathbf{Ch}_{\mathcal{G}}(\mathbf{X}), \mathbf{An}_{\mathcal{G}}(\mathbf{X}), \mathbf{Desc}_{\mathcal{G}}(\mathbf{X})$ to denote the set of parents, children, ancestors and descendants of nodes $\mathbf{X}$ in $\mathcal{G}$. A **directed cycle** in $\mathcal{G}$ occurs when $X \rightarrow Y \in \mathbf{E}$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. An **almost directed cycle** in $\mathcal{G}$ occurs when $X \leftrightarrow Y \in \mathbf{E}$ and $Y \in \mathbf{An}_{\mathcal{G}}(X)$. Given a path p in a mixed graph, a non-endpoint node V on p is called a **collider** if the two edges incident to V on p are both into V. Otherwise V is called a **non-collider**. A path $p = \langle X, Y, Z \rangle$, where X and Z are not adjacent in $\mathcal{G}$ is called an **unshielded triple**. If Y is a collider on this path, the triple is called an **unshielded collider**.

MAGs and SMCMs are graphical models that represent both causal relations and conditional independencies among a set of measured (observed) variables $\mathbf{O}$, and can be viewed as generalizations of causal Bayesian networks that can account for latent confounders. MAGs can also account for selection bias, but in this work we assume selection bias is not present.

3.1 Semi-Markov Causal Models

Semi-Markov causal models (SMCMs), introduced by Tian and Pearl (2003), often also reported as Acyclic Directed Mixed Graphs (ADMGs), are causal models that implicitly model hidden confounders using bi-directed edges. A directed edge $X \dashrightarrow Y$ denotes that X is a *direct* cause of Y in the context of the variables included in the model. A bi-directed edge $X \dashleftarrow\dashrightarrow Y$ denotes that X and Y are confounded by an unobserved variable. Two variables can be joined by at most two edges, one directed and one bi-directed.

Semi-Markov causal models are designed to represent marginals of causal Bayesian networks. In DAGs, the probabilistic properties of the distribution of variables included in the model can be determined graphically using the criterion of d-separation. The natural extension of d-separation to mixed causal graphs is called m-separation:

Definition 1 (m-connection, m-separation.) *In a mixed graph $\mathcal{G} = (\mathbf{E}, \mathbf{V})$, a path p between A and B is **m-connecting** given (conditioned on) a set of nodes $\mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \backslash \{A, B\}$ if*

1. *Every non-collider on p is not a member of $\mathbf{Z}$.*
2. *Every collider on the path is an ancestor of some member of $\mathbf{Z}$.*

*A and B are said to be m-**separated** by $\mathbf{Z}$ if there is no m-connecting path between A and B relative to $\mathbf{Z}$. Otherwise, we say they are m-**connected** given $\mathbf{Z}$. We use the notation $\mathcal{J}_m(\mathcal{G})$ to denote the set of m-separations that hold in $\mathcal{G}$.*

Let $\mathcal{G}$ be a SMCM over a set of variables $\mathbf{O}$, Π the joint probability distribution (JPD) over the same set of variables and $\mathcal{J}(\Pi)$ the independence model, defined as the set of conditional independencies that hold in Π. We use $\langle \mathbf{X}, \mathbf{Y} | \mathbf{Z} \rangle$ to denote the conditional

independence of variables in $\mathbf{X}$ with variables in $\mathbf{Y}$ given variables in $\mathbf{Z}$. Under the Causal Markov (**CMC**) and Faithfulness (**FC**) conditions (Spirtes et al., 2001), *every m-separation present in $\mathcal{G}$ corresponds to a conditional independence in $\mathcal{J}(\Pi)$ and vice-versa: $\mathcal{J}_m(\mathcal{G}) = \mathcal{J}(\Pi)$.*

In causal Bayesian networks, every missing edge in $\mathcal{G}$ corresponds to a conditional independence in $\mathcal{J}(\Pi)$ (resp. an m-separation in $\mathcal{G}$), meaning there exists a subset of the variables in the model that renders the two non-adjacent variables independent. Respectively, every conditional independence in $\mathcal{J}(\Pi)$ corresponds to a missing edge in the DAG $\mathcal{G}$. This is not always true for SMCMs. Figure 1 illustrates an example of a SMCM where two non-adjacent variables are not independent given any subset of observed variables.

Evans and Richardson (2010, 2011) deal with the factorization and parameterization of SMCMs for discrete variables. Based on this parameterization, score-based methods have also recently been explored (Richardson et al., 2012; Shpitser et al., 2013), but are still limited to small sets of discrete variables. The skeleton of a SMCM is not uniquely identifiable by the corresponding conditional independence model on the same variables (see Figure 1 for an example). Richardson and Spirtes (2002) overcome this obstacle by introducing a causal mixed graph with slightly different semantics, the maximal ancestral graph.

3.2 Maximal Ancestral Graphs

Maximal ancestral graphs (MAGs) (Richardson and Spirtes, 2002), are **ancestral** mixed graphs, meaning that they contain no directed or almost directed cycles, where an almost directed cycle occurs if $X \leftrightarrow Y$ and X causes Y. Every pair of variables X, Y in an ancestral graph is joined by at most one edge. The orientation of this edge represents (non) causal ancestry: A bi-directed edge $X \leftrightarrow Y$ denotes that X does not cause Y and Y does not cause X, but (under the faithfulness assumption) the two share a latent confounder. A directed edge $X \rightarrow Y$ denotes causal ancestry: X is a *causal ancestor* of Y. Thus, if X causes Y (not necessarily directly in the context of observed variables) and they are also confounded, there is an edge $X \rightarrow Y$ in the corresponding MAG. Undirected edges can also be present in MAGs that account for selection bias. As mentioned above, we assume no selection bias in this work and the theory of MAGs presented here is restricted to MAGs with no undirected edges.

Like SMCMs, ancestral graphs are also designed to represent marginals of causal Bayesian networks. Thus, under the causal Markov and faithfulness conditions for a MAG $\mathcal{M}$ and a jpd Π, X and Y are m-separated given $\mathbf{Z}$ in an ancestral graph $\mathcal{M}$ if and only if $\langle X, Y|\mathbf{Z}\rangle$ is in the corresponding independence model $\mathcal{J}(\Pi)$. Still, like in SMCMs, a missing edge does not necessarily correspond to a conditional independence. The following definition describes a subset of ancestral graphs in which every missing edge (non-adjacency) corresponds to a conditional independence:

Definition 2 (Maximal Ancestral Graph, MAG) *A mixed graph is called* ancestral *if it contains no directed and almost directed cycles. An ancestral graph $\mathcal{G}$ is called* maximal *if for every pair of non-adjacent nodes (X, Y), there is a (possibly empty) set $\mathbf{Z}$, $X, Y \notin \mathbf{Z}$ such that $\langle X, Y|\mathbf{Z}\rangle \in \mathcal{J}_m(\mathcal{G})$.*

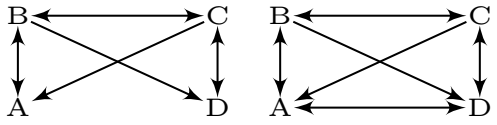


Figure 1: Maximality and primitive inducing paths.(a) Both (i) a semi Markov causal model over variables $\{A,\ B,\ C,\ D\}$; variables A and D are m-connected given any subset of observed variables, but they do not share a direct relationship in the context of observed variables and (ii) a non-maximal ancestral graph over variables $\{A,\ B,\ C,\ D\}$. (b) The corresponding MAG. A and D are adjacent, since they cannot be m-separated given any subset of $\{B, C\}$. Path $\langle A, B, C, D\rangle$ is a primitive inducing path. This example was presented in Zhang (2008b).

Figure 1 illustrates an ancestral graph that is not maximal, and the corresponding maximal ancestral graph. MAGs are closed under marginalization (Richardson and Spirtes, 2002). Thus, if $\mathcal{G}$ is a MAG faithful to Π, then there is a unique MAG $\mathcal{G}'$ faithful to any marginal distribution of Π.

We use $[_{\mathbf{L}}$ to denote the act of marginalizing out variables $\mathbf{L}$, thus, if $\mathcal{G}$ is a MAG over variables $\mathbf{O} \cup \mathbf{L}$ faithful to a joint probability distribution Π, $\mathcal{G}[_{\mathbf{L}}$ is the MAG over $\mathbf{O}$ faithful to the marginal joint probability distribution of Π. We use $\mathcal{J}[_{\mathbf{L}}$ to denote the *marginal independence model* of $\mathcal{J}$, i.e. the set of conditional independencies $\{X \perp\!\!\!\perp Y \mid \mathbf{Z} \in \mathcal{J} : (X \cup Y \cup \mathbf{Z}) \cap \mathbf{L} = \emptyset\}$. Obviously, the DAG of a causal Bayesian network is also a MAG. For a MAG $\mathcal{G}$ over $\mathbf{O}$ and a set of variables $\mathbf{L} \subset \mathbf{O}$, the marginal MAG $\mathcal{G}[_{\mathbf{L}}$ is defined as follows:

Definition 3 (Marginal MAG) *(Richardson and Spirtes, 2002) MAG $\mathcal{G}[_{\mathbf{L}}$ has node set $\mathbf{O} \setminus \mathbf{L}$ and edges specified as follows: If X, Y are s.t. $\forall \mathbf{Z} \subset \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$, X and Y are m-connected given $\mathbf{Z}$ in $\mathcal{G}$, then*

$$\text{if } \left\{ \begin{array}{l} X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) \\ X \in \mathbf{An}_{\mathcal{G}}(Y); Y \notin \mathbf{An}_{\mathcal{G}}(X) \\ X \notin \mathbf{An}_{\mathcal{G}}(Y); Y \in \mathbf{An}_{\mathcal{G}}(X) \end{array} \right\} \text{ then } \left\{ \begin{array}{l} X \leftrightarrow Y \\ X \rightarrow Y \\ X \leftarrow Y \end{array} \right\} \text{ in } \mathcal{G}[_{\mathbf{L}}$$

The following theorem was proved in Richardson and Spirtes (2002):

Theorem 4 *If $\mathcal{G}$ is a MAG over $\mathbf{V} = \mathbf{O} \cup \mathbf{L}$, then $\mathcal{J}_m(\mathcal{G}[_{\mathbf{L}}) = \mathcal{J}_m(\mathcal{G})[_{\mathbf{L}}$.*

Proof See proof of Theorem 4.18 in Richardson and Spirtes (2002). ■

As mentioned above, every conditional independence in an independence model $\mathcal{J}$ corresponds to a missing edge in the corresponding faithful MAG $\mathcal{G}$. Conversely, if X and Y are dependent given every subset of observed variables, then X and Y are adjacent in $\mathcal{G}$. Thus, given an oracle of conditional independence it is possible to learn the skeleton of a MAG $\mathcal{G}$ over variables $\mathbf{O}$ from a data set. Still, some of the orientations of $\mathcal{G}$ are not

distinguishable by mere observations. The set of MAGs $\mathcal{G}$ faithful to distributions Π that entail a set of conditional independencies $\mathcal{J}(\Pi)$ form a **Markov equivalence class**.

It is well known that two DAGs are Markov equivalent if and only if they share the same adjacencies and unshielded colliders. Markov equivalent MAGs also share adjacencies and unshielded colliders, but this is not sufficient to characterize Markov equivalent graphs. The emergence of bi-directed edges imposes also a set of shielded colliders on the Markov equivalent MAGs. These colliders are discriminated by *discriminating paths*:

Definition 5 (Discriminating path) *A path $p = \langle X, \dots, W, V, Y \rangle$ is called **discriminating** for V if X is not adjacent to Y and every node on the path from X to V is a collider and a parent of Y.*

Discriminating paths, their properties and their connection to Markov equivalence is discussed in detail in Ali et al. (2009). Unfortunately, two Markov equivalent MAGs may not share the same discriminating paths. Moreover, a triple may be discriminated to be a collider in MAG $\mathcal{M}_1$ but not in MAG $\mathcal{M}_2$ in the same Markov equivalence class. There exists however, a subset of discriminating paths that (a) are present in all the Markov equivalent MAGs and (b) the colliders discriminated by these paths are necessary and sufficient for Markov equivalence (Ali et al., 2009). The following definition from Ali et al. (2009) is relevant:

Definition 6 (Colliders with order) *Let $\mathfrak{D}_i, i \geq 0$ be a set of triples of order i in MAG $\mathcal{M}$, defined recursively as follows:*

- *Order 0: A triple $\langle X, Y, Z \rangle \in \mathfrak{D}_0$ if X and Z are not adjacent.*
- *Order i: A triple $\langle X, Y, Z \rangle \in \mathfrak{D}_{i+1}$ if,*
  1. *for all $j < i + 1, \langle X, Y, Z \rangle \notin \mathfrak{D}_j$ and*
  2. *There is a discriminating path $\langle W, V_1, \dots, V_n, Y, Q \rangle$ such that either $\langle X, Y, Z \rangle = \langle V_n, Y, Q \rangle$ or $\langle X, Y, Z \rangle = \langle Q, Y, V_n \rangle$ and the n colliders:*

$$\langle W, V_1, V_2 \rangle, \dots, \langle V_{n-1}, V_n, Y \rangle \in \bigcup_{j \leq i} \mathfrak{D}_j$$

*If $\langle X, Y, Z \rangle \in \mathfrak{D}_i$, the triple has order i. If the triple has order i for some i, then we say the triple has order. If $\langle X, Y, Z \rangle$ is a triple with order and $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ is in $\mathcal{M}$, then the triple is a **collider with order** i in $\mathcal{M}$. Otherwise, the triple is a **definite non-collider with order** in $\mathcal{M}$. A discriminating path p has order i if all colliders on the path (except from the collider $\langle V_n, Y, Q \rangle$ discriminated by the path) have order at most $i - 1$, and there exists at least one collider with order $i - 1$. If a discriminating path has order i for some i, then we say that the discriminating path has order. In this work we (abusively) call (non) colliders with order ≥ 1 **discriminating (definite non) colliders**.*

Note that not every triple on a mixed graph has order. The order (if any) of a shielded triple is the minimum of the orders of all discriminating paths with order for that triple. Triples with order 0 are the unshielded triples. Discriminating paths with order ≥ 1 are

present in all Markov equivalent MAGs, and therefore colliders with order ≥ 1 are the triples that are colliders in all the Markov equivalent MAGs. Colliders with order, along with adjacencies, are necessary and sufficient to characterize Markov equivalent MAGs:

Theorem 7 *Two MAGs over the same variable set are Markov equivalent if and only if they share the same edges and the same colliders with order.*

Proof See proof of Theorem 3.7 in Ali et al. (2009). ■

We use $[\mathcal{G}]$ to denote the class of MAGs that are Markov equivalent to $\mathcal{G}$. A **partial ancestral graph (PAG)** is a representative graph of this class, and has the skeleton shared by all the graphs in $[\mathcal{G}]$, and all the orientations invariant in all the graphs in $[\mathcal{G}]$. Endpoints that can be either arrows or tails in different MAGs in $\mathcal{G}$ are denoted with a circle "$\circ$" in the representative PAG. We use the symbol $\star$ as a wild card to denote any of the three marks. We use the notation $\mathcal{M} \in \mathcal{P}$ to denote that MAG $\mathcal{M}$ belongs to the Markov equivalence class represented by PAG $\mathcal{P}$.

For a MAG $\mathcal{M}$ and a probability distribution Π faithful to each other, $\mathcal{J}_m(\mathcal{M}) = \mathcal{J}(\Pi)$. Thus, the set of m-separations entailed in $\mathcal{M}$ are exactly the conditional independencies that hold in Π. **FCI** Algorithm (Spirtes et al., 2001; Zhang, 2008a) is a sound and complete algorithm for learning the complete (maximally informative) PAG of the MAGs faithful to a distribution Π over variables $\mathbf{O}$ in which a set of conditional independencies $\mathcal{J}(\Pi)$ hold. An important advantage of FCI is that it employs CMC, faithfulness and some graph theory to reduce the number of tests required to identify the correct PAG.

3.3 Correspondence Between SMCMs and MAGs

Semi Markov Causal Models and Maximal Ancestral Graphs both represent causally insufficient causal structures. They both entail the conditional independence structure and the causal ancestry structure of the observed variables. Thus, under CMC and FC, the SMCM $\mathcal{G}$ and the MAG $\mathcal{M}$ over a set of variables $\mathbf{O}$ entail the same independence model: $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$. They also entail the same ancestral relationships: X is an ancestor of Y in $\mathcal{S}$ if and only if X is an ancestor of Y in $\mathcal{M}$.

Nevertheless, SMCMs and MAGs also have significant differences: SMCMs describe the causal relations among observed variables, while MAGs encode independence structure with partial causal ordering. Edge semantics in SMCMs are closer to the semantics of causal Bayesian networks, whereas edge semantics in MAGs are more complicated. On the other hand, unlike in DAGs and MAGs, a missing edge in a SMCM does not necessarily correspond to a conditional independence (SMCMs do not obey a pairwise Markov property).

Figure 2 summarizes the main differences of SMCMs and MAGs. It shows two different DAGs, and the corresponding marginal SMCMs and MAGs over four observed variables. SMCMs have a many-to-one relationship to MAGs: For a MAG $\mathcal{M}$, there can exist more than one SMCMs that entail the same probabilistic and causal ancestry relations. On the other hand, for any given SMCM there exists only one MAG entailing the same probabilistic and causal ancestry relations. This is clear in Figure 2, where a unique MAG, $\mathcal{M}_1 = \mathcal{M}_2$ entails the same information as two different SMCMs, $\mathcal{S}_1$ and $\mathcal{S}_2$ in the same figure.

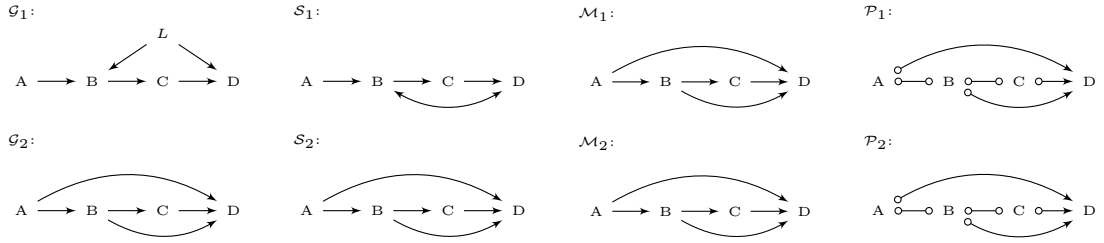


Figure 2: An example two different DAGs and the corresponding mixed causal graphs over observed variables. On the left we can see DAGs $\mathcal{G}_1$ over variables $\{A,\ B,\ C,\ D,\ L\}$ (top) and $\mathcal{G}_2$ over variables $\{A,\ B,\ C,\ D\}$ (bottom). From left to right, on the same row as the underlying causal DAG, we can see the respective SMCMs $\mathcal{S}_1$ and $\mathcal{S}_2$ over $\{A,\ B,\ C,\ D\}$; the respective MAGs $\mathcal{M}_1 = \mathcal{G}_1[_L$ and $\mathcal{M}_2 = \mathcal{G}_2$ over variables $\{A,\ B,\ C,\ D\}$; finally, the respective PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$. Notice that, $\mathcal{M}_1$ and $\mathcal{M}_2$ are identical, despite representing different underlying causal structures.

Directed edges in a SMCM denote a causal relation that is *direct* in the context of observed variables. In contrast, a directed edge in a MAG merely denotes causal ancestry; the causal relation is not necessarily direct. An edge $X \dashrightarrow Y$ can be present in a MAG even though X does not directly cause Y; this happens when X is a causal ancestor of Y and the two cannot be rendered independent given any subset of observed variables. Depending on the structure of latent variables, this edge can be either missing or bi-directed in the respective SMCM.

In Figure 2 we can see examples of both cases. For example, A is a causal ancestor of D in DAG $\mathcal{G}_1$, but not a direct cause (in the context of observed variables). Therefore, the two are not adjacent in the corresponding SMCM $\mathcal{S}_1$ over $\{A, B, C, D\}$. However, the two cannot be rendered independent given any subset of $\{B, C\}$, and therefore $A \dashrightarrow D$ is in the respective MAG $\mathcal{M}_1$.

On the same DAG, B is another causal ancestor (but not a direct cause) of D. The two variables share the common cause L. Thus, in the corresponding SMCM $\mathcal{S}_1$ over $\{A, B, C, D\}$ we can see the edge $B \dashleftarrow\dashrightarrow D$. However, a bi-directed edge between B and D is not allowed in MAG $\mathcal{M}_1$, since it would create an almost directed cycle. Thus, $B \dashrightarrow D$ is in $\mathcal{M}_1$.

We must also note that unlike SMCMs, MAGs only allow one edge per variable pair. Thus, if X directly causes Y and the two are also confounded, both edges will be in a relevant SMCM ($X \rightleftarrows Y$), while the two will share a directed edge from X to Y in the corresponding MAG.

Overall, a SMCM has a subset of the adjacencies (but not necessarily edges) of its MAG counterpart. These extra adjacencies in MAGs correspond to pairs of variables that cannot be m-separated given any subset of observed variables, but neither directly causes the other, and the two are not confounded. These adjacencies can be checked in a SMCM using a special type of path, called **inducing path** (Richardson and Spirtes, 2002).

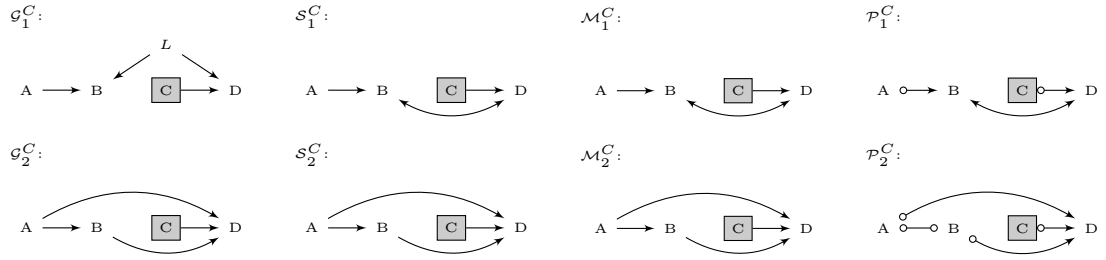


Figure 3: Effect of manipulating variable C on the causal graphs of Figure 2. From right to left we can see the manipulated DAGs $\mathcal{G}_1^C$ (top) and $\mathcal{G}_2^C$ (bottom), the manipulated SMCMs $\mathcal{S}_1^C$ (top) and $\mathcal{S}_2^C$ (bottom) over variables $\{A,\ B,\ C,\ D\}$, the manipulated MAGs $\mathcal{M}_1^C = \mathcal{G}_1^C[_L$ (top) and $\mathcal{M}_2^C = \mathcal{G}_2^C$ (bottom) over the same set of variables, and the corresponding PAGs $\mathcal{P}_1^C$ (top) and $\mathcal{P}_2^C$ (bottom). Notice that edge $A \longrightarrow D$ is removed in $\mathcal{M}_1^C$, even though it is not adjacent to the manipulated variable. Moreover, on the same graph, edge $B \longrightarrow D$ is now $B \longleftrightarrow D$.

Definition 8 (Inducing path) *A path $p = \langle V_1, V_2, \ldots, V_n \rangle$ on a mixed causal graph $\mathcal{G}$ over a set of variables $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$ is called **inducing** with respect to $\mathbf{L}$ if every non-collider on the path is in $\mathbf{L}$ and every collider is an ancestor of either V_1 or V_n. A path that is inducing with respect to the empty set is called a **primitive** inducing path.*

Obviously, an edge joining X and Y is a primitive inducing path. Intuitively, an inducing path with respect to $\mathbf{L}$ is m-connecting given any subset of variables that does not include variables in $\mathbf{L}$. Path $A \longrightarrow B \longleftarrow L \longrightarrow D$ is an inducing path with respect to L in $\mathcal{G}_1$ of Figure 2, and $A \longrightarrow B \longleftrightarrow D$ is an inducing path with respect to the empty set in $\mathcal{S}_1$ of the same figure. Inducing paths are extensively discussed in Richardson and Spirtes (2002), where the following theorem is proved:

Theorem 9 *If $\mathcal{G}$ is an ancestral graph over variables $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$, and $X, Y \in \mathbf{O}$ then the following statements are equivalent:*

1. *X and Y are adjacent in $\mathcal{G}[_{\mathbf{L}}$.*
2. *There is an inducing path with respect to $\mathbf{L}$ in $\mathcal{G}$.*
3. *$\forall \mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$, X and Y are m-connected given $\mathbf{Z}$ in $\mathcal{G}$.*

Proof See proof of Theorem 4.2 in Richardson and Spirtes (2002). ■

This theorem links inducing paths in an ancestral graph to m-separations in the same graph and to adjacencies in any marginal ancestral graph. The equivalence of (ii) and (iii) can also be proved for SMCMs, using the proof presented in Richardson and Spirtes (2002) for Theorem 9:

Theorem 10 *If $\mathcal{G}$ is a SMCM over variables $\mathbf{V} = \mathbf{O} \dot{\cup} \mathbf{L}$, and $X, Y \in \mathbf{O}$ then the following statements are equivalent:*

1. *There is an inducing path with respect to* $\mathbf{L}$ *in* $\mathcal{G}$.

2. $\forall \mathbf{Z}$, $\mathbf{Z} \subseteq \mathbf{V} \setminus \mathbf{L} \cup \{X, Y\}$, X *and* Y *are m-connected given* $\mathbf{Z}$ *in* $\mathcal{G}$.

Proof See proof of Theorem 4.2 in Richardson and Spirtes (2002). ■

The following proposition follows from Theorems 9 and 10:

Proposition 12 . *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$. *Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{O}$ *that is faithful to* $\mathcal{J}$ *and* $\mathcal{M}$ *be the MAG over the same variables that is faithful to* $\mathcal{J}$. *Let* $X, Y \in \mathbf{O}$. *Then there is an inducing path between* X *and* Y *with respect to* $\mathbf{L}$, $\mathbf{L} \subseteq \mathbf{O}$ *in* $\mathcal{S}$ *if and only if there is an inducing path between* X *and* Y *with respect to* $\mathbf{L}$ *in* $\mathcal{M}$.

Proof See Appendix A. ■

Primitive inducing paths are connected to the notion of maximality in ancestral graphs: Every ancestral graph can be transformed into a maximal ancestral graph with the addition of a finite number of bi-directed edges. Such edges are added between variables X, Y that are m-connected through a **primitive inducing path** (Richardson and Spirtes, 2002). Path $A \leftrightarrow B \leftrightarrow C \leftrightarrow D$ in Figure 1 is an example of a primitive inducing path.

Inducing paths are crucial in this work because adjacencies and non-adjacencies in marginal ancestral graphs can be translated into existence or absence of inducing paths in causal graphs that include some additional variables. For example, path $A \rightarrow B \leftarrow L \rightarrow D$ is an inducing path w.r.t. L in $\mathcal{G}_1$ in Figure 2, and therefore A and D are adjacent in $\mathcal{M}_1$. Thus, inducing paths are useful for combining causal mixed graphs over overlapping variables.

Inducing paths are also necessary to decide whether two variables in an SMCM will be adjacent in a MAG over the same variables without having to check all possible m-separations. Algorithm 1 describes how to turn a SMCM into a MAG over the same variables.

Algorithm 1 takes as input a SMCM $\mathcal{S}$ and adds the necessary edges to transform it into a MAG $\mathcal{M}$ by looking for primitive inducing paths. The procedure can be viewed as a special case of marginalizing out variables in DAGs, presented in Spirtes and Richardson (1996) and Zhang (2008b). Similar algorithms are also presented in Sadeghi (2012), where the relationship among different types of mixed causal graphs representing the same independence model is discussed in detail. The algorithm is sound, i.e. the output MAG shares the same causal ancestry relations and entails the same independence model as $\mathcal{S}$:

Theorem 13 . *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$. *Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{V}$ *that is faithful to* $\mathcal{J}$. *Let* $\mathcal{M}$ = *SMCMtoMAG(*$\mathcal{S}$*). Then* $\mathcal{S}$ *and* $\mathcal{M}$ *share the same ancestry relations and* $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$.

Proof See Appendix A. ■

Algorithm 1: SMCMtoMAG

input : SMCM $\mathcal{S}$
output: MAG $\mathcal{M}$

$\mathcal{M} \leftarrow \mathcal{S}$;
foreach *ordered pair of variables* X, Y *not adjacent in* $\mathcal{S}$ **do**
  if $\exists$ *primitive inducing path from* X *to* Y *in* $\mathcal{S}$ **then**
    if $X \in \mathbf{An}_{\mathcal{S}}(Y)$ **then**
      add $X \dashrightarrow Y$ to $\mathcal{M}$;
    else if $Y \in \mathbf{An}_{\mathcal{S}}(X)$ **then**
      add $Y \dashrightarrow X$ to $\mathcal{M}$;
    else
      add $Y \dashleftarrow\dashrightarrow X$ to $\mathcal{M}$;
    end
  end
end
foreach $X \overleftrightarrows Y$ *in* $\mathcal{M}$ **do**
  remove $X \dashleftarrow\dashrightarrow Y$;
end

The algorithm is also complete, since there only exists one such MAG. The inverse procedure, converting a MAG into the underlying SMCM, is not possible, since we cannot know in general which of the edges correspond to direct causation or confounding and which are there because of a (non-trivial) primitive inducing path. Note though that, there exist sound and complete algorithms that identify all edges for which such a determination is possible (Borboudakis et al., 2012). In addition, in the next section we show that co-examining manipulated distributions can indicate that some edges stand for indirect causality (or indirect confounding).

3.4 Manipulations Under Causal Insufficiency

An important motivation for using causal models is to predict causal effects. In this work, we focus on hard manipulations, where the value of the manipulated variables is set exclusively by the manipulation procedure. We also adopt the assumption of locality, denoting that the intervention of each manipulated variable should not directly affect any variable other than its direct target, and more importantly, local mechanisms for other variables should remain the same as before the intervention (Zhang, 2006). Thus, the intervention is merely a local surgery with respect to causal mechanisms. These assumptions may seem a bit restricting, but this type of experiment is fairly common in several modern fields where the technical capability for precise interventions is available, such as, for example, molecular biology. Finally, we assume that the manipulated model is faithful to the corresponding manipulated distributions.

In the context of causal Bayesian networks, hard interventions are modeled using what is referred to as "graph surgery", in which all edges incoming to the manipulated variables are removed from the graph. The resulting graph is referred to as the **manipulated graph**.

Naturally, DAGs are closed under manipulation. We use the term **intervention target** to denote a set of manipulated variables. For a DAG $\mathcal{G}$ and an intervention target $\mathbf{I}$, we use $\mathcal{G}^{\mathbf{I}}$ to denote the manipulated DAG. Parameters of the distribution that refer to the probability of manipulated variables given their parents are replaced by the parameters set by the manipulation procedure, while all other parameters remain intact. We use $\Pi^{\mathbf{I}}$ to denote this **manipulated joint probability distribution**, and $\mathcal{J}^{\mathbf{I}}$ to denote the corresponding **manipulated independence model**.

Graph surgery can be easily extended to SMCMs: One must simply remove edges into the manipulated variables. Again, we use the notation $\mathcal{S}^{\mathbf{I}}$ to denote the graph resulting from a SMCM $\mathcal{S}$ after the manipulation of variables in $\mathbf{I}$. In contrast, predicting the effect of manipulations in MAGs is not trivial. Due to the complicated semantics of the edges, the manipulated graph is usually not unique.

This becomes more obvious by looking at Figures 2 and 3. Figure 2 shows two different causal DAGs and the corresponding SMCMs and MAGs, and Figure 3 shows the effect of a manipulation on the same graphs. In Figure 2 the marginals of DAGs $\mathcal{D}_1$ and $\mathcal{D}_2$ are represented by the same MAG $\mathcal{M}_1 = \mathcal{M}_2$. However, after manipulating variable C, the resulting manipulated MAGs $\mathcal{M}_1^C$ and $\mathcal{M}_2^C$ do not belong to the same equivalence class (they do not even share the same skeleton). We must point out, that the indistinguishability of $\mathcal{M}_1$ and $\mathcal{M}_2$ refers to m-separation only; the absence of a direct causal edge between A and D could be detected using other types of tests, like the Verma constraint (Verma and Pearl, 2003).

While we cannot predict the effect of manipulations on a MAG $\mathcal{M}$, given a data set measuring variables $\mathbf{O}$ when variables in $\mathbf{I} \subset \mathbf{O}$ are manipulated, we can obtain (assuming an oracle of conditional independence) the PAG representative of the actual manipulated MAG $\mathcal{M}^{\mathbf{I}}$. We use $\mathcal{P}^{\mathbf{I}}$ to denote this PAG.

We must point out here that we use $\mathcal{P}^{\mathbf{I}}$ as the representative of the Markov equivalence class of models that are faithful to the manipulated conditional independence model $\mathcal{J}(\Pi^{\mathbf{I}})$, as opposed to the representative of the *interventional Markov equivalence class* of manipulated MAGs. The information on manipulations, not included in the present use of $\mathcal{P}^{\mathbf{I}}$, defines a smaller Markov equivalence class: For example, in Figure 3, MAGs in the interventional Markov equivalence class of $\mathcal{M}_1^C$ share the additional invariant characteristic of a tail into C on the edge $C \circ\!\!\rightarrow D$. This invariant feature however is not oriented in $\mathcal{P}_1^C$. To the best of our knowledge, no sound and complete algorithm for identifying the maximally informative PAG for the *interventional Markov equivalence class of* $\mathcal{M}^{\mathbf{I}}$ exists (however, orienting all edges out of the manipulated variables is a trivially sound method).

By observing PAGs $\{\mathcal{P}^{\mathbf{I}_i}\}$ that stem from known, different manipulations of the same underlying distribution, we can infer some refined information for the underlying causal model. Let's suppose, for example, that $\mathcal{G}_1$ in Figure 2 is the true underlying causal graph for variables $\{A, B, C, D, L\}$ and that we have the learnt PAGs $\mathcal{P}_1^A$ and $\mathcal{P}_1^C$ from relevant data sets. Graph $\mathcal{P}_1^A$ is not shown, but is identical to $\mathcal{P}_1$ in Figure 2 since A has no incoming edges in the underlying DAG (and SMCM). $\mathcal{P}_1^C$ is illustrated in Figure 3. Edge $A \circ\!\!-\!\!\circ D$ is present in $\mathcal{P}_1^A$, but is missing in $\mathcal{P}_1^C$ even though neither A nor D are manipulated in $\mathcal{P}_1^C$. By reasoning on the basis of both graphs, we can infer that edge $A \rightarrow D$ in $\mathcal{P}_1^A$ cannot denote a *direct* causal relation among the two variables, but must be the result of a primitive, non-trivial inducing path.

4. Learning Causal Structure From Multiple Data Sets Measuring Overlapping Variables Under Different Manipulations

In the previous section we described the effect of manipulation on MAGs and saw an example of how co-examining PAGs faithful to different manipulations of the same underlying distribution can help classify an edge between two variables as not direct.

In this section, we expand this idea and present a general, constraint-based algorithm for learning causal structure from overlapping manipulations. The algorithm takes as input a set of data sets measuring overlapping variable sets $\{\mathbf{O}_i\}_{i=1}^N$; in each data set, some of the observed variables can be manipulated. The set of manipulated variables in experiment i is also provided and is denoted with $\mathbf{I}_i$.

In the rest of this paper, we make the following assumptions:

A1 We assume that there exists an underlying causal mechanism over a set of variables $\mathbf{O}$ that can be described with a semi Markov causal model $\mathcal{G}$ over $\mathbf{O}$. If Π is the joint probability distribution over $\mathbf{O}$, we assume that Π and $\mathcal{G}$ are faithful to each other, i.e. $\mathcal{J}_m(\mathcal{G}) = \mathcal{J}(\Pi)$. We also say that $\mathcal{G}$ is faithful to the independence model $\mathcal{J}(\Pi)$.

A2 We assume that we collect data sets in N different experiments, where in experiment i we observe variables $\mathbf{O}_i \subseteq \mathbf{O}$, while variables $\mathbf{L}_i = \mathbf{O} \setminus \mathbf{O}_i$ are latent and variables $\mathbf{I}_i \subset \mathbf{O}$ are manipulated. We also assume $\mathbf{O} = \bigcup_{i=1}^N \mathbf{O}_i$. We assume that manipulations are ideal hard interventions and that they result in removal of all edges in $\mathcal{G}$ that are incoming to the manipulated variables.

A3 We assume faithfulness for the manipulated SMCMs and distributions, i.e. $\mathcal{J}_m(\mathcal{G}^{\mathbf{I}_i}) = \mathcal{J}(\Pi^{\mathbf{I}_i})$.

Unless mentioned otherwise, the following notation is used:

- $\mathbf{O}_i$ denotes the set of observed variables in experiment i.
- $\mathbf{I}_i$ denotes the set of manipulated variables in experiment i.
- $\mathbf{O} = \cup_i \mathbf{O}_i$ denotes the union of observed variables.
- $\mathbf{L}_i = \mathbf{O} \setminus \mathbf{O}_i$ denotes the set of latent variables (with respect to the union of observed variables) in experiment i.
- $\mathbf{D}_i$ denotes a data set for experiment i, sampled from the mechanism described by $(\mathcal{G}^{\mathbf{I}_i}, \Pi^{\mathbf{I}_i})$, measuring variables in $\mathbf{O}_i$.
- $\mathcal{J}_i$ denotes the independence model that holds in data set $\mathbf{D}_i$. In the sample limit, $\mathcal{J}_i$ is equal to the set of m-separations that hold for sets of variables in $\mathbf{O}_i$ after manipulating $\mathbf{I}_i$ in the underlying causal model: $\mathcal{J}_i = \mathcal{J}(\Pi^{\mathbf{I_i}})[_{\mathbf{L_i}} = \mathcal{J}_m(\mathcal{G}^{\mathbf{I_i}})[_{\mathbf{L_i}}$.
- $\mathcal{P}_i$ denotes the maximally informative PAG for the (observational) Markov equivalence class of MAGs faithful to $\mathcal{J}_i$. Thus, for any MAG $\mathcal{M}_i \in \mathcal{P}_i$, $\mathcal{J}_m(\mathcal{M}_i) = \mathcal{J}_i$. Notice that, since SMCMs and MAGs over the same variables represent the same independence model, for an oracle of conditional independence, $\mathcal{P}_i = [\text{SMCMtoMAG}(\mathcal{G}^{\mathbf{I}_i})[_{\mathbf{L}_i}]$.

Under the assumptions described above, we are interested in combining information across data sets collected from different manipulations and marginalizations of the same system under study, to identify features of the possible underlying causal mechanism. If $\mathcal{S}$ is a SMCM that describes this underlying causal mechanism, then this SMCM must agree with all the observed independence models $\{\mathcal{J}_i\}_{i=1}^N$. This means that for each experiment, the respective manipulated $\mathcal{S}^{\mathbf{I}_i}$ must entail all and only the conditional independencies that hold in data set $\mathbf{D}_i$ (in the sample limit $\mathcal{J}_i$ can be obtained correctly from the data). For the family of independence models $\{\mathcal{J}_i\}_{i=1}^N$, and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$ a **possibly underlying SMCM** is defined as follows:

Definition 11 (Possibly underlying SMCM) *If $\{\mathcal{J}_i\}_{i=1}^N$ is a family of independence models over variable sets $\{\mathbf{O}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ is a family of intervention targets such that $\mathbf{I}_i \subseteq \mathbf{O}_i \quad \forall i$, then a SMCM $\mathcal{S}$ is a **possibly underlying SMCM** for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ iff:*

$$\forall X, Y, \mathbf{Z} \subseteq \mathbf{O}_i,\ [X \text{ is } m\text{-separated from } Y \text{ given } \mathbf{Z} \text{ in } \mathcal{S}^{\mathbf{I}_i}] \Leftrightarrow X \perp\!\!\!\perp Y \mid \mathbf{Z} \in \mathcal{J}_i,$$

Intuitively, $\mathcal{S}$ is a SMCM such that once the effects of manipulations are modeled (i.e. $\mathcal{S}^{\mathbf{I}_i}$ is constructed), it entails all and only the independencies $\mathcal{J}_i$ observed in the corresponding data set. Thus, $\mathcal{S}$ is a possible causal model that explains all data. Since each independence model $\mathcal{J}_i$ can be graphically represented by a PAG $\mathcal{P}_i$, one can recast this definition in graph-theoretic terms: $\mathcal{S}$ is a possibly underlying SMCM if, after graph surgery, results in a marginal MAG that belongs in $\mathcal{P}_i$:

Theorem 14 *If $\mathcal{S}$ is a SMCM, $\{\mathcal{J}_i\}_{i=1}^N$ is a family of independence models, $\{\mathbf{I}_i\}_{i=1}^N$ is a family of intervention targets and $\mathcal{P}_i$ is the PAG of the Markov equivalence class of MAGs faithful to $\mathcal{J}_i$, the following statements are equivalent:*

- *$\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
- *$\mathcal{M}_i \in \mathcal{P}_i \quad \forall i$, where $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$.*

Proof See Appendix A. ■

As mentioned above, PAGs $\mathcal{P}_i$ here denote the maximally informative representatives of the Markov equivalence class of MAGs that entail independence models $\mathcal{J}_i$, instead of the *interventional* Markov equivalence class of MAGs that entail both $\mathcal{J}_i$ and the interventional constraints following the manipulation of targets $\mathbf{I_i}$. Hence, this graphical criterion may seem incomplete, since the actual MAGs belong to thinner equivalence classes, which include some additional orientations: tails towards any manipulated variable and additional orientations stemming from the combination of m-separation and acyclicity with these aforementioned tails. However, MAGs $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$ are constructed after graph surgery has been applied to the (candidate) possibly underlying SMCM and abide by definition the constraints that correspond to interventional information (i.e. tail orientations towards manipulated variables), since $\mathcal{S}^{\mathbf{I}_i}$ and $\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})$ share the same ancestral relations. Thus, the resulting MAGs $\mathcal{M}_i$ belong (by construction) to the thinner *interventional* Markov equivalence class of MAGs, and testing Markov equivalence

in the observational sense is a sound and complete graphical criterion to determine whether a SMCM is possibly underlying for a family of independence models coupled with a family of intervention targets.

Notice that PAG $\mathcal{P}_i$ can be learnt with a sound and complete algorithm such as FCI. We can now benefit by the compact representation of Markov equivalence classes of MAGs described in Theorem 7, to check whether a SMCM $\mathcal{S}$ is possibly underlying for a family of independence models $\{\mathcal{J}_i\}_{i=1}^N$ and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$: Instead of checking *all* conditional dependencies (resp. independencies) in $\mathcal{J}_i$ to be m-connections (resp. m-separations) in the corresponding SMCM $\mathcal{S}^{\mathbf{I}_i}$, we can construct the corresponding MAGs $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$ for each experiment and check whether they belong to the Markov equivalence class represented by $\mathcal{P}_i$. By Theorem 7, we only need to check adjacencies and colliders with order.

In the next section, we present an algorithm that converts the problem of identifying a SMCM $\mathcal{S}$ that is possibly underlying for a family of observed independence models $\{\mathcal{J}_i\}_{i=1}^N$ and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$ into a constraint satisfaction problem. Specifically, we will create a satisfiability instance s.t. a SMCM is possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ if and only if it corresponds to a truth-setting assignment for the SAT instance. For a family of independence models $\{\mathcal{J}_i\}_{i=1}^N$ and a family of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$, several SMCMs may be possibly underlying. We can then use the equivalent SAT instance to query properties shared by all possibly underlying SMCMs, or to identify a single possibly underlying SMCM with some desirable characteristics. In this work, we use the equivalent SAT instance to identify all edges and endpoints that are invariant in all possibly underlying SMCMs.

4.1 Conversion to SAT

Theorem 14 implies that $\mathcal{M}_i$ has the same edges (adjacencies), and the same colliders with order (unshielded colliders and discriminating colliders with order) as any MAG in $\mathcal{P}_i$, for all i. We impose these constraints on $\mathcal{S}$ by converting them to a SAT instance. We express the constraints in terms of the following **core** variables, denoting edges and orientations in any possibly underlying SMCM $\mathcal{S}$.

- edge(X, Y): true if X and Y are adjacent in $\mathcal{S}$, false otherwise.
- tail(X, Y): true if there exists an edge between X and Y in $\mathcal{S}$ that is out of Y, false otherwise.
- arrow(X, Y): true if there exists an edge between X and Y in $\mathcal{S}$ that is into Y, false otherwise.

Variables tail and arrow are not mutually exclusive, enabling us to represent $X \leftrightarrows Y$ edges when $tail(Y, X) \wedge arrow(Y, X)$. Each independence model $\mathcal{J}_i$ is entailed by the (non) adjacencies and (non) colliders in each observed PAG $\mathcal{P}_i$. These structural characteristics correspond to paths in any possibly underlying SMCM as follows:

1. $\forall X, Y \in \mathbf{O}_i$, X and Y are adjacent in $\mathcal{P}_i$ if and only if there exists an inducing path between X and Y with respect to $\mathbf{L_i}$ in $\mathcal{S}^{\mathbf{I}_i}$ (by Theorems 9 and 10 and Proposition 12).

$$\textbf{adjacent}(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} : inducing(p_{XY}, i)$$

/* X and Y are **adjacent** in $\mathcal{P}_i$ iff
there exists an inducing path from X to Y with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. */

$$\textbf{collider}(\langle X, Y, Z\rangle, \mathcal{P}_i) \leftrightarrow \neg ancestor(Y, X, i) \wedge \neg ancestor(Y, Z, i)$$

/* Triple $\langle X, Y, Z\rangle$ is **collider** in $\mathcal{P}_i$ iff
Y is not an ancestor of X or Z in $\mathcal{S}^{\mathbf{I}_i}$. */

$$\textbf{unshielded}(\langle X, Y, Z\rangle, \mathcal{P}_i) \leftrightarrow adjacent(X, Y, \mathcal{P}_i) \wedge adjacent(Y, Z, \mathcal{P}_i) \wedge \neg adjacent(X, Z, \mathcal{P}_i)$$

/* Triple $\langle X, Y, Z\rangle$ is **unshielded** in $\mathcal{P}_i$ iff
(X, Y), (Y, Z) are adjacent in $\mathcal{P}_i$, (X, Z) are not adjacent in $\mathcal{P}_i$. */

$$\begin{aligned}\textbf{discriminating}(\langle V_0, \ldots, V_{n-1}, V_n, V_{n+1}\rangle, V_n, \mathcal{P}_i) \leftrightarrow \\ \neg adjacent(V_0, V_{n+1}, \mathcal{P}_i) \wedge \forall j \in [0, \ldots, n] adjacent(V_j, V_{j+1}, \mathcal{P}_i) \wedge \\ \forall j \in [1, \ldots, n-1]\big(collider(\langle V_{j-1}, V_j, V_{j+1}\rangle, \mathcal{P}_i) \\ \wedge adjacent(V_j, V_{n+1}, \mathcal{P}_i) \wedge ancestral(V_j, V_{n+1}, i)\big)\end{aligned}$$

/* Path $\langle V_0, \ldots, V_{n+1}\rangle$ is **discriminating** for V_n in $\mathcal{P}_i$ iff
V_0, V_{n+1} are not adjacent in $\mathcal{P}_i$, $V_0, \ldots, V_{n+1}$ is a path in $\mathcal{P}_i$,
every node between V_0 and V_n is a collider on the path
and a parent of V_{n+1} in $\mathcal{P}_i$. */

Figure 4: Formulae relating properties of observed PAGs to the underlying SMCM $\mathcal{S}$. In each PAG, all features that are necessary and sufficient for Markov equivalence impose constraints on possibly underlying SMCMs. Constraints are expressed using the literals and formulae introduced here. Index i is used to denote properties of an underlying SMCM in experiment i, where variables $\mathbf{L}_i$ are latent and variables $\mathbf{I}_i$ are manipulated. We use use p_{XY} to denote a path between X and Y in $\mathcal{S}$. Conjunction and disjunction are assumed to have precedence over implication with regard to bracketing. Each formula is followed by an explanation in natural language (in star-slash comments).

$$\textbf{inducing}(\langle V_0,\ldots,V_{n+1}\rangle, i) \leftrightarrow$$
$$(n=0 \rightarrow edge(V_0, V_{n+1})) \wedge$$
$$\big(n>0 \rightarrow (\forall j \in [1,\ldots,n]\ unblocked(\langle V_{j-1}, V_j, V_{j+1}\rangle, V_0, V_{n+1}, i))\big) \wedge$$
$$(V_0 \in \mathbf{I}_i \rightarrow tail(V_1, V_0)) \wedge (Y \in \mathbf{I}_i \rightarrow tail(V_n, V_{n+1}))$$

/* Path $\langle V_0,\ldots,V_{n+1}\rangle$ is **inducing** with respect to $\mathbf{L_i}$ in $\mathcal{S}^{\mathbf{I}_i}$ iff
if the path has only two variables, V_0 is adjacent to V_n in $\mathcal{S}$
else each triple is **unblocked** for the endpoints with respect to $\mathbf{L_i}$,
if V_0 (V_{n+1}) is manipulated in i then the path is out of V_0 (V_{n+1}) in $\mathcal{S}$. */

$$\textbf{unblocked}(\langle Z, V, W\rangle, X, Y, i) \leftrightarrow$$
$$edge(Z, V) \wedge edge(V, W) \wedge$$
$$[V \in \mathbf{L}_i \rightarrow \neg head2head(\langle Z, V, W\rangle, i) \vee ancestor(V, X, i) \vee ancestor(V, Y, i)] \wedge$$
$$[V \notin \mathbf{L}_i \rightarrow head2head(\langle Z, V, W\rangle, i) \wedge (ancestor(V, X, i) \vee ancestor(V, Y, i))]$$

/* Triple $\langle Z, V, W\rangle$ is **unblocked** for X, Y with respect to $\mathbf{L_i}$ iff
(Z, V) (V, W) are adjacent in $\mathcal{S}$
if V is latent, if V is **head2head** then it is an ancestor of X or Y in $\mathcal{S}^{\mathbf{I}_i}$
if V is not latent, V is a **head2head** and an ancestor of X or Y in $\mathcal{S}^{\mathbf{I}_i}$. */

$$\textbf{head2head}(\langle X, Y, Z\rangle, i) \leftrightarrow Y \notin \mathbf{I}_i \wedge arrow(X, Y) \wedge arrow(Z, Y)$$

/* Triple $\langle X, Y, Z\rangle$ is **head2head** in $\mathcal{S}^{\mathbf{I}_i}$ iff
Y is not manipulated in experiment i, X is into Y, Z is into Y in $\mathcal{S}$. */

$$\textbf{ancestor}(X, Y, i) \leftrightarrow \exists p_{XY} : ancestral(p_{XY}, i)$$

/* X is an **ancestor** of Y in experiment i iff
there exists an ancestral path from X to Y in $\mathcal{S}^{\mathbf{I}_i}$. */

$$\textbf{ancestral}(\langle V_0,\ldots,V_{n+1}\rangle, i) \leftrightarrow$$
$$\forall j \in [1,\ldots,n+1]\big(V_j \notin \mathbf{I}_i \wedge (edge(V_{j-1}, V_j) \wedge tail(V_j, V_{j-1}) \wedge arrow(V_{j-1}, V_j))\big)$$

/* Path $\langle V_0,\ldots,V_{n+1}\rangle$ is **ancestral** in experiment i iff
every variable (apart from possibly V_0) is not manipulated in $\mathcal{S}^{\mathbf{I}_i}$
every variable is a parent of the next in $\mathcal{S}$. */

Figure 5: Formulae reducing path properties of the graphs $\mathcal{S}^{\mathbf{I}_i}$ to the core variables: Graph properties of $\mathcal{S}$ in each experiment, inferred by the observed PAGs using the formulae in Figure 4, are now expressed as boolean formulae using the "core" variables *edge*, *arrow* and *tail*. Index i is used to denote properties of an underlying SMCM in experiment i, where variables $\mathbf{L}_i$ are latent and variables $\mathbf{I}_i$ are manipulated. Conjunction and disjunction are assumed to have precedence over implication with regard to bracketing. Each formula is followed by an explanation in in natural language (in star-slash comments).

2. If $\langle X, Y, Z\rangle$ is an unshielded definite non collider in $\mathcal{P}_i$, then $\langle X, Y, Z\rangle$ is an unshielded triple in $\mathcal{P}_i$ and Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).

3. If $\langle X, Y, Z\rangle$ is an unshielded collider in $\mathcal{P}_i$, then $\langle X, Y, Z\rangle$ is an unshielded triple in $\mathcal{P}_i$ and Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).

4. If $\langle W, \ldots, X, Y, Z\rangle$ is a discriminating collider in $\mathcal{P}_i$, then $\langle W \ldots, X, Y, Z\rangle$ is a discriminating path for Y in $\mathcal{P}_i$ and Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).

5. If $\langle W, \ldots, X, Y, Z\rangle$ is a discriminating definite non collider in $\mathcal{P}_i$, then $\langle W \ldots, X, Y, Z\rangle$ is a discriminating path for Y in $\mathcal{P}_i$ and Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$ (by the semantics of edges in MAGs).

These constraints are expressed using the core variables (edges, tails and arrows), as described in Figures 4 and 5. Figure 4 describes how features of a PAG are imposed as path constraints in a possibly underlying SMCM. More specifically, an adjacency, a tail and an arrowhead in a PAG $\mathcal{P}_i$ correspond to an inducing path, a causal ancestry and the lack of causal ancestry on any possibly underlying SMCM, respectively. Unshielded triples and discriminating paths are expressed on the basis of these basic PAG features. In each PAG, the observed features depend on the latent and manipulated variables. When constraints are imposed on the candidate underlying SMCMs, the latent and manipulated variables in the experiment are taken under consideration: If an adjacency is observed in $\mathcal{P}_i$ in experiment i, where variables $\mathbf{L}_i$ are latent and $\mathbf{I}_i$ are manipulated, then any path on $\mathcal{S}$ that explains this adjacency must be inducing with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. Any truth-assignment to the core variables that does not entail the presence of such an inducing path should not satisfy the SAT instance. The following constraints are added to ensure that the graphs satisfying constraints 1-5 above are SMCMs:

6. $\forall X, Y \in \mathbf{O}$, either X is not an ancestor of Y or Y is not an ancestor of X in $\mathcal{S}$ (no directed cycles).

7. $\forall X, Y \in \mathbf{O}$, at most one of $tail(X, Y)$ and $tail(Y, X)$ can be true (no selection bias).

8. $\forall X, Y \in \mathbf{O}$, at least one of $tail(X, Y)$ and $arrow(Y, X)$ must be true.

Naturally, Constraints 7 and 8 are meaningful only if X and Y are adjacent (if edge(X, Y) is true), and redundant otherwise.

4.2 Algorithm COmbINE

We now present algorithm **COmbINE** (Causal discovery from Overlapping INtErventions) that learns causal features from multiple, heterogeneous data sets. The algorithm takes as input a set of data sets $\{\mathbf{D}_i\}_{i=1}^N$ over a set of overlapping variable sets $\{\mathbf{O}_i\}_{i=1}^N$. In each data set, a (possibly empty) subset of the observed variables $\mathbf{I}_i \subset \mathbf{O}_i$ may be manipulated. Each data set entails an independence model $\mathcal{J}_i$. FCI is run on each data set and the corresponding PAGs $\{\mathcal{P}_i\}_{i=1}^N$ are produced. The algorithm then creates a candidate underlying SMCM $\mathcal{H}_{in}$. Subsequently, for each PAG $\mathcal{P}_i$, the features of $\mathcal{P}_i$ are translated into

Algorithm 2: COmbINE

input : data sets $\{\mathbf{D}_i\}_{i=1}^N$, sets of intervention targets $\{\mathbf{I}_i\}_{i=1}^N$, FCI parameters *params*, maximum path length *mpl*, conflict resolution strategy *str*

output: Summary graph $\mathcal{H}$

1 **foreach** i **do** $\mathcal{P}_i \leftarrow$ `FCI`$(\mathbf{D}_i, \textit{params})$;
2 $\mathcal{H}_{in} \leftarrow$ `initializeSMCM` $(\{\mathcal{P}_i\}_{i=1}^N)$;
3 $(\Phi, \mathcal{F}) \leftarrow$ `addConstraints` $(\mathcal{H}, \{\mathcal{P}_i\}_{i=1}^N, \{\mathbf{I}_i\}_{i=1}^N, \textit{mpl})$;
4 $\mathcal{F}' \leftarrow$ select a subset of non-conflicting literals $\mathcal{F}'$ according to strategy *str*;
5 $\mathcal{H} \leftarrow$ `backBone` $(\Phi \wedge \mathcal{F}')$

constraints, expressed in terms of edges and endpoints in $\mathcal{H}_{in}$, using the formulae in Figures 4 and 5 . In the sample limit (and under the assumptions discussed above), the SAT formula $\Phi \wedge \mathcal{F}$ produced by this procedure is satisfied by all and only the possibly underlying SMCMs for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}$. In the presence of statistical errors, however, $\Phi \wedge \mathcal{F}$ may be unsatisfiable. To handle conflicts, the algorithm takes as input a strategy for selecting a non-conflicting subset of constraints $\mathcal{F}'$ and ignores the rest. Finally, COmbINE queries the SAT formula for variables that have the same truth-value in all satisfying assignments, translates them into graph features, and returns a graph that summarizes the invariant edges and orientations of all possibly underlying SMCMs. In the rest of this paper we call the graphical output of COmbINE a **summary graph**.

The pseudocode for COmbINE is presented in Algorithm 2. Apart from the set of data sets described above, COmbINE takes as input the chosen parameters for FCI (threshold α, maximum conditioning set $maxK$), the maximum length of paths to consider and a strategy for selecting a subset of non-conflicting constraints.

Initially, the algorithm runs FCI on each data set $\mathbf{D}_i$ and produces the corresponding PAG $\mathcal{P}_i$. Then the candidate SMCM $\mathcal{H}_{in}$ is initialized: $\mathcal{H}_{in}$ is the graph upon which all path constraints will be imposed. Path constraints are realized on the basis of the *plausible configurations* of $\mathcal{H}_{in}$. We say that a path p in $\mathcal{H}_{in}$ is **possibly inducing with respect to L**, if we can create a graph $\mathcal{H}'_{in}$ by orienting circle endpoints in $\mathcal{H}_{in}$ such that path p is inducing with respect to $\mathbf{L}$ in $\mathcal{H}'_{in}$. We say that a path p in $\mathcal{H}_{in}$ is **possibly ancestral**, if we can create a graph $\mathcal{H}'_{in}$ by orienting circle endpoints in $\mathcal{H}_{in}$ such that path p is ancestral $\mathcal{H}'_{in}$. To ensure the soundness of the algorithm, if p is an inducing (ancestral) path in $\mathcal{S}$, it must be a possibly inducing (ancestral) path in $\mathcal{H}_{in}$. Thus, $\mathcal{H}_{in}$ must have at least a superset of edges and at most a subset of orientations of any possibly underlying SMCM $\mathcal{S}$.

An obvious–yet not very smart–choice for $\mathcal{H}_{in}$ would be the complete unoriented graph. However, looking for possibly inducing and possibly ancestral paths on the complete unoriented graph over the union of variables could make the problem intractable even for small input sizes. To reduce the number of possibly inducing and possibly ancestral paths, we use Algorithm 3 to construct $\mathcal{H}_{in}$.

Algorithm 3 constructs a graph $\mathcal{H}_{in}$ that has all edges observed in any PAG $\mathcal{P}_i$ as well as some additional edges that would not have been observed even if they existed: Edges connecting variables that have never been observed together, and edges connecting variables that have been observed together, but at least one of them was manipulated in each joint

Algorithm 3: initializeSMCM

input : PAGs $\{\mathcal{P}_i\}_{i=1}^N$
output: initial graph $\mathcal{H}_{in}$

1 $\mathcal{H}_{in} \leftarrow$ empty graph over $\cup \mathbf{O}_i$;
2 **foreach** i **do**
3 | $\mathcal{H}_{in} \leftarrow$ add all edges in $\mathcal{P}_i$ unoriented;
4 **end**
5 Orient only arrowheads that are present in every $\mathcal{P}_i$;
/* Add edges between variables never measured unmanipulated together */
6 **foreach** *pair* X, Y *of non-adjacent nodes* **do**
7 | **if** $\nexists i$ *s.t.* $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$ **then**
8 | | add $X \circ\!\!-\!\!\circ Y$ to $\mathcal{H}_{in}$;
9 | | **if** $\exists i$ *s.t.* $X, Y \in \mathbf{O}_i$, $X \in \mathbf{I}_i$, $Y \notin \mathbf{I}_i$ **then** add arrow into X;
10 | | **if** $\exists i$ *s.t.* $X, Y \in \mathbf{O}_i$, $Y \in \mathbf{I}_i$, $X \notin \mathbf{I}_i$ **then** add arrow into Y;
11 | **end**
12 **end**

appearance in a data set. For example, variables $X9$ and $X15$ in Figure 6 are measured together in two data sets: $\mathbf{D}_2$ and $\mathbf{D}_3$. If $X9 \dashrightarrow X15$ in the underlying SMCM, this edge would be present in $\mathcal{P}_3$. Similarly, if $X15 \dashrightarrow X9$ in the underlying SMCM, the variables would be adjacent in $\mathcal{P}_2$. We can therefore rule out the possibility of a directed edge between the two variables in $\mathcal{S}$. However, it is possible that $X15$ and $X9$ are confounded in $\mathcal{S}$, and the edge disappears by the manipulation procedure in both $\mathcal{P}_2$ and $\mathcal{P}_3$. Thus, Algorithm 3 will add these possible edges in $\mathcal{H}_{in}$. In addition, in Line 5, Algorithm 3 adds all the orientations found so far in all $\mathcal{P}_i$'s that are invariant.[1] The resulting graph has, in the sample limit, a superset of edges and a subset of orientations compared to the actual underlying SMCM. Lemma 15 in Appendix A formalizes and proves this property.

Having initialized the search graph, Algorithm 2 proceeds to generate the constraints. This procedure is described in detail in Algorithm 4, that is the core of COmbINE. These are: (i) the bi-conditionals regarding the presence/absence of edges (Line 4), (ii) conditionals regarding unshielded and discriminating colliders (Lines 14, 13, 20 and 19), (iii) constraints that ensure that any truth-setting assignment is a SMCM, i.e., it has no directed cycles and that every edge has at least one arrowhead (Lines 8 and 9 respectively). Literal *col* (resp. *dnc*) is used to represent both unshielded and discriminating colliders (resp. unshielded and discriminating non colliders).

The constraints are realized on the basis of the *plausible* configurations of $\mathcal{H}_{in}$: Thus, for the constraints corresponding to $adjacent(X, Y, i)$ the algorithm finds all paths between

1. Other options would be to keep all non-conflicting arrows, or keep non-conflicting arrows and tails after some additional analysis on definitely visible edges (see Zhang 2008b, Borboudakis et al. 2012 for more on this subject). These options are asymptotically correct and would constrain search even further. Nevertheless, orientation rules in FCI seem to be prone to error propagation and we chose a more conservative strategy giving a chance to the conflict resolution strategy to improve the learning quality. Naturally, if an oracle of conditional independence is available or there is a reason to be confident on certain features, one can opt to make additional orientations.

Algorithm 4: addConstraints

input : $\mathcal{H}_{in}$, $\{\mathcal{P}_i\}_{i=1}^N$, $\{\mathbf{I}_i\}_{i=1}^N$, *mpl*
output: Φ, list of literals $\mathcal{F}$

$\Phi \leftarrow \emptyset$ **foreach** X, Y **do**
  foreach i **do**
    posIndPaths$\leftarrow$ paths in $\mathcal{H}_{in}$ of maximum length *mpl* that are possibly inducing with respect to $\mathbf{L}_i$;
    $\Phi \leftarrow \Phi \wedge \big[adjacent(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} \in$**posIndPaths** s. t. $inducing(p_{XY}, i)\big]$;
    if *X, Y are adjacent in* $\mathcal{P}_i$ **then** add $adjacent(X, Y, \mathcal{P}_i)$ to $\mathcal{F}$;
    else add $\neg adjacent(X, Y, \mathcal{P}_i)$ to $\mathcal{F}$;
  end
  $\Phi \leftarrow \Phi \wedge \big[\neg ancestor(X, Y) \vee \neg ancestor(Y, X)\big]$;
  $\Phi \leftarrow \Phi \wedge \big[\neg tail(X, Y) \vee \neg tail(Y, X)\big] \wedge \big[(arrow(X, Y) \vee tail(X, Y)\big]$;
end
foreach i **do**
  foreach *unshielded triple in* $\mathcal{P}_i$ **do**
    $\Phi \leftarrow \Phi \wedge \big[col(X, Y, Z, \mathcal{P}_i) \rightarrow unshielded(X, Y, Z, \mathcal{P}_i) \wedge collider(X, Y, Z, \mathcal{P}_i)\big]$;
    $\Phi \leftarrow \Phi \wedge \big[dnc(X, Y, Z, \mathcal{P}_i) \rightarrow unshielded(X, Y, Z, \mathcal{P}_i) \wedge \neg collider(X, Y, Z, \mathcal{P}_i)\big]$;
    if $\langle X, Y, Z\rangle$ *is a collider in* $\mathcal{P}_i$ **then** add $col(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
    else add $dnc(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
  end
  foreach *discriminating path* $p_{WZ} = \langle W, \ldots, X, Y, Z\rangle$ **do**
    $\Phi \leftarrow \Phi \wedge \big[col(X, Y, Z, \mathcal{P}_i) \rightarrow discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge collider(X, Y, Z, \mathcal{P}_i)\big]$;
    $\Phi \leftarrow \Phi \wedge \big[dnc(X, Y, Z, \mathcal{P}_i) \rightarrow discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge \neg collider(X, Y, Z, \mathcal{P}_i)\big]$;
    if *X, Y, Z is a collider in* $\mathcal{P}_i$ **then** add $col(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
    else add $dnc(X, Y, Z, \mathcal{P}_i)$ to $\mathcal{F}$;
  end
end

X and Y in $\mathcal{H}_{in}$ that are possibly inducing. Then, for the literal $adjacent(X, Y, i)$ to be true, at least one of these paths is constrained to be inducing; for the opposite, none of these paths is allowed to be inducing. This step is the most computationally expensive part of the algorithm. The parameter *mpl* controls the length of the possibly inducing paths; instead of finding *all* paths between X and Y that are possibly inducing, the algorithm looks for all paths of length at most *mpl*. This plays a major part in the ability of the algorithm to scale up, since finding all possible paths between every pair of variables can blow up even in relatively small networks, particularly in the presence of unoriented cliques or in relatively dense networks.

Notice that the information on manipulations is included in the satisfiability instance through the encoding of the constraints: For every adjacency between X and Y observed

in $\mathcal{P}_i$, the plausible inducing paths are consistent with the respective intervention targets: No inducing path is allowed to include an edge that is incoming to a manipulated variable.

As an example, consider the following variation of the instance presented in Figure 7. Assume that variable X is manipulated in experiment 1, and no variable is manipulated in experiment 2. Since no information concerning experiments is employed up to the initialization of the search graph, the resulting PAGs are the $\mathcal{P}_1$ and $\mathcal{P}_2$ shown in Figure 7. Thus, in the initial search graph $\mathcal{H}_{in}$, $X \circ\!\!-\!\!\circ Y$ and $X \circ\!\!-\!\!\circ Z \circ\!\!-\!\!\circ Y$ are the two possibly inducing paths for X and Y in experiment i. Then the following constraint will be imposed:

$$adjacent(X, Y, 1) \leftrightarrow inducing(\langle X, Y\rangle, 1) \vee inducing(\langle X, Z, Y\rangle, 1)$$

For path $\langle X, Y\rangle$, the corresponding constraint is reduced to the properties of $\mathcal{S}$ as follows:

$$inducing(\langle X, Y\rangle, 1) \leftrightarrow \\ (X \in \mathbf{I}_1 \rightarrow tail(Y, X)) \wedge (Y \in \mathbf{I}_1 \rightarrow tail(X, Y)) \wedge edge(X, Y)$$

which is then added in Φ as $inducing(\langle X, Y\rangle, 1) \leftrightarrow tail(Y, X) \wedge edge(X, Y)$ since $X \in \mathbf{I}_1$ is true and $Y \in \mathbf{I}_1$ is false. For the path $\langle X, Z, Y\rangle$ the corresponding constraint finally added in Φ is

$$inducing(\langle X, Z, Y\rangle) \leftrightarrow \\ tail(Z, X) \wedge [\neg head2head(\langle X, Z, Y\rangle) \vee ancestral(Z, X) \vee ancestral(Z, Y)]$$

Thus, in a SMCM that satisfies the final formula, if $inducing(\langle X, Y\rangle, i)$ is true, there will be an inducing path from X to Y consistent with the manipulation information.

Also notice how this constraint is *propagated* in the SAT: For example, $X \star\!\!-\!\!\star Z \star\!\!-\!\!\star Y \star\!\!-\!\!\star W$ is a plausible skeleton for a possibly underlying SMCM. By the constraints mentioned above, $X \rightarrow Z \star\!\!-\!\!\star Y$ is the inducing path for X and Y with respect to $L_1 = Z$. By the constraints added for the definite non collider $\langle X, Z, W\rangle$ for $\mathcal{P}_2$, Z has to be an ancestor of either X or Y in $\mathcal{S}^{\emptyset}$. Therefore, the path $Z \star\!\!-\!\!\star Y \star\!\!-\!\!\star W$ has to be an ancestral path in $\mathcal{S}$, which implies that $Y \dashrightarrow Z$ in $\mathcal{S}$. Thus, the orientation $Y \dashrightarrow Z$ is imposed by a combination of constraints stemming from different PAGs, for two variables never jointly measured.

As mentioned above, in the absence of statistical errors, all the constraints stemming from all PAGs $\mathcal{P}_i$ are simultaneously satisfiable. In practical settings however, it is possible that some of the PAGs have some erroneous features due to statistical errors, and these features can lead to conflicting constraints. To tackle this problem, Algorithm 4 uses the following technique: For every observed feature, instead of imposing the implied constraints on the formula Φ, the algorithm adds a bi-conditional connecting the feature to the constraints. For example, if X and Y are found adjacent in $\mathcal{P}_i$, then instead of adding the constraints $\exists p_{XY} : inducing(X, Y, i)$ to Φ, we add the bi-conditional $adjacent(X, Y, \mathcal{P}_i) \leftrightarrow \exists p_{XY} : inducing(X, Y, i)$. The antecedents of the conditionals are stored in a list of literals $\mathcal{F}$. The conflict resolution strategy is then imposed on this list of literals, selecting a subset $\mathcal{F}'$ that results in a satisfiable SAT formula $\Phi \wedge \mathcal{F}'$. The formula $\Phi \wedge \mathcal{F}'$ is expressed in Conjunctive Normal Form (CNF) so it can be input to standard SAT solvers.

Recall that the propositional variables of Φ correspond to the features of the actual underlying SMCM (its edges and endpoints). Some of these variables have the same value

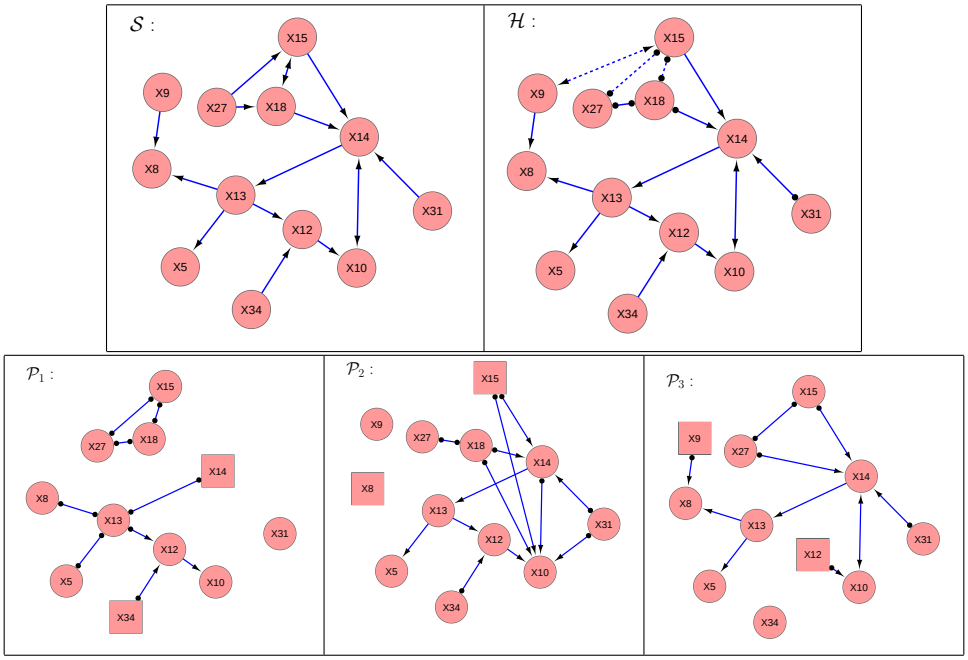


Figure 6: An example of COmbINE input - output. Graph $\mathcal{S}$ is the actual, data-generating, underlying SMCM over 12 variables. PAGs $\mathcal{P}_1, \mathcal{P}_2$ and $\mathcal{P}_3$ are the output of FCI ran with an oracle of conditional independence on three different marginals of $\mathcal{G}$. $\mathcal{H}$ is the output of COmbINE algorithm. The sets of latent variables (with respect to the union of observed variables) per data set are: $\mathbf{L}_1 = \{X9\}$, $\mathbf{L}_2 = \{\emptyset\}$, $\mathbf{L}_3 = \{X18\}$. The sets of manipulated variables (annotated as rectangle nodes instead of circles in the respective graphs) are: $\mathbf{I}_1 = \{X14, X34\}$, $\mathbf{I}_2 = \{X15, X8\}$, $\mathbf{I}_3 = \{X9, X12\}$. Notice that $X10$ and $X31$ are adjacent in $\mathcal{P}_2$, but not in $\mathcal{P}_1$ or $\mathcal{P}_3$. This happens because there exists an inducing path in the underlying SMCM ($X31 \longrightarrow X14 \longleftrightarrow X10$ in $\mathcal{S}$) that is "broken" by the manipulation of $X14$ and $X12$, respectively. Also notice a dashed edge between $X9$ and $X15$, which cannot be excluded since the variables have never been observed unmanipulated together. Even if the link existed, it would be destroyed in both $\mathcal{P}_2$ and $\mathcal{P}_3$, where both variables are observed. All graphs were visualized in Cytoscape (Smoot et al., 2011).

in all the possible truth-setting assignments of $\Phi \wedge \mathcal{F}'$, meaning the respective features are invariant in all possibly underlying SMCMs. Such variables are called **backbone** variables of $\Phi \wedge \mathcal{F}'$ (Hyttinen et al., 2013). The actual value of a backbone variable is called the polarity of the variable. For sake of brevity, we say an edge or endpoint has polarity 0/1 if the corresponding variable is a backbone variable in $\Phi \wedge \mathcal{F}'$ and has polarity 0/1. Based on the backbone of $\Phi \wedge \mathcal{F}'$, the final step of COmbINE is to construct the summary graph $\mathcal{S}$. $\mathcal{S}$ has the following types of edges and endpoints:

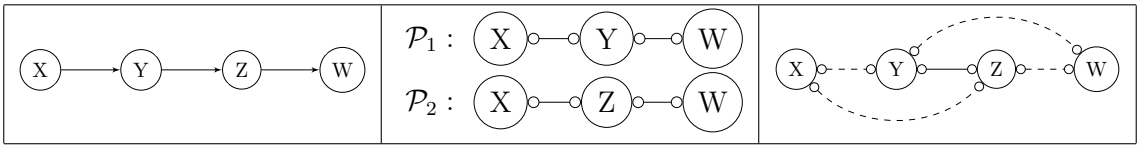


Figure 7: A detailed example of a non-trivial inference. From left to right: The true underlying SMCM over variables X, Y, Z, W; PAGs $\mathcal{P}_1$ and $\mathcal{P}_2$ over $\{X, Y, W\}$ and $\{X, Z, W\}$, respectively; The output $\mathcal{H}$ of Algorithm 2 ran with an oracle of conditional independence. Notice that, the edges in $\mathcal{P}_1$ can not both simultaneously occur in a consistent SMCM $\mathcal{S}$: This would make $X \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\circ W$ an inducing path for X and W with respect to $\mathbf{L}_2 = \{Y\}$ and contradict the features of $\mathcal{P}_2$, where X and W are not adjacent. Similarly, $X \circ\!\!-\!\!\circ Z \circ\!\!-\!\!\circ W$ cannot occur in any possibly underlying SMCM $\mathcal{S}$. The only possible edge structures that explain all the observed adjacencies and definite non colliders are $X \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\circ Z \circ\!\!-\!\!\circ W$ or $X \circ\!\!-\!\!\circ Z \circ\!\!-\!\!\circ Y \circ\!\!-\!\!\circ W$. Either way, Y and Z share an edge in all consistent SMCMs, and the algorithm will predict a solid edge between Y and Z, even if the two have not been measured in the same data set. This example is discussed in detail in (Tsamardinos et al., 2012).

- **Solid Edges:** Edges in $\mathcal{H}$ that have polarity 1 in $\Phi \wedge \mathcal{F}'$, meaning that they are present in all possibly underlying SMCMs.
- **Absent Edges:** Edges that are not in $\mathcal{H}$ or edges in $\mathcal{H}$ that have polarity 0 in $\Phi \wedge \mathcal{F}'$, meaning that they are absent in all possibly underlying SMCMs.
- **Dashed Edges:** Edges in $\mathcal{H}$ that are not backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that there exists at least one possibly underlying SMCM where this edge is present and one where this edge is absent.
- **Solid Endpoints:** Endpoints in $\mathcal{H}$ that are backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that this orientation is invariant in all possibly underlying SMCMs.
- **Dashed (circled) Endpoints:** Endpoints in $\mathcal{H}$ that are not backbone variables in $\Phi \wedge \mathcal{F}'$, meaning that there exists at least one possibly underlying SMCM where this orientation does not hold.

We use the term **solid features** of the summary graph to denote the set of solid edges, absent edges and solid endpoints of the summary graph.

Overall, *Algorithm 2 takes as input a set of data sets and a list of parameters and outputs a summary graph that has all invariant edges and orientations of the SMCMs that satisfy as many constraints as possible (according to some strategy).* The algorithm is capable of non-trivial inferences, like for example the presence of a solid edge among variables never measured together. Figures 6 and 7 illustrate the output of Algorithm 2, along with the corresponding input PAGs.

We claim that, given an oracle of conditional independence, the SAT-generating procedure described in Algorithm 4 results in a SAT instance $\Phi \wedge \mathcal{F}$ that is satisfied by all

and only the possibly underlying SMCMs for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ (i.e., every SMCM that entails the exact same conditional independencies as those obtained by the oracle for every experiment, after the removal of edges incoming to the manipulated variables). Lemma 17 proves that the every possibly underlying SMCM satisfies $\Phi \wedge \mathcal{F}$, while Lemma 19 proves that if $\mathcal{S}$ is a mixed graph satisfying $\Phi \wedge \mathcal{F}$, $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.

In all subsequent lemmas, theorems and proofs we employ the assumptions A1-A3 and the notation presented in the beginning of Section 4. We also assume the algorithms are run with an oracle of conditional independence and infinite maximum conditioning set size and maximum path length. We only present the main lemmas and theorems here. Auxiliary lemmas and all proofs can be found in Appendix A.

Lemma 17 *For an oracle of conditional independence, if $\mathcal{S}$ is a possibly underlying model for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, and $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, $\mathcal{S}$ satisfies $\Phi \wedge \mathcal{F}$.*

Proof See Appendix A. ■

Lemma 19. *For an oracle of conditional independence, if $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, and $\mathcal{S}$ a mixed graph that satisfies $\Phi \wedge \mathcal{F}$, then $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*

Proof See Appendix A. ■

Soundness and completeness of Algorithm 2 stems from the Lemmas 17 and 19: For the summary graph that is the output of COmbINE soundness means that if a feature is solid in $\mathcal{H}$, the feature is present in all possibly underlying SMCMs for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$. Completeness means that if a feature is dashed in $\mathcal{H}$, there exists at least two possibly underlying SMCM where this feature has different truth values. Since $\Phi \wedge \mathcal{F}$ implicitly represents the entire solution space, and it is satisfied by all and only the possibly underlying SMCMs, soundness and completeness of Algorithm 2 easily follows.

Theorem 20 ***(Soundness and completeness of Algorithm 2)*** *If $\mathcal{H}$ is the output of Algorithm 2, then the following hold:*
Soundness*: If a feature (edge, absent edge, endpoint) is* solid *in $\mathcal{H}$, then this feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
Completeness*: If a feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, the feature is solid in $\mathcal{H}$.*

Proof See Appendix A. ■

4.3 A Strategy for Conflict Resolution Based on the Maximum Posterior Ratio

In this section, we present a method for assigning a measure of confidence to every literal in list $\mathcal{F}$ described in Algorithm 2, and a strategy for selecting a subset of non-conflicting constraints. List $\mathcal{F}$ includes four types of literals, expressing different statistical information:

1. $adjacent(X, Y, \mathcal{P}_i)$: X and Y are not independent given any subset of $\mathbf{O}_i$.
2. $\neg adjacent(X, Y, \mathcal{P}_i)$: X and Y are independent given some $\mathbf{Z} \subset \mathbf{O}_i$
3. $col(\langle X, Y, Z\rangle, \mathcal{P}_i)$: Y is in no subset of $\mathbf{O}_i$ that renders X and Z independent.
4. $dnc(\langle X, Y, Z\rangle, \mathcal{P}_i)$: Y is in every subset of $\mathbf{O}_i$ that renders X and Z independent.

For the scope of this work, we will focus on ranking the first two types of antecedents: Adjacencies and non-adjacencies. We will then assign colliders and non-colliders with order to the same rank as the non-adjacency of the corresponding discriminating path's endpoints. Naturally, this criterion of sorting colliders and non-colliders is merely a heuristic, as more than one tests of independence are involved in deciding that a triple is a (non) collider.

Assigning a measure of likelihood or posterior probability to every single (non) adjacency would enable their comparison. A non-adjacency in a PAG corresponds to a conditional independence given some subset of the observed variables. In contrast, an adjacency corresponds to the lack of such a subset. Thus, an edge between X and Y should be present in $\mathcal{P}_i$ if the evidence (data) is less in favor of hypothesis:

$$H_0 : \exists \mathbf{Z} \subset \mathbf{O}_i : X \perp\!\!\!\perp Y \mid \mathbf{Z} \text{ than the alternative } H_1 : \nexists \mathbf{Z} \subset \mathbf{O}_i : X \perp\!\!\!\perp Y \mid \mathbf{Z} \tag{1}$$

This is a complicated set of hypotheses, that involves multiple tests of independence. We try to approximate testing by using a single test of independence as a surrogate: During FCI, several conditioning sets are tested for every pair of variables X and Y. Let $\mathbf{Z}_{XY}$ be the conditioning test for which the highest p-value is identified for the given pair of variables. Notice that it is this maximum p-value that is employed in FCI and similar algorithms to determine whether an edge is included in the output or not. We use the set of hypotheses

$$H_0 : X \perp\!\!\!\perp Y \mid \mathbf{Z}_{XY} \text{ against the alternative } H_1 : X \not\perp\!\!\!\perp Y \mid \mathbf{Z}_{XY}$$

as a surrogate for the set of hypotheses in Equation 1. Under the null hypothesis, the p-values follow a uniform $\mathcal{U}([0, 1])$ distribution,[2] also known as the $Beta(1, 1)$ distribution. Under the alternative hypothesis, the density of the p-values should be decreasing in p. One class of decreasing densities is the $Beta(\xi, 1)$ distribution for $0 < \xi < 1$, with density $f(p|\xi) = \xi p^{\xi-1}$. Thus, we can approximate the null and alternative hypotheses in terms of the p-value as

$$H_0 : p_{XY.\mathbf{Z}} \sim Beta(1, 1) \text{ against } H_1 : p_{XY.\mathbf{Z}} \sim Beta(\xi, 1) \text{ for some } \xi \in (0, 1). \tag{2}$$

Taking the Beta alternatives was presented as a method for calibrating p-values in Sellke et al. (2001). For the purpose of this work, we use them to estimate whether dependence

2. This is actually an approximation in this case, since these p-values are maximum p-values over several tests.

is more probable than independence for a given p-value p, by estimating which of the Beta alternatives it is most likely to follow.

Let $\mathcal{F}$ be a set of M literals corresponding to adjacencies and non-adjacencies, and $\{p_j\}_{j=1}^{M}$ the respective maximum p-values: If the j-th literal in $\mathcal{F}$ is $(\neg)adjacent(X, Y, \mathcal{P}_i)$, then p_j is the maximum p-value obtained for X, Y during FCI over $\mathbf{D}_i$. We assume that this population of p-values follows a mixture of $Beta(\xi, 1)$ and $Beta(1, 1)$ distribution. If π_0 is the proportion of p-values following $Beta(\xi, 1)$, the probability density function is

$$f(p|\xi, \pi_0) = \pi_0 + (1 - \pi_0)\xi p^{\xi-1}$$

and the likelihood for a set of p-values $\{p_j\}_{j=1}^{M}$ is

$$L(\xi, \pi_0) = \prod_j (\pi_0 + (1 - \pi_0)\xi p_j^{\xi-1}).$$

The respective negative log likelihood is

$$-LL(\xi, \pi_0) = -\sum_j log(\pi_0 + (1 - \pi_0)\xi p_j^{\xi-1}). \tag{3}$$

For given estimates $\hat{\pi_0}$ and $\hat{\xi}$, the posterior ratio of H_0 against H_1 is

$$E_0(p) = \frac{P(p|H_0)P(H_0)}{P(p|H_1)P(H_1)} = \frac{P(p|p \sim Beta(1, 1))P(p \sim Beta(1, 1))}{P(p|p \sim Beta(\hat{\xi}, 1))P(p \sim Beta(\hat{\xi}, 1))} = \frac{\hat{\pi_0}}{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}.$$

$E_0(p) > 1$ implies that for the test of independence represented by the p-value p, independence is more probable than dependence, while $E_0(p) < 1$ implies the opposite. Moreover, the value of $E_0(p)$ *quantifies* this belief. Conversely, the corresponding posterior ratio of H_1 against H_0 is

$$E_1(p) = \frac{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}{\hat{\pi_0}}.$$

We define the **maximum posterior ratio (MPR)** for a p-value p to be the maximum between the two:

$$E(p) = max\Big\{\frac{\hat{\pi_0}}{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}, \frac{\hat{\xi} p^{\hat{\xi}-1}(1 - \hat{\pi_0})}{\hat{\pi_0}}\Big\}. \tag{4}$$

MPR estimates heuristically quantify our confidence in the observed adjacencies and non-adjacencies and are employed to create a list of literals as follows: Let X and Y be a pair of observed variables, and p_{XY} be the maximum p-value reported during FCI for these variables. Then, if $E_0(p_{XY}) > E_1(p_{XY})$, the literal $\neg adjacent(X, Y, i)$ is added to $\mathcal{F}$ with confidence estimate $E(p_{XY})$. Otherwise, the literal $adjacent(X, Y, i)$ is added to $\mathcal{F}$ with a confidence estimate $E(p_{XY})$. The list can then be sorted in order of confidence, and the literals can be satisfied incrementally. Whenever a literal in the list is encountered that cannot be satisfied in conjunction with the ones already selected, it is ignored.

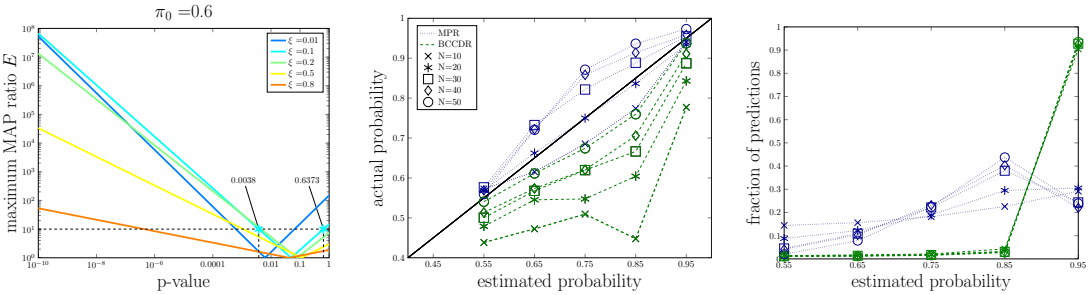


Figure 8: Behaviour and calibration of MPR estimates. (left) Log of the maximum posterior ratio $E(p)$ versus log of the p-value p for $\hat{\pi_0} = 0.6$ and various $\hat{\xi}$. For $\hat{\xi} = 0.1$, an adjacency supported by a maximum p-value of 0.0038 corresponds to the same E as a non-adjacency supported by a p-value of 0.6373. The intersection point of the line with the x axis is the p for which $E_0(p) = E_1(p) = 1$. (center) Probability calibration plots for confidence estimates obtained using MPR estimates $(1/(1 + E_0(p))$ for adjacencies, $E_0(p)/(1 + E_0(p))$ for non-adjacencies). For each interval of length 0.1 in [0.5, 1], the estimated confidences are plotted against the actual frequency of correctness of the corresponding constraints. The green lines correspond to estimates obtained using BCCDR (see Section 5) The confidence estimates correspond to the experiments presented in Figure 10. (right). Number of confidences in each interval.

Notice that, it is possible that for a p-value $E_0(p_{XY}) > E_1(p_{XY})$ (i.e., MPR determines independence is more probable), even though p_{XY} is smaller than the FCI threshold used. In other words, given a fixed FCI threshold, dependence maybe accepted; but, when analyzing the set of p-values encountered to compute MPR, independence seems more probable. The reverse situation is also possible. The pseudo-code in Algorithm 5 (Lines 6—10) accepts the MPR decisions for dependencies and independencies; *this implies that some of the decisions made by FCI will be reversed*. Nevertheless, in anecdotal experiments we found that the literals for which this situation occurs are near the end of the sorted list; thus, whether one accepts the initial decisions of FCI based on a fixed threshold, or a dynamic threshold based on MPR usually does not have a large impact on the output of the algorithm.

Figure 8 shows how the MPR varies with the p-value for $\hat{\pi_0} = 0.6$ and several $\hat{\xi}$'s. The lowest possible value of the MPR is 1, and corresponds to the p-value p for which $E_0(p) = E_1(p)$. Naturally, for the same ξ, this p-value (where the odds switch in favor of non-adjacency) is larger for a lower π_0. In Figure 8 for $\pi_0 = 0.6$ we can see an example of two p-values that correspond to the same E: An adjacency represented by a p-value of 0.0038 (0.0038 being the *maximum* p-value of any test performed by FCI for the pair of variables) is as likely as a non-adjacency represented by a p-value of 0.6373 (0.6373 being the p-value based on which FCI removed this edge).

To obtain MPR estimates, we need to estimate π_0 and ξ. We used the method described in Storey and Tibshirani (2003) to estimate π_0 on the pooled (maximum) p-values $\{p_j\}_{j=1}^{M}$

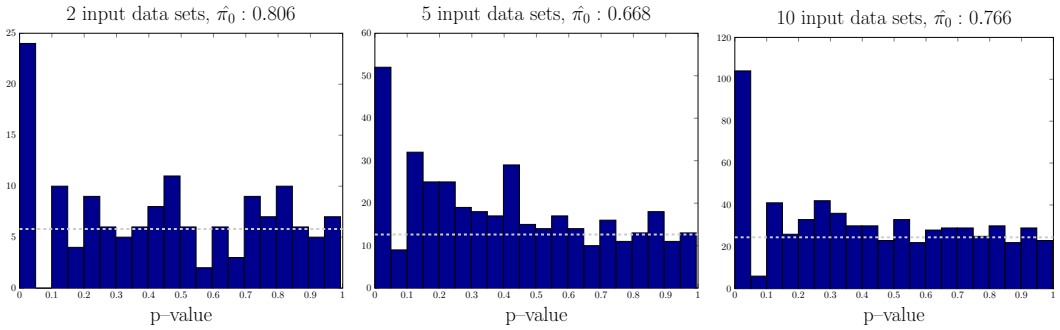


Figure 9: Distribution of p-values and estimated $\hat{\pi_0}$. We used the method of Storey and Tibshirani (2003) to estimate $\hat{\pi_0}$ for a sample of p-values corresponding to 2 (left), 5 (center) and 10 (right) input data sets. We generated networks by manipulating a marginal of the ALARM network (Beinlich et al., 1989) consisting of 14 variables. In each experiment, at most 3 variables were latent and at most 2 variables were manipulated. We simulated data sets of 100 samples each from the resulting manipulated graphs. We ran FCI on each data set with $\alpha = 0.1$ and $maxK = 5$ and cached the maximum p-value reported for each pair of variables. We used the p-values from all data sets to estimate $\hat{\pi_0}$. The dashed line corresponds to the proportion of p-values that come from the null distribution based on the estimated $\hat{\pi_0}$.

over all data sets obtained during FCI. For a given $\hat{\pi_0}$, Equation 3 can then be easily optimized for ξ.

The method used to obtain $\hat{\pi_0}$ assumes independent p-values, which is of course not the case since the test schedule of FCI depends on previous decisions. In addition, each p-value may be the maximum of several p-values; these maximum p-values may not follow a uniform distribution even when the non-adjacency (null hypothesis) is true. Finally, given that p-values stem from tests over different conditioning set sizes, p-values corresponding to adjacencies do not necessarily follow the same beta distribution. Thus, the approach presented here is at best an approximation.

In the algorithm as presented, a single beta is fit from the pooled p-values of FCI runs over all data sets. This strategy is perhaps more appropriate when individual data sets have a small number of p-values, so the pooled set provides a larger sample size for the fitting. Other strategies though, are also possible. One could instead fit a different beta for each data-set and its corresponding set of p-values. This approach could perhaps be more appropriate in case the PAG structures $\mathcal{P}_i$ vary greatly in terms of sparseness. In addition, one could also fit different beta distributions for each conditioning set size. Figure 9 shows the empirical distribution of p-values and the estimated $\hat{\pi_0}$ based on the p-values returned from FCI on 2, 5 and 10 input data sets, simulated from a network of 14 variables.

The strategy for selecting non-conflicting constraints based on the MPR is presented in Algorithm 5. MPR is a general criterion that can be used to compare confidence in dependencies and independencies. The method is based on p-values and thus, can be

Algorithm 5: MPRstrategy

input : SAT formula Φ, list of literals $\mathcal{F}$, their corresponding p-values $\{p_j\}$
output: List of non conflicting literals $\mathcal{F}'$

$\mathcal{F}' \leftarrow \emptyset$;
Estimate $\hat{\pi_0}$ from $\{p_j\}$ using the method described in Storey and Tibshirani (2003);
Find $\hat{\xi}$ that minimizes $-\sum_j log(\hat{\pi_0} + (1-\hat{\pi_0})\xi p_j^{\xi-1})$;
foreach *literal* $(\neg)adjacent(X, Y, \mathcal{P}_i) \in \mathcal{F}$ *with p-value* p_j **do**
    $E_0(p_j) \leftarrow \frac{\hat{\pi_0}}{\hat{\xi} p_j^{\hat{\xi}-1}(1-\hat{\pi_0})}$, $E_1(p_j) \leftarrow \frac{\hat{\xi} p_j^{\hat{\xi}-1}(1-\hat{\pi_0})}{\hat{\pi_0}}$;
    if $E_1(p_j) < E_0(p_j)$ **then**
        add $\neg adjacent(X, Y, \mathcal{P}_i)$ to $\mathcal{F}$
    else
        add $adjacent(X, Y, \mathcal{P}_i)$ in $\mathcal{F}$
    end
    $Score(literal) \leftarrow max\{E_0(p_j), E_1(p_j)\}$;
end
foreach *literal* $collider(X, Y, Z, \mathcal{P}_i)$, $dnc(X, Y, Z, \mathcal{P}_i)$ **do**
    if X, Y, Z *is an unshielded triple in* $\mathcal{P}_i$ **then**
        $Score(literal) \leftarrow Score(X, Z, \mathcal{P}_i)$;
    else if $\langle W \ldots X, Y, Z \rangle$ *is discriminating for* Y *in* $\mathcal{P}_i$ **then**
        $Score(literal) \leftarrow Score(W, Z, \mathcal{P}_i)$;
    end
end
$\mathcal{F} \leftarrow$ sort $\mathcal{F}$ by descending score;
foreach $\phi \in \mathcal{F}$ **do**
    if $\Phi \wedge \phi$ *is satisfiable* **then**
        $\Phi \leftarrow \Phi \wedge \phi$;
        Add ϕ to $\mathcal{F}'$;
    end
end

applied in different types of data (e.g., continuous and discrete) in conjunction with any appropriate test of independence. Moreover, since it is based on cached p-values, and fitting a beta distribution is efficient, it adds minimal computational complexity. On the other hand, the estimation of maximum posterior ratios is based on heuristic assumptions and approximations. Nevertheless, experiments presented in the following section showcase that the method works similarly if not better than other conflict resolution methods, while being orders of magnitude computationally more efficient.

5. Experimental Evaluation

We present a series of experiments to characterize how the behavior of COmbINE is affected by the characteristics of the problem instance and compare it against another alternative

Problem attribute	Default value used
Number of variables in the generating DAG	20
Maximum number of parents per variable	5
Number of input data sets	5
Maximum number of latent variables per data set	3
Maximum number of manipulated variables per data set	2
Sample size per data set	1000

Table 1: Default values used in generating experiments in each iteration of COmbINE. Unless otherwise stated, the input data sets of COmbINE were generated according to these values.

algorithm in the literature. We also present a comparative evaluation of conflict resolution methods, including the one based on the proposed MPR estimation technique. Finally, we present a proof-of-concept application on real mass cytometry data on human T-cells. In more detail, we initially compare the complete version of COmbINE (i.e., without restrictions on the maximum path length or the conditioning set) against SBCSD (Hyttinen et al., 2013) in ideal conditions (i.e., both algorithms are provided with an independence oracle). We perform a series of experiments to explore the (a) learning accuracy of COmbINE as a function of the maximum path length considered by the algorithm, the density and size of the network to reconstruct, the number of input data sets, the sample size, and the number of latent variables, and (b) the computational time as a function of the above factors.

All experiments were performed on data simulated from randomly generated networks as follows. The graph of each network is a DAG with a specified number of variables and maximum number of parents per variable. Variables are randomly sorted topologically and for each variable the number of parents is uniformly selected between 0 and the maximum allowed. The parents of each variable are selected with uniform probability from the set of preceding nodes. Each DAG is then coupled with random parameters to generate conditional linear Gaussian networks. To avoid very weak interactions, minimum absolute conditional correlation was set to 0.2. Before generating a data set, the variables of the graph are partitioned to unmanipulated, manipulated, and latent. Mean value and standard deviation for the manipulated variables were set to 0 and 1, respectively. Subsequently, data instances are sampled from the network distribution, considering the manipulations and removing the latent variables. All experiments are performed on **conservative** families of targets; the term was introduced in Hauser and Bühlmann (2012) to denote families of intervention targets in which all variables have been observed unmanipulated at least once.

For each invocation of the algorithm, the problem instance (set of data sets) is generated using the parameters shown in Table 1. COmbINE default parameters were set as follows: maximum path length = 3, $\alpha = 0.1$ and maximum conditioning set $maxK = 5$, and the Fisher z-test of conditional independence. As far as orientations are concerned, in our experience, FCI is very prone to error propagation, we therefore used the rule in (Ramsey et al., 2006) for *conservative* colliders. Unless otherwise stated, Algorithm 5 is employed to resolve conflicts. SAT instances were solved using MINISAT2.0 (Eén and Sörensson, 2004) along with the modifications presented in Hyttinen et al. (2013) for iterative solving and

computing the backbone with some minor modifications for sequentially performing literal queries. In the subsequent experiments, *one of the problem parameters in Table 1 is varied each time, while the others retain the values above.*

To measure learning performance, ideally one should know the correct output, i.e., the structure that the algorithm would learn if ran with an oracle of conditional independence, and unrestricted infinite maxK and maximum path length parameters. Notice that *the original generating DAG structure cannot serve as the correct output for comparison.* This is because the presence of manipulated and latent variables implies that not all structural features of the generating DAG can be recovered. For example, for the problem instance presented in Figure 7 (middle), the correct output, shown in Figure 7 (right), has one solid edge out of 5, no solid endpoint, one absent, and four dashed edges. Dashed edges and endpoints in the output of the algorithm can only be evaluated if one knows this correct output. Unfortunately, the correct output cannot be recovered in a timely fashion in most problems involving more than 15 variables, as shown in Section 5.1.

As a surrogate, we defined metrics that do not consider dashed edges or endpoints and can be directly computed by comparing the "solid" features of the output with the original data generating graph. Specifically, we used two types of precision and recall; one for edges (s-Precision/s-Recall) and one for orientations (o-Precision/o-Recall). Let $\mathcal{G}$ be the graph that generated the data (the SMCM stemming from the initial random DAG after marginalizing out variables latent in all data sets), and $\mathcal{H}$ be the summary graph returned by COmbINE. s-Precision and s-Recall were then calculated as follows:

$$\text{s-Precision} = \frac{\#\text{ solid edges in } \mathcal{H} \text{ that are also in } \mathcal{G}}{\#\text{ solid edges in } \mathcal{H}}$$

and

$$\text{s-Recall} = \frac{\#\text{ solid edges in } \mathcal{H} \text{ that are also in } \mathcal{G}}{\#\text{ edges in } \mathcal{G}}.$$

Similarly, orientation precision and recall are calculated as follows:

$$\text{o-Precision} = \frac{\#\text{ endpoints in } \mathcal{G} \text{ correctly oriented in } \mathcal{H}}{\#\text{ of orientations(arrows/tails) in } \mathcal{H}}$$

and

$$\text{o-Recall} = \frac{\#\text{ endpoints in } \mathcal{G} \text{ correctly oriented in } \mathcal{H}}{\#\text{ endpoints in } \mathcal{G}}.$$

Since dashed edges and endpoints do not contribute to these metrics, precision in particular could be favorable for conservative algorithms that tend to categorize all edges (endpoints) as dashed. To alleviate this problem, we accompany each precision / recall figure with the percentage of dashed edges out of all edges in the output graph to indicate how conservative is the algorithm. Similarly, we present the percentage of dashed (circled) endpoints out of all endpoints in the output graph. Finally, we note that in the experiments that follow, unless otherwise stated, we report the median, 5, and 95 percentile over 100 runs of the algorithm with the same settings.

# variables	# max parents	Running time **Median** (5 %ile, 95 %ile)			Completed instances/ total instances		
		COmbINE	SBCSD	SBCSD′	COmbINE	SBCSD	SBCSD′
10	3	**17**$(1, 113)$	**149**$(14, 470)^*$	**91**$(30, 369)^*$	50/50	30/50	48/50
	5	**80**$(4, 1192)$	**365**$(133, 500)^*$	**264**$(68, 554)^*$	50/50	16/50	32/50
14	3	**28**$(4, 6361)^*$	–	**451**$(407, 492)^*$	49/50	0/50	4/50
	5	**272**$(23, 16107)^*$	–	–	43/50	0/50	0/50

Table 2: Comparison of running times for COmbINE and SBCSD for networks of 10 and 14 variables. The table reports the median running time along with the 5 and 95 percentiles, as well as the number of instances (problem inputs) in which each algorithm managed to complete; *numbers are computed only on the problems for which the algorithm completed.

5.1 COmbINE vs. SBCSD

Hyttinen et al. (2013) presented a similar algorithm, named SAT-based causal structure discovery (SBCSD). SBCSD is also capable of learning causal structure from manipulated data sets over overlapping variable sets. In addition, if linearity is assumed, it can admit feedback cycles. SBCSD also uses similar techniques for converting conditional (in)dependencies into a SAT instance. However, the algorithm requires all m-connections to constrain the search space (at least the ones that guarantee completeness), while COmbINE uses inducing paths to avoid that. For each adjacency $X \star\!\!-\!\!\star Y$ in a data set, COmbINE creates a constraint specifying that at least one path between the variables is inducing with respect to $\mathbf{L_i}$. In contrast, SBCSD creates a constraint specifying that at least one path between the variables is m-connecting path given each possible conditioning set. So, both algorithms are forced to check every possible path, yet COmbINE examines each path once (with respect to $\mathbf{L_i}$), while SBCSD examines it for multiple possible conditioning sets. The latter choice may be necessary to deal with cyclic structures, but leads to significantly larger SAT problems when acyclicity is assumed.

SBCSD is not presented with a conflict resolution strategy and so it can only be tested by using an oracle of conditional independence. Equipping SBCSD with such a strategy is possible, but it may not be straightforward: SBCSD computes the SAT backbone incrementally for efficiency, which complicates pre-ranking constraints according to some criterion. Since SBCSD cannot handle conflicts, we compared it to the complete version of our algorithm (infinite maxK and maximum path length) using an oracle of conditional independence. Since no statistical errors are assumed, the initial search graph for COmbINE includes all observed arrows. Both algorithms are sound and complete, hence we only compare running time. SBCSD uses a path-analysis heuristic to limit the number of tests to perform. However, the authors suggest that in cases of acyclic structures, this heuristic could be substituted with the FCI test schedule. To better characterize the behavior of SBCSD on acyclic structures, we equipped the original implementation as suggested.[3] We denote this version of the algorithm as SBCSD′. Also note, that the available implementation of

3. However, we do not include the Possible d-Separating step of FCI; this step hardly influences the quality of the algorithm (Colombo et al., 2012). Thus, the timing results of Table 2 are a lower bound on the execution time of the SBCSD algorithm.

SBCSD by its authors has an option to restrict the search to acyclic structures, which was employed in the comparative evaluation. Finally, we note that SBCSD is implemented in C, while COmbINE is implemented in Matlab.

For the comparative evaluation, we simulated random acyclic networks with 10 and 14 variables. The default parameters were used to generate 50 problem instances for networks with 3 and 5 maximum parents per variable. Both algorithms were run on the same computer, with 4GB of available memory. SBCSD reached maximum memory and aborted without concluding in several cases for networks of 10 variables, and *in all cases for networks of 14 variables*. SBCSD$'$ slightly improves the running time over SBCSD. Median running time along with the 5 and 95 percentiles as well as number of cases completed are reported in Table 2. The metrics for each algorithm were calculated only on the cases where the algorithm completed.

The results in Table 2 indicate that COmbINE is more time-efficient than SBCSD and SBCSD$'$. While the running times do depend on implementation, the fact that SBCSD have much higher memory requirements indicates that the results must be at least in part due to the more compact representation of constraints by COmbINE . COmbINE managed to complete all cases for networks of 10 and most cases for 14 variables, while SBCSD completed less than 50% and 0%, respectively. SBCSD$'$ completed most cases for 10 variables but only 4% of cases for 14 variables. Interestingly, the percentiles for COmbINE are quite wide spanning two orders of magnitude for problems with maxParents equal to 5 (we cannot compute the actual 95 percentile for SBCSD since it did not complete for most problems). Thus, performance highly depends on the input structure. Such heavy-tailed distributions are well-noted in the constraint satisfaction literature (Gomes et al., 2000). We also note the fact that COmbINE seems to depend more on the sparsity and less on the number of variables, while SBCSD's time increases monotonically with the number of variables. Based on these results, we would suggest the use of COmbINE for problems where acyclicity is a reasonable assumption and the number of variables is relatively high.

5.2 Evaluation of Conflict Resolution Strategies

In this section we evaluate our Maximum Map Ratio strategy (**MPR**) against three other alternatives: A ranking strategy where constraints are sorted based on Bayesian probabilities as proposed in Claassen and Heskes (2012) (**BCCDR**), as well as a Max-SAT (**MaxSAT**) and a weighted max-SAT (**wMaxSAT**) approach.

MPR: This strategy sorts constraints according to the Maximum Map Ratio (Algorithm 5) and greedily satisfies constraints in order of confidence; whenever a new constraint is not satisfiable given the ones already selected, it is ignored (Lines 21- 25 in Algorithm 5).

BCCDR: BCCDR sorts constraints according to Bayesian probability estimates of the literals in $\mathcal{F}$ as presented in Claassen and Heskes (2012). The same greedy strategy for satisfying constraints in order is employed. Briefly, the authors propose a method for calculating Bayesian probabilities for any feature of a causal graph (e.g. adjacency, m-connection, causal ancestry). To estimate the probability of a feature, for a given data set $\mathbf{D}$, the authors calculate the score of all DAGs of N variables. Let $\mathcal{G} \vdash f$ denote that a feature f is present in DAG $\mathcal{G}$. The probability of the feature is then calculated as $P(f) = \sum_{\mathcal{G} \vdash \mathrm{f}} P(\mathbf{D}|\mathcal{G})P(\mathcal{G})$. Scoring all DAGs is practically infeasible for networks with

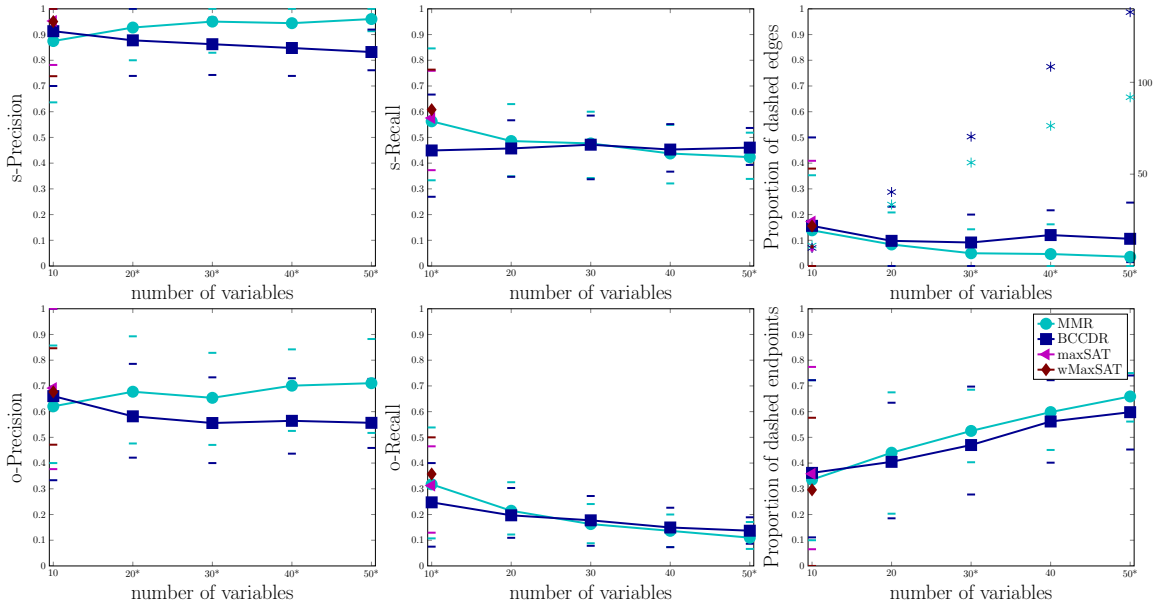


Figure 10: Learning performance of COmbINE with various conflict resolution strategies. From left to right: Median s-Precision, s-Recall, proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints (bottom) for networks of several sizes for various conflict resolution strategies. Each data set consists of 100 samples. The numbers for wMaxSAT and maxSAT correspond to 22 and 23 cases, respectively, in which the algorithms managed to return a solution within 500 seconds. Coloured bars indicate 5 and 95 percentiles. Asterisks in the top right figure show the absolute number of literals rejected by each strategy (y axis on the right). Asterisks on x tick labels indicate cases where the behaviour of MPR and BCCDR are significantly different (paired t-test of equality of means with unknown but equal variances).

more than 5 or 6 variables. Thus, for data sets with more variables, a subset of variables must be selected for the calculation of the probability of a feature. Following (Claassen and Heskes, 2012), we use 5 as the maximum N attempted.

The literals in $\mathcal{F}$ represent information on adjacencies: $(\neg)adjacent(X, Y, \mathcal{P}_i)$ and colliders: $(\neg)collider(X, Y, Z, \mathcal{P}_i)$. To apply the method above for a given feature, we have to select the variables used in the DAGs, a suitable scoring function, and suitable DAG priors. For (non) adjacencies $X \star\!\!-\!\!\star Y$ in PAG $\mathcal{P}_i$, we scored the DAGs over variables X, Y and $\mathbf{Z}$, for the conditioning set $\mathbf{Z}$ maximizing the p-value of the tests $X \perp\!\!\!\perp Y \mid \mathbf{Z}$ performed by FCI. Since the total number of variables cannot exceed 5, the maximum conditioning set for FCI is limited to 3 in all experiments in this section for a fair comparison. (Non) colliders are assigned the same score as the non adjacency of their endpoints.

We use the BGE metric for Gaussian distributions (Geiger and Heckerman, 1994) as implemented in the BDAGL package Eaton and Murphy (2007a) to calculate the likelihoods of the DAGs. This metric is score equivalent, so we pre-computed representatives of the

Markov equivalent networks of up to 5 nodes, and scored only one network per equivalence class to speed up the method. Priors for the DAGs were also pre-computed to be consistent with respect to the maximum attempted number of nodes (i.e. 5) as suggested in Claassen and Heskes (2012).

MaxSAT: This approach tries to satisfy as many literals in $\mathcal{F}$ as possible. Recall that the SAT problem consists of a set of hard-constraints (conditionals, no cycles, no tail-tail edges), which should always be satisfied (hard constraints), and a set of literals $\mathcal{F}$. Maximum SAT solvers cannot be directly applied to the entire SAT formula since they do not distinguish between hard and soft constraints. To maximize the number of literals satisfied, while ensuring all hard-constraints are satisfied we resorted to the following technique: we use the akmaxsat (Kuegel, 2010) *weighted* max SAT solver that tries to maximize the sum of the weights of the satisfied clauses. Each literal is assigned a weight of 1, and each hard-constraint is assigned a weight equal to the sum of all weights in $\mathcal{F}$ plus 10000. The summary graph returned by Algorithm 2 is based on the backbone of the subset of literals selected by akmaxsat.

wMaxSAT: Finally, we augmented the above technique with a different weighted strategy that considers the importance of each literal. Specifically, each literal was weighted proportionally to the logarithm of the corresponding MPR. Again, each hard-constraint was assigned a weight equal to the sum of all weights in $\mathcal{F}$ plus 10000, to ensure that the solver will always satisfy these statements. The summary graph returned by Algorithm 2 is based on the backbone of the subset of literals selected by akmaxsat.

We ran all methods for networks of 10, 20, 30, 40 and 50 variables for data sets of 100 samples to test them on cases where statistical errors are common. For each network size we performed 50 iterations. **MaxSAT** and **wMaxSAT** often failed to complete in a timely fashion; to complete the experiments we aborted the solver after 500 seconds. We note that this amount of time corresponds to more than 10 times the maximum running time of the MPR method (calculating MPRs and solving the SAT instance), and more than twice times the maximum running time of the BCCDR-based method (for 50 variables). Cases where the solver did not complete were not included in the reported statistics. Unfortunately, *the methods using weighted max SAT solving failed to complete in most cases for 10 variables*, and all cases for more than 10 variables.

The results are shown in Figure 10, where we can see the median performance of both algorithms over 50 iterations. Overall, **MPR** exhibits better Precision and identifies more solid edges, while **BCCDR** exhibits slightly better Recall. **BCCDR** is better for variable size equal to 10, which could be explained from the fact that **MPR** is not provided with sufficient number of p-values to estimate $\hat{\pi}_0$ and $\hat{\xi}$. In terms of computational complexity, for networks of 50 variables, estimating the **BCCDR** ratios takes about 150 seconds on average, while estimating the **MPR** ratios takes less than a second. The more sophisticated search strategies **MaxSAT** and **wMaxSAT** do not seem to offer any significant quality benefits, at least for the single variable size for which we could evaluate them. Based on these results, we believe that **MPR** is a reasonable and relatively efficient conflict resolution strategy.

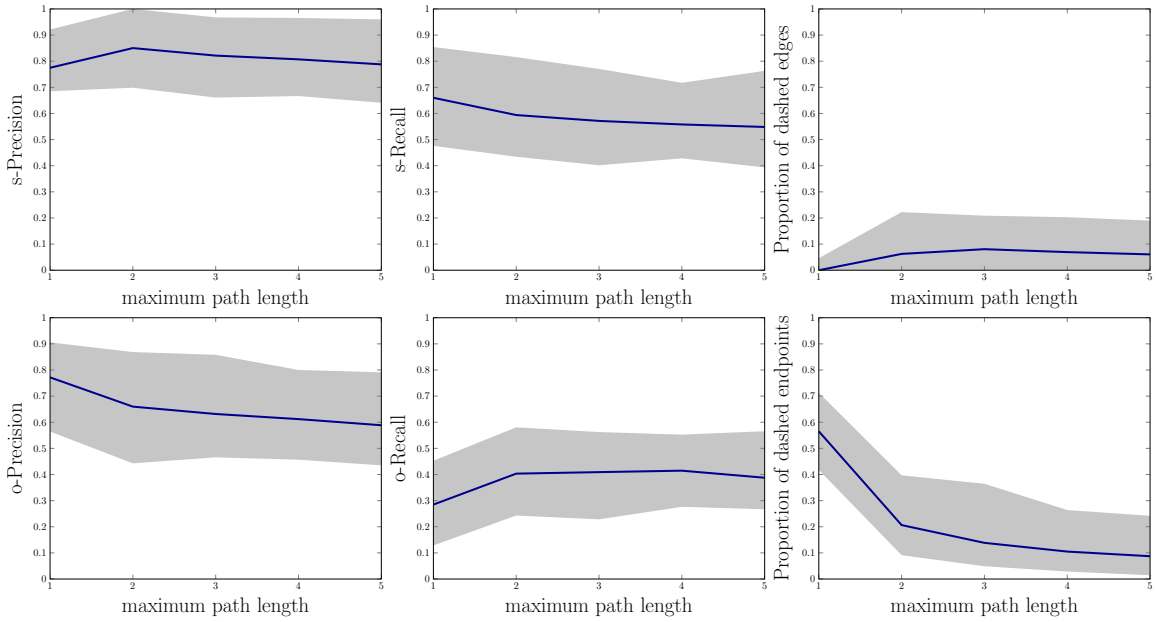


Figure 11: Learning performance of COmbINE against maximum path length. From left to right: s-Precision, s-Recall, percentage dashed edges and o-Precision, o-Recall and percentage of dashed endpoints (bottom) for varying maximum path length, averaged over all networks. Shaded area ranges from the 5 to the 95 percentile. Maximum path length 3 seems to be a be a reasonable trade-off between performance, percentage of dashed features, and efficiency.

5.3 COmbINE Performance with Increasing Maximum Path Length

In this section, we examine the behavior of the algorithm when the length of the paths considered is limited, in which case the output is an approximation of the actual solution. The COmbINE pseudo-code in Algorithm 2 accepts the maximum path length as a parameter.

Learning performance as a function of the maximum path length is shown in Figure 11. Notice that when the path length is increased from 1 to 2 there is drop in the percentage of dashed endpoints, implying more orientations are possible. For length equal to 1, only unshielded and discriminating colliders are identified, while for length larger than 2 further orientations become possible thanks to reasoning with the inducing paths. When length is 1, notice that there are almost no dashed edges (except for the edges added in Line 5 of Algorithm 3). When the maximum length increases, adjacencies in one data set, can be explained with longer inducing paths in the underlying graph and more dashed edges appear. The learning performance of the algorithm is not monotonic with the maximum length. Explaining an association (adjacency) through the presence of a long inducing path may be necessary for asymptotic correctness. However, in the presence of statistical errors, allowing such long paths could lead to complicated solutions or the propagation of errors.

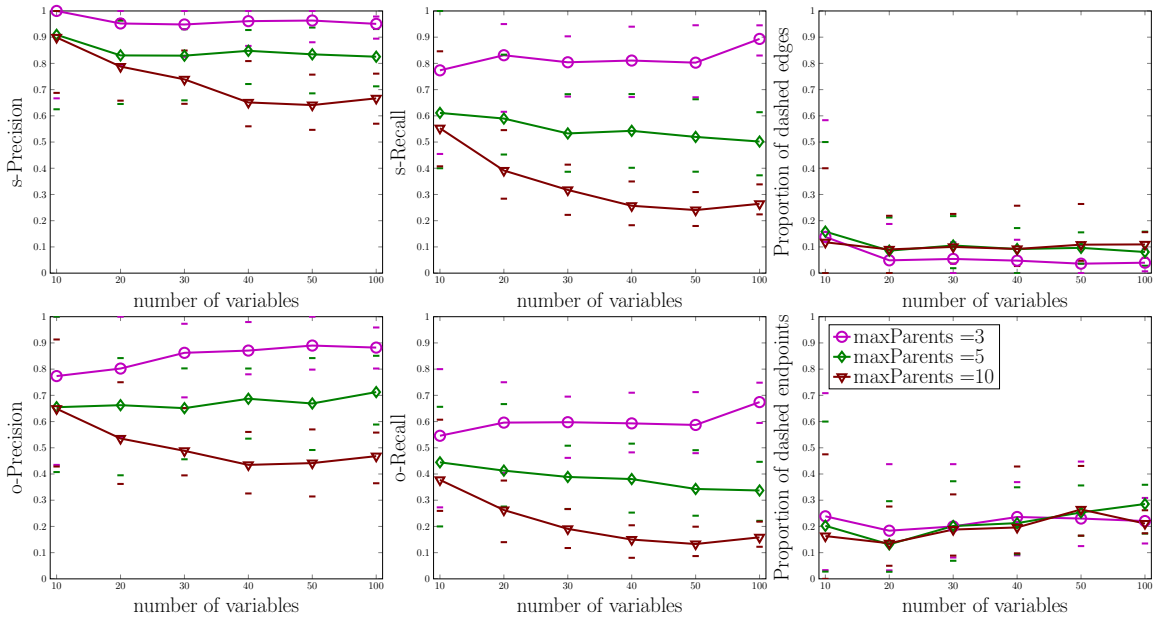


Figure 12: Learning performance of COmbINE for various network sizes and densities. From left to right: Median s-Precision, s-Recall, proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints (bottom) for varying network size and density. Density is controlled by limiting the number of possible parents per variable. Coloured bars indicate 5 and 95 percentiles. As expected, the performance deteriorates as networks become denser.

Overall, it seems any increase of the maximum path length above 3 does not significantly affect performance. It seems that a maximum path length of 3 is a reasonable trade-off among learning performance (precision and recall), percentage of uncertainties, and computational efficiency. These experiments justify our choice of maximum length 3 as the default parameter value of the algorithm.

5.4 COmbINE Performance as a Function of Network Density and Size

In Figure 12 the learning performance of the algorithm is presented as a function of network density and size. Density was controlled by the maximum parents allowed per variable, set by parameter maxParents during the generation of the random networks. For all network sizes, learning performance monotonically decreases with increased density, while the percentage of dashed features does not significantly vary. The size of the network has a smaller impact on the performance, particularly for the sparser networks. For dense networks, performance is relatively poor and becomes worse with larger sizes.

We also calculated confusion matrices for edges and endpoints inferred by COmbINE against the *correct output* structure $\mathcal{H}$ for networks of 10 variables, where $\mathcal{H}$ can be obtained by running COmbINE with an oracle of conditional independence and unrestricted path length and conditioning set size. Table 3 shows the resulting confusion matrices for

		Actual $\mathcal{H}$					
		maxParents 3			maxParents 5		
	Edges	solid	dashed	absent	solid	dashed	absent
$\hat{\mathcal{H}}$	solid	**8.0** (4.0, 12.0)	**0.0** (0.0, 5.0)	**0.0** (0.0, 4.0)	**9.0** (3.0, 13.0)	**1.0** (0.0, 10.0)	**1.0** (0.0, 5.0)
	dashed	**0.0** (0.0, 3.0)	**0.0** (0.0, 4.0)	**0.0** (0.0, 2.0)	**0.5** (0.0, 4.0)	**0.5** (0.0, 3.0)	**1.0** (0.0, 2.0)
	absent	**1.0** (0.0, 4.0)	**0.0** (0.0, 3.0)	**31.0** (24.0, 36.0)	**2.5** (0.0, 8.0)	**1.5** (0.0, 9.0)	**24.0** (14.0, 34.0)
	Endpoints	arrow	circle	tail	arrow	circle	tail
	arrow	**8.0** (4.0, 12.0)	**1.0** (0.0, 5.0)	**0.0** (0.0, 3.0)	**8.0** (4.0, 13.0)	**3.0** (0.0, 8.0)	**2** (0.0, 5.0)
	circle	**1.0** (0.0, 3.0)	**3.0** (0.0, 14.0)	**0.0** (0.0, 2.0)	**1.0** (0.0, 5.0)	**3.0** (0.0, 8.0)	**1.0** (0.0, 4.0)
	tail	**0.0** (0.0, 2.0)	**0.0** (0.0, 5.0)	**4.0** (0.0, 8.0)	**1.0** (0.0, 5.0)	**1** (0.0, 54.0)	**3.0** (1.0, 6.0)

Table 3: Confusion matrices reporting edge and endpoint counts of the output of COmbINE $\hat{\mathcal{H}}$ versus the actual summary graph $\mathcal{H}$. Results are shown for 10 variables and 5 data sets of 1000 samples each. $\mathcal{H}$ was obtained using COmbINE with an oracle of conditional independence, and unconstrained maxK and maximum path length parameters. The table reports median values (bold) along with the 5 and 95 percentiles (in parenthesis). Results are in agreement with the metrics used for larger networks.

maxParents 3 and 5 and 5 data sets of sample size 1000. Overall, the results are in concordance with the metrics used for larger networks, and confirm that the method works best for sparser networks. Notice that for dense networks (for N=10 and maxParents =5, the networks have about 40% of all possible edges), there are cases where the actual correct output includes a large proportion of dashed edges, while constricting the maximum path length forces the algorithm to accept more solid features (hence the wide percentiles).

5.5 COmbINE Performance over Sample Size and Number of Input Data Sets

Figure 13 shows the performance of the algorithm with increasing the number of input data sets. As expected, the percentage of uncertainties (dashed features) is steadily decreasing with increased number of input data sets. Recall also steadily improves, while Precision is relatively unaffected. Figure 14 holds the number of input data set constant to the default value 5, while increasing the sample size per data set. Recall in particular improves with larger sample sizes, while the percentage of dashed endpoints drops.

5.6 COmbINE Performance for Increasing Number of Latent Variables

We also examine the effect of confounding to the performance of COmbINE . To do so, we generated semi-Markov causal models instead of DAGs in the generation of the experiments: We generated random DAG networks of 30 variables and then marginalized out a percentage of the variables. Figure 15 depicts COmbINE's performance against 3, 6, and 9 of latent variables, corresponding to 10%, 20% and 30% of the total number of variables in the graph, respectively. Overall, confounding does not seem to greatly affect the performance of COmbINE. We must point out however, that s-Recall is lower than the s-Recall with no confounded variables for the same network size (see Figure 12).

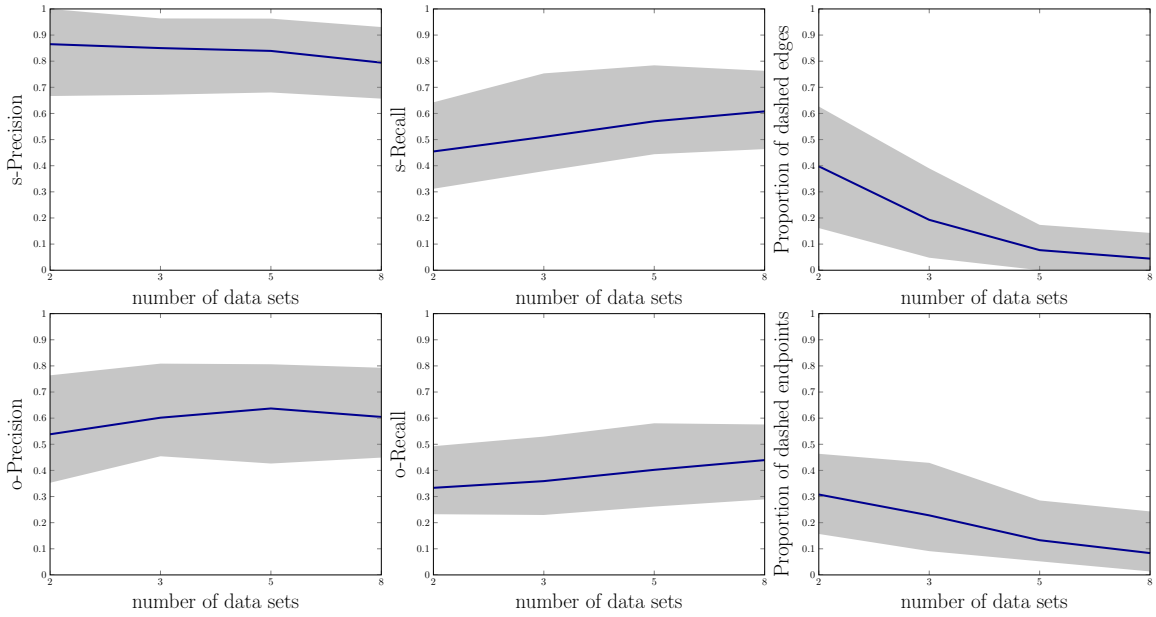


Figure 13: Learning performance of COmbINE for varying number of input data sets. From left to right: Median s-Precision, s-Recall, Proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints of (bottom) for varying number of input data sets. Shaded area ranges from the 5 to the 95 percentile. Increasing the number of input data sets improves the performance of the algorithm.

5.7 Running Time for COmbINE

The running time of COmbINE depends on several factors, including the ones examined in the previous experiments: Maximum path length, number of input data sets and sample size, and, naturally, the number of variables. Figure 16 illustrates the running time of COmbINE against these factors. As we can see in Figure 16, the restriction on the maximum path length is the most critical factor for the scalability of the algorithm.

5.8 A Case Study: Mass Cytometry Data

Mass cytometry (Bendall et al., 2011) is a recently introduced technique that enables measuring protein activity in cells, and its main use is to classify hematopoietic cells and identify signaling profiles in the immune system. Therefore, the proteins are usually measured in a sample of cells and then in a different sample of the same (type of) cells after they have been stimulated with a compound that triggers some kind of signaling behavior. Identifying the causal succession of events during cell signaling is crucial to designing drugs that can trigger or suppress immune reaction. Therefore in several studies both stimulated and un-stimulated cells are treated with several perturbing compounds to monitor the potential effect on the signaling pathway.

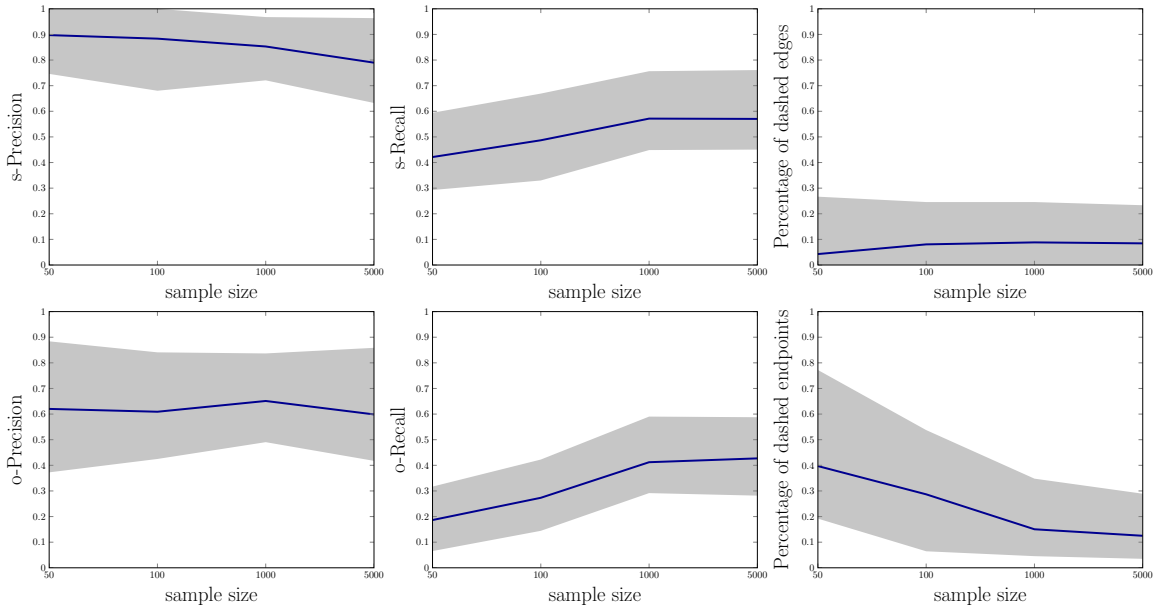


Figure 14: Learning performance of COmbINE for varying sample size per data set. From left to right: s-Precision, s-Recall, Proportion of dashed edges (top) and o-Precision, o-Recall and proportion of dashed endpoints of (bottom) for varying sample size per data set. Shaded area ranges from the 5 to the 95 percentile. Increasing the sample size improves the performance of the algorithm.

Mass cytometry data seem to be a suitable test-bed for causal discovery methods: The proteins are measured in single cells instead of representing tissue averages, the latter being known to be problematic for causal discovery (Chu et al., 2003), and the samples range in thousands. However, the mass cytometer can measure only up to 34 variables, which may be too low a number to measure all the variables involved in a signaling pathway. Moreover, about half of these variables are surface proteins that are necessary to distinguish (gate) the cells into sub-populations, but are not functional proteins involved in the signaling pathway. It is therefore reasonable for scientists to perform experiments measuring overlapping variable sets.

Bendall et al. (2011) and Bodenmiller et al. (2012) both use mass cytometry to measure protein abundance in cells of the immune system. In both studies, the samples were treated with several different signaling stimuli. Some of the stimuli were common in both studies. After stimulation with each activating compound, Bodenmiller et al. (2012) also test the cell's response to 27 inhibitors. One of these inhibitors is also used in Bendall et al. (2011). For this inhibitor, Bendall et al. (2011) measured bone marrow cell samples of a single donor. In Bodenmiller et al. (2012), measurements were taken from peripheral blood mononuclear cell (PBMC) samples of a (different) single donor. Despite differences in the experimental setup, the signaling pathway of every stimulus and every sub-population of cells is considered universal across (healthy) donors, so the data should reflect the same underlying causal structure.

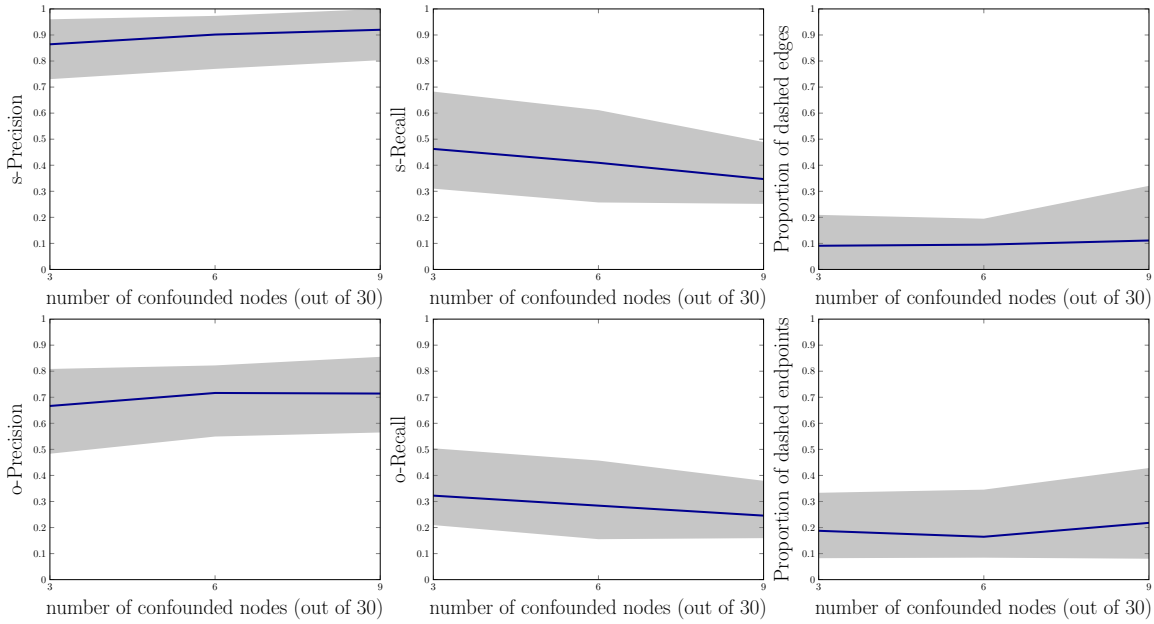


Figure 15: Learning performance of COmbINE for varying percentage of confounded variables. From left to right: s-Precision, s-Recall, percentage of dashed edges (top) and o-Precision, o-Recall and percentage of dashed endpoints (bottom) for varying number of confounded nodes for networks of 30 variables. Shaded area ranges from the 5 to the 95 percentile. Overall, the number of confounding variables does not seem to greatly affect the algorithm' s performance.

We focused on two sub-populations of the cells, CD4+ and CD8+ T-cells, which are known to play a central role in immune signaling. The data were manually gated by the researchers in the original studies. We also focused on one of the stimuli present in both studies, PMA-Ionomycin, which is known to have prominent effects on T-cells. Proteins pBtk, pStat3, pStat5, pNfkb, pS6, pp38, pErk, pZap70, pSHP2 and pPlcg2 are measured in both data sets (initial p denotes that the concentration of the phosphorylated protein is measured). Four additional variables were included in the analysis, pAkt, pLat and pStat1 measured only in Bodenmiller et al. (2012) and pMAPK measured only in Bendall et al. (2011). To be able to detect signaling behavior, we formed data sets that contain both stimulated and unstimulated samples.

As mentioned above, the cells were treated with several inhibitors. Some of these inhibitors target a specific protein, and some of them perturb the system in a more general or unidentified way. Specific inhibitors can be abundance inhibitors, which affect the level of measured protein, and activity inhibitors, which affect the function of measured proteins. The former are closer to ideal hard interventions. Activity inhibitors have been modelled in several ways in the literature. Sachs et al. (2005) model them as ideal interventions by manually setting the values to the lowest discretization level. Itani et al. (2010) propose splitting the target variable in two nodes, one used to represent the inhibition and the other

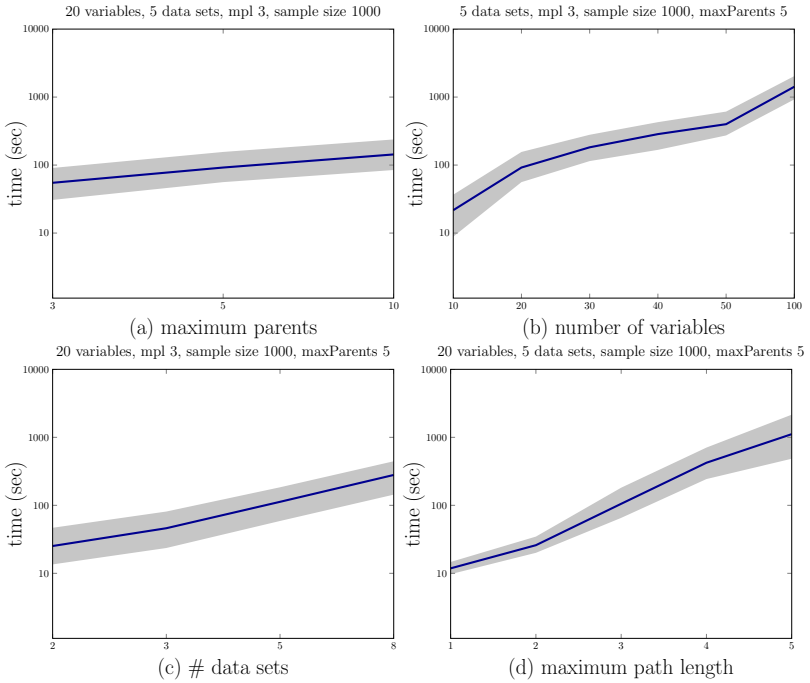


Figure 16: Running time of COmbINE. From left to right: Running time (in seconds) is plotted in logarithmic scale against maximum parents per variable and number of variables (top row); number of data sets and maximum path length (bottom row). Shaded area ranges from the 5 to the 95 percentile. The number of variables and the maximum path length seem to be the most critical factors of computational performance. Notice that, COmbINE scales up to problems with 100 total variables for limited path length and relatively sparse networks.

used to represent the abundance. Mooij and Heskes (2013) propose modelling activity inhibitions by removing outgoing edges of the target variable. Notice that this type of modelling can be easily encoded in a SAT representation.

We used abundance inhibitors that we believe can be modeled as hard interventions (i.e. the compounds used to target these proteins are known to be specific and to have an effect in the phosphorylation levels of the target). The maximum dosage of each inhibitor was used. For all three interventions, the distribution of the target variable under zero dosage is differs significantly (according to a Kolmogorov-Smirnov test with significance threshold 0.05) from the distribution of the target variable for the maximum dosage, indicating that the inhibitor has an effect on the abundance of the target protein. Nevertheless, we must point out that the interventions may not be entirely ideal. More information on the specific compounds can be found in the respective publications.

We ended up with four data sets for each sub-population. Details can be found in Table 4. Protein interactions are typically non-linear, so we discretized the data into 4 bins. We ran Algorithm 2 with maximum path length 3. We used the G^2 test of independence for

Data set	Source	latent ($\mathbf{L_i}$):	manipulated($\mathbf{I_i}$)	Donor
$\mathbf{D_1}$	Bodenmiller et al. (2012)	pMAPK	pAkt	1
$\mathbf{D_2}$	Bodenmiller et al. (2012)	pMAPK	pBtk	1
$\mathbf{D_3}$	Bodenmiller et al. (2012)	pMAPK	pErk	1
$\mathbf{D_4}$	Bendall et al. (2011)	pAkt, pLat, pStat1	pErk	2

Table 4: Summary of the mass cytometry data sets co-analyzed with COmbINE. The procedure was repeated for two sub-populations of cells, CD4+ cells and CD8+ cells.

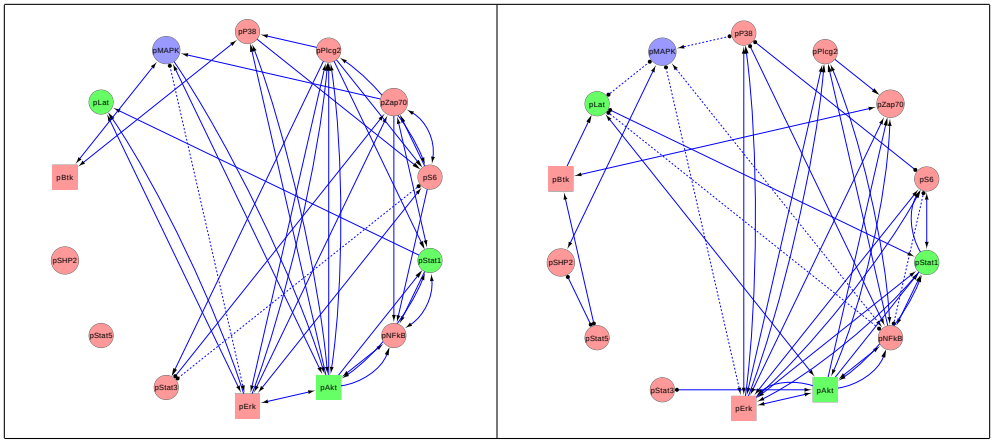


Figure 17: A case study for COmbINE: Mass cytometry data. COmbINE was run on 4 different mass cytometry data for two different cell populations: CD4+ T-cells (left) and CD8+ T-cells (right). In each data set, one variable was manipulated (pAkt, pBTk, pErk, pErk respectively). Variables pAkt, pLat and pStat1 are only measured in data sets 1-3, while pMAPK is only measured in data set 4.

FCI with $\alpha = 0.05$ and maxK=5. We used Cytoscape (Smoot et al., 2011) to visualize the summary graphs produced by COmbINE, illustrated in Figure 17.

Unfortunately, the ground truth for this problem is not known for a full quantitative evaluation of the results. Nevertheless, this set of experiments demonstrates the availability of real and important data sets and problems that are suited integrative causal analysis. Second, these experiments provide a proof-of-concept for the specific algorithm. One type of interesting type of inference possible with COmbINE and similar algorithms is the prediction of a direct relation of pAkt and pMAPK in CD4+ cells, *even though the variables are not jointly measured in any of the input data sets*. Thus, methods for learning causal structure from multiple manipulations over overlapping variables potentially constitute a powerful tool in the field of mass cytometry.

We do not make any claims for the validity of the output graphs and they are presented only as a proof-of-concept, as there are several potential pitfalls. In addition to the potential imperfect manipulations described above, COmbINE also assumes lack of feedback cycles,

which is not guaranteed in this system. We note however, that acyclic networks have been successfully used for reverse engineering protein pathways in the past (Sachs et al., 2005).

6. Conclusions and Future Work

We have presented COmbINE, a sound and complete algorithm that performs causal discovery from multiple data sets that measure overlapping variable sets under different interventions in acyclic domains. COmbINE works by converting the constraints on inducing paths in the sought out semi Markov causal model (SMCMs) that stem from the discovered (in)dependencies into a SAT instance. COmbINE outputs a summary of the structural characteristics of the underlying SMCM, distinguishing between the characteristics that are identifiable from the data (e.g., causal relations that are postulated as present), and the ones that are not (e.g., relations that could be present or not). In the empirical evaluation the algorithm outperforms in efficiency a recently published similar one (Hyttinen et al., 2013) that, given an oracle of conditional independence, performs the same inferences by checking all m-connections necessary for completeness.

COmbINE is equipped with a conflict resolution technique that ranks dependencies and independencies discovered according to confidence as a function of their p-values. This technique allows it to be applicable on real data that may present conflicting constraints due to statistical errors. To the best of our knowledge, COmbINE is the only implemented algorithm of its kind that can be applied on real data.

The algorithm is empirically evaluated in various scenarios, where it is shown to exhibit high precision and recall and reasonable behavior against sample size and number of input data sets. It scales up to networks with up to 100 variables for relatively sparse networks. Moreover, it is possible for the user to trade the number of inferences for improved computational efficiency by limiting the maximum path length considered by the algorithm. As a proof-of-concept application, we used COmbINE to analyze a real set of experimental mass-cytometry data sets measuring overlapping variables under three different interventions.

COmbINE outputs a summary of the characteristics of the underlying SMCM that can be identified by computing the backbone of the corresponding SAT instance. The conversion of a causal discovery problem to a SAT instance makes COmbINE easily extendable to other inference tasks. One could instead produce all SAT solutions and obtain all the SMCMs that are plausible (i.e., fit all data sets). In this case, COmbINE with input a single PAG would output all SMCMs that are Markov Equivalent with the PAG; there is no other known procedure for this task. Alternatively, one could easily query whether there are solution models with certain structural characteristics of interest (e.g., a directed path from A to B); this is easily done by imposing additional SAT clauses expressing the presence of these features. Incorporating certain types of prior knowledge such as causal precedence information can also be achieved by imposing additional path constraints. Future work includes extending this work for admitting soft interventions and known instrumental variables. The conflict resolution technique proposed could be employed to standard causal discovery algorithms that learn from single data sets, in an effort to improve their learning quality.

Acknowledgements

We thank the anonymous reviewers and the action editor for their constructive comments, their thorough reviews really helped improve the manuscript. We also thank Vincenzo Lagani and Giorgos Borboudakis for comments and suggestions on early versions of this work, and Tom Claassen for providing clarifications on the BCCD algorithm. ST and IT were funded by the STATegra EU FP7 project, No 306000. IT was partially funded by the ERC Consolidator Grant No 617393 CAUSALPATH, as well as the EPILOGEAS GSRT ARISTEIA II project, No 3446, which is part of the NSRF 2007-2013 Education and Lifelong Learning Program, co-financed by the European Union (European Social Fund) and national resources.

Appendix A. Proofs

We now present proofs for propositions and theorems presented in the main section.

Proposition 12 *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$*. Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{O}$ *that is faithful to* $\mathcal{J}$ *and* $\mathcal{M}$ *be the MAG over the same variables that is faithful to* $\mathcal{J}$*. Let* $X, Y \in \mathbf{O}$*. Then there is an inducing path between* X *and* Y *with respect to* $\mathbf{L}$*,* $\mathbf{L} \subseteq \mathbf{O}$ *in* $\mathcal{S}$ *if and only if there is an inducing path between* X *and* Y *with respect to* $\mathbf{L}$ *in* $\mathcal{M}$*.*

Proof ($\Rightarrow$) Assume there exists a path p in $\mathcal{S}$ that is inducing w.r.t. $\mathbf{L}$. Then by Theorem 10 there exists no $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{S}$, and since $\mathcal{S}$ and $\mathcal{M}$ entail the same m-separations there exists no $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{M}$. Thus, by Theorem 9 there exists an inducing path between X and Y with respect to $\mathbf{L}$ in $\mathcal{M}$.
($\Leftarrow$) Similarly, assume there exists a path p in $\mathcal{M}$ that is inducing w.r.t. $\mathbf{L}$. Then by Theorem 9 there exists no $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{M}$, and since $\mathcal{S}$ and $\mathcal{M}$ entail the same m-separations there exists no $\mathbf{Z} \subseteq \mathbf{O} \setminus \mathbf{L} \cup \{X, Y\}$ such that X and Y are m-separated given $\mathbf{Z}$ in $\mathcal{S}$. Thus, by Theorem 10 there exists an inducing path between X and Y with respect to $\mathbf{L}$ in $\mathcal{S}$. ■

Theorem 13 *Let* $\mathbf{O}$ *be a set of variables and* $\mathcal{J}$ *the independence model over* $\mathbf{O}$*. Let* $\mathcal{S}$ *be a SMCM over variables* $\mathbf{O}$ *that is faithful to* $\mathcal{J}$*. Let* $\mathcal{M}$ *= SMCMtoMAG(*$\mathcal{S}$*). Then* $\mathcal{S}$ *and* $\mathcal{M}$ *share the same ancestry relations and* $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$*, hence the two graphs entail the same independence model.*

Proof $\mathcal{S}$ and $\mathcal{M}$ share the same ancestry relations, since during Algorithm 1 a directed edge $X \rightarrow Y$ is added only if X is an ancestor of Y in $\mathcal{S}$, and no directed edges are removed. To prove that the $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{M})$, consider a DAG $\mathcal{G}$ constructed from $\mathcal{S}$ as follows: For every bi-directed edge $X \leftrightarrow Y$, introduce a new node L_{XY}. Remove $X \leftrightarrow Y$ and add $X \leftarrow L_{XY} \rightarrow Y$. Let $\{L_{V_i V_j}\}$ be the set of nodes added by this procedure. Obviously, $\mathcal{G}$ is a DAG and $\mathcal{G}$ and $\mathcal{S}$ share the same ancestry relations and the same m-separations for variables in $\mathbf{O}$, thus $\mathcal{J}_m(\mathcal{S}) = \mathcal{J}_m(\mathcal{G})[_{\mathbf{L}}$. If $\langle X, V_1, \ldots, V_n, Y \rangle$ is a primitive inducing path

in $\mathcal{S}$, then $\langle X, L_{XV_1}, V_1, \ldots, L_{V_{n-1}V_n}, V_n, L_{V_nY}, Y\rangle$ is an inducing path with respect to $\mathbf{L}$ in $\mathcal{G}$ and vice versa. Thus, X and Y are adjacent in $\mathcal{G}[_{\mathbf{L}}$ if only if there exists a primitive inducing path between X and Y in $\mathcal{S}$, and $\mathcal{G}$ shares the same ancestry relations with $\mathcal{S}$ for variables in $\mathbf{O}$, thus by Definition 3, $\mathcal{G}[_{\mathbf{L}} = \mathcal{M}$. By Theorem 4 (Richardson and Spirtes, 2002) $\mathcal{J}_m(\mathcal{M}) = \mathcal{J}_m(\mathcal{G}[_{\mathbf{L}}) = \mathcal{J}_m(\mathcal{G})[_{\mathbf{L}} = \mathcal{J}_m(\mathcal{S})$. ■

In all subsequent lemmas, theorems and proofs we employ the assumptions and notation presented in Section 4 (Assumptions A1-A3 and notation presented beneath them). We also assume the algorithms are run with an oracle of conditional independence and infinite maximum conditioning set size and maximum path length.

The following theorem proves that a $\mathcal{S}$ is possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^n$ and $\{\mathbf{I}_i\}_{i=1}^N$ if and only if the result of manipulating $\mathbf{I}_i$, adding necessary edges to create a Markov equivalent MAG and then marginalizing out variables in $\mathbf{L}_i$ produces a MAG $\mathcal{M}_i$ that belongs to the Markov equivalence class represented by $\mathcal{P}_i$ for all experiments.

Theorem 14 *If $\mathcal{S}$ is a SMCM, $\{\mathcal{J}_i\}_{i=1}^N$ is a family of independence models, $\{\mathbf{I}_i\}_{i=1}^N$ is a family of intervention targets and $\mathcal{P}_i$ is the PAG of the Markov equivalence class of MAGs faithful to $\mathcal{J}_i$, the following statements are equivalent:*

- *$\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
- *$\forall i, \mathcal{M}_i \in \mathcal{P}_i$, where $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}$.*

Proof The following hold:

$$\mathcal{S} \text{ is a possibly underlying SMCM for } \{\mathcal{J}_i\}_{i=1}^N \text{ and } \{\mathbf{I}_i\}_{i=1}^N \Leftrightarrow \mathcal{J}_m(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} = \mathcal{J}_i \quad \forall i$$

(by definition)

$$\mathcal{J}_m(\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i}))[_{\mathbf{L}_i} = \mathcal{J}_m(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} = \mathcal{J}_i \quad \forall i \quad \text{(by Theorem 13)}$$

$$\mathcal{J}_m(\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i}) = \mathcal{J}_m(\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i}))[_{\mathbf{L}_i} = \mathcal{J}_i \quad \forall i \quad \text{(by Theorem 4)}$$

$$\mathcal{J}_m(\mathcal{M}_i) = \mathcal{J}_i \quad \forall i, \text{ and by definition of } \mathcal{P}_i, \quad \mathcal{M}_i \in \mathcal{P}_i \quad \forall i.$$

■

The following Lemma proves that no inducing and ancestral paths present in the true underlying SMCM are ruled out during the construction of the initial search graph, and is necessary for subsequent proofs. We prove that $\mathcal{H}_{in}$ has a superset of edges and a subset of orientations compared to $\mathcal{S}$.

Lemma 15 *If $\mathcal{H}_{in}$ is the initial search graph returned by Algorithm 3 for $\{\mathcal{P}_i\}_{i=1}^N$, and $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, then the following hold: If p is an ancestral path in $\mathcal{S}$, then p is a possibly ancestral path in $\mathcal{H}_{in}$. Similarly, if p is an inducing path with respect to $\mathbf{L}$ in $\mathcal{S}$, then p is a possibly inducing path with respect to $\mathbf{L}$ in $\mathcal{H}_{in}$.*

Proof We will first prove that $\mathcal{H}_{in}$ has a superset of edges compared to $\mathcal{S}$, and therefore any path in $\mathcal{S}$ is a path also in $\mathcal{H}_{in}$. If X and Y are adjacent in $\mathcal{S}$, then one of the following holds:

1. $\exists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$. Then the edge is present in $\mathcal{S}^{\mathbf{I}_i}$, and X and Y are adjacent in $\mathcal{P}_i$: the edge is added to $\mathcal{H}_{in}$ in Line 3 of Algorithm 3.

2. $\nexists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$. Then the edge is added to $\mathcal{H}_{in}$ in Line 8 of Algorithm 3.

Therefore, every edge in $\mathcal{S}$ is present also in $\mathcal{H}_{in}$. We must also prove that no orientation in $\mathcal{H}$ is oriented differently in $\mathcal{S}$: $\mathcal{H}_{in}$ has only arrowhead orientations, so we must prove that, if $X \ast\!\!\rightarrow Y$ in $\mathcal{H}_{in}$ and X and Y are adjacent in both graphs, $X \ast\!\!\rightarrow Y$ in $\mathcal{S}$.

Arrowheads are added to $\mathcal{H}_{in}$ in Lines 5, 9 or 10 of the Algorithm. Arrowheads added in Line 5 occur in all $\mathcal{P}_i$. If $X \ast\!\!\rightarrow Y$ in any $\mathcal{P}_i$, this means that Y is not an ancestor of X in $\mathcal{S}^{\mathbf{I}_i}$. Assume that $X \leftarrow\!\!\ast Y$ in $\mathcal{S}$: If X in $\mathbf{I}_i$, the edge would be absent in $\mathcal{S}^{\mathbf{I}_i}$ and $\mathcal{P}_i$. If $X \notin \mathbf{I}_i$, X would be ancestor of Y in $\mathcal{S}^{\mathbf{I}_i}$, which is a contradiction. Therefore, if X and Y are adjacent in $\mathcal{S}$, $X \ast\!\!\rightarrow Y$ in $\mathcal{S}$.

Arrows added to $\mathcal{H}_{in}$ in Lines 9 and 10 correspond to cases where an edge is not present in any $\mathcal{P}_i$, $\nexists i$ s.t. $X, Y \in \mathbf{O}_i \setminus \mathbf{I}_i$, but $\exists i$ s.t. $X, Y \in \mathbf{O}_i$, $X \in \mathbf{I}_i$ and $Y \notin \mathbf{I}_i$. Then an arrow is added towards X. Assume the opposite holds: $X \rightarrow Y$ in $\mathcal{S}$, then $X \rightarrow Y$ in $\mathcal{S}^{\mathbf{I}_i}$, and since both variables are observed in experiment i the edge would be present in $\mathcal{P}_i$, which is a contradiction. Thus, if the edge is present in $\mathcal{S}$, the edge is oriented into X.

Thus, $\mathcal{H}_{in}$ has a superset of edges of $\mathcal{S}$, and for any edge present in both graphs, the orientations are the same. Thus, if p is an ancestral path in $\mathcal{S}$, then p is a possibly ancestral path in $\mathcal{H}_{in}$. Similarly, if p is a possibly inducing path with respect to $\mathbf{L}$ in $\mathcal{S}$, then p is a possibly inducing path with respect to $\mathbf{L}$ in $\mathcal{H}_{in}$. ■

We can now prove that if a SMCM $\mathcal{S}$ entails all and only the observed conditional independencies for all experiments (and is therefore a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$), then $\mathcal{S}$ satisfies $\Phi \wedge \mathcal{F}$. We say that $\mathcal{S}$ *satisfies a constraint* ϕ if the truth-values assigned to *edge*, *arrow* and *tail* variables by their corresponding configuration in $\mathcal{S}$ satisfies ϕ. To simplify the proof, we first prove the following lemma:

Lemma 16 *If $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, and $X \rightarrow Y$ is in $\mathcal{P}_i$, then $\mathcal{S}$ satisfies $ancestor(X, Y, i)$. Similarly, if $X \circ\!\!\rightarrow Y$ is in $\mathcal{P}_i$, then $\mathcal{S}$ satisfies $\neg ancestor(Y, X, i)$.*

Proof By Theorem 14 SMCMtoMAG $(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i$. Thus, if $X \rightarrow Y$ is in $\mathcal{P}_i$, then X is an ancestor of Y in $\mathcal{S}^{\mathbf{I}_i}$ (there exists an ancestral path from X to Y in $\mathcal{S}^{\mathbf{I}_i}$). Let $p_1, \ldots, p_M$ be the possibly ancestral paths (there exists at least one: if $X \rightarrow Y$ in $\mathcal{P}_i$, then $X \ast\!\!\ast Y$ is a possibly inducing path in $\mathcal{H}_{in}$) from X to Y in $\mathcal{H}_{in}$. The constraint $ancestor(X, Y, i)$ is realized in $\Phi \wedge \mathcal{F}$ as $ancestor(Y, X, i) \wedge [ancestor(Y, X, i) \leftrightarrow ancestral(p_1, i) \vee ancestral(p_2, i) \cdots \vee ancestral(p_M, i)]$. This is equivalent to $ancestral(p_1, i) \vee ancestral(p_2, i) \cdots \vee ancestral(p_M, i)$. If a path is ancestral in $\mathcal{S}^{\mathbf{I}_i}$, the path is also ancestral in $\mathcal{S}$. By Lemma 15, if a path is ancestral in $\mathcal{S}$, the path is possibly ancestral in $\mathcal{H}_{in}$. Hence, at least one of $p_1, \ldots, p_M$ is ancestral in $\mathcal{S}^{\mathbf{I}_i}$, and $\mathcal{S}$ satisfies $ancestor(X, Y, i)$.

If $X \circ\!\!\rightarrow Y$ is in $\mathcal{P}_i$, then, since SMCMtoMAG $(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i$, there can be no ancestral path from Y to X in $\mathcal{S}^{\mathbf{I}_i})$. Let $p_1, \ldots, p_M$ be the possibly ancestral paths (if any) from Y to X in $\mathcal{H}_{in}$. The constraint $\neg ancestral(Y, X, i)$ is realized in $\Phi \wedge \mathcal{F}$ as $\neg ancestor(Y, X, i) \wedge [ancestor(Y, X, i) \leftrightarrow ancestral(p_1, i) \vee ancestral(p_2, i) \cdots \vee ancestral(p_M, i)]$. This is equivalent to $\neg ancestral(p_1, i) \wedge \neg ancestral(p_2, i) \cdots \wedge \neg ancestral(p_M, i)$. None of these paths are ancestral in $\mathcal{S}^{\mathbf{I}_i}$, therefore $\mathcal{S}$ satisfies $ancestor(X, Y, i)$. ■

We can now prove that any possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ satisfies $\Phi \wedge \mathcal{F}$.

Lemma 17 *For an oracle of conditional independence, if $\mathcal{S}$ is a possibly underlying model for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, and $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, $\mathcal{S}$ satisfies $\Phi \wedge \mathcal{F}$.*

Proof By Theorem 14, since $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i \quad \forall i$.

1. **Constraints added in Lines 8, 9 of Algorithm 4**. These constraints are satisfied since $\mathcal{S}$ is an acyclic mixed graph.

2. **Adjacency constraints added in Lines 4, 5, 6 of Algorithm 4**. Assume that for a pair of variables X, Y adjacent in $\mathcal{P}_i$, there exist M possibly inducing paths in $\mathcal{H}_{in}$, namely $p_1, , \ldots, p_M$. For this adjacency, the following constraint is added in $\Phi \wedge \mathcal{F}$ in Lines 4 and 5 of Algorithm 4:

   $$adjacent(X, Y, \mathcal{P}_i) \wedge [adjacent(X, Y, \mathcal{P}_i) \leftrightarrow inducing(p_1, i) \vee \cdots \vee inducing(p_M, i)],$$

   which is equivalent to

   $$inducing(p_1, i) \vee \cdots \vee inducing(p_M, i).$$

   Since $\mathcal{M}_i \in \mathcal{P}_i$, X and Y are adjacent in $\mathcal{M}_i$. By Proposition 12 there exists an inducing path p^* between X and Y with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. By Lemma 15, this path is a possibly inducing path in $\mathcal{H}_{in}$, thus, $\exists i \in [1, \ldots, M]$ such that $p^* = p_i$. Thus, the constraint $inducing(p_1, i) \vee \cdots \vee inducing(p_M, i)$ is satisfied by $\mathcal{S}$.

   Similarly, if X and Y are not adjacent in $\mathcal{P}_i$, the constraint

   $$\neg adjacent(X, Y, \mathcal{P}_i) \wedge [adjacent(X, Y, \mathcal{P}_i) \leftrightarrow inducing(p_1, i) \vee \cdots \vee inducing(p_M, i)]$$

   is added to $\Phi \wedge \mathcal{F}$ in Lines 4 and 6 of Algorithm 4. The constraint is equivalent to

   $$\neg inducing(p_1, i) \wedge \cdots \wedge \neg inducing(p_M, i).$$

   Since X and Y are not adjacent in $\mathcal{M}_i$, by Proposition 12 there exists no inducing path with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. Thus, none of the paths (if any) $p_1, \ldots, p_M$ is inducing with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$, and the constraint $\neg inducing(p_1, i) \wedge \cdots \wedge \neg inducing(p_M, i)$ is satisfied by $\mathcal{S}$.

3. **Unshielded (non) collider constraints added in Lines 13,14, 15,16 of Algorithm 4.** For an unshielded collider $X \star\!\!-\!\!\star Y \star\!\!-\!\!\star Z$ in $\mathcal{P}_i$, the constraint

$$col(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \big[col(\langle X,Y,Z\rangle,\mathcal{P}_i) \rightarrow unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i)\big],$$

which is equivalent to

$$unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i)$$

is added in Lines 14 and 15. As shown in Figure 4,

$$unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \leftrightarrow adjacent(X,Y,\mathcal{P}_i) \wedge adjacent(Y,Z,\mathcal{P}_i) \wedge \neg adjacent(X,Z,\mathcal{P}_i)$$

and

$$collider(\langle X,Y,Z\rangle,\mathcal{P}_i) \leftrightarrow \neg ancestor(Y,X,i) \wedge \neg ancestor(Y,Z,i)$$

. Since $\mathcal{M}_i \in \mathcal{P}_i$, $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ is an unshielded triple in $\mathcal{M}_i$, $adjacent(X,Y,\mathcal{P}_i) \wedge adjacent(Y,Z,\mathcal{P}_i) \wedge \neg adjacent(X,Z,\mathcal{P}_i)$ is satisfied (as described above for adjacency constraints). Since $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ in $\mathcal{P}_i$, by Lemma 16 constraints $\neg ancestor(Y,X,i) \wedge \neg ancestor(Y,Z,i)$ are satisfied by $\mathcal{S}$.

For an unshielded definite non collider $X \star\!\!-\!\!\star Y \star\!\!-\!\!\star Z$ in $\mathcal{P}_i$, the constraint

$$dnc(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \big[dnc(\langle X,Y,Z\rangle,\mathcal{P}_i) \rightarrow unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \neg collider(\langle X,Y,Z\rangle,\mathcal{P}_i)\big],$$

is added in Lines 13 and 16 of Algorithm 4, which is equivalent to

$$unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \neg collider(\langle X,Y,Z\rangle,\mathcal{P}_i).$$

Since $\mathcal{M}_i \in \mathcal{P}_i$, $X \star\!\!-\!\!\star Y \star\!\!-\!\!\star Z$ is an unshielded triple in $\mathcal{M}_i$, so $unshielded(\langle X,Y,Z\rangle,\mathcal{P}_i)$ is satisfied by $\mathcal{S}$ as described above. Moreover, since either $Y \rightarrow X$ in $\mathcal{M}_i$, or $Y \rightarrow Z$ in $\mathcal{M}_i$, by Lemma 16 $ancestor(Y,X,i) \vee ancestor(Y,Z,i)$ is satisfied by $\mathcal{S}$.

4. **Discriminating (non) collider constraints added in Lines 19, 20,21, 22 of Algorithm 4**. If $\langle W,\ldots,X,Y,Z\rangle$ is a discriminating path for Y in $\mathcal{P}_i$, and Y is a collider on the path in $\mathcal{P}_i$, the following constraint is added in $\Phi \wedge \mathcal{F}$ and in Lines 19 and 21 of Algorithm 4:

$$col(\langle X,Y,Z\rangle,\mathcal{P}_i) \wedge \big[col(\langle X,Y,Z\rangle,\mathcal{P}_i) \rightarrow discriminating(p_{WZ},Y,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i)\big],$$

which is equivalent to

$$discriminating(p_{WZ},Y,\mathcal{P}_i) \wedge collider(\langle X,Y,Z\rangle,\mathcal{P}_i).$$

Since $\mathcal{M}_i \in \mathcal{P}_i$, the path is discriminating for Y in $\mathcal{M}_i$ and the triple is a collider in $\mathcal{M}_i$. The constraint for the discriminating path is analyzed as a conjunction of the individual features ((non) adjacencies and endpoints) of the path as shown in Figure

4. Since the path is discriminating in $\mathcal{M}_i$, all these adjacency and ancestry constraints are satisfied by $\mathcal{S}$, by the proof for adjacency constraints and Lemma 16. In addition, the triple is a collider in $\mathcal{M}_i$, thus $collider(\langle X, Y, Z\rangle, \mathcal{P}_i)$ is satisfied by $\mathcal{S}$ as described for unshielded colliders.

Similarly, if $\langle W, \ldots, X, Y, Z\rangle$ is a discriminating path for Y in $\mathcal{P}_i$, and Y is a definite non collider on the path in $\mathcal{P}_i$, the following constraint is added in $\Phi \wedge \mathcal{F}$ and in Lines 20 and 22 of Algorithm 4:

$$dnc(\langle X, Y, Z\rangle, \mathcal{P}_i) \wedge \\ \big[dnc(\langle X, Y, Z\rangle, \mathcal{P}_i) \rightarrow discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge \neg collider(\langle X, Y, Z\rangle, \mathcal{P}_i)\big],$$

which is equivalent to

$$discriminating(p_{WZ}, Y, \mathcal{P}_i) \wedge \neg collider(\langle X, Y, Z\rangle, \mathcal{P}_i).$$

Since $\mathcal{M}_i \in \mathcal{P}_i$, the path is discriminating for Y in $\mathcal{M}_i$ and the triple is a non-collider in $\mathcal{M}_i$. The constraint for the discriminating path satisfied by $\mathcal{S}$ as described above. In addition, the triple is a non-collider in $\mathcal{M}_i$, thus $\neg collider(\langle X, Y, Z\rangle, \mathcal{P}_i)$ is satisfied by $\mathcal{S}$ as described for unshielded definite non colliders.

Thus, $\mathcal{S}$ satisfies all constraints in $\Phi \wedge \mathcal{F}$. ■

To prove completeness for Algorithm 4, we must show that the opposite also holds: If $\mathcal{S}$ is a truth-setting assignment of $\Phi \wedge \mathcal{F}$, $\mathcal{S}$ entails all and only the conditional independencies observed in $\{\mathcal{J}_i\}_{i=1}^N$ for each experiment. According to Theorem 14, we need to show that any truth setting assignment of $\Phi \wedge \mathcal{F}$ results, in each experiment i (after the respective procedures of manipulation, conversion to MAG and marginalization) in a MAG $\mathcal{M}_i$ that belongs to the Markov equivalence class represented by $\mathcal{P}_i$. Thus, we need to show that $\mathcal{M}_i$ has the same adjacencies and colliders with order as any MAG $\mathcal{M}' \in \mathcal{P}_i$. Proving that $\mathcal{M}_i$ and any $\mathcal{M}' \in \mathcal{P}_i$ have the same adjacencies is straight-forward. We then use induction to the order of the triple to show that the two MAGs also share the same colliders with order. The following lemma proves that discriminating paths with order are present in all members of the equivalence class, and therefore they are (definite) discriminating paths with order in $\mathcal{P}_i$ (Lemma 18.) Thus, all (non) colliders with order in $\mathcal{P}_i$ are identified and added to the SAT formula in Lines 19 and 20 of Algorithm 4.

Lemma 18 *If $p = \langle W, V_1, \ldots, V_n, Y, Q\rangle$ is a discriminating path with order r in $\mathcal{M}$, then the path is a discriminating path with order r in $\mathcal{P} = [\mathcal{M}]$.*

Proof We will show that the path is a discriminating path with order r in any $\mathcal{M}' \in \mathcal{P}$. Since $\mathcal{M}'$ and $\mathcal{M}$ are Markov equivalent, the two share the same colliders with order. Thus, every triple $\langle V_{i-1}, V_i, V_{i+1}\rangle$ is a collider with order in $\mathcal{M}$. Lemma 3.10 in Ali et al. (2009) states that if a path $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating for Y in a MAG $\mathcal{M}$, then in any Markov equivalent MAG $\mathcal{M}'$ in which V_i are colliders on the same path, $V_i \rightarrow Q$ in $\mathcal{M}'$ for $i = 1, \ldots, N$, and therefore the path is discriminating with order r in $\mathcal{M}'$. Thus, the path is discriminating with order r in all members of $[\mathcal{M}]$. It is therefore a discriminating path

with order r in $\mathcal{P}$. ■

We can now prove that any truth-setting assignment for $\Phi \wedge \mathcal{F}$ corresponds to a SMCM $\mathcal{S}$ that is possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.

Lemma 19 *For an oracle of conditional independence, if $\Phi \wedge \mathcal{F}$ is the conjunction of the outputs of Algorithm 4, and $\mathcal{S}$ a mixed graph that satisfies $\Phi \wedge \mathcal{F}$, then $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*

Proof We need to prove that (a) $\mathcal{S}$ is an acyclic mixed graph and (b) $\mathcal{M}_i = \text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[_{\mathbf{L}_i} \in \mathcal{P}_i \quad \forall i$. To prove the latter, we need to prove that for each i, if $\mathcal{M}' \in \mathcal{P}_i$, $\mathcal{M}_i$ and $\mathcal{M}'$ are Markov equivalent. Thus, we must show that $\mathcal{M}_i$ and $\mathcal{M}'$ share the same edges and colliders with order.

- **$\mathcal{S}$ is a SMCM:** $\mathcal{S}$ satisfies the constraints added in Lines 8 and 9 respectively. Therefore, $\mathcal{S}$ has no tail-tail edges, every endpoint is an arrow or a tail (not exclusively) and $\mathcal{S}$ has no directed cycles.

- **$\mathcal{M}_i$ and $\mathcal{M}'$ share the same edges**: If X and Y are adjacent in $\mathcal{M}'$, then X and Y are adjacent in $\mathcal{P}_i$. $\mathcal{S}$ satisfies the constraints added in Line 4 of Algorithm 4, therefore there exists an inducing path with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$. Thus, X and Y are adjacent in $\mathcal{M}_i$. If X and Y are not adjacent in $\mathcal{M}'$, X and Y are not adjacent in $\mathcal{P}_i$ and by the same constraints there exists no inducing path with respect to $\mathbf{L}_i$ in $\mathcal{S}^{\mathbf{I}_i}$, therefore X and Y are not adjacent in $\mathcal{M}_i$.

- **$\mathcal{M}_i$ and $\mathcal{M}'$ share the same colliders with order:** We will prove this by induction to order r: For order $= 0$, if $\langle X, Y, Z\rangle$ is an unshielded collider in $\mathcal{M}'$, the triple is an unshielded collider in $\mathcal{P}_i$. Since $\mathcal{M}'$ and $\mathcal{M}_i$ share the same edges, $X \star\!\!\rightarrow\!\!\star Y \star\!\!\leftarrow\!\!\star Z$ is an unshielded triple in $\mathcal{M}_i$. $\mathcal{S}$ satisfies the constraints added in Line 13 of Algorithm 4, and therefore Y is not an ancestor of X nor Z in $\mathcal{S}^{\mathbf{I}_i}$. Thus, $X \star\!\!\rightarrow Y \leftarrow\!\!\star Z$ in $\mathcal{M}_i$. If the triple is an unshielded collider in $\mathcal{M}_i$, then the triple is unshielded in $\mathcal{M}'$. If the triple is a non-collider in $\mathcal{M}'$, then $\mathcal{S}$ satisfies the constraints added in Line 14 of Algorithm 4, and Y is an ancestor of either X or Z in $\mathcal{S}^{\mathbf{I}_i}$. But then the triple is a non-collider in $\mathcal{M}_i$, which is a contradiction. Thus, $\mathcal{M}_i$ and $\mathcal{M}'$ share the same colliders with order 0.

  For the induction step, we assume that $\mathcal{M}_i$ and $\mathcal{M}'$ share the same colliders with order $s < r$. We will show that the two MAGs also share the same colliders with order r. We will first show that a path $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating for $\langle V_n, Y, Q\rangle$ with order r in $\mathcal{M}_i$ iff the path is discriminating for $\langle V_n, Y, Q\rangle$ with order r in $\mathcal{M}'$.

  If $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating with order r in $\mathcal{M}'$, by Lemma 18 the path is discriminating with order r in $\mathcal{P}_i$. $\mathcal{S}$ satisfies the constraints added in Lines 20 and 19 and therefore the path is discriminating in $\mathcal{M}_i$. Moreover, every triple on the path is a collider with order $< r$ in $\mathcal{M}'$ and by the induction hypothesis $\mathcal{M}'$ and $\mathcal{M}_i$ share the same colliders with order $< r$, thus the path has order r in $\mathcal{M}_i$.

  If $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is discriminating with order r in $\mathcal{M}_i$, then, by the induction hypothesis, every triple on the path is a collider with the same order $< r$ in $\mathcal{M}'$.

We will show that $V_i \to Q \quad \forall i$, and therefore $\langle W, V_1, \ldots, V_n, Y, Q\rangle$ is a discriminating path with order r in $\mathcal{M}'$.

The proof is similar to that of Lemma 3.10 in Ali et al. (2009). We will use induction on i. First, consider the (V_1, Q) edge in $\mathcal{M}'$. If $V_1 \leftarrow\!\star Q$, then $W \star\!\rightarrow V_1 \leftarrow\!\star Q$ forms a collider with order 0 in $\mathcal{M}'$, but an non-collider with order 0 in $\mathcal{M}_i$, which is a contradiction. Thus, $V_1 \rightarrow Q$ in $\mathcal{M}'$.

Suppose that $V_j \rightarrow Q$ for $1 \leq j \leq i$ in $\mathcal{M}'$. Then, the path $\langle W, V_1, \ldots, V_i, Q\rangle$ forms a discriminating path for V_i with the same order $< r$ in both graphs, and $\langle V_{i-1}, V_i, Q\rangle$ is a non-collider in $\mathcal{M}_i$. By Lemma 18, the path is a discriminating path with order in $\mathcal{P}_i$, and therefore $\Phi \wedge \mathcal{F}$ includes discriminating path constraints for this path added in Lines 19 and 21 or 20 and 22 of Algorithm 4. Thus, the triple can only be a non-collider in $\mathcal{M}_i$ if it is a non-collider in $\mathcal{M}'$. Since $V_{i-1} \leftrightarrow V_i$ in $\mathcal{M}'$, $V_i \rightarrow Q \quad \forall i$ and the path is discriminating in $\mathcal{M}'$ with order r.

We have shown that $\mathcal{M}_i$ and $\mathcal{M}'$ share the same discriminating paths with order r. It is now easy to show that a triple is a collider with order r in $\mathcal{M}'$ iff it is a collider with order r in $\mathcal{M}_i$. If $\langle V_n, Y, Z\rangle$ is a collider with order r in $\mathcal{M}'$, then there exists a discriminating path with order r in both graphs and in $\mathcal{P}_i$. Thus, $\mathcal{S}$ satisfies the constraints added in Lines 19 and 21 of Algorithm 4, by which Y is not an ancestor of V_n nor Q in $\mathcal{S}^{\mathbf{I}_i}$, and therefore the triple is a collider in $\mathcal{M}_i$, and it has order at most r. But by the induction hypothesis, the $\mathcal{M}'$ and $\mathcal{M}_i$ share the same colliders with order $< r$, thus the triple has order r in $\mathcal{M}_i$. Similarly, if the triple is a collider with order r in $\mathcal{M}_i$, there exists a discriminating path with order r in $\mathcal{M}$; and therefore in $\mathcal{P}_i$. Thus, $\mathcal{S}$ satisfies the constraints added in Lines 19 and 21 of Algorithm 4 or in Lines 20 and 22 of Algorithm 4. Hence, the triple must be in $\mathcal{M}'$, otherwise the triple would be a non-collider in $\mathcal{M}_i$. In addition, the triple has order at most r in $\mathcal{M}'$ and by the induction hypothesis the triple can not have order $< r$ in $\mathcal{M}'$, so the triple has order r in $\mathcal{M}'$. Thus, $\mathcal{M}'$ and $\mathcal{M}_i$ share the same colliders with order.

Thus, if $\mathcal{S}$ a mixed graph that satisfies $\Phi \wedge \mathcal{F}$, then $\mathcal{S}$ is a SMCM and $\text{SMCMtoMAG}(\mathcal{S}^{\mathbf{I}_i})[\mathbf{L}_i \in \mathcal{P}_i \quad \forall i$, so by Theorem 14, $\mathcal{S}$ is a possibly underlying SMCM for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$. ■

We can now prove soundness and completeness of Algorithm 2:

Theorem 20 (Soundness and completeness of Algorithm 2) *If $\mathcal{H}$ is the output of Algorithm 2, then the following hold:*
***Soundness**: If a feature (edge, absent edge, endpoint) is* solid *in $\mathcal{H}$, then this feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$.*
***Completeness**: If a feature is present in* all *SMCMs that are possibly underlying for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$, the feature is solid in $\mathcal{H}$.*

Proof *Soundness:* Solid features correspond to backbone variables. By Lemma 17 every possibly underlying SMCM $\mathcal{S}$ for $\{\mathcal{J}_i\}_{i=1}^N$ and $\{\mathbf{I}_i\}_{i=1}^N$ satisfies the final formula $\Phi \wedge \mathcal{F}$. Thus, if a core variable has the same value in all the possible truth-setting assignments of $\Phi \wedge \mathcal{F}$, this feature is present in all possibly underlying SMCMs. *Completeness:* By Lemma 19 the final formula $\Phi \wedge \mathcal{F}$ of Algorithm 2 is satisfied only by possibly underlying SMCMs. Thus,

if a core variable is present in *all* consistent SMCMs, the corresponding core variable will be a backbone variable for $\Phi \wedge \mathcal{F}$. ■

References

RA Ali, TS Richardson, and P Spirtes. Markov equivalence for ancestral graphs. *The Annals of Statistics*, 37(5B):2808–2837, 2009.

IA Beinlich, HJ Suermondt, RM Chavez, and GF Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Second European Conference on Artificial Intelligence in Medicine*, 1989.

SC Bendall, EF Simonds, P Qiu, El-ad D Amir, PO Krutzik, R Finck, RV Bruggner, R Melamed, A Trejo, OI Ornatsky, RS Balderas, SK Plevritis, K Sachs, D Peér, SD Tanner, and GP Nolan. Single-cell mass cytometry of differential immune and drug responses across a human hematopoietic continuum. *Science*, 332(6030):687–696, 2011.

B Bodenmiller, ER Zunder, R Finck, TJ Chen, ES Savig, RV Bruggner, EF Simonds, SC Bendall, K Sachs, PO Krutzik, et al. Multiplexed mass cytometry profiling of cellular states perturbed by small-molecule regulators. *Nature Biotechnology*, 30(9):858–867, 2012.

G Borboudakis, S Triantafillou, and I Tsamardinos. Tools and algorithms for causally interpreting directed edges in maximal ancestral graphs. In *Sixth European Workshop on Probabilistic Graphical Models*, 2012.

T Chu, C Glymour, R Scheines, and P Spirtes. A statistical problem for inference to regulatory structure from associations of gene expression measurements with microarrays. *Bioinformatics*, 19(9):1147–1152, 2003.

T Claassen and T Heskes. Causal discovery in multiple models from different experiments. In *Twenty-fourth Annual Conference on Neural Information Processing Systems*, 2010a.

T Claassen and T Heskes. Learning causal network structure from multiple (in) dependence models. In *Fifth European Workshop on Probabilistic Graphical Models*, 2010b.

T Claassen and T Heskes. A Bayesian approach to constraint based causal inference. In *Twenty-eighth Conference on Uncertainty in Artificial Intelligence*, 2012.

D Colombo, MH Maathuis, M Kalisch, and TS Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, 40 (1):294–321, 2012.

GF Cooper and Ch Yoo. Causal discovery from a mixture of experimental and observational data. In *Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

D Eaton and K Murphy. BDAGL: Bayesian DAG learning. `http://www.cs.ubc.ca/~murphyk/Software/BDAGL/`, 2007a.

D Eaton and KP Murphy. Exact bayesian structure learning from uncertain interventions. In *Eleventh International Conference on Artificial Intelligence and Statistics*, 2007b.

F Eberhardt and R Scheines. Interventions and causal inference. *Philosophy of Science*, 74 (5):981–995, 2007.

N Eén and N Sörensson. An extensible SAT-solver. In *Theory and Applications of Satisfiability Testing*, 2004.

RJ Evans and TS Richardson. Maximum likelihood fitting of acyclic directed mixed graphs to binary data. In *Twenty-sixth International Conference on Uncertainty in Artificial Intelligence*, 2010.

RJ Evans and TS Richardson. Marginal log-linear parameters for graphical markov models. *arXiv preprint arXiv:1105.6075*, 2011.

RA Fisher. *The Design of Experiments*. Hafner Publishing, New York, 1935.

D Geiger and D Heckerman. Learning Gaussian networks. In *Tenth Conference on Uncertainty in Artificial Intelligence*, 1994.

CP Gomes, B Selman, N Crato, and H Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1-2):67–100, 2000.

A Hauser and P Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13 (1):2409–2464, 2012.

A Hyttinen, F Eberhardt, and PO Hoyer. Learning linear cyclic causal models with latent variables. *Journal of Machine Learning Research*, 13:3387–3439, 2012a.

A Hyttinen, F Eberhardt, and PO Hoyer. Causal discovery of linear cyclic models from multiple experimental data sets with overlapping variables. In *Twenty-eighth Conference on Uncertainty in Artificial Intelligence*, 2012b.

A Hyttinen, PO Hoyer, F Eberhardt, and M Järvisalo. Discovering cyclic causal models with latent variables: A general SAT-based procedure. In *Twenty-ninth Conference on Uncertainty in Artificial Intelligence*, 2013.

S Itani, M Ohannessian, K Sachs, GP Nolan, and MA Dahleh. Structure learning in causal cyclic networks. In *Journal of Machine Learning Research Workshop and Conference Proceedings*, volume 6, pages 165 – 176, 2010.

A Kuegel. Improved exact solver for the weighted max-SAT problem. In *Workshop Pragmatics of SAT*, 2010.

S Meganck, S Maes, P Leray, and B Manderick. Learning semi-Markovian causal models using experiments. In *Third European Workshop on Probabilistic Graphical Models*, 2006.

JM Mooij and T Heskes. Cyclic causal discovery from continuous equilibrium data. In *Twenty-ninth Conference on Uncertainty in Artificial Intelligence*, 2013.

K Murphy. Active learning of causal Bayes net structure. Technical report, UC Berkeley, 2001.

J Ramsey, P Spirtes, and J Zhang. Adjacency faithfulness and conservative causal inference. In *Twenty-second Conference on Uncertainty in Artificial Intelligence*, 2006.

TS Richardson. Markov properties for acyclic directed mixed graphs. *Scandinavian Journal of Statistics*, 30(1):145–157, 2003.

TS Richardson and P Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030, 2002.

TS Richardson, JM Robins, and I Shpitser. Nested Markov properties for acyclic directed mixed graphs. In *Twenty-eighth Conference on Uncertainty in Artificial Intelligence*. 2012.

K Sachs, O Perez, D Pe'er, DA Lauffenburger, and GP Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

K Sadeghi. *Graphical Representation of Independence Structures*. PhD thesis, Oxford University, 2012.

T Sellke, MJ Bayarri, and JO Berger. Calibration of p-values for testing precise null hypotheses. *The American Statistician*, 55(1):62–71, 2001.

I Shpitser, R Evans, TS Richardson, and JM Robins. Sparse nested Markov models with log-linear parameters. In *Twenty-ninth Conference on Uncertainty in Artificial Intelligence*. 2013.

ME Smoot, K Ono, J Ruscheinski, PL Wang, and T Ideker. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics*, 27(3):431–432, 2011.

P Spirtes and TS Richardson. A polynomial time algorithm for determining DAG equivalence in the presence of latent variables and selection bias. In *Sixth International Workshop on Artificial Intelligence and Statistics*, 1996.

P Spirtes, C Glymour, and R Scheines. *Causation, Prediction, and Search*. The MIT Press, second edition, 2001.

JD Storey and R Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440, 2003.

J Tian and J Pearl. On the identification of causal effects. Technical Report R-290-L, UCLA Cognitive Systems Laboratory, 2003.

RE Tillman and P Spirtes. Learning equivalence classes of acyclic models with latent and selection variables from multiple datasets with overlapping variables. In *Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

RE Tillman, D Danks, and C Glymour. Integrating locally learned causal structures with overlapping variables. In *Twenty-Second Annual Conference on Neural Information Processing Systems*, 2008.

S Tong and D Koller. Active learning for structure in Bayesian networks. In *Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

S Triantafillou, I Tsamardinos, and IG Tollis. Learning causal structure from overlapping variable sets. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010.

I Tsamardinos, S Triantafillou, and V Lagani. Towards integrative causal analysis of heterogeneous data sets and studies. *Journal of Machine Learning Research*, 13:1097–1157, 2012.

TS Verma and J Pearl. Equivalence and synthesis of causal models. Technical Report R-150, UCLA Department of Computer Science, 2003.

J Zhang. *Causal Inference and Reasoning in Causally Insufficient Systems*. PhD thesis, Carnegie Mellon University, 2006.

J Zhang. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence*, 172(16-17):1873–1896, 2008a.

J Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9(1):1437–1474, 2008b.

Journal of Machine Learning Research 16 (2015) 2207-2230 Submitted 8/14; Revised 11/14; Published 11/15

Existence and Uniqueness of Proper Scoring Rules

Evgeni Y. Ovcharov TRULR6@YAHOO.COM
Heidelberg Institute for Theoretical Studies
Schloss-Wolfsbrunnenweg 35, D-69118 Heidelberg
Germany

Editor: Zhihua Zhang

Abstract

To discuss the existence and uniqueness of proper scoring rules one needs to extend the associated entropy functions as sublinear functions to the conic hull of the prediction set. In some natural function spaces, such as the Lebesgue L^p-spaces over $\mathbb{R}^d$, the positive cones have empty interior. Entropy functions defined on such cones have directional derivatives only, which typically exist on large subspaces and behave similarly to gradients. Certain entropies may be further extended continuously to open cones in normed spaces containing signed densities. The extended entropies are Gâteaux differentiable except on a negligible set and have everywhere continuous subgradients due to the supporting hyperplane theorem. We introduce the necessary framework from analysis and algebra that allows us to give an affirmative answer to the titular question of the paper. As a result of this, we give a formal sense in which entropy functions have uniquely associated proper scoring rules. We illustrate our framework by studying the derivatives and subgradients of the following three prototypical entropies: Shannon entropy, Hyvärinen entropy, and quadratic entropy.

Keywords: proper scoring rules, entropy, characterisation, existence, uniqueness, quasi-interior, directional derivative, Gâteaux derivative, subgradient, sublinear, convex analysis

1. Introduction

Proper scoring rules have attracted a lot of interest in recent years in disparate fields such as statistics, decision theory, machine learning, game theory, finance, meteorology, etc. They provide practical measures for assessing the accuracy and precision of probabilistic forecasts. In this paper, we build a general measure-theoretic framework for proper scoring rules that allows us to consider their existence and uniqueness as subgradients of sublinear functions.

1.1 Definitions

Let $(\Omega, \mathcal{A}, \mu)$ be a measure space and $\mathcal{P}$ be a convex set of probability densities on Ω with respect to the measure μ. A random variable X takes values in Ω with unknown true density $p \in \mathcal{P}$. We refer to $\mathcal{P}$ and its elements as a *prediction set* and *predictive densities* for X, respectively. By $\mathcal{L}(\mathcal{P})$ we denote the set of all μ-measurable functions $f : \Omega \to \mathbb{R}$ such that

$$\int_\Omega |f(x)|\, p(x) d\mu(x) < \infty$$

for all $p \in \mathcal{P}$. We call the elements of $\mathcal{L}(\mathcal{P})$ $\mathcal{P}$-*integrable* functions.

A *scoring rule* $S : \mathcal{P} \to \mathcal{L}(\mathcal{P})$ assigns for each predictive density $q \in \mathcal{P}$ a $\mathcal{P}$-integrable function $S(q)$. The value of $S(q)$ at $x \in \Omega$ is interpreted as a numerical score assigned to the outcome x. We take scoring rules to be *positively orientated*, that is, they are viewed as incentives which a forecaster wishes to maximise. It is customary to term S *proper* if the expected value of S at q,

$$p \cdot S(q) := \int_\Omega S(q)(x)p(x)d\mu(x),$$

is maximised in q at the true density $q = p$, and *strictly proper*, if the true density is the only maximiser.

Strictly proper scoring rules could be used as a bonus system under which truth-telling is the only optimal long-term strategy (Gneiting and Raftery, 2007). For such an S, the optimal expected reward is the *(negative) entropy* induced by S,

$$\Phi : \mathcal{P} \to \mathbb{R}, \quad \Phi(p) = p \cdot S(p),$$

(Parry et al., 2012). In what follows, we refer to Φ simply as the entropy function associated to S, as there is no danger of confusion between negative and positive entropy functions in the present context. The *regret* for quoting q instead of the true density p is expressed by the function

$$D : \mathcal{P} \times \mathcal{P} \to \mathbb{R}, \quad D(p, q) = p \cdot S(p) - p \cdot S(q),$$

which in the statistics literature is also known as the *divergence* induced by S. In the present paper, we shall use the notions of entropy and divergence in a more general sense by replacing strict propriety with propriety.

General overviews of proper scoring rules may be found in Gneiting and Raftery (2007); Gneiting and Katzfuss (2014) in connection to probabilistic forecasting, and also in Dawid and Musio (2014), where the emphasis is on statistical inference. Theoretical aspects of proper scoring rules are studied in Dawid (2007); Grünwald and Dawid (2004); Williamson (2014). Frongillo and Kash (2014) investigate proper scoring rules in connection with the elicitation of private information. The remaining references throughout the text provide links to more specific uses of scoring rules.

1.2 Motivation and Scope of the Paper

In this paper we adopt the theoretical framework of Hendrickson and Buehler (1971). This approach is characterised by exploiting a beautiful connection with *Euler's homogeneous function theorem*, which presupposes that we extend our quantities of interest as homogeneous functions to the conic hull of the prediction set. To that end, we introduce the *prediction cone* $\mathcal{P}^+ = \{\lambda p \,|\, \lambda > 0, p \in \mathcal{P}\}$ and extend S and Φ to $\mathcal{P}^+$ as homogeneous functions of degrees zero and one, respectively. Any $\mathcal{P}$-integrable function q^* satisfying

$$\Phi(p) \geq p \cdot q^*, \quad \forall p \in \mathcal{P}^+,$$

with equality for $p = q$, is called a *$\mathcal{P}$-integrable subgradient* of Φ at q. The subgradient is called *strict* if the above inequality is strict for all $p \in \mathcal{P}^+$ not positively collinear to q. Suppose that Φ has a subgradient $S(q) \in \mathcal{L}(\mathcal{P})$ at each $q \in \mathcal{P}^+$ and the resulting

map $S : \mathcal{P}^+ \to \mathcal{L}(\mathcal{P})$ is homogeneous of degree zero. We call S a *$\mathcal{P}$-integrable subgradient* of Φ on $\mathcal{P}^+$. We recall that a (strictly) convex homogeneous function of degree one is a *(strictly) sublinear function*. We may now state Hendrickson and Buehler's classical result in a slightly more contemporary language.

Theorem 1.1 *Let $\mathcal{P}$ be a prediction set with respect to the measure space $(\Omega, \mathcal{A}, \mu)$. A scoring rule $S : \mathcal{P}^+ \to \mathcal{L}(\mathcal{P})$ is (strictly) proper if and only if there is a (strictly) sublinear function $\Phi : \mathcal{P}^+ \to \mathbb{R}$ such that S is a subgradient of Φ on $\mathcal{P}^+$.*

Theorem 1.1 provides us with a basic but insufficient theoretical framework to discuss the titular question of this paper. In support of this claim, in Example B.2 we show the existence of a sublinear function that has unique but non-$\mathcal{P}$-integrable subgradients at some points of its domain, while at other points it has multiple $\mathcal{P}$-integrable subgradients. The most important structure missing in Theorem 1.1 is the notion of *interior* of a convex domain, which lies at the intersection of geometry, algebra, and topology, and may have different incarnations depending on the context (Borwein and Vanderwerff, 2010; Rockafellar, 1972). For example, studying proper local scoring rules on discrete sample spaces, Dawid et al. (2012) apply Theorem 1.1 in a context where the prediction cone is the interior of the positive orthant in $\mathbb{R}^d$. In this case, well-known results from convex analysis give necessary and sufficient conditions for an affirmative answer to our basic question. The real focus of our paper is thus the non-Euclidean case in the abstract measure-theoretic setting introduced above.

In Proposition 2.4 and Example B.3, we show that at boundary points sublinear functions have either no subgradient, or infinitely many. Therefore, it is paramount to try to define entropy functions on interiors of positive cones. In infinite dimensions, however, this is not always possible. Indeed, it is well-known that the positive cones in many natural function spaces (such as the Lebesgue L^p-spaces over $\mathbb{R}^d$) have empty interiors (Borwein and Lewis, 1992) and are negligible sets in terms of Baire category. This calls for a more subtle approach to our problem in which we need to refine our notion of interior and boundary. Inspired by geometric functional analysis, we adapt an algebraic refinement of the notion of interior of convex sets, whose better known topological analogues are often referred to as *quasi-interior* (Fullerton and Braunschweiger, 1963; Borwein and Lewis, 1992). Common entropies whose domains are positive cones with empty interior but nonempty quasi-interior are the *Shannon entropy*, the *Hyvärinen entropy*, and in principle, the entropies associated with the *proper local scoring rules of arbitrary orders*. These entropies are formally not differentiable functions but possess directional derivatives on large subspaces, which display similar properties to standard gradients.

Other entropies, such as those that are associated with the families of *power scoring rules* and *pseudospherical scoring rules* may be extended continuously to open cones in normed spaces that contain signed densities. Geometrically, this setting is similar to the Euclidean setting. One applies the supporting hyperplane theorem and other standard results in analysis relating subgradients and Gâteaux derivatives. The latter entropies are Gâteaux differentiable (either everywhere or outside a negligible set), which we illustrate in the context of the *quadratic scoring rule*.

The original part of the paper is concerned with the analysis of the notion of $\mathcal{P}$-integrable subgradient introduced by Hendrickson and Buehler (1971) and the associated most ba-

sic general framework for proper scoring rules. To address the question of existence and uniqueness of proper scoring rules, we equip this framework with a notion of algebraic quasi-interior. As an illustration, we show that the Hyvärinen scoring rule is the unique 0-homogeneous $\mathcal{P}$-integrable subgradient of its entropy function on the (non-empty) quasi-interior of a suitable positive cone.

The paper is organised as follows. In Section 2, we introduce the notation and present all the background facts. Section 3 contains our main results which formulate necessary and sufficient conditions for existence and uniqueness of subgradients of entropy functions. In Section 4, we illustrate the theory with applications to three prototypical entropy functions, namely, the Shannon, Hyvärinen, and quadratic entropy. These examples formalise the meaning with respect to which we may consider each entropy to have a uniquely associated proper scoring rule. We complete the main part of the paper in Section 5 with some closing remarks. The proofs of all formal assertions made in the text are given in Appendix A. In Appendix B, we present additional facts that illustrate various points made in the Introduction or later in the text.

2. Notation and Preliminaries

Let E, E_1, E_2 be sets of μ-measurable functions on Ω. For $\alpha \in \mathbb{R}$, we use the notation

$$\alpha E_1 = \{\alpha f \,|\, f \in E_1\}$$
$$E_1 + E_2 = \{f + g \,|\, f \in E_1, g \in E_2\}.$$

The (*blunt*) *cone* of E is the set $E^+ = \{\lambda f \,|\, \lambda > 0, f \in E\}$, while the *pointed cone* of E is the set $E^+ \cup \{0\}$. The *convex hull* of E,

$$\operatorname{co} E = \left\{ \sum_{i=1}^{k} \alpha_i f_i \,\middle|\, k \geq 1,\ f_i \in E,\ \alpha_i \geq 0,\ \sum_{i=1}^{k} \alpha_i = 1 \right\},$$

is the set of all convex combinations of elements of E. The *conic hull* of E,

$$\operatorname{cone} E = \left\{ \sum_{i=1}^{k} \alpha_i f_i \,\middle|\, k \geq 1,\ f_i \in E,\ \alpha_i \geq 0 \right\},$$

is the set of all conic combinations of elements of E. By

$$\operatorname{span} E = \left\{ \sum_{i=1}^{k} \alpha_i f_i \,\middle|\, k \geq 1,\ f_i \in E,\ \alpha_i \in \mathbb{R} \right\}$$

we denote the set of all linear combinations of elements of E, and we refer to it as the *linear span* of E.

A set E is called *convex* if $\operatorname{co} E = E$, a *cone* if $E = E^+$ or $E = E^+ \cup \{0\}$, a *convex cone* if $E = \operatorname{cone} E$ or $E = \operatorname{cone} E \setminus \{0\}$, and a *linear space* if $E = \operatorname{span} E$. If E is convex, $E^+ = \operatorname{cone} E \setminus \{0\}$ is a convex cone.

The *epigraph* of $\Phi : E \to \mathbb{R}$ is the set in $\operatorname{span} E \times \mathbb{R}$ given by

$$\operatorname{epi} \Phi = \{(f, y) \,|\, f \in E,\ y \in \mathbb{R},\ y \geq \Phi(f)\}.$$

The *graph* of Φ is the set $\{(f, \Phi(f)) \,|\, f \in E\}$.

A function $\Phi : E \to \mathbb{R}$ is called *convex* if its epigraph is a convex set. The definition implies that E is convex. Therefore, Φ is convex if, for any $f, g \in E$ and $\lambda \in (0, 1)$, Φ satisfies

$$\Phi((1-\lambda)f + \lambda g) \leq (1-\lambda)\Phi(f) + \lambda\Phi(g).$$

If the inequality is strict for $f \neq g$, then Φ is called *strictly convex*.

A function $\Phi : E^+ \to \mathbb{R}$ is said to be *(positively) homogeneous of degree k*, for $k \in \mathbb{R}$, or *(positively) k-homogeneous*, if for every $f \in E^+$ and every $\lambda > 0$, it holds $\Phi(\lambda f) = \lambda^k \Phi(f)$. A function $\Phi : E \to \mathbb{R}$ is said to be *subadditive* if Φ satisfies

$$\Phi(f + g) \leq \Phi(f) + \Phi(g)$$

for all $f, g \in E$, and *strictly subadditive*, if the above inequality is strict for $f \neq g$. We need to modify slightly the latter definition in the case when $\Phi : E^+ \to \mathbb{R}$ is 1-homogeneous. Then we say that Φ is *strictly subadditive* if the above inequality is strict whenever $f, g \in E^+$ are not positively collinear. Functions that are 1-homogeneous and (strictly) subadditive are called *(strictly) sublinear*. It is easy to see that $\Phi : E^+ \to \mathbb{R}$ is (strictly) sublinear if and only if Φ is (strictly) convex on E and 1-homogeneous on E^+.

Let $\mathcal{P}$ be a prediction set with respect to $(\Omega, \mathcal{A}, \mu)$ and let $E \subset \operatorname{span} \mathcal{P}$. By $E^\perp$ we denote the *annihilator* of E in $\mathcal{L}(\mathcal{P})$, that is, all $f \in \mathcal{L}(\mathcal{P})$ such that

$$p \cdot f = 0$$

for all $p \in E$. Clearly, $E^\perp$ is a linear subspace of $\mathcal{L}(\mathcal{P})$. In the case when $E^\perp = \{0\}$, we say that E has a *trivial annihilator*.

By a *direction* in a vector space we understand the equivalence class of all positively collinear vectors to a given nonzero vector. Note that any 0-homogeneous function is a function of directions. For $q \in \mathcal{P}^+$, we define the *set of directions* from q to the points in $\mathcal{P}^+$ as

$$\begin{aligned}\mathcal{D}(q) &= \{p \in \operatorname{span} \mathcal{P} \,|\, \exists \epsilon_p > 0,\, \forall t \in [0, \epsilon_p],\, q + tp \in \mathcal{P}^+\} \\ &= \{p \in \operatorname{span} \mathcal{P} \,|\, \exists \epsilon_p > 0,\, q + \epsilon_p p \in \mathcal{P}^+\}.\end{aligned}$$

We have the latter identity due to the convexity of $\mathcal{P}^+$.

A point $q \in \mathcal{P}^+$ is an *algebraically interior* point of $\mathcal{P}^+$ if $\mathcal{D}(q) = \operatorname{span} \mathcal{P}$. The collection of all algebraically interior points of $\mathcal{P}^+$ is called the *algebraic interior* of $\mathcal{P}^+$. In the case of a topological vector space, the topological interior of a set is always contained in the algebraic interior of the set. Moreover, when the topological interior is not empty, the two notions coincide. If q is not algebraically interior for $\mathcal{P}^+$, that is, $\mathcal{D}(q) \neq \operatorname{span} \mathcal{P}$, we say that q is a *boundary point* for $\mathcal{P}^+$. If $\mathcal{P}^+$ has empty algebraic interior, then the prediction cone consists entirely of boundary points. This case occurs frequently in the context of continuous sample spaces, see e.g. Proposition B.1.

Lemma 2.1 *For each $q \in \mathcal{P}^+$, we have the representation*

$$\mathcal{D}(q) = \operatorname{cone}(\mathcal{P}^+ - q).$$

For a point $q \in \mathcal{P}^+$, we define $\mathcal{O}(q) = \mathcal{D}(q) \cap -\mathcal{D}(q)$. This is the subset of directions in $\mathcal{D}(q)$ whose inverse is also in $\mathcal{D}(q)$. The set may be identified with these directions in $\operatorname{span}\mathcal{P}$ along which there is an open line segment that contains q and is contained in $\mathcal{P}^+$. Clearly, q is algebraically interior for $\mathcal{P}^+$ if and only if $\mathcal{O}(q) = \mathcal{D}(q) = \operatorname{span}\mathcal{P}$. By construction, $\mathcal{O}(q)$ is a linear subspace of $\operatorname{span}\mathcal{P}$. The sets of directions $\mathcal{D}(q)$ and $\mathcal{O}(q)$ are instrumental for defining various notions of directional derivatives.

The most basic directional derivative is the following one.

Definition 2.2 *For a function $\Phi : \mathcal{P}^+ \to \mathbb{R}$, the* right directional derivative *of Φ at $q \in \mathcal{P}^+$ along $p \in \mathcal{D}(q)$ is defined as*

$$\Phi'_+(p, q) = \lim_{t \to 0^+} \frac{\Phi(q + tp) - \Phi(q)}{t} \tag{1}$$

if the limit exists.

We gather below the main properties of $\Phi'_+(p, q)$.

Proposition 2.3 *Let $\Phi : \mathcal{P}^+ \to \mathbb{R}$ be a sublinear function and $q \in \mathcal{P}^+$. We have*

(a) for each $p \in \mathcal{D}(q)$,

$$\Phi'_+(p, q) = \inf_{t>0} \frac{\Phi(q + tp) - \Phi(q)}{t} \in \mathbb{R} \cup \{-\infty\},$$

and the infimum is finite for $p \in \mathcal{O}(q)$;

(b) $\Phi'_+(\cdot, q) : \mathcal{D}(q) \to \mathbb{R} \cup \{-\infty\}$ is sublinear;

(c) for each $\lambda > 0$, $\Phi'_+(p, \lambda q) = \Phi'_+(p, q)$;

(d) for each $p \in \mathcal{P}^+$,

$$\Phi(p) \geq \Phi'_+(p, q),$$

with equality for $p = q$;

(e) for each $p \in \mathcal{O}(q)$, $-\Phi'_+(-p, q) \leq \Phi'_+(p, q)$;

(f) the set

$$\mathcal{O}'(q) = \{p \in \mathcal{O}(q) \mid -\Phi'_+(-p, q) = \Phi'_+(p, q)\}$$

is a linear subspace of $\mathcal{O}(q)$ and the restriction $\Phi'_+(\cdot, q)\big|_{\mathcal{O}'(q)}$ is linear.

We next consider the other two types of directional derivatives. First, if we take the limit (1) with the restriction $t \leq 0$ instead $t \geq 0$, we obtain the *left directional derivative* of Φ, denoted $\Phi'_-(\cdot, q)$. It is easy to see that $\Phi'_-(\cdot, q)$ can be defined on $\mathcal{O}(q)$ and we have $\Phi'_-(p, q) = -\Phi'_+(-p, q)$, for each $p \in \mathcal{O}(q)$. Thus part (e) above can be rewritten as

$$\Phi'_-(p, q) \leq \Phi'_+(p, q)$$

for all $p \in \mathcal{O}(q)$. On the subspace $\mathcal{O}'(q)$ introduced above in part (f), we have that

$$\Phi'_-(\cdot, q) = \Phi'_+(\cdot, q)$$

is in fact the *two-sided directional derivative* of Φ at q, denoted $\Phi'(\cdot, q)$. The latter can be defined as the limit (1) without any restriction on t. In the most important case in practice, we have that $\mathcal{O}(q) = \mathcal{O}'(q)$. If in addition $\mathcal{O}(q) \neq \operatorname{span} \mathcal{P}$, then Φ has no standard functional derivative. For an illustration of this fact in the context of Shannon and Hyvärinen entropies, see Section 4.

By $\operatorname{Lin} \mathcal{P}$ we denote the space of all real-valued linear functionals on $\operatorname{span} \mathcal{P}$, i.e., the algebraic dual of $\operatorname{span} \mathcal{P}$. By "$\cdot$" we denote the bilinear pairing on $\operatorname{span} \mathcal{P} \times \operatorname{Lin} \mathcal{P}$, so if $q \in \operatorname{span} \mathcal{P}$ and $q^* \in \operatorname{Lin} \mathcal{P}$, $q \cdot q^*$ is the value of q^* at q.

Let $\Phi : \mathcal{P}^+ \to \mathbb{R}$ be 1-homogeneous. We say that $q^* \in \operatorname{Lin} \mathcal{P}$ is a *subgradient* of Φ at q if

$$\Phi(p) \geq p \cdot q^*$$

for all $p \in \mathcal{P}^+$, with equality for $p = q$. The collection of all subgradients of Φ at q is called the *subdifferential* of Φ at q and is denoted by $\partial\Phi(q)$. A subgradient q^* is *strict* if and only if the inequality $\Phi(p) > p \cdot q^*$ holds for all $p \in \mathcal{P}^+$ not positively collinear with q.

If $h \in \operatorname{Lin} \mathcal{P}$, the *hyperplane* H in $\operatorname{span} \mathcal{P} \times \mathbb{R}$ given by

$$z = p \cdot h, \quad \forall p \in \operatorname{span} \mathcal{P},$$

supports Φ at q if the epigraph of Φ lies above H, and H contains the point $(q, \Phi(q))$. Clearly, H supports Φ at q if and only if $h \in \partial\Phi(q)$.

The following proposition describes the intimate connection between one-sided and two-sided directional derivatives and the subdifferential of a sublinear function.

Proposition 2.4 *For a point $q \in \mathcal{P}^+$, we have*

(a) $q^* \in \partial\Phi(q)$ if and only if

$$p \cdot q^* \leq \Phi'_+(p, q)$$

for all $p \in \mathcal{P}^+$, with equality for $p = q$;

(b) if $\mathcal{D}(q) = \operatorname{span} \mathcal{P}$ and $\Phi'(\cdot, q)$ exists on $\operatorname{span} \mathcal{P}$, then $\partial\Phi(q) = \{\Phi'(\cdot, q)\}$;

(c) if $\mathcal{D}(q) = \operatorname{span} \mathcal{P}$ and $\Phi'(\cdot, q)$ does not exist on $\operatorname{span} \mathcal{P}$, then $\partial\Phi(q)$ has multiple elements;

(d) if $\mathcal{D}(q) \neq \operatorname{span} \mathcal{P}$ and $\Phi'_+(p, q)$ is finite for all $p \in \mathcal{P}^+$, then $\partial\Phi(q)$ has multiple elements;

(e) if $\mathcal{D}(q) \neq \operatorname{span} \mathcal{P}$ and there is $p \in \mathcal{P}^+$ such that $\Phi'_+(p, q) = -\infty$, then $\partial\Phi(q) = \emptyset$.

Part (a) above is the standard characterisation of the subdifferential of a sublinear function. Parts (b) and (c) give additional information in the case of algebraically interior points. Parts (d) and (e) do the same for boundary points. Notice that the latter imply the statement from the Introduction that at boundary points either the existence or uniqueness of subgradient fails. (See also Example B.3.) In the next section, we show that uniqueness might be sometimes recovered at certain boundary points if we confine ourselves to a regularity class such as $\mathcal{L}(\mathcal{P})$.

We next give a formal definition of a scoring rule and elaborate some of its implications.

Definition 2.5 *Let $\mathcal{P}$ be a prediction set with respect to the measure space $(\Omega, \mathcal{A}, \mu)$. Any 0-homogeneous map $S : \mathcal{P}^+ \to \mathcal{L}(\mathcal{P})$ is called a* scoring rule.

If X is a random variable on Ω with unknown true density $p \in \mathcal{P}$, then for each predictive density $q \in \mathcal{P}^+$, $S(q)(X)$ is a random function of X. The condition $S(q) \in \mathcal{L}(\mathcal{P})$ guarantees that the expectation of S is always finite. The *uncertainty function* associated to S is the function $\Phi : \mathcal{P}^+ \to \mathbb{R}$, $\Phi(p) = p \cdot S(p)$. Clearly, Φ is 1-homogeneous. When S is proper, it is customary to call Φ an *entropy function.*

Suppose now that $S : \mathcal{P}^+ \to \mathcal{L}(\mathcal{P})$ is a proper scoring rule with entropy Φ. The condition that the expected score of S is maximised in q at the true density $q = p$ means that S satisfies the inequality

$$\Phi(p) \geq p \cdot S(q),$$

for each $p, q \in \mathcal{P}^+$, with equality for $q = p$. If S is strictly proper, then p is the only maximiser up to a scaling factor. In this case, the inequality above is strict for any q that is not positively collinear to p. So, the assumption of propriety is equivalent to S being a subgradient of Φ on $\mathcal{P}^+$. Moreover, strict propriety corresponds to strict subgradients on $\mathcal{P}^+$. The existence of a subgradient on $\mathcal{P}^+$ implies that Φ is sublinear, see Lemma A.1. We conclude that (strictly) proper scoring rules are $\mathcal{P}$-integrable subgradients of (strictly) sublinear functions. Therefore, it is reasonable in the context of scoring rules to restrict the notion of subgradient to the class $\mathcal{L}(\mathcal{P}) \subset \mathrm{Lin}(\mathcal{P})$. In the next section, and in particular in Theorem 3.1 and Theorem 3.2, we discuss the existence and uniqueness of $\mathcal{P}$-integrable subgradients.

In some special cases, we may add to our notion of subgradient a topological structure. Let $\mathcal{P}^+$ be a prediction cone such that $\operatorname{span} \mathcal{P}$ may be identified with a normed space $(N, \|\cdot\|)$, and let the continuous dual of N, denoted N^*, be a subset of $\mathcal{L}(\mathcal{P})$. Suppose that $\mathcal{P}^+ \subset \mathcal{C}$, where $\mathcal{C}$ is an open convex cone in N, and Φ may be extended to $\mathcal{C}$ as a continuous sublinear function.

We recall that Φ is *Gâteaux differentiable* at $q \in \mathcal{C}$ if there is $q^* \in N^*$ such that for every $p \in N$, the limit

$$p \cdot q^* = \lim_{t \to 0} \frac{\Phi(q + tp) - \Phi(q)}{t}$$

exists. The functional q^* is called the *Gâteaux derivative* of Φ at q and is also denoted by $\nabla\Phi(q)$. Notice that by definition the Gâteaux derivative is applicable only to interior points. See Theorems 3.3 and 3.4 for an answer to our two main questions.

If Φ is Gâteaux differentiable at q, taking $p = q$ in the above limit, we recover *Euler's homogeneous function* theorem

$$q \cdot \nabla\Phi(q) = \Phi(q).$$

More generally, if Φ is sublinear and has a subgradient S on $\mathcal{P}^+$, then we have that $q \cdot S(q) = \Phi(q)$, for every $q \in \mathcal{P}^+$, (Hendrickson and Buehler, 1971). The proof also follows from Proposition 2.4 (a) and Proposition 2.3 (d). This beautiful generalisation of Euler's theorem is only visible after extending S and Φ to denormalised densities as homogeneous functions.

Suppose now that a scoring rule $S : \mathcal{P} \to \mathcal{L}(\mathcal{P})$ is given. Then, setting

$$S(q) = S\left(\frac{q}{q \cdot 1}\right)$$

for any $q \in \mathcal{P}^+$, extends S as a 0-homogeneous function to the prediction cone. Here

$$q \cdot 1 = \int_\Omega q(x) d\mu(x)$$

is the normalising constant of q. Similarly, let an entropy function $\Phi : \mathcal{P} \to \mathbb{R}$ be given. Setting

$$\Phi(q) = (q \cdot 1)\Phi\left(\frac{q}{q \cdot 1}\right)$$

for any $q \in \mathcal{P}^+$, extends Φ as a 1-homogeneous function to the prediction cone. See Section 4 for an illustration. Working directly with denormalised predictive densities could also be advantageous in numerical computation (Hyvärinen, 2005, 2007; Dawid and Musio, 2012, 2014).

3. Main Results

Our first result gives a necessary and sufficient condition for existence of a $\mathcal{P}$-integrable subgradient at a point. The result can be easily generalised to subgradients on $\mathcal{P}^+$.

Theorem 3.1 *Let $\Phi : \mathcal{P}^+ \to \mathbb{R}$ be a sublinear function. Then Φ has a $\mathcal{P}$-integrable subgradient at a point $q \in \mathcal{P}^+$ if and only if there is $q^* \in \mathcal{L}(\mathcal{P})$ such that*

$$p \cdot q^* \leq \Phi'_+(p, q)$$

for all $p \in \mathcal{P}^+$, with equality for $p = q$.

In the light of Theorem 1.1 and the above result, we call any sublinear function Φ an *entropy* if Φ has a $\mathcal{P}$-integrable subgradient at each point of its domain. In most cases of practical interest, one may choose the prediction cone appropriately so that $\Phi'_+(\cdot, q) = q^*$ for some $q^* \in \mathcal{L}(\mathcal{P})$. This means that $\Phi'_+(\cdot, q)$ is a $\mathcal{P}$-integrable subgradient of Φ at q and that $\Phi'_+(\cdot, q) = \Phi'(\cdot, q)$ is also a two-sided directional derivative on the subspace $\mathcal{O}(q)$ of $\operatorname{span}\mathcal{P}$. In our next result, we show that if $\mathcal{O}(q)$ is a sufficiently large subspace, then $\Phi'_+(\cdot, q)$ is the unique $\mathcal{P}$-integrable subgradient of Φ at q.

Theorem 3.2 *Let $\mathcal{P}$ be a prediction set and $\Phi : \mathcal{P}^+ \to \mathbb{R}$ be a sublinear function. Suppose that at a point $q \in \mathcal{P}^+$ the subspace $\mathcal{O}(q)$ of $\operatorname{span}\mathcal{P}$ has a trivial annihilator in $\mathcal{L}(\mathcal{P})$. If there is a $q^* \in \mathcal{L}(\mathcal{P})$ such that*

$$p \cdot q^* = \Phi'_+(p, q) \tag{2}$$

for all $p \in \mathcal{P}^+$, then q^* is the unique $\mathcal{P}$-integrable subgradient of Φ at q.

In the above result, the condition that $\mathcal{O}(q)$ has a trivial annihilator in $\mathcal{L}(\mathcal{P})$ can be interpreted to say that the set of directions at which $q \in \mathcal{P}^+$ is boundary to the cone $\mathcal{P}^+$ is negligible. The latter condition represents an algebraic analogue to the property of q being a *quasi-interior point* of $\mathcal{P}^+$, which is better known in its topological forms presented in Fullerton and Braunschweiger (1963); Borwein and Lewis (1992). The collection of all quasi-interior points of $\mathcal{P}^+$ is the *quasi-interior* of $\mathcal{P}^+$. As an illustration, in the next section we define Shannon and Hyvärinen entropies on positive cones with nonempty quasi-interiors.

Presently, however, we do not investigate the proposed variant of quasi-interior in full. This analysis is not necessary for the application of Theorem 3.2 and may be a subject of future work. Notice also that uniqueness of subgradient is understood and valid only within the class $\mathcal{L}(\mathcal{P})$.

We now consider the case of topological subgradients. Our main assumption is the following:

$$\begin{cases} \mathcal{P}^+ \subset \mathcal{C}, \text{ where } \mathcal{C} \text{ is an open convex cone in a normed space } N \\ \Phi : \mathcal{C} \to \mathbb{R} \text{ is a continuous sublinear function.} \end{cases} \tag{3}$$

Theorem 3.3 *If* (3) *holds, then* Φ *admits a subgradient* $S : \mathcal{C} \to N^*$.

The result is generally known as the *supporting hyperplane theorem*. For proof see e.g. Niculescu and Persson (2006); Borwein and Vanderwerff (2010); Zalinescu (2002); Rudin (1973). Any subgradient $S : \mathcal{C} \to N^*$ of Φ may be identified with a proper scoring rule on $\mathcal{P}^+$ by restricting S to $\mathcal{P}^+$.

Theorem 3.4 *Assume* (3). *Then,* Φ *is Gâteaux differentiable on* $\mathcal{C}$ *if and only if* Φ *admits a unique subgradient* $S : \mathcal{C} \to N^*$. *In this case* $S = \nabla\Phi$ *is the Gâteaux derivative of* Φ.

This is a standard result in convex analysis. See e.g. Borwein and Vanderwerff (2010); Zalinescu (2002). See Example B.2 for an illustration of the case where the assumption $N^* \subset \mathcal{L}(\mathcal{P})$ is not satisfied.

4. Applications

In this section, we apply our main results to three important entropies: *Shannon entropy*, *Hyvärinen entropy*, and *quadratic entropy*. For each entropy, we investigate an appropriate domain with nonempty quasi-interior for which we show the existence of a unique subgradient.

4.1 Shannon Entropy

The *Shannon entropy function* for densities on $\mathbb{R}^d$ is given by

$$\Phi(p) = \int_{\mathbb{R}^d} p(x) \ln \frac{p(x)}{p \cdot 1} dx \tag{4}$$

where $p(x) \geq 0$ is assumed to be sufficiently regular. More facts about Shannon entropy may be found e.g. in Dawid (2007); Parry et al. (2012); Dawid et al. (2012).

We first show that Shannon entropy may only be defined for nonnegative functions in a natural way. The kernel of Φ is the function $\phi(t) = t \ln t$ for $t > 0$ and $\phi(0) = 0$. Clearly, $\phi(t)$ is strictly convex on $t \geq 0$ since, for $t > 0$, $\phi''(t) = 1/t > 0$, and ϕ is continuous at the endpoint $t = 0$. Notice that $\phi(t)$ has a vertical tangent at $t = 0$ since $\phi'(t) = \ln t + 1$. We conclude that $\phi(t)$ cannot be extended as a convex function to $t < 0$. This furnishes our claim.

The positive cone of $L^1(\mathbb{R}^d)$ comprises of all nonnegative functions in $L^1(\mathbb{R}^d)$ and is denoted by $L^1_+(\mathbb{R}^d)$. In Proposition B.1 we give a direct proof that $L^1_+(\mathbb{R}^d)$ is a nowhere

dense subset of $L^1(\mathbb{R}^d)$. Since the domain of Shannon entropy is a subset of $L^1_+(\mathbb{R}^d)$, it too is a nowhere dense set.

We now proceed to find a suitable prediction set. For $a \geq d+1$, we set

$$\mathcal{P}^+ = \left\{ p \in C(\mathbb{R}) \,\middle|\, p(x) > 0\,, \exists C_1, C_2 > 0 \,:\, \frac{C_1}{(1+|x|)^a} \leq p(x) \leq \frac{C_2}{(1+|x|)^{d+1}} \right\}.$$

Notice that $\mathcal{L}(\mathcal{P}) \subset L^1_{\mathrm{loc}}(\mathbb{R}^d)$. Indeed, for any $f \in \mathcal{L}(\mathcal{P})$ consider

$$p_t(x) = \begin{cases} 1 & 0 < |x| < t \\ \left(\frac{1+t}{1+|x|}\right)^{d+1} & t \leq |x|\,. \end{cases}$$

Since $p_t \in \mathcal{P}^+$, the $\mathcal{P}$-integrability of f implies that

$$\int_{|x|\leq t} |f(x)|\, dx < \infty$$

for all $t > 0$.

Let us next see that for any $q \in \mathcal{P}^+$, $\mathcal{O}(q)$ has a trivial annihilator in $\mathcal{L}(\mathcal{P})$. Clearly, $\mathcal{O}(q)$ contains all $p \in \operatorname{span}\mathcal{P}$ that have faster or equal decay at infinity compared to q. Suppose that $f \in \mathcal{O}(q)^\perp$. Choosing an appropriate approximation of the identity, $\{p_n\}$, $p_n \in \mathcal{O}(q)$, we get that $f * p_n(x) \to f(x)$ for every x in the Lebesgue set of f. Hence $f = 0$ a.e. on $\mathbb{R}^d$. We conclude that $\mathcal{O}(q)^\perp = \{0\}$.

After this preparation, we may now define Φ rigorously as the map from $\mathcal{P}^+$ to $\mathbb{R}$ given by (4). Strict convexity of Φ follows from the strict convexity of $t \ln t$, for $t \geq 0$, while its 1-homogeneity is trivial. Therefore, Φ is strictly sublinear on $\mathcal{P}^+$. Let us compute the right directional derivative of Φ.

For $q \in \mathcal{P}^+$ and $p \in \mathcal{D}(q)$, we set $q_t = q + tp$. We have

$$\begin{aligned} \lim_{t\to 0^+} \frac{\Phi(q+tp) - \Phi(q)}{t} &= \left.\frac{d}{dt}\right|_{t=0} \left(q_t \cdot \ln \frac{q_t}{q_t \cdot 1} \right) \\ &= p \cdot \ln \frac{q}{q\cdot 1} + q \cdot \left(\frac{p}{q} - \frac{p\cdot 1}{q \cdot 1} \right) \\ &= p \cdot \ln \frac{q}{q \cdot 1}. \end{aligned}$$

Therefore,

$$\Phi'_+(p,q) = \int_{\mathbb{R}^d} p(x) \ln \frac{q(x)}{q\cdot 1} dx.$$

Clearly, the function

$$S(q)(x) = \ln \frac{q(x)}{q \cdot 1}$$

is in $\mathcal{L}(\mathcal{P})$. Indeed, the claim follows from the fact that $S(q)$ is continuous in x and grows logarithmically as $|x| \to \infty$. In view of Theorem 3.2, S is the unique $\mathcal{P}$-integrable subgradient of Φ on $\mathcal{P}^+$ since $\Phi'_+(p,q) = p \cdot S(q)$ for every $p, q \in \mathcal{P}^+$. The map is known as the *logarithmic scoring rule*.

The uniqueness of the logarithmic scoring rule as a subgradient of Shannon entropy is in no way an absolute fact. Using the Hahn-Banach theorem as illustrated in Example B.3 and the fact that $L^1_+(\mathbb{R}^d)$ consists entirely of boundary points, one may construct other subgradients of Φ that lie outside $\mathcal{L}(\mathcal{P})$. Moreover, if q lies on the *quasi-boundary* of $\mathcal{P}^+$ (i.e. the points where the condition $\mathcal{O}(q)^\perp = \{0\}$ is violated), then uniqueness will fail even within $\mathcal{L}(\mathcal{P})$.

4.2 Hyvärinen Entropy

Hyvärinen entropy for densities on $\mathbb{R}^d$ is defined as

$$\Phi(p) = \int_{\mathbb{R}^d} \frac{|\nabla p(x)|^2}{p(x)} dx. \tag{5}$$

Here ∇ is the gradient on $\mathbb{R}^d$. Hyvärinen and related entropies are considered e.g. in Parry et al. (2012); Ehm and Gneiting (2012); Forbes and Lauritzen (2014); Dawid and Musio (2012); Hyvärinen (2005, 2007); Sánchez-Moreno et al. (2012).

We first show that there is no natural way to extend Hyvärinen entropy to signed densities. For simplicity, we confine ourselves to the case $d = 1$. Suppose that p changes sign at some $x_0 \in \mathbb{R}$ that has multiplicity one. The assumption is generic and it means that x_0 is not an inflection point of p. It follows that the above integral is divergent at x_0. Indeed, the claim is a direct consequence of the asymptotic expansion of the term

$$\frac{|p'(x)|^2}{p(x)} = \frac{1}{x - x_0} + O(x - x_0)$$

near x_0. On the other hand, if p has a zero of higher multiplicity at x_0, one may check that the above asymptotics will be bounded and the integral will be convergent in a neighbourhood of x_0. Nevertheless, the example shows that Φ cannot be generally defined for densities that change sign.

We proceed to define a suitable domain for Φ. Suppose that $\mathcal{P}^+$ consists of all positive, twice continuously differentiable functions $p(x)$ on $\mathbb{R}^d$ that satisfy the bounds:

(a) there are $C_1 > 0$ and $k > 0$ such that

$$\left|\frac{\nabla p(x)}{p(x)}\right| + \left|\frac{\Delta p(x)}{p(x)}\right| \leq C_1(1 + |x|)^k;$$

(b) there is $C_2 > 0$ such that

$$|p(x)| \leq \frac{C_2}{(1 + |x|)^{d+1+k^2}},$$

where $\Delta = \partial^2/\partial x_1^2 + \cdots + \partial^2/\partial x_d^2$ is the *Laplacian* on $\mathbb{R}^d$. In view of the above, we have the following limit

$$\lim_{R\to\infty} \frac{1}{R} \int_{|y|=R} \left(\frac{y\nabla q(y)}{q(y)}\right) p(y) dy = 0 \tag{6}$$

for any $p, q \in \mathcal{P}^+$. Note that here

$$y\nabla q(y) = y_1 \frac{\partial q(y)}{\partial y_1} + \cdots + \frac{y_d \partial q(y)}{\partial y_d}$$

denotes the *scalar product* of y and $\nabla q(y)$ and the integral in (6) is a surface integral over the sphere centred at the origin of radius R. The class $\mathcal{P}$ is broad, e.g. it contains the Gaussians, and all positive continuous densities that have bounded first and second-order derivatives and decay at infinity sufficiently fast. Just like in Section 4.1, we have that $\mathcal{L}(\mathcal{P}) \subset L^1_{\mathrm{loc}}(\mathbb{R}^d)$ and that for any $q \in \mathcal{P}^+$ the annihilator of $\mathcal{O}(q)$ in $\mathcal{L}(\mathcal{P})$ is trivial. In the light of Proposition B.1, $\mathcal{P}^+$ is nowhere dense in $L^1(\mathbb{R}^d)$ as $\mathcal{P}^+ \subset L^1_+(\mathbb{R}^d)$.

We now formally define Hyvärinen entropy as the map from $\mathcal{P}^+$ to $\mathbb{R}$ given in (5). Convexity of Φ follows from the convexity of the function

$$\phi(t, t_1, \ldots, d_d) = \frac{t_1^2 + \cdots + t_d^2}{t}, \qquad \text{for } t > 0,\ (t_1, \ldots, t_d) \in \mathbb{R}^d,$$

while its 1-homogeneity is trivial. Hence, Φ is sublinear. Let us compute its right directional derivative.

For $q \in \mathcal{P}^+$ and $p \in \mathcal{D}(q)$, we set $q_t = q + tp$. We have

$$\begin{aligned}\lim_{t \to 0^+} \frac{\Phi(q+tp) - \Phi(q)}{t} &= \int_{\mathbb{R}^d} \left.\frac{d}{dt}\right|_{t=0} \left(\frac{|\nabla q_t(x)|^2}{q_t(x)} \right) dx \\ &= \int_{\mathbb{R}^d} \left(2\frac{\nabla q(x)}{q(x)} \frac{\nabla p(x)}{p(x)} - \frac{|\nabla q(x)|^2}{q^2(x)} \right) p(x) dx.\end{aligned}$$

By integration by parts we get

$$\begin{aligned}&\int_{|x| \le R} \left(\frac{2\nabla q(x) \nabla p(x)}{q(x)} - \frac{|\nabla q(x)|^2}{q^2(x)} p(x) \right) dx \\ &\qquad = \int_{|x| \le R} \left(-\frac{2\Delta q(x)}{q(x)} + \frac{|\nabla q(x)|^2}{q^2(x)} \right) p(x) dx + \frac{2}{R} \int_{|y|=R} \left(\frac{y \nabla q(y)}{q(y)} \right) p(y) dy.\end{aligned}$$

Letting $R \to \infty$ and using (6), we obtain

$$\Phi'_+(p, q) = \int_{\mathbb{R}^d} \left(-\frac{2\Delta q(x)}{q(x)} + \frac{|\nabla q(x)|^2}{q^2(x)} \right) p(x) dx.$$

The assumptions on $\mathcal{P}^+$ guarantee that

$$S(q)(x) = -\frac{2\Delta q(x)}{q(x)} + \frac{|\nabla q(x)|^2}{q^2(x)}$$

is $\mathcal{P}$-integrable for every $q \in \mathcal{P}^+$. In view of Theorem 3.2, $S(q)$ is the unique $\mathcal{P}$-integrable subgradient of Φ on $\mathcal{P}^+$. The map is known as the *Hÿvarinen scoring rule* (Parry et al., 2012).

In fact, $S(q)$ is a strict subgradient of Φ on $\mathcal{P}^+$. This can be shown if we notice that the divergence induced by S has the representation

$$p \cdot S(p) - p \cdot S(q) = \int_{\mathbb{R}^d} \left| \frac{\nabla p(x)}{p(x)} - \frac{\nabla q(x)}{q(x)} \right|^2 p(x) dx.$$

The latter identity can be proved by integration by parts. The divergence is zero if and only if

$$\nabla(\ln p(x) - \ln q(x)) = 0.$$

This is equivalent to $p = Cq$ for some constant $C > 0$, i.e., p and q being positively collinear. This concludes the proof of the claim.

4.3 Quadratic Entropy

Here we consider the *quadratic entropy*

$$\Phi(q) = \frac{1}{q \cdot 1} \int_\Omega q^2(x) dx, \tag{7}$$

where $(\Omega, \mathcal{A}, \mu)$ is a Lebesgue measure space with $\Omega \subset \mathbb{R}^d$. In what follows, we show that its Gâteaux derivative is the *quadratic scoring rule*, also known as *Brier score* (Brier, 1950). The quadratic entropy is a member of the important family of *power entropy functions*. The corresponding *power scoring rules* have been studied in connection to robust estimation e.g. in Basu et al. (1998); Kanamori and Fujisawa (2015, 2014).

We proceed to choose a suitable domain for Φ. In contrast to the previous two entropies we now introduce a topology. To that end, we begin with a description of some normed spaces. Let $w : \Omega \to [0, \infty)$ be a measurable function which we call a *weight*. By $L^p(\Omega, w)$, for $p \geq 1$, we denote the Lebesgue space of functions on Ω whose p-th power is absolutely integrable with respect to the weight $w(x)$. By $\|\cdot\|_{p,w}$ we denote the corresponding weighted L^p-norm. When w is identically equal to one we get the usual Lebesgue space and norm. In this case we drop w from our notation. We now set

$$w(x) = (1 + |x|)^{d+1}.$$

Notice that $L^2(\Omega, w)$ embeds continuously in $L^1(\Omega)$. Indeed, for $f \in L^1(\Omega)$, we have

$$\begin{aligned} \int_\Omega |f(x)| \, dx &= \int_\Omega w^{-1/2}(x) \, |f(x)| \, w^{1/2}(x) dx \\ &\leq \left(\int_\Omega w^{-1}(x) dx \right)^{\frac{1}{2}} \left(\int_\Omega |f(x)|^2 \, w(x) dx \right)^{\frac{1}{2}} \\ &\leq C \, \|f\|_{2,w} \, , \end{aligned}$$

where $C > 0$ is a constant. Clearly, $L^2(\Omega, w)$ also embeds continuously in $L^2(\Omega)$ and hence the same conclusion holds for $L^2(\Omega, w)$ for all intermediate spaces $L^p(\Omega)$ with $1 \leq p \leq 2$. Hence, we have the inequality

$$\|f\|_p \leq C \, \|f\|_{2,w}$$

for some fixed $C > 0$ and all $p \in [1, 2]$.

We have that $f \in L^2(\Omega, w)$ if and only if $fw^{1/2} \in L^2(\Omega)$. Clearly, the weight is needed only when Ω is unbounded as otherwise the weighted and the ordinary L^p-norms are equivalent. The continuous dual space of $L^2(\Omega, w)$ may be identified with the space $L^2(\Omega, w^{-1})$. Therefore, $g \in L^2(\Omega, w^{-1})$ if and only if $gw^{-1/2} \in L^2(\Omega)$. Hence, the dual space $L^2(\Omega, w^{-1})$ contains the constants and also the elements of $L^2(\Omega, w)$.

We now specify a prediction set $\mathcal{P} \subset L^2_+(\Omega, w)$ with the following property: there are constants $k_1 > 0$ and $k_2 > 0$ such that

$$k_1 \leq \|q\|_{2,w} \leq k_2$$

for all $q \in \mathcal{P}$. Choose $0 < \epsilon < \min(1, k_1)$. For $p \in L^2(\Omega)$, let $B_\rho(p)$ denote the open ball about p of radius $\rho > 0$. Choose $\delta > 0$ so small that for every $p \in B_\delta(0)$ we have $\|p\|_1 \leq \epsilon$ and $\|p\|_{2,w} \leq \epsilon$. Let $q \in \mathcal{P}$ and consider $r \in B_\delta(q)$. It is easy to show that

$$k_1 - \epsilon \leq \|r\|_{2,w} \leq k_2 + \epsilon$$

for all $r \in B_\delta(q)$. Similarly, we also have

$$1 - \epsilon \leq r \cdot 1 \leq 1 + \epsilon$$

for all $r \in B_\delta(q)$. Here we have used the fact that $r = p + q$, where $q \cdot 1 = 1$ and $\|p\|_1 \leq \epsilon$. We now set

$$\mathcal{C}_0 = \mathcal{P} + B_\delta(0) = \cup_{q \in \mathcal{P}} B_\delta(q).$$

It follows that $\mathcal{C}_0$ is convex as both $\mathcal{P}$ and $B_\delta(0)$ are convex. Finally, let $\mathcal{C} = \mathcal{C}_0^+$ be the cone of $\mathcal{C}_0$. Clearly, $\mathcal{C}$ is an open convex cone in $L^2(\Omega, w)$.

We may now formally define Φ as the map from $\mathcal{C}$ to $\mathbb{R}$ given by (7). We have that Φ is strictly convex on $\mathcal{C}_0$ as the kernel function $\phi(t) = t^2$ is strictly convex for $t \in \mathbb{R}$. Therefore, Φ is strictly sublinear on $\mathcal{C}$. It is not hard to see that Φ is also continuous on $\mathcal{C}$. Theorem 3.3 implies that Φ has a subgradient on $\mathcal{C}$. The following computation shows that Φ is Gâteaux differentiable. Indeed, for $q \in \mathcal{C}$ and $p \in L^2(\Omega, w)$, we have

$$\begin{aligned}\lim_{t \to 0} \frac{\Phi(q + tp) - \Phi(q)}{t} &= \int_\Omega \frac{d}{dt}\bigg|_{t=0} \frac{(q(x) + tp(x))^2}{(q + tp) \cdot 1} dx \\ &= 2 \int_\Omega \frac{q(x)p(x)}{q \cdot 1} dx - \int_\Omega \frac{q^2(x)}{(q \cdot 1)^2} dx \int_\Omega p(x) dx.\end{aligned}$$

We obtain that

$$\nabla\Phi(q) = \frac{2q}{q \cdot 1} - \frac{q \cdot q}{(q \cdot 1)^2}$$

is the Gâteaux derivative of Φ as clearly $\nabla\Phi(q) \in L^2(\Omega, w^{-1})$. In view of Theorem 3.4, $S = \nabla\Phi|_{\mathcal{P}^+}$ defines a strictly proper scoring rule on $\mathcal{P}^+$. We have that $\nabla\Phi$ is the unique subgradient of quadratic entropy on the cone $\mathcal{C}$, but as discussed before, by using the Hahn-Banach theorem one may show that uniqueness fails on $\mathcal{P}^+$ when Ω is unbounded. The rule S is known as the *quadratic scoring rule*.

5. Conclusion

We were originally motivated to understand the implications of the fact that Shannon and Hyvärinen entropies are only finite on domains with empty interiors. As no notion of functional derivative is applicable to these entropies, the question whether the logarithmic and Hyvärinen scoring rules are the unique subgradients of their respective entropy functions is not obvious. In contrast, the quadratic entropy may be continuously extended to signed densities, which allows us to interpret the quadratic scoring rule as the Gâteaux derivative of its entropy. We realised that in order to answer the titular question of the paper, one must introduce additional structures to the basic measure-theoretic framework known in the literature of scoring rules (Hendrickson and Buehler, 1971). The most important new aspect is the notion of interior and its refinement (known as quasi-interior) in the context of domains with empty interior. Another crucially important idea is to use directional derivatives to describe the subdifferentials of entropy functions. Finally, our approach marks a shift in emphasis from proper scoring rules to a greater focus on entropy functions.

Acknowledgments

Much of this work was done while the author was a PostDoc at the University of Heidelberg, Germany. The author has been supported by the European Union Seventh Framework Programme under grant agreement no. 290976.

Appendix A. Proofs

Lemma A.1 *Let $\mathcal{P}$ be a prediction set and $\Phi : \mathcal{P}^+ \to \mathbb{R}$ be a 1-homogeneous function. If Φ has a (strict) subgradient on $\mathcal{P}^+$, then Φ is a (strictly) sublinear function.*

Proof Let $S : \mathcal{P}^+ \to \operatorname{Lin}\mathcal{P}$ be a (strict) subgradient of Φ. Then S (strictly) satisfies

$$\Phi(p) \geq p \cdot S((1-\lambda)p + \lambda q)$$
$$\Phi(q) \geq q \cdot S((1-\lambda)p + \lambda q)$$

for every $p, q \in \mathcal{P}^+$ (p and q not positively collinear), and every $0 < \lambda < 1$. Multiplying the first inequality by $1-\lambda$, the second one by λ, and then adding them up, we obtain that Φ (strictly) satisfies

$$\Phi(1-\lambda)p + \lambda q) \leq (1-\lambda)\Phi(p) + \lambda\Phi(q).$$

■

Proof [of Lemma 2.1] We first show that $\operatorname{cone}(\mathcal{P}^+ - q) \subset \mathcal{D}(q)$. It is easy to see that $\mathcal{D}(q)$ is closed under taking conic combinations. The claim follows from the fact that $(\mathcal{P}^+ - q) \subset \mathcal{D}(q)$. We now show that $\mathcal{D}(q) \subset \operatorname{cone}(\mathcal{P}^+ - q)$. If $p \in \mathcal{D}(q)$, then there is $\epsilon_p > 0$ and $r \in \mathcal{P}^+$ such that $q + \epsilon_p p = r$. Then $p = (r-q)\epsilon_p^{-1}$ and hence $p \in \operatorname{cone}(\mathcal{P}^+ - q)$. ■

Proof [of Proposition 2.3] (a) For $p \in \mathcal{D}(q)$ arbitrary, consider the line in $\operatorname{span}\mathcal{P}$ with parametric equation

$$\gamma(t) = q + t(p - q), \quad t \in \mathbb{R},$$

passing through q and p. Clearly, $\gamma(0) = q$ and $\gamma(1) = p$. Moreover, there is some $\epsilon > 0$ such that the interval $[0, \epsilon]$ is mapped entirely in $\mathcal{P}^+$ under γ (if $p \in \mathcal{P}^+$, then $\epsilon \geq 1$). Then the function

$$\phi(t) = \Phi(q + t(p - q)), \quad t \in [0, \epsilon],$$

is convex and its slope function

$$s_\phi(t_1, t_2) = \frac{\phi(t_2) - \phi(t_1)}{t_2 - t_1}, \quad t_1, t_2 \in [0, \epsilon],$$

is nondecreasing (Rockafellar, 1972; Niculescu and Persson, 2006). We have that

$$\Phi'_+(p, q) = \lim_{t_2 \to 0+} \frac{\phi(t_2) - \phi(0)}{t_2} = \inf_{t_2 > 0} \frac{\phi(t_2) - \phi(0)}{t_2}.$$

If $p \in \mathcal{O}(q)$, then there is some $\delta > 0$ such that the interval $[-\delta, \delta]$ is mapped entirely in $\mathcal{P}^+$ under γ. Let $-\delta \leq t_1 < 0 < t_2 \leq \delta$. To prove that $\Phi'_+(p, q)$ is finite, we consider

$$\frac{\phi(0) - \phi(t_1)}{-t_1} \leq \frac{\phi(t_2) - \phi(0)}{t_2},$$

and take the infimum in t_2.

(b) Homogeneity of $\Phi'_+(\cdot, q)$ follows from:

$$\begin{aligned}\Phi_+(\lambda p, q) = \lim_{\tau \to 0^+} \frac{\Phi(q + \tau\lambda p) - \Phi(q)}{\tau} &\leq \lambda \lim_{\tau \to 0^+} \frac{\Phi(q + \lambda\tau p) - \Phi(q)}{\lambda\tau} \\ &= \lambda\Phi_+(p, q).\end{aligned}$$

Let $p_1, p_2 \in \mathcal{D}(q)$. Subadditivity of $\Phi'_+(\cdot, q)$ follows from:

$$\begin{aligned}\Phi'_+(p_1 + p_2, q) &= \lim_{\tau \to 0^+} \frac{\Phi(q + \tau(p_1 + p_2)) - \Phi(q)}{\tau} \\ &\leq \lim_{\tau \to 0^+} \frac{\Phi(q/2 + \tau p_1) - \Phi(q)/2}{\tau} + \lim_{\tau \to 0^+} \frac{\Phi(q/2 + \tau p_2) - \Phi(q)/2}{\tau} \\ &= \lim_{\tau \to 0^+} \frac{\Phi(q + 2\tau p_1) - \Phi(q)}{2\tau} + \lim_{\tau \to 0^+} \frac{\Phi(q + 2\tau p_2) - \Phi(q)}{2\tau} \\ &= \Phi'_+(p_1, q) + \Phi'_+(p_2, q).\end{aligned}$$

(c) The claim follows from

$$\begin{aligned}\Phi'_+(p, \lambda q) = \lim_{\tau \to 0^+} \frac{\Phi(\lambda q + \tau p) - \Phi(\lambda q)}{\tau} &= \lim_{\tau \to 0^+} \frac{\Phi(q + \tau p/\lambda) - \Phi(q)}{\tau/\lambda} \\ &= \Phi'_+(p, q).\end{aligned}$$

(d) We have

$$\Phi(p) \geq \Phi(q+p) - \Phi(q) \geq \frac{\Phi(q+\tau p) - \Phi(q)}{\tau} \geq \Phi'_+(p,q),$$

where $0 < \tau < 1$. The first inequality follows from sublinearity of Φ, while the second and third follow from the fact that the slope function of Φ is nondecreasing. It remains to show that $\Phi(q) = \Phi'_+(q,q)$. This follows immediately from

$$\begin{aligned}\Phi(q) &= \lim_{\tau \to 0^+} \frac{(1+\tau)\Phi(q) - \Phi(q)}{\tau} = \lim_{\tau \to 0^+} \frac{\Phi(q+\tau q) - \Phi(q)}{\tau} \\ &= \Phi'_+(q,q).\end{aligned}$$

(e) The claim is a direct consequence of

$$0 = \Phi'_+(0,q) = \Phi'_+(p-p,q) \leq \Phi'_+(p,q) + \Phi'_+(-p,q).$$

(f) To show that $\mathcal{O}'(q)$ is a linear subspace of $\mathcal{O}(q)$ it is enough to show that it is closed under scalar multiplication and vector addition. Let $\lambda \in \mathbb{R}$ and $p \in \mathcal{O}'(q)$. Then, for $\lambda \geq 0$, $\Phi'_+(\lambda p, q) = \lambda \Phi'_+(p,q)$. Analogously, for $\lambda < 0$ we have

$$\Phi'_+(\lambda p, q) = \Phi'_+(-\lambda(-p), q) = -\lambda \Phi'_+(-p,q) = \lambda(-\Phi'_+(-p,q)) = \lambda \Phi'_+(p,q).$$

Therefore, $\Phi'_+(\lambda p, q) = \lambda \Phi'_+(p,q)$ for any $\lambda \in \mathbb{R}$ and $p \in \mathcal{O}'(q)$. Then multiplying by λ both sides of the identity

$$-\Phi'_+(-p,q) = \Phi'_+(p,q)$$

and using the previous identity, we get that $\lambda p \in \mathcal{O}'(q)$. Hence, $\mathcal{O}'(q)$ is closed under scalar multiplication.

Suppose now that $p, r \in \mathcal{O}'(q)$. We have

$$\begin{aligned}\Phi'_+(p+r,q) &\leq \Phi'_+(p,q) + \Phi'_+(r,q) = -(\Phi'_+(-p,q) + \Phi'_+(-r,q)) \\ &\leq -\Phi'_+(-p-r,q) \leq \Phi'_+(p+r,q),\end{aligned}$$

where the last inequality follows from (e). Clearly, we must have equalities throughout. In particular,

$$-\Phi'_+(-p-r,q) = \Phi'_+(p+r,q)$$

and

$$\Phi'_+(p+r,q) = \Phi'_+(p,q) + \Phi'_+(r,q).$$

Hence $p+r \in \mathcal{O}'(q)$. We conclude that $\mathcal{O}'(q)$ is a linear subspace and $\Phi'_+(\cdot,q)\big|_{\mathcal{O}'(q)}$ is linear. ■

Proof [of Proposition 2.4] (a) The sufficient part of the claim follows from Proposition 2.3 (d). Let us now show the necessary part. To that end, let $q^* \in \operatorname{Lin} \mathcal{P}$ be a subgradient of Φ at q, and let $p \in \mathcal{P}^+$ be arbitrary. Setting $q_t = q + (1-t)p$, we have $\Phi(q_t) \geq q_t \cdot q^*$ for

all $t \in [0,1]$. Subtracting $\Phi(q)$ from both sides of the inequality and dividing by $(1-t)$, for $t \in (0,1)$, we get

$$\frac{\Phi(q+(1-t)p) - \Phi(q)}{1-t} \geq p \cdot q^*.$$

Letting $t \uparrow 1$, we get

$$\Phi'_+(p,q) \geq p \cdot q^*$$

as desired.

(b) The claim follows by restricting Φ to 1-dimensional affine spaces through q. On these spaces Φ is convex and differentiable and therefore has a unique subgradient. Since these subspaces cover the whole of $\operatorname{span} \mathcal{P}$, it follows that the directional derivative $\Phi'(\cdot, q)$ is the unique subgradient of Φ there.

(c) In view of Proposition 2.3 (a), $\Phi'_+(p,q)$ is finite for each $p \in \mathcal{O}(q) = \operatorname{span} \mathcal{P}$. The hypothesis implies that there is at least one 1-dimensional linear subspace of $\operatorname{span} \mathcal{P}$ on which $\Phi'_+(\cdot, q)$ is not linear. There are infinitely many ways we can choose a linear function on that space that is dominated by $\Phi'_+(\cdot, q)$. The claim now follows from the Hahn-Banach theorem stated below as Theorem B.4.

(d) Since $\mathcal{O}(q) \neq \operatorname{span} \mathcal{P}$, it follows that $\mathcal{P}^+ \setminus \mathcal{O}(q)$ is nonempty. Take any p in that set and consider the 1-dimensional linear space generated by the span of p. Since $\Phi'_+(\cdot, q)$ is defined only on its positive half-space, there are infinitely many linear functions that are dominated by $\Phi'_+(\cdot, q)$ on the whole space. The proof now follows from Theorem B.4.

(e) There is no element of $\operatorname{Lin} \mathcal{P}$ that satisfies the condition in part (a) of this proposition. Therefore, $\partial\Phi(q) = \emptyset$. ■

Proof [of Theorem 3.1] Suppose that $q^* \in \mathcal{L}(\mathcal{P})$ satisfies $p \cdot q^* \leq \Phi'_+(p,q)$ for all $p \in \mathcal{P}^+$, with equality for $p = q$. In view of Proposition 2.3 (d), we have that $p \cdot q^* \leq \Phi(p)$ for all $p \in \mathcal{P}^+$, and $q \cdot q^* = \Phi(q)$. Hence, q^* is a $\mathcal{P}$-integrable subgradient of Φ at q.

The converse claim, that is, if q^* is a $\mathcal{P}$-integrable subgradient of Φ at q, then $p \cdot q^* \leq \Phi'_+(p,q)$ for all $p \in \mathcal{P}^+$, with equality for $p = q$, follows from Proposition 2.4 (a). ■

Proof [of Theorem 3.2] The hypothesis implies that $\Phi'_+(\cdot, q)$ is linear on $\mathcal{O}(q) \subset \mathcal{P}^+$. By restricting Φ to 1-dimensional subspaces of $\mathcal{O}(q)$ it follows immediately that any subgradient of Φ must agree with q^* on $\mathcal{O}(q)$. The assumption that $\mathcal{O}(q)^\perp = \{0\}$ implies that Φ may have at most one $\mathcal{P}$-integrable subgradient at q. Then the claim follows from the fact that q^* is a subgradient of Φ at q. ■

Appendix B. Some Additional Facts

The positive cones in many standard function spaces are nowhere dense sets. Let us show this for the Lebesgue space $L^1(\mathbb{R}^d)$. The positive cone of $L^1(\mathbb{R}^d)$ consists of all Lebesgue integrable functions $f \geq 0$ a.e. on $\mathbb{R}^d$ and is denoted by $L^1_+(\mathbb{R}^d)$. We recall that a set in a topological vector space is *nowhere dense* if its closure has empty interior.

Proposition B.1 *The positive cone of $L^1(\mathbb{R}^d)$ is nowhere dense.*

Proof We show that for every $f \geq 0$ a.e., there is $g \geq 0$ a.e. such that, for every $\alpha > 0$, $f - \alpha g \notin L^1_+(\Omega)$. This means that no open ball about f is contained in $L^1_+(\mathbb{R}^d)$. Since $L^1_+(\mathbb{R}^d)$ is closed, then this would imply that $L^1_+(\mathbb{R}^d)$ is nowhere dense.

To prove our claim, we use the fact that there is no absolutely convergent series with a slowest rate of decay at infinity. We begin by partitioning $\mathbb{R}^d$ into dyadic regions

$$\omega_k = \{2^k \leq |x| < 2^{k+1}\}$$

for $k \in \mathbb{Z}$. For $f \in L^1(\mathbb{R}^d)$, we set

$$a_k = \int_{\omega_k} f(x)dx.$$

We have that the series

$$\sum_{k=0}^{\infty} a_k = \int_{\mathbb{R}^d} f(x)dx$$

is absolutely convergent. If $r_k = \sum_{i \geq k} a_i$ is the tail of the series for each k, then the series $\sum_{k \geq 0} a_k/\sqrt{r_k}$ is also convergent (Rudin, 1976). Notice that the ratio of the common term of the second to the first series tends to infinity as $k \to \infty$. Therefore, the second series has a strictly slower rate of convergence. There exists a function $g \in L^1_+(\mathbb{R}^d)$ such that the integrals of g on ω_k are $b_k = a_k/\sqrt{r_k}$ and

$$\sum_{k=0}^{\infty} b_k = \int_{\mathbb{R}^d} g(x)dx.$$

Clearly, for any $\alpha > 0$, the difference $f - \alpha g$ changes sign for some ω_k, and hence $f - \alpha g \notin L^1_+(\mathbb{R}^d)$. ■

The next example illustrates the notion of topological subgradient in the case when the assumption $N^* \subset \mathcal{L}(\mathcal{P})$ is not satisfied.

Example B.2 Consider a Lebesgue measure space $(\Omega, \mathcal{A}, \mu)$ with Ω a compact subset of $\mathbb{R}^d$. We set $\mathcal{P}^+$ to be the positive cone of $C(\Omega)$, that is, the set of all nonnegative continuous functions on Ω. The continuous dual of $C(\Omega)$ is the space of all real-valued Radon measures on Ω. The fact that $\mathcal{P}^+$ contains constants implies that $\mathcal{L}(\mathcal{P}) \subseteq L^1(\Omega)$. Actually, $\mathcal{L}(\mathcal{P}) = L^1(\Omega)$ and hence the $\mathcal{P}$-integrable functions are the Radon measures that have a Lebesgue density. Since $L^1(\Omega) \subsetneq (C(\Omega))^*$, we see that in this case the notion of a $\mathcal{P}$-integrable subgradient is more restrictive than that of a topological subgradient.

We proceed to examine the implications of the latter observation on a concrete sublinear function. Let $\Phi : C(\Omega) \to \mathbb{R}$ be the *supremum function*, that is,

$$\Phi(p) = \sup_{x \in \Omega} p(x).$$

It is easy to check that Φ is non-strictly sublinear and continuous. The supporting hyperplane theorem guarantees the existence of a topological subgradient of Φ at each point in its

domain that is a *real Radon measure*. Let us see whether the subgradient is regular enough to be a proper scoring rule.

We first demonstrate that there are points $q \in \mathcal{P}^+$ at which Φ has no subgradient in $\mathcal{L}(\mathcal{P})$. To that end, let $\mathcal{M}(q)$ denote the set of *modes* of q, that is, the subset of Ω where q reaches its maximum. Notice that $\mathcal{M}(q)$ is always compact. It can be shown that

$$\Phi'_+(p, q) = \sup_{x \in \mathcal{M}(q)} p(x),$$

the proof of which is left to the reader. When $\mathcal{M}(q) = \{x_0\}$ is a singleton, $\Phi'_+(\cdot, q) = \delta(x - x_0)$ is Dirac's delta function. Clearly, in this case Φ is Gâteaux differentiable with derivative $\delta(x - x_0)$. We claim that Φ has no $\mathcal{P}$-integrable subgradient for any density q with $\mu(\mathcal{M}(q)) = 0$.

Suppose conversely that $q^* \in \mathcal{L}(\mathcal{P})$, $q^* \neq 0$, is a subgradient of Φ at q. Then

$$\Phi'_+(p, q) \geq p \cdot q^*$$

for all $p \in \mathcal{P}^+$. We shall show that this inequality implies $q^*(x) \leq 0$ a.e. on Ω, which leads to a contradiction with $\Phi(q) = q \cdot q^* > 0$.

To show the latter claim, notice that $\Omega \setminus \mathcal{M}(q)$ is open, and hence for any $y \in \Omega \setminus \mathcal{M}(q)$, there is $\epsilon_y > 0$ such that the ball about y of radius ϵ_y lies in the complement of $\mathcal{M}(q)$ with respect to Ω. Let $\{p_k\}$ be a sequence of densities approximating $\delta(x - y)$ entirely supported on this ball. Since $\Phi'_+(p_k, q) = 0$, we get that $p_k \cdot q^* \leq 0$. If y is a Lebesgue point of q^*, then we have the limit

$$\lim_{k \to \infty} p_k \cdot q^* = \delta(\cdot - y) \cdot q^* = q^*(y).$$

Since almost every point of q^* is a Lebesgue point, we get that $q^*(x) \leq 0$ a.e. on Ω. This completes the proof of the claim.

In the case $\mu(\mathcal{M}(q)) > 0$, we may find a $\mathcal{P}$-integrable subgradient of Φ at q. Consider the function

$$q^*(x) = \begin{cases} \frac{1}{\mu(\mathcal{M}(q))} & x \in \mathcal{M}(q) \\ 0 & x \in \Omega \setminus \mathcal{M}(q). \end{cases}$$

Clearly, $q \cdot q^* = \sup_{x \in \Omega} q(x)$ and $p \cdot q^* \leq \sup_{x \in \Omega} p(x)$ for all $p \in \mathcal{P}^+$. This furnishes our claim.

In our final example, we illustrate the fact that at boundary points a sublinear function has either no subgradient, or infinitely many.

Example B.3 Take $\Phi(x, y) = x + y$ on $\mathbb{R}^2_+ = \{(x, y) \,|\, x \geq 0,\, y \geq 0\}$. The graph of Φ is a part of a plane, so it is easy to see that Φ has infinitely many supporting planes at the boundaries of $\mathbb{R}^2_+$. Consider now

$$\Phi(x, y) = x \ln \frac{x}{x + y} + y \ln \frac{y}{x + y}$$

on $\mathbb{R}^2_+$, which is Shannon entropy for binary variables. A computation shows that

$$\nabla \Phi(x, y) = \ln \frac{x}{x + y} + \ln \frac{y}{x + y}$$

and hence $\nabla\Phi(x,y) \to -\infty$ when (x,y) tends to the boundary of $\mathbb{R}^2_+$. This means that Φ has vertical tangent planes through the coordinate axes, which implies that Φ has no subgradient on the boundary of its domain.

The situation is the same when $\mathcal{P}^+$ is a subset of an infinite dimensional vector space. For example, one may use the Hahn-Banach theorem presented below to show the existence of multiple supporting hyperplanes at boundary points q for which $\Phi'_+(p,q)$ is finite for all $p \in \mathcal{P}^+$. If, instead, there is $p \in \mathcal{P}^+$ for which $\Phi'_+(p,q) = -\infty$, then Φ has no subgradient at q.

We now state a slight generalisation of the classical Hahn-Banach theorem. Let E be a real vector space and $K \subset E$ be a convex cone.

Theorem B.4 (Hahn-Banach theorem) *Let $\phi : K \to \mathbb{R}$ be a sublinear function and $l_0 : E_0 \to \mathbb{R}$ be a linear functional on a linear subspace $E_0 \subseteq E$ which is dominated by ϕ on $E_0 \cap K$, i.e.*

$$l_0(q) \leq \phi(q), \qquad \forall q \in E_0 \cap K.$$

Then there exists a linear extension $l : E \to \mathbb{R}$ of l_0 to the whole space E such that

$$\begin{aligned} l(q) &= l_0(q), \qquad \forall q \in E_0, \\ l(q) &\leq \phi(q), \qquad \forall q \in E \cap K. \end{aligned}$$

In the classical formulation of the theorem, we have $K = E$. The proof of the version with $K \subset E$ is the same. In fact, if anything, the condition $K \subset E$ is easier to satisfy than $K = E$ when extending l_0.

References

A. Basu, I.R. Harris, N. L. Hjort, and M. C. Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

J. Borwein and J. Vanderwerff. *Convex Functions: Constructions, Characterizations and Counterexamples*. Number 109 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 2010.

J. M. Borwein and A. S. Lewis. Partially finite convex programming, Part I: Quasi relative interiors and duality theory. *Mathematical Programming*, 57(1–3):15–48, 1992.

G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950.

A. P. Dawid. The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*, 59:77–93, 2007.

A. P. Dawid and M. Musio. Estimation of spatial processes using local scoring rules. *AStA Advances in Statistical Analysis*, 96:1–7, 2012. Spatial special issue.

A. P. Dawid and M. Musio. Theory and applications of proper scoring rules. *Metron*, 72: 169–183, 2014.

A. P. Dawid, S. Lauritzen, and M. Parry. Proper local scoring rules on discrete sample spaces. *The Annals of Statistics*, 40(1):593–608, 2012.

W. Ehm and T. Gneiting. Local proper scoring rules of order two. *The Annals of Statistics*, 40(1):609–637, 2012.

P. G. M. Forbes and S. Lauritzen. Linear estimating equations for exponential families with application to Gaussian linear concentration models. *Linear Algebra and its Applications*, 2014. In press.

R. M. Frongillo and I. Kash. General truthfulness characterizations via convex analysis. In *Lecture Notes in Computer Science*, volume 8877 of *Web and Internet Economics*, pages 354–370. Springer, 2014.

R. E. Fullerton and C. C. Braunschweiger. Quasi-interior points of cones. *Technical Report 2, University of Delaware, Newark, Delaware*, 1963.

T. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102:359–376, 2007.

P. D. Grünwald and A. P. Dawid. Game theory, maximum entropy, minimum discrepancy, and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.

A. Hendrickson and R. Buehler. Proper scores for probability forecasters. *The Annals of Mathematical Statistics*, 42(6):1916–1921, 1971.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

A. Hyvärinen. Some extensions of score matching. *Computational Statistics & Data Analysis*, 51:2499–2512, 2007.

T. Kanamori and H. Fujisawa. Affine invariant divergences associated with proper composite scoring rules and their applications. *Bernoulli*, 20(4):2278–2304, 2014.

T. Kanamori and H. Fujisawa. Robust estimation under heavy contamination using enlarged models. *Biometrika*, 2015. doi:10.1093/biomet/asv014.

C. Niculescu and L.-E. Persson. *Convex Functions and Their Applications*. CMS Books in Mathematics. Springer, 2006.

M. Parry, A. P. Dawid, and S. Lauritzen. Proper local scoring rules. *The Annals of Statistics*, 40(1):561–592, 2012.

R. T. Rockafellar. *Convex Analysis*. Princeton Mathematical Series. Princeton University Press, second edition, 1972.

W. Rudin. *Functional Analysis*. McGraw-Hill, 1973.

W. Rudin. *Principles of Mathematical Analysis*. International Series in Pure and Applied Mathematics. McGraw-Hill, third edition, 1976.

P. Sánchez-Moreno, A. Zarzo, and J. S. Dehesa. Jensen divergence based on Fisher's information. *Journal of Physics A: Mathematical and Theoretical*, 45, 2012. 125305.

R. C. Williamson. The geometry of losses. *Conference On Learning Theory (COLT)*, 35: 1078–1108, 2014.

C. Zalinescu. *Convex Analysis in General Vector Spaces*. World Scientific, 2002.

Journal of Machine Learning Research 16 (2015) 2231-2271 Submitted 3/13; Revised 12/14; Published 8/15

Adaptive Strategy for Stratified Monte Carlo Sampling

Alexandra Carpentier A.CARPENTIER@STATSLAB.CAM.AC.UK
Statistical Laboratory
Center for Mathematical Sciences
Wilberforce Road
CB3 0WB Cambridge, United Kingdom

Remi Munos* MUNOS@GOOGLE.COM
Google DeepMind
London, UK

András Antos† ANTOS@CS.BME.HU
Budapest University of Technology and Economics
3 Műegyetem rkp.
1111 Budapest, Hungary

Editor: Nicolas Vayatis

Abstract

We consider the problem of stratified sampling for Monte Carlo integration of a random variable. We model this problem in a K-armed bandit, where the arms represent the K strata. The goal is to estimate the integral mean, that is a weighted average of the mean values of the arms. The learner is allowed to sample the variable n times, but it can decide on-line which stratum to sample next. We propose an UCB-type strategy that samples the arms according to an upper bound on their estimated standard deviations. We compare its performance to an ideal sample allocation that knows the standard deviations of the arms. For sub-Gaussian arm distributions, we provide bounds on the total regret: a distribution-dependent bound of order $\text{poly}(\lambda_{\min}^{-1})\widetilde{O}(n^{-3/2})$[1] that depends on a measure of the disparity $\lambda_{\min}$ of the per stratum variances and a distribution-free bound $\text{poly}(K)\widetilde{O}(n^{-7/6})$ that does not. We give similar, but somewhat sharper bounds on a proxy of the regret. The problem-independent bound for this proxy matches its recent minimax lower bound in terms of n up to a $\log n$ factor.

Keywords: adaptive sampling, bandit theory, stratified Monte Carlo, minimax strategies, active learning

1. Introduction

Estimation of mean values (or, especially, probabilities) can be considered as a special case of most problems in stochastic machine learning (e.g., regression function estimation, classification, clustering), thus understanding all of its aspects is crucial to tackle more

∗. Also affiliated to Inria Lille - Nord Europe, France

†. During parts of this work he was with the Computer and Automation Research Institute of the Hungarian Academy of Sciences, Budapest, Hungary.

1. The notation $a_n = \text{poly}(b_n)$ means that there exist $C,\alpha > 0$ such that $a_n \leq C b_n^\alpha$ for n large enough. Moreover, $a_n = \widetilde{O}(b_n)$ means that $a_n/b_n = \text{poly}(\log n)$ for n large enough.

complex problems. Consider a polling institute that has to estimate as accurately as possible the average income of a country, given a finite budget for polls. The institute has call centers in every region in the country, and gives a part of the total sampling budget to each center so that they can call random people in the area and ask about their income. A naive method would allocate a budget proportionally to the number of people in each area. However some regions show a high variability in the income of their inhabitants whereas others are very homogeneous. Now if the polling institute knows the level of variability within each region, it could adjust the budget allocated to each region in a more clever way (allocating more polls to regions with high variability) in order to reduce the final estimation error.

This example is just one of many for which an efficient method of sampling a function with natural strata (i.e., the regions) is of great importance. Note that even in the case that there are no natural strata, it is always a good strategy to design arbitrary strata and allocate a budget to each stratum that is proportional to the size of the stratum, compared to a crude Monte Carlo. There are many good surveys on the topic of stratified sampling for Monte Carlo (Glasserman, 2004; Rubinstein and Kroese, 2008, Subsection 5.5). It is sometimes used in conjunction with other variance reduction techniques, such as importance sampling, antithetic sampling, or control-variables. However, in contrast with those mentioned above, stratified sampling can be used even without substantial knowledge about the function to be evaluated or the sampling distribution (though, to construct effective strata, some knowledge on the variance on different domain areas is better).

The main problem for performing an efficient sampling is that the variances within the strata (in the previous example, the income variability per region) are unknown. One possibility is to estimate the variances *online* while sampling the strata. There is some interesting research along this direction (Arouna, 2004; Etoré and Jourdain, 2010; Kawai, 2010). The work of Etoré and Jourdain (2010) matches exactly our problem of designing an efficient adaptive sampling strategy. In this paper, they propose to sample according to the empirical estimates of the standard deviations of the strata, whereas Kawai (2010) addresses a computational complexity problem which is slightly different from ours. The recent work of Etoré et al. (2011) describes a strategy that enables to sample *asymptotically* according to the (unknown) standard deviations of the strata and at the same time adapts the shape (and number) of the strata online. This is a very difficult problem, especially in high dimension, that we will not address here, although we think this is a very interesting and promising direction for further research.

These works provide asymptotic convergence of the variance of the estimate to the targeted stratified variance divided by the sample size (Rubinstein and Kroese, 2008, Subsection 5.5), see also (5) in this paper. They also prove that the number of pulls within each stratum converges asymptotically to the desired number of pulls, that is, the optimal allocation if the variances per stratum were known. Like Etoré and Jourdain (2010), we consider a stratified Monte Carlo setting with fixed strata. Our contribution is to design a sampling strategy for which we can derive a finite-time analysis (where 'time' refers to the number of samples). This enables us to predict the quality of our estimate for any given budget n.

We model this problem using the setting of multi-armed bandits where our goal is to estimate a weighted average of the mean values of the arms. For quite complete surveys on the classical bandit setting, see for example, the surveys of Cesa-Bianchi and Lugosi (2006);

Bubeck and Cesa-Bianchi (2012), and see also the seminal papers of Lai and Robbins (1985), and Auer et al. (2002). Although our goal is different from a usual bandit problem where the objective is to play the best arm as often as possible, this problem also exhibits an *exploration-exploitation trade-off*. The arms have to be pulled both in order to estimate the initially unknown variability of the arms (exploration) and to allocate correctly the budget according to our current knowledge of the variability (exploitation).

This topic has already been formalized in terms of a bandit problem in the master thesis of Grover (2009), where an algorithm named GAFS-WL (Greedy Allocation with Forced Selection - Weighted Loss) is presented. It deals with stratified sampling, that is, it targets an allocation which is proportional to the standard deviation (and not to the variance) of a stratum times its size, see the book of Rubinstein and Kroese (2008) and also as explained later on in this paper. Grover (2009) defines a proxy on the overall mean squared error (MSE, defined in Equation 1 below), the weighted sum of the per stratum MSE's (defined in Equation 3 below), that he calls *loss*. He proves that the difference between this loss of GAFS-WL and the optimal static loss is of order $\text{poly}(K)\widetilde{O}(n^{-3/2})$, where the $\widetilde{O}(\cdot)$ depends of the arm distributions. Another approach for this problem, still with a bandit formalism, can be found in the paper of Carpentier and Munos (2011), where another algorithm, based on Upper-Confidence-Bounds (UCB) on the standard deviations, was proposed. This algorithm is inspired by the celebrated UCB strategy (Auer et al., 2002), that is designed for the classical bandit setting. The algorithm, called MC-UCB, samples the arms proportionally to an UCB on the standard deviation times the size of the stratum. The authors provided finite-time, problem-dependent and problem-independent bounds for the weighted MSE loss of this algorithm. The first one corresponds to the bound in the work of Grover (2009), the latter one differs from it. Finally, Carpentier and Munos (2012) developed a lower bound for this problem, stating that the pseudo-regret (defined in Section 2 below) of any algorithm for this problem cannot be significantly smaller in a problem-independent minimax sense than $\frac{K^{1/3}}{n^{4/3}}$. In addition, they prove that the problem-independent upper bound on the pseudo-regret of MC-UCB matches this bound up to some $\log n$ factor.

Note that a different, but closely analogous problem is when, instead of a weighted sum of the per arm MSE's, the maximum of these MSE's have to be minimized (e.g., because the weights are unknown). This is dealt with by Carpentier et al. (2011, 2015) for UCB-type algorithms (CH-AS, B-AS) and by Antos et al. (2010) for GAFS-type algorithm (GAFS-MAX).

Recall that in our original stratified sampling problem, however, the natural intuitive measure of performance is not the weighted MSE loss defined by Grover (2009); Carpentier and Munos (2011, 2012), but the total MSE of estimating the weighted average of the mean values of the strata. It is a very important open question to link this total MSE loss to the weighted MSE loss. Without this link, the theoretical analyses which are provided do not give bounds in terms of the natural performance measure.

Contributions. In this paper we extend the analysis of MC-UCB by Carpentier and Munos (2011). Our contributions are the following:

- We provide finite-time bounds on the MSE of the estimate of the mean value. To the best of our knowledge, these are the first finite-time results for the problem of adaptive

stratified Monte Carlo which target directly a usual loss measure (i.e., the total MSE). These consist of: (i) A distribution-dependent bound of order $\text{poly}(\lambda_{\min}^{-1})\widetilde{O}(n^{-3/2})$ that depends on the disparity $\lambda_{\min}$ of the strata (a measure of the problem complexity defined in Equation 6 below), and which corresponds to a stationary regime where the budget n is large compared to this complexity. (ii) A distribution-free bound of order $\text{poly}(K)\widetilde{O}(n^{-7/6})$ that does not depend on the disparity of the strata, and corresponds to a transitory regime where n is small compared to the problem complexity. (iii) The latter bound is sharpened to order $\text{poly}(K)\widetilde{O}(n^{-4/3})$ when each arm distribution is symmetric. Notably, all these bounds yield $o(1/n)$ regret rate.

- We detail the proofs of Carpentier and Munos (2011), which have not been published in full version due to space constraints. They correspond to two pseudo-regret bounds: a distribution-dependent one of order $\lambda_{\min}^{-3/2}\widetilde{O}(n^{-3/2})$ and a distribution-free one of order $K^{1/3}\widetilde{O}(n^{-4/3})$.

The rest of the paper is organized as follows. In Section 2 we formalize the problem and introduce the notations used throughout the paper. Section 3 introduces the MC-UCB algorithm and reports performance bounds on the number of pulls, the weighted MSE loss, the total MSE loss, and the pseudo-loss under sub-Gaussian assumption on the arm distributions. We then discuss the results in Section 4. Finally, Section 5 concludes the paper and suggests future works. The appendices contain useful lemmata and the proofs.

2. Preliminaries

The allocation problem mentioned in the previous section is formalized as a K-armed bandit problem where each arm (stratum) $k = 1, \dots, K$ is characterized by a distribution ν_k with mean value μ_k and variance σ_k^2. At each round $t \geq 1$, an allocation strategy (or algorithm) $\mathcal{A}$ selects an arm k_t adaptively based on past samples, and then receives a sample drawn from ν_{k_t} that is conditionally independent of the past samples given k_t. Let $(w_k)_{k=1,\dots,K}$ denote a known set of positive weights (measure of stratum i) which sum to 1. The goal is to define a strategy that estimates as precisely as possible $\mu = \sum_{k=1}^K w_k \mu_k$ using a total budget of n samples.

Let $\mathbb{I}\{E\}$ be the indicator variable of event E, that is, $\mathbb{I}\{E\} = 1$ if and only if E holds, otherwise $\mathbb{I}\{E\} = 0$. Let us write $T_{k,t} = \sum_{s=1}^t \mathbb{I}\{k_s = k\}$ for the number of times arm k has been pulled up to time t and $\hat{\mu}_{k,t} = \frac{1}{T_{k,t}} \sum_{s=1}^{T_{k,t}} X_{k,s}$ for the empirical estimate of the mean μ_k at time t, where $X_{k,s}$ denotes the sample received when pulling arm k for the s^{th} time. After n rounds, an algorithm $\mathcal{A}$ returns the empirical estimate $\hat{\mu}_{k,n}$ of μ_k for each arm and also their weighted average $\hat{\mu}_n = \sum_{k=1}^K w_k \hat{\mu}_{k,n}$ as the empirical estimate of μ.

For any algorithm $\mathcal{A}$, we use the *total mean (expected) squared error* (MSE) loss of $\hat{\mu}_n$ as performance measure in estimating μ:

$$\bar{L}_n(\mathcal{A}) = \mathbb{E}\big[(\hat{\mu}_n - \mu)^2\big] = \mathbb{E}\bigg[\Big(\sum_{k=1}^K w_k(\hat{\mu}_{k,n} - \mu_k)\Big)^2\bigg], \tag{1}$$

where $\mathbb{E}[\cdot]$ is the expectation integrated over all the samples of all arms. The goal is to define an allocation strategy that minimizes the total MSE loss defined by (1). The total

MSE loss can be decomposed as

$$\bar{L}_n(\mathcal{A}) = \underbrace{\sum_{k=1}^{K} w_k^2 \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)^2\big]}_{L_n(\mathcal{A})} + \sum_{k=1}^{K} \sum_{k' \neq k} w_k w_{k'} \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{k',n} - \mu_{k'})\big]. \tag{2}$$

Here the *weighted MSE loss*

$$L_n(\mathcal{A}) = \sum_{k=1}^{K} w_k^2 \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)^2\big] \tag{3}$$

is equal to the loss defined by Grover (2009); Carpentier and Munos (2011). Thus our analysis for stratified sampling problem implicitly covers the other problem, where instead of estimating μ, the goal is estimating all μ_k simultaneously under a weighted MSE loss $L'_n(\mathcal{A}) = \sum_{k=1}^{K} p_k(\hat{\mu}_{k,n} - \mu_k)^2$, since this loss is essentially the same as $L_n(\mathcal{A})$. Such a setting is referred to sometimes as an *active learning* (or active regression estimation) problem in the literature (e.g., Grover, 2009). This case is even simpler in the sense that we do not have to bother with the cross product-terms in (2).

Note that if all the $T_{k,n}$ are *deterministic*, then in the cross product-terms

$$\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{k',n} - \mu_{k'})\big] = \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)\big]\mathbb{E}\big[(\hat{\mu}_{k',n} - \mu_{k'})\big] = 0 \cdot 0 = 0,$$

and also $\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)^2\big] = \sigma_k^2 / T_{k,n}$. This implies that in this case

$$\bar{L}_n(\mathcal{A}) = L_n(\mathcal{A}) = \sum_{k=1}^{K} w_k^2 \frac{\sigma_k^2}{T_{k,n}}. \tag{4}$$

This gives rise to the definition of

$$\widetilde{L}_n(\mathcal{A}) = \sum_{k=1}^{K} w_k^2 \mathbb{E}\left[\frac{\sigma_k^2}{T_{k,n}}\right]$$

for any algorithm $\mathcal{A}$ (with sample dependent $T_{k,n}$'s) as an alternative performance measure. We call $\widetilde{L}_n(\mathcal{A})$ *pseudo-loss*, as it is a proxy of $\bar{L}_n(\mathcal{A})$ and $L_n(\mathcal{A})$. It is obviously equal to them for deterministic $T_{k,n}$'s.

2.1 Optimal Allocation

Although (4) does not hold when the numbers of pulls of an adaptive algorithm depend on the observed samples and thus are random, it holds when each arm is pulled a deterministic number of times. Thus if the variances of the arms were known in advance, one could design an optimal deterministic (i.e., static, non-adaptive) allocation strategy $\mathcal{A}^*$ by choosing $T_{k,n} = T^*_{k,n}$ such that they minimize $\bar{L}_n$ under the constraint $\sum_{k=1}^{K} T^*_{k,n} = n$. This optimal deterministic allocation of $\mathcal{A}^*$ is to pull each arm k proportionally to $w_k \sigma_k$ (up to rounding effects), that is, given by

$$T^*_{k,n} = \frac{w_k \sigma_k}{\sum_{i=1}^{K} w_i \sigma_i} n.$$

This achieves the loss

$$\bar{L}_n(\mathcal{A}^*) = L_n(\mathcal{A}^*) = \widetilde{L}_n(\mathcal{A}^*) = \frac{\Sigma_w^2}{n}, \tag{5}$$

where $\Sigma_w \stackrel{\text{def}}{=} \sum_{k=1}^K w_k \sigma_k$. We assume in the sequel that $\Sigma_w > 0$, that is, $\exists k$ that $\sigma_k > 0$. We define also $\bar{\Sigma} \stackrel{\text{def}}{=} \max_k \sigma_k$. In the following, we write

$$\lambda_k \stackrel{\text{def}}{=} \frac{T^*_{k,n}}{n} = \frac{w_k \sigma_k}{\Sigma_w}$$

for the optimal allocation proportion for arm k and

$$\lambda_{\min} \stackrel{\text{def}}{=} \min_{1 \le k \le K} \lambda_k \qquad , \qquad \underline{w} \stackrel{\text{def}}{=} \min_{1 \le k \le K} w_k. \tag{6}$$

Note that a small $\lambda_{\min}$ means a large disparity of the quantities $\{w_k \sigma_k\}_{k \le K}$. It will turn out that $\lambda_{\min}$ seems to characterize the hardness of a problem.

2.2 Uniform Allocation

Another possible deterministic allocation is the *proportional* or *uniform strategy* $\mathcal{A}^u$ which assumes uniform standard deviations (e.g., since the σ_k's are unknown and thus the optimal allocation is out of reach), that is, allocates such that $T_k^u = \frac{w_k}{\sum_{i=1}^K w_i} n = w_k n$. Its loss is

$$\bar{L}_n(\mathcal{A}^u) = L_n(\mathcal{A}^u) = \widetilde{L}_n(\mathcal{A}^u) = \sum_{k=1}^K \frac{w_k \sigma_k^2}{n} = \frac{\Sigma_{w,2}}{n},$$

where $\Sigma_{w,2} = \sum_{k=1}^K w_k \sigma_k^2$. Note that using either Jensen's or Cauchy-Schwarz's inequality, we can see that $\Sigma_w^2 \le \Sigma_{w,2}$ with equality if and only if all the σ_k's are equal. Thus $\mathcal{A}^*$ is always at least as good as $\mathcal{A}^u$. In addition, since $\sum_k w_k = 1$, we have

$$\Sigma_{w,2} - \Sigma_w^2 = \sum_k w_k (\sigma_k - \Sigma_w)^2.$$

The difference between those two quantities is the weighted quadratic variation of the σ_k's ($1 \le k \le K$) around their weighted mean Σ_w. As a result the gain of $\mathcal{A}^*$ compared to $\mathcal{A}^u$ grows with the disparity of the σ_k's.

We would like to do better than the uniform strategy by considering an adaptive strategy $\mathcal{A}$ that would estimate all σ_k at the same time as it tries to implement an allocation strategy as close as possible to the optimal allocation algorithm $\mathcal{A}^*$. This introduces a natural trade-off between exploration needed to improve the estimates of the variances and exploitation of the current estimates to allocate the pulls near optimally.

2.3 Definition of Regret

In order to assess how well $\mathcal{A}$ solves the *exploration-exploitation trade-off* above and manages to sample according to the true standard deviations *without knowing them in advance*, we compare its performance to that of the optimal allocation strategy $\mathcal{A}^*$. For this purpose

we define the notion of *total/weighted MSE regret* of an adaptive algorithm $\mathcal{A}$ as the difference between the total/weighted MSE loss incurred by $\mathcal{A}$ and the optimal loss, respectively:

$$\bar{R}_n(\mathcal{A}) = \bar{L}_n(\mathcal{A}) - \frac{\Sigma_w^2}{n} \quad , \quad R_n(\mathcal{A}) = L_n(\mathcal{A}) - \frac{\Sigma_w^2}{n}.$$

The total MSE regret indicates how much we loose in terms of MSE by not knowing in advance the standard deviations σ_k. Note that since $\bar{L}_n(\mathcal{A}^*) \propto 1/n$ by (5), a consistent strategy, that is, one which is asymptotically equivalent to the optimal strategy, is obtained whenever its regret is negligible compared to $1/n$.

We also define the *pseudo-regret*, a proxy for the MSE regret, as the difference between the pseudo-loss incurred by the algorithm and the optimal loss:

$$\widetilde{R}_n(\mathcal{A}) = \widetilde{L}_n(\mathcal{A}) - \frac{\Sigma_w^2}{n}.$$

It is important to derive bounds for $\bar{R}_n(\mathcal{A})$ when $T_{k,n}$'s are random. Taking the decomposition (2), a natural way to proceed is to prove that both
(i) $R_n(\mathcal{A})$ is small and
(ii) the cross product-terms $\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{k',n} - \mu_{k'})\big]$ are small.

Note that for $K = 1$, for any $\mathcal{A}$, $T_{1,n} = T^*_{1,n} = n$ and $\bar{R}_n(\mathcal{A}) = R_n(\mathcal{A}) = \widetilde{R}_n(\mathcal{A}) = 0$, thus we assume $K \geq 2$ from now on.

3. Allocation Based on Monte Carlo Upper Confidence Bound

We now describe the main algorithm and the associated bounds.

3.1 The Algorithm

In this section, we introduce our adaptive algorithm for the allocation problem, called *Monte Carlo Upper Confidence Bound* (MC-UCB). The algorithm computes a high-probability bound on the standard deviation of each arm and samples the arms proportionally to their bounds times the corresponding weights. The MC-UCB algorithm, $\mathcal{A}_{\text{MC-UCB}}$, is described in Figure 1. It requires a parameter β as input, which should be chosen as explained below after Assumption 1.

Input: β
Initialize: Pull each arm twice.
for $t = 2K + 1, \ldots, n$ **do**
  Compute $B_{k,t}$ using (7) for each arm $1 \leq k \leq K$
  Pull an arm $k_t \in \arg\max_{1 \leq k \leq K} B_{k,t}$
end for
Output: $\hat{\mu}_{k,n}$ for each arm $1 \leq k \leq K$ and $\hat{\mu}_n$

Figure 1: The pseudo-code of the MC-UCB algorithm.

The algorithm starts by pulling each arm twice in rounds $t = 1$ to $2K$. From round $t = 2K + 1$ on, it computes an upper confidence bound

$$B_{k,t} = \frac{w_k}{T_{k,t-1}}\Big(\hat{\sigma}_{k,t-1} + \frac{2\beta}{\sqrt{T_{k,t-1}}}\Big) \tag{7}$$

on the standard deviation σ_k for each arm k, and then pulls the one with largest $B_{k,t}$. The bounds $B_{k,t}$ are built by using Lemma 10 (and Corollary 16) and based on the empirical standard deviation $\hat{\sigma}_{k,t-1}$:

$$\hat{\sigma}^2_{k,t-1} = \frac{1}{T_{k,t-1} - 1} \sum_{i=1}^{T_{k,t-1}} (X_{k,i} - \hat{\mu}_{k,t-1})^2, \tag{8}$$

where $X_{k,i}$ is the i-th sample received when pulling arm k and $T_{k,t-1}$ is the number of pulls allocated to arm k up to time $t - 1$. After n rounds, $\mathcal{A}_{\text{MC-UCB}}$ returns the empirical mean $\hat{\mu}_{k,n}$ for each arm $1 \le k \le K$ and also their weighted average $\hat{\mu}_n$.

The motivation to use such an adaptive algorithm instead of classical strategies using, for example, a limited pre-run to get preliminary estimates of the variances is that the latter needs to know the sample size in advance, and will not be able to adapt the length of the exploration phase to the difficulty of the problem. For instance, a strategy that uses e.g., $\approx n^{2/3}$ samples for variance estimation will have minimax-optimal problem-independent rate (up to a log factor) but will display a suboptimal problem-dependent regret rate, i.e., $n^{-4/3}$. On the other hand, a strategy that uses e.g., $\approx n/\log n$ samples for variance estimation will have an optimal problem-dependent regret (of order $n^{-3/2}$ up to a log factor). The main advantage of adaptive strategies such as the one we provide is that it adapts the length of exploration phase to the difficulty of the problem.

We are giving two analyses of $\mathcal{A}_{\text{MC-UCB}}$, a *problem-dependent* and a *problem-independent* one, which are interesting in the stationary and the transitory regimes of the run time of the algorithm, respectively. We will comment on this later in Section 4.

3.2 Assumption on the Arm Distributions and Setting β

Before stating the main results of this section, we state the assumption that the distributions are sub-Gaussian, which includes, for example, Gaussian or bounded distributions. See the paper of Buldygin and Kozachenko (1980) for more precision.

Assumption 1 *There exist $c_1, c_2 > 0$ such that for all $1 \le k \le K$ and any $\epsilon > 0$,*

$$\mathbb{P}_{X\sim\nu_k}(|X - \mu_k| \ge \epsilon) \le c_2 \exp(-\epsilon^2/c_1). \tag{9}$$

The parameters c_1 and c_2 characterize the maximal heaviness of the tails of the arm distributions. Since (9) is equivalent to

$$\mathbb{P}\left(|X_{k,t} - \mu_k| \ge \sqrt{c_1 \log(c_2/\delta)}\right) \le \delta \qquad \text{for any } 0 < \delta < c_2,$$

$\sqrt{c_1 \log(c_2/\delta)}$ can be seen as a high probability bound on the centered samples.

For bounded arm distributions, parameter β of $\mathcal{A}_{\text{MC-UCB}}$ should be generally set as $c\sqrt{\log(2/\delta)}$, where c is the maximum range of the distributions and δ is a chosen significance level corresponding to the estimation of the standard deviations (see Theorem 12). In particular, δ will be chosen as an appropriate decreasing function of n (here $n^{-9/2}$) giving $\beta = \beta_n \propto c\sqrt{\log n}$.

For unbounded distributions satisfying Assumption 1, the role of c is taken by $\propto \sqrt{c_1 \log(c_2/\delta)}$, and the expressions become more involved. Then β will be set as the following function of c_1, c_2, δ, and the total sample size n

$$\beta = \beta_n(\delta) \stackrel{\text{def}}{=} 2\sqrt{c_1 \log(c_2/\delta) \log(2/\delta)} + \frac{\sqrt{c_1 n \delta \log(ec_2/\delta)}}{2(1-\delta)}. \tag{10}$$

This particular form comes from the way we extend a tail inequality for sub-Gaussian random variables in Proposition 14 of Appendix B. In particular, substituting $\delta = n^{-9/2}$ into (10) $\beta = \beta_n$ will be set as the following function of n, c_1, and c_2

$$\beta_n \stackrel{\text{def}}{=} \sqrt{c_1 \log(c_2^2 n^9) \log(4n^9)} + \frac{\sqrt{c_1 \log(ec_2 n^{4.5})}}{2(1-n^{-4.5})n^{7/4}}. \tag{11}$$

To help the reader, subscript n will be used after this substitution. Moreover, note that $B_{k,t}$, k_t, $T_{k,t}$, $\hat{\mu}_{k,t}$, and $\hat{\sigma}_{k,t}$, beside depending on the time step $t \le n$, depend, possibly in an indirect way, also on β, and so on δ, the budget n, c_1, and c_2. An accurate notation would denote also these in some indices to avoid confusion. However, since we consider mostly fixed n, δ, c_1, and c_2, we keep the lighter notations above for the sake of concision.

3.3 High-Probability Bounds on the Number of Pulls

For $2 \le t \le n$, $1 \le k \le K$, write

$$\hat{s}_{k,t}^2 \stackrel{\text{def}}{=} \frac{1}{t-1} \sum_{i=1}^{t} \Big(X_{k,i} - \frac{1}{t} \sum_{t'=1}^{t} X_{k,t'} \Big)^2 \tag{12}$$

for the unbiased empirical variances corresponding to the first t samples from arm k and also $\hat{s}_{k,t} \stackrel{\text{def}}{=} \sqrt{\hat{s}_{k,t}^2}$. Then we have $\hat{\sigma}_{k,t} = \hat{s}_{k,T_{k,t}}$ as computed in (8).

To conduct our analysis, first we state upper and lower bounds on the difference between the allocation $T_{k,n}$ implemented by the MC-UCB algorithm run by parameter β and the optimal allocation $T_{k,n}^*$ for each arm which hold on the event that all standard deviation estimations $\hat{s}_{k,t}$ are quite accurate, namely on

$$\xi = \xi_{K,n}(\delta) \stackrel{\text{def}}{=} \bigcap_{1 \le k \le K,\, 2 \le t \le n} \left\{ |\hat{s}_{k,t} - \sigma_k| \le \frac{2\beta}{\sqrt{t}} \right\}, \tag{13}$$

where β is given by (10). Later Corollary 16 will show that a small δ implies a high probability $\mathbb{P}(\xi)$ under Assumption 1, thus we can use these results to derive the various regret bounds in Subsections 3.4–3.7 for the algorithm. The proofs of Lemma 1 and 2 are in Appendix A.

Problem-dependent bound. All of our problem-dependent bounds (Lemma 1, Propositions 3, 8, partially Proposition 6 and Theorem 7) contain $1/\lambda_{\min}$ and so become void (actually trivial) if $\lambda_{\min} = 0$.[2] Thus we assume $\lambda_{\min} > 0$ in their proofs.

Lemma 1 *Let Assumption 1 hold. For any $0 < \delta \leq 1$, $n \geq 4K$, and any arm $1 \leq p \leq K$, on ξ, the allocation $T_{p,n}$ implemented by $\mathcal{A}_{MC\text{-}UCB}$ satisfies*

$$\frac{w_p \sigma_p}{T_{p,n}} \leq \frac{\Sigma_w}{n} + \frac{12\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w}{n^2}, \tag{14}$$

and consequently $T_{p,n} - T^*_{p,n}$ satisfies

$$-4\lambda_p \left(\frac{3\beta}{\Sigma_w \lambda_{\min}^{3/2}} \sqrt{n} + K \right) \leq T_{p,n} - T^*_{p,n} \leq 4 \left(\frac{3\beta}{\Sigma_w \lambda_{\min}^{3/2}} \sqrt{n} + K \right), \tag{15}$$

where β is given by (10).

In (15), $|T_{p,n} - T^*_{p,n}|$ is bounded by a quantity of order $\sqrt{n}$. This is directly linked to the parametric rate of convergence of the estimation of σ_k, which is of order $1/\sqrt{n}$. Note that (15) also shows the inverse dependency on the smallest optimal allocation proportion $\lambda_{\min}$.

Problem-independent bound.

Lemma 2 *Let Assumption 1 hold. For any $0 < \delta \leq 1$, $n \geq 4K$, and any arm $1 \leq p \leq K$, on ξ, the allocation $T_{p,n}$ implemented by $\mathcal{A}_{MC\text{-}UCB}$ satisfies*

$$T_{p,n} \geq \frac{(w_p n)^{2/3}}{\gamma^2} \qquad \textit{and} \tag{16}$$

$$\frac{w_p \sigma_p}{T_{p,n}} \leq \frac{\Sigma_w}{n} + \frac{12K^{1/3}\beta\gamma}{n^{4/3}} + \frac{4K\Sigma_w}{n^2}, \tag{17}$$

and consequently $T_{p,n} - T^*_{p,n}$ satisfies

$$-4\lambda_p \left(\frac{3K^{1/3}\beta\gamma}{\Sigma_w} n^{2/3} + K \right) \leq T_{p,n} - T^*_{p,n} \leq 4 \left(\frac{3K^{1/3}\beta\gamma}{\Sigma_w} n^{2/3} + K \right),$$

where β is given by (10) *and $\gamma = \gamma_n(\delta) \stackrel{\text{def}}{=} (\bar{\Sigma}/\beta + \sqrt{8})^{1/3}$.*

Unlike in the bounds proved in Lemma 1, here $|T_{p,n} - T^*_{p,n}|$ is bounded by a quantity of order $n^{2/3}$ without any inverse dependency on $\lambda_{\min}$.

2. There are good chances in this case that by refined analyses and setting $\lambda_{\min} = \min_{1 \leq k \leq K : \lambda_k > 0} \lambda_k$ (that is > 0), the same formulae can be proven giving finite bounds.

3.4 Bounds on the Weighted MSE Regret of $\mathcal{A}_{\text{MC-UCB}}$

To simplify our bounds, we introduce

$$C_\beta = C_{\beta,n} \overset{\text{def}}{=} \sqrt{c_1}(9\log n + 1.6\log(c_2+1)) \qquad \text{and} \tag{18}$$

$$C_\xi = C_{\xi,n} \overset{\text{def}}{=} c_1 \log(ec_2 n^{7/2}/2K) \tag{19}$$
$$(\ < c_1(7\log n/2 + \log c_2) \qquad \text{for } K \geq 2 \ ),$$

which depend only polynomially on $\log n$, $\sqrt{c_1}$, and $\log c_2$. We now report the bounds on $R_n(\mathcal{A}_{\text{MC-UCB}})$. The proofs are given in Appendix D.

Problem-dependent bound. This result depends crucially on $\lambda_{\min}^{-1}$ which is a measure of the disparity of the products of the standard deviations and the weights. For this reason we refer to it as "distribution-dependent" result. Its proof relies on the upper- and lower bounds on $T_{k,t} - T_{k,t}^*$ in Lemma 1.

Proposition 3 *Let Assumption 1 be verified for two parameters c_1, $c_2 \geq 1$. If β_n is given by (11), then for $n \geq 4K$ it holds for $\mathcal{A}_{MC\text{-}UCB}$ that*

$$R_n(\mathcal{A}_{MC\text{-}UCB}) \leq \frac{24\Sigma_w C_\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{288 C_\beta^2}{n^2\lambda_{\min}^3} + \frac{\sqrt{K}C_\xi + 32K\Sigma_w^2}{2n^2},$$

where C_β and C_ξ are given by (18) and (19).

Problem-independent bound. Now we report our second bound on $R_n(\mathcal{A}_{\text{MC-UCB}})$ that does not depend on $\lambda_{\min}^{-1}$ at all. This is obtained at the price of the worse rate $K^{1/3}\widetilde{O}(n^{-4/3})$. Its proof relies on the upper- and lower bounds on $T_{k,t} - T_{k,t}^*$ in Lemma 2.

Proposition 4 *Let Assumption 1 be verified for two parameters c_1, $c_2 \geq 1$. If β_n is given by (11), then for $n \geq 4K$ it holds for $\mathcal{A}_{MC\text{-}UCB}$ that*

$$R_n(\mathcal{A}_{MC\text{-}UCB}) \leq \frac{36K^{1/3}\Sigma_w C_\beta}{n^{4/3}} + \frac{K^{2/3}(2058C_\beta^2 + 32\Sigma_w^2) + K^{1/6}C_\xi}{(2n)^{5/3}},$$

where C_β and C_ξ are given by (18) and (19).

Note that this bound is not entirely distribution free, since Σ_w appears. But, as proven in Appendix B.3 using Assumption 1, $\Sigma_w^2 \leq c_1\log(ec_2)$.

For Gaussian distributions with variance 1, we can take $c_1 = c_2 = 1$, and the main coefficient of $\log n/(n\lambda_{\min})^{3/2}$ in Proposition 3 and of $K^{1/3}\log n/n^{4/3}$ in Proposition 4 are upper bounded by 216 and 324, respectively.

3.5 Bounds on the Cross Product-Terms

The difficulty in bounding the cross product-terms, that is, the second term in the right-hand side of (2), comes from the fact that the $(T_{k,n})_{k\leq K}$ depend on the samples (in particular, for $\mathcal{A}_{\text{MC-UCB}}$, on the empirical standard deviations $(\hat{\sigma}_{k,t})_{k\leq K, t\leq n}$). This dependence can make correlation between $\hat{\mu}_{k,n}$ and $\hat{\mu}_{k',n}$. Thus, for general distributions, we cannot see

obvious, direct reason why a cross product-term should be equal to the product of the corresponding biases, and so be close to 0. We give three results for these cross product-terms. The first one corresponds to the specific case where the distributions of the arms are symmetric. The next two provide a problem-dependent and a problem-independent bound in the general case. All these are partial results for proving bounds on $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$ and proven in Appendix E.

Arms with symmetric distributions. The first result holds in the specific case of symmetric distributions. Intuitively speaking, in this setting, conditioning on the empirical standard deviations does not change the mean of the samples (and sample averages). This implies that for $k \neq k'$, $\hat{\mu}_{k,n} - \mu_k$ and $\hat{\mu}_{k',n} - \mu_{k'}$ are conditionally uncorrelated. From that we deduce the following result.

Proposition 5 *Assume that each distribution ν_k is symmetric around μ_k, respectively. For $\mathcal{A}_{\text{MC-UCB}}$ launched with any parameter β_n, we have that*

$$\sum_{k=1}^{K} \sum_{k' \neq k} w_k w_{k'} \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{k',n} - \mu_{k'})\big] = 0.$$

Though mostly of theoretical interest, the significance of this result is its indication that the rate might be improvable for other distributions, as well.

Problem-dependent and problem-independent bound in general. The following proposition gives bounds on the cross product-terms. This can be seen as an intermediary step in linking the weighted MSE regret and the true regret. Its proof relies on the specific structure of $\mathcal{A}_{\text{MC-UCB}}$ through the use of Lemma 1 and 2.

Proposition 6 *Let Assumption 1 be verified for two parameters c_1, $c_2 \geq 1$. If β_n is given by* (11)*, then (for n large enough compared to K, c_1, $\log c_2$, and $1/\Sigma_w$) the cross product-terms for $\mathcal{A}_{\text{MC-UCB}}$ are bounded as*

$$\sum_{k=1}^{K} \sum_{q \neq k} w_k w_q \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{q,n} - \mu_q)\big] \leq \text{poly}(\Sigma_w c_1 \log c_2 / \lambda_{\min}) \widetilde{O}(n^{-3/2}),$$

and

$$\sum_{k=1}^{K} \sum_{q \neq k} w_k w_q \mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{q,n} - \mu_q)\big] \leq \text{poly}(K \Sigma_w c_1 \log c_2 / \underline{w}) \widetilde{O}(n^{-7/6}),$$

where $\underline{w}$ is given by (6) *(and $\widetilde{O}(\cdot)$ does not depend on $\lambda_{\min}$).*

Note that the latter bound, depending on $\underline{w}$, is not really problem-independent (considering w_k's to be part of the problem), but it is independent of the arm distributions, particularly of $\lambda_{\min}$.

3.6 Bounds on the Total-Regret

From the decomposition (2) for $\mathcal{A}_{\text{MC-UCB}}$ and Propositions 3, 4, 6, and 5, we can deduce our main result, a bound on the true regret $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$:

Theorem 7 *Let Assumption 1 be verified for two parameters $c_1 > 0$, $c_2 \geq 1$. If β_n is given by* (11)*, then (for n large enough compared to K, c_1, $\log c_2$, and $1/\Sigma_w$) the true regret of $\mathcal{A}_{MC\text{-}UCB}$ is bounded as*

$$\bar{R}_n(\mathcal{A}_{MC\text{-}UCB}) = \text{poly}(\Sigma_w c_1 \log c_2/\lambda_{\min})\widetilde{O}(n^{-3/2}),$$

and

$$\bar{R}_n(\mathcal{A}_{MC\text{-}UCB}) = \text{poly}(K\Sigma_w c_1 \log c_2/\underline{w})\widetilde{O}(n^{-7/6})$$

(thus, in particular, $\bar{R}_n = o(1/n)$). If each distribution ν_k is symmetric around μ_k, then the cross product-terms are 0, and the following tighter problem-independent bound holds

$$\bar{R}_n(\mathcal{A}_{MC\text{-}UCB}) = R_n(\mathcal{A}_{MC\text{-}UCB}) = \text{poly}(K\Sigma_w c_1 \log c_2)\widetilde{O}(n^{-4/3}).$$

3.7 Bounds on the Pseudo-Regret

We bound $\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}})$ by a problem-dependent and a problem-independent upper bound that are of the same order in n as the bounds in Propositions 3 and 4, respectively. The proofs are given in Appendix C.

Problem-dependent bound.

Proposition 8 *Let Assumption 1 be verified for two parameters $c_1 > 0$, $c_2 \geq 1$. If β_n is given by* (11)*, then the pseudo-regret of $\mathcal{A}_{MC\text{-}UCB}$ launched with $n \geq 4K$ is bounded as*

$$\widetilde{R}_n(\mathcal{A}_{MC\text{-}UCB}) \leq \frac{12\Sigma_w C_\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{(4K + \sqrt{2}/16)\Sigma_w^2}{n^2},$$

where C_β is given by (18)*.*

Problem-independent bound.

Proposition 9 *Let Assumption 1 be verified for two parameters $c_1 > 0$, $c_2 \geq 1$. If β_n is given by* (11)*, then the pseudo-regret of $\mathcal{A}_{MC\text{-}UCB}$ launched with $n \geq 4K$ is bounded as*

$$\widetilde{R}_n(\mathcal{A}_{MC\text{-}UCB}) \leq \frac{18K^{1/3}\Sigma_w C_\beta}{n^{4/3}} + \frac{(4K + \sqrt{2}/16)\Sigma_w^2}{n^2},$$

where C_β is given by (18)*.*

For Gaussian distributions with variance 1, we can consider $c_1 = c_2 = 1$, and the main coefficient of $\log n/(n\lambda_{\min})^{3/2}$ in Proposition 8 and of $K^{1/3}\log n/n^{4/3}$ in Proposition 9 are upper bounded by 108 and 162, respectively.

4. Discussion on the Results

We make several comments on the algorithm MC-UCB in this section.

4.1 Problem-Dependent and -Independent Bounds on $R_n(\mathcal{A})$ and $\widetilde{R}_n(\mathcal{A})$

Our problem-dependent $\lambda_{\min}^{-3}\widetilde{O}(n^{-3/2})$ upper bound on $R_n(\mathcal{A}_{\text{MC-UCB}})$ in Proposition 3 is similar and comparable to the one provided for GAFS-WL by Grover (2009), where the loss measure is $L_n(\mathcal{A}_{\text{GAFS-WL}})$. Beside this $\lambda_{\min}$-dependent bound for $\mathcal{A}_{\text{MC-UCB}}$, Propositions 4 gives a $\lambda_{\min}$-independent bound of order $K^{1/3}\widetilde{O}(n^{-4/3})$. (Note however, that when there is an arm with 0 variance, GAFS-WL is likely to perform better than MC-UCB, as it will only sample this arm $O(\sqrt{n})$ times, while MC-UCB usually samples it $\Omega(n^{2/3})$ times.) Similarly, Proposition 8 provides a pseudo-regret bound of order $\lambda_{\min}^{-3/2}\widetilde{O}(n^{-3/2})$, whereas Proposition 9 gives a $\lambda_{\min}$-independently bound of order $K^{1/3}\widetilde{O}(n^{-4/3})$.

Hence, for a given problem, that is, a given $\lambda_{\min}$, the distribution-free results of Proposition 4 and 9 are more informative than the distribution-dependent results of Proposition 3 and 8, respectively, in the *transitory regime*, that is, when n is small compared to $\lambda_{\min}^{-1}$. Proposition 3 and 8 is better in the *stationary regime*, that is, for n large enough. This distinction reminds us of the difference between distribution-dependent and distribution-free bounds for the UCB algorithm in usual multi-armed bandits. In that setting, the distribution dependent bound is in $O(K\log n/\Delta)$, where Δ is the difference between the mean value of the two best arms, and the distribution-free bound is in $O(\sqrt{Kn})$ as explained by Auer et al. (2002); Audibert and Bubeck (2009). In many works, these two types of results are called *individual* and *uniform* bounds. For several models, the two bounds correspond with each other, at least in their convergence rates in the sample size for the best possible algorithms (i.e., in some minimax sense). See the thesis of Antos (1999) for a discussion. Our results and proofs suggest that our stratified sampling model is another interesting exception, where these two types of rates must be different.

At first sight, the problem of Monte Carlo integration seems to be more related to the problem of *pure exploration* (Bubeck et al., 2011; Audibert et al., 2010) than to the usual bandit setting: indeed, similarly to the setting of pure exploration, an intermediate objective (linked to the overall objective) is to allocate the number of pulls of the arms proportionally to some unknown problem-dependent quantities. However, we believe that our problem is actually more related to the standard bandit problem, since it gives rise to an exploration-exploitation trade-off.

4.2 The Parameter β of the Algorithm

We saw in (11) that the parameter β_n of $\mathcal{A}_{\text{MC-UCB}}$ should depend on n, c_1, c_2. It is actually such that $\beta_n \approx c'\log n$, where c' can be interpreted as a high probability bound on the range of the samples. We thus simply require a rough bound on the magnitude of the samples. As we saw, in the case of bounded distributions, β_n can be chosen as $\beta_n = c\sqrt{5\log n}$, where c is a true bound on the range of the variables. This is easy to deduce by comparing Corollary 13 and Proposition 14 in Appendix B. The interpretation of this parameter β is quite similar to the interpretation of the parameter in the UCB algorithm of Auer et al. (2002), and its order of magnitude is roughly the same. (In that paper, it is assumed that the distributions of the arms are bounded.) On the other hand, the interpretation of this quantity is quite different from the interpretation of the parameter a of the algorithm UCB-E of Audibert et al. (2010), which characterizes here the complexity of the problem. This is yet another

illustration from the fact that this problem is somehow more related to the standard bandit problem than to the problem of pure exploration.

4.3 Finite-Time Bounds for $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$ and Asymptotic Optimality

The first result in Theorem 7 states that $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$ is of order $\text{poly}(\lambda_{\min}^{-1})\widetilde{O}(n^{-3/2})$. This corresponds to the $\lambda_{\min}$-dependent bound on $R_n(\mathcal{A}_{\text{MC-UCB}})$. Theorem 7 also states that an upper bound on $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$ is of order $\text{poly}(K)\widetilde{O}(n^{-7/6})$. This corresponds to the $\lambda_{\min}$-independent bound on $R_n(\mathcal{A}_{\text{MC-UCB}})$. Unfortunately, in this case, we do not obtain the same order for $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$ as for $R_n(\mathcal{A}_{\text{MC-UCB}})$, that is, $\text{poly}(K)\widetilde{O}(n^{-4/3})$. This comes from the fact that the bound on the cross product-terms in Proposition 6 is of order $\text{poly}(K/\underline{w})\widetilde{O}(n^{-7/6})$. Whether this bound is tight or not is an open problem.

As we bound $\bar{R}_n(\mathcal{A}_{\text{MC-UCB}})$ as $o(1/n)$, $\bar{L}_n(\mathcal{A}_{\text{MC-UCB}})$ is asymptotically not more than $\bar{L}_n(\mathcal{A}^*) = \Sigma_w^2/n$ for *any* problem satisfying Assumption 1. This can be said as $\mathcal{A}_{\text{MC-UCB}}$ is (weakly) consistent; just like the algorithms of Kawai (2010); Etoré and Jourdain (2010).

Note also that whenever there is some disparity among the arms, that is, when $\Sigma_w^2 < \Sigma_{2,w}$, $\mathcal{A}_{\text{MC-UCB}}$ is asymptotically strictly more efficient than the uniform strategy.

4.4 Pseudo-Regret of $\mathcal{A}_{\text{MC-UCB}}$ and the Lower Bound

Carpentier and Munos (2012) provided a $\lambda_{\min}$-independent minimax lower bound for $\widetilde{R}_n(\mathcal{A})$ that is of order $K^{1/3}\Omega(n^{-4/3})$. An important achievement is that the $\lambda_{\min}$-independent upper bound on $\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}})$ in Proposition 9 is of the same order up to a logarithmic factor. Thus, regarding $\widetilde{R}_n(\mathcal{A})$, it is impossible to improve this strategy uniformly for every sub-Gaussian problem more than by a log factor.

Although we do not have a $\lambda_{\min}$-dependent lower bound on $\widetilde{R}_n(\mathcal{A})$ yet, we believe that the $\widetilde{O}(n^{-3/2})$ rate of Proposition 8 cannot be improved in n for general distributions. As it seems from the proofs in Appendix A and C, this rate is a direct consequence of the high probability bounds on the estimates of the standard deviations of the arms which are in $O(1/\sqrt{n})$, and *those bounds are tight*. Because of the minimax lower bound that is of order $K^{1/3}\Omega(n^{-4/3})$, it is however clear that there exists no algorithm with a regret of order $n^{-3/2}$ without any dependence on $\lambda_{\min}^{-1}$ (or another related problem-dependent quantity).

4.5 Making $\mathcal{A}_{\text{MC-UCB}}$ Anytime

An interesting question is whether and how it is possible to make $\mathcal{A}_{\text{MC-UCB}}$ anytime, that is, not requiring the knowledge of the sample horizon n in advance. Although we will not provide formal proofs of this result in this paper, we believe that setting a δ that evolves with the current time as $\delta_t = t^{-9/2}$, is sufficient to make all the regret bounds of this paper hold with slightly modified constants. Some ideas on how to prove these results can be found in the literature (Grover, 2009; Antos et al., 2010; Auer et al., 2002).

4.6 Domains of Application

Monte Carlo integration has many relevant applications in machine learning. Being able to compute precisely an integral is a prerequisite in many methods or algorithms in this field.

Some examples of possible application of the stratified Monte Carlo technique are listed below.

- There are more and more applications in machine learning that are targeting the allocation and placement of various kinds of sensors (as e.g., pollution sensors, temperature sensors, cameras of various kinds, network sensors, etc.). It is a challenge to find a way to place them efficiently, or choose at which frequency to observe their output. The placement of these sensors should depend of the objective that they have to fulfill. In some cases, one wants to use these sensors to compute an integral (for instance, the average pollution level or temperature in a region, the average amount of traffic at a certain time, the average number of customers in a given place in a supermarket, or the average amount of exchange in a network, etc.). The approach of this paper can be used to decide adaptively how to place these sensors, how frequently to inspect them, or how many of them to put depending on the area. In some other cases, the objective is that the sensors provide a good estimate of what they measure in each zone (e.g., local water pressure on a dyke). As mentioned earlier, our algorithm minimizes, with respect to the sample allocation, a weighted (over the strata) mean squared error of estimations. Therefore, our approach also provides good results in such a setting where the objective is to estimate the mean value in each zone, rather than an overall integral.

- A huge domain that is commonly handled in the machine learning community, and in which the aim is often to compute precisely integrals is Bayesian methodology. Indeed, expectations under the posterior distribution are often good estimators for some relevant parameters of the model. Being able to compute these expectations (which are well defined integrals) fast and precisely is both desirable and challenging, and our method provides an alternative for MCMC methods in the computation of such integrals.

- There are many applications in mathematical finance, for example, in the domain of pricing (which essentially sums up to the computation of a complex stochastic integral).

As mentioned below (3), omitting the cross product-terms and focusing on the weighted MSE loss our setting can be interpreted as an active learning framework. This can be a suitable model also in production quality testing, adaptive study design, drug discovery, crowd-sourcing, etc.

5. Conclusions

We provide a finite-time analysis for stratified sampling for Monte Carlo in the case of fixed strata with sub-Gaussian distributions. We report two bounds on the weighted MSE regret: (i) a distribution dependent bound of order $\text{poly}(\lambda_{\min}^{-1})\widetilde{O}(n^{-3/2})$ which is of interest when n is large compared to a measure of disparity $\lambda_{\min}^{-1}$ of the standard deviations (*stationary regime*), and (ii) a distribution free bound of order $\widetilde{O}(K^{1/3}n^{-4/3})$ which is of interest when n is small compared to $\lambda_{\min}^{-1}$ (*transitory regime*). We also link the weighted MSE loss to

the total MSE loss of algorithm MC-UCB, that is the natural measure of performance for the problem. We provide $\text{poly}(\lambda_{\min}^{-1})\widetilde{O}(n^{-3/2})$ problem-dependent and $\text{poly}(K)\widetilde{O}(n^{-7/6})$ problem-independent bounds for the total MSE regret, as well. In case of symmetric arm distributions, the latter rate is improved to $\text{poly}(K)\widetilde{O}(n^{-4/3})$. We give a distribution dependent bound of order $\text{poly}(\lambda_{\min}^{-1})\widetilde{O}(n^{-3/2})$ and a distribution free bound of order $\widetilde{O}(K^{1/3}n^{-4/3})$ also on the pseudo-regret. The latter matches its minimax lower bound in terms of n up to a $\log n$ factor.

Possible directions for future work include: (i) making the MC-UCB algorithm anytime (i.e., not requiring the knowledge of n in advance) and (ii) deriving distribution-dependent lower bound for this problem determining the necessary dependence on $\lambda_{\min}$.

Acknowledgements This research was partially supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 270327 (CompLACS).

Appendices

These appendices contain the proofs of the theorems in the paper. Their organization is as follows.

- Appendix A contains the proofs of the (problem-dependent) Lemma 1) and the (problem-independent) Lemma 2) stating that the number of pulls of any arm is not too far from the optimal allocation for that arm on event ξ.
- Appendix B states some preliminary results which are useful in the regret bound proofs. It first gives (conditional) variance bound for sub-Gaussian random variables. Then it shows that ξ has high probability. It also contains the proof that for any $t \leq n$, $T_{k,t}$ is a stopping time, and applies Wald's identity to the samples from an arm. Next, it states bounds on some other technical quantities outside ξ that are used afterwards. Finally, it gives bounds on the parameters β_n and γ_n.[3]
- Appendix C contains the proofs of the (problem-dependent) Proposition 8 and the (problem-independent) Proposition 9 upper bounding $\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}})$ based on Lemma 1 and 2, respectively. These proofs are simpler than those in Appendix D and can serve as an introduction for the latter.
- Appendix D contains the proofs of the (problem-dependent) Proposition 3 and the (problem-independent) Proposition 4 upper bounding $R_n(\mathcal{A}_{\text{MC-UCB}})$ based on Lemma 1 and 2, respectively. These proofs are quite similar to the ones for bounding $\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}})$ in Appendix C. However, those have to be extended by additional technical steps, for example, using Wald's second identity for sums with random number of terms, to bound $R_n(\mathcal{A}_{\text{MC-UCB}})$ with a quantity reminding to $\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}})$.
- Appendix E provides the proofs of the three bounds on the cross product-terms. The first one holds when the arm distributions are symmetric: then the cross product-terms are exactly 0. The two other bounds, a problem-dependent and a problem-independent

3. As for β, γ_n will be used for γ after this substituting $\delta = n^{-9/2}$.

one, concern the general sub-Gaussian case. These bounds rely on Lemma 1 and 2. Using these together with the results in Appendix D gives bounds on the total regret.

- Appendix F provides the proof of some general technical lemmata.

Appendix A. Proof of the Bounds on the Number of Pulls of the Arms

In this section, we prove Lemma 1 and 2. Recall that their statements hold on the event ξ. This event plays an important role in the proofs of the regret bounds; several statements will be proven on ξ. We transcribe the definition (13) of ξ into the following lemma when the number of samples $T_{k,t}$ are random.

Lemma 10 *For $k = 1, \dots, K$ and $t = 2K, \dots, n$, let $T_{k,t}$ be any random variable taking values in $\{2, \dots, n\}$. Let $\hat{\sigma}^2_{k,t}$ be the empirical variance computed from (8) and β be given by (10). Then, on ξ, we have:*

$$|\hat{\sigma}_{k,t} - \sigma_k| \leq \frac{2\beta}{\sqrt{T_{k,t}}}.$$

All statements in the proofs of this section are meant to hold on ξ.

A.1 Problem-Dependent Bound; Proof of Lemma 1

Proof of Lemma 1 The proof consists of the following three main steps.

Step 1. Properties of the algorithm. Recall the definition of the upper bound used in $\mathcal{A}_{\text{MC-UCB}}$ when $t > 2K$:

$$B_{q,t+1} = \frac{w_q}{T_{q,t}} \left(\hat{\sigma}_{q,t} + \frac{2\beta}{\sqrt{T_{q,t}}} \right), \qquad 1 \leq q \leq K.$$

From Lemma 10, we obtain the following upper and lower bounds for $B_{q,t+1}$ on ξ:

$$\frac{w_q \sigma_q}{T_{q,t}} \leq B_{q,t+1} \leq \frac{w_q}{T_{q,t}} \left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,t}}} \right). \tag{20}$$

Note that as $n \geq 4K$, there exists an arm pulled after the initialization. Let k be such an arm and $t + 1 > 2K$ be the time step when k is pulled for the last time, that is, $T_{k,t} = T_{k,n} - 1 \geq 2$ and $T_{k,t+1} = T_{k,n}$. Since arm k is chosen at time $t + 1$, we have for any arm p

$$B_{p,t+1} \leq B_{k,t+1}. \tag{21}$$

From (20) and the fact that $T_{k,t} = T_{k,n} - 1$, we obtain on ξ

$$B_{k,t+1} \leq \frac{w_k}{T_{k,t}} \left(\sigma_k + \frac{4\beta}{\sqrt{T_{k,t}}} \right) = \frac{w_k}{T_{k,n} - 1} \left(\sigma_k + \frac{4\beta}{\sqrt{T_{k,n} - 1}} \right). \tag{22}$$

Using the lower bound in (20) and the fact that $T_{p,t} \leq T_{p,n}$, we may lower bound $B_{p,t+1}$ on ξ as

$$B_{p,t+1} \geq \frac{w_p \sigma_p}{T_{p,t}} \geq \frac{w_p \sigma_p}{T_{p,n}}. \tag{23}$$

Combining (21), (22), and (23), we obtain on ξ

$$\frac{w_p\sigma_p}{T_{p,n}} \leq \frac{w_k}{T_{k,n}-1}\left(\sigma_k + \frac{4\beta}{\sqrt{T_{k,n}-1}}\right). \tag{24}$$

Note that at this point there is no dependency on t, and on ξ, (24) holds for any p and for any k such that $T_{k,n} > 2$.

Step 2. Lower bound on $T_{p,n}$. From the constraints $\sum_k (T_{k,n}-2) = n-2K$ and $\sum_k \lambda_k = 1$, we can deduce (by indirect proof) that there exists an arm k with $T_{k,n}-2 \geq \lambda_k(n-2K) > 0$, that is, $T_{k,n} > 2$. Thus k satisfies (24). Using (24), $T_{k,n}-1 > \lambda_k(n-2K)$, and $\lambda_k = w_k\sigma_k/\Sigma_w$ implies for any arm p

$$\frac{w_p\sigma_p}{T_{p,n}} < \frac{w_k}{n\lambda_k}\frac{1}{1-2K/n}\left(\sigma_k + \frac{4\beta}{\sqrt{n\lambda_k(1-2K/n)}}\right) \leq \frac{\Sigma_w}{n} + \frac{4K\Sigma_w}{n^2} + \frac{8\sqrt{2}\beta}{n^{3/2}\lambda_k^{3/2}},$$

because $n \geq 4K$. The previous inequality combined with the fact that $\lambda_k \geq \lambda_{\min}$ gives the first inequality (14) of the lemma

$$\frac{w_p\sigma_p}{T_{p,n}} \leq \frac{\Sigma_w}{n} + \frac{12\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w}{n^2}.$$

By rearranging it, we obtain the lower bound on $T_{p,n}$ in (15)

$$T_{p,n} \geq \frac{w_p\sigma_p}{\frac{\Sigma_w}{n} + \frac{12\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w}{n^2}} \geq T^*_{p,n} - 4\lambda_p\left(\frac{3\beta}{\Sigma_w\lambda_{\min}^{3/2}}\sqrt{n} + K\right), \tag{25}$$

where in the second inequality we use $1/(1+x) \geq 1-x$ (for $x > -1$). Note that the lower bound holds on ξ for any arm p.

Step 3. Upper bound on $T_{p,n}$. Using (25) and the fact that $\sum_k T_{k,n} = n$, we obtain

$$T_{p,n} = n - \sum_{k\neq p} T_{k,n} \leq \left(n - \sum_{k\neq p} T^*_{k,n}\right) + \sum_{k\neq p} 4\lambda_k\left(\frac{3\beta}{\Sigma_w\lambda_{\min}^{3/2}}\sqrt{n} + K\right).$$

Since $\sum_{k\neq p}\lambda_k \leq 1$ and $\sum_k T^*_{k,n} = n$, we deduce

$$T_{p,n} \leq T^*_{p,n} + 4\left(\frac{3\beta}{\Sigma_w\lambda_{\min}^{3/2}}\sqrt{n} + K\right). \tag{26}$$

The lemma follows by combining the lower and upper bounds in (25) and (26). ■

A.2 Problem-Independent Bound; Proof of Lemma 2

Proof of Lemma 2

Step 1. Lower bound of order $\Omega(n^{2/3})$. Recall the definition of the upper bound $B_{q,t+1}$ used in $\mathcal{A}_{\text{MC-UCB}}$ when $t > 2K$:

$$B_{q,t+1} = \frac{w_q}{T_{q,t}}\left(\hat{\sigma}_{q,t} + \frac{2\beta}{\sqrt{T_{q,t}}}\right), \qquad 1 \le q \le K.$$

Using Lemma 10 it follows that on ξ, for any q such that $T_{q,t} \ge 2$,

$$\frac{w_q \sigma_q}{T_{q,t}} \le B_{q,t+1} \le \frac{w_q}{T_{q,t}}\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,t}}}\right). \tag{27}$$

Let k be the index of an arm that is such that $T_{k,n} - 2 \ge w_k(n-2K)$. Such an arm always exists for any possible allocation strategy, as $n - 2K = \sum_q (T_{q,n} - 2)$ and $\sum_q w_q = 1$. This implies $T_{k,n} \ge 3$ as $n \ge 4K$, thus arm k is pulled after the initialization. Let $t+1 \le n$ be the last time at which it was pulled, that is, $T_{k,t} = T_{k,n} - 1$ and $T_{k,t+1} = T_{k,n}$. From (27) and the fact that $T_{k,t} > w_k(n-2K)$ and $T_{k,t} \ge 2$, we obtain on ξ

$$B_{k,t+1} \le \frac{w_k}{T_{k,t}}\left(\sigma_k + \frac{4\beta}{\sqrt{T_{k,t}}}\right) < \frac{\max_p \sigma_p + \sqrt{8}\beta}{n-2K}. \tag{28}$$

Since at time $t+1$ the arm k has been pulled, then for any arm q, we have

$$B_{q,t+1} \le B_{k,t+1}. \tag{29}$$

From the definition of $B_{q,t+1}$, and also using the fact that $T_{q,t} \le T_{q,n}$, we deduce on ξ that

$$B_{q,t+1} \ge \frac{2\beta w_q}{T_{q,t}^{3/2}} \ge \frac{2\beta w_q}{T_{q,n}^{3/2}}. \tag{30}$$

Combining (28)–(30), we obtain on ξ

$$\frac{2\beta w_q}{T_{q,n}^{3/2}} < \frac{\max_p \sigma_p + \sqrt{8}\beta}{n-2K} = \frac{\bar{\Sigma} + \sqrt{8}\beta}{n-2K}.$$

Finally, this implies on ξ that for any q,

$$T_{q,n} \ge \left(\frac{2\beta w_q (n-2K)}{\bar{\Sigma} + \sqrt{8}\beta}\right)^{2/3} = \left(\frac{2 - 4K/n}{\bar{\Sigma}/\beta + \sqrt{8}} w_q n\right)^{2/3} \ge \frac{(w_q n)^{2/3}}{(\bar{\Sigma}/\beta + \sqrt{8})^{2/3}} = \frac{(w_q n)^{2/3}}{\gamma^2}$$

recalling $\gamma = (\bar{\Sigma}/\beta + \sqrt{8})^{1/3}$, which proves (16).

Step 2. Properties of the algorithm. We follow a similar analysis to Step 1 of the proof of Lemma 1. Note that as $n \ge 4K$, there exists an arm pulled after the initialization. Let q be any such arm and $t+1 > 2K$ be the time step when q is pulled for the last time, that is, $T_{q,t} = T_{q,n} - 1 \ge 2$ and $T_{q,t+1} = T_{q,n}$. Since arm q is chosen at time $t+1$, we have for any arm p

$$B_{p,t+1} \le B_{q,t+1}. \tag{31}$$

From (27) and $T_{q,t} = T_{q,n} - 1$, we obtain on ξ

$$B_{q,t+1} \leq \frac{w_q}{T_{q,t}}\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,t}}}\right) = \frac{w_q}{T_{q,n}-1}\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,n}-1}}\right). \tag{32}$$

Furthermore, since $T_{p,t} \leq T_{p,n}$ and $T_{p,t} \geq 2$ (as $t \geq 2K$), then on ξ

$$B_{p,t+1} \geq \frac{w_p\sigma_p}{T_{p,t}} \geq \frac{w_p\sigma_p}{T_{p,n}}. \tag{33}$$

Combining (31)–(33), we obtain on ξ

$$\frac{w_p\sigma_p}{T_{p,n}}(T_{q,n}-1) \leq w_q\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,n}-1}}\right).$$

Note that this inequality holds on ξ for any p and for any q such that $T_{q,n} \geq 3$. Summing over all such q on both sides, we obtain on ξ for any arm p

$$\frac{w_p\sigma_p}{T_{p,n}} \sum_{q:T_{q,n}\geq 3}(T_{q,n}-1) \leq \sum_{q:T_{q,n}\geq 3} w_q\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,n}-1}}\right).$$

This implies

$$\frac{w_p\sigma_p}{T_{p,n}}(n-2K) \leq \sum_{q=1}^{K} w_q\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,n}-1}}\right), \tag{34}$$

because $\sum_{q:T_{q,n}\geq 3}(T_{q,n}-1) = n - K - \sum_{q:T_{q,n}\leq 2}(T_{q,n}-1) \geq n - K - K = n - 2K$.

Step 3. Lower bound. Plugging (16) into (34),

$$\begin{aligned}
\frac{w_p\sigma_p}{T_{p,n}}(n-2K) &\leq \sum_q w_q\left(\sigma_q + \frac{4\beta}{\sqrt{T_{q,n}-1}}\right) \\
&\leq \sum_q w_q\left(\sigma_q + 4\beta\sqrt{\frac{2\gamma^2}{(w_q n)^{2/3}}}\right) \\
&\leq \Sigma_w + \frac{4\sqrt{2}\beta\gamma}{n^{1/3}}\sum_q w_q^{2/3} \leq \Sigma_w + \frac{6\beta\gamma K^{1/3}}{n^{1/3}},
\end{aligned}$$

on ξ, since $T_{q,n} - 1 \geq \frac{T_{q,n}}{2}$ (as $T_{q,n} \geq 2$) and because $\sum_q w_q^{2/3} \leq K^{1/3}$ by Jensen's inequality. Finally as $n \geq 4K$, we obtain on ξ the first inequality (17) of the lemma

$$\frac{w_p\sigma_p}{T_{p,n}} \leq \frac{\Sigma_w}{n} + \frac{12K^{1/3}\beta\gamma}{n^{4/3}} + \frac{4K\Sigma_w}{n^2}.$$

We now invert this bound and obtain on ξ the final lower bound on $T_{p,n}$ as follows

$$T_{p,n} \geq \frac{w_p\sigma_p}{\frac{\Sigma_w}{n} + 12K^{1/3}\beta\gamma n^{-4/3} + \frac{4K\Sigma_w}{n^2}} \geq T^*_{p,n} - 4\lambda_p\left(\frac{3K^{1/3}\beta\gamma}{\Sigma_w}n^{2/3} + K\right),$$

as $\frac{1}{1+x} \geq 1-x$. Note that this lower bound holds with high probability for any arm p.

Step 4. Upper bound. An upper bound on $T_{p,n}$ on ξ follows by using $T_{p,n} = n - \sum_{q \neq p} T_{q,n}$ and the previous lower bound, that is

$$T_{p,n} \leq n - \sum_{q \neq p} T^*_{q,n} + \sum_{q \neq p} 4\lambda_q \left(\frac{3K^{1/3}\beta\gamma}{\Sigma_w} n^{2/3} + K \right) \leq T^*_{p,n} + 4 \left(\frac{3K^{1/3}\beta\gamma}{\Sigma_w} n^{2/3} + K \right),$$

because $\sum_{q \neq p} \lambda_q \leq 1$. ■

Appendix B. Main Tools for the Bounds on the Regrets

In this section, we first give a high probability uniform upper bound on the estimation errors of the unbiased empirical standard deviations for sub-Gaussian random variables, then describe other technical tools, properties, and inequalities. Several of these use the following simple lemma giving (conditional) variance bound for sub-Gaussian random variables proven in Appendix F:

Lemma 11 *Let A be an event with $\mathbb{P}(A) \leq \delta$. Let X have a distribution with $\mu \stackrel{\text{def}}{=} \mathbb{E}X$ satisfying* (9) *of Assumption 1 with $c_1 > 0$, $c_2 \geq \delta$, and any $\epsilon > 0$. Then*

$$\mathbb{E}\left[|X-\mu|^2 \mathbb{I}\{A\}\right] \leq \delta c_1 \log(ec_2/\delta).$$

Particularly, the case $\mathbb{P}(A) = \delta = 1$ gives $\operatorname{Var} X \leq c_1 \log(ec_2)$ if $c_2 \geq 1$.

B.1 High Probability Uniform Upper Bound on the Variance Estimation Errors

In this subsection, let $n \geq 2$, $X_1, \ldots, X_n$ be i.i.d. random variables with mean μ, variance σ^2, and unbiased empirical variances

$$\hat{s}_t^2 = \frac{1}{t-1} \sum_{i=1}^{t} \left(X_i - \frac{1}{t} \sum_{t'=1}^{t} X_{t'} \right)^2 \tag{35}$$

corresponding to the first t variables, and also $\hat{s}_t = \sqrt{\hat{s}_t^2}$ $(2 \leq t \leq n)$.

The upper confidence bounds $B_{k,t}$ used in the MC-UCB algorithm is motivated by the following theorem of Maurer and Pontil (2009) (see also the paper of Audibert et al., 2009, for a variant), that gives a high probability bound on the estimation error of $\hat{s}_t$:

Theorem 12 (Theorem 10 of Maurer and Pontil, 2009) *If $\forall t \leq n$, $X_t \in [a, a+c]$, then for $0 < \delta \leq 2$, with probability at least $1-\delta$*

$$|\hat{s}_n - \sigma| \leq c \sqrt{\frac{2\log(2/\delta)}{n-1}}.$$

Using the union bound and $t/(t-1) \leq 2$ for $t \geq 2$ this implies the following uniform bound:

Corollary 13 *If $\forall t \leq n$, $X_t \in [a, a+c]$, then for $0 < \delta \leq 2$, the event*

$$\bigcap_{2 \leq t \leq n} \left\{ |\hat{s}_t - \sigma| \leq 2c\sqrt{\frac{\log(2/\delta)}{t}} \right\}.$$

has probability at least $1 - n\delta$.

We extend this result to sub-Gaussian random variables:

Proposition 14 *Let the distribution of X_t's satisfy (9) of Assumption 1 with $c_1 > 0$, $c_2 \geq 1$, and any $\epsilon > 0$. Define the following event for any $0 < \delta < 1/e$*

$$\xi_n(\delta) = \bigcap_{2 \leq t \leq n} \left\{ |\hat{s}_t - \sigma| \leq \frac{2\beta}{\sqrt{t}} \right\},$$

where β is given by (10). Then $\mathbb{P}(\xi_n(\delta)) \geq (1 - n\delta)^2$.

Proof of Proposition 14 *Step 1. Truncating sub-Gaussian variables.* Let the conditional variance of X_t be $\tilde{\sigma}^2 \stackrel{\text{def}}{=} \mathrm{Var}[X_t | (X_t - \mu)^2 \leq c_1 \log(c_2/\delta)]$. We characterize $\tilde{\sigma}$ by the following lemma (proven in Appendix F):

Lemma 15 *Let X have a distribution with $\mu \stackrel{\text{def}}{=} \mathbb{E}X$ and $\sigma^2 \stackrel{\text{def}}{=} \mathrm{Var}\, X$ satisfying (9) of Assumption 1 with $c_1 > 0$, $c_2 \geq 1$, and any $\epsilon > 0$. Let $0 < \delta < 1/e$, $A \stackrel{\text{def}}{=} \{|X - \mu|^2 \leq c_1 \log(c_2/\delta)\}$, and $\tilde{\sigma}^2 \stackrel{\text{def}}{=} \mathrm{Var}[X|A]$. Then $\mathbb{P}(A^C) \leq \delta$ and*

$$0 \leq \sigma - \tilde{\sigma} \leq \frac{\sqrt{c_1 \delta \log(ec_2/\delta)}}{1 - \delta}.$$

Step 2. Application of tail inequalities. Define the event

$$\xi_1 = \xi_{1,n}(\delta) = \bigcap_{1 \leq t \leq n} \left\{ |X_t - \mu|^2 \leq c_1 \log(c_2/\delta) \right\}.$$

We have that $\mathbb{P}(\xi_1^C) \leq n\delta$ using the union bound and (9). Given ξ_1, $(X_t)_{1 \leq t \leq n}$ are n i.i.d. bounded random variables with common conditional variance $\tilde{\sigma}^2$.

Now let $\xi_2 = \xi_{2,n}(\delta)$ be the event:

$$\xi_2 = \bigcap_{2 \leq t \leq n} \left\{ |\hat{s}_t - \tilde{\sigma}| \leq 4\sqrt{c_1 \log(c_2/\delta) \frac{\log(2/\delta)}{t}} \right\}.$$

Using Corollary 13, we deduce that $\mathbb{P}(\xi_2 | \xi_1) \geq 1 - n\delta$, and thus

$$\mathbb{P}(\xi_1 \cap \xi_2) = \mathbb{P}(\xi_2 | \xi_1)\mathbb{P}(\xi_1) \geq (1 - n\delta)^2.$$

Moreover, from Lemma 15, we have $0 \leq \sigma - \tilde{\sigma} \leq \frac{\sqrt{c_1 \delta \log(ec_2/\delta)}}{1 - \delta}$, and thus on ξ_2, for all $2 \leq t \leq n$:

$$|\hat{s}_t - \sigma| \leq |\hat{s}_t - \tilde{\sigma}| + |\tilde{\sigma} - \sigma| \leq 4\sqrt{c_1 \log(c_2/\delta) \frac{\log(2/\delta)}{t}} + \frac{\sqrt{c_1 \delta \log(ec_2/\delta)}}{1 - \delta} \leq \frac{2\beta}{\sqrt{t}},$$

implying $\xi_2 \subseteq \xi_n(\delta)$. From this, we deduce

$$\mathbb{P}(\xi_n(\delta)) \geq \mathbb{P}(\xi_2) \geq \mathbb{P}(\xi_1 \cap \xi_2) \geq (1 - n\delta)^2$$

proving the proposition. ■

Corollary 16 *Let $n \geq 2$. Let Assumption 1 hold with $c_1 > 0$, $c_2 \geq 1$, and any $\epsilon > 0$. For any $0 < \delta < 1/e$ and for event ξ defined by (13), $\mathbb{P}(\xi) \geq (1 - n\delta)^{2K} \geq 1 - 2nK\delta$.*

Proof of Corollary 16 Since for each $1 \leq k \leq K$, Proposition 14 implies that the probability of

$$\bigcap_{2 \leq t \leq n} \left\{ |\hat{s}_{k,t} - \sigma_k| \leq \frac{2\beta}{\sqrt{t}} \right\}$$

is at least $(1 - n\delta)^2$, the intersection of these independent events, ξ, has probability at least $(1 - n\delta)^{2K}$. The last inequality comes from the convexity of $(1 - x)^{2K}$. ■

B.2 $T_{k,t}$ is Stopping Time, Wald's Identity for the Variance of the Sum of $T_{k,t}$ Centered Samples of One Arm

For a given k, let $(\mathcal{F}_t^{(k)})_{t \leq n}$ be the filtration associated to the process $(X_{k,t})_{t \leq n}$, and $\mathcal{E}_{-k} = \mathcal{E}_{-k,n}$ be the σ-algebra generated by $(X_{k',t'})_{t' \leq n, k' \neq k}$ ("environment"). Define the filtration $(\mathcal{G}_t^{(k)})_{t \leq n}$ by

$$\mathcal{G}_t^{(k)} = \mathcal{G}_t^{(k,n)} \stackrel{\text{def}}{=} \sigma(\mathcal{F}_t^{(k)}, \mathcal{E}_{-k}).$$

$T_{k,t}$ is a stopping time. We prove the following proposition.

Proposition 17 *For each $1 \leq n' \leq n$, $T_{k,n'}$ is a stopping time w.r.t. $(\mathcal{G}_t^{(k)})_{t \leq n}$.*

Proof We prove the statement for fixed budget n by induction for $n' = 1, \ldots, n$.

For $n' \leq 2K$ (initialization), $T_{k,n'}$ is deterministic, so for any t, $\{T_{k,n'} \leq t\}$ is either the empty set or the whole probability space (and is thus measurable according to $\mathcal{G}_t^{(k)}$).

Let us now assume that for a given time step $2K \leq n' < n$, and for any t, $\{T_{k,n'} \leq t\}$ is $\mathcal{G}_t^{(k)}$-measurable. We consider now time step $n' + 1$. Note first that for $t = 0$, $\{T_{k,n'+1} \leq t\} = \{T_{k,n'+1} \leq 0\}$ is the empty set and is thus $\mathcal{G}_t^{(k)}$-measurable. If $t > 0$, then

$$\{T_{k,n'+1} \leq t\} = \big(\{T_{k,n'} = t\} \cap \{k_{n'+1} \neq k\}\big) \cup \{T_{k,n'} \leq t - 1\}. \tag{36}$$

By induction assumption, $\{T_{k,n'} = t\}$ and $\{T_{k,n'} \leq t - 1\}$ are $\mathcal{G}_t^{(k)}$-measurable (since *for any t'*, $\{T_{k,n'} \leq t'\}$ is $\mathcal{G}_{t'}^{(k)}$-measurable). On $\{T_{k,n'} = t\}$, $k_{n'+1}$ is also $\mathcal{G}_t^{(k)}$-measurable since it is determined only by the values of the upper bounds $\{B_{q,n'+1}\}_{1 \leq q \leq K}$ (which depend only on $\{X_{k',t'}\}_{t' \leq n, k' \neq k}$ and on $(X_{k,1}, \ldots, X_{k,t})$). Hence, $\{T_{k,n'} = t\} \cap \{k_{n'+1} \neq k\}$ is $\mathcal{G}_t^{(k)}$-measurable, and thus using (36), we have that $\{T_{k,n'+1} \leq t\}$ is $\mathcal{G}_t^{(k)}$-measurable, as well.

We have thus proved by induction that $T_{k,n'}$ is a stopping time w.r.t. the filtration $(\mathcal{G}_t^{(k)})_{t\le n}$. ■

Wald's second identity for the variance. We also need to express the variance of the sum of random number of centered terms when this random number is a stopping time. Thus, we recall the following theorem from Athreya and Lahiri (2006) (this variant is quoted from Lemma 10 of Antos et al. (2010))

Proposition 18 (Theorem 13.2.14 of Athreya and Lahiri (2006)) *Let $(\mathcal{F}_t)_{t=1,\dots,n}$ be a filtration and $(X_t)_{t=1,\dots,n}$ be an $\mathcal{F}_t$ adapted sequence of i.i.d. random variables with finite expectation μ and variance σ^2. Assume that $\mathcal{F}_t$ and $\sigma(\{X_s : s \ge t+1\})$ are independent for any $t \le n$, and let $T(\le n)$ be a stopping time w.r.t. $\mathcal{F}_t$. Then*

$$\mathbb{E}\left[\Big(\sum_{t=1}^{T}(X_t-\mu)\Big)^2\right] = \mathbb{E}[T]\sigma^2.$$

Application to arm k and samples $(X_{k,t})_{t\le n}$.

Corollary 19 *For any $1 \le k \le K$ and $n' \le n$,*

$$\mathbb{E}\left[\Big(\sum_{t=1}^{T_{k,n'}}(X_{k,t}-\mu_k)\Big)^2\right] = \mathbb{E}[T_{k,n'}]\sigma_k^2.$$

Proof Proposition 17, the fact that $G_t^{(k)}$ and $\sigma(\{X_{k,s} : s \ge t+1\})$ are independent for any $t \le n$, and $T_{k,n'} \le n$ guarantee that we can apply Proposition 18 with filtration $(\mathcal{G}_t^{(k)})_{t\le n}$, $(X_{k,t})_{t\le n}$, and $T_{k,n'}$ leading to the equality. ■

B.3 Other Technical Inequalities

Now we state and prove some further technical inequalities.

Bounds on the loss and the variance of the sum of the centered samples of one arm on event ξ^C.

Lemma 20 *Let $n \ge 2$ and $0 < \delta < 1/e$. Let Assumption 1 hold with $c_2 \ge \max(1, 2nK\delta)$. Then for each arm k,*

$$\mathbb{E}\big[|\hat{\mu}_{k,n}-\mu_k|^2\mathbb{I}\{\xi^C\}\big] \le Kn^2\delta C_\xi(\delta) \qquad \text{and}$$
$$\mathbb{E}\left[\Big(\sum_{t=1}^{T_{k,n}}(X_{k,t}-\mu_k)\Big)^2\mathbb{I}\{\xi^C\}\right] \le 2Kn^3\delta C_\xi(\delta),$$

where $C_\xi(\delta) = C_{\xi,n}(\delta) \stackrel{\text{def}}{=} c_1 \log(ec_2/2nK\delta)$. *Consequently, for every arms* k *and* q,

$$\left|\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{q,n} - \mu_q)\mathbb{I}\{\xi^C\}\big]\right| \le Kn^2\delta C_\xi(\delta) \qquad \text{and}$$
$$\left|\mathbb{E}\left[\Big(\sum_{t=1}^{T_{k,n}}(X_{k,t} - \mu_k)\Big)\Big(\sum_{t=1}^{T_{q,n}}(X_{q,t} - \mu_q)\Big)\mathbb{I}\{\xi^C\}\right]\right| \le 2Kn^3\delta C_\xi(\delta).$$

Proof of Lemma 20 $c_2 \ge 1$ and Corollary 16 imply $\mathbb{P}(\xi^C) \le 2nK\delta$. Due to this, $c_2 \ge 2nK\delta$, and Assumption 1, for any $1 \le k \le K$ and $1 \le t \le n$, Lemma 11 implies

$$\mathbb{E}\left[(X_{k,t} - \mu_k)^2\mathbb{I}\{\xi^C\}\right] \le 2nK\delta c_1 \log(ec_2/2nK\delta) = 2Kn\delta C_\xi(\delta).$$

The first claim follows from the fact that

$$\begin{aligned}\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)^2\mathbb{I}\{\xi^C\}\big] &\le \mathbb{E}\left[\frac{\sum_{t=1}^{T_{k,n}}(X_{k,t} - \mu_k)^2}{T_{k,n}}\mathbb{I}\{\xi^C\}\right] \\ &\le \sum_{t=1}^{n}\frac{\mathbb{E}\big[(X_{k,t} - \mu_k)^2\mathbb{I}\{\xi^C\}\big]}{2} \le Kn^2\delta C_\xi(\delta).\end{aligned}$$

The second claim follows from the fact that

$$\left(\sum_{t=1}^{T_{k,n}}(X_{k,t} - \mu_k)\right)^2 \le \left(\sum_{t=1}^{n}|X_{k,t} - \mu_k|\right)^2 \le n\sum_{t=1}^{n}(X_{k,t} - \mu_k)^2,$$

and so

$$\mathbb{E}\left[\left(\sum_{t=1}^{T_{k,n}}(X_{k,t} - \mu_k)\right)^2\mathbb{I}\{\xi^C\}\right] \le n\sum_{t=1}^{n}\mathbb{E}\big[(X_{k,t} - \mu_k)^2\mathbb{I}\{\xi^C\}\big] \le 2Kn^3\delta C_\xi(\delta).$$

The third claim follows from the first one by Cauchy-Schwarzs inequality

$$\left|\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{q,n} - \mu_q)\mathbb{I}\{\xi^C\}\big]\right| \le \sqrt{\mathbb{E}[(\hat{\mu}_{k,n} - \mu_k)^2\mathbb{I}\{\xi^C\}]}\sqrt{\mathbb{E}[(\hat{\mu}_{q,n} - \mu_q)^2\mathbb{I}\{\xi^C\}]},$$

and the fourth one follows from the second one, analogously. ■

We get the following corollary by substituting $\delta = n^{-9/2}$:

Corollary 21 *Let* $n \ge K \ge 2$. *Let Assumption 1 hold with* $c_2 \ge 1$. *Then for each arm* k,

$$\mathbb{E}\big[|\hat{\mu}_{k,n} - \mu_k|^2\mathbb{I}\{\xi^C\}\big] \le \frac{KC_\xi}{n^{5/2}} \qquad \text{and}$$
$$\mathbb{E}\left[\Big(\sum_{t=1}^{T_{k,n}}(X_{k,t} - \mu_k)\Big)^2\mathbb{I}\{\xi^C\}\right] \le \frac{2KC_\xi}{n^{3/2}}.$$

where $C_\xi = C_\xi(n^{-9/2}) = c_1 \log(ec_2 n^{7/2}/2K)$ *as in* (19). *Consequently, for every arms* k *and* q,

$$\left|\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{q,n} - \mu_q)\mathbb{I}\{\xi^C\}\big]\right| \le \frac{KC_\xi}{n^{5/2}} \qquad \text{and}$$
$$\left|\mathbb{E}\left[\Big(\sum_{t=1}^{T_{k,n}}(X_{k,t} - \mu_k)\Big)\Big(\sum_{t=1}^{T_{q,n}}(X_{q,t} - \mu_q)\Big)\mathbb{I}\{\xi^C\}\right]\right| \le \frac{2KC_\xi}{n^{3/2}}.$$

Upper and lower bound on β_n *of* (11) *for* $\delta = n^{-9/2}$. Using $n \ge 4K \ge 8$, $c_2 \ge 1$, and monotonicity in n we have

$$\begin{aligned}
\beta_n &= \sqrt{c_1 \log(c_2^2 n^9)\log(4n^9)} + \frac{\sqrt{c_1 \log(ec_2 n^{4.5})}}{2(1-n^{-4.5})n^{7/4}} \\
&\le \sqrt{c_1}\frac{\log(c_2^2 n^9) + \log(4n^9)}{2} + \frac{\sqrt{c_1 \log(ec_2 8^{4.5})}}{2(1-8^{-4.5})8^{7/4}} \\
&\le \sqrt{c_1}\left(9\log n + \log(4c_2^2)/2 + \frac{\log(e^2 c_2^2 8^9)}{2^{5/4}(8^2 - 8^{-2.5})\sqrt{\log(e8^{4.5})}}\right).
\end{aligned}$$

Using

$$\log(e^2 c_2^2 8^9) \le \log(e^2 c_2^2 (c_2+1)^{27}) \le 29\log(c_2+1) + 2\log(c_2+1)/\log 2 \le 32\log(c_2+1)$$

and $4c_2^2 \le (c_2+1)^3$

$$\beta_n \le \sqrt{c_1}\left(9\log n + 1.5\log(c_2+1) + \frac{32\log(c_2+1)}{489}\right) \le \sqrt{c_1}(9\log n + 1.6\log(c_2+1)) = C_\beta$$

recalling (18). On the other hand, keeping only the first term of β_n

$$\beta_n \ge \sqrt{c_1 \log(c_2^2 n^9)\log(4n^9)} \ge \sqrt{c_1 \log(8^9 c_2^2) 29\log 2} \ge \sqrt{58 c_1 \log 2 \log(ec_2)} \ge \sqrt{40 c_1 \log(ec_2)}.$$

Upper bound on γ_n *of Lemma 2 when* $\delta = n^{-9/2}$. If Assumption 1 is satisfied with $c_2 \ge 1$ then Lemma 11 implies $\sigma_k^2 \le c_1 \log(ec_2)$ for any $1 \le k \le K$, thus recalling $\bar{\Sigma} = \max_p \sigma_p$ we have $\Sigma_w \le \bar{\Sigma} \le \sqrt{c_1 \log(ec_2)}$. For $\delta = n^{-9/2}$, the lower bound above on β_n leads to $\bar{\Sigma}/\beta_n \le 1/\sqrt{40}$ and

$$\gamma_n = (\bar{\Sigma}/\beta_n + \sqrt{8})^{1/3} \le (1/\sqrt{40} + \sqrt{8})^{1/3} < 1.5.$$

Appendix C. Proof of Proposition 8 and 9

In this section, we use Lemmata 1 and 2 to prove Proposition 8 and 9, respectively.

C.1 Proof of Proposition 8

Proof of Proposition 8 By definition, the pseudo-loss of the algorithm is

$$\begin{aligned}\widetilde{L}_n(\mathcal{A}_{\text{MC-UCB}}) &= \sum_{k=1}^K w_k^2\sigma_k^2\mathbb{E}\Big[\frac{\mathbb{I}\{\xi\}}{T_{k,n}}\Big] + \sum_{k=1}^K w_k^2\sigma_k^2\mathbb{E}\Big[\frac{\mathbb{I}\{\xi^C\}}{T_{k,n}}\Big] \\ &\le \sum_{k=1}^K \frac{w_k^2\sigma_k^2\mathbb{P}(\xi)}{\inf_{\omega\in\xi} T_{k,n}(\omega)} + \sum_{k=1}^K w_k^2\sigma_k^2\frac{\mathbb{P}(\xi^C)}{2}, \end{aligned} \tag{37}$$

because $T_{k,n} \ge 2$ by the definition of $\mathcal{A}_{\text{MC-UCB}}$. Recalling (14) from Lemma 1 that upper bounds $w_k\sigma_k/T_{k,n}$ on ξ, we obtain

$$\sum_{k=1}^K \frac{w_k^2\sigma_k^2\mathbb{P}(\xi)}{\inf_\xi T_{k,n}} \le \sum_{k=1}^K w_k\sigma_k\Big(\frac{\Sigma_w}{n} + \frac{12\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w}{n^2}\Big) = \frac{\Sigma_w^2}{n} + \frac{12\Sigma_w\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w^2}{n^2}$$

using $\sum_k w_k\sigma_k = \Sigma_w$. Finally, using (37) and the previous inequality and recalling $\mathbb{P}(\xi^C) \le 2nK\delta$ from Corollary 16, $\delta = n^{-9/2}$, and $\beta_n \le C_\beta$ from Appendix B.3, we have

$$\begin{aligned}\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}}) &= \widetilde{L}_n(\mathcal{A}_{\text{MC-UCB}}) - \frac{\Sigma_w^2}{n} \\ &\le \frac{12\Sigma_w\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w^2}{n^2} + nK\delta\sum_{k=1}^K w_k^2\sigma_k^2 \\ &\le \frac{12\Sigma_w C_\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w^2}{n^2} + \frac{K\Sigma_w^2}{n^{7/2}} \\ &\le \frac{12\Sigma_w C_\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{(4K+\sqrt{2}/16)\Sigma_w^2}{n^2}, \end{aligned}$$

that concludes the proof. ■

C.2 Proof of Proposition 9

Proof of Proposition 9 We decompose and bound $\widetilde{L}_n(\mathcal{A}_{\text{MC-UCB}})$ on ξ and ξ^C again as in (37). Recalling (17) from Lemma 2 that upper bounds $w_k\sigma_k/T_{k,n}$ on ξ, we obtain

$$\sum_{k=1}^K \frac{w_k^2\sigma_k^2\mathbb{P}(\xi)}{\inf_\xi T_{k,n}} \le \sum_{k=1}^K w_k\sigma_k\Big(\frac{\Sigma_w}{n} + \frac{12K^{1/3}\beta_n\gamma_n}{n^{4/3}} + \frac{4K\Sigma_w}{n^2}\Big) = \frac{\Sigma_w^2}{n} + \frac{12K^{1/3}\Sigma_w\beta_n\gamma_n}{n^{4/3}} + \frac{4K\Sigma_w^2}{n^2}$$

using $\sum_k w_k\sigma_k = \Sigma_w$. Finally, using (37) and the previous inequality and recalling $\mathbb{P}(\xi^C) \leq 2nK\delta$ from Corollary 16, $\delta = n^{-9/2}$, $\beta_n \leq C_\beta$, and $\gamma_n < 1.5$ from Appendix B.3, we have

$$\begin{aligned}
\widetilde{R}_n(\mathcal{A}_{\text{MC-UCB}}) &= \widetilde{L}_n(\mathcal{A}_{\text{MC-UCB}}) - \frac{\Sigma_w^2}{n} \\
&\leq \frac{12K^{1/3}\Sigma_w\beta_n\gamma_n}{n^{4/3}} + \frac{4K\Sigma_w^2}{n^2} + nK\delta\sum_{k=1}^K w_k^2\sigma_k^2 \\
&\leq \frac{18K^{1/3}\Sigma_w C_\beta}{n^{4/3}} + \frac{4K\Sigma_w^2}{n^2} + \frac{K\Sigma_w^2}{n^{7/2}} \\
&\leq \frac{18K^{1/3}\Sigma_w C_\beta}{n^{4/3}} + \frac{(4K+\sqrt{2}/16)\Sigma_w^2}{n^2},
\end{aligned}$$

that concludes the proof. ■

Appendix D. Bounds on $R_n(\mathcal{A}_{\text{MC-UCB}})$

This section contains the proofs of the regret bounds for $\mathcal{A}_{\text{MC-UCB}}$.

D.1 Problem-Dependent Bound

Proof of Proposition 3 By definition, we have

$$L_n(\mathcal{A}_{\text{MC-UCB}}) = \sum_{k=1}^K w_k^2\mathbb{E}\Big[(\hat{\mu}_{k,n}-\mu_k)^2\mathbb{I}\{\xi\}\Big] + \sum_{k=1}^K w_k^2\mathbb{E}\Big[(\hat{\mu}_{k,n}-\mu_k)^2\mathbb{I}\{\xi^C\}\Big]. \tag{38}$$

Using the definition of $\hat{\mu}_{k,n}$, we have

$$(\hat{\mu}_{k,n}-\mu_k)^2\mathbb{I}\{\xi\} \leq \frac{\big(\sum_{t=1}^{T_{k,n}}(X_{k,t}-\mu_k)\big)^2}{\inf_{\omega\in\xi} T_{k,n}^2(\omega)}\mathbb{I}\{\xi\} \leq \frac{\big(\sum_{t=1}^{T_{k,n}}(X_{k,t}-\mu_k)\big)^2}{\inf_{\xi} T_{k,n}^2}.$$

Taking expectation and using Corollary 19

$$\mathbb{E}\big[(\hat{\mu}_{k,n}-\mu_k)^2\mathbb{I}\{\xi\}\big] \leq \frac{\mathbb{E}\Big[\big(\sum_{t=1}^{T_{k,n}}(X_{k,t}-\mu_k)\big)^2\Big]}{\inf_{\xi} T_{k,n}^2} = \frac{\mathbb{E}[T_{k,n}]\sigma_k^2}{\inf_{\xi} T_{k,n}^2},$$

so we bound the first sum of (38) as

$$\sum_{k=1}^K w_k^2\mathbb{E}\big[(\hat{\mu}_{k,n}-\mu_k)^2\mathbb{I}\{\xi\}\big] \leq \sum_{k=1}^K w_k^2\frac{\sigma_k^2\mathbb{E}[T_{k,n}]}{\inf_{\xi} T_{k,n}^2}. \tag{39}$$

Recalling (14) from Lemma 1 that upper bounds $w_k\sigma_k/T_{k,n}$ on ξ, we obtain

$$\sum_{k=1}^K w_k^2\frac{\sigma_k^2\mathbb{E}[T_{k,n}]}{\inf_{\xi} T_{k,n}^2} \leq \sum_{k=1}^K \Big(\frac{\Sigma_w}{n} + \frac{12\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w}{n^2}\Big)^2\mathbb{E}[T_{k,n}]. \tag{40}$$

Since $\sum_k T_{k,n} = n$, we have $\sum_k \mathbb{E}[T_{k,n}] = n$, (40) can be rewritten as

$$\begin{aligned}\sum_{k=1}^{K} w_k^2 \frac{\sigma_k^2 \mathbb{E}[T_{k,n}]}{\inf_\xi T_{k,n}^2} &\leq \Big(\frac{\Sigma_w}{n} + \frac{12\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{4K\Sigma_w}{n^2}\Big)^2 n \\ &\leq \Big(\frac{\Sigma_w^2}{n^2} + \frac{24\Sigma_w\beta_n}{n^{5/2}\lambda_{\min}^{3/2}} + \frac{8K\Sigma_w^2}{n^3} + \frac{288\beta_n^2}{n^3\lambda_{\min}^3} + \frac{32K^2\Sigma_w^2}{n^4}\Big) n \\ &\leq \frac{\Sigma_w^2}{n} + \frac{24\Sigma_w\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + 16\frac{K\Sigma_w^2}{n^2} + \frac{288\beta_n^2}{n^2\lambda_{\min}^3}.\end{aligned}$$

Finally, using (38), (39), and the previous inequality and recalling $\delta = n^{-9/2}$, Corollary 21, and $\beta_n \leq C_\beta$ from Appendix B.3 we have

$$\begin{aligned}R_n(\mathcal{A}_{\text{MC-UCB}}) &= L_n(\mathcal{A}_{\text{MC-UCB}}) - \frac{\Sigma_w^2}{n} \\ &\leq \frac{24\Sigma_w\beta_n}{n^{3/2}\lambda_{\min}^{3/2}} + 16\frac{K\Sigma_w^2}{n^2} + \frac{288\beta_n^2}{n^2\lambda_{\min}^3} + \frac{KC_\xi}{n^{5/2}} \\ &\leq \frac{24\Sigma_w C_\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{288C_\beta^2}{n^2\lambda_{\min}^3} + 16\frac{K\Sigma_w^2}{n^2} + \frac{\sqrt{K}C_\xi}{2n^2} \\ &\leq \frac{24\Sigma_w C_\beta}{n^{3/2}\lambda_{\min}^{3/2}} + \frac{288C_\beta^2}{n^2\lambda_{\min}^3} + \frac{\sqrt{K}C_\xi + 32K\Sigma_w^2}{2n^2}.\end{aligned}$$

This concludes the proof. ■

D.2 Problem-Independent Bound

Proof of Proposition 4 Again, we decompose $L_n(\mathcal{A}_{\text{MC-UCB}})$ on ξ and ξ^C as in (38), and bound it on ξ as in (39). Recalling (17) from Lemma 2 that upper bounds $w_k\sigma_k/T_{k,n}$ on ξ, we obtain

$$\sum_{k=1}^{K} w_k^2 \frac{\sigma_k^2 \mathbb{E}[T_{k,n}]}{\inf_\xi T_{k,n}^2} \leq \sum_{k=1}^{K} \Big(\frac{\Sigma_w}{n} + \frac{12K^{1/3}\beta_n\gamma_n}{n^{4/3}} + \frac{4K\Sigma_w}{n^2}\Big)^2 \mathbb{E}[T_{k,n}]. \tag{41}$$

Since $\sum_k T_{k,n} = n$, we have $\sum_k \mathbb{E}[T_{k,n}] = n$, (41) can be rewritten as

$$\begin{aligned}\sum_{k=1}^{K} w_k^2 \frac{\sigma_k^2 \mathbb{E}[T_{k,n}]}{\inf_\xi T_{k,n}^2} &\leq \Big(\frac{\Sigma_w}{n} + \frac{12K^{1/3}\beta_n\gamma_n}{n^{4/3}} + \frac{4K\Sigma_w}{n^2}\Big)^2 n \\ &\leq \Big(\frac{\Sigma_w^2}{n^2} + \frac{24K^{1/3}\Sigma_w}{n^{7/3}}\beta_n\gamma_n + \frac{8K\Sigma_w^2}{n^3} + \frac{288K^{2/3}}{n^{8/3}}\beta_n^2\gamma_n^2 + \frac{32K^2\Sigma_w^2}{n^4}\Big) n \\ &\leq \frac{\Sigma_w^2}{n} + \frac{24K^{1/3}\Sigma_w}{n^{4/3}}\beta_n\gamma_n + \frac{288K^{2/3}}{n^{5/3}}\beta_n^2\gamma_n^2 + \frac{16K\Sigma_w^2}{n^2}.\end{aligned}$$

Finally, using (38), (39), and the previous inequality and recalling $\delta = n^{-9/2}$, Corollary 21, $\beta_n \le C_\beta$, and $\gamma_n < 1.5$ from Appendix B.3 we have

$$\begin{aligned} R_n(\mathcal{A}_{\text{MC-UCB}}) &= L_n(\mathcal{A}_{\text{MC-UCB}}) - \frac{\Sigma_w^2}{n} \\ &\le \frac{24K^{1/3}\Sigma_w}{n^{4/3}}\beta_n\gamma_n + \frac{288K^{2/3}}{n^{5/3}}\beta_n^2\gamma_n^2 + \frac{16K\Sigma_w^2}{n^2} + \frac{KC_\xi}{n^{5/2}} \\ &\le \frac{36K^{1/3}\Sigma_w C_\beta}{n^{4/3}} + \frac{648K^{2/3}C_\beta^2}{n^{5/3}} + \frac{\sqrt{K}C_\xi + 32K\Sigma_w^2}{2n^2} \\ &\le \frac{36K^{1/3}\Sigma_w C_\beta}{n^{4/3}} + \frac{K^{2/3}(2058C_\beta^2 + 32\Sigma_w^2) + K^{1/6}C_\xi}{(2n)^{5/3}}. \end{aligned}$$

This concludes the proof. ■

Remark 22 *Observe that in the proof of Proposition 8 and 9, we already bounded a linear combination of $\mathbb{E}[\mathbb{I}\{\xi\}/T_{k,n}]$ (leading to the desired rates), that is clearly upper bounded also by $\mathbb{E}[T_{k,n}]/\inf_\xi T_{k,n}^2$ appearing in both proofs above. Unfortunately, a reverse inequality does not directly hold, thus here we had to proceed in a more involved way leading to looser constants. If one could derive such a reverse inequality and then use the bounds on $\widetilde{R}_n(\mathcal{A}_{MC\text{-}UCB})$, that might give sharper constants also in the bounds on $R_n(\mathcal{A}_{MC\text{-}UCB})$.*

Appendix E. Bounds on the Cross Product-Terms

In this appendix, we prove Proposition 5 and 6 stating that the cross product-terms in (2) are 0 for symmetric distributions and decrease at polynomial rate in n in the general sub-Gaussian case.

E.1 Vanishing of the Terms for Symmetric Arm Distributions

Proof of Proposition 5

Step 1: Conditioning on a pair of numbers of pulls. Recall that $(\hat{s}_{k,t})_{k\le K, t\le n}$ are the unbiased empirical variances (see Equation 12). At each time step $t > 2K$, $\mathcal{A}_{\text{MC-UCB}}$ chooses k_t based on the values of $(B_{p,t})_{p\le K}$, which depend on $\{T_{p,t-1}\}_{p\le K}$ and $\{\hat{\sigma}_{p,t-1}\}_{p\le K}$. Thus $\{T_{p,t}\}_{p\le K}$ is a deterministic map of $\{T_{p,t-1}\}_{p\le K}$ and $\{\hat{\sigma}_{p,t-1}\}_{p\le K}$. Hence, by induction, each $T_{k,n}$ is a deterministic function of $\{\hat{\sigma}_{p,t}\}_{p\le K, t<n}$, and so of $\{\hat{s}_{p,t}\}_{p\le K, t\le n}$, as well.

Now fix arms k,k' and $1 \le s$,$s' \le n$ such that $\mathbb{P}(T_{k,n} = s, T_{k',n} = s') > 0$. Then we have

$$\begin{aligned} &\mathbb{E}\big[(\hat{\mu}_{k,n} - \mu_k)(\hat{\mu}_{k',n} - \mu_{k'})\big|T_{k,n} = s, T_{k',n} = s'\big] \\ &= \mathbb{E}\bigg[\Big(\frac{1}{s}\sum_{t=1}^{s} X_{k,t} - \mu_k\Big)\Big(\frac{1}{s'}\sum_{t=1}^{s'} X_{k',t} - \mu_{k'}\Big)\bigg|T_{k,n} = s, T_{k',n} = s'\bigg] \qquad (42) \\ &= \mathbb{E}\bigg[\mathbb{E}\Big[\Big(\frac{1}{s}\sum_{t=1}^{s} X_{k,t} - \mu_k\Big)\Big(\frac{1}{s'}\sum_{t=1}^{s'} X_{k',t} - \mu_{k'}\Big)\Big|\{\hat{s}_{p,t}\}_{p\le K, t\le n}\Big]\bigg|T_{k,n} = s, T_{k',n} = s'\bigg]. \end{aligned}$$

Since the full sample sequences of the individual arms are independent, the sequences $(X_{k,1},\dots,X_{k,s})$ and $(X_{k',1},\dots,X_{k',s'})$ remain conditionally independent conditioning on $\{\hat{s}_{p,t}\}_{p\le K,t\le n}$. This leads to:

$$\begin{aligned}&\mathbb{E}\Big[\Big(\frac{1}{s}\sum_{t=1}^{s}X_{k,t}-\mu_k\Big)\Big(\frac{1}{s'}\sum_{t=1}^{s'}X_{k',t}-\mu_{k'}\Big)\Big|\{\hat{s}_{p,t}\}_{p\le K,t\le n}\Big] \\ &=\mathbb{E}\Big[\frac{1}{s}\sum_{t=1}^{s}X_{k,t}-\mu_k\Big|\{\hat{s}_{p,t}\}_{p\le K,t\le n}\Big]\mathbb{E}\Big[\frac{1}{s'}\sum_{t=1}^{s'}X_{k',t}-\mu_{k'}\Big|\{\hat{s}_{p,t}\}_{p\le K,t\le n}\Big].\end{aligned} \tag{43}$$

Step 2: For any $k\le K$ and $s\le n$, $\mathbb{E}\big[\frac{1}{s}\sum_{t=1}^{s}X_{k,t}-\mu_k\big|\{\hat{s}_{p,t}\}_{p\le K,t\le n}\big]=0$ a.s. We first state the following Lemma proven in Appendix F:

Lemma 23 *Let ν be a symmetric distribution on $\mathbb{R}$ around 0, $X=(X_1,\dots,X_n)$ be generated in an i.i.d. way according to ν, and $\hat{s}_2$,...,$\hat{s}_n$ are the unbiased empirical standard deviations given by* (35)*. Then for $1\le t\le n$, $\mathbb{E}[X_t|\{\hat{s}_{t'}\}_{t'\le n}]=0$ a.s.*

As ν_k is symmetric, Lemma 23 applies to $X=(X_{k,1}-\mu_k,\dots,X_{k,n}-\mu_k)$ and $\{\hat{s}_{k,t}\}_{t\le n}$, that is,

$$\mathbb{E}\Big[\frac{1}{s}\sum_{t=1}^{s}X_{k,t}-\mu_k\Big|\{\hat{s}_{k,t}\}_{t\le n}\Big]=\frac{1}{s}\sum_{t=1}^{s}\mathbb{E}[X_{k,t}-\mu_k|\{\hat{s}_{k,t'}\}_{t'\le n}]=0\qquad\text{a.s.}$$

By definition, $\{\hat{s}_{p,t}\}_{p\neq k,t\le n}$ is independent of $(X_{k,1},\dots,X_{k,s},\{\hat{s}_{k,t}\}_{t\le n})$, hence

$$\mathbb{E}\Big[\frac{1}{s}\sum_{t=1}^{s}X_{k,t}-\mu_k\Big|\{\hat{s}_{p,t}\}_{p\le K,t\le n}\Big]=\mathbb{E}\Big[\frac{1}{s}\sum_{t=1}^{s}X_{k,t}-\mu_k\Big|\{\hat{s}_{k,t}\}_{t\le n}\Big]=0\quad\text{a.s.} \tag{44}$$

Step 3: The cross product-terms $\mathbb{E}[(\hat{\mu}_{k,n}-\mu_k)(\hat{\mu}_{k',n}-\mu_{k'})]=0$. We combine (42), (43), and (44) to get in case of $\mathbb{P}(T_{k,n}=s,T_{k',n}=s')>0$

$$\mathbb{E}[(\hat{\mu}_{k,n}-\mu_k)(\hat{\mu}_{k',n}-\mu_{k'})|T_{k,n}=s,T_{k',n}=s']=\mathbb{E}[0\cdot 0|T_{k,n}=s,T_{k',n}=s']=0.$$

Conditioning on $\{T_{k,n}=s,T_{k',n}=s'\}$ and using the equation above

$$\begin{aligned}&\mathbb{E}\Big[(\hat{\mu}_{k,n}-\mu_k)(\hat{\mu}_{k',n}-\mu_{k'})\Big] \\ &=\sum_{s=2}^{n}\sum_{s'=2}^{n}\mathbb{E}\Big[(\hat{\mu}_{k,n}-\mu_k)(\hat{\mu}_{k',n}-\mu_{k'})|T_{k,n}=s,T_{k',n}=s'\Big]\mathbb{P}\big(T_{k,n}=s,T_{k',n}=s'\big)=0.\end{aligned}$$

Taking the weighted sum over k and k' concludes the proof. ■

E.2 Bounds on the Terms for General Arm Distributions

The following lemma proven in Appendix F will be used for the proof:

Lemma 24 *Let X be a random variable. Let $(\Omega_u)_{u=1,\dots,p}$ be a partition of an event Ω' of the probability space. Let $a_u \in \mathbb{R}$ for $u = 1, \dots, p$, and $\underline{a} = \min_{1\le u\le p} a_u$, $\bar{a} = \max_{1\le u\le p} a_u$. We have*

$$\Big|\mathbb{E}\Big[X\sum_{u=1}^{p} a_u \mathbb{I}\{\Omega_u\}\Big]\Big| - \big|\underline{a}\mathbb{E}[X\mathbb{I}\{\Omega'\}]\big| \le \Big|\mathbb{E}\Big[X\sum_{u=1}^{p} a_u \mathbb{I}\{\Omega_u\}\Big] - \underline{a}\mathbb{E}[X\mathbb{I}\{\Omega'\}]\Big| \le (\bar{a}-\underline{a})\mathbb{E}|X\mathbb{I}\{\Omega'\}|.$$

Proof of Proposition 6 For any given $k \neq q$, introduce

$$Z_{kq} \stackrel{\text{def}}{=} \left(\sum_{t=1}^{T_{k,n}} (X_{k,t}-\mu_k)\right)\left(\sum_{t=1}^{T_{q,n}} (X_{q,t}-\mu_q)\right) = T_{k,n}T_{q,n}(\hat{\mu}_{k,n}-\mu_k)(\hat{\mu}_{q,n}-\mu_q).$$

Then it suffices to bound $|w_k w_q \mathbb{E}[Z_{kq}/(T_{k,n}T_{q,n})]|$.

Step 1: $\mathbb{E}[Z_{kq}] = 0$. Let $\mathcal{T}_{k,t} \stackrel{\text{def}}{=} \min\{s \ge 1 : T_{k,s} \ge t\}$, that is, that random time step when $\mathcal{A}_{\text{MC-UCB}}$ pulls arm k the t^{th} time. ($\mathcal{T}_{k,t} = \infty$ if k is not pulled t times.) Now

$$\begin{aligned}
\mathbb{E}[Z_{kq}] &= \mathbb{E}\Big[\Big(\sum_{t=1}^{n}(X_{k,t}-\mu_k)\mathbb{I}\{T_{k,n}\ge t\}\Big)\Big(\sum_{t=1}^{n}(X_{q,t}-\mu_q)\mathbb{I}\{T_{q,n}\ge t\}\Big)\Big] \\
&= \sum_{t=1}^{n}\sum_{t'=1}^{n}\mathbb{E}\big[(X_{k,t}-\mu_k)(X_{q,t'}-\mu_q)\mathbb{I}\big\{T_{k,n}\ge t \wedge T_{q,n}\ge t'\big\}\big] \\
&= \sum_{t=1}^{n}\sum_{t'=1}^{n}\mathbb{E}\big[(X_{k,t}-\mu_k)(X_{q,t'}-\mu_q)\mathbb{I}\big\{T_{k,n}\ge t \wedge T_{q,n}\ge t' \wedge \mathcal{T}_{k,t} < \mathcal{T}_{q,t'}\big\}\big] \\
&\quad + \sum_{t=1}^{n}\sum_{t'=1}^{n}\mathbb{E}\big[(X_{k,t}-\mu_k)(X_{q,t'}-\mu_q)\mathbb{I}\big\{T_{k,n}\ge t \wedge T_{q,n}\ge t' \wedge \mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\big\}\big].
\end{aligned}$$

Fix any $1 \le t, t' \le n$. Proposition 17 implies that $\{T_{k,n} \le t-1\} \in \mathcal{G}_{t-1}^{(k)}$ (defined in Proposition 17), and thus also $\{T_{k,n} \ge t\} \in \mathcal{G}_{t-1}^{(k)}$. $\mathcal{T}_{k,t} > \mathcal{T}_{q,t'}$ means that for some time step $s \ge t'$, $\{T_{q,s} \ge t'\}$, but $\{T_{k,s} < t\}$. Thus,

$$\{\mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\} = \bigcup_{s=t'}^{\infty}\{T_{q,s}\ge t'\}\cap\{T_{k,s}<t\}.$$

Intersecting this by $\{T_{k,n}\ge t\}$ and noting that for $s \ge n$, $\{T_{k,s} < T_{k,n}\} = \emptyset$

$$\{\mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\}\cap\{T_{k,n}\ge t\} = \bigcup_{s=t'}^{n-1}\{T_{q,s}\ge t'\}\cap\{T_{k,s}<t\le T_{k,n}\}.$$

Now, by Proposition 17 for any $s \leq n$, $\{T_{k,s} < t\} = \{T_{k,s} \leq t-1\} \in \mathcal{G}_{t-1}^{(k)}$. Moreover, on $\{T_{k,s} < t\}$, $T_{q,s}$ is $\mathcal{G}_{t-1}^{(k)}$-measurable, thus $\{T_{q,s} \geq t'\} \cap \{T_{k,s} < t\} \in \mathcal{G}_{t-1}^{(k)}$. Hence $\{\mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\} \cap \{T_{k,n} \geq t\} \in \mathcal{G}_{t-1}^{(k)}$, as well. Observe that $T_{k,n} \geq t$ and $\mathcal{T}_{k,t} > \mathcal{T}_{q,t'}$ together imply $T_{q,n} \geq t'$, so $\mathbb{I}\{\mathcal{T}_{k,t} > \mathcal{T}_{q,t'} \wedge T_{k,n} \geq t \wedge T_{q,n} \geq t'\}$ is $\mathcal{G}_{t-1}^{(k)}$-measurable. Also $X_{q,t'}$ is obviously $\mathcal{G}_{t-1}^{(k)}$-measurable, while $X_{k,t}$ is independent of $\mathcal{G}_{t-1}^{(k)}$. Thus, conditioning on $\mathcal{G}_{t-1}^{(k)}$, we have

$$\begin{aligned}
&\mathbb{E}\big[(X_{k,t}-\mu_k)(X_{q,t'}-\mu_q)\mathbb{I}\{T_{k,n} \geq t \wedge T_{q,n} \geq t' \wedge \mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\}\big] \\
&= \mathbb{E}\Big[(X_{q,t'}-\mu_q)\mathbb{I}\{T_{k,n} \geq t \wedge T_{q,n} \geq t' \wedge \mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\}\,\mathbb{E}\big[X_{k,t}-\mu_k|\mathcal{G}_{t-1}^{(k)}\big]\Big] \\
&= \mathbb{E}\big[(X_{q,t'}-\mu_q)\mathbb{I}\{T_{k,n} \geq t \wedge T_{q,n} \geq t' \wedge \mathcal{T}_{k,t} > \mathcal{T}_{q,t'}\}\,0\big] = 0.
\end{aligned}$$

By summing for t,t' and repeating the same reasoning for the other term of $\mathbb{E}[Z_{kq}]$ with arm q, we obtain that $\mathbb{E}[Z_{kq}] = 0$.

Step 2: Bounding the terms on ξ^C. By Corollary 21 we have

$$\left|\mathbb{E}\left[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi^C\}\right]\right| \leq \frac{KC_\xi}{n^{5/2}} \qquad \text{and} \qquad \left|\mathbb{E}\left[Z_{kq}\mathbb{I}\{\xi^C\}\right]\right| \leq \frac{2KC_\xi}{n^{3/2}}, \tag{45}$$

implying, since $\mathbb{E}[Z_{kq}] = 0$ (Step 1), also

$$|\mathbb{E}[Z_{kq}\mathbb{I}\{\xi\}]| \leq \frac{2KC_\xi}{n^{3/2}}. \tag{46}$$

Step 3: Bounding the terms on ξ. We recall that under Assumption 1, $n \geq 4K$, and $\delta = n^{-9/2}$, combining Lemmata 1 (Equation 15) and 2 we have that $\mathcal{A}_{\text{MC-UCB}}$ run by β_n given by (11) satisfies on ξ for all arm p, $-\lambda_p M \leq T_{p,n} - T^*_{p,n} \leq M$, where

$$M \stackrel{\text{def}}{=} 4\min\left(\frac{3\beta_n}{\Sigma_w\lambda_{\min}^{3/2}}\sqrt{n} + K, K^{1/3}\frac{3\beta_n\gamma_n}{\Sigma_w}n^{2/3} + K\right)$$

and $\gamma_n = (\bar{\Sigma}/\beta_n + \sqrt{8})^{1/3}$ as in Lemma 2. Recalling $\beta_n \leq C_\beta$ and $\gamma_n < 1.5$ from Appendix B.3 M is upper bounded by $\min\left(B\sqrt{n}, An^{2/3}\right)$, where

$$B \stackrel{\text{def}}{=} \frac{12C_\beta}{\Sigma_w\lambda_{\min}^{3/2}} + 2\sqrt{K} \qquad \text{and} \qquad A \stackrel{\text{def}}{=} K^{1/3}\left(\frac{18C_\beta}{\Sigma_w} + 4^{1/3}\right).$$

Moreover, by (16) of Lemma 2,

$$T_{p,n} \geq \frac{(w_p n)^{2/3}}{\gamma_n^2} > 4(\underline{w}n)^{2/3}/9 = En^{2/3} \qquad \text{on } \xi,$$

where $\underline{w} \stackrel{\text{def}}{=} \min_k w_k$ and $E \stackrel{\text{def}}{=} 4\underline{w}^{2/3}/9 > 0$. Note that B displays a dependency on $\lambda_{\min}^{-1}$, but A and E do not. Summarizing these inequalities on $T_{p,n}$ we have

$$T_{p,n} \geq \max\left(T^*_{p,n} - \lambda_p\min\left(B\sqrt{n}, An^{2/3}\right), En^{2/3}\right) \stackrel{\text{def}}{=} \underline{T}_{p,n}$$

$$\text{and} \qquad T_{p,n} \leq T^*_{p,n} + \min\left(B\sqrt{n}, An^{2/3}\right) \stackrel{\text{def}}{=} \bar{T}_{p,n}$$

on ξ. Note that using $n \geq 4K \geq 8$, $\Sigma_w^2 \leq c_1 \log(ec_2)$, $c_2 \geq 1$, and $\lambda_{\min} \leq 1/K$ it is easy to see that each $\bar{T}_{p,n} > 643$. Since now

$$\{\{T_{k,n} = t, T_{q,n} = t'\} \cap \xi : \underline{T}_{k,n} \leq t \leq \bar{T}_{k,n}, \underline{T}_{q,n} \leq t' \leq \bar{T}_{q,n}\}$$

is a partition of ξ, we have by Lemma 24

$$\begin{aligned}\Big|\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi\}\Big]\Big| &= \Big|\mathbb{E}\Big[Z_{kq}\sum_{t=\underline{T}_{k,n}}^{\bar{T}_{k,n}}\sum_{t'=\underline{T}_{q,n}}^{\bar{T}_{q,n}}\frac{1}{tt'}\mathbb{I}\big\{\{T_{k,n}=t, T_{q,n}=t'\}\cap\xi\big\}\Big]\Big| \\ &\leq \mathbb{E}|Z_{kq}\mathbb{I}\{\xi\}|\Big(\frac{1}{\underline{T}_{k,n}\underline{T}_{q,n}} - \frac{1}{\bar{T}_{k,n}\bar{T}_{q,n}}\Big) + \frac{1}{\bar{T}_{k,n}\bar{T}_{q,n}}|\mathbb{E}[Z_{kq}\mathbb{I}\{\xi\}]|.\end{aligned}$$

Note now that by Cauchy-Schwarz's inequality

$$\mathbb{E}|Z_{kq}\mathbb{I}\{\xi\}| \leq \mathbb{E}|Z_{kq}| \leq \sqrt{\mathbb{E}\Big[\Big(\sum_{t=1}^{T_{k,n}}(X_{k,t}-\mu_k)\Big)^2\Big]\mathbb{E}\Big[\Big(\sum_{t=1}^{T_{q,n}}(X_{q,t}-\mu_q)\Big)^2\Big]}.$$

Using Corollary 19 the right-hand side is bounded by $\sqrt{\mathbb{E}T_{k,n}\sigma_k^2\mathbb{E}T_{q,n}\sigma_q^2}$. Since

$$\mathbb{E}T_{k,n} = \mathbb{E}[T_{k,n}\mathbb{I}\{\xi\}] + \mathbb{E}[T_{k,n}\mathbb{I}\{\xi^C\}] \leq \bar{T}_{k,n}\mathbb{P}(\xi) + 2Kn^2\delta \leq \bar{T}_{k,n} + 2Kn^{-5/2} \leq \bar{T}_{k,n} + \sqrt{2}/64$$

by definition of $\bar{T}_{k,n}$ and $\bar{T}_{k,n} > 643$, $\mathbb{E}T_{k,n} < (1+\sqrt{2}/41152)\bar{T}_{k,n} < 1.01\bar{T}_{k,n}$. Similarly, $\mathbb{E}T_{q,n} < 1.01\bar{T}_{q,n}$. Thus we have $\mathbb{E}|Z_{kq}\mathbb{I}\{\xi\}| \leq 1.01\sigma_k\sigma_q\sqrt{\bar{T}_{k,n}\bar{T}_{q,n}}$. From this and (46), one gets

$$\begin{aligned}w_kw_q\Big|\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi\}\Big]\Big| &\leq 1.01w_k\sigma_kw_q\sigma_q\sqrt{\bar{T}_{k,n}\bar{T}_{q,n}}\Big(\frac{1}{\underline{T}_{k,n}\underline{T}_{q,n}} - \frac{1}{\bar{T}_{k,n}\bar{T}_{q,n}}\Big) + \frac{2w_kw_q}{\bar{T}_{k,n}\bar{T}_{q,n}}\frac{KC_\xi}{n^{3/2}} \\ &\leq 1.01\frac{w_k\sigma_kw_q\sigma_q}{\underline{T}_{k,n}\underline{T}_{q,n}}\frac{\bar{T}_{k,n}\bar{T}_{q,n} - \underline{T}_{k,n}\underline{T}_{q,n}}{\sqrt{\bar{T}_{k,n}\bar{T}_{q,n}}} + \frac{1.3KC_\xi}{10^6n^{3/2}}.\end{aligned}$$

Now for n large enough (compared to K, c_1, $\log c_2$, $1/\Sigma_w$, and $\log n$), $n \geq 8A^3$ (i.e., $An^{2/3} \leq n/2$) holds. Thus

$$\underline{T}_{p,n} \geq T^*_{p,n} - A\lambda_p n^{2/3} = \lambda_p(n - An^{2/3})$$

implies also $\frac{w_p\sigma_p}{\underline{T}_{p,n}} \leq \frac{\Sigma_w}{n-An^{2/3}} \leq 2\frac{\Sigma_w}{n}$ for any arm p. This leads to the bound

$$w_kw_q\Big|\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi\}\Big]\Big| \leq 4.04\frac{\Sigma_w^2}{n^2}\frac{\bar{T}_{k,n}\bar{T}_{q,n} - \underline{T}_{k,n}\underline{T}_{q,n}}{\sqrt{\bar{T}_{k,n}\bar{T}_{q,n}}} + \frac{1.3KC_\xi}{10^6n^{3/2}}. \tag{47}$$

Step 4: problem-dependent upper bound. We deduce that

$$\begin{aligned}\frac{\bar{T}_{k,n}\bar{T}_{q,n}-\underline{T}_{k,n}\underline{T}_{q,n}}{\sqrt{\bar{T}_{k,n}\bar{T}_{q,n}}} &\le \frac{\big(n\lambda_k+B\sqrt{n}\big)\big(n\lambda_q+B\sqrt{n}\big)-\big(n\lambda_k-B\lambda_k\sqrt{n}\big)\big(n\lambda_q-B\lambda_q\sqrt{n}\big)}{\sqrt{n\lambda_k n\lambda_q}}\\ &= \frac{B(\lambda_k+\lambda_q+2\lambda_k\lambda_q)n\sqrt{n}+B^2(1-\lambda_k\lambda_q)n}{n\sqrt{\lambda_k\lambda_q}}\\ &\le B\sqrt{n}\left(\frac{1+B/\sqrt{n}}{\sqrt{\lambda_k\lambda_q}}+2\sqrt{\lambda_k\lambda_q}\right)\\ &\le B\left(\frac{1+B/\sqrt{8}}{\lambda_{\min}}+1\right)\sqrt{n}\end{aligned}$$

using $n \ge 4K \ge 8$ and $\lambda_k\lambda_q \le 1/4$. Thus, we have from this and (47)

$$w_k w_q\Big|\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi\}\Big]\Big| \le 5B\left(\frac{1+B/\sqrt{8}}{\lambda_{\min}}+1\right)\frac{\Sigma_w^2}{n^{3/2}}+\frac{1.3KC_\xi}{10^6 n^{3/2}} = \frac{C_1+1.3KC_\xi/10^6}{n^{3/2}},$$

where $C_1 \stackrel{\text{def}}{=} 5B((1+B/\sqrt{8})/\lambda_{\min}+1)\Sigma_w^2$.

Finally, using (45), we have

$$\begin{aligned}\Big|w_k w_q\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\Big]\Big| &\le w_k w_q\Big|\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi\}\Big]\Big| + w_k w_q\Big|\mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n}T_{q,n}}\mathbb{I}\{\xi^C\}\Big]\Big|\\ &\le \frac{C_1+1.3KC_\xi/10^6}{n^{3/2}}+\frac{KC_\xi}{4n^{5/2}}\\ &\le \frac{C_1+(1.3K/10^6+1/16)C_\xi}{n^{3/2}},\end{aligned}$$

where C_1 and C_ξ depend only polynomially on $\log n$, $\lambda_{\min}^{-1}$, K, Σ_w, c_1, and $\log c_2$. This concludes the proof for the problem-dependent bound.

Step 4': problem-independent upper bound. Using $\bar{T}_{k,n} \ge \underline{T}_{k,n} \ge En^{2/3}$, which implies that $\bar{T}_{k,n} \ge \max(\lambda_k n, En^{2/3})$, we deduce that

$$\begin{aligned}\frac{\bar{T}_{k,n}\bar{T}_{q,n}-\underline{T}_{k,n}\underline{T}_{q,n}}{\sqrt{\bar{T}_{k,n}\bar{T}_{q,n}}} &\le \frac{\big(n\lambda_k+An^{2/3}\big)\big(n\lambda_q+An^{2/3}\big)-\big(n\lambda_k-A\lambda_k n^{2/3}\big)\big(n\lambda_q-A\lambda_q n^{2/3}\big)}{\sqrt{\max\big(\lambda_k n, En^{2/3}\big)\max\big(\lambda_q n, En^{2/3}\big)}}\\ &= \frac{A(\lambda_k+\lambda_q+2\lambda_k\lambda_q)nn^{2/3}+A^2(1-\lambda_k\lambda_q)n^{4/3}}{\sqrt{\max\big(\lambda_k\lambda_q n^2, E\max(\lambda_k,\lambda_q)nn^{2/3}, E^2n^{4/3}\big)}}\\ &\le A\left[\frac{(\lambda_k+\lambda_q)n^{5/3}}{\sqrt{E\max(\lambda_k,\lambda_q)n^{5/3}}}+\frac{2\lambda_k\lambda_q n^{5/3}}{\sqrt{\lambda_k\lambda_q}n}+\frac{An^{4/3}}{En^{2/3}}\right]\\ &\le An^{5/6}\left[\frac{\sqrt{\lambda_k+\lambda_q}}{\sqrt{E/2}}+\frac{2\sqrt{\lambda_k\lambda_q}}{n^{1/6}}+\frac{A}{En^{1/6}}\right]\\ &\le \frac{A}{\sqrt{2}}\left(\frac{2}{\sqrt{E}}+1+\frac{A}{E}\right)n^{5/6}\end{aligned}$$

using $n \geq 4K \geq 8$ and $\lambda_k \lambda_q \leq 1/4$. Thus, we have from this and (47) that

$$w_k w_q \Big| \mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n} T_{q,n}} \mathbb{I}\{\xi\} \Big] \Big| \leq 2.02\sqrt{2} A \left(\frac{2}{\sqrt{E}} + 1 + \frac{A}{E} \right) \frac{\Sigma_w^2}{n^{7/6}} + \frac{1.3 K C_\xi}{10^6 n^{3/2}} \leq \frac{C_2 + 9K^{2/3} C_\xi / 10^7}{n^{7/6}},$$

where $C_2 = 3A(2/\sqrt{E} + 1 + A/E)\Sigma_w^2$.

Finally, using (45), we have

$$\begin{aligned} \Big| w_k w_q \mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n} T_{q,n}} \Big] \Big| &\leq w_k w_q \Big| \mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n} T_{q,n}} \mathbb{I}\{\xi\} \Big] \Big| + w_k w_q \Big| \mathbb{E}\Big[\frac{Z_{kq}}{T_{k,n} T_{q,n}} \mathbb{I}\{\xi^C\} \Big] \Big| \\ &\leq \frac{C_2 + 9K^{2/3} C_\xi / 10^7}{n^{7/6}} + \frac{K C_\xi}{4 n^{5/2}} \\ &\leq \frac{C_2 + (9K^{2/3}/10^7 + 1/32) C_\xi)}{n^{7/6}}, \end{aligned}$$

where C_2 and C_ξ depend only polynomially on $\log n$, K, Σ_w, c_1, $\log c_2$, and $1/\underline{w}$. This concludes the proof for the problem-independent bound. ■

Appendix F. Proofs of Technical Lemmata

Proof of Lemma 11 Using that $\log(c_2/\delta) \geq 0$

$$\begin{aligned} \mathbb{E}\left[|X - \mu|^2 \mathbb{I}\{A\} \right] &= \int_0^\infty \mathbb{P}\big(|X - \mu|^2 > \epsilon, A \big)\, d\epsilon \\ &\leq \int_0^{c_1 \log(c_2/\delta)} \mathbb{P}(A)\, d\epsilon + \int_{c_1 \log(c_2/\delta)}^\infty \mathbb{P}\big(|X - \mu|^2 > \epsilon \big)\, d\epsilon \\ &\leq \delta c_1 \log(c_2/\delta) + \int_{c_1 \log(c_2/\delta)}^\infty c_2 e^{-\epsilon/c_1}\, d\epsilon = \delta c_1 \log(e c_2/\delta). \end{aligned}$$

■

Proof of Lemma 15 Using (9) for $\epsilon^2 = c_1 \log(c_2/\delta) (> 0)$ we have

$$\mathbb{P}(A^C) \leq c_2 e^{-c_1 \log(c_2/\delta)/c_1} = \delta \qquad \text{and} \qquad \mathbb{P}(A) \geq 1 - \delta > 0, \tag{48}$$

so $\mathrm{Var}[X|A]$ and also $\tilde{\mu} \stackrel{\text{def}}{=} \mathbb{E}[X|A] = \mathbb{E}[X\mathbb{I}\{A\}]/\mathbb{P}(A)$ make sense. If $\mathbb{P}(A) = 1$ then $\tilde{\sigma} = \sigma$, and the claim follows. Now assume $\mathbb{P}(A) < 1$. Since $\mathbb{E}[|X - \mu|^2 | A^C] \geq c_1 \log(c_2/\delta) \geq \mathbb{E}[|X - \mu|^2 | A]$, we have

$$\sigma^2 = \mathbb{E}[|X - \mu|^2] = \mathbb{E}[|X - \mu|^2 | A^C]\mathbb{P}(A^C) + \mathbb{E}[|X - \mu|^2 | A]\mathbb{P}(A) \geq \mathbb{E}[|X - \mu|^2 | A]. \tag{49}$$

Moreover,

$$\tilde{\sigma}^2 = \mathbb{E}[|X - \tilde{\mu}|^2 | A] = \mathbb{E}[|X - \mu|^2 | A] - |\mu - \tilde{\mu}|^2,$$

and thus

$$\sigma^2 - \tilde{\sigma}^2 = \sigma^2 - \mathbb{E}[|X-\mu|^2|A] + |\mu - \tilde{\mu}|^2 \geq 0 \tag{50}$$

by (49). But (49) implies also that

$$\begin{aligned}\sigma^2 - \mathbb{E}[|X-\mu|^2|A] &= \frac{\sigma^2\mathbb{P}(A) - \mathbb{E}[|X-\mu|^2\mathbb{I}\{A\}]}{\mathbb{P}(A)} = \frac{\mathbb{E}[|X-\mu|^2\mathbb{I}\{A^C\}] - \sigma^2\mathbb{P}(A^C)}{\mathbb{P}(A)} \\ &= \frac{\mathbb{E}[(|X-\mu|^2 - \sigma^2)\mathbb{I}\{A^C\}]}{\mathbb{P}(A)}. \end{aligned} \tag{51}$$

Using that $\delta \leq 1/e$ and Lemma 11 imply $c_1 \log(c_2/\delta) \geq c_1 \log(ec_2) \geq \sigma^2$, we have

$$\begin{aligned}&\mathbb{E}\big[(|X-\mu|^2 - \sigma^2)\mathbb{I}\{A^C\}\big] = \int_0^\infty \mathbb{P}(|X-\mu|^2 - \sigma^2 > \epsilon', A^C)\, d\epsilon' = \int_{\sigma^2}^\infty \mathbb{P}(|X-\mu|^2 > \epsilon, A^C)\, d\epsilon \\ &= \int_{\sigma^2}^{c_1\log(c_2/\delta)} \mathbb{P}(A^C)\, d\epsilon + \int_{c_1\log(c_2/\delta)}^\infty \mathbb{P}(|X-\mu|^2 > \epsilon)\, d\epsilon \\ &\leq \delta(c_1\log(c_2/\delta) - \sigma^2) + \int_{c_1\log(c_2/\delta)}^\infty c_2 e^{-\epsilon/c_1}\, d\epsilon \qquad \text{(by Equations 48 and 9)} \\ &= \delta c_1 \log(c_2/\delta) - \delta\sigma^2 + c_1 c_2 e^{-c_1\log(c_2/\delta)/c_1} = \delta c_1 \log(ec_2/\delta) - \delta\sigma^2.\end{aligned}$$

This, (51), and (48) imply

$$\sigma^2 - \mathbb{E}[|X-\mu|^2|A] \leq \delta\frac{c_1\log(ec_2/\delta) - \sigma^2}{1-\delta}. \tag{52}$$

For the last term of (50), noticing that $\mathbb{E}[X\mathbb{I}\{A^C\}] + \mathbb{E}[X\mathbb{I}\{A\}] = \mu$ we have

$$\begin{aligned}|\mu - \tilde{\mu}| &= \left|\frac{\mu\mathbb{P}(A) - \mathbb{E}[X\mathbb{I}\{A\}]}{\mathbb{P}(A)}\right| = \frac{\left|\mathbb{E}[X\mathbb{I}\{A^C\}] - \mu\mathbb{P}(A^C)\right|}{\mathbb{P}(A)} = \frac{\left|\mathbb{E}[(X-\mu)\mathbb{I}\{A^C\}]\right|}{\mathbb{P}(A)} \\ &\leq \frac{\sqrt{\mathbb{E}[|X-\mu|^2]\mathbb{E}[\mathbb{I}\{A^C\}]}}{\mathbb{P}(A)} \qquad \text{(by Cauchy-Schwarz's inequality)} \\ &= \frac{\sigma\sqrt{\mathbb{P}(A^C)}}{\mathbb{P}(A)} \leq \frac{\sigma\sqrt{\delta}}{1-\delta}\end{aligned} \tag{53}$$

using again (48). From (50), (52), and (53), we derive

$$\sigma^2 - \tilde{\sigma}^2 \leq \delta\frac{c_1\log(ec_2/\delta) - \sigma^2}{1-\delta} + \frac{\delta\sigma^2}{(1-\delta)^2} \leq c_1\delta\frac{\log(ec_2/\delta)}{(1-\delta)^2}.$$

Since $(\sigma - \tilde{\sigma})^2 \leq (\sigma + \tilde{\sigma})(\sigma - \tilde{\sigma}) = \sigma^2 - \tilde{\sigma}^2$, the claim follows. ■

Proof of Lemma 23 Denote $(\hat{s}_2, \ldots, \hat{s}_n)$ by $\hat{S}(X)$. Then $\hat{S}(X) = \hat{S}(-X)$ holds due to the quadratic form of the empirical variances. Thus, by the symmetry of ν,

$$\mathbb{E}[X_t|\hat{S}(X)] = \mathbb{E}[-X_t|\hat{S}(-X)] = -\mathbb{E}[X_t|\hat{S}(X)] \quad \text{a.s.},$$

implying $\mathbb{E}[X_t|\{\hat{s}_{t'}\}_{t'\le n}] = \mathbb{E}[X_t|\hat{S}(X)] = 0$ a.s. ■

Proof of Lemma 24 By the definition of $\bar{a}$ and $\underline{a}$,

$$X\sum_{u=1}^{p} a_u \mathbb{I}\{\Omega_u\} \le X\mathbb{I}\{X \ge 0\}\,\bar{a}\mathbb{I}\{\Omega'\} + X\mathbb{I}\{X < 0\}\,\underline{a}\mathbb{I}\{\Omega'\}\,.$$

This implies

$$\begin{aligned}
\mathbb{E}\Big[X\sum_{u=1}^{p} a_u \mathbb{I}\{\Omega_u\}\Big] &\le \mathbb{E}\Big[X\mathbb{I}\{X \ge 0\}\,\bar{a}\mathbb{I}\{\Omega'\} + X\mathbb{I}\{X < 0\}\,\underline{a}\mathbb{I}\{\Omega'\}\Big] \\
&= \mathbb{E}\Big[(\bar{a}-\underline{a})X\mathbb{I}\{X \ge 0\}\,\mathbb{I}\{\Omega'\} + \underline{a}X(\mathbb{I}\{X<0\}+\mathbb{I}\{X\ge 0\})\mathbb{I}\{\Omega'\}\Big] \\
&= (\bar{a}-\underline{a})\mathbb{E}\Big[X\mathbb{I}\{X \ge 0\}\,\mathbb{I}\{\Omega'\}\Big] + \underline{a}\mathbb{E}[X\mathbb{I}\{\Omega'\}] \\
&\le (\bar{a}-\underline{a})\mathbb{E}|X\mathbb{I}\{\Omega'\}| + \underline{a}\mathbb{E}[X\mathbb{I}\{\Omega'\}].
\end{aligned}$$

By applying the inequality above for $-X$ we have

$$\mathbb{E}\Big[X\sum_{u=1}^{p} a_u \mathbb{I}\{\Omega_u\}\Big] \ge -(\bar{a}-\underline{a})\mathbb{E}|X\mathbb{I}\{\Omega'\}| + \underline{a}\mathbb{E}[X\mathbb{I}\{\Omega'\}].$$

Those two inequalities lead to the second inequality of the lemma, while the first one follows from the triangle inequality. ■

References

A. Antos. *Performance Limits of Nonparametric Estimators.* PhD thesis, Technical University of Budapest, May 1999. URL `http://www.cs.bme.hu/~antos/ps/dr.pdf`.

A. Antos, V. Grover, and Cs. Szepesvári. Active learning in heteroscedastic noise. *Theoretical Computer Science*, 411(29–30):2712–2728, June 17 2010. doi: 10.1016/j.tcs.2010.04.007. Special Issue for ALT 2008. Available online 10 April 2010.

B. Arouna. Adaptative Monte Carlo method, a variance reduction technique. *Monte Carlo Methods and Applications*, 10(1):1–24, 2004.

K.B. Athreya and S.N. Lahiri. *Measure Theory and Probability Theory.* Springer, 2006.

J.-Y. Audibert and S. Bubeck. Minimax policies for adversarial and stochastic bandits. In S. Dasgupta and A. Klivans, editors, *Proceedings of the* 22nd *Annual Conference on Learning Theory*, pages 217–226. Omnipress, June 2009.

J.-Y. Audibert, R. Munos, and Cs. Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009.

J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In *Proceedings of the* 23rd *Annual Conference on Learning Theory*, pages 41–53, 2010.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002. ISSN 0885-6125.

S. Bubeck and N. Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, December 12, 2012. doi: 10.1561/2200000024.

S. Bubeck, R. Munos, and G. Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, April 22, 2011. ISSN 03043975. doi: 10.1016/j.tcs.2010.12.059.

V.V Buldygin and Y.V. Kozachenko. Sub-gaussian random variables. *Ukrainian Mathematical Journal*, 32(6):483–489, 1980.

A. Carpentier and R. Munos. Finite time analysis of stratified sampling for monte carlo. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1278–1286. Curran Associates, 2011.

A. Carpentier and R. Munos. Minimax number of strata for online stratified sampling given noisy samples. In N.H. Bshouty, G. Stoltz, N. Vayatis, and T. Zeugmann, editors, *Proceedings of the* 23rd *International Conference, Algorithmic Learning Theory 2012*, volume 7568 of *LNCS/LNAI*, pages 229–244, Berlin, Heidelberg, 2012. Springer-Verlag.

A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, and P. Auer. Upper-confidence-bound algorithms for active learning in multi-armed bandits. In J. Kivinen, C. Szepesvári, E. Ukkonen, and Th. Zeugmann, editors, *Proceedings of the* 22^{nd} *International Conference, Algorithmic Learning Theory 2011*, volume 6925 of *LNCS/LNAI*, pages 189–203, Berlin, Heidelberg, 2011. Springer-Verlag.

A. Carpentier, A. Lazaric, M. Ghavamzadeh, R. Munos, P. Auer, and A. Antos. Upper-confidence-bound algorithms for active learning in multi-armed bandits. *ArXiv e-prints*, July 2015. URL `http://arxiv.org/abs/1507.04523`. Technical Report.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge Univ Press, 2006. ISBN 0521841089.

P. Etoré and B. Jourdain. Adaptive optimal allocation in stratified sampling methods. *Methodol. Comput. Appl. Probab.*, 12(3):335–360, September 2010.

P. Etoré, G. Fort, B. Jourdain, and É. Moulines. On adaptive stratification. *Annals of Operations Research*, 189(1):127–154, September 2011. doi: 10.1007/s10479-009-0638-9. Published online: November 21, 2009.

P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer Verlag, 2004. ISBN 0387004513.

V. Grover. Active learning and its application to heteroscedastic problems. Master's thesis, Department of Computing Science, Univ. of Alberta, Edmonton, AB, Canada, 2009.

R. Kawai. Asymptotically optimal allocation of stratified sampling with adaptive variance reduction by strata. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 20(2):1–17, 2010. ISSN 1049-3301.

T.L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

A. Maurer and M. Pontil. Empirical Bernstein bounds and sample-variance penalization. In *Proceedings of the* 22nd *Annual Conference on Learning Theory*, pages 115–124, 2009.

R.Y. Rubinstein and D.P. Kroese. *Simulation and the Monte Carlo Method*. Wiley-interscience, 2008. ISBN 0470177942.

Journal of Machine Learning Research 16 (2015) 2273-2328 Submitted 12/13; Revised 10/14; Published 8/15

Concave Penalized Estimation of Sparse Gaussian Bayesian Networks

Bryon Aragam BRYON@STAT.UCLA.EDU
Qing Zhou ZHOU@STAT.UCLA.EDU
Department of Statistics
University of California, Los Angeles
Los Angeles, CA 90024, USA

Editor: Max Chickering

Abstract

We develop a penalized likelihood estimation framework to learn the structure of Gaussian Bayesian networks from observational data. In contrast to recent methods which accelerate the learning problem by restricting the search space, our main contribution is a fast algorithm for score-based structure learning which does not restrict the search space in any way and works on high-dimensional data sets with thousands of variables. Our use of concave regularization, as opposed to the more popular ℓ_0 (e.g. BIC) penalty, is new. Moreover, we provide theoretical guarantees which generalize existing asymptotic results when the underlying distribution is Gaussian. Most notably, our framework does not require the existence of a so-called faithful DAG representation, and as a result, the theory must handle the inherent nonidentifiability of the estimation problem in a novel way. Finally, as a matter of independent interest, we provide a comprehensive comparison of our approach to several standard structure learning methods using open-source packages developed for the `R` language. Based on these experiments, we show that our algorithm obtains higher sensitivity with comparable false discovery rates for high-dimensional data and scales efficiently as the number of nodes increases. In particular, the total runtime for our method to generate a solution path of 20 estimates for DAGs with 8000 nodes is around one hour.

Keywords: Bayesian networks, concave penalization, directed acyclic graphs, coordinate descent, nonconvex optimization

1. Introduction

The problem of estimating Bayesian networks (BNs) has received a significant amount of attention over the past decade, with applications ranging from medicine and genetics to expert systems and artificial intelligence. The idea of using directed graphical models such as Bayesian networks to model real-world phenomena is certainly nothing new, and while the calculus of these models has been well-developed, the development of fast algorithms to accurately estimate these models in high-dimensions has been slow. The basic problem can be formulated as follows: Given observations from a probability distribution, is it possible to construct a directed acyclic graph (DAG) which decomposes the distribution into a sparse Bayesian network?

Based on observational data alone, it is well-known that there are many Bayesian networks that are consistent in the Markov sense with a given distribution. What we are

interested in is finding the sparsest possible Bayesian network, estimated purely from i.i.d. observations without any experimental data. When the number of variables is small, there are many practical algorithms for solving this problem. Unfortunately, as the number of variables increases, this problem becomes notoriously difficult: the learning problem is nonconvex, NP-hard, and scales super-exponentially with the number of variables (Chickering, 1996; Chickering and Meek, 2002; Robinson, 1977). Since many realistic networks can have upwards of thousands or even tens of thousands of nodes—genetic networks being a prominent example of great importance—the development of new statistical methods for learning the structure of Bayesian networks is critical.

In this work, we use a penalized likelihood estimation framework to learn the structure of Gaussian Bayesian networks from observational data. Our framework is based on recent work by Fu and Zhou (2013) and van de Geer and Bühlmann (2013), who show how these ideas lead to a family of estimators with good theoretical properties and whose estimation performance is competitive with traditional approaches. Neither of these works, however, consider the computational challenges associated with high-dimensional data sets for which the dimension scales to thousands of variables, which is a key challenge in Bayesian network learning. With these computational challenges in mind, we sought to develop a score-based method that:

- Does not restrict or prune the search space in any way;
- Does not assume faithfulness;
- Does not require a known variable ordering;
- Works on observational data (i.e. without experimental interventions);
- Works effectively in high dimensions ($p \gg n$);
- Is capable of handling graphs with several thousand variables.

While various methods in the literature cover a few of these requirements, none that we are aware of simultaneously cover *all* of them. The main contribution of the present work is a fast algorithm for score-based structure learning that accomplishes precisely that.

One of the key developments in our method is the application of modern regularization techniques, including both ℓ_1 and concave penalties. Although ℓ_1 regularization is well-understood with attractive high-dimensional and computational properties (Bühlmann and van de Geer, 2011), as we shall see, in the context of Bayesian networks many of these advantages disappear. While our approach still allows for ℓ_1-based penalties in practice, our results will indicate that concave penalties such as the SCAD (Fan and Li, 2001) and MCP (Zhang, 2010) offer improved performance. This is in line with recent advances in sparse learning that have highlighted the advantages of nonconvex regularization in linear and generalized linear models (Lv and Fan, 2009; Fan and Lv, 2010, 2011; Zhang and Zhang, 2012; Huang et al., 2012; Fan and Lv, 2013). Notwithstanding, both our theory and our method apply to a general class of penalties which can be chosen based on the application at hand.

In this light, our method also represents a major conceptual departure from existing methods in the literature on Bayesian networks through its deep involvement of recent developments in sparse regression, as well as using parametric modeling via structural equations as its foundation, in contrast to the more common approach using graph theory and Markov equivalence. These techniques have long been known to be useful in regression modeling, covariance estimation, matrix factorization, and image processing, but their application to Bayesian networks, as far as we can tell, is a recent development (Schmidt et al., 2007; Xiang and Kim, 2013; Fu and Zhou, 2013, 2014). Finally, our method offers new insights into accelerating score-based algorithms in order to compete with hybrid and constraint-based methods which, as we will show, are generally faster and more effective than traditional score-based algorithms.

The organization of the rest of this paper is as follows: In the remainder of this section we review previous work and compare our contributions with the existing literature. In Section 2, we establish the necessary preliminaries for our approach via structural equations. In Section 3 we define and discuss the penalized estimator that is the focus of this paper. Section 4 then provides the necessary finite-dimensional theory to justify the use of our estimator. A complete description of our algorithm is outlined in Section 5, followed by an empirical evaluation of the algorithm in Section 6. Section 6 also offers a side-by-side comparison of our algorithm with four other structure learning algorithms, and Section 7 provides an evaluation of these algorithms using a real-world data set. We finally conclude with a discussion of some future directions for this research.

1.1 Related Work

The idea of using penalized likelihood estimation and sparse regression to learn Gaussian Bayesian networks in high dimensions is a recent development, and the theoretical basis for ℓ_0 penalization has been instigated by van de Geer and Bühlmann (2013). Their work relies on the interpretation of Gaussian Bayesian networks in terms of structural equation models (Drton and Richardson, 2008; Drton et al., 2011), which provides a natural interpretation of network edges in terms of coefficients of a regression model. To the best of our knowledge, the work of van de Geer and Bühlmann (2013) is the first high-dimensional analysis of a score-based approach in the literature, and has not yet been generalized to the case of continuous ℓ_1 or concave penalties. As the nontrivial and novel nature of this analysis would detract from our primary goal of addressing computational challenges, we will not pursue a corresponding high-dimensional theory here. Given this foundational work, our goal is to show that these ideas can be translated into a family of fast algorithms for score-based learning of Bayesian network structures.

While the traditional approach to estimating Bayesian networks uses ℓ_0-based penalties such as the Bayesian information criterion (BIC), Fu and Zhou (2013) recently introduced the idea of using continuous penalties via the adaptive ℓ_1 penalty and showed that it can be competitive in practice. They combine a novel method of enforcing acyclicity with a block coordinate descent algorithm in order to compute an ℓ_1-penalized maximum likelihood estimator for structure learning. Their algorithm is adapted to the case of intervention data and does not exploit the underlying convexity of the Gaussian likelihood function; as a result, it cannot be used on high-dimensional data and is limited to graphs with

200 or so nodes. By contrast, the method proposed here adapts this algorithm for use with observational, high-dimensional data, and takes explicit advantage of convexity and sparsity. We also extend these ideas to a general class of penalties which includes both ℓ_0 and ℓ_1 regularization as special cases. The result is an algorithm which easily handles thousands of nodes in a matter of minutes. Moreover, in contrast to the theory proposed in Fu and Zhou (2013), our theory does not rely on faithfulness or identifiability.

1.2 Review of Structure Learning

Traditionally, there are three main approaches to learning Gaussian Bayesian networks.

Score-based. In the score-based approach, a scoring function is defined over the space of DAG structures, and one searches this space for a structure that optimizes the chosen scoring function. The most commonly used scoring functions are based on the a posteriori probability of a network structure (Geiger and Heckerman, 2013), while others use minimum-description length, which is equivalent to BIC (Lam and Bacchus, 1994). In terms of implementation, the standard algorithmic approach is greedy hill-climbing (Heckerman et al., 1995), for which various improvements have been offered over the years (e.g. Chickering, 2003). Monte Carlo methods have also been used to sample network structures according to an a posteriori distribution (Ellis and Wong, 2008; Zhou, 2011).

Constraint-based. In the constraint-based approach, repeated conditional independence tests are used to check for the existence of edges between nodes. The idea is to search for statistical independence between variables, which indicates that an edge cannot exist in the underlying DAG structure as long as certain assumptions are satisfied. These assumptions tend to be very strong in practice, and this constitutes the main drawback of this approach. Conversely, since the tests of independence can be efficient, constraint-based approaches tend to be faster than score-based approaches. Two popular approaches in this spirit are the PC algorithm (Spirtes and Glymour, 1991; Kalisch and Bühlmann, 2007) and the MMPC algorithm (Tsamardinos et al., 2006).

Hybrid. In the hybrid approach, constraint-based search is used to prune the search space (e.g. to find the skeleton or a moral graph representation), which is then used as an input to restrict a score-based search. By removing as many edges as possible in the first step, the second step can be significantly faster than unrestricted score-based searching. This technique has been shown to work well in practice by combining the advantages of score-based and constraint-based approaches (Tsamardinos et al., 2006; Gámez et al., 2011, 2012).

As previously noted, the main issue with modern approaches to structure learning is scaling algorithms to data sets of ever-increasing sizes. Tsamardinos et al. (2006) show how their hybrid MMHC algorithm scales to 5,000 variables, although the running time of 13 days left much to be desired. By assuming the underlying DAG is sparse, Kalisch and Bühlmann (2007) show how exploiting sparsity in the PC algorithm leads to significant computational gains. More recently, Gámez et al. (2012) have proposed modifications to hybrid hill-climbing that scale to 1000 or so variables. By taking advantage of distributed computation, Scutari (2014) shows how to scale constraint-based approaches to thousands of variables. Notably, none of these methods fall into the first category of score-based methods. In contrast, the method proposed in the present work is a genuine score-based

method and scales efficiently to graphs with thousands of variables. To the best of our knowledge, this is one of the first purely score-based methods that accomplishes this in the sense that we rely neither on significance tests (as in the constraint-based approach) nor pruning the search space (as in the hybrid approach).

2. Preliminaries

We will develop our framework by using a multivariate Gaussian distribution as our starting point, which we will then decompose into a Bayesian network in order to define our estimator. Our approach is purely algebraic, relying on the uniqueness of the Cholesky decomposition in order to factorize a Gaussian distribution into a set of linear structural equations. In what follows, the reader may recall that the structure of a Bayesian network is completely determined by a directed acyclic graph, and hence learning the structure of a Bayesian network reduces to learning directed acyclic graphs. In order to maintain consistency and ease of translation, much of our notation is adapted from van de Geer and Bühlmann (2013).

2.1 Background and Notation

We assume throughout that the data are generated from a p-variate Gaussian distribution,

$$(X_1, \ldots, X_p) \sim \mathcal{N}(0, \Sigma_0), \tag{1}$$

where the covariance matrix $\Sigma_0 \in \mathbb{R}^{p \times p}$ is positive definite. Such a model can always be written as a set of Gaussian structural equations as follows (see Dempster, 1969):

$$X_j = \sum_{i=1}^{p} \beta_{ij}^0 X_i + \varepsilon_j, \quad j = 1, \ldots, p, \tag{2}$$

where the ε_j are mutually independent with $\varepsilon_j \sim \mathcal{N}(0, (\omega_j^0)^2)$, ε_j is independent of $\Pi_j^0 = \{X_i : \beta_{ij}^0 \neq 0\}$, and $\beta_{jj}^0 = 0$. This decomposition is not unique, and we will let $B_0 = (\beta_{ij}^0)$ denote any matrix of coefficients that satisfies (2). The matrix $B_0 = (\beta_{ij}^0)$ can then be regarded as the weighted adjacency matrix of a directed acyclic graph and represents a Bayesian network for the distribution $\mathcal{N}(0, \Sigma_0)$. Recall that a *directed acyclic graph* B is a directed graph containing no directed cycles. In a slight abuse of notation, we will identify a DAG B with its weighted adjacency matrix, which we will also denote by $B = (\beta_{ij})$.

The nodes of B are in one-to-one correspondence with the random variables $X_1, \ldots, X_p$ in our model. Following tradition, we make no distinction between random variables and nodes or vertices, and will use these terms interchangeably. We say that X_k is a *parent* of X_j if $X_k \to X_j$, and the set of parents of X_j will be denoted by $\Pi_j := \Pi_j(B)$. We will denote the number of edges in B by $s_B := |\{\beta_{ij} \neq 0\}|$. When the underlying graph is clear from context, we will suppress the dependence on B and simply denote the number of edges by s.

Unless otherwise noted, $\|\cdot\|$ shall always mean the standard Euclidean norm and $\|\cdot\|_F$ will denote the standard ℓ_2 Frobenius norm on matrices. For a general matrix $A = (a_{ij})_{n \times p} \in \mathbb{R}^{n \times p}$, its columns will be denoted using lowercase and single subscripts, so that

$$A = [a_1 \,|\, \cdots \,|\, a_p], \quad a_i \in \mathbb{R}^n \text{ for } i = 1, \ldots, p.$$

The square brackets signal that A is a matrix with p columns given by $a_1, \ldots, a_p$. In particular, we will write $B = [\beta_1 \,|\, \cdots \,|\, \beta_p]$ for an arbitrary DAG. The support of a matrix is defined by $\mathrm{supp}(B) := \{(i,j) : \beta_{ij} \neq 0\}$.

If $X = [x_1 \,|\, \cdots \,|\, x_p]$ is an $n \times p$ data matrix of i.i.d. observations from (1), then we can rewrite (2) as a matrix equation,

$$X = XB_0 + E, \tag{3}$$

where $E \in \mathbb{R}^{n \times p}$ is the matrix of noise vectors. This model has $p(p-1) + p = p^2$ free parameters, which we encode through two matrices given by (B_0, Ω_0). Here, $\Omega_0 = \mathrm{diag}((\omega_1^0)^2, \ldots, (\omega_p^0)^2)$ is the matrix of error variances. We denote the matrix of error variances by Ω in order to avoid confusion with the covariance matrix Σ.

There are thus two unknown parameters in (2):

$$B := (\beta_{ij}) \in \mathbb{R}^{p \times p},$$
$$\Omega := \mathrm{diag}(\omega_1^2, \ldots, \omega_p^2) \in \mathbb{R}^{p \times p}.$$

Given n i.i.d. observations of the variables $(X_1, \ldots, X_p)$, the negative log-likelihood of the data $X \in \mathbb{R}^{n \times p}$ is easily seen to be

$$L(B, \Omega \,|\, X) = \sum_{j=1}^{p} \left[\frac{n}{2} \log(\omega_j^2) + \frac{1}{2\omega_j^2} \|x_j - X\beta_j\|^2 \right]. \tag{4}$$

Observe that the function in (4) is nonconvex; this fact will play an important role in the development of our method.

Remark 1. The vast majority of the literature on Bayesian networks focuses on discrete data, in contrast to our method which assumes the data are Gaussian. As the motivation for this work is to scale penalized likelihood methods for high-dimensional data, the Gaussian case is a natural starting point, as much of the high-dimensional statistical theory is tailored towards this case. Recent work has shown how to adapt our techniques to the discrete case via multi-logit regression (Fu and Zhou, 2014). Further generalizations to more general continuous distributions remain for future work. Finally, even though our method implicitly assumes the data are Gaussian, one may naively use our algorithm on discrete data and still obtain reasonable results (see Section 7).

Thus far we have viewed the distribution $\mathcal{N}(0, \Sigma_0)$ as the data-generating mechanism, rewriting this in terms of (B_0, Ω_0) by using well-known properties of the Gaussian distribution. We could just as well have gone the other way around: Given a DAG B and variance matrix $\Omega = \mathrm{diag}(\omega_1^2, \ldots, \omega_p^2)$, the parameters (B, Ω) uniquely define a structural equation model as in (2), and this model defines a $\mathcal{N}(0, \Sigma)$ distribution. By (3), we have for any (B, Ω),

$$\Sigma = (I - B)^{-T} \Omega (I - B)^{-1}, \tag{5}$$

and hence Σ is uniquely determined by (B, Ω). Considering instead the inverse covariance matrix $\Theta = \Sigma^{-1}$, we can define

$$\Theta = \Theta(B, \Omega) = (I - B)\Omega^{-1}(I - B)^T. \tag{6}$$

By using (6) and defining $S_n := X^T X$, the negative log-likelihood in (4) can be rewritten in terms of $\Theta = \Theta(B, \Omega)$ directly as

$$L(\Theta \,|\, X) = -\frac{n}{2} \log\det \Theta + \frac{1}{2} \operatorname{tr}(\Theta S_n). \tag{7}$$

By combining (4) and (7), we have $L(B, \Omega \,|\, X) = L(\Theta(B, \Omega) \,|\, X)$. This expression shows how the weighted adjacency matrix of a DAG can be considered as a reparameterization of the usual normal distribution, and gives us an explicit connection between inverse covariance estimation and DAG estimation, which will be explored further in the next subsection.

Since the decomposition of a normal distribution as a linear structural equation model (SEM) as in (2) is not unique, we can define the following equivalence class of DAGs:

$$\mathcal{E}(\Theta) := \{(B, \Omega) : \Theta(B, \Omega) = \Theta\}. \tag{8}$$

When $(B, \Omega) \in \mathcal{E}(\Theta)$, we shall say that B *represents*, or *is consistent with*, Θ. Two DAGs $(B, \Omega), (B', \Omega')$ will be called *equivalent* if they belong to the same equivalence class $\mathcal{E}(\Theta)$.

This definition of equivalence in terms of equivalent parameterizations is indeed different from the usual definition of *distributional* or *Markov equivalence* that is common in the Bayesian network literature. Furthermore, while it is commonplace to assume that the true underlying distribution is faithful to the DAG B_0—which roughly speaking entails that B_0 contains exactly the same conditional independence constraints as the true distribution—we have deliberately sidestepped considerations of this hypothesis since our theory does not rely on faithfulness.

Remark 2. Strictly speaking, a Gaussian Bayesian network is specified by *both* a weighted adjacency matrix B and a variance matrix Ω, however, we will frequently refer to a BN simply by its adjacency matrix B. Although it may not be explicitly mentioned, when there is any ambiguity one may assume that there is an assumed variance matrix Ω paired with B.

2.2 Comparison of Graphical Models

The previous section showed how the weighted adjacency matrix of a DAG can be considered as a reparameterization of the usual normal distribution, and gave an explicit connection between inverse covariance estimation and DAG estimation: Equation (6) shows how any DAG (B, Ω) uniquely defines an inverse covariance matrix $\Theta = \Theta(B, \Omega)$. It follows that any estimate $(\widehat{B}, \widehat{\Omega})$ of the true DAG yields an estimate of Θ_0 given by $\widehat{\Theta} := \Theta(\widehat{B}, \widehat{\Omega})$. In the context of the PC algorithm, this has been studied by Rütimann and Bühlmann (2009). As a result, one may also view our framework as defining an estimator for the inverse covariance matrix. Covariance selection and precision matrix estimation have a long history in the statistical literature (Dempster, 1972), with recent approaches employing regularization in various incarnations (e.g. Meinshausen and Bühlmann, 2006; Chaudhuri et al., 2007; Banerjee et al., 2008; Friedman et al., 2008; Ravikumar et al., 2011). A detailed survey of recent progress in this area can be found in Pourahmadi (2013). We will not pursue this connection in detail here, however, a few comments are in order.

First, while these two problems are deeply connected, estimating an inverse covariance matrix is significantly easier: The estimation problem is statistically identifiable and the

parameter space is convex. This stands in stark contrast to the more difficult problem of estimating an underlying DAG, which we have shown to be simultaneously *nonidentifiable* and *nonconvex*. As a result, while the high-dimensional properties of regularized covariance estimation are well-understood, the high-dimensional properties of DAG estimation have proven much more difficult to ascertain. The only significant results we are aware of are in van de Geer and Bühlmann (2013) and Kalisch and Bühlmann (2007).

Second, our approach is also distinct from existing methods that directly regularize Cholesky factors (Huang et al., 2006; Lam and Fan, 2009), as they make implicit use of an *a priori* ordering amongst the variables. As such, the consistency theory in Lam and Fan (2009) for the sparse Cholesky decomposition does not apply directly to our method. Finally, while there are important similarities between Bayesian networks and other undirected models such as Markov random fields and Ising models, our framework has so far only been applied to the former. For applications of Bayesian networks to inferring so-called Markov blankets, see Aliferis et al. (2010a,b).

Part of the justification for our framework is that it produces sparse BNs that yield good fits to the true distribution, which is tantamount to producing good estimates of the inverse covariance matrix Θ_0. This will be established through the theory presented in Section 4, as well as empirically via the simulations discussed in Section 6. Because of the significance and popularity of covariance selection methods, it would of course be interesting to compare our estimate of Θ_0 to the methods cited in the above discussion. As our desire is to keep the focus on estimating Bayesian networks, such comparisons are left to future work.

2.3 Permutations and Equivalence

In this section we wish to exhibit the connection between equivalent DAGs as defined in (8) and the choice of a permutation of the variables. Recall that a *topological sort* of a directed graph is an ordering on the nodes, often denoted by $\prec$, such that the existence of a directed edge $X_k \to X_j$ implies $X_k \prec X_j$ in the ordering. A directed graph has a topological sort if and only if it is acyclic, and in general such a sort need not be unique.

When describing equivalent DAGs, it is easier to interpret an ordering in terms of a permutation of the variables. Let $\mathcal{P}$ denote the collection of all permutations of the indices $\{1, \dots, p\}$. For an arbitrary matrix A and any $\pi \in \mathcal{P}$, let us denote by $P_\pi A$ the matrix obtained by permuting the rows and columns of A according to π, so that $(P_\pi A)_{ij} = a_{\pi(i)\pi(j)}$. Then a DAG can be equivalently defined as any graph whose adjacency matrix B admits a permutation π such that $P_\pi B$ is strictly triangular. When the order of the nodes in $P_\pi B$ matches a topological sort of B, that is if $X_k \prec X_j \implies \pi^{-1}(k) < \pi^{-1}(j)$, then the matrix $P_\pi B$ will be strictly upper triangular. For our purposes, however, it will be easier to use a *lower*-triangularization, which we now describe.

A DAG B will be called *compatible with the permutation* π if $P_\pi B$ is lower-triangular, which is equivalent to saying that $X_k \to X_j$ (i.e. $X_k \prec X_j$) in B implies $\pi^{-1}(k) > \pi^{-1}(j)$. Similarly, π will also be called *compatible* with B. Such a permutation π may be obtained by simply reversing any topological sort for B, so that parents come *after* their children. Formally, suppose $X_1 \prec X_2 \prec \cdots \prec X_p$ is a topological sort of B. Then the permutation

$$\pi(i) = p - i + 1, \quad i = 1, \dots, p,$$

is compatible with B. Our decision to use lower-triangular matrices is for consistency with existing literature and to allow a convenient interpretation of the matrix B as the weighted adjacency matrix of a graph. This will also simplify the technical discussion below (e.g. compare equation (6) above with (9) below).

Suppose Θ_0 is a fixed positive definite matrix and $\pi \in \mathcal{P}$. Then the matrix $P_\pi \Theta_0$ represents the same covariance structure as Θ_0 up to a reordering of the variables. We may use the Cholesky decomposition to write $P_\pi \Theta_0$ uniquely as

$$P_\pi \Theta_0 = (I - L)D^{-1}(I - L)^T = \Theta(L, D), \tag{9}$$

where L is strictly lower triangular and D is diagonal. It follows from Lemma 8 in the Appendix that $P_\pi \Theta(L, D) = \Theta(P_\pi L, P_\pi D)$ for any π, so we can rewrite (9) as

$$\Theta_0 = \Theta(P_{\pi^{-1}} L, P_{\pi^{-1}} D).$$

For each π, define

$$\tilde{B}_0(\pi) := P_{\pi^{-1}} L,$$
$$\tilde{\Omega}_0(\pi) := P_{\pi^{-1}} D.$$

By (6), this gives us the unique decomposition of Θ_0 into a DAG $(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$ that is compatible with the permutation π. The DAGs $(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$ that are compatible with some permutation π define a subset of the equivalence class $\mathcal{E}(\Theta_0)$; it is easy to check that in fact, this subset is the entire equivalence class.

Lemma 1. *Suppose Σ_0 is a positive definite covariance matrix and let $\Theta_0 := \Sigma_0^{-1}$. Then*

$$\begin{aligned}\mathcal{E}(\Theta_0) &= \{(P_{\pi^{-1}} L, P_{\pi^{-1}} D) : P_\pi \Theta_0 = \Theta(L, D),\ \pi \in \mathcal{P}\} \\ &= \{(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi)) : \pi \in \mathcal{P}\}.\end{aligned}$$

Note that the relationship between DAGs and permutations is not bijective: multiple permutations can lead to the same DAG. For example, the trivial DAG with no edges is compatible with all possible permutations.

The question now arises: which DAG $(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$ do we want to estimate? In the presence of experimental data, one may consider issues of causality, in which case each DAG represents a different causal structure. In the absence of such data, however, we can make no such distinctions. All of the DAGs in $\mathcal{E}(\Theta_0)$ are statistically indistinguishable based on observational data alone, so a natural objective is to estimate the DAG that most parsimoniously represents the parameter Θ_0 in the sense that it has the fewest number of edges. This choice can also be motivated as it represents a so-called *minimal I-map*.

Under this assumption, there is an obvious connection between our approach and the *sparse Cholesky factorization* problem: Given a symmetric, positive definite matrix A, find a permutation π such that the Cholesky factor of $P_\pi A$ has the fewest number of nonzero entries possible. In the oracle setting in which we know Θ_0, this is exactly the same problem as finding a permutation π such that $\tilde{B}_0(\pi)$ has the fewest number of edges. This connection has been studied in more detail by Raskutti and Uhler (2014). They show that in this oracle setting, there is an equivalence between ℓ_0-penalized estimation and

sparse Cholesky factorization. In contrast, here we seek to estimate Θ_0 *as well as* find a sparse permutation π, and in this sense we provide a non-oracular, computationally feasible alternative to searching across all $p!$ permutations when p is large.

Example 1. Suppose the DAG B_0 has the structure $X_1 \to X_2 \to X_3$ with edge weights $\beta_{12} = 1$ and $\beta_{23} = 1$, and $\omega_j = 1$ for each j. In this case, we have

$$B_0 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \quad \Omega_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \Theta(B_0, \Omega_0) = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

A topological sort for B_0 is $X_1 \prec X_2 \prec X_3$ (i.e. B_0 is already sorted), but B_0 is lower triangularized by the permutation $\pi_0 = (3, 2, 1)$ that swaps X_1 and X_3. Thus $B_0 = \tilde{B}_0(\pi_0)$.

Now consider another DAG, defined by

$$B_1 = \begin{pmatrix} 0 & 1/2 & 1 \\ 0 & 0 & 0 \\ 0 & 1/2 & 0 \end{pmatrix}, \quad \Omega_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \quad \Theta(B_1, \Omega_1) = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix}.$$

Since $\Theta(B_1, \Omega_1) = \Theta(B_0, \Omega_0)$, the DAG (B_1, Ω_1) is equivalent to (B_0, Ω_0). Thus, according to Lemma 1, there must be a permutation π_1 such that $B_1 = \tilde{B}_0(\pi_1)$ and $\Omega_1 = \tilde{\Omega}_0(\pi_1)$. Indeed, if we let $\pi_1 = (2, 3, 1)$, one can check (by (9)) that these identities hold. Furthermore, if we reverse the order of the variables in π_1, we obtain a topological sort for B_1: $X_1 \prec X_3 \prec X_2$.

This example highlights two important points: (i) For the reader familiar with Markov equivalence of DAGs, it is obvious that B_0 and B_1 are not Markov equivalent, so our definition of equivalence is indeed different; and (ii) Equivalent DAGs in the sense we have defined need not have the same number of edges. This is the primary complication our framework must manage: Amongst all the DAGs which are equivalent to the true parameter Θ_0, we wish to find one which has the fewest number of edges.

2.4 Structural Equation Modeling

We have chosen to focus on the problem of structure estimation of Bayesian networks, which is not to be confused with the problem of causal inference. We view the data-generation mechanism as a multivariate Gaussian distribution as in (1). From this perspective, there are many linear structural equations (2) that may generate (1). Our focus is on finding the most parsimonious representation of the true distribution as a set of structural equations.

Alternatively, one could view the structural equation model (2) as the data-generating mechanism, in which case there is a *particular* set of structural equations that we wish to estimate. This is the perspective commonly adopted in the social sciences and in public health, in which the structural equations model causal relationships between the variables. In this set-up, it is well-known that one cannot expect to recover the directionality of causal relationships based on observational data alone, and the issues of causality, confounding and identifiability take center stage. Since we are only considering observational data, our framework does not address these questions.

3. The Concave Penalization Framework

Now that the necessary preliminaries have been discussed, in the remainder of the paper we will develop the estimation framework thus far described at a high-level. Our approach is to use a penalized maximum likelihood estimator to estimate a sparse DAG B_0 that represents Θ_0. Recall that the negative log-likelihood is given by $L(B, \Omega \,|\, X)$ in (4). This will be our loss function, however in order to promote sparsity and avoid overfitting, we will minimize a penalized loss instead. In what follows, let $p_\lambda : [0, \infty) \to \mathbb{R}$ be a nonnegative and nondecreasing penalty function that depends on the tuning parameter λ and possibly one or more additional shape parameters. Our framework is valid for a general class of penalties, so in what follows we will allow $p_\lambda(\cdot)$ to be arbitrary. The details of choosing the penalty function will be discussed in Section 3.3.

Once p_λ is chosen, one may seek to find a solution to

$$\arg\min_{B,\Omega} \big\{ L(B, \Omega \,|\, X) + n \sum_{i,j} p_\lambda(|\beta_{ij}|) : B \text{ is a DAG} \big\}. \tag{10}$$

When L is taken to be a more general scoring function such as a posterior probability, (10) resembles most familiar score-based methods. When $p_\lambda(\cdot)$ is taken to be the ℓ_0 penalty, we recover the estimator discussed in van de Geer and Bühlmann (2013). Our approach differs from the aforementioned in two ways:

1. Our choice of the penalty term $p_\lambda(\cdot)$ is different from traditional approaches and results in a continuous optimization problem,

2. Due to the nonconvexity of the loss function, we reparameterize the problem in order to obtain a convex loss function.

Thus, in general our estimator will not be the same as (10).

Remark 3. If we further constrain the minimization problem in (10) to include only DAGs which are compatible with a fixed topological sort, we can reduce the problem to a series of p individual regression problems. Given a topological sort $\prec$, the parents of X_j must be a subset of the variables that precede X_j in $\prec$. In terms of the permutation π described in Section 2.3, we require $\Pi_j^0 \subset \{X_k : \pi^{-1}(k) > \pi^{-1}(j)\}$. The true neighbourhood of X_j can then be determined by projecting X_j onto this subset of nodes, which can be done via penalized least squares. Consistency in structure learning and parameter estimation can then be established through standard penalized regression theory.

3.1 Reparameterization

One of the drawbacks of the loss in (4) is that it is nonconvex, which complicates the minimization of the penalized loss. If we minimize (4) with respect to Ω and use the adaptive Lasso penalty, we obtain the estimator described in Fu and Zhou (2013). By keeping the p variance terms, however, we can exploit a clever reparameterization of the problem, introduced in Städler et al. (2010), which leads to a convex loss.

The idea is to define new variables by $\rho_j = 1/\omega_j$ and $\phi_{ij} = \beta_{ij}/\omega_j$, which yields the reparameterized negative log-likelihood

$$L(\Phi, R \,|\, X) = \sum_{j=1}^{p} \left[-n \log(\rho_j) + \frac{1}{2} \|\rho_j x_j - X\phi_j\|^2 \right], \tag{11}$$

where $\Phi = [\phi_1 \,|\, \cdots \,|\, \phi_p]$ and $R = \operatorname{diag}(\rho_1, \ldots, \rho_p)$. The loss function in (11) is easily seen to be convex. Furthermore, if we interpret Φ as the adjacency matrix of a directed graph, then Φ has exactly the same edges and nonzero entries as B, and thus in particular Φ is acyclic if and only if B is acyclic.

In analogy with the parameterization (B, Ω), define

$$\Theta(\Phi, R) = (R - \Phi)(R - \Phi)^T, \tag{12}$$

which gives a formula for the inverse covariance matrix in the parameterization (Φ, R). Note that if $\Phi = \Phi(B, \Omega)$ and $R = R(B, \Omega)$, then $\Theta(B, \Omega) = \Theta(\Phi, R)$, and hence also $L(B, \Omega) = L(\Phi, R)$.

This reparameterization is *not* the same as the likelihood in (7), which is well-known to lead to a convex program (see, for instance, Boyd and Vandenberghe, 2009, §7.1). Indeed, plugging (6) into (7) leads back to (4), which is nonconvex in the parameters β_{ij} and ω_j. To wit, the problem is convex in Θ but not in (B, Ω). The key insight from Städler et al. (2010) is to observe that one may recover convexity by switching to the alternate parameterization in terms of ϕ_{ij} and ρ_j. Unfortunately, the DAG constraint in (10) is still nonconvex. The idea behind this reparameterization is to allow our algorithm to exploit convexity wherever possible in order to reap at least *some* computational and analytical gains. As we shall see, the gains are indeed significant.

3.2 The Estimator

We are now prepared to introduce the formal definition of the DAG estimator which is the focus of this work.

Fix a penalty function $p_\lambda(\cdot)$. Then given

$$\begin{aligned} (\widehat{\Phi}, \widehat{R}) &:= \operatorname*{arg\,min}_{\Phi, R} \big\{ L(\Phi, R \,|\, X) + n \sum_{i,j} p_\lambda(|\phi_{ij}|) : \Phi \text{ is a DAG} \big\} \\ &= \operatorname*{arg\,min}_{\Phi, R} \big\{ \sum_{j=1}^{p} \left[-n \log(\rho_j) + \frac{1}{2} \|\rho_j x_j - X\phi_j\|^2 \right] \\ &\qquad\qquad + n \sum_{i,j} p_\lambda(|\phi_{ij}|) : \Phi \text{ is a DAG} \big\}, \end{aligned} \tag{13}$$

we define our estimator to be

$$(\widehat{B}, \widehat{\Omega}) = \begin{cases} \hat{\beta}_{ij} = \hat{\phi}_{ij}/\hat{\rho}_j, & i \neq j \\ \hat{\beta}_{jj} = 0, & \\ \hat{\omega}_j^2 = 1/\hat{\rho}_j^2, & j = 1, \ldots, p \end{cases} \tag{14}$$

where $\hat{\phi}_{ij}$ and $\hat{\rho}_j$ denote the respective components of $(\widehat{\Phi}, \widehat{R})$. When we wish to emphasize the estimator's dependence on λ, we shall denote it by $(\widehat{\Phi}(\lambda), \widehat{R}(\lambda))$.

There is an intuitive interpretation of the problem in (13): By the identity $L(\Phi, R \,|\, X) = L(\Theta(\Phi, R) \,|\, X)$, it is evident that the loss function for (Φ, R) is simply the negative log-likelihood of the resulting estimate of $\Theta = \Theta(\Phi, R)$. In this sense, we are implicitly approximating the true parameter Θ_0. The key ingredient, however, is the penalty term: We only penalize the edge weights ϕ_{ij}, which has the effect of self-selecting for DAGs which are sparse. In this way, the solution to (13) produces a sparse Bayesian network whose distribution is close to the true, underlying distribution.

Remark 4. For most choices of the penalty, the solution to (13) is *not* the same as the solution to (10) since we are penalizing different terms. In the original parameterization, we penalize the coefficients β_{ij}, whereas after reparameterizing we are penalizing the rescaled coefficients $\phi_{ij} = \beta_{ij}/\omega_j$. Thus we are also penalizing choices of coefficients which overfit the data, i.e., which have small ω_j. A notable exception, however, occurs when $p_\lambda(\cdot)$ is taken to be the ℓ_0 penalty. In this special case, the problems in (10) and (13) are the same, and thus in particular the analysis in van de Geer and Bühlmann (2013) applies.

3.3 Choice of Penalty Function

The standard approach in the Bayesian network literature is to use AIC or BIC to penalize overly complex models, although ℓ_1-based methods have been slowly gaining in popularity. Traditionally, ℓ_1 regularization is viewed as a convex relaxation of optimal ℓ_0 regularization, which results in a convex program that is computationally efficient to solve. Unfortunately, in our situation the constraint that B is a DAG is also nonconvex, so there is little hope to recover a convex program. Thus, there is nothing lost in using concave penalties, which have more attractive theoretical properties than ℓ_1-based alternatives. We will briefly review the details here.

Fan and Li (2001) introduce the fundamental theory of concave penalized likelihood estimation and outline three principles that should guide any variable selection procedure: unbiasedness, sparsity, and continuity. They argue that the following conditions are sufficient to guarantee that a penalized least squares estimator has these properties:

1. (Unbiasedness) $p'_\lambda(t) = 0$ for large t;
2. (Sparsity) The minimum of $t + p'_\lambda(t)$ is positive;
3. (Continuity) The minimum of $t + p'_\lambda(t)$ is attained at zero.

Condition (1) only guarantees unbiasedness for large values of the parameter; in general we cannot expect a penalized procedure to be totally unbiased. Note also that (1-3) imply that p_λ must be a concave function of t.

In the methodological developments which follow, it will not be necessary to assume that the penalty function is concave. The theory developed in Section 4 will illuminate how the properties of the penalty function influence the theoretical properties of the estimator (13, 14), however, the only strict requirement on the penalty function needed for the proposed algorithm is that there exists a corresponding threshold function $S(\cdot, \lambda)$ to perform

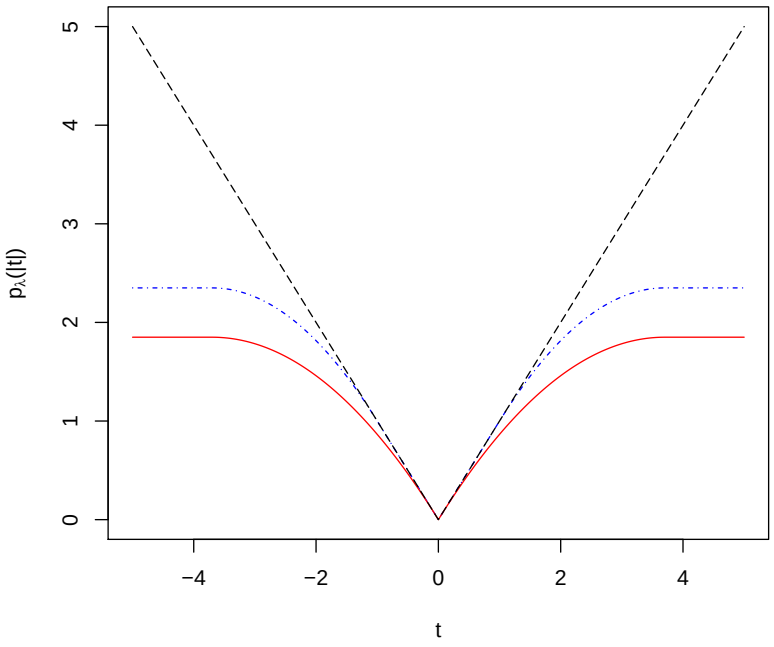


Figure 1: Comparison of penalty functions. The solid red line is the minimax concave penalty (MCP), the dot-dashed blue line is the smoothly clipped absolute deviation penalty (SCAD), and the dashed black line is the ℓ_1 or Lasso penalty. Both the MCP and SCAD represent smooth interpolations of the ℓ_1 and ℓ_0 penalties and hence have better statistical properties, whereas the ℓ_1 penalty exhibits bias due to its divergence as $t \to \infty$.

the single parameter updates (see Section 5.2 for details). Examples of common penalty functions in the literature include ℓ_1 (or Lasso, Tibshirani, 1996), SCAD (Fan and Li, 2001) and MCP (Zhang, 2010). The SCAD penalty represents a smooth quadratic interpolation between the ℓ_1 and ℓ_0 penalties, and the MCP translates the ℓ_1 portion of the SCAD to the origin. See Figure 1 for a visual comparison of these three penalties. The key difference between the ℓ_1 penalty and SCAD or MCP is the flat part of the penalty, which helps to reduce bias.

In our computations we chose to use the MCP, defined for $t \geq 0$ by

$$p_\lambda(t;\gamma) := \lambda\left(t - \frac{t^2}{2\lambda\gamma}\right)1(t < \lambda\gamma) + \frac{\lambda^2\gamma}{2}1(t \geq \lambda\gamma) \tag{15}$$
$$= \begin{cases} \lambda\left(t - \frac{t^2}{2\lambda\gamma}\right), & t < \lambda\gamma, \\ \frac{\lambda^2\gamma}{2}, & t \geq \lambda\gamma. \end{cases}$$

The γ parameter in the MCP controls the concavity of the penalty: As $\gamma \to 0$, MCP approaches the ℓ_0 penalty and as $\gamma \to \infty$, it approaches the ℓ_1 penalty. In the sequel we will thus refer to γ as the *concavity parameter* and λ as the *regularization parameter*. From the above formula, MCP is easily seen to be a quadratic spline between the origin and the

ℓ_0 penalty with a knot at $t = \lambda\gamma$. To demonstrate the differences and potential advantages of a concave penalty, we also implemented our method with the ℓ_1 penalty, $p_\lambda(|t|) = \lambda|t|$.

As the ℓ_1 penalty does not satisfy the unbiasedness condition (Condition (1) above), it yields biased estimates in general. Allowing ourselves to be motivated by some recent developments in regression theory, we can say even more. There the assumptions required for consistency are rather strong and require a so-called *irrepresentability condition* (Zhao and Yu, 2006), also known as *neighbourhood stability* (Meinshausen and Bühlmann, 2006). The bias issues can be circumvented by employing the adaptive Lasso (Zou, 2006), an idea which has been explored in Fu and Zhou (2013). Recent theoretical analysis of regularization with concave penalties has shown that, compared to ℓ_1 penalties, the assumptions on the data needed for consistency can be relaxed substantially. Generalizing these ideas to Bayesian network models, we will show in Section 4 how our estimator is consistent in both parameter estimation and structure learning when concave regularization is used; with ℓ_1 regularization we only obtain parameter estimation consistency. These theoretical results are supported by the comparisons in Section 6.

3.4 The Role of Sparsity

For a given Θ_0, the equivalence class $\mathcal{E}(\Theta_0)$ will typically consist of graphs with different numbers of edges, and in general there need not be a sparse representation $(\tilde{B}_0(\pi), \tilde{\Omega}(\pi))$ with $s_{\tilde{B}_0(\pi)} := \tilde{s}_0(\pi) \ll p^2$. Moreover, the asymptotic theory to be developed in Section 4 will not require such an assumption. When we evaluate our method in Sections 5-7, however, we will focus our attention on the case where there exists a DAG in $\mathcal{E}(\Theta_0)$ which is sparse, that is, satisfying the condition $\tilde{s}_0(\pi) = O(p)$.

Our justification for this assumption is both practical and theoretical. In terms of the true graph, sparsity implies that we expect either (a) only a subset of the variables are truly involved, or (b) on average, each variable has only a few parents. In case (a), estimating a Bayesian network is similar to the variable screening problem. Both of these scenarios are commonly encountered in practice, as many realistic DAG models tend to be sparse in one of these two senses. Moreover, for data sets with p very large, we typically have fewer observations than variables. In fact, we expect $p \gg n$, with p on the order of thousands or tens of thousands. When this happens, we can only expect to obtain reasonable results when each node has at most n parents, although in practice far fewer than n parents is typical. For these reasons, we chose to tailor our algorithm to the sparse, high-dimensional regime. Along with the nonconvexity of the constraint space, this is the main reason for emphasizing the use of concave penalties, whose superior performance in the $p \gg n$ regime has been already established for regression models. Furthermore, by assuming that the true graph is sparse, we can take advantage of several computational enhancements that allow our algorithm to leverage sparsity for speed. The result is an efficient algorithm when we are confident that the underlying model admits a sparse representation.

4. Asymptotic Theory

In this section we provide theoretical justification for the use of the estimator (13, 14) in the finite-dimensional regime. That is, we will assume p is fixed and let $n \to \infty$. The purpose of this section is not to provide novel theoretical insights, but rather simply to

show that under the right conditions we can always guarantee that the estimator defined in the previous section has good estimation properties. Most importantly, we establish that these conditions can always be satisfied when the MCP is used for regularization.

In the statistics literature, a procedure which attains consistency in structure learning with high probability is sometimes referred to as *model selection consistent*. This can be confusing as model selection is also used to refer to the problem of selecting the tuning parameter λ. In the sequel, we use the following conventions: (i) A procedure is *structure estimation consistent* if $P(\text{supp}(\widehat{B}) = \text{supp}(B_0)) \to 1$, (ii) A procedure is *parameter estimation consistent* if $\|\widehat{B} - B_0\|_F \xrightarrow{P} 0$, and (iii) *Model selection* will refer only to the problem of choosing λ.

4.1 Nonidentifiability and Sparsity

Since our optimization problem is nonconvex, we must be careful when discussing "solutions" to (13). The estimator is defined to be the global minimum of the penalized loss, but theoretical guarantees are generally only available for local minimizers. Our theory is no exception, and it is furthermore complicated by identifiability issues: Based on observational data alone, the inverse covariance matrix Θ_0 is identifiable, but the DAG (B_0, Ω_0) is not. The usual theory of maximum likelihood estimation assumes identifiability, but it is possible to derive similar optimality results when the true parameter is nonidentifiable (see for instance Redner, 1981).

When the model is identifiable, one establishes the existence of a consistent local minimizer for the true parameter, which is unique (e.g as in Fan and Li, 2001). It turns out that even if the model is nonidentifiable, we can still obtain a consistent local minimizer for each equivalent parameter. As long as there are finitely many equivalent parameters, these minimizers are unique to each parameter. In particular, in the context of DAG estimation, there are up to $p!$ equivalent parameters in the equivalence class $\mathcal{E}_0$ (Lemma 1). Thus we have a finite collection of local minimizers that serve as "candidates" for the global minimum; the question that remains is which one of these minimizers does our estimator produce?

Each equivalent parameter has the same likelihood, so the only quantity we have to distinguish these minimizers is the penalty term. Our theory will show that by properly controlling the amount of regularization, it is possible to distinguish the *sparsest* DAGs in $\mathcal{E}_0$ in the sense that they will each have strictly smaller penalized loss than their competitors. Moreover, this analysis can be transferred over to the *empirical* local minimizers, so that the sparsest local minimizer has the smallest penalized loss. Because of nonconvexity, however, it is hard to guarantee that these minimizers are the *only* local minimizers, and hence that the sparsest DAGs are the global minimizers. The simulations in Section 6 give us good empirical evidence that our estimator indeed approximates the sparsest DAG representation of Θ_0, as opposed to another DAG with many more edges.

The remainder of this section undertakes the details of this analysis. To stay consistent with the literature, instead of minimizing the penalized loss (13) we will maximize the penalized log-likelihood, which is of course only a technical distinction. We begin with a discussion of the technical results and assumptions which establish the existence of consis-

tent local maximizers before stating our main result in Section 4.3. We also briefly discuss the high-dimensional scenario in which p is allowed to depend on n.

Remark 5. For some classes of models, including nonlinear and non-Gaussian models, the DAG estimation problem considered here is known to be identifiable based on observational data alone (Shimizu et al., 2006; Peters et al., 2012), and some methods have been developed to estimate such models (Hyvärinen et al., 2010; Anandkumar et al., 2013). In contrast to these developments, the main technical difficulty in our analysis is the nonidentifiability of the general Gaussian model.

4.2 Existence of Local Maximizers

In the ensuing theoretical analysis, it will be easier to work with a single parameter vector (vs. the two matrices Φ and R), so we first transform our parameter space in this way without any loss of generality. To the end, define $U := R + \Phi$ and let $\boldsymbol{\nu} = \mathrm{vec}(U) = \mathrm{vec}(R + \Phi) \in \mathbb{R}^{p^2}$ be the vectorized copy of U in $\mathbb{R}^{p^2}$. Our parameter space is then the subset $\mathcal{D}$ of $\mathbb{R}^{p^2}$ such that $\boldsymbol{\nu} \in \mathcal{D}$ implies (Φ, R) is a DAG, where $\boldsymbol{\nu} = \mathrm{vec}(R + \Phi)$. In the sequel, we will refer to such a $\boldsymbol{\nu}$ as a DAG. For a more in-depth treatment of the abstract framework, see Section A.1 in the Appendix.

The true distribution is uniquely defined by its inverse covariance matrix, Θ_0. By equation (12), given $(\widehat{\Phi}, \widehat{R})$ we may consider the resulting estimate of the inverse covariance matrix $\widehat{\Theta} = \Theta(\widehat{\Phi}, \widehat{R})$. By analogy, for any DAG $\boldsymbol{\nu} \in \mathbb{R}^{p^2}$, we may define in the obvious way the matrix $\Theta(\boldsymbol{\nu})$. Thus the parameter $\boldsymbol{\nu}$ is simply another parameterization of the normal distribution: For any Θ_0, there exists $\boldsymbol{\nu} \in \mathcal{D}$ such that $\Theta_0 = \Theta(\boldsymbol{\nu})$. Let $\mathcal{E}_0 = \mathcal{E}(\Theta_0) = \{\boldsymbol{\nu} \in \mathbb{R}^{p^2} : \Theta(\boldsymbol{\nu}) = \Theta_0\}$. We will denote an arbitrary element of $\mathcal{E}_0$ by $\boldsymbol{\nu}_0$ and a minimal-edge DAG in $\mathcal{E}_0$ by $\boldsymbol{\nu}^*$.

As is customary, we denote the support set of a vector by $\mathrm{supp}(\boldsymbol{\nu}) := \{j : \nu_j \neq 0\}$, and likewise for matrices $\mathrm{supp}(B) := \{(i,j) : \beta_{ij} \neq 0\}$. Let $\ell_n(\boldsymbol{\nu} \,|\, X)$ be the unpenalized log-likelihood of the parameter vector $\boldsymbol{\nu}$ and define

$$p_\lambda(\boldsymbol{\nu}) = \sum_{i \neq j} p_\lambda(|u_{ij}|), \tag{16}$$

where u_{ij} denote the elements of U. Note that we are penalizing only the off-diagonal elements of U, which correspond to the elements of Φ. Now let

$$F(\boldsymbol{\nu}) := \ell_n(\boldsymbol{\nu} \,|\, X) - n\, p_{\lambda_n}(\boldsymbol{\nu}). \tag{17}$$

We are interested in maximizing F over $\mathcal{D}$.

For any $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ which represents a DAG $(\Phi_0, R_0) = ((\phi^0_{ij}), (\rho^0_j))$ as described above, define two sequences which depend on the choice of penalty p_λ:

$$a_n(\boldsymbol{\nu}_0) := \max\{|p'_{\lambda_n}(|\phi^0_{ij}|)| : \phi^0_{ij} \neq 0\}, \tag{18}$$

$$b_n(\boldsymbol{\nu}_0) := \max\{|p''_{\lambda_n}(|\phi^0_{ij}|)| : \phi^0_{ij} \neq 0\}. \tag{19}$$

When it is clear from context, the dependence of a_n and b_n on $\boldsymbol{\nu}_0$ will be suppressed. Finally, let $\tau(\lambda) := \sup_t p_\lambda(t)$, which may be infinite. For the MCP we have $\tau(\lambda) = \lambda^2\gamma/2$ and for the ℓ_1 penalty $\tau(\lambda) = +\infty$.

The following result, which is similar in spirit to Theorem 2 of Fu and Zhou (2013), guarantees the existence of a consistent local maximizer:

Theorem 2. *Fix $p \geq 1$. If there exists $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ with $b_n(\boldsymbol{\nu}_0) \to 0$, then there is a local maximizer $\widehat{\boldsymbol{\nu}}_n$ of $F(\boldsymbol{\nu})$ such that*

$$\|\widehat{\boldsymbol{\nu}}_n - \boldsymbol{\nu}_0\| = O_P(n^{-1/2} + a_n(\boldsymbol{\nu}_0)).$$

When $a_n = O(n^{-1/2})$, we obtain a $n^{1/2}$-consistent estimator of $\boldsymbol{\nu}_0$. Note that by Lemma 1, if $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ then $\boldsymbol{\nu}_0 = (\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$ for some permutation π. For this reason, in the sequel we shall refer to the local maximizer $\widehat{\boldsymbol{\nu}}_n$ as the *π-local maximizer* of F for the permutation π. This theorem says that as long as the curvature of the penalty at $(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$ tends to zero, the penalized likelihood has a π-local maximizer that converges to $(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$ as $n \to \infty$.

Under additional assumptions on the penalty function, we may further strengthen this result to include consistency in structure estimation when p remains fixed:

Theorem 3. *Assume that the penalty function satisfies*

$$\liminf_{n\to\infty} \liminf_{t\to 0^+} p'_{\lambda_n}(t)/\lambda_n > 0. \tag{20}$$

Assume further that $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ satisfies $a_n(\boldsymbol{\nu}_0) = O(n^{-1/2})$, $b_n(\boldsymbol{\nu}_0) \to 0$, and let $\widehat{\boldsymbol{\nu}}_n$ be a π-local maximizer from Theorem 2. If $\lambda_n \to 0$ and $\lambda_n n^{1/2} \to \infty$, then

$$P(\operatorname{supp}(\widehat{\boldsymbol{\nu}}_n) = \operatorname{supp}(\boldsymbol{\nu}_0)) \to 1. \tag{21}$$

In fact, this follows immediately from Theorem 2 above and Theorem 2 in Fan and Li (2001). An obvious corollary is that $P(\hat{s}_n = s_0) \to 1$.

We must be careful in interpreting these theorems correctly: They do not imply necessarily that the estimator defined in (13, 14) is consistent. These theorems simply show that under the right conditions, there is a local maximizer of F that is consistent. It remains to establish that the global maximizer of F is indeed one of these local maximizers.

Remark 6. If we assume that the conditions of Theorems 2 and 3 hold for *all* $\boldsymbol{\nu}_0 \in \mathcal{E}_0$, then we can conclude that every equivalent DAG has a π-local maximizer that selects the correct sparse structure. This is trivial since we assume p to be fixed as $n \to \infty$, which allows us to bound the probabilities over all $p!$ choices of $\boldsymbol{\nu}_0$ simultaneously. Since the number of equivalent DAGs grows super-exponentially as p increases, bounding these probabilities when $p = p_n$ grows with n is the main obstacle to achieving useful results in high-dimensions.

The proofs of these two theorems are found in the appendix. In the course of the proofs, we will need the following lemma:

Lemma 4. *If $B_1 \neq B_2$ are DAGs that have a common topological sort, then for any choices of Ω_1 and Ω_2, we have $\Theta(B_1, \Omega_1) \neq \Theta(B_2, \Omega_2)$. A similar result holds in the parameterization (Φ, R).*

The assumption that two DAGs have a common topological sort is equivalent to each DAG being compatible with the same permutation π. The following lemma shows that the $\boldsymbol{\nu}_0$ are isolated, which guarantees that π-local maximizers do not cluster around multiple $\boldsymbol{\nu}_0$. For any $\varepsilon > 0$, we denote the ε-neighbourhood of $\boldsymbol{\nu}_0$ in $\mathcal{D}$ by $B(\boldsymbol{\nu}_0, \varepsilon) := \{\boldsymbol{\nu} \in \mathcal{D} : \|\boldsymbol{\nu} - \boldsymbol{\nu}_0\| < \varepsilon\}$.

Lemma 5. *For any positive definite Θ_0 there exists $\varepsilon > 0$ such that $\mathcal{E}_0 \cap B(\boldsymbol{\nu}_0, \varepsilon) = \{\boldsymbol{\nu}_0\}$ for any $\boldsymbol{\nu}_0 \in \mathcal{E}_0$.*

The proofs of these lemmas are also found in the appendix.

4.3 The Main Result

We will now significantly strengthen Theorems 2 and 3 by showing that, under a concave penalty, a sparsest DAG $\boldsymbol{\nu}^* \in \mathcal{E}_0$ maximizes the penalized likelihood amongst all the possible equivalent representations of the covariance matrix Θ_0. Under the assumptions of Theorem 2, there is a π-local maximizer $\widehat{\boldsymbol{\nu}}_n^*$ of $F(\boldsymbol{\nu})$ such that $\|\widehat{\boldsymbol{\nu}}_n^* - \boldsymbol{\nu}^*\| = O_P(n^{-1/2} + a_n(\boldsymbol{\nu}^*))$. Ideally, when $\boldsymbol{\nu}_0$ has more edges than $\boldsymbol{\nu}^*$, we would like these π-local maximizers to satisfy $F(\widehat{\boldsymbol{\nu}}_n^*) > F(\widehat{\boldsymbol{\nu}}_n)$ with high probability.

Intuitively, when $a_n(\boldsymbol{\nu}_0) = b_n(\boldsymbol{\nu}_0) = 0$, all of the nonzero coefficients lie in the flat part of the penalty where $p'_{\lambda_n}(|\phi_{ij}^0|) = p''_{\lambda_n}(|\phi_{ij}^0|) = 0$. When this happens, the penalty "acts" like the ℓ_0 penalty by penalizing all of the coefficients equally by the amount $\tau(\lambda_n)$, and any DAG with more edges than $\boldsymbol{\nu}^*$ will see a heavier penalty. In order to quantify "how close" $\boldsymbol{\nu}_0$ is to lying in the flat part of the penalty, we define

$$c_n(\boldsymbol{\nu}_0) := \min\{p_{\lambda_n}(|\phi_{ij}^0|) : \phi_{ij}^0 \neq 0\}.$$

When $c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n)$, the penalty mimics the ℓ_0 penalty, and since the likelihood $\ell_n(\boldsymbol{\nu}_0 \,|\, X)$ is constant for all $\boldsymbol{\nu}_0$, we would then have

$$p_{\lambda_n}(\boldsymbol{\nu}^*) < p_{\lambda_n}(\boldsymbol{\nu}_0) \iff \ell_n(\boldsymbol{\nu}^* \,|\, X) - n\, p_{\lambda_n}(\boldsymbol{\nu}^*) > \ell_n(\boldsymbol{\nu}_0 \,|\, X) - n\, p_{\lambda_n}(\boldsymbol{\nu}_0).$$

One would hope that for local maximizers $\widehat{\boldsymbol{\nu}}_n$ that are sufficiently close to the $\boldsymbol{\nu}_0$, the continuity of F would guarantee that this intuition persists. As long as the amount of regularization grows fast enough, this is precisely the case:

Theorem 6. *Suppose that $p_\lambda(t)$ is nondecreasing and concave for $t \geq 0$ with $p_\lambda(0) = 0$. Assume further that the conditions for Theorem 3 hold for all $\boldsymbol{\nu}_0 \in \mathcal{E}_0$. Recall that $\tau(\lambda_n) := \sup_t p_{\lambda_n}(t)$. If*

1. *$c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n) + O(n^{-1/2})$ for all $\boldsymbol{\nu}_0 \in \mathcal{E}_0$,*
2. *$\limsup_n \tau(\lambda_n) < \infty$,*
3. *$\tau(\lambda_n) n^{1/2} \to \infty$,*

then for any DAG $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ with strictly more edges than $\boldsymbol{\nu}^*$, $P(F(\widehat{\boldsymbol{\nu}}_n^*) > F(\widehat{\boldsymbol{\nu}}_n)) \to 1$ as $n \to \infty$.

The restriction to $\boldsymbol{\nu}_0$ with strictly more edges than $\boldsymbol{\nu}^*$ is necessary since $\boldsymbol{\nu}^*$ may not be unique in general. Theorem 6 essentially answers the question of which DAG in the true equivalence class $\mathcal{E}_0$ our estimator approximates. As we have discussed, there is a subtle technicality in which it is possible that there are *other* maximizers of $F(\boldsymbol{\nu})$ besides the π-local maximizers, but this is unlikely in practice.

These theorems provide general technical statements which can be used when weaker assumptions are necessary. By imposing all the conditions in Theorems 2, 3, and 6 uniformly, we can combine all of the results in order to characterize the behaviour of the estimates in terms of the parameterization $(\widehat{B}, \widehat{\Omega})$ given by (14). Before stating the main theorem, we will need some notation to distinguish π-local maximizers. Assuming the conditions of Theorem 2 hold for all π, denote the collection of π-local maximizers by $\mathcal{M}_n$. Continuing our notation from the previous section, we also let (B^*, Ω^*) denote any graph in $\mathcal{E}_0$ with the fewest number of edges, and let $(\widehat{B}^*, \widehat{\Omega}^*)$ be the corresponding π-local maximizer. Recall that given a DAG estimate $(\widehat{B}, \widehat{\Omega})$, we define $\widehat{\Theta} = \Theta(\widehat{B}, \widehat{\Omega})$.

Theorem 7. *Suppose that $p_\lambda(t)$ is nondecreasing and concave for $t \geq 0$ with $p_\lambda(0) = 0$. Fix $p \geq 1$ and assume that the penalty function satisfies*

$$\liminf_{n\to\infty} \liminf_{t\to 0^+} p'_{\lambda_n}(t)/\lambda_n > 0.$$

Assume further that $a_n(\boldsymbol{\nu}_0) = O(n^{-1/2})$, $b_n(\boldsymbol{\nu}_0) \to 0$, and $c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n) + O(n^{-1/2})$ for each DAG in $\mathcal{E}_0$. If $\lambda_n \to 0$, $\lambda_n n^{1/2} \to \infty$, $\limsup_n \tau(\lambda_n) < \infty$, and $\tau(\lambda_n) n^{1/2} \to \infty$, then for any permutation π, there is a local maximizer $(\widehat{B}, \widehat{\Omega})$ of F such that

1. $\|\widehat{B} - \tilde{B}_0(\pi)\|_F + \|\widehat{\Omega} - \tilde{\Omega}_0(\pi)\|_F = O_P(n^{-1/2})$,
2. $P(\operatorname{supp}(\widehat{B}) = \operatorname{supp}(\tilde{B}_0(\pi))) \to 1$,
3. $\|\widehat{\Theta} - \Theta_0\|_F = O_P(n^{-1/2})$.

Furthermore,

$$P\left(F(\widehat{B}^*, \widehat{\Omega}^*) = \max_{(\widehat{B},\widehat{\Omega})\in\mathcal{M}_n} F(\widehat{B}, \widehat{\Omega})\right) \to 1.$$

The proof of Theorem 7 is immediate from the properties of the Frobenius norm and Theorems 2, 3, and 6.

Remark 7. Using an adaptive ℓ_1 penalty, Fu and Zhou (2013) first obtained results similar to Theorems 2 and 3. These results assume a weakened form of faithfulness, however, and require experimental data with interventions in order to guarantee identifiability of the true causal DAG. The results here generalize this theory to observational data without needing faithfulness. The keys to this generalization are the notion of parametric equivalence in (8) (as opposed to Markov equivalence) and the use of a concave penalty to rule out equivalent DAGs with too many edges. The role of concavity is highlighted by the observation that convex penalties cannot satisfy the conditions for Theorem 6.

4.4 Discussion of the Assumptions

The general theme behind the theory described in the previous sections is that as long as the penalty is chosen cleverly enough, there will be a consistent local maximizer for the constrained penalized likelihood problem (13). We pause now to discuss these conditions more carefully, and show that they can always be satisfied.

The parameters $a_n(\boldsymbol{\nu}_0)$ and $b_n(\boldsymbol{\nu}_0)$ measure respectively the maximum slope and concavity of the penalty function, and the conditions on these terms are derived directly from Fan and Li (2001). The idea is that as long as the concavity of the penalty is overcome by the local convexity of the log-likelihood function, our intuition from classical maximum likelihood theory continues to hold true. In order to simultaneously guarantee consistency in parameter estimation and structure learning, it is necessary that these parameters vanish asymptotically.

Furthermore, the assumptions on a_n and b_n in Theorems 2 and 3 highlight the advantages of concave regularization over ℓ_1 regularization. In particular, the ℓ_1 penalty trivially satisfies $b_n \to 0$, but cannot simultaneously satisfy $a_n(\boldsymbol{\nu}_0) = \lambda_n = O(n^{-1/2})$ and $\lambda_n n^{1/2} \to \infty$. Thus, for the ℓ_1 penalty, we may apply Theorem 2 to obtain a local maximizer which is consistent in *parameter estimation*, but we cannot guarantee structure estimation consistency through Theorem 3. In contrast, these conditions are easily satisfied by a concave penalty; in particular they are satisfied when p_λ is the MCP. These observations were first made in Fan and Li (2001).

The conditions on $\tau(\lambda_n)$ in Theorem 6 are more interesting. When the true parameter is identifiable, there is no concern about dominating the penalized likelihood for nonsparse parameters. Since our set-up is decidedly nonidentifiable—there are up to $p!$ choices of the "true" graph—it is essential to control the growth of the penalty, and more specifically, how the penalty grows at the various equivalent DAGs $\boldsymbol{\nu}_0 \in \mathcal{E}_0$. As long as this grows at the right rate, nonsparse graphs will see the penalty term dominate, and as a result the sparsest graph (B^*, Ω^*) emerges as the best estimate of the true graph. Since $\tau(\lambda_n) = +\infty$ for any convex penalty, Theorem 6 along with the remainder of this discussion do not apply to ℓ_1 regularization.

In order to quantify the behaviour of the penalty, we need to control the growth of two different quantities: the maximum penalty $\tau(\lambda_n)$, and the rate of convergence of $c_n(\boldsymbol{\nu}_0)$. By rate of convergence, we refer to the fact that the assumptions on $a_n(\boldsymbol{\nu}_0)$ and $b_n(\boldsymbol{\nu}_0)$ alone require that $c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n) + o(1)$, or equivalently $p_{\lambda_n}(|\phi^0_{ij}|) = \tau(\lambda_n) + o(1)$ whenever $\phi^0_{ij} \neq 0$. The stronger assumption that $c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n) + O(n^{-1/2})$ in Theorem 6 shows that it is not enough that this convergence occurs at an arbitrary rate. One may think of this as a requirement on the zeroth-order convergence of p_{λ_n}, in contrast to the first- and second-order convergence required by Theorems 2 and 3. In practice, it is sufficient to have $c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n)$ for sufficiently large n, and hence also $a_n = b_n = 0$.

Of course, none of this is relevant if we cannot construct a penalty which satisfies all of these conditions simultaneously along with associated regularization parameters λ_n. When the penalty is chosen to be the MCP, all of the conditions required for Theorem 7 are satisfied as long as

$$\limsup_n \lambda_n \gamma_n < \min_{\boldsymbol{\nu}_0 \in \mathcal{E}_0} \min\{|\phi^0_{ij}| : \phi^0_{ij} \neq 0\} \quad \text{and} \quad \lambda_n = O(n^{-\alpha}),\ 0 < \alpha < 1/2. \tag{22}$$

Remark 8. To better understand the conditions on $\tau(\lambda_n)$ in Theorems 6 and 7, it is instructive to consider the simplified case in which the penalty factors as $p_{\lambda_n}(t) = \lambda_n \rho(t)$ for some function $\rho(t)$ (not to be confused with the parameters ρ_j in our model). In this case, the penalty is bounded as long as $\lim_{t\to\infty} \rho(t) < \infty$ and the conditions on $\tau(\lambda_n)$ in Theorem 6 reduce to $\limsup_n \lambda_n < \infty$ and $\lambda_n n^{1/2} \to \infty$. When $\lambda_n \to 0$, these conditions are simply the assumptions in Theorem 3. Thus, the extra conditions on $\tau(\lambda_n)$ in Theorems 6 and 7 are redundant when the penalty factors in this way.

Example 2. Although the usual formula for the MCP does not satisfy the factorization property in Remark 8, we may reparameterize it so that it does. To do this, define a new penalty by

$$\overline{p}_\lambda(t;\delta) := \lambda\left(t - \frac{t^2}{2\delta}\right)1(t<\delta) + \frac{\lambda\delta}{2}1(t\geq\delta), \qquad t \geq 0.$$

Then $\overline{p}_\lambda(t;\delta) = \lambda \cdot \overline{p}_{\lambda=1}(t;\delta)$, and by choosing $\delta = \lambda\gamma$ we may recover the usual formula for the MCP given by (15). Furthermore, the condition in (22) becomes

$$\limsup_n \delta_n < \min_{\boldsymbol{\nu}_0 \in \mathcal{E}_0} \min\{|\phi^0_{ij}| : \phi^0_{ij} \neq 0\},$$

which is independent of λ_n.

4.5 Score-Based Theory in High-Dimensions

The theory in this section so far has assumed that p is fixed with $n > p$, the classical low-dimensional scenario. It would be interesting to obtain results for this method when p is allowed to depend on n, and in particular the case when $p > n$. While the simulations in Section 6 give good empirical evidence that our method is applicable to this scenario, formal theoretical results are not available yet. Here we take a moment to discuss some current work in this direction.

If we fix a permutation π, we have already described in Remark 3 how to modify our method in order to estimate the equivalent DAG that is compatible with π, which we have denoted by $(\tilde{B}_0(\pi), \tilde{\Omega}_0(\pi))$. When the order of the variables is fixed, the problem reduces to standard multiple regression with a concave penalty, in which case Theorems 2 and 3 can be generalized to high-dimensions, for instance using the results in Fan and Lv (2010). This is in the spirit of similar results in the ℓ_1 case obtained by Shojaie and Michailidis (2010). Of course, in our set-up, we do not know in advance which permutation is optimal, so this does not tell the whole story. Theorem 6 shows how our estimator selects the right permutation automatically based on the data, and eliminates the need to assume this prior knowledge.

Recently, van de Geer and Bühlmann (2013) obtained some positive results using ℓ_0 regularization in which it is not assumed that π is known in advance. Under the same Gaussian framework we have adopted in this work, they show the following: When $p_\lambda(t) = \lambda^2 1(t \neq 0)$ and under certain strong regularity conditions, any *global minimizer* of (10) satisfies

$$\|\widehat{B} - \tilde{B}_0(\hat{\pi})\|_F^2 + \|\widehat{\Omega} - \tilde{\Omega}_0(\hat{\pi})\|_F^2 = O_P(\lambda^2 s_0), \tag{23}$$

where $\hat{\pi}$ is the permutation compatible with $(\widehat{B}, \widehat{\Omega})$. Furthermore, they establish that the estimated number of edges are all of the same order: $\hat{s} = O_P(\tilde{s}_0(\hat{\pi})) = O_P(s_0)$. These results represent the first significant analysis of score-based structure learning in high-dimensions that we know of, however, they have some drawbacks. First, they do not guarantee structure estimation consistency, and instead only give an upper bound on the number of estimated edges, which is to be of the same order as a minimal-edge DAG. With respect to computations, these results only hold for the intractable ℓ_0 penalty, and no suggestions are made to allow computation of this estimator in practice. Furthermore, since the optimization problem is nonconvex, theoretical guarantees for global minimizers are less practical than guarantees for local minimizers. We have already observed (Remark 4) that the estimator defined in van de Geer and Bühlmann (2013) is a special case of (13), and so this theory applies to our framework under ℓ_0 regularization.

A common interpretation of concave penalization is as a continuous relaxation of the discrete ℓ_0 penalty. Our framework can thus be seen in this light. Previous work has shown that penalized likelihood estimators can have near optimal performance when compared with the ℓ_0 estimator (Zhang and Zhang, 2012), and thus we have good reason to believe the same holds true for our estimator. The key idea from the analysis in van de Geer and Bühlmann (2013) is to control the behaviour of the estimates over all $p!$ possible permutations, which requires careful analysis using exponential-type concentration inequalities. Based on our preliminary work, we believe that such an analysis can be carried out for more general penalties, however, the details remain to be worked out and are expected to be technical.

Recently there has been some reported progress in high-dimensions for hybrid methods that consist of multiple learning stages. The general outline of these methods is the following:

1. Estimate an initial (undirected, directed, or partially directed) graph $\mathcal{G}_0$,
2. Search for an optimal DAG structure $\widehat{\mathcal{G}}$ subject to the constraint that $\widehat{\mathcal{G}}$ is a subgraph of $\mathcal{G}_0$.

This approach is motivated by the fact that searching for an undirected or partially directed graph in the first step can be substantially faster than searching for a DAG. In this light, Loh and Bühlmann (2013) consider using inverse covariance estimation to restrict the search space, and Bühlmann et al. (2014) convert the problem into three separate steps: preliminary neighborhood selection, order search, and maximum likelihood estimation. Since these ideas use multiple stages, they do not apply directly to the framework developed here.

5. Algorithm Details

Both the objective function and the constraint set in (13) are nonconvex, which makes traditional gradient descent algorithms for performing the necessary minimization inapplicable. One could employ naive gradient descent to find a local minimizer of (13), but it would still be difficult to enforce the DAG constraint. Thus, a different approach must be taken altogether. Extending the algorithm of Fu and Zhou (2013), we employ a cyclic coordinate-descent based algorithm that relies on checking the DAG constraint at each update. By

properly exploiting the sparsity of the estimates and the reparameterization (11), however, we will be able to perform the single parameter updates and enforce the constraint with ruthless efficiency.

5.1 Overview

Before outlining the technical details of implementing our algorithm, we pause to provide a high-level overview of our approach.

The idea behind cyclic coordinate descent is quite simple: Instead of minimizing the objective function over the entire parameter space simultaneously, we restrict our attention to one variable at a time, perform the minimization in that variable while holding all others constant (hereafter referred to as a *single parameter update*), and cycle through the remaining variables. This procedure is repeated until convergence. Coordinate descent is ideal in situations in which each single parameter update can be performed quickly and efficiently. For more details on the statistical perspective on coordinate descent, see Wu and Lange (2008); Friedman et al. (2007).

Moreover, due to acyclicity, we know *a priori* that the parameters ϕ_{kj} and ϕ_{jk} cannot simultaneously be nonzero for $k \neq j$. This suggests performing the minimization in blocks, minimizing over $\{\phi_{kj}, \phi_{jk}\}$ simultaneously. An immediate consequence of this is that we reduce the number of free parameters from p^2 to $p(p-1)/2 + p$, a substantial savings.

In order to enforce acyclicity, we use a simple heuristic: For each block $\{\phi_{kj}, \phi_{jk}\}$, we check to see if adding an edge from $X_k \to X_j$ induces a cycle in the estimated DAG. If so, we set $\phi_{kj} = 0$ and minimize with respect to ϕ_{jk}. Alternatively, if the edge $X_j \to X_k$ induces a cycle, we set $\phi_{jk} = 0$ and minimize with respect to ϕ_{kj}. If neither edge induces a cycle, we minimize over both parameters simultaneously.

Before we outline the details, let us introduce some functions which will be useful in the sequel. Define

$$Q(\Phi, R) := L(\Phi, R) + \sum_{i,j} p_\lambda(|\phi_{ij}|) \tag{24}$$

to be our objective function for coordinate descent. Note that we have suppressed the dependence of the log-likelihood on the data X as well as the dependence of the penalty term on n. In fact, in the computations we may treat n as fixed, so we can absorb this term into the penalty function p_λ. This simply amounts to rescaling the regularization parameter λ, which causes no problems in computing $(\widehat{\Phi}, \widehat{R})$. Thus solving (13) is equivalent to minimizing Q.

Now define the single-variable functions

$$Q_1(\phi_{kj}) = \frac{1}{2}\left\|\rho_j x_j - \sum_{i=1}^{p} \phi_{ij} x_i\right\|^2 + p_\lambda(|\phi_{kj}|), \tag{25}$$

$$Q_2(\rho_j) = -n \log \rho_j + \frac{1}{2}\left\|\rho_j x_j - \sum_{i=1}^{p} \phi_{ij} x_i\right\|^2. \tag{26}$$

The function Q_1 is $Q(\Phi, R)$ in (24) considered as a function of the single parameter ϕ_{kj}, while holding the other $p^2 - 1$ variables fixed and ignoring terms that do not depend on ϕ_{kj},

Algorithm 1 CCDr Algorithm

Input: Initial estimates (Φ^0, R^0); penalty parameters (λ, γ); error tolerance $\varepsilon > 0$; maximum number of iterations M.

1. Cycle through ρ_j for $j = 1, \ldots, p$, minimizing Q_2 with respect to ρ_j at each step.
2. Cycle through the $p(p-1)/2$ blocks $\{\phi_{kj}, \phi_{jk}\}$ for $j, k = 1, \ldots, p$, $j \neq k$, minimizing with respect to each block:
   (a) If $\phi_{kj} \Leftarrow 0$, then minimize Q_1 with respect to ϕ_{jk} and set $(\phi_{kj}, \phi_{jk}) = (0, \phi^*_{jk})$, where $\phi^*_{jk} = \arg\min Q_1(\phi_{jk})$;
   (b) If $\phi_{jk} \Leftarrow 0$, then minimize Q_1 with respect to ϕ_{kj} and set $(\phi_{kj}, \phi_{jk}) = (\phi^*_{kj}, 0)$, where $\phi^*_{kj} = \arg\min Q_1(\phi_{kj})$;
   (c) If neither 2(a) nor 2(b) applies, then choose the update which leads to a smaller value of Q.
3. Repeat steps 1 and 2 l times, until either $\max_{j,k} |\phi^{(l-1)}_{kj} - \phi^{(l)}_{kj}| < \varepsilon$ or $l > M$.
4. Transform the final estimates $(\widehat{\Phi}, \widehat{R})$ back to the original parameter space $(\widehat{B}, \widehat{\Omega})$ (see equation (14)) and output these values.

and Q_2 is the corresponding function for the parameter ρ_j. We express the dependence of Q_1 and Q_2 on k and/or j implicitly through their respective argument, ϕ_{kj} or ρ_j.

An overview of the algorithm is given in Algorithm 1. We use the notation $\phi_{kj} \Leftarrow 0$ to mean that ϕ_{kj} must be set to zero due to acyclicity, as outlined above. The remainder of this section is devoted to the details of implementing the above algorithm, which we call Concave penalized Coordinate Descent with reparameterization (CCDr).

5.2 Coordinate Descent

In what follows, we assume that the data have been appropriately normalized so that each column x_j has unit norm, $\|x_j\|^2 = \sum_h x^2_{hj} = 1$. Furthermore, although the details of the algorithm do not depend on the choice of penalty, we will focus on the MCP and ℓ_1 penalties, as these are the methods implemented and discussed in Sections 6 and 7.

5.2.1 Update for ϕ_{kj}

Mazumder et al. (2011) show that the minimum of (25) can be found by solving

$$\arg\min_{\beta} Q^1(\beta), \quad \text{where } Q^1(\beta) := \frac{1}{2}(\beta - \tilde{\beta})^2 + p_\lambda(|\beta|). \tag{27}$$

The solution to (27) is given by a so-called threshold function which is associated to each choice of penalty. For the MCP with $\gamma > 1$ this is defined by

$$S_\gamma(\tilde{\beta}, \lambda) = \begin{cases} 0, & |\tilde{\beta}| \leq \lambda, \\ \operatorname{sgn}(\tilde{\beta}) \left(\frac{|\tilde{\beta}| - \lambda}{1 - 1/\gamma} \right), & \lambda < |\tilde{\beta}| \leq \lambda\gamma, \\ \tilde{\beta}, & |\tilde{\beta}| > \lambda\gamma. \end{cases} \tag{28}$$

For the ℓ_1 penalty, we have

$$S(\tilde{\beta}, \lambda) = \begin{cases} 0, & |\tilde{\beta}| \leq \lambda, \\ \operatorname{sgn}(\tilde{\beta})(|\tilde{\beta}| - \lambda), & |\tilde{\beta}| > \lambda. \end{cases} \tag{29}$$

To see how to convert (25) into (27), note that

$$\begin{aligned} Q_1(\phi_{kj}) &= \frac{1}{2} \sum_{h=1}^n \left(\rho_j x_{hj} - \sum_{i \neq k} \phi_{ij} x_{hi} - \phi_{kj} x_{hk} \right)^2 + p_\lambda(|\phi_{kj}|) \qquad (30) \\ &= \frac{1}{2} \sum_{h=1}^n x_{hk}^2 \left(\frac{1}{x_{hk}} r_{kj}^{(h)} - \phi_{kj} \right)^2 + p_\lambda(|\phi_{kj}|), \end{aligned}$$

where $r_{kj}^{(h)} := \rho_j x_{hj} - \sum_{i \neq k} \phi_{ij} x_{hi}$. Expanding the square in the last line and using $\sum_h x_{hk}^2 = 1$,

$$\begin{aligned} Q_1(\phi_{kj}) &= \frac{1}{2} \left\{ \sum_{h=1}^n (r_{kj}^{(h)})^2 - 2\phi_{kj} \sum_{h=1}^n x_{hk} r_{kj}^{(h)} + \phi_{kj}^2 \right\} + p_\lambda(|\phi_{kj}|) \qquad (31) \\ &= \frac{1}{2} \left(\phi_{kj} - \sum_{h=1}^n x_{hk} r_{kj}^{(h)} \right)^2 + p_\lambda(|\phi_{kj}|) + \text{const.} \qquad (32) \end{aligned}$$

The constant term in (32) does not depend on ϕ_{kj} and hence does not affect the minimization of Q_1. Thus minimizing $Q_1(\phi_{kj})$ is equivalent to minimizing $Q^1(\beta)$ in (27) with $\tilde{\beta} = \sum_h x_{hk} r_{kj}^{(h)}$. Hence for MCP with $\gamma > 1$,

$$\arg\min Q_1(\phi_{kj}) = S_\gamma \left(\sum_h x_{hk} r_{kj}^{(h)}, \lambda \right), \tag{33}$$

and similarly for the ℓ_1 penalty. The existence of a closed-form solution to the single parameter update for ϕ_{kj} is a key ingredient to our method, and is one of the reasons we chose the MCP and ℓ_1 penalties in our comparisons. Many other penalty functions, however, allow for closed-form solutions to (27), and our algorithm applies for any such penalty function.

5.2.2 Update for ρ_k

The single parameter update for ρ_j is straightforward to compute and is given by

$$\arg\min Q_2(\rho_j) = \frac{c + \sqrt{c^2 + 4n}}{2}, \quad \text{with } c = \sum_{i \neq j} \phi_{ij} \sum_h x_{hi} x_{hj}. \tag{34}$$

Since $Q_2(\rho_j)$ is a strictly convex function, this is the only minimizer.

5.3 Regularization Paths

In practice, it is difficult to select optimal choices of the penalty parameters (λ, γ) in advance. Thus it is necessary to compute several models at many discrete choices of (λ_i, γ_j), and then perform model selection. In testing, we observed a dependence on the concavity parameter γ, however, for simplicity we will consider γ fixed in the sequel, and postpone further study of the method's dependence on γ to future work.

The regularization parameter λ, on the other hand, has a strong effect on the estimates. In particular, as $\lambda \to \infty$, $\widehat{\Phi}(\lambda) \to \mathbf{0}$, and as $\lambda \to 0$ we obtain the unpenalized maximum likelihood estimates. It is thus desirable to obtain a sequence of estimates $(\widehat{\Phi}(\lambda_i), \widehat{R}(\lambda_i))$ for some sequence $\lambda_i > \lambda_{i+1} > 0$, $i = 0, 1, \ldots, L$. In practice, we will always choose λ_0 so that $\widehat{\Phi}(\lambda_0) = \mathbf{0}$, with successive values of λ_i decreasing on a linear scale. One can easily check that if we use an initial guess of $\Phi^0 = \mathbf{0}$, then the choice $\lambda_0 = n^{1/2}$ ensures that the null model is a local minimizer of Q.

Once we have estimated a sequence of models $(\widehat{\Phi}(\lambda_i), \widehat{R}(\lambda_i))$, $i = 0, 1, \ldots, L$, we must choose the best model from these $L + 1$ models. This is the model selection problem, and is beyond the scope of this paper. The present work should be considered a "proof of concept," showing that under the right conditions, there exists a λ that estimates the true DAG with high fidelity. The problem of correctly selecting this parameter is left for future work, but some preliminary empirical analysis is provided in Section 6.5. See Wang et al. (2007) for some positive results concerning the SCAD penalty, and Fu and Zhou (2013) for a relevant discussion of some difficulties that are idiosyncratic to structure estimation of BNs. In particular, it is worth re-emphasizing here that cross-validation is suboptimal, and should be avoided.

5.4 Implementation Details

As presented so far, the CCDr algorithm is not particularly efficient. Fortunately, there are several computational enhancements we can exploit to greatly improve the efficiency of the algorithm. Many of these ideas are adapted from Friedman et al. (2010), and the reader is urged to refer to this paper for an excellent introduction to coordinate descent for penalized regression problems.

In implementing the CCDr algorithm, we use warm starts and an active set of blocks as described in Friedman et al. (2010); Fu and Zhou (2013). We also use a sparse implementation of the parameter matrix Φ to speed up internal calculations. Naive recomputation of the n weighted residual factors $r_{kj}^{(h)}$ for $h = 1, \ldots, n$ for every update incurs a cost of $O(np)$ operations, which is prohibitive in general, and is the main bottleneck in the algorithm.

Friedman et al. (2010) observe that this calculation can be reduced to $O(p)$ operations by noting that the sum in (33) can be written as

$$\sum_{h=1}^{n} x_{hk} r_{kj}^{(h)} = \rho_j \langle x_j, x_k \rangle - \sum_{i \neq k} \phi_{ij} \langle x_i, x_k \rangle. \tag{35}$$

The inner products above do not change as the algorithm progresses, and hence can be computed once at a cost of $O(n^2 \log n)$ operations. This is a substantial improvement over several million $O(np)$ computations, which is typical for large p.

Similar reasoning applies to the computation of (34), which highlights why the reparameterization (11) is useful: the single parameter update for each ρ_j only requires $O(p)$ operations, compared with $O(p^2)$ required operations for the standard residual estimate for ω_j^2 in the original parameterization. Since we perform p of these updates in each cycle, we reduce the total number of operations per cycle from $O(p^3)$ down to $O(p^2)$, which is a substantial savings. Moreover, by leveraging sparsity, both (33) and (34) become $O(1)$ calculations when the maximum number of parents per node is bounded.

As stated, our algorithm will take a pre-specified sequence of λ-values and compute an estimate $(\widehat{\Phi}(\lambda_i), \widehat{R}(\lambda_i))$ for all $L+1$ choices of λ_i. In general, we do not know in advance what the smallest value of λ appropriate for the data is, and we typically choose λ_L as some small value. Since the model complexity (in terms of the number of edges) increases as λ decreases, more and more time is spent computing complex models for small λ. We can exploit these facts in order to avoid wasting time on computing unnecessarily complex models. As the algorithm proceeds calculating estimates for each λ_i, if the estimated number of edges $\hat{s}_i := s_{\widehat{B}(\lambda_i)}$ is too large, we know that we need not continue computing new models for smaller λ. We can justify this as follows: *either* the true model is sparse, in which case we know that complex models with $\hat{s}_i$ large can be ignored, *or* the true model is *not* sparse, in which case our algorithm is less competitive. Thus, in this sense, prior knowledge or intuition of the sparsity of the true model is needed. In practice, we implement this by halting the algorithm whenever $\hat{s}_i > \alpha p$, where $\alpha > 0$ is a pre-specified parameter. While the choice of α should be application driven, we will use $\alpha = 3$ unless reported otherwise. In the sequel, α shall be referred to as the *threshold parameter*.

5.5 Full Algorithm

A complete, detailed description of the algorithm is given in Algorithm 2, including the implementation details discussed in the previous section. We refer to steps (1-2) of Algorithm 1 as a single "sweep" of the algorithm (i.e. performing a single parameter update for every parameter in the active set).

Finally, note that it is trivial to adapt the *SparseNet* procedure from Mazumder et al. (2011) to our algorithm in order to compute a *grid* of estimates

$$(\widehat{\Phi}(\lambda_i, \gamma_j), \widehat{R}(\lambda_i, \gamma_j)), \quad i = 0, \dots, L,\ j = 0, \dots, J,$$

if one wishes to adjust the γ parameter in addition to λ.

Algorithm 2 Full CCDr Algorithm

Input: Initial estimates (Φ_0^0, R_0^0); sequence of regularization parameters $\lambda_0 > \lambda_1 > \cdots > \lambda_L$; concavity parameter $\gamma > 1$; error tolerance $\varepsilon > 0$.

1. Normalize the data so that $\|x_j\|^2 = 1$ and compute the inner products $\langle x_i, x_j \rangle$ for all $i, j = 1, \ldots, p$.
2. For each λ_i:
    1. If $i > 0$, set $(\Phi_i^0, R_i^0) = (\widehat{\Phi}(\lambda_{i-1}), \widehat{R}(\lambda_{i-1}))$.
    2. Perform a full sweep of all parameters using (Φ_i^0, R_i^0) as initial values, and identify the active set.
    3. Sweep over the active set l times, until either $\max_{j,k} |\phi_{kj}^{(l-1)} - \phi_{kj}^{(l)}| < \varepsilon$ or $l > M$.
    4. Repeat (2-3) m times (using the current estimates as initial values) until the active set does not change, or $m > M$.
    5. If $\hat{s}_i > \alpha p$, then halt the algorithm. If not, continue by computing $(\widehat{\Phi}(\lambda_{i+1}), \widehat{R}(\lambda_{i+1}))$.
3. Transform the final estimates $(\widehat{\Phi}(\lambda_i), \widehat{R}(\lambda_i))$ back to the original parameter space $(\widehat{B}(\lambda_i), \widehat{\Omega}(\lambda_i))$ (see equation (14)) and output these values.

6. Numerical Simulations and Results

In order to assess the accuracy and efficiency of the CCDr algorithm, we compared our algorithm with four other well-known structure learning algorithms: the PC algorithm (Spirtes and Glymour, 1991), the max-min hill-climbing algorithm (MMHC; Tsamardinos et al., 2006), Greedy Equivalent Search (GES; Chickering, 2003), and standard greedy hill-climbing (HC). This selection was based on a pre-screening in which we compared the performance of several more algorithms in order to select those which showed the best performance in terms of accuracy and efficiency, and is by no means intended to be exhaustive. We were mainly interested in the accuracy and timing performance of each algorithm as a function of the model parameters (p, s_0, n). Details on the implementations used and our experimental choices will be discussed in Section 6.1.

Our comparisons thus consist of two score-based methods (GES, HC), one constraint-based method (PC), and one hybrid method (MMHC). For brevity, in the ensuing discussion we will frequently refer to both PC and MMHC as constraint-based methods since both methods employ some form of constraint-based search whereas GES and HC do not. In order to compare the effects of regularization, we also compared each of these algorithms to two implementations of CCDr: One using MCP as the penalty (CCDr-MCP), and a second with the ℓ_1 penalty (CCDr-ℓ_1). This gives us a total of six algorithms overall. To offer a sense of scale, the experiments in this section total over $140,000$ individual DAG estimates for almost 1,000 "gold-standard" DAGs.

We begin with a comprehensive evaluation in low-dimensions ($n \geq p$) of all six algorithms using randomly generated DAGs, the main purpose of which is to show that hill-climbing and GES are significantly slower and less accurate in comparison with the other approaches. This supports our first claim that CCDr represents a clear improvement over existing score-based methods. We then move onto a similar assessment for high-dimensional data, which will show the advantages of our method over the constraint-based methods when sample sizes are limited and the number of nodes increases. Once this has been done, we show that our method scales efficiently on graphs with up to 2000 nodes as well as discuss some issues related to model selection and timing. We conclude this section with some detailed discussions about our experiments.

6.1 Experimental Set-Up

All of the algorithms were implemented in the `R` language for statistical computing (R Core Team, 2014). For the PC and GES algorithms, we used the `pcalg` package (version 2.0-3, Kalisch et al., 2012), and for the MMHC and HC algorithms we used the `bnlearn` package (version 3.6, Scutari, 2010). Both packages employ efficient, optimized implementations of each algorithm, and were updated as recently as July 2014. At the time of the experiments, these were the most up-to-date publicly available versions of either package. All of the tests were performed on a late 2009 Apple iMac with a 2.66GHz Intel Core i5 processor and 4GB of RAM, running Mac OS X 10.7.5.

For all the experiments described in this section, DAGs were randomly generated according to the Erdös-Renyi model, in which edges are added independently with equal probability of inclusion. In each experiment, an array of values were chosen for each of the three main parameters: p, s_0, and n. For every possible combination of (p, s_0, n), N individual tests were then run with these parameters fixed. For each test, a DAG was randomly generated using the `pcalg` function `randomDAG` with p nodes and s_0 expected edges, and then n random samples were generated using the function `rmvDAG`, according to the structural model (2). For tests involving different choices of the sample size, the same DAG was used for each choice of n to generate data sets of different sizes. Since the edges were selected at random, the simulated DAGs did not have *exactly* s_0 edges, but instead s_0 edges on average. For each simulation, the nonzero coefficients β^0_{ij} were chosen randomly and uniformly from the interval $[0.5, 2]$ and the error variances were fixed at $\omega^0_j = 1$ for all j.

With the exception of HC and GES, each algorithm has a tuning parameter which strongly affects the accuracy of the final estimates. For CCDr, this is λ, which controls the amount of regularization, and for PC and MMHC it is α, the significance level. In order to study the dependence of each algorithm on these parameters, we chose a sequence of parameters to use for each algorithm. For CCDr, we used a linear sequence of 20 values, starting from $\lambda_{\max} = n^{1/2}$. For both PC and MMHC, we used

$$\alpha \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\}.$$

Our choices for α were motivated by the recommendations in Kalisch and Bühlmann (2007) and Tsamardinos et al. (2006), respectively, as well as by computational concerns: It was necessary to use a much smaller sequence for these algorithms since their running times are significantly longer than CCDr. Furthermore, we found that setting $\alpha < 0.0001$ results in

estimates with too few edges, and setting $\alpha > 0.05$ can lead to runtimes well in excess of 24 hours.

When using the MCP, we must also select the concavity parameter γ in addition to λ. In order to keep our experiments constrained to a reasonable size, we elected not to study the effect of this parameter in detail. Based on the extensive evaluations in Zhang (2010), we chose $\gamma = 2$, which was supported by internal tests to gauge the effect of this parameter. This value represents a fair balance between convexity ($\gamma \to \infty$) and complexity ($\gamma \to 0$). The CCDr algorithm also has three other user-specific parameters: ε, M, and α. Based on our simulations, ε and M have a minimal impact on the accuracy of the estimates, and can simply be chosen to be small and large respectively. The default parameters we used in these simulations were: $\varepsilon = 10^{-4}$, $M = p^{1/2} \vee 10$, and $\alpha = 3$. Recall that in the full algorithm (Algorithm 2), for each λ_i there are at most $M^2 = p \vee 100$ sweeps. When p is small a maximum of 100 iterations is more than enough.

Remark 9. Traditionally, the PC algorithm produces either a skeleton or a CPDAG, depending on how many phases of the algorithm are run (for the definition of a CPDAG and its relation to the PC algorithm, see Kalisch and Bühlmann, 2007). As discussed in Rütimann and Bühlmann (2009), however, it is possible to orient a DAG given its CPDAG using the function `pdag2dag` from the `pcalg` package. This works well in practice, although we found that in some cases the provided method was not able to orient the edges in the CPDAG successfully. In this case, we were able to compare skeletons but not DAGs for the PC algorithm. In the analysis, we treated this situation agnostically by ignoring such problematic estimates and entering them as missing values in the final analysis. This situation arose in less than 5% of cases, so it was not a significant issue.

6.2 Performance Metrics

Our emphasis will be on the performance of each algorithm with respect to structure learning; that is, how well each algorithm reconstructs the DAG which is used to generate the data. Thus for every estimated structure, we compare both the final oriented DAG and its skeleton (i.e. the undirected graph that results by ignoring the directionality of the edges) to those of the true DAG. For a directed graph, we distinguish between *true edges* (or *true positives*)—edges which are estimated with the correct orientation—and *reversed edges*—edges which are in the skeleton but have the wrong direction. No such distinction can be made for the skeletons, of course. A *false positive* is any edge—regardless of directionality—which is not in the skeleton of the true graph.

We gauge the performance of the algorithms on the following metrics:

1. P = number of estimated (predicted) edges,
2. TP = number of true positives,
3. R = number of reversed edges,
4. FP = number of false positives,
5. SHD of the estimated DAG,

6. SHD of the estimated skeleton,
7. Test-data log-likelihood,
8. Test-data BIC,
9. Total and average running time in seconds.

SHD refers to the structural Hamming distance, which measures the minimum number of edge reversals, additions, and/or removals necessary to convert an estimated graph into the true graph. This is a useful metric since it gives an absolute sense of "how far" away the estimates are from the true graph. For the precise definition of the structural Hamming distance, see Tsamardinos et al. (2006). Also, in order to compute the log-likelihood and BIC, it is necessary to estimate the parameters given the estimated structures, which we did by simple ordinary linear regression. As p increases the time to compute these parameters becomes burdensome, and so comparisons of the log-likelihood and BIC were only performed for the low-dimensional experiments with $p \leq 200$. While our primary concern in these evaluations is accuracy in structure learning, these two metrics give us a sense of the implied parameter estimation consistency.

We will also sometimes refer to the following common normalizations of the above metrics:

1. False discovery rate (FDR) $= (R + FP)/P$,
2. True positive rate (TPR) $= TP/T$,
3. False positive rate (FPR) $= (R + FP)/F$,

Here, T is number of edges in the true graph and $F = \frac{1}{2}p(p-1) - T$ is the number of edges absent from the true graph. In some literature, the complement of the false discovery rate (i.e. $1 - \text{FDR}$) is sometimes called *specificity*, while TPR is also variously called *recall* or *sensitivity*.

Finally, when comparing the timing data it is important to recall that each algorithm computes a different number of estimates: HC and GES only produce one, the implementations of PC and MMHC used here produce exactly six, and both CCDr approaches produce up to 20 estimates. Thus it is necessary to consider both the total running time for each algorithm as well as the average time per estimate, which gives a better sense of the computational complexity of each approach. In the sequel, the *total runtime* is defined as the real processor time required to run an algorithm over a full sequence of tuning parameters, and the *average runtime* is defined as the total runtime divided by the number of graphs estimated, i.e., the number of tuning parameters in the sequence.

6.3 Experiments on Random Graphs

In this section we provide detailed results comparing the performance of each algorithm on randomly generated DAGs, across a wide range of choices of (p, s_0, n), using the metrics described in Section 6.2.

In order to properly compare the algorithms, a single model needed to be selected from each sequence of estimates generated by each algorithm. To keep things simple, and since

we have not considered a theoretical analysis of consistent model selection, we simply chose the most accurate model produced by each algorithm by selecting the DAG estimate with the smallest SHD. While this may seem artificial, it provides a good assessment of the potential of each approach. This choice of model selection results in DAGs with somewhat low sensitivity, but nonetheless it still provides a consistent method of comparing the performance of different algorithms. In Section 6.5 we will discuss some interesting issues related to model selection.

6.3.1 Low-Dimensions

We first generated relatively small random graphs along with low-dimensional data sets according to the following settings:

- $p \in \{50, 100, 200\}$;
- $s_0/p \in \{0.2, 0.5, 1.0, 2.0\}$;
- $n/p \in \{1, 5\}$;
- Algorithms: CCDr-MCP, CCDr-ℓ_1, GES, HC, MMHC, PC.

For all combinations of (p, s_0, n), we ran $N = 50$ tests each. The result was 600 random DAGs, 1200 data sets, and 86,400 individual estimates across all six algorithms tested.

The results are shown in Table 1 and Figure 2. For each p, the results are averaged over all 50 tests and each value of s_0 and n. In the low-dimensional regime, it is expected that constraint-based algorithms will show good performance as the statistical tests on which they rely are more reliable and consistent when $n \geq p$. As expected, in our experiments, both PC and MMHC produced the most accurate results in this setting (Table 1). This is further substantiated by the seemingly counterintuitive observation that the performance of both algorithms improves as p increases; this is explained by recalling that n also increases as p increases, so for larger p the statistical tests also have increased power.

The score-based algorithms GES and HC, on the other hand, easily perform the worst in terms of structure learning: these algorithms include far too many edges and as a result obtain high sensitivity but also high false discovery rates. For example, when $p = 200$ and the simulated DAGs had 185 edges on average, both HC and GES estimate well over 500 edges, almost three times the true number, and exhibit false discovery rates greater than 70%. Notwithstanding, GES does noticeably outperform HC, which was anticipated.

Both CCDr methods fall in the middle, with CCDr-MCP outperforming CCDr-ℓ_1 by a few edges in each case. Both methods estimate fewer edges than their score-based competitors—150 and 140 edges respectively when $p = 200$—but slightly more than the constraint-based methods, which estimate 135 edges (PC) and 129 edges (MMHC). This shows that CCDr represents a clear improvement over both GES and HC, and this is even without consideration of efficiency, which we will discuss shortly (Section 6.3.3).

The results for the test-data log-likelihood and the BIC score highlight several difficulties with existing methods which the proposed methods help to overcome. GES and HC both show higher log-likelihood than the others, and since the results are computed based on *test data*, this cannot be attributed to overfitting. What's more, even though both methods produce far more edges than the others, they each only estimate roughly 3 edges per node,

$p = 50$, $T = 46.48$	CCDr-MCP	CCDr-ℓ_1	GES	HC	MMHC	PC
P	26.50	22.98	109.83	113.78	26.46	26.39
TP	14.35	11.86	**33.20**	27.49	15.88	16.64
R	8.38	**7.96**	8.19	12.29	9.14	8.26
FP	3.78	3.15	68.44	74.00	**1.44**	1.48
SHD (DAG)	35.92	37.77	81.72	92.99	32.04	**31.32**
SHD (skeleton)	27.54	29.81	73.53	80.69	**22.89**	23.06
TPR	0.31	0.26	**0.71**	0.59	0.34	0.36
FDR	0.46	0.48	0.70	0.76	0.40	**0.37**
$p = 100$, $T = 91.48$	CCDr-MCP	CCDr-ℓ_1	GES	HC	MMHC	PC
P	67.14	60.32	241.71	256.20	60.97	60.33
TP	36.40	30.85	**74.30**	60.24	39.03	39.85
R	18.95	19.87	**12.90**	23.16	18.71	17.33
FP	11.79	9.60	154.51	172.81	3.22	**3.15**
SHD (DAG)	66.86	70.23	171.69	204.05	55.67	**54.78**
SHD (skeleton)	47.91	50.36	158.79	180.88	**36.95**	37.45
TPR	0.40	0.34	**0.81**	0.66	0.43	0.44
FDR	0.46	0.49	0.69	0.76	0.36	**0.34**
$p = 200$, $T = 185.06$	CCDr-MCP	CCDr-ℓ_1	GES	HC	MMHC	PC
P	150.44	140.51	553.78	591.55	134.72	128.73
TP	83.60	73.28	**158.38**	127.69	90.74	89.23
R	39.05	42.58	**22.35**	45.65	37.59	34.28
FP	27.79	24.65	373.06	418.21	6.39	**5.22**
SHD (DAG)	129.24	136.43	399.74	475.58	100.70	**96.69**
SHD (skeleton)	90.19	93.86	377.39	429.93	**63.12**	65.25
TPR	0.45	0.40	**0.86**	0.69	0.49	0.48
FDR	0.44	0.48	0.71	0.78	0.33	**0.31**

Table 1: Average estimation performance of algorithms in low-dimensions.

which is further evidence that these methods are not necessarily overfitting. Rather, going back to (7), we see that the log-likelihood is a function of Θ alone, which means the test-data log-likelihood is not influenced by the accuracy of the graph structure estimated by an algorithm. This results in two distinct issues in evaluating algorithms on the basis of test-data log-likelihood:

- Even if $\|\widehat{\Theta} - \Theta_0\|_F$ is small, i.e. $\widehat{\Theta}$ is a good estimate of the true parameter, the estimated equivalence class can still be different from the true equivalence class;
- Even if the equivalence class is correctly estimated, the chosen representation may not be the sparsest.

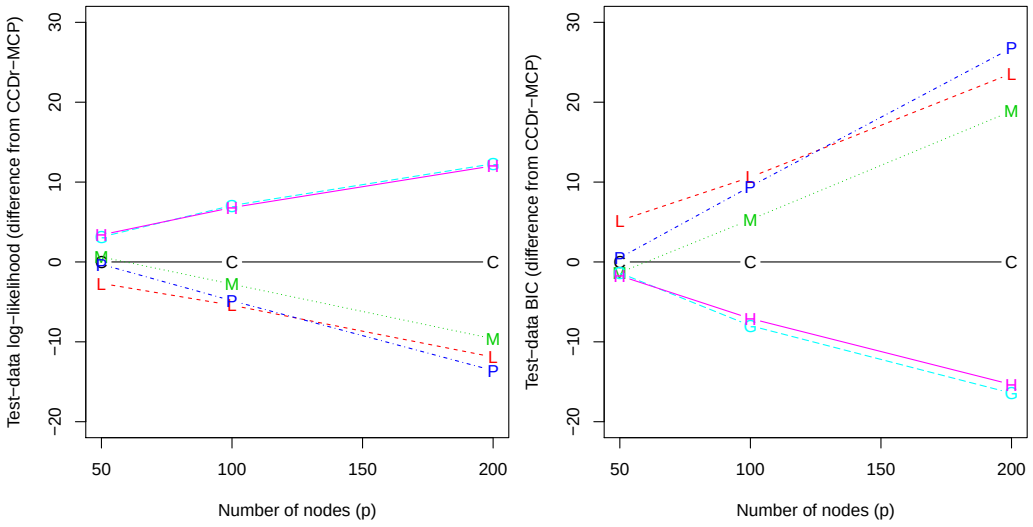


Figure 2: Comparison of test-data log-likelihood and BIC scores (low dimensions). The data are presented relative to the scores for CCDr-MCP. For log-likelihood, larger scores (positive values in the plot) are better; for BIC smaller scores (negative values) are better. (C = CCDr-MCP, L = CCDr-ℓ_1, G = GES, H = HC, M = MMHC, P = PC)

This explains why GES and HC perform the best on this metric: They do a good job of estimating Θ_0, as opposed to a sparse Bayesian network. By contrast, the constraint-based methods do not use the log-likelihood at all and thus exhibit the worst generalization in terms of log-likelihood. For methods which estimate approximately the same number of edges, CCDr-MCP is optimal, falling in between the score-based and constraint-based approaches (Figure 2). A similar discussion applies to the BIC scores, with the added complication of the BIC penalty. The fact that GES and HC still perform the best with respect to BIC—in spite of estimating far too many edges—underscores the fact that the BIC penalty is too lenient for estimating DAGs. This observation is further substantiated and discussed in more detail in Section 6.5.

6.3.2 High-Dimensions

In this section we use the same random set-up as in the previous section, however, our focus is now on high-dimensional estimation. Both HC and GES were omitted in this experiment because of their poor performance—both in terms of accuracy and timing—in the low-dimensional setting. This allowed us to scale up the experiments to $p = 500$. In order to ensure a reasonable signal was detectable in each test, we fixed $n = 50$ for the tests. The following settings were used:

- $p \in \{100, 200, 500\}$;
- $s_0/p \in \{0.2, 0.5, 1.0, 2.0\}$;

- $n = 50$ fixed for all models;
- Algorithms: CCDr-MCP, CCDr-ℓ_1, MMHC, PC.

For all combinations of (p, s_0, n), we ran $N = 20$ tests each, resulting in 240 tests. These tests give us a better sense of the performance of the algorithms when the sample size is small relative to p.

The results are shown in Table 2. As before, the results are presented for each value of p, averaged over all tests and each value of s_0 (note that n did not change in these tests). In contrast to the low-dimensional scenario in which the constraint-based methods outperform our method, in high-dimensions we begin to see the advantages of CCDr in structure learning. As p increases and n remains fixed, the gap between CCDr-MCP and both PC and MMHC increases. In particular, across each value of p, the false discovery rates for all the methods are comparable, however, the increased sensitivity (true positive rate) and lower SHD indicates that CCDr-MCP provides a higher quality reconstruction of the true network. The numbers are illuminating: when $p = 500$, for graphs which have 460 edges on average, CCDr-MCP estimates approximately 100 more edges while maintaining roughly the same false discovery rate and including 50-70 *more* true edges on average.

By comparison, CCDr-ℓ_1 estimates fewer edges, obtaining lower sensitivity, and more closely mirrors the performance of PC and MMHC. This discrepancy in the performance of concave and ℓ_1 regularization in high dimensions highlights the advantages of concave regularization and supports the conclusions in the literature on sparse regression. This is not altogether surprising since our framework is closely tied to the Gaussian linear model and regression analysis.

Comparing Tables 1 and 2 when $p = 100, 200$, we also see that the CCDr methods are more robust to smaller sample sizes. When $p = 200$, for example, the net decrease in true positives between low- and high-dimensions is roughly 18 edges for CCDr-MCP, 26 edges for CCDr-ℓ_1, 46 edges for MMHC, and 42 edges for PC. Similar patterns are observed for $p = 100$, and for other metrics as well. This confirms what we already know about constraint-based methods: they are more reliable when sample sizes are large. Moreover, in spite of the fact that GES and HC were omitted from the high-dimensional experiments, we of course do not expect *improved* performance when n decreases. These observations confirm our expectations that regularization can improve the performance of structure learning algorithms in high-dimensions, with concave regularization providing a noticeable improvement upon ℓ_1 regularization.

6.3.3 Timing Comparison

A comparison of the total and average runtimes for all the algorithms is provided by Figures 3 and 4. The results are displayed graphically here; detailed tables can be found in the Supplementary Materials (Tables S1 and S2).

In low-dimensions, both GES and HC produce a single DAG estimate and take 15s and 25s, respectively, to estimate graphs with 200 nodes. This is compared with 3-5s for both CCDr-MCP and CCDr-ℓ_1, in which time both methods compute approximately 20 estimates. Amongst all the compared methods, the fastest alternative is the PC algorithm, however, the difference in timing is still roughly an order of magnitude: When $p = 200$, PC

$p = 100$, $T = 92.31$	CCDr-MCP	CCDr-ℓ_1	MMHC	PC
P	52.74	43.95	43.02	43.89
TP	**27.59**	21.48	23.82	24.12
R	16.95	16.29	**16.07**	16.19
FP	8.20	6.19	**3.12**	3.58
SHD (DAG)	72.92	77.03	**71.61**	71.76
SHD (skeleton)	55.98	60.74	**55.54**	55.58
TPR	**0.30**	0.23	0.26	0.26
FDR	0.48	0.51	**0.45**	**0.45**
$p = 200$, $T = 181.89$	CCDr-MCP	CCDr-ℓ_1	MMHC	PC
P	122.05	97.36	82.71	86.41
TP	**65.14**	47.40	44.71	46.70
R	35.75	34.89	**31.40**	33.17
FP	21.16	15.07	6.60	**6.54**
SHD (DAG)	**137.91**	149.56	143.78	141.72
SHD (skeleton)	**102.16**	114.67	112.38	108.55
TPR	**0.36**	0.26	0.25	0.26
FDR	0.47	0.51	**0.46**	**0.46**
$p = 500$, $T = 460.21$	CCDr-MCP	CCDr-ℓ_1	MMHC	PC
P	319.94	252.56	195.07	202.64
TP	**172.34**	121.75	101.49	104.33
R	88.51	89.33	**75.50**	82.60
FP	59.09	41.49	18.09	**15.71**
SHD (DAG)	**346.96**	379.95	376.81	371.60
SHD (skeleton)	**258.45**	290.62	301.31	289.00
TPR	**0.37**	0.26	0.22	0.23
FDR	**0.46**	0.52	0.48	0.49

Table 2: Average estimation performance of algorithms in high-dimensions.

takes a little less than 4s on average for a single estimate, whereas CCDr takes approximately *one-fifth of a second* per estimate. This translates to a total runtime of less than 4s for 20 CCDr estimates—faster than the time to compute a single model, on average, for the PC algorithm. Furthermore, CCDr-MCP is slightly faster than CCDr-ℓ_1, although the difference is small. Similar observations continue to hold in high-dimensions up to the tested limit of $p = 500$. Interestingly, both PC and MMHC are significantly faster in high-dimensions than in low-dimensions (see Tables S1 and S2 in the Supplementary Materials), which we suspect is due to how these algorithms scale with n: data sets with more samples require more time to process (see Section 6.6 for more details).

Combined with the improved performance in high-dimensions (Section 6.3.2), these results support our claim that CCDr is an improvement in both timing and accuracy over

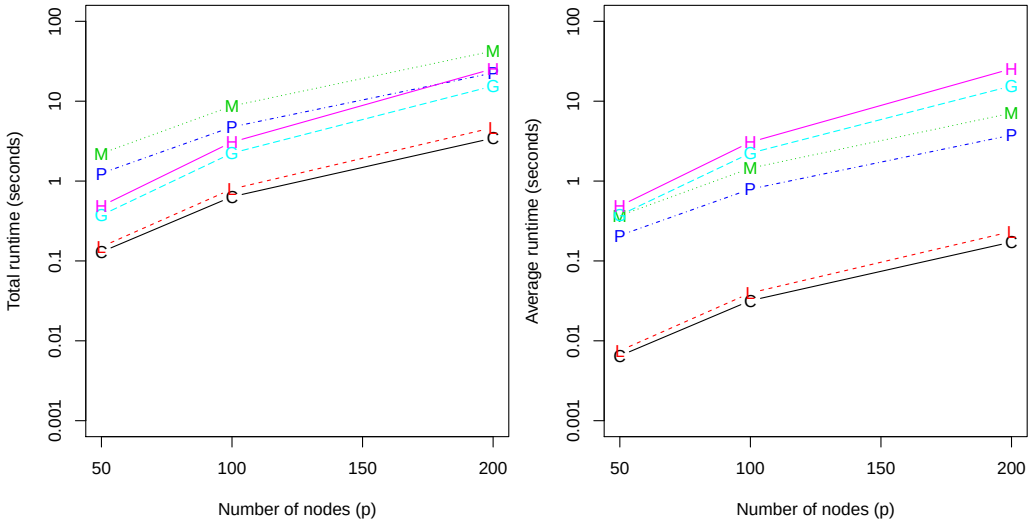


Figure 3: Timing comparison in low dimensions for all six algorithms (C = CCDr-MCP, L = CCDr-ℓ_1, G = GES, H = HC, M = MMHC, P = PC).

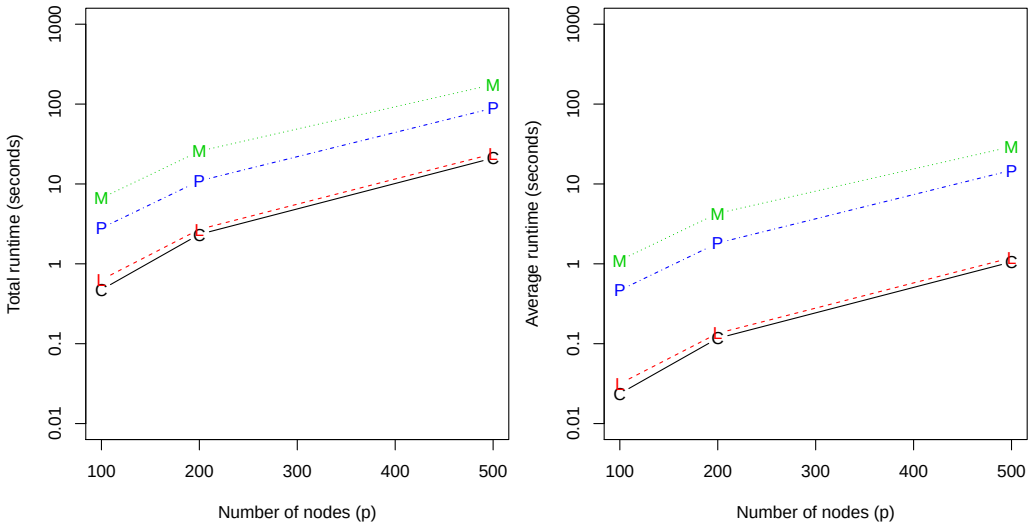


Figure 4: Timing comparison in high dimensions, excluding GES and HC (C = CCDr-MCP, L = CCDr-ℓ_1, M = MMHC, P = PC).

existing methods for high-dimensional data when $p \leq 500$. To see how CCDr performs when $p > 500$, we will show in the next subsection that the CCDr algorithm scales efficiently to high-dimensional problems with thousands of variables with almost no loss in reconstruction accuracy.

6.4 Large Graphs

The previous section offered a detailed assessment of the performance of the CCDr algorithm when $p \leq 500$. In order to test how our algorithm scales as the number of nodes increases, we ran further tests up to $p = 2000$ using CCDr-MCP. The purpose of these tests is to show how the proposed method scales as p increases in terms of timing and accuracy. Since the timing is acutely dependent on the relationship between the dimension, the sparsity of the true graph, and the number of samples, we opted to compare the timing over random choices of the latter two parameters. This also gives us a sense of how the algorithm performs when faced with a more realistic scenario in which the relationship between p, s_0, and n can be unpredictable. Specifically, we ran $N = 20$ tests with the following parameters:

- $p \in \{100, 200, 500, 1000, 1500, 2000\}$;
- $s_0/p \in \{0.2, 0.3, 0.4, \ldots, 2\}$;
- $n/p \in \{0.1, 0.2, 0.3, \ldots, 5\}$.

The parameters s_0 and n were chosen randomly from the above sets in each test, which resulted in an average sparsity level of $s_0/p = 1.06$. The results are displayed in Table 3 and Figure 5. Since the timing of the algorithm depends crucially on the total number of models estimated, and also on the threshold parameter α, we have plotted both the total and average runtimes for two scenarios: The time it took to estimate DAGs with up to p edges, and then the full running time with the edge threshold set at $\alpha = 3$. When $p = 1000$, the total running time is just under six minutes, with an average time per model of about 20 seconds. When $p = 2000$, the total running time is just under thirty minutes, with an average time per model of about 85 seconds.

In terms of accuracy, Table 3 shows that the results are comparable to those in Section 6.3. Furthermore, as p increases we notice that TPR increases while FDR decreases, which is likely due to the increased number of samples (on average) as p increases; when $p = 100$, there were $n = 114$ samples on average vs. $n = 2260$ when $p = 2000$. Combined with the timing data in Figure 5, this confirms that CCDr scales efficiently in terms of both n and p when the underlying graph is sparse.

After these experiments in this work were completed, the performance of our method was further improved, so that the total runtime for $p = 2000$ is now less than five minutes.[1] These changes were made to the underlying codebase, and *not* to the algorithm, thus the improvements were purely in terms of code efficiency. Using this updated implementation, we can report that our method has been successfully tested on graphs with up to 8000 nodes, with comparable accuracy to the results exhibited in Table 3. The total runtime for 20 estimates was 75 minutes, which may be compared with the 13 days reported for

1. A comprehensive comparison of the updated implementation vs. the numbers reported here can be found in Figure S1 in the Supplementary Materials.

Number of nodes (p)	100	200	500	1000	1500	2000
Number of samples (n)	114	190	520	1280	1470	2260
T	83.15	237.15	538.15	1186.35	1550.15	2057.95
P	66.15	191.90	488.30	1082.20	1434.20	1926.90
TP	36.15	111.50	279.80	636.70	854.25	1156.10
R	20.75	46.45	115.80	226.45	323.75	447.90
FP	9.25	33.95	92.70	219.05	256.20	322.90
SHD (DAG)	56.25	159.60	351.05	768.70	952.10	1224.75
SHD (skeleton)	35.50	113.15	235.25	542.25	628.35	776.85
TPR	0.43	0.47	0.52	0.54	0.55	0.56
FDR	0.45	0.42	0.43	0.41	0.40	0.40

Table 3: Average estimation performance of CCDr-MCP from Section 6.4, averaged over $N = 20$ random choices of s_0 and n for each p.

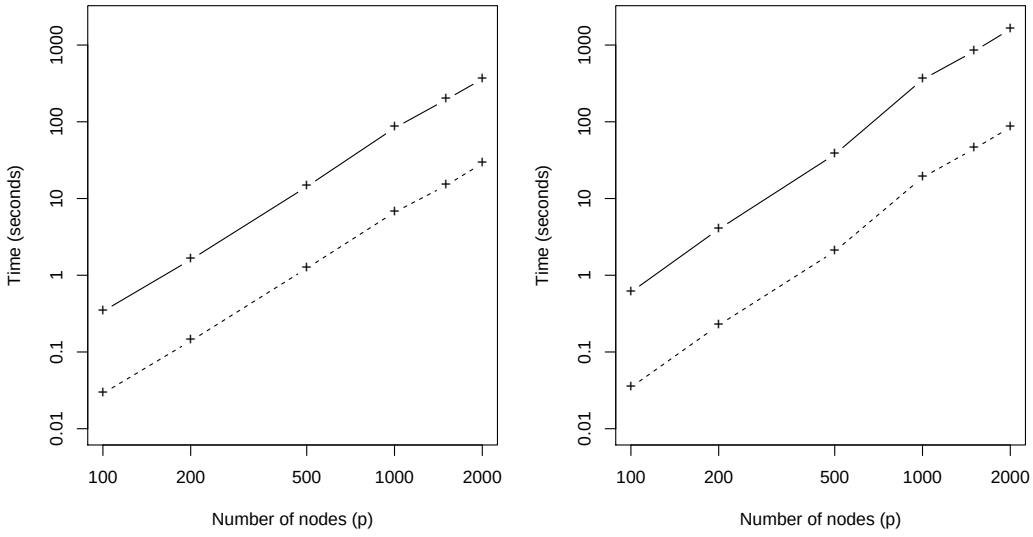


Figure 5: Timing data for CCDr-MCP up to $p = 2000$. The solid line is the total runtime and the dashed line is the average runtime. (left) Time to estimate graphs with at most p edges; (right) Full runtime with edge threshold $\alpha = 3$.

MMHC on a graph with $p = 5000$ in Tsamardinos et al. (2006). Regarding the internal implementation of our method, we did not make use of an internal cache, memoization, or efficient data structures (i.e. besides standard vectors), all of which are common strategies used in existing methods. It stands to reason that an optimized implementation would yield even faster results. For instance, we perform the acyclicity check statically with each edge addition; one could imagine a more sophisticated strategy such as incremental topological sorting would lead to significant performance enhancements.

6.5 Model Selection

Thus far, we have used the "best estimate" according to distance from the true graph, measured by SHD, in order to select models from the estimated solution paths for CCDr, MMHC, and PC. This choice provides a consistent comparison, but results in relatively sparse estimates since missing edges are penalized equally against false positives. One of the advantages of CCDr is that it is able to estimate models with higher sensitivity much more efficiently than PC or MMHC. Alternatively, one could use empirical model selection techniques such as BIC or cross-validation. It has already been noted that these empirical model selection techniques are suboptimal in high-dimensions, particularly for graphical models. This has been previously reported in the literature, see for instance Fu and Zhou (2013). Here we briefly discuss the results of some tests to confirm this behaviour for our method.

Using both conventional BIC and the extended BIC for high-dimensional problems developed in Foygel and Drton (2010), we selected the tuning parameters for CCDr-MCP, CCDr-ℓ_1, PC, and MMHC. The results confirm that BIC tends to select models with too many edges by insufficiently penalizing the model complexity, consistent with Figure 2. One may ask if all the algorithms suffer equally, and the answer is no. For the reasons already discussed, we were not able to test the performance of either PC or MMHC for $\alpha > 0.05$, which is the regime in which more edges tend to be selected. Thus, in using BIC to select the significance level, the maximum value of $\alpha = 0.05$ was over-represented. We suspect that if we had run PC and MMHC with $\alpha > 0.05$ in order to produce estimates with extraneous edges, BIC would also select these models. As a result of these limitations, in selecting models based on BIC, CCDr appeared to perform worse relative to either PC or MMHC than reported in previous sections.

To correct for this, we ran the same model selection test using BIC as the selection criterion, but this time restricting the set of CCDr candidates to those with at most as many edges as the most produced by either the PC algorithm or the MMHC algorithm. Using the same data as in Section 6.3.1, the results resemble those previously reported (Table S3 in the Supplementary Materials). Across the board, graphs with more edges were selected, but the qualitative observations between CCDr and PC / MMHC remain the same.

6.6 Further Discussion

The experiments and results described already, while providing a general overview of the performance of the algorithms tested, also raise several questions which we address briefly in this section.

While we tested a variety of sparsity levels in Section 6.3, we have not provided a detailed assessment of how the performance of the algorithms varies as the sparsity increases or decreases. An analysis of the effect of sparsity shows that the same qualitative behaviour observed in Sections 6.3.1 and 6.3.2 persists (see Figures S3 and S4 in the Supplementary Materials). We do observe a small decrease in reconstruction accuracy for the CCDr methods when the graph is more dense ($s_0/p = 2$); improving our method when the true graph is more dense remains for future work.

For the CCDr algorithm, in order to provide a reasonable balance of complexity and efficiency in the resulting estimation problem, we fixed $\gamma = 2$. Nonetheless, this parameter was observed to have a non-negligible effect on the results and a more in-depth study in the future would account for the effect of this parameter. Another parameter which we have not discussed is the maximum neighbourhood size in the true graph, which we controlled in our simulations by controlling the expected neighbourhood size. Keeping the neighbourhoods small is critical for keeping the running time of the PC algorithm reasonable. Further simulations in which we allowed each node to have arbitrarily many parents showed that the running time of the CCDr algorithm does not depend on this parameter. Moreover, restricting the maximum size of the conditioning sets used in the conditional independence tests in the PC algorithm, as suggested by the work of Anandkumar et al. (2012), also had a negligible effect. Finally, both PC and MMHC show relatively poor computational complexity with respect to the sample size n, with more instances requiring more time to process. Our tests indicate that the complexity of CCDr is essentially independent of n—the only dependence on sample size enters through the computation of the correlation matrix in the first step.

7. Real Networks

While the random set-up in the previous section provided a convenient setting to test many random structures quickly and efficiently, random graphs may not be good representatives of realistic network structures. For this reason, we augmented these experiments with tests on real network structures, using both simulated and scientific (unsimulated) data. Our first experiment uses network structures from the Bayesian Network Repository,[2] a standardized collection of networks which is commonly used as a benchmark for structure learning methods, as well as a simulated scale-free network. In order to assess the impact of these methods on actual scientific data, we also compare the performance of the algorithms on the well-known flow cytometry data set (Sachs et al., 2005).

7.1 Bayesian Network Repository

All of the networks examined in this experiment were loaded using the `bnlearn` package.[3] We then used the graph structures to generate data according to a structural equation model as in the previous section. Furthermore, in order to keep the focus on high-dimensional estimation, we fixed the number of samples at $n = 50$, which narrowed the choice of networks to those that satisfy $p > 50$. Seven such network structures were tested, to which we added

2. The original repository can be found at: `http://www.cs.huji.ac.il/site/labs/compbio/Repository/`.
3. A mirror of the repository used by the `bnlearn` package can be found at: `http://www.bnlearn.com/bnrepository/`.

one randomly generated scale-free structure with 200 nodes. The scale-free network was created using the `igraph` package. For each network, we generated random coefficients in the interval $[0.5, 1]$ for each edge and generated a single random data set with unit variances for testing. This procedure was replicated $N = 50$ times, and the number of true positives and false positives were tracked for each algorithm. We also increased the length of the regularization path used for the CCDr methods to 50 estimates while keeping both PC and MMHC fixed at six estimates for each graph. Based on the results in the previous section—particularly with respect to timing—both HC and GES were excluded from these tests.

We have already observed in Section 6.5 how traditional model selection techniques such as BIC and cross-validation perform poorly. For this reason, we chose to present the results graphically by their ROC curves in order to compare the true positive rate against the false positive rate as a function of the tuning parameters. The resulting ROC curves are displayed in Figure 6.

In terms of reconstruction accuracy, with only one exception, we see that the CCDr methods perform as well or better than the other methods in these experiments. Consistent with the previously reported experiments on random graphs, the CCDr methods tend to show higher sensitivity with comparable false positive rates in high dimensions. In some cases the improvements are dramatic—for instance, `pathfinder`, `scalefree`, and `pigs`. The one exception is the `win95pts` network, in which the PC algorithm attains slightly higher sensitivity and lower FDR compared with the CCDr methods as well as MMHC. These results further highlight the tradeoffs in learning between each approach and confirm the patterns observed previously in the literature: constraint-based methods tend to miss edges in the true skeleton, resulting in lower false discovery rates and lower sensitivity, whereas regularization tends to increase overall sensitivity with the risk of higher false positive rates if the amount of regularization is not calibrated properly.

More interesting is the comparison between CCDr-MCP and CCDr-ℓ_1. Compared with the simulation results in Section 6, there is a more pronounced difference between the performance of concave vs ℓ_1 regularization, with the former outperforming the latter. This is most visible in the `hailfinder` and `pigs` networks, where both methods show comparable sensitivity but CCDr-MCP exhibits lower false positive rates. The only network in which ℓ_1 regularization is preferable is `pathfinder`, where CCDr-ℓ_1 obtains higher sensitivity later in the solution path.

Consistent with the previous experiments, however, the main advantages of CCDr come in the form of efficiency: Figure S2 in the Supplementary Materials contains a comparison of runtime for each network and method. Unlike in the previous experiments, for these experiments the estimated solution path for the CCDr methods was 2.5 times longer, with up to 50 estimates per solution path. Notwithstanding, the CCDr methods were consistently the fastest. For example, using PC and MMHC, the `pathfinder` network with $p = 135$ nodes took 110x and 150x longer on average per estimate to compute, respectively. At the other end of the spectrum, the hardest graph to reconstruct was the `pigs` network, which took 39s for CCDr-MCP, 29s for CCDr-ℓ_1, 71s for PC, and 147s for MMHC. In both cases CCDr-MCP easily did the best job reconstructing the true networks.

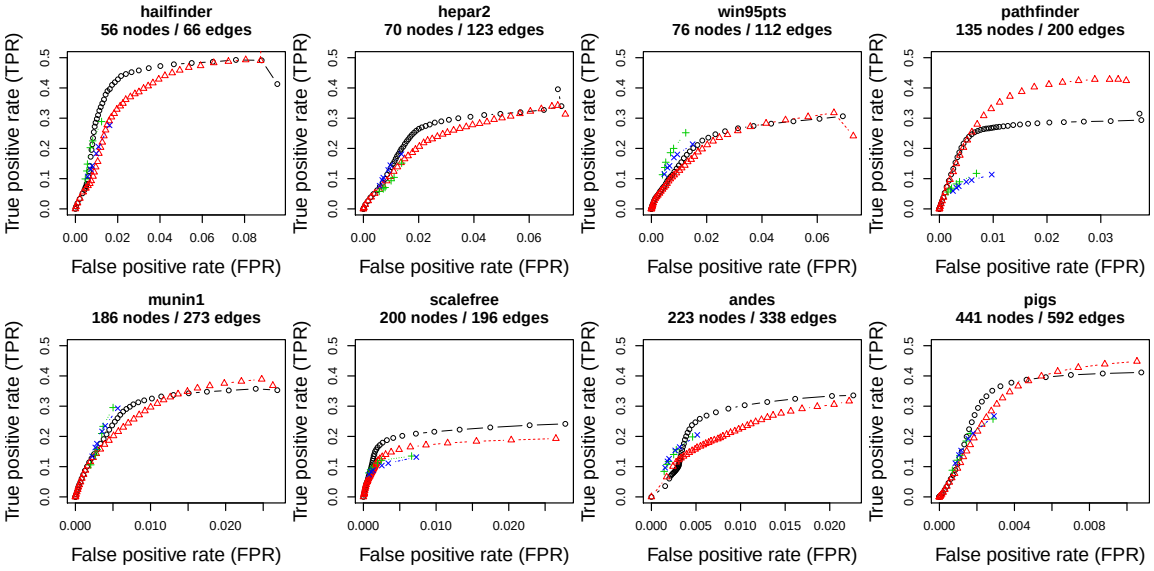


Figure 6: ROC curves for real networks (black ◯ = CCDr-MCP, red △ = CCDr-ℓ_1, blue × = MMHC, green + = PC).

7.2 Application to Real Data

We analyzed the well-known flow cytometry data set, generated by Sachs et al. (2005), which has been previously analyzed by Fu and Zhou (2013); Shojaie and Michailidis (2010); Friedman et al. (2008) among others. The data set contains $n = 7466$ measurements of $p = 11$ continuous variables corresponding to proteins and phospholipids in human immune system cells. The underlying network, constructed through a careful series of biological experiments, has $s_0 = 20$ edges, and represents a gold-standard for comparison currently accepted by the biology community. Hereafter, we regard this consensus network as the true network in order to assess the algorithms. While this data set is hardly high-dimensional, it represents one of the few continuous data sets for which we have oracular knowledge of the true underlying DAG *as well as* real data from which to infer the true structure.

The original data set contains a mixture of both observational and experimental data. Since the methods presented here assume the data are normally distributed, we first tested the original continuous variables for normality, and much as expected the data were highly non-normal. To correct for this, we applied a logarithm transform, which produced variables that were much closer to Gaussian. This data set was used for our tests on continuous data.

We also analyzed a discretized version of the data set containing $n = 5400$ measurements, created by transforming the continuous data into three nonnegative levels which correspond to *high*, *medium*, and *low*, so that magnitudes were partially preserved (Sachs et al., 2005). This data set is especially interesting for a number of reasons. First, it represents a test of model misspecification: Our method was developed for continuous data, but nothing prevents us from naively feeding this data set into the algorithm. By treating the three

levels as numeric values (*high* = 2, *medium* = 1, *low* = 0), we can compute the correlation matrix and proceed with the second and third steps in Algorithm 2. Since the data are clearly not Gaussian, the results of this test give us a sense of how well our method performs on discrete, non-Gaussian data. Second, as a result of postprocessing to clean up the data as well as the discretization itself, it is much less noisy than the original data set, which provides an interesting side-by-side comparison.

A few changes were made to the set-up used in previous experiments. First, since the number of variables was small, it was feasible to run the constraint-based methods on a longer sequence of significance levels. Thus, we used a sequence of 10 levels:

$$\alpha \in \{10^{-6}, 5 \times 10^{-6}, 10^{-5}, 5 \times 10^{-5}, 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 5 \times 10^{-3}, 0.01, 0.05\}.$$

Furthermore, in a majority of the tests we ran, the PC algorithm was unable to orient all the edges in the final step, leading to a partially directed graph (formally a CPDAG, see Remark 9). As a result, we had to modify our metrics to allow for undirected edges. We did this favourably for the PC algorithm by counting an undirected edge as a true edge as long as the same edge exists in the skeleton of the true graph. Any edge that was successfully oriented by the PC algorithm was treated as a directed edge. Finally, we split each data set in half in order to obtain a testing data set on which to compute the log-likelihood of the estimated models. Since the PC algorithm was not able to estimate DAGs, log-likelihood scores could not be computed for the continuous data set.

Tables 4 and 5 summarize the results for a sample run, which are indicative of the general behaviour when different random splits are tested. Instead of selecting the best estimates as in Section 6, we chose estimates with comparable numbers of edges, selected to match the true graph as closely as possible with $s_0 = 20$. The results for CCDr-MCP are visualized in Figure 7. Both GES and HC consistently estimated too many edges, which matches the behaviour observed in Section 6.3.1. For the continuous data set, CCDr-MCP and MMHC perform the best with almost identical metrics, while for the discrete data set CCDr-MCP is clearly optimal with fewer false positives and smaller SHD across the board. This indicates that even though this method was developed with continuous Gaussian data in mind, it can still be applied to discrete data with reasonable results.

Due to the small size of the graph with only $p = 11$ nodes, the differences in timing are largely negligible, taking fractions of a second to complete. Because of this, the processor time is subject to fluctuations in low-level bottlenecks most likely unrelated to the core algorithms themselves, and so we do not report exact times here. At a high level we did observe that HC and GES show much improved performance relative to PC and MMHC, however, the CCDr methods are still consistently the fastest.

8. Conclusion

We have introduced a general penalized likelihood framework for estimating sparse Bayesian networks, along with a fast algorithm that is easily implemented on a personal computer. In the finite dimensional scenario, the resulting estimator has good theoretical properties. Through a series of tests designed to test the limits of this new algorithm, we have shown that our approach accurately estimates networks with 2000 nodes while scaling efficiently to handle networks with up to 8000 nodes. The proposed method is compatible with high-

$p = 11$, $T = 20$	CCDr-MCP	CCDr-ℓ_1	GES	HC	MMHC	PC
P	20	20	41	38	20	20
TP	7	7	9	**10**	7	7
R	2	**1**	7	6	2	2
FP	**11**	12	25	22	**11**	**11**
SHD (DAG)	**24**	25	36	32	**24**	25
SHD (skeleton)	**22**	24	29	26	**22**	**22**
Test Log-likelihood	-2.05	-2.19	**-0.34**	-1.09	-2.03	—

Table 4: Structure estimation performance for all algorithms using the log-transformed continuous cytometry data.

$p = 11$, $T = 20$	CCDr-MCP	CCDr-ℓ_1	GES	HC	MMHC	PC
P	20	20	43	35	20	20
TP	6	3	**13**	7	3	6
R	5	6	4	7	5	**2**
FP	**9**	11	26	21	12	12
SHD (DAG)	**23**	28	33	34	29	26
SHD (skeleton)	**18**	22	29	27	24	24
Test Log-likelihood	-0.68	-1.86	**-0.10**	0.18	-2.32	-2.01

Table 5: Structure estimation performance for all algorithms using discretized cytometry data.

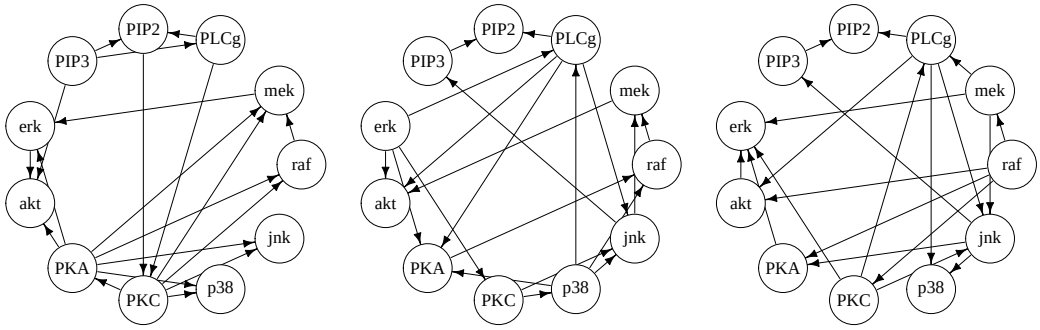


Figure 7: Comparison of the consensus network (left) against the DAGs estimated by the CCDr-MCP algorithm for both data sets: (middle) Log-transformed continuous data set; (right) Discretized data set.

dimensional data where $p \gg n$, and outperforms many existing methods in both speed and accuracy in this regime. Tests on real networks have validated the performance and applicability of this method in a variety of domains.

Our focus in this work has been on structure recovery, which is closely related to statistical inference and should not be confused with the complementary problem of *prediction*. For this reason, the metrics we employed require knowledge of the true underlying graph. Alternatively, one could inquire into the predictive performance and generalizability of learning methods, in which case metrics such as the prediction loss and test-data likelihood can be assessed without prior knowledge of the true graph. Indeed, our simulations indicate that existing score-based methods such as GES may perform better with respect to such predictive metrics. We have already discussed in Section 6.3.1 why this may be, and it remains for future work to study this phenomenon in more depth.

While we have focused on the use of cyclic coordinate descent to minimize the penalized log-likelihood, it would be interesting to compare more sophisticated optimization techniques such as adaptive and stochastic coordinate descent. It also remains to incorporate prior knowledge either via whitelists and blacklists, or through a more sophisticated hybrid Bayesian approach. As nonconvex optimization is a rapidly developing field of study, the methods presented here merely scratch the surface of how such techniques can be applied to the structure learning problem for Bayesian networks. An `R` package which implements the proposed algorithm along with some of these improvements is currently under development.

The central theme of exploiting convexity to solve nonconvex problems is an intriguing prospect for the development of new algorithms in statistics and machine learning. Indeed, the main difficulties with nonconvex regularization are computational in nature. Although recent progress has broken this barrier in the case of least squares regression, to our knowledge the algorithm presented here is one of the first to approximate this type of nonconvex optimization problem when p is in the thousands. Moreover, since our method revolves around a continuous optimization problem, we avoid approaches that rely on individual edge additions and removals, which are intrinsically discrete. As a result, future advances in nonconvex optimization will directly affect how we solve the maximum likelihood problem presented here.

Acknowledgements

We would like to thank the referees for their helpful comments. We would also like to thank Marco Scutari and Sara van de Geer for their thoughtful discussions, as well as Damon Alexander for his assistance and suggestions in implementing the algorithm. This work was supported by NSF grants DMS-1055286 and DMS-1308376 (to Q.Z.) and NSF graduate research fellowships DGE-1144087 and DGE-0707424. B.A. was also supported by a UCLA Dissertation Year Fellowship.

Appendix A. Proofs of Main Results

We collect here the proofs of our main results.

A.1 Formal Preliminaries

Conceptually our theory is quite simple: we have a function F on $\mathbb{R}^{p^2}$ which we would like to maximize over a subset defined by the space of DAGs, $\mathcal{D}$. In order to properly specify a topology for this space, and to ensure that the translation between our statistical model for

(B, Ω) and the mathematical model for $\boldsymbol{\nu}$ is coherent, we carefully outline the mathematical set-up here.

Given a DAG (B, Ω), consider the reparameterization (Φ, R) given by

$$\Phi = B\Omega^{-1/2} \tag{36}$$

$$R = \Omega^{-1/2}. \tag{37}$$

This is of course just the matrix version of the reparameterization that leads to (11). Now define the following function which maps $(\Phi, R) \in \mathbb{R}^{p\times p} \times \mathbb{R}^{p\times p}$ into $\mathbb{R}^{p^2}$:

$$\boldsymbol{\nu}(\Phi, R) = \text{vec}(U) = (u_1, \dots, u_p), \quad \text{where } U = [u_1 \,|\, \dots \,|\, u_p] = R + \Phi.$$

Recall that Φ has zeroes on the diagonal and R is a diagonal matrix, so that the sum $U := R + \Phi$ has the same number of nonzero entries as R and Φ separately. Furthermore, the sparsity pattern of the off-diagonal elements of U exactly matches that of Φ.

In the proofs, when there is no confusion we will simply write $\boldsymbol{\nu} = U = (\Phi, R) = (B, \Omega)$ to mean that these are all equivalent representations of the same DAG in various parameterizations. In particular, for any $\boldsymbol{\nu}_0 \in \mathcal{E}_0$, we have $\boldsymbol{\nu}_0 = U_0 = (\Phi_0, R_0) = (B_0, \Omega_0)$. Mathematically, we will work with $\boldsymbol{\nu}$, however, our results should always be interpreted in terms of the original model (B, Ω).

The space of DAGs is formally defined as follows:

$$\mathcal{D} := \left\{ \boldsymbol{\nu} = \boldsymbol{\nu}(\Phi, R) \in \mathbb{R}^{p^2} : \Phi \in \mathbb{R}^{p\times p} \text{ is a DAG, } \rho_j > 0 \text{ for all } j \right\}.$$

This space inherits its topology from the ambient space $\mathbb{R}^{p^2}$, and it is this space on which we wish to maximize the function $F(\boldsymbol{\nu}) = \ell_n(\boldsymbol{\nu}) - np_{\lambda_n}(\boldsymbol{\nu})$.

A.2 Proof of Theorem 2

We begin by formalizing some of the background material on the Cholesky decomposition used in Section 2.3, which will also be used in the proof of Lemma 4. First recall the following standard result:

Lemma 8. *For any symmetric positive definite matrix $A \in \mathbb{R}^{p\times p}$ and permutation $\pi \in \mathcal{P}$, the Cholesky decomposition $A = LDL^T$ satisfies*

$$P_\pi A = (P_\pi L)(P_\pi D)(P_\pi L)^T,$$

where L is lower triangular and D is a diagonal matrix.

Now suppose Θ is given and use the Cholesky decomposition to write $\Theta = \Theta(L, D)$ as in (9). Then, taking $A = \Theta(L, D)$ in Lemma 8, we obtain $P_\pi \Theta(L, D) = \Theta(P_\pi L, P_\pi D)$. Alternatively, suppose $(B, \Omega) \in \mathcal{E}(\Theta)$ and suppose $\pi \in \mathcal{P}$ is compatible with (B, Ω). Since $P_\pi B$ is lower-triangular, by taking $A = \Theta(P_\pi B, P_\pi \Omega)$, we may similarly deduce

$$P_{\pi^{-1}} \Theta(P_\pi B, P_\pi \Omega) = \Theta(B, \Omega) \implies \Theta(P_\pi B, P_\pi \Omega) = P_\pi \Theta(B, \Omega).$$

This proves the following lemma, which will be useful:

Lemma 9. *Let (B, Ω) be a DAG. For any permutation $\pi \in \mathcal{P}$ that is compatible with (B, Ω), we have*

$$P_\pi \Theta(B, \Omega) = \Theta(P_\pi B, P_\pi \Omega).$$

We now prove Lemma 4, which will be used in the proof of Theorem 2.

Proof of Lemma 4 We only prove this for the original parameterization (B, Ω); the reparameterized case is similar.

Since B_1 and B_2 have a common topological sort, there is a permutation π of the vertices that orders B_1 and B_2 simultaneously, so that $P_\pi B_1$ and $P_\pi B_2$ are both strictly lower triangular. Suppose then that $\Theta(B_1, \Omega_1) = \Theta(B_2, \Omega_2) := \tilde{\Theta}$, so that (using Lemma 9 above)

$$\begin{aligned} & P_\pi \Theta(B_1, \Omega_1) = P_\pi \Theta(B_2, \Omega_2) \\ \iff & \Theta(P_\pi B_1, P_\pi \Omega_1) = \Theta(P_\pi B_2, P_\pi \Omega_2) \\ \iff & (I - P_\pi B_1)(P_\pi \Omega_1)^{-1}(I - P_\pi B_1)^T = (I - P_\pi B_2)(P_\pi \Omega_2)^{-1}(I - P_\pi B_2)^T. \end{aligned}$$

The last expression is equal to $P_\pi \tilde{\Theta}$, which is a symmetric positive definite matrix. By the uniqueness of the Cholesky factorization, we must have

$$\begin{aligned} I - P_\pi B_1 &= I - P_\pi B_2 \\ (P_\pi \Omega_1)^{-1} &= (P_\pi \Omega_2)^{-1}, \end{aligned}$$

which implies

$$B_1 = B_2, \quad \Omega_1 = \Omega_2.$$

Since B_1 was assumed to be distinct from B_2, this contradiction establishes the desired result. ■

Proof of Theorem 2 Suppose $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ with $b_n(\boldsymbol{\nu}_0) \to 0$. It suffices to check Conditions (A)-(C) from Fan and Li (2001), which are simply the standard regularity conditions for asymptotic efficiency of ordinary maximum likelihood estimates. Model identifiability is not an issue since the same analysis can be carried out for any equivalent parameter (see Section 4.1). Since the densities $f(\cdot \,|\, \boldsymbol{\nu})$ are Gaussian, the only condition that needs to be checked is that the Fisher information is positive definite at $\boldsymbol{\nu}_0$ restricted to the DAG space $\mathcal{D}$. Theorem 2 will then follow immediately from Theorem 1 in Fan and Li (2001).

Let $I(\boldsymbol{\nu}_0)$ denote the usual Fisher information matrix at this point; we will show that $I(\boldsymbol{\nu}_0)$ is positive definite. Since f is always a Gaussian density, it will suffice to show that $f(\cdot \,|\, \boldsymbol{\nu}) \neq f(\cdot \,|\, \boldsymbol{\nu}_0)$ for $\boldsymbol{\nu}$ in a sufficiently small neighbourhood of $\boldsymbol{\nu}_0$.

Now suppose $\boldsymbol{\nu} = (\Phi, R)$ is in an arbitrarily small neighbourhood of $\boldsymbol{\nu}_0 = (\Phi_0, R_0)$. Then it must hold that $\phi_{ij} \neq 0$ whenever $\phi^0_{ij} \neq 0$. Indeed, otherwise

$$\|\Phi - \Phi_0\|^2 \geq (\phi_{ij} - \phi^0_{ij})^2 = |\phi^0_{ij}|^2.$$

Thus, $\phi^0_{ij} \neq 0$ implies $\phi_{ij} \neq 0$, or $i \to j$ in Φ_0 implies $i \to j$ in any DAG close to Φ_0. In particular, Φ contains all the edges (including orientation) in Φ_0, with the possible addition of extra edges. That is, Φ_0 is a subgraph of Φ. It follows that there is an ordering of the vertices that is compatible with Φ and Φ_0 simultaneously. Since $\Phi \neq \Phi_0$, it follows from Lemma 4 that $\Theta(\boldsymbol{\nu}) \neq \Theta(\boldsymbol{\nu}_0)$, whence $f(\cdot \,|\, \boldsymbol{\nu}) \neq f(\cdot \,|\, \boldsymbol{\nu}_0)$. ■

Proof of Lemma 5 Note that Lemma 1 implies that the equivalence class $\mathcal{E}_0$ is finite. Set $\varepsilon = \min_{\boldsymbol{\nu}_0 \in \mathcal{E}_0} \min_{i,j}\{|\phi^0_{ij}| : \phi^0_{ij} \neq 0\} > 0$. Then if $\|\Phi - \Phi_0\| \leq \|\boldsymbol{\nu} - \boldsymbol{\nu}_0\| < \varepsilon$, the arguments in the proof of Theorem 2 guarantee the existence of an ordering that is compatible with Φ and Φ_0, and the result follows from Lemma 4. ■

A.3 Proof of Theorem 6

Instead of directly proving Theorem 6, we will prove a slightly more general statement under weaker assumptions. Theorem 6 will then follow as a special case.

The following technical lemmas ensure that the objective function $F(\boldsymbol{\nu})$ is well-behaved with respect to taking limits. The first is a standard application of the uniform law of large numbers (see, for example, Ferguson, 1996, §16) and the second is a direct consequence of concavity.

Lemma 10. *Fix $\boldsymbol{\nu}_0$ and suppose $\boldsymbol{\nu}_n$ is a sequence with $\|\boldsymbol{\nu}_n - \boldsymbol{\nu}_0\| = o(1)$. If the empirical log-likelihood $\ell_n(\boldsymbol{\nu})$ is continuous for all n, then*

$$P\left(\lim_{n\to\infty} \frac{1}{n}\ell_n(\boldsymbol{\nu}_n) = \lim_{n\to\infty} \frac{1}{n}\ell_n(\boldsymbol{\nu}_0)\right) = 1.$$

Lemma 11. *Suppose that $p_\lambda(t)$ is nondecreasing and concave for $t \geq 0$ with $p_\lambda(0) = 0$. If $\limsup_n \tau(\lambda_n) < \infty$, then for any $x_0 > 0$ there exists a constant C, depending only on x_0, such that*

$$|p_{\lambda_n}(x) - p_{\lambda_n}(x_0)| \leq C|x - x_0| \quad \textit{for all } x \geq 0 \textit{ and all } n.$$

Recall that $f(n) = \omega(g(n)) \iff g(n) = o(f(n))$, that is, for every $C > 0$,

$$f(n) \geq Cg(n) \quad \text{for all large } n.$$

As in Section 4, we use $\widehat{\boldsymbol{\nu}}_n$ and $\widehat{\boldsymbol{\nu}}^*_n$ to denote the local maximizers close to $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}^*$, respectively, whose existence is guaranteed by Theorem 2.

Theorem 12. *Suppose that $p_\lambda(t)$ is nondecreasing and concave for $t \geq 0$ with $p_\lambda(0) = 0$. Let $\boldsymbol{\nu}_0 \in \mathcal{E}_0$ be a DAG with strictly more edges than $\boldsymbol{\nu}^*$. Assume further that the conditions for Theorem 3 hold for both $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}^*$. If*

1. *$c_n(\boldsymbol{\nu}^*) = \tau(\lambda_n) + O(n^{-1/2})$ and $c_n(\boldsymbol{\nu}_0) = \tau(\lambda_n) + O(n^{-1/2})$,*
2. *$\limsup_n \tau(\lambda_n) < \infty$,*

3. $\tau(\lambda_n) = \omega(n^{-1/2})$,

then for every $\varepsilon > 0$,

$$P\left(\ell_n(\widehat{\boldsymbol{\nu}}_n^*) - n\, p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n^*) > \ell_n(\widehat{\boldsymbol{\nu}}_n) - n\, p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n)\right) \geq 1 - \varepsilon \quad \textit{for sufficiently large } n.$$

Proof Since we assume Theorem 3 holds for both $\boldsymbol{\nu}_0$ and $\boldsymbol{\nu}^*$, we may assume without loss of generality that $\mathrm{supp}(\widehat{\boldsymbol{\nu}}_n^*) = \mathrm{supp}(\boldsymbol{\nu}^*)$ and $\mathrm{supp}(\widehat{\boldsymbol{\nu}}_n) = \mathrm{supp}(\boldsymbol{\nu}_0)$.

Since ℓ_n is continuous for each n, $\|\widehat{\boldsymbol{\nu}}_n - \boldsymbol{\nu}_0\| = O_P(n^{-1/2})$, and $\|\widehat{\boldsymbol{\nu}}_n^* - \boldsymbol{\nu}^*\| = O_P(n^{-1/2})$, Lemma 10 implies that

$$\frac{1}{n}(\ell_n(\widehat{\boldsymbol{\nu}}_n) - \ell_n(\widehat{\boldsymbol{\nu}}_n^*)) \to 0$$

almost surely. It is easy to show that in fact $n^{-1}(\ell_n(\widehat{\boldsymbol{\nu}}_n) - \ell_n(\widehat{\boldsymbol{\nu}}_n^*)) = O_P(n^{-1/2})$.

It will suffice to show that for any $\varepsilon > 0$, there exists an N such that for all $n > N$, we have

$$P\left(p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n) - p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n^*) - \frac{1}{n}(\ell_n(\widehat{\boldsymbol{\nu}}_n) - \ell_n(\widehat{\boldsymbol{\nu}}_n^*)) > 0\right) \geq 1 - \varepsilon.$$

Given $\varepsilon > 0$, there exists $M > 0$ such that

$$P\left(\frac{1}{n}(\ell_n(\widehat{\boldsymbol{\nu}}_n) - \ell_n(\widehat{\boldsymbol{\nu}}_n^*)) \leq M n^{-1/2}\right) \geq 1 - \varepsilon,$$

so that it suffices to check that $p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n) - p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n^*) > M n^{-1/2}$ for sufficiently large n.

Lemma 11 implies that for each $\phi_{ij}^0 \neq 0$,

$$|p_{\lambda_n}(\widehat{\phi}_{ij}^0) - p_{\lambda_n}(\phi_{ij}^0)| \leq C|\widehat{\phi}_{ij}^0 - \phi_{ij}^0| = O(n^{-1/2}),$$

and similarly for all $\phi_{ij}^* \neq 0$. Thus we can write $p_{\lambda_n}(\widehat{\boldsymbol{\nu}}_n) = p_{\lambda_n}(\boldsymbol{\nu}_0) + O_P(n^{-1/2})$ and similarly for $\widehat{\boldsymbol{\nu}}^*$. It thus suffices to show that

$$p_{\lambda_n}(\boldsymbol{\nu}_0) - p_{\lambda_n}(\boldsymbol{\nu}^*) = \omega(n^{-1/2}).$$

Now, using Condition 1,

$$\begin{aligned}
p_{\lambda_n}(\boldsymbol{\nu}_0) - p_{\lambda_n}(\boldsymbol{\nu}^*) &= \sum_{\phi_{ij}^0 \neq 0} p_{\lambda_n}(|\phi_{ij}^0|) - \sum_{\phi_{ij}^* \neq 0} p_{\lambda_n}(|\phi_{ij}^*|) \\
&\geq s_0 c_n(\boldsymbol{\nu}_0) - s^* \tau(\lambda_n) + s^* \tau(\lambda_n) - \sum_{\phi_{ij}^* \neq 0} p_{\lambda_n}(|\phi_{ij}^*|) \\
&= (s_0 - s^*)\tau(\lambda_n) + O(n^{-1/2}) + \sum_{\phi_{ij}^* \neq 0} (\tau(\lambda_n) - p_{\lambda_n}(\phi_{ij}^*)) \\
&\geq (s_0 - s^*)\tau(\lambda_n) + O(n^{-1/2}).
\end{aligned}$$

Since $\tau(\lambda_n) = \omega(n^{-1/2})$ (Condition 3), it follows that $p_{\lambda_n}(\boldsymbol{\nu}_0) - p_{\lambda_n}(\boldsymbol{\nu}^*) \geq \omega(n^{-1/2})$, from which the claim follows. ■

Proof of Theorem 6 Condition 3 in Theorem 12 is equivalent to $\tau(\lambda_n)/n^{-1/2} \to \infty$, and Theorem 6 follows as a special case since the equivalence class $\mathcal{E}_0$ is finite. ■

References

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification Part I: Algorithms and empirical evaluation. *The Journal of Machine Learning Research*, 11:171–234, 2010a.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification Part II: Analysis and extensions. *The Journal of Machine Learning Research*, 11:235–284, 2010b.

Animashree Anandkumar, Vincent YF Tan, Furong Huang, and Alan S Willsky. High-dimensional Gaussian graphical model selection: Walk summability and local separation criterion. *The Journal of Machine Learning Research*, 13(1):2293–2337, 2012.

Animashree Anandkumar, Daniel Hsu, Adel Javanmard, and Sham Kakade. Learning linear Bayesian networks with latent variables. In *Proceedings of The 30th International Conference on Machine Learning*, pages 249–257, 2013.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2009.

Peter Bühlmann and Sara van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer, 2011.

Peter Bühlmann, Jonas Peters, Jan Ernest, et al. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

Sanjay Chaudhuri, Mathias Drton, and Thomas S Richardson. Estimation of a covariance matrix with zeros. *Biometrika*, 94(1):199–216, 2007.

David Maxwell Chickering. Learning Bayesian networks is NP-complete. In *Learning From Data*, pages 121–130. Springer, 1996.

David Maxwell Chickering. Optimal structure identification with greedy search. *The Journal of Machine Learning Research*, 3:507–554, 2003.

David Maxwell Chickering and Christopher Meek. Finding optimal Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*, pages 94–102. Morgan Kaufmann Publishers Inc., 2002.

Arthur P Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

Arthur Pentland Dempster. *Elements of Continuous Multivariate Analysis*, volume 388. Addison-Wesley Reading, Mass., 1969.

Mathias Drton and Thomas S Richardson. Graphical methods for efficient likelihood inference in Gaussian covariance models. *The Journal of Machine Learning Research*, 9: 893–914, 2008.

Mathias Drton, Rina Foygel, and Seth Sullivant. Global identifiability of linear structural equation models. *The Annals of Statistics*, 39(2):865–886, 2011.

Byron Ellis and Wing Hung Wong. Learning causal Bayesian network structures from experimental data. *Journal of the American Statistical Association*, 103(482), 2008.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

Jianqing Fan and Jinchi Lv. Nonconcave penalized likelihood with NP-dimensionality. *IEEE Transactions on Information Theory*, 57(8):5467–5484, 2011.

Yingying Fan and Jinchi Lv. Asymptotic equivalence of regularization methods in thresholded parameter space. *Journal of the American Statistical Association*, 108(503):1044–1061, 2013.

Thomas Shelburne Ferguson. *A Course in Large Sample Theory*, volume 38. CRC Press, 1996.

Rina Foygel and Mathias Drton. Extended Bayesian information criteria for Gaussian graphical models. In *Advances in Neural Information Processing Systems*, pages 604–612, 2010.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the Graphical Lasso. *Biostatistics*, 9(3):432–441, 2008.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1, 2010.

Fei Fu and Qing Zhou. Learning sparse causal Gaussian networks with experimental intervention: Regularization and coordinate descent. *Journal of the American Statistical Association*, 108(501):288–300, 2013.

Fei Fu and Qing Zhou. Penalized estimation of sparse directed acyclic graphs from categorical data under intervention. *arXiv Preprint arXiv:1403.2310*, 2014.

José A Gámez, Juan L Mateo, and José M Puerta. Learning Bayesian networks by hill climbing: Efficient methods based on progressive restriction of the neighborhood. *Data Mining and Knowledge Discovery*, 22(1-2):106–148, 2011.

José A Gámez, Juan L Mateo, and José M Puerta. One iteration CHC algorithm for learning Bayesian networks: An effective and efficient algorithm for high dimensional problems. *Progress in Artificial Intelligence*, 1(4):329–346, 2012.

Dan Geiger and David Heckerman. Learning Gaussian networks. *arXiv Preprint arXiv:1302.6808*, 2013.

David Heckerman, Dan Geiger, and David M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

Jian Huang, Patrick Breheny, and Shuangge Ma. A selective review of group selection in high-dimensional models. *Statistical Science*, 27(4), 2012.

Jianhua Z Huang, Naiping Liu, Mohsen Pourahmadi, and Linxu Liu. Covariance matrix selection and estimation via penalised normal likelihood. *Biometrika*, 93(1):85–98, 2006.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-Gaussianity. *The Journal of Machine Learning Research*, 11:1709–1731, 2010.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *The Journal of Machine Learning Research*, 8:613–636, 2007.

Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012.

Clifford Lam and Jianqing Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37(6B):4254, 2009.

Wai Lam and Fahiem Bacchus. Learning Bayesian belief networks: An approach based on the MDL principle. *Computational Intelligence*, 10(3):269–293, 1994.

Po-Ling Loh and Peter Bühlmann. High-dimensional learning of linear causal networks via inverse covariance estimation. *arXiv Preprint arXiv:1311.3492*, 2013.

Jinchi Lv and Yingying Fan. A unified approach to model selection and sparse recovery using regularized least squares. *The Annals of Statistics*, 37(6A):3498–3528, 2009.

Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. SparseNet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495): 1125–1138, 2011.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

Jonas Peters, Joris Mooij, Dominik Janzing, and Bernhard Schölkopf. Identifiability of causal graphs using functional models. *arXiv Preprint arXiv:1202.3757*, 2012.

Mohsen Pourahmadi. *High-Dimensional Covariance Estimation*. John Wiley & Sons, 2013.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2014. URL `http://www.R-project.org`.

Garvesh Raskutti and Caroline Uhler. Learning directed acyclic graphs based on sparsest permutations. *arXiv Preprint arXiv:1307.0366*, 2014.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Richard Redner. Note on the consistency of the maximum likelihood estimate for nonidentifiable distributions. *The Annals of Statistics*, 9(1):225–228, 1981.

Robert W Robinson. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V*, pages 28–43. Springer, 1977.

Philipp Rütimann and Peter Bühlmann. High dimensional sparse covariance estimation via directed acyclic graphs. *Electronic Journal of Statistics*, 3:1133–1160, 2009.

Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P Nolan. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Mark Schmidt, Alexandru Niculescu-Mizil, and Kevin Murphy. Learning graphical model structure using L1-regularization paths. In *AAAI*, volume 7, pages 1278–1283, 2007.

Marco Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(i03), 2010.

Marco Scutari. Bayesian network constraint-based structure learning algorithms: Parallel and optimised implementations in the bnlearn R package. *arXiv Preprint arXiv:1406.7648*, 2014.

Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-Gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research*, 7:2003–2030, 2006.

Ali Shojaie and George Michailidis. Penalized likelihood methods for estimation of sparse high-dimensional directed acyclic graphs. *Biometrika*, 97(3):519–538, 2010.

Peter Spirtes and Clark Glymour. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1):62–72, 1991.

Nicolas Städler, Peter Bühlmann, and Sara Van De Geer. ℓ_1-penalization for mixture regression models. *Test*, 19(2):209–256, 2010.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2006.

Sara van de Geer and Peter Bühlmann. ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

Hansheng Wang, Runze Li, and Chih-Ling Tsai. Tuning parameter selectors for the Smoothly Clipped Absolute Deviation method. *Biometrika*, 94(3):553–568, 2007.

Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for Lasso penalized regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

Jing Xiang and Seyoung Kim. A* Lasso for learning a sparse Bayesian network structure for continuous variables. In *Advances in Neural Information Processing Systems*, pages 2418–2426, 2013.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

Qing Zhou. Multi-domain sampling with applications to structural inference of Bayesian networks. *Journal of the American Statistical Association*, 106(496):1317–1330, 2011.

Hui Zou. The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Journal of Machine Learning Research 16 (2015) 2329-2355 Submitted 4/13, Revised 2/14; Published 8/15

Agnostic Insurability of Model Classes

Narayana Santhanam NSANTHAN@HAWAII.EDU
Dept of Electrical Engineering
University of Hawaii at Manoa
Honolulu, HI 96822

Venkat Anantharam ANANTH@EECS.BERKELEY.EDU
EECS Dept
University of California, Berkeley
Berkeley, CA 94720

Editor: Gabor Lugosi

Abstract

Motivated by problems in insurance, our task is to predict finite upper bounds on a future draw from an unknown distribution p over natural numbers. We can only use past observations generated independently and identically distributed according to p. While p is unknown, it is known to belong to a given collection $\mathcal{P}$ of probability distributions on the natural numbers.

The support of the distributions $p \in \mathcal{P}$ may be unbounded, and the prediction game goes on for *infinitely* many draws. We are allowed to make observations without predicting upper bounds for some time. But we must, with probability 1, start and then continue to predict upper bounds after a finite time irrespective of which $p \in \mathcal{P}$ governs the data.

If it is possible, without knowledge of p and for any prescribed confidence however close to 1, to come up with a sequence of upper bounds that is never violated over an infinite time window with confidence at least as big as prescribed, we say the model class $\mathcal{P}$ is *insurable*.

We completely characterize the insurability of any class $\mathcal{P}$ of distributions over natural numbers by means of a condition on how the neighborhoods of distributions in $\mathcal{P}$ should be, one that is both necessary and sufficient.

Keywords: insurance, ℓ_1 topology of probability distributions over countable sets, non-parametric approaches, prediction of quantiles of distributions, universal compression

1. Introduction

Insurance is a means of managing risk by transferring a potential sequence of losses to an *insurer* for a price paid on a regular basis, the *premium*. The insurer attempts to break even by balancing the possible loss that may be suffered by a few with the guaranteed premiums of many. We aim to study the fundamentals of this problem when the losses can be unbounded and a precise model for the probability distribution of the aggregate loss in each period either does not exist or is infeasible to get.

A systematic, theoretical, as opposed to empirical, study of insurance goes back to 1903 when Filip Lundberg defined a natural probabilistic setting as part of his thesis—see, *e.g.,* the chapter on Lundberg in Englund and Martin-Löf (2001). In particular, Lundberg formulated a collective risk problem pooling together the risk of all the insured parties into a single entity, which we call the insured. A substantial part of statistical work on insurance depend on working with specific models for the loss distribution, e.g. compound Poisson models, after which questions of interest in practice, such as the relation between the size of the premiums charged and the probability of the insurer going bankrupt, can be analyzed.

A rather comprehensive theory of insurance along these lines has evolved, see Cramer (1969) and more recently in Asmussen and Albrecher (2010). This theory is able to incorporate several model classes for the distribution of the losses over time other than compound Poisson processes, including some heavy tailed distribution classes.

We will outline our approach in the next subsection. In addition, from a learning theory perspective, our formulation is positioned in between the notions of uniform and pointwise consistency as we outline in subsection 1.2. Finally, in subsection 1.3 we compare our approach to the closely connected notions of universal compression and regret.

1.1 Approach

Our approach departs from the existing literature on insurance in two important respects.

No upper bound on loss. The first departure relates to the practice among insurers to limit payments to a predetermined ceiling, even if the loss suffered by the insured exceeds this ceiling. In both the insurance industry and the legal regulatory framework surrounding it, this is assumed to be common sense. But is it always necessary to impose such ceilings? Moreover, in scenarios such as reinsurance, a ceiling on compensation is not only undesirable, but may also limit the very utility of the business. As we will see, we may be able to handle scenarios where the loss can be unbounded.

Universal approach. The second aspect of our approach arises from our motivation to deal with several new settings for which some sort of insurance is desirable, but where insurers are hesitant to enter the market due to lack of sufficient data. Examples of such settings include insuring against network outages or attacks against future smart grids, where the cascade effect of outages or attacks could be catastrophic. In these settings, it is not clear today what should constitute a reasonable risk model because of the absence of usable information about what might cause the outages or motivate the attacks.

We address the second issue by working with a *class* of models, *i.e.,* a set of probability laws over loss sequences that adheres to any assumptions the insurer may want to make or any information it may already have. In this paper we will only consider loss models that are independent and identically distributed (*i.i.d.*) from period to period, so we can equivalently think of a model class as defined in terms of its one dimensional marginals.

As an example, we may want to consider the set of all finite moment probability distributions over the nonnegative integers as our class of possible models for the loss distribution in each period. Now, we ask the question: what classes of models are the ones on which the insurer can learn from observations and set premiums so as to remain solvent? In this paper, we completely answer this question by giving a necessary and sufficient condition that characterizes what classes of models lend themselves to this insurance task.

This setup for insurability is very reminiscent of the universal compression/estimation/prediction approaches as seen in Shtarkov (1987); Fittingoff (1972); Rissanen (1984) and Ryabko (2008). There is also extensive work regarding learning from experts that has a related flavor, see Cesa-Bianchi and Lugosi (2006) for a survey. We will discuss the compression angle in more depth shortly as well.

Formulation. Formally, we adopt the collective risk approach, namely, we abstract the problem to include just two agents, the insurer and the insured. Losses incurred by the insured are considered to

form a discrete time sequence of random variables, with the sequence of losses denoted by $\{X_i,\ i \geq 1\}$, and we assume that $X_i \in \mathbb{N}$ for all $i \geq 1$, where $\mathbb{N}$ denotes the set of natural numbers, $\{0, 1, 2, \ldots\}$.

A *model class* $\mathcal{P}^\infty$ is a collection of measures on infinite length loss sequences, and is to be thought of as the set of all potential probability laws governing the loss sequence. Each element of $\mathcal{P}^\infty$ is a *model* for the sequence of losses. Any prior knowledge on the structure of the problem is accounted for in the definition of $\mathcal{P}^\infty$. We focus on measures corresponding to *i.i.d.* samples, i.e. each member of $\mathcal{P}^\infty$ induces marginals that are product distributions. We denote by $\mathcal{P}$ the set of distributions on $\mathbb{N}$ obtained as one dimensional marginals of $\mathcal{P}^\infty$. Since there is no risk of confusion, we will also refer to the distributions in $\mathcal{P}$ as models and to $\mathcal{P}$ as the model class.

The actual model in $\mathcal{P}$ governing the law of the loss in each period remains unknown to the insurer. We assume no ceiling on the loss, and require the insurer to compensate the insured in full for the loss in each period at the end of that period. The insurer is assumed to start with some initial capital $\Pi_0 \in \mathbb{R}^+$, a nonnegative real number. The insurer then sets a sequence of premiums based on the past losses—at time i, the insurer collects a premium $\Pi(X_1^{i-1})$ at the beginning of the period, and pays out full compensation for loss X_i at the end of the period. If the built up capital till step i (including $\Pi(X_1^{i-1})$, and after having paid out all past losses) is less than X_i, the insurer is said to be *bankrupted*.

Given a class $\mathcal{P}^\infty$ of loss models, we ask if, for every prescribed upper bound $\eta > 0$ on the probability of bankruptcy, the insurer can set (finite) premiums at every time step based only on the loss sequence observed thus far and with no further knowledge of which law $p \in \mathcal{P}^\infty$ governs the loss sequence, while simultaneously ensuring that the insurer remains solvent with probability bigger than $1 - \eta$ under p irrespective of which $p \in \mathcal{P}^\infty$ is in effect. If the probability of the insurer ever going bankrupt over an infinite time window can be made arbitrarily small in this sense, the class of *i.i.d.* loss measures $\mathcal{P}^\infty$ is said to be *insurable*.

A couple of clarifications are in order here. First, to make the problem non-trivial, we allow the insurer to observe the loss sequence for some arbitrary finite length of time without having to provide compensations. We require that the insurer has to eventually provide insurance with probability 1 no matter which $p \in \mathcal{P}^\infty$ is in effect. The insurer cannot quit providing insurance once it has entered into the insurance contract with the insured. Premiums set before the entry time can be thought of as being 0 and the question of bankruptcy only arises after the insurer has entered into the contract. Secondly, at this point of research, we do not concern ourselves with incentive compatibility issues on the part of the insured and assume that the insured will accept the contract once the insurer has entered, agreeing to pay the premiums as set by the insurer.

It turns out that the fact that the capital available to the insurer at any time is built up from past premiums does not play any role in whether a model class is insurable or not. In fact, the problem is basically one of finding a sequence of finite upper bounds $\Phi(X_1^{i-1})$ on the loss X_i for all $i \geq 1$. We refer to the sequence $\{\Phi(X_1^{i-1}),\ i \geq 1\}$ as the *loss dominating sequence* and call $\Phi(X_1^{i-1})$ the *loss dominant* at step i.

The notion of insurability of a model class $\mathcal{P}$ comes down to whether for each $\eta > 0$ there is a way of choosing the loss dominants such that the probability of the loss X_i ever exceeding the loss dominant $\Phi(X_1^{i-1})$ is smaller than η irrespective of which model p in the model class $\mathcal{P}^\infty$ is in effect. Here again we allow some initial finite number of periods for which the loss dominant can be set to ∞, but it must become finite with probability 1 under each $p \in \mathcal{P}^\infty$ and stay finite from that point onwards.

Results. For a model class to be insurable, roughly speaking, close distributions must have comparable percentiles. Distributions in the model class that, in every neighborhood, have some other distribution with arbitrarily different percentiles are said to be *deceptive*. In Section 3, we define what

it means for distributions to be close, and what it means for distributions to have comparable percentiles. In Section 4, we provide several examples of insurable and non-insurable model classes. Our main result is Theorem 1 of Section 3, which states that $\mathcal{P}^\infty$ is insurable iff it has no deceptive distributions. We prove this theorem in Sections 5 and 6. In the rest of this section, we discuss the problem of insurability in the broader contexts of uniform and pointwise convergence of estimators and universal compression.

1.2 Pointwise vs. Uniform Convergence

Theoretically, the flexibility we have permitted regarding when to start proposing finite loss dominants allows us to categorize the insurance problem formulated above as one that admits what we call *data-derived* pointwise convergent estimators.

When dealing with large alphabets or high dimensions, it may be too restrictive to only deal with model classes or problem formulations that admit uniformly convergent estimators. We are particularly interested in richer cases where uniform learnability may not be possible.

In such cases, often guarantees on estimators are provided that hold *pointwise* for all models. Typically, the results proven in such cases are of consistency, or bounds on rate of convergence of estimators that depend on the parameters of the model in force—of course, the model by itself is usually not known *a-priori*. Therefore the practical issue with such pointwise guarantees is that they may not say much about what is happening with the specific sample at hand. Namely, if we know that an estimator for a problem is consistent and we have model dependent convergence bounds, for a given sample (from an unknown model) there may be no way of telling how good or bad the estimate currently is.

It can be shown that the insurance problem outlined here is equivalent to learning an upper bound on every percentile of the unknown distribution from $\mathcal{P}$, using only samples of *i.i.d.* draws from the distribution. However, we allow for model classes that are rich enough that there may be no bound on a given percentile that holds uniformly over the entire class $\mathcal{P}$.

Yet, we can still salvage the situation if for any given finite sample, we had some way to tell from the sample if the estimate was doing well or not relative to the true unknown model. In other words, we ask for model dependent convergence bound that depends on the model only through the sample that we observe.

Data-driven pointwise convergence is at the heart of insurability as well, and it shows up specifically because we provide a finite (but not bounded uniformly over all models and observations) observation window before the insurer enters the game. While we consider complex $\mathcal{P}$ where there may be no possible way to bound any given percentile over the entire class, we can tell when the bounds obtained from a sample is good against the model that generated the sample. The point at which we decide our bounds are good against the model underlying the sample observed is related to the entry point defined in the formulation above. In section 4 we provide several examples of classes that are insurable and those that are not.

This principle of using the sample to gauge the performance of a pointwise consistent estimate has been incorporated into several other setups as well. A few examples that stand out are the notions of luckiness NML in compression proposed in Grunwald (2007), in the PAC-Bayesian bounds for classification in McAllester (2013) and in the estimation of slow mixing Markov processes in Asadi et al. (2014). For the luckiness NML formulation, an appropriate slack function can be interpreted as a bound on how far we are from the code length of the underlying source. The PAC-Bayesian bounds of McAllester (2013) can also be used to provide data-dependent confidence bounds for classifiers in scenarios where classes may be very rich. In Asadi et al. (2014) again, data-dependent confidence bounds are provided

for the estimates of transition and stationary probabilities of an underlying slow mixing, long memory Markov sources.

To illustrate this concept, we provide a simple running example below that we will also use to demonstrate connections with compression in subsection 1.3.

Example 1 [Birthday Problem] Consider the problem of estimating the size of a discrete, finite, set $S \subset \mathbb{N}$ (specifically, there is no upper bound on the size of S) if we can draw as many random samples from S as needed. If we have independent, uniform draws from S, one simple way to estimate the size of S is keep sampling till some element from S is drawn twice. This is the first *repeat*. A simple back of the envelope calculation analogous to the Birthday Problem shows that if N_1 is the sample size when the first repeat occurred, then a good estimate of the set size is $N_1^2/2$. One can then provide PAC-learning kind of bounds that, with some confidence, the above estimate based on the first repeat is accurate to a certain level.

This is an example where there can be no uniformly convergent estimator of the set size. Given a fixed sample of size n, if the size of S is $\Omega(n^2)$ the sample consists of n distinct symbols with probability close to 1. If we can assume no further structure on S, there is no way to distinguish between samples obtained from any two sets with size $\Omega(n^2)$—thus no estimator can distinguish between these large sets. It is therefore futile, with a finite sample size, to expect an estimator that can estimate the set size to any non-trivial accuracy no matter what the set is. Equivalently, there can be no uniformly convergent estimator of the set size.

The simple "Birthday Problem" estimator above only converges pointwise. It may take an arbitrarily larger sample for some models to give an answer. That is the nature of the problem. But the estimator is imbued with a very useful property—with a guaranteed confidence, the estimator does not make a mistake even though it may not always have an answer. If the sample has no repeats yet, the estimator does not overreach and volunteer a wrong answer. Hence, we can tell from the sample if we can do well or not. □

1.3 Universal Compression

The approach we take is not unconnected with the universal compression literature, as well as certain learning formulations involving regret with log-loss. There are many variations in how compression problems are formulated, and the following example illustrates the main variants studied. It also serves to provide an example of the "data-derived" pointwise estimation introduced in the last section, and hence places insurability in context of the universal compression literature.

Example 2 We will study the so-called "Birthday problem" from the previous example in a little more depth. Let $\mathcal{B}$ denote the collection of all distributions p_M ($M \geq 1$), where p_M is a uniform distribution over $\{0, \ldots, M\}$. We use this example to distinguish between uniformly good compressors (strongly universal) and compressors that are only good pointwise (weakly universal).

Suppose we consider one draw from an unknown distribution $p \in \mathcal{B}$. The worst case regret or worst-case *redundancy* quantifies the minimum possible excess code length of a universal distribution q over $\mathbb{N}$ over the (unknown) distribution p

$$\inf_q \sup_{p \in \mathcal{B}} \sup_{x \in \mathbb{N}} \log \frac{p(x)}{q(x)}. \tag{1}$$

Note that the regret of any class of distributions is always ≥ 0. We could, of course, consider a sequence of n independent draws from an unknown $p \in \mathcal{B}$, and ask now for a measure q over infinite sequences

of numbers that is universal for all the *i.i.d.* measures corresponding to distributions in $\mathcal{B}$. We then concentrate on the *redundancy*

$$R_n(\mathcal{B}) \stackrel{\text{def}}{=} \inf_q \sup_{p\in\mathcal{B}} \sup_{x^n\in\mathbb{N}^n} \frac{1}{n}\log\frac{p(x^n)}{q(x^n)},$$

where we abuse notation and write for all $p \in \mathcal{B}$

$$p(x^n) = \prod_{j=1}^{n} p(x_j)$$

to be the probability assigned to x^n by independent draws from p. Of course, we could similarly define redundancy for length n sequences for any collection of measures over infinite sequences from a countable alphabet, not necessarily *i.i.d.*. *Strongly compressible* classes are those sets $\mathcal{P}^\infty$ of measures over infinite sequences satisfying

$$\lim_{n\to\infty} R_n(\mathcal{P}^\infty) \to 0.$$

For the single-letter formulation in (1), clearly the optimal universal distribution gives any number x a probability proportional to the highest probability that number gets from any model in $\mathcal{B}$, followed by a normalization. But the highest probability a model in $\mathcal{B}$ gives any $x \in \mathbb{N}$ is $1/(x+1)$, which is not summable over x. Thus the redundancy is infinite here—or equivalently, no matter what universal distribution q we choose and no matter how large a number M we pick, there is a $p' \in \mathcal{B}$ and a number $x' \in \mathbb{N}$ such that

$$\log\frac{p'(x')}{q(x')} > M.$$

In this case, we will therefore not have redundancy bounds holding uniformly for the model class. We say $\mathcal{B}$ is not strongly compressible. With a very similar argument, it is easy to see that $R_n(\mathcal{B}^\infty) = \infty$ for all n, where we use $\mathcal{B}^\infty$ to denote the set of *i.i.d.* measures over infinite sequences constructed as above from marginals in $\mathcal{B}$.

But we can say something more. Consider again compressing sequences of numbers drawn *i.i.d.* from an unknown distribution in $\mathcal{B}$. Noting that $\mathcal{B}$ is countable, we focus on a measure q over infinite sequences that gives a sequence x^n the probability

$$q_w(x^n) = \sum_{p_i\in\mathcal{B}} \frac{1}{i(i+1)} p_i(x^n).$$

It is easy to verify that q_w above satisfies

$$\sup_{p\in\mathcal{B}^\infty} \lim_{n\to\infty} \sup_{x^n\in\mathbb{N}^n} \frac{1}{n}\log\frac{p(x^n)}{q_w(x^n)} = 0, \tag{2}$$

or that q_w matches every p *pointwise* over the model class $\mathcal{B}$. Such classes of sources are *weakly universal*. The code length of the universal measure q_w matches that of p for every $p \in \mathcal{B}$, but at arbitrarily slower rates for some sources (since the class cannot be strongly compressed).

A couple of points. Note that admittedly it has been easy to define q_w here since $\mathcal{B}$ was countable to begin with. If not, a condition reminiscent of countability above is necessary and sufficient for a class to be weakly universal as shown in (Kieffer, 1978). To emphasize, q_w is guaranteed to satisfy (2),

implying that it does not underestimate relative to any p for sufficiently long sequences. But how long is "sufficiently long" depends on p, and for a given length-n sequence without knowing p, it may not be possible to say if q_w is doing well or not. This second aspect of knowing when an estimator is good is crucial to insurance formulations.

One could also replace the sup over x^n with expectation, and get average case versions for both strong and weak' universality. □

Strong compression is well known and is the more studied version of universal compression and regret formulations involving log loss. As one might expect, we show in (Santhanam and Anantharam, 2012) that strong compression implies insurability but not vice-versa.

However, the insurance formulation has more in common with weak compression and pointwise convergence, rather than strong compression and uniform convergence. The connection between insurability and weak compression turns out to be rather interesting. In (Santhanam and Anantharam, 2012), we show classes of models that can be weakly compressed but are not insurable. At the same time, we also construct classes of models in (Santhanam and Anantharam, 2012) that are insurable, but cannot be weakly compressed.

To summarize, our formulation is interesting precisely in cases where the strong notions of (worst-case or average-case) redundancy fail. Namely, classes of distributions whose redundancy is not finite. The universal compression formulation closer to our notion of insurability here is in the idea of weak universal compression in Kieffer (1978). However, weak compression formulations thus far have never included the aspect of determining from the data at hand *when* a compressor is doing well—a crucial part of our problem.

There may be one insight that we conjecture can be generalized beyond insurability to all problems with the flavor of data-derived pointwise convergence of estimates. What matters seems to be the local complexity as opposed to global complexity of model classes. Insurability of model classes does not depend on global complexity measures of model classes—as with the (strong) redundancy of model classes (which is determined by the integral of the square root of absolute Fisher information over the entire model class) or the Rademacher complexity. Instead, insurability is related to how local neighborhoods look; in particular it depends on *local tightness* as we will see in Section 3.

2. Precise Formulation of Insurability

We model the loss at each time by a random variable taking values in $\mathbb{N} = \{0, 1, \ldots\}$. Denote the sequence of losses by $X_1, X_2 \ldots$ where $X_i \in \mathbb{N}$. Let $\mathbb{N}^*$ be the set of all finite length sequences from $\mathbb{N}$, including the empty sequence. We will write x^n for the sequence $x_1, \ldots, x_n$. Where it appears, x^0 denotes the empty sequence. A loss distribution is a probability distribution on $\mathbb{N}$. Let $\mathcal{P}$ be a set of loss distributions. $\mathcal{P}^\infty$ is the collection of *i.i.d.* measures over infinite sequences of symbols from $\mathbb{N}$ such that the set of one dimensional marginals over $\mathbb{N}$ they induce is $\mathcal{P}$.

We write $\mathbb{R}^+$ for the set of nonnegative real numbers and use $:=$ for equality by definition.

Consider an insurer with an *initial capital* $\Pi_0 \in \mathbb{R}^+$. An *insurance scheme* for $\mathcal{P}$ is comprised of a pair (τ, Π).

Here $\tau : \mathbb{N}^* \mapsto \{0, 1\}$ satisfies $\tau(x_1, \ldots, x_n) = 1 \implies \tau(x_1, \ldots, x_{n+1}) = 1$ for all x^n and also $p(\sup_n \tau(X^n) = 1) = 1$ for all $p \in \mathcal{P}^\infty$. τ should be thought of as defining an *entry time* for the insurer with the property that once the insurer has entered it stays entered and that the insurer enters with probability 1 irrespective of which $p \in \mathcal{P}^\infty$ is in effect. Here we say the insurer enters after seeing the sequence $x^n \in \mathbb{N}^*$ (possibly the empty sequence) if $\tau(x^n) = 1$. The other ingredient of an insurance

scheme is the *premium setting scheme* $\Pi : \mathbb{N}^* \to \mathbb{R}^+$, satisfying $\Pi(x^n) = 0$ if $\tau(x^n) = 0$, with $\Pi(x^n)$ being interpreted as the premium demanded by the insurer from the insured after the loss sequence $x^n \in \mathbb{N}^*$ is observed.

Let $1(\cdot)$ denote the indicator function of its argument. The event that the insurer goes bankrupt is the event that

$$\Pi_0 + \sum_{i=1}^{n} (\Pi(X^{i-1}) - X_i) 1(\tau(X^{i-1}) = 1) < 0 \text{ for some } n \geq 1 \ .$$

In words, this is the event that in some period $n \geq 1$ after the insurer has entered, the loss X_n incurred by the insured exceeds the built up capital of the insurer, namely the sum of its initial capital and all the premiums it has collected after it has entered (including the currently charged premium $\Pi(X^{n-1})$) less all the losses paid out so far.

Definition 1 A class $\mathcal{P}^\infty$ of laws on loss sequences is called insurable by an insurer with initial capital $\Pi_0 \in \mathbb{R}^+$ if $\forall\ \eta > 0$, there exists an insurance scheme (τ, Π) such that $\forall\ p \in \mathcal{P}^\infty$,

$$p((\tau, \Pi) \text{ goes bankrupt }) < \eta \ .$$

We should remark that despite the apparent role of the initial capital of the insurer in this definition, it plays no role from a mathematical point of view. To see this note first that if a model class $\mathcal{P}^\infty$ is insurable by an insurer with capital Π_0 it is clearly insurable by all insurers with initial capital at least Π_0, since such an insurer can use the same entry time and premium setting scheme as the insurer with initial capital Π_0. On the other hand, an insurer with initial capital less than Π_0 can use the same entry time as an insurer with initial capital Π_0 and simply charge an additional premium at the time of entry which in effect builds up its initial capital to Π_0, and then proceed with the same premium setting scheme as that used by the insurer with initial capital Π_0. This feature is an artifact of the complete flexibility we give the insurer in setting premiums; for more on this see the concluding remarks in Section 7.

As indicated in the introductory Section 1, we will first show that whether a model class of loss distributions is insurable is equivalent to whether we can find suitable loss-domination sequences for the sequence of losses. We next make this connection and the associated terminology precise.

Definition 2 A *loss-domination scheme* for $\mathcal{P}$ is a mapping $\Phi : \mathbb{N}^* \mapsto \mathbb{R}^+ \cup \{\infty\}$, where for $x^n \in \mathbb{N}^*$, we interpret $\Phi(x^n)$ as an estimated upper bound on x_{n+1}. We call $\{\Phi(X^{i-1}),\ i \geq 1\}$ the loss-domination sequence and $\Phi(X^{i-1})$ the loss dominant at step i. We require for all $x^n \in \mathbb{N}^*$ that

$$\Phi(x_1, \ldots, x_n) < \infty \Longrightarrow \Phi(x_1, \ldots, x_{n+1}) < \infty$$

and also that for all $p \in \mathcal{P}^\infty$,

$$p(\inf_{n \geq 1} \Phi(X^n) < \infty) = 1. \qquad \square$$

We think of $\Phi(x^n) = \infty$ as saying that the scheme has not yet committed to proposing finite loss dominants after having seen the sequence x^n, while if $\Phi(x^n) < \infty$ it has. Once the scheme commits to proposing finite loss dominants it has to continue to propose finite loss dominants from that point onwards. Further, with probability 1 under every $p \in \mathcal{P}^\infty$, the scheme has to eventually start proposing finite loss dominants.

Definition 3 Given our motivation from the insurance problem, we will say the loss-domination scheme Φ goes *bankrupt* if $\Phi(X^{n-1}) < X_n$ for some $n \geq 1$. □

The connection between the insurance problem and the problem of selecting loss dominants can now be made precise as follows.

Observation 1 Let $\mathcal{P}^\infty$ be a model class and $\eta > 0$. Let $\Pi_0 \in \mathbb{R}^+$. An insurer with initial capital Π_0 can find an insurance scheme (τ, Π) such that the probability of remaining solvent is bigger than $1 - \eta$ irrespective of which $p \in \mathcal{P}^\infty$ is in effect if and only if there is a loss-domination scheme Φ such that the probability of it going bankrupt is less than η irrespective of which $p \in \mathcal{P}^\infty$ is in effect.

Proof Given an insurance scheme (τ, Π) consider the loss-domination scheme Φ that has $\Phi(x^n) := \infty$ iff $\tau(x^n) = 0$ and

$$\Phi(X^{n-1}) := \Pi_0 + \sum_{i=1}^{n-1} (\Pi(X^{i-1}) - X_i) 1(\tau(X^{i-1}) = 1) \; + \; \Pi(X^{n-1}) \; ,$$

if $\tau(X^n) = 1$. Since τ enters (become equal to 1) with probability 1 under each $p \in \mathcal{P}^\infty$ and stays equal to 1 once it has become 1, Φ becomes finite with probability 1 under each $p \in \mathcal{P}^\infty$ and stays finite once it has become finite. Thus Φ is indeed a loss-domination scheme. It is straightforward to check that if the insurance scheme (τ, Π) stays solvent with probability bigger than $1 - \eta$ irrespective of which $p \in \mathcal{P}^\infty$ is in effect then the loss-domination scheme Φ becomes bankrupt with probability less than η irrespective of which $p \in \mathcal{P}^\infty$ is in effect.

Conversely, given a loss-domination scheme Φ define the insurance scheme (τ, Π) by setting $\tau(x^n) := 0$ iff $\Phi(X^n) = \infty$ (and $\tau(x^n) := 1$ iff $\Phi(x^n) < \infty$) and defining $\Pi(x^n) := 0$ if $\Phi(x^n) = \infty$ and $\Pi(x^n) := \Phi(x^n)$ if $\Phi(x^n) < \infty$.

One sees that τ as defined becomes 1 with probability 1 under each $p \in \mathcal{P}^\infty$ and stays equal to 1 once it becomes 1. Further, the premiums set at each time are finite and equal to 0 till the entry time. Thus (τ, Π) as defined is indeed an insurance scheme.

It is straightforward to check if Φ becomes bankrupt with probability less than η irrespective of which $p \in \mathcal{P}^\infty$ is in effect, then (τ, Π) stays solvent with probability bigger than $1 - \eta$ irrespective of which $p \in \mathcal{P}^\infty$ is in effect. Hence the above observation. □

We may therefore conclude that a model class $\mathcal{P}^\infty$ is insurable iff for all $\eta > 0$ there is a loss-domination scheme Φ such that the probability of going bankrupt under Φ is less than η irrespective of which $p \in \mathcal{P}^\infty$ is in effect. In the rest of the paper we will therefore focus mainly on whether the model class $\mathcal{P}^\infty$ is such that for every $\eta > 0$ a loss-domination sequence Φ exists with its probability of bankruptcy being less than η irrespective of which model in the model class governs the sequence of losses.

In Theorem 1, we provide a condition on $\mathcal{P}$ that is both necessary and sufficient for insurability.

3. Statement of Main Result

We go through a few technical points before spelling out the results in detail in 3.3.

3.1 Close Distributions

Insurability of $\mathcal{P}^\infty$ depends on the neighborhoods of the probability distributions among its one dimensional marginals $\mathcal{P}$. The relevant measure of closeness between distributions in $\mathcal{P}$ that decides the

neighborhoods is

$$\mathcal{J}(p,q) := D\left(p||\frac{p+q}{2}\right) + D\left(q||\frac{p+q}{2}\right).$$

Note that the above is the Jensen-Shannon divergence (with an additional factor of 2) and is *not* a true distance. Here $D(p||q)$ denotes the relative entropy of p with respect to q, where p and q are probability distributions on $\mathbb{N}$, defined by

$$D(p||q) := \sum_{y\in\mathbb{N}} p(y)\log\frac{p(y)}{q(y)} \ .$$

The logarithm is assumed to be taken to base 2 (we use ln for the logarithm to the natural base).

The reason for choosing the Jensen-Shannon (JS) divergence is that it has two convenient properties—(i) for the necessary part, it becomes easy to quantify how "close" distributions yield very similar measures on sequences, (ii) for the sufficient part, we bound the JS divergence with the ℓ_1 norm in Lemma 4 which in turn lets us work with the ℓ_1 topology induced on the class $\mathcal{P}$ of distributions. Specifically, we show that if p and q are probability distributions on $\mathbb{N}$, then

$$\frac{1}{4\ln 2}|p-q|_1^2 \leq \mathcal{J}(p,q) \leq \frac{1}{\ln 2}|p-q|_1 \ .$$

3.2 Cumulative Distribution Function

Since we would like to discuss percentiles, it is convenient to use a non-standard definition for the cumulative distribution function of a probability distribution on $\mathbb{N}$.

For our purposes, the cumulative distribution function of any probability distribution p on $\mathbb{N}$ is a function $F_p : \mathbb{R}^+\cup\{\infty\} \to [0,1]$ defined in an unconventional way. We obtain F_p by first defining F_p on points in the support of p in the way cumulative distribution functions are normally defined. We define F_p for all other nonnegative real numbers by linearly interpolating between the values in the support of p. Finally, $F_p(\infty) := 1$.

Let $F_p^{-1} : [0,1] \mapsto \mathbb{R}^+ \cup \{\infty\}$ denote the inverse function of F_p. Then $F_p^{-1}(x) = 0$ for all $0 \leq x < F_p(0)$. If p has infinite support then $F_p^{-1}(1) = \infty$, else $F_p^{-1}(1)$ is the smallest natural number y such that $F_p(y) = 1$.

Two simple and useful observations can now be made. Consider a probability distribution p with support $\mathcal{A} \subset \mathbb{N}$. For $\delta > 0$, let (T for tail)

$$T_{p,\delta} := \{y \in \mathcal{A} : y \geq F_p^{-1}(1-\delta)\},$$

and let (H for head)

$$H_{p,\delta} := \{y \in \mathcal{A} : y \leq 2F_p^{-1}(1-\delta/2)\}.$$

It is easy to see that

$$p(T_{p,\delta}) > \delta \tag{3}$$

and that

$$p(H_{p,\delta}) > 1-\delta. \tag{4}$$

Suppose that for some $\delta > 0$ we have $F_p^{-1}(1-\delta) > 0$ and the loss dominant at the beginning of period $i \geq 1$ happens to be set to $F_p^{-1}(1-\delta)$, then the probability under p of the loss in period i exceeding the loss dominant is bigger than δ. If the loss dominant at the beginning of period i happens to be set to $2F_p^{-1}(1-\delta/2)$, then the probability that the loss in period i exceeds the loss dominant is less than δ. We will use these observations in the proofs to follow.

3.3 Condition that is Necessary and Sufficient for Insurability

Existence of close distributions with very different quantiles is what kills insurability. A loss-domination scheme could be "deceived" by some process $p \in \mathcal{P}^\infty$ into setting low loss dominants, while a close enough distribution hits the scheme with too high a loss. The conditions for insurability of $\mathcal{P}^\infty$ are phrased in terms of the set of its one dimensional marginals, $\mathcal{P}$.

Formally, a distribution p in $\mathcal{P}$ is *not deceptive* if some neighborhood around p is *tight*. For a more complete development of the concept of tightness, see *e.g.,* Billingsley (1995). Specifically, $\exists\ \epsilon_p > 0$, such that $\forall\ \delta > 0$, $\exists\ f(\delta) \in \mathbb{R}$, such that all distributions $q \in \mathcal{P}$ with

$$\mathcal{J}(p, q) < \epsilon_p$$

satisfy

$$F_q^{-1}(1 - \delta) \leq f(\delta).$$

Equivalently, a probability distribution p in $\mathcal{P}$ is *deceptive* if no neighbourhood of $\mathcal{P}$ around p is tight. Specifically, $\forall\ \epsilon > 0$, $\exists\ \delta > 0$ such that that no matter what $f(\delta) \in \mathbb{R}^+$ is chosen, $\exists$ a (bad) distribution $q \in \mathcal{P}$ such that

$$\mathcal{J}(p, q) < \epsilon$$

and

$$F_q^{-1}(1 - \delta) > f(\delta).$$

In the above definition, $f(\delta)$ is simply an arbitrary nonnegative real number. However, it is useful to think of this number as the evaluation of a function $f : (0, 1) \to \mathbb{R}$ at δ.

Our main theorem is the following, which we prove in Sections 5 and 6.

Theorem 1 $\mathcal{P}^\infty$ is insurable, iff no $p \in \mathcal{P}$ is deceptive. □

4. Examples

Consider $\mathcal{U}$, the collection of all uniform distributions over finite supports of form $\{m, m + 1, \ldots, M\}$ for all positive integers m and M with $m \leq M$. Let the sequence of losses be *i.i.d.* samples from distributions in $\mathcal{U}$—call the resulting model class over infinite loss sequences $\mathcal{U}^\infty$.

Note that no distribution in $\mathcal{U}$ is deceptive. Around each distribution in $\mathcal{U}$ is a neighborhood that that contains no other distribution of $\mathcal{U}$.

Example 3 $\mathcal{U}^\infty$ is insurable.

Proof If the threshold probability of ruin is η, choose the loss-domination scheme Φ as follows. For all sequences x^n with $n \leq \log \frac{1}{\eta} + 1$ set $\Phi(x^n) = \infty$. For all sequences x^n with $n > \log \frac{1}{\eta} + 1$, the loss dominant $\Phi(x^n)$ is set to be twice the largest loss observed thus far. It is easy to see that this scheme is bankrupted with probability less than η irrespective of which $p \in \mathcal{U}^\infty$ is in effect. □

Consider the set $\mathcal{N}_1^\infty$ of all *i.i.d.* processes such that the one dimensional marginals have finite first moment. Namely, $\forall p \in \mathcal{N}_1^\infty$, $\mathbb{E}_p X < \infty$ where $X \in \mathbb{N}$ is distributed according to the single letter marginal of p. If $\mathcal{N}_1$ is the collection of single letter marginals from $\mathcal{N}_1^\infty$, it is easy to verify as below that every distribution in $\mathcal{N}_1$ is deceptive.

Example 4 $\mathcal{N}_1^\infty$ is not insurable.

Proof Note that the loss process that puts probability 1 on the all zero sequence exists in $\mathcal{N}_1^\infty$, since it corresponds to the one dimensional marginal loss distribution that produces loss 0 in each period. Since every loss-domination scheme enters with probability 1 no matter which $p \in \mathcal{N}_1^\infty$ is in force, every loss-domination scheme must enter after seeing some finite number of zeros. Fix any loss-domination scheme Φ. Suppose the scheme starts to set finite loss dominants after seeing N losses of size 0. To show that $\mathcal{N}_1^\infty$ is not insurable, we show that $\exists \eta > 0$ and $\exists p \in \mathcal{N}_1^\infty$ such that

$$p(\ \Phi \text{ goes bankrupt }) \geq \eta.$$

Fix $\delta = 1 - \eta$. Let ϵ be small enough that

$$(1-\epsilon)^N > 1 - \delta/2,$$

and let M be a number large enough that

$$(1-\epsilon)^M < \delta/2.$$

Note that since $1 - \delta/2 \geq \delta/2$, we have $N < M$. Let L be greater than any of the loss dominants set by Φ for the sequences $0^N, 0^{N+1}, \ldots 0^M$. Let $p \in \mathcal{N}_1^\infty$ satisfy, for all i,

$$p(X_i) = \begin{cases} 1-\epsilon & \text{if } X_i = 0 \\ \epsilon & \text{if } X_i = L. \end{cases}$$

For the *i.i.d.* loss process having the law p, the insurer is bankrupted on all sequences that contain loss L in between the N-th and M-th steps. These sequences, $0^N L, 0^{N+1} L, \ldots, 0^{M-1} L$, have respective probabilities (under p)

$$(1-\epsilon)^N \epsilon, (1-\epsilon)^{N+1}\epsilon, \ldots, (1-\epsilon)^{M-1},$$

and they also form a prefix free set. Therefore, summing up the geometric series and using the assumptions on ϵ above,

$$p(\ \Phi \text{ is bankrupted }) \geq (1-\epsilon)^N - (1-\epsilon)^M \geq 1 - \delta/2 - \delta/2 = \eta. \qquad \square$$

A reading of the proof above shows that we can say something much stronger. The distributions that break insurability have *all* their moments finite. Suppose $\mathcal{N}_*^\infty$ is the collection of measures each of whose single letter marginal has all moments finite. Namely for all $p \in \mathcal{N}_*^\infty$ and all finite $r \geq 0$, $\mathbb{E}_p X_1^r < \infty$. It follows that

Example 5 $\mathcal{N}_*^\infty$ is not insurable. $\square$

Consider the collection of all *i.i.d.* loss distributions with monotone one dimensional marginals. A monotone probability distribution p on $\mathbb{N}$ is one that satisfies $p(y+1) \leq p(y)$ for all $y \in \mathbb{N}$. Let $\mathcal{M}^\infty$ be the set of all *i.i.d.* loss processes, with one dimensional marginal distribution from $\mathcal{M}$, the collection of all monotone probability distributions over $\mathbb{N}$.

Again, it is easily shown that every distribution in $\mathcal{M}$ is deceptive. It follows from Theorem 1 that

Example 6 $\mathcal{M}^\infty$ is not insurable.

Proof To see that any distribution $p \in \mathcal{M}$ is deceptive, consider distributions of form $p' = (1-\epsilon)p + \epsilon q$, where $q \in \mathcal{U}$ is a monotone uniform distribution and $\epsilon > 0$.

Clearly, the ℓ_1 distance between p' and q is $\leq 2\epsilon$ (and therefore so is the JS divergence, up to a constant factor). But for all $M > 0$ and $\delta < \epsilon$, we can pick $q \in \mathcal{U}$ over a sufficiently large support that the $1-\delta-$percentile of p' can be made $\geq M$. Therefore, no neighborhood around p' is tight. $\square$

Note also that if p has finite entropy, so does every p' obtained by the above construction. Let $\mathcal{M}_* \subset \mathcal{M}$ be the collection of finite entropy monotone distributions. The above example also implies that

Example 7 $\mathcal{M}_*^\infty$ is not insurable. □

Now for $h > 0$, we consider the set $\mathcal{M}_h \subset \mathcal{M}$ of all monotone distributions over $\mathbb{N}$ whose entropy is bounded above by h. Let $\mathcal{M}_h^\infty$ be the set of all *i.i.d.* loss processes with one dimensional marginals from $\mathcal{M}_h$. Then

Example 8 $\mathcal{M}_h^\infty$ is insurable.
Proof From Markov inequality, if $p \in \mathcal{M}_h$ and $X \sim p$,

$$p(X > M) = p(\log(X+1) > \log(M+1)) < \frac{E_p \log(X+1)}{\log(M+1)} \leq \frac{E_p \log \frac{1}{p(X)}}{\log(M+1)} \leq \frac{h}{\log(M+1)}.$$

To see the second inequality above, note that p is monotone therefore for $i \in \mathbb{N}$, $p(i) \leq \frac{1}{i+1}$. Therefore, for all $p \in \mathcal{M}_h$,

$$F_p^{-1}(1-\delta) \leq 2^{\frac{h}{\delta}}.$$

Thus no $p \in \mathcal{M}_h$ is deceptive, and $\mathcal{M}_h^\infty$ is insurable. □

In the class $\mathcal{U}$ above, there was a neighborhood around each distribution $p \in \mathcal{U}$ with no other model from $\mathcal{U}$, Hence $\mathcal{U}$ trivially satisfied the local tightness condition that we will prove is necessary and sufficient for insurability. The above case is another extreme—the entire model class $\mathcal{M}_h$ is tight. The following example illustrates a insurable class of models where neither extreme holds.

For a distribution q over $\mathbb{N}$, let $q^{(R)}(i+R) = q(i)$ for all $i \in \mathbb{N}$. Furthermore let the *span* of any finite support probability distribution over naturals be the largest natural number which has non-zero probability. Then, let

$$\mathcal{F}_h = \left\{(1-\epsilon)p_1 + \epsilon p_2^{(\text{span}(p_1)+1)} : \forall p_1 \in \mathcal{U}, p_2 \in \mathcal{M}_h \text{ and } 1 > \epsilon > 0\right\}.$$

As always $\mathcal{F}_h^\infty$ is the set of measures on infinite sequences obtained by *i.i.d.* sampling from distributions in $\mathcal{F}_h$.

Example 9 $\mathcal{F}_h^\infty$ is insurable.
Proof Let the *base* of any probability distribution over the naturals be the smallest natural number which has non-zero probability. Consider any distribution $p = (1-\epsilon)p_1 + \epsilon p_2^{(\text{span}(p_1)+1)} \in \mathcal{F}_h$ with $p_1 \in \mathcal{U}$, $p_2 \in \mathcal{M}_h$, and $1 > \epsilon > 0$. Let m denote base(p), and $m + M - 1$ denote span(p_1). Thus $|\text{support}(p_1)| = M$, and we have $M \geq 1$. Of course, we also have base(p_1) = base(p).

Consider any other distribution $q = (1-\epsilon')q_1 + \epsilon' q_2^{(\text{span}(q_1)+1)} \in \mathcal{F}_h$, where $q_1 \in \mathcal{U}$, $q_2 \in \mathcal{M}_h$, and $1 > \epsilon' > 0$. Let m' denote base(q) (which equals base(q_1)) and let $m' + M' - 1$ denote span(q_1), so $|\text{support}(q_1)| = M'$, and we have $M' \geq 1$.

It suffices to show that there is an ℓ_1 ball around p of sufficiently small radius, such that for all $\delta > 0$ we can find a uniform bound on the $(1-\delta)$-th percentile of all q in this ball.

If $m' > m$, then the ℓ_1 distance between p and q is at least $\frac{1-\epsilon}{M}$. Hence, whenever the ℓ_1 distance between p and q is strictly less than $\frac{1-\epsilon}{M}$ we must have $m' \leq m$. Thus we may assume that $m' \leq m$.

Suppose $m' + M' - 1 \geq m + \frac{2M}{1-\epsilon}$. Then $\frac{M}{M'} \leq \frac{1-\epsilon}{2}$, from which, because $\text{support}(p_1) \subseteq \text{support}(q_1)$, we can conclude that the ℓ_1 distance between q and p is at least $\frac{1-\epsilon}{2}$. Thus we may assume that $m' + M' - 1 < m + \frac{2M}{1-\epsilon}$.

Now, for any $i \geq 0$, we have

$$q(m' + M' + i) = \epsilon' q_2(i) \leq \epsilon' \frac{1}{i+1} \leq \frac{1}{i+1} \ .$$

Thus for any $K \geq 0$ we have

$$q(X > m + \frac{2M}{1-\epsilon} + K) \leq q(X > m' + M' + K) \leq \frac{h}{\log(K+1)}$$

by an argument similar to that in the preceding example, which gives the desired conclusion that no $p \in \mathcal{F}_h^\infty$ is deceptive, and hence that $\mathcal{F}_h^\infty$ is insurable. □

5. Necessary Condition for Insurability

Theorem 2 shows that lack of deceptive distributions in $\mathcal{P}$ is necessary for insurability of $\mathcal{P}^\infty$.

Theorem 2 If $\mathcal{P}^\infty$ is insurable, then no $p \in \mathcal{P}$ is deceptive.

Proof To keep notation simple, we will denote by p (or q) both a measure in $\mathcal{P}^\infty$ as well as the corresponding one dimensional marginal distribution, which is a member of $\mathcal{P}$. The context will clarify which of the two is meant. We prove the contrapositive of the theorem: if some $p \in \mathcal{P}$ is deceptive, then $\mathcal{P}^\infty$ is not insurable.

Pick $\alpha > 0$. Suppose $p \in \mathcal{P}$ is deceptive. Let Φ be any loss-domination scheme. Recall that Φ enters on p with probability 1, in the sense that the loss dominants set by Φ will eventually become finite with probability 1 under p. For all $n \geq 1$, let

$$R_n := \{x^n : \Phi(x^n) < \infty\}$$

be the set of sequences of length n on which Φ has entered and let $N \geq 1$ be a number such that

$$p(R_N) > 1 - \alpha/2. \tag{5}$$

Fix $0 < \eta < \frac{1}{2}(1 - \alpha - \frac{2}{N})(1 - 1/e^2)$. We prove that $\mathcal{P}^\infty$ is not insurable by finding, for each loss-domination scheme Φ, a probability distribution $q \in \mathcal{P}$ such that

$$q(\ \Phi \text{ goes bankrupt }) \geq \eta.$$

The basic idea is that because Φ has to enter with probability 1 under p, it would have been forced to set premiums that are too low for q.

For any sequence x^n, let $A(x^n)$ be the set of symbols that appear in it. Recall that the head of the distribution p, $H_{p,\gamma}$, was defined in Section 3.2 to be the set $\{y \in \mathcal{A}_p : y \leq 2F_p^{-1}(1 - \gamma/2)\}$, where $\mathcal{A}_p$ is the support of p. Further, define for all $\gamma > 0$

$$R_{p,\gamma,n} := \{x^n \in R_n : A(x^n) \subseteq H_{p,\gamma})\}.$$

Given $\delta > 0$, pick $\gamma_p(\delta)$ so small that

$$(1 - \gamma_p(\delta))^{N+1/\delta} \geq 1 - \alpha/2. \tag{6}$$

A word about this parameter $\gamma_p(\delta)$, since it may not be immediately apparent why this should be defined. The advantage of doing so is technical—we will be able to handle p (and q which will be chosen later) as though they were distributions with finite span.

Set[1] $\epsilon = \frac{1}{16(\ln 2)N^8}$. Applying Lemma 5 to distributions over length-N sequences induced by the distributions $p, q \in \mathcal{P}$ such that $\mathcal{J}(p,q) \leq \epsilon$, we have for each $\delta > 0$ that

$$q(X^N \in R_{p,\gamma_p(\delta),N}) \geq 1 - \alpha - \frac{2}{N} \ . \tag{7}$$

This implies that Φ has entered with probability (under q) at least $1 - \alpha - \frac{2}{N}$ for length N sequences. We will find a suitable $\delta > 0$ and a suitable q such that $\mathcal{J}(p,q) \leq \epsilon$, and such that the scheme Φ is bankrupted with probability $> \eta$.

Since p is deceptive, there exists $\delta' > 0$ such that

$$\sup_{\substack{q \in \mathcal{P}: \\ \mathcal{J}(p,q) < \epsilon}} F_q^{-1}(1-\delta') = \infty.$$

Define

$$\Delta_{p,\epsilon} = \{\delta' : \sup_{q: \mathcal{J}(p,q) < \epsilon} F_q^{-1}(1-\delta') = \infty\}.$$

In connection with this definition, note that if the δ' tails of distributions in the $\epsilon-$neighborhood of p are not bounded, neither are the δ'' tails for all $\delta'' < \delta'$. Furthermore if $\sup \Delta_{p,\epsilon} \geq 1/2$, we are done since for some $\delta \geq 1/2$, there is a q satisfying $\mathcal{J}(p,q) \leq \epsilon$ and

$$F_q^{-1}(1-\delta) \geq \max_{x^N \in R_{p,\gamma_p,N}} \Phi(x^N).$$

Therefore, conditioned on the event $\{X^N \in R_{p,\gamma_p(\delta),N}\}$, this q will be bankrupted with probability $\geq 1/2$. From (7) above we thus have

$$q(\ \Phi \text{ goes bankrupt }) \geq \frac{1 - \alpha - \frac{2}{N}}{2} > \eta.$$

If not, we have $\Delta_{p,\epsilon} < 1/2$. Pick δ and $r = 2\delta$ such that $\delta \in \Delta_{p,\epsilon}$, but $r \notin \Delta_{p,\epsilon}$. For convenience, let $M = \lceil \frac{1}{\delta} \rceil$. We consider now a set $\mathcal{S}$ of strings of lengths ranging from N to $N+M$ defined by the following properties:

- every string in $\mathcal{S}$ has its prefix of length N belonging to $R_{p,\gamma_p(\delta),N}$;
- every string of length k in $\mathcal{S}$, $N+1 \leq k \leq N+M$, has all its symbols at times $N+1$ through to k belonging to $H_{q',2r}$ for some $q' \in \mathcal{P}$ such that $\mathcal{J}(p,q') < \epsilon$.

To clarify this definition, we recall once again that $H_{q',2r}$ is the $\{y \in \mathcal{A}_{q'} \ : \ y \leq 2F_{q'}^{-1}(1-r)\}$, where $\mathcal{A}_{q'}$ denotes the support of q'. Note that since $r \notin \Delta_{p,\epsilon}$, $\mathcal{S}$ is finite. Again, we pick q satisfying $\mathcal{J}(p,q) \leq \epsilon$ and

$$F_q^{-1}(1-\delta) \geq \max_{\substack{x^k \in \mathcal{S} \\ N \leq k \leq N+M-1}} \Phi(x^k).$$

1. Please note that in the interest of simplicity, we have not attempted to provide the best scaling for ϵ or the tightest possible bounds in arguments below

Therefore if a symbol from $T_{q,\delta}$ follows any string in $\mathcal{S}$, the scheme goes bankrupt under q. There may be symbols in the complement of $T_{q,\delta}$ that also bankrupt the scheme if they follow a string in $\mathcal{S}$. Taking a different perspective, it could happen that no string in $\mathcal{S}$ contains symbols in $T_{q,\delta}$, or that strings $\mathcal{S}$ contain symbols from $T_{q,\delta}$. We consider both these variations below.

Let q_1 be the probability (under q) of all sequences in $R_{p,\gamma_p,N}$ under which the scheme Φ has not yet been bankrupted, and let q_2 be the probability (under q) of all sequences in $R_{p,\gamma_p,N}$ where Φ has already been bankrupted. Therefore $q_1 + q_2 = q(R_{p,\gamma_p,N})$.

Define a sequence in $\mathcal{S}$ to be a *survivor* if the loss-domination scheme Φ has not yet been bankrupted on this sequence under q. Thus, for instance, q_2 is the probability, under q, of survivor sequences of length N. To continue, we need to consider two cases:

Case (i). Strings in $\mathcal{S}$ do not contain symbols of $T_{q,\delta}$ (when $\sup_{q':\mathcal{J}(p,q')<\epsilon} 2F_{q'}^{-1}(1-r) \leq F_q^{-1}(1-\delta)$).

In this case, $H_{q,2r}$ is contained in the complement of $T_{q,\delta}$ and we show the probability under q with which Φ is bankrupted is bounded below by

$$q_2 + q_1 \left(\delta + (1-r)\delta + \ldots + (1-r)^M \delta\right).$$

Given that the sequence seen so far is a survivor sequence, one can classify the next symbol in one of four ways: (a) it is in $T_{q,\delta}$ (which automatically implies bankruptcy); (b) it is in the complement of $T_{q,\delta} \cup H_{q,2r}$ which we ignore; (c) it is in $H_{q,2r}$ and results in bankruptcy; (d) it is in $H_{q,2r}$ and does not result in bankruptcy. We ignore case (b) since we are only interested in a lower bound. In case (c) the contribution to the conditional probability of bankruptcy given a survivor sequence is 1, but we lower bound it for survivor sequences of length $N + l$ ($0 \leq l \leq M-1$) by the running sum $\left(\delta + (1-r)\delta + \ldots + (1-r)^{M-l-1}\delta\right)$. This has the effect that the lower bound looks as though symbols in $H_{q,2r}$ never contribute to bankruptcy (and hence always lead to survivor sequences).

Case (ii). Strings in $\mathcal{S}$ could contain symbols of $T_{q,\delta}$ (when $\sup_{q':\mathcal{J}(p,q')<\epsilon} 2F_{q'}^{-1}(1-r) > F_q^{-1}(1-\delta)$.)

We show here that the probability under q with which Φ is bankrupted is bounded below by

$$q_2 + q_1 \left(\delta + (1-\delta)\delta + \ldots + (1-\delta)^M \delta\right).$$

This can be seen, as in the preceding case, by classifying the next symbol following a survivor sequence into three types: (a) it is in $T_{q,\delta}$ (which implies bankruptcy); (b) it is the complement of $T_{q,\delta}$ and results in bankruptcy; (c) it is the complement of $T_{q,\delta}$ and does not result in bankruptcy. In case (b), the conditional probability of bankruptcy given a survivor sequence of length $N+l$ ($0 \leq l \leq M-1$) is 1, but, as before we lower bound the contribution to by the running sum $\left(\delta + (1-\delta)\delta + \ldots + (1-\delta)^{M-l-1}\delta\right)$. Similar to the prior case, this has the effect that the lower bound looks as though symbols in the complement of $T_{q,\delta}$ never contribute to bankruptcy (and hence always lead to survivor sequences).

However, we have $1 - \delta \geq 1 - r$, so once again in case (ii) the probability under q with which Φ is bankrupted is bounded below by

$$q_2 + q_1 \left(\delta + (1-r)\delta + \ldots + (1-r)^M \delta\right).$$

Thus we see that irrespective of which case is in force, under q the loss-domination scheme Φ is bankrupted with probability at least

$$\begin{aligned}
&q_2 + q_1\big(\delta + (1-r)\delta + \ldots + (1-r)^M\delta\big) \\
&= q_2 + q_1\left(\frac{1-(1-2\delta)^{\lceil 1/\delta\rceil}}{2}\right) \\
&\geq \frac{1}{2}q(R_{p,\gamma_p,N})\Big(1-(1-2\delta)^{\lceil 1/\delta\rceil}\Big) \\
&\geq \frac{1}{2}\left(1-\alpha-\frac{2}{N}\right)\left(1-\frac{1}{e^2}\right).
\end{aligned}$$

The theorem follows. □

6. Sufficient Condition for Insurability

When no $p \in \mathcal{P}$ is deceptive, for any $\eta > 0$ Theorem 3 constructs a loss-domination scheme that goes bankrupt with probability $\leq \eta$.

If no $p \in \mathcal{P}$ is deceptive, there is for each $p \in \mathcal{P}$ a number $\epsilon_p > 0$ such that, for every percentile $\delta > 0$, there is a uniform bound on the δ-percentile over the set of probability distributions in the neighborhood

$$\{p' \in \mathcal{P} : \mathcal{J}(p', p) < \epsilon_p\},.$$

We pick such an ϵ_p for each $p \in \mathcal{P}$ and call it the *reach* of p. For $p \in \mathcal{P}$, the set

$$B_p = \{p' \in \mathcal{P} : \mathcal{J}(p, p') < \epsilon_p\},$$

where ϵ_p is the reach of p, will play the role of the set of probability distributions in $\mathcal{P}$ for which it will be okay to eventually set loss dominants assuming p is in force.

To prove that $\mathcal{P}^\infty$ is insurable if no distribution among its one dimensional marginals $\mathcal{P}$ is deceptive, we will need to find a way to cover $\mathcal{P}$ with countably many sets of the form B_p above. Unfortunately, $\mathcal{J}(p, q)$ is not a metric, so it is not immediately clear how to go about doing this. On the other hand note that $\mathcal{J}(p', p) \leq |p - p'|_1/\ln 2$, where $|p - p'|_1$ denotes the ℓ_1 distance between p and p' (see Lemma 4 in the Appendix). Therefore, we can instead bootstrap off an understanding of the topology induced on $\mathcal{P}$ by the ℓ_1 metric.

6.1 Topology of $\mathcal{P}$ with ℓ_1 Metric

The topology induced on $\mathcal{P}$ by the ℓ_1 metric is Lindelöf, i.e. any covering of $\mathcal{P}$ with open sets in the ℓ_1 topology has a countable subcover. See *e.g.,* (Dugundji, 1970, Defn. 6.4) for definitions and properties of Lindelöf topological spaces.

We can show that $\mathcal{P}$ with the ℓ_1 topology is Lindelöf by appealing to the fact that the set of all probability distributions on $\mathbb{N}$ with the ℓ_1 topology, is second countable, i.e. that it has a countable basis. The set of all distributions on $\mathbb{N}$ along with ℓ_1 topology has a countable basis because it has a countable norm-dense set (consider the set of all probability distributions on $\mathbb{N}$ with finite support and with all probabilities being rational). Now, $\mathcal{P}$, as a topological subspace of a second countable topological space is also second countable (Dugundji, 1970, Theorem 6.2(2)). Finally, every second countable topological space is Lindelöf (Dugundji, 1970, Thm. 6.3), hence $\mathcal{P}$ is Lindelöf.

6.2 Sufficient Condition

We now have the machinery required to prove that if no $p \in \mathcal{P}$ is deceptive, then $\mathcal{P}^\infty$ is insurable, which is the other direction of Theorem 1, as stated next.

Theorem 3 If no $p \in \mathcal{P}$ is deceptive, then $\mathcal{P}^\infty$ is insurable.

Proof The proof is constructive. For any $0 < \eta < 1$, we obtain a loss-domination scheme Φ such that for all $p \in \mathcal{P}^\infty$, $p(\Phi \text{ goes bankrupt }) < \eta$.

For $p \in \mathcal{P}$, let

$$Q_p = \left\{ q : |p - q|_1 < \frac{{\epsilon_p}^2 (\ln 2)^2}{16} \right\},$$

where ϵ_p is the reach of p. We will call Q_p as the *zone* of p. The set Q_p is non-empty when $\epsilon_p > 0$.

For large enough n, the set of loss sequences of length n with empirical distribution in Q_p will ensure that the loss-domination scheme Φ to be proposed enters with probability 1 when p is in force. Note that if $\epsilon_p > 0$ is small enough then $Q_p \cap \mathcal{P} \subset B_p$—we will assume wolog that $\epsilon_p > 0$ is always taken so that $Q_p \cap \mathcal{P} \subset B_p$.

Since no $p \in \mathcal{P}$ is deceptive, none of the zones Q_p are empty and the space $\mathcal{P}$ of distributions can be covered by the sets $Q_p \cap \mathcal{P}$, namely

$$\mathcal{P} = \cup_{p \in \mathcal{P}} (Q_p \cap \mathcal{P}).$$

From Section 6.1, we know that $\mathcal{P}$ is Lindelöf under the ℓ_1 topology. Thus, there is a countable set $\tilde{\mathcal{P}} \subseteq \mathcal{P}$, such that $\mathcal{P}$ is covered by the collection of relatively open sets

$$\{Q_{\tilde{p}} \cap \mathcal{P} : \tilde{p} \in \tilde{\mathcal{P}}\}.$$

We let the above collection be denoted by $\mathcal{Q}_{\tilde{\mathcal{P}}}$. We will refer to $\tilde{\mathcal{P}}$ as the *quantization* of $\mathcal{P}$ and to elements of $\tilde{\mathcal{P}}$ as *centroids* of the quantization, borrowing from commonly used literature in classification.

We index the countable set of centroids, $\tilde{\mathcal{P}}$ (and reuse the index for the corresponding elements of $\mathcal{Q}_{\tilde{\mathcal{P}}}$) by $\iota : \tilde{\mathcal{P}} \to \mathbb{N}$.

We now describe the loss-domination scheme Φ having the property that for all $p \in \mathcal{P}^\infty$,

$$p(\Phi \text{ goes bankrupt }) < \eta.$$

Preliminaries. Consider a length-n sequence x^n on which Φ has not entered thus far. Let the empirical distribution of the sequence be q, and let

$$\mathcal{P}'_q := \{p' \in \tilde{\mathcal{P}} : q \in Q_{p'}\}$$

be the set of centroids in the quantization of $\mathcal{P}$ (elements of $\tilde{\mathcal{P}}$) which can potentially *capture* q. Note that q in general need not belong to $\tilde{\mathcal{P}}$ or $\mathcal{P}$.

If $\mathcal{P}'_q \neq \emptyset$, we will further refine the set of distributions that could capture q further to $\mathcal{P}_q \subset \mathcal{P}'_q$ as described below. Refining $\mathcal{P}'_q$ to $\mathcal{P}_q$ ensures that models in $\mathcal{P}'_q$ do not prematurely capture loss sequences.

Let p be the model in force, which remains unknown. The idea is that we want sequences generated by (unknown) p to be captured by those centroids of the quantization $\tilde{\mathcal{P}}$ that have p in their reach. We will require (8) below to ensure that the probability (under the unknown p) of all sequences that may

get captured by centroids $p' \in \mathcal{P}_q$ not having p in its reach remains small. In addition, we impose (9) as well to resolve a technical issue since q need not, in general, belong to $\mathcal{P}$.

For $p' \in \mathcal{P}'_q$, let the reach of p' be $\epsilon_{p'}$, and define

$$D_{p'} := \frac{\epsilon_{p'}{}^4 (\ln 2)^4}{256} .$$

In case the underlying distribution p happens to be out of the reach of p' (wrong capture), the quantity $D_{p'}$ will later lower bound the distance of the empirical q in question from the underlying p.

Specifically, we place p' in $\mathcal{P}_q$ if n satisfies

$$\exp\left(-nD_{p'}/18\right) \leq \frac{\eta}{2C(p')\iota(p')^2 n(n+1)}, \tag{8}$$

and

$$2F_q^{-1}(1 - \sqrt{D_{p'}}/6) \leq \log C(p'), \tag{9}$$

where $C(p')$ is

$$C(p') := 2^{2^{\left(\sup_{r \in B_{p'}} F_r^{-1}(1-\sqrt{D_{p'}}/6)\right)}}.$$

Note that $C(p')$ is finite since p' is not deceptive. Comparison with Lemma 7 will give a hint as to why the equations above look the way they do.

Description of Φ. For the sequence x^n with type q, if $\mathcal{P}_q = \emptyset$, the scheme does not enter yet. If $\mathcal{P}_q \neq \emptyset$, let p_q denote the distribution in $\mathcal{P}_q$ with the smallest index.

All sequences with prefix x^n (namely sequences obtained by concatenating x^n with by any other sequence of symbols) are then said to be *trapped* by p_q—namely, loss dominants will be based on p_q. The loss dominant assigned for a length-m sequence trapped by p_q is

$$2g_{p_q}\left(\frac{\eta}{4n(n+1)}\right) := 2 \sup_{r \in B_{p_q}} F_r^{-1}\left(1 - \frac{\eta}{4n(n+1)}\right).$$

Φ enters with probability 1. First, we verify that the scheme enters with probability 1, no matter what distribution $p \in \mathcal{P}$ is in force. Every distribution $p \in \mathcal{P}$ is contained in at least one of the elements of the cover $\mathcal{Q}_{\tilde{\mathcal{P}}}$.

Recall the enumeration of $\tilde{\mathcal{P}}$. Let p' be centroid with the smallest index among all centroids in $\tilde{\mathcal{P}}$ whose zones contain p. Let Q be the zone of p'. There is thus some $\gamma > 0$ such that the neighborhood around p given by

$$I(p, \gamma) := \{q : |p - q|_1 < \gamma\}$$

satisfies $I(p, \gamma) \subseteq Q$. Note in particular that p is in the reach of p'.

With probability 1, sequences generated by p will have their empirical distribution within $I(p, \gamma)$. For the proof of this well known result in the countably infinite alphabet case, see Chung (1961) or for an alternate approach, see Lemma 7. Next (8) will hold for all sequences whose empirical distributions that fall in $I(p, \gamma)$ whose length n is large enough—since $C(p')$ and $\iota(p')$ do not change with n, the right hand side diminishes to zero polynomially with n while the left hand side diminishes exponentially to zero. Thus we conclude (8) will be satisfied with probability 1.

Next, (9) will also hold almost surely, for if q is the empirical probability of sequences generated by p, then (with a little abuse of notation)

$$F_q^{-1}(1-\sqrt{D_{p'}}/6) \to F_p^{-1}(1-\sqrt{D_{p'}}/6)$$

with probability 1. Note that the quantity on the left is actually a random variable that is sequence dependent (since q is the empirical distribution of the sequence). Furthermore, we also have

$$\begin{aligned} 2F_p^{-1}(1-\sqrt{D_{p'}}/6) &\leq 2\left(\sup_{r\in B_{p'}} F_r^{-1}(1-\sqrt{D_{p'}}/6)\right) \\ &= \log C(p'), \end{aligned}$$

where the first inequality follows since p is in the reach of p'.

Thus the scheme enters with probability 1 no matter which $p \in \mathcal{P}$ is in force.

Probability of bankruptcy $\leq \eta$. We now analyze the scheme. Consider any $p \in \mathcal{P}$. Among sequences on which Φ has entered, we will distinguish between those that are in *good* traps and those in *bad* traps. If a sequence x^n is trapped by p' such that $p \in B_{p'}$, p' is a good trap. Conversely, if $p \notin B_{p'}$, p' is a bad trap.

(Good traps) Suppose a length-n sequence x^n is in a good trap, namely, it is trapped by a distribution p' such that $p \in B_{p'}$. Recall that the loss dominant assigned is

$$2g_{p'}\left(\frac{\eta}{4n(n+1)}\right) \geq 2F_p^{-1}\left(1-\frac{\eta}{4n(n+1)}\right),$$

where the inequality follows because p' is not deceptive, and p is within the reach of p'. Therefore from (4), given any sequence in a good trap the scheme is bankrupted with conditional probability at most $\delta' = \eta/2n(n+1)$ in the next step. Therefore, summing over all n, sequences in good traps contribute at most $\eta/2$ to the probability of bankruptcy.

(Bad traps) We will show that the probability with which sequences generated by p fall into bad traps $\leq \eta/2$. Pessimistically, the conditional probability of bankruptcy in the very next step given a sequence falls into a bad trap is going to be bounded above by 1. Thus the contribution to bankruptcy by sequences in bad traps is at most $\eta/2$.

Let q be any length-n empirical distribution trapped by $\tilde{p}$ with reach $\tilde{\epsilon}$ such that $p \notin B_{\tilde{p}}$.

If p is "far" from $\tilde{p}$ (because p is not in $\tilde{p}$'s reach), namely

$$\mathcal{J}(\tilde{p}, p) \geq \tilde{\epsilon},$$

but q is "close" to $\tilde{p}$ (because q has to be in $\tilde{p}$'s zone to be captured by it), namely

$$|\tilde{p} - q|_1 < \frac{\tilde{\epsilon}^2(\ln 2)^2}{16},$$

then we would like q to be far from p. That is exactly what we obtain from the triangle-inequality like Lemma 6, namely that

$$\mathcal{J}(p, q) \geq \frac{\tilde{\epsilon}^2 \ln 2}{16}$$

and hence, for all q trapped by $\tilde{p}$ that

$$|p-q|_1^2 \geq \mathcal{J}^2(p,q)(\ln 2)^2 \geq \frac{\tilde{\epsilon}^4(\ln 2)^4}{256} = D_{\tilde{p}}^2.$$

We need not be concerned that the right side above depends on $\tilde{p}$, and there may be actually no way to lower bound the rhs as a function of just p. Rather, we take care of this issue by setting the entry point appropriately via (8).

Thus, for $p \in \mathcal{P}^\infty$, the probability length-n sequences with empirical distribution q is trapped by a bad $\tilde{p}$ is, using (8) and (9)

$$\begin{aligned}
&\leq p\left(|q-p|^2 \geq D_{\tilde{p}} \text{ and } 2F_q^{-1}(1-\frac{\sqrt{D_{\tilde{p}}}}{6}) \leq \log C(\tilde{p})\right)\\
&\overset{(a)}{\leq} (C(\tilde{p})-2)\exp\left(-\frac{nD_{\tilde{p}}}{18}\right)\\
&\overset{(b)}{\leq} \frac{\eta(C(\tilde{p})-2)}{2C(\tilde{p})\iota(\tilde{p})^2 n(n+1)}\\
&\leq \frac{\eta}{2\iota(\tilde{p})^2 n(n+1)},
\end{aligned}$$

where the inequality (a) follows from Lemma 7 and (b) from (8). Therefore, the probability of sequences falling into bad traps

$$\leq \sum_{n\geq 1}\sum_{\tilde{p}\in\tilde{\mathcal{P}}} \frac{\eta}{2\iota(\tilde{p})^2 n(n+1)} \leq \eta/2$$

since $\sum_{\tilde{p}\in\tilde{\mathcal{P}}} \frac{1}{\iota(\tilde{p})^2} \leq \sum_{n\geq 1}\frac{1}{n(n+1)} = 1$. The theorem follows. □

7. Concluding Remarks

7.1 Observations

We make a few observations about insurability that, while evident from the proofs and approaches we have taken, are interesting in themselves and worth highlighting.

Finite unions of insurable classes. The first is relative simple—finite unions of insurable model classes are insurable in themselves. If $\mathcal{P}_1,\ldots,\mathcal{P}_m$ are m insurable model collections, then $\cup_{i=1}^m \mathcal{P}_i$ is insurable.

Countable unions of insurable classes. The union of countably infinitely many classes of models each of which is insurable need not be insurable. As we have seen, the collection of monotone distributions with entropy $\leq h$, $\mathcal{M}_h$, is insurable for all $h \in \mathbb{N}$. However, the collection $\mathcal{M}_*^\infty = \cup_{h\in\mathbb{N}}\mathcal{M}_h$ is not insurable.

Countable unions of tight sets. Note that from our arguments while proving the sufficiency criterion, it follows that every insurable model class is a countable union of tight sets. The converse is not however true. Note that $\mathcal{M}_h$ is a tight set for any $h > 0$, yet $\mathcal{M}_*^\infty = \cup_{h\in\mathbb{N}}\mathcal{M}_h$ is not insurable.

7.2 General Remarks

The loss-domination problem formulated and solved in this paper appears to be of natural interest. However, there are several features of the insurance problem formulated here that might appear troubling even to the casual reader. In practice an insured party entering into an insurance contract would expect some stability in the premiums that are expected to be paid. A natural direction for further research is therefore to study how the notion of insurability of a model class changes when one imposes restrictions on how much the premium set by the insurer can vary from period to period. Another obvious shortcoming of the formulation of the insurance problem studied here is the assumption that the insured will accept any contract issued by the insurer. Since the insured in our model represents an aggregate of individual insured parties, a natural direction to make the framework more realistic would be to think of the insured parties as being of different *types*. This would in effect make the total realized premium from the insured (the aggregate of the insured parties) and the distribution of the realized loss in each period a function of the size of the premium per insured party set by the insurer in that period. Characterizing which model classes are insurable when the realized premium and the realized loss are functions of a set premium per insured party would be of considerable interest.

Both for the loss-domination problem and for the insurance problem, working with model classes for the loss sequence that allow for dependencies in the loss from period to period, for instance Markovian dependencies, would be another interesting direction for further research. Considering models with multiple, possibly competing insurers, as well as considering an insurer operating in multiple markets, where losses in one market can be offset by gains in another, also seem to be useful directions to investigate.

Acknowledgments

We are very grateful to anonymous reviewers whose insightful comments has made this paper much better. We thank C. Nair (Chinese Univ of Hong Kong) and K. Viswanathan (HP Labs) for helpful discussions. N. Santhanam was supported by NSF Grants CCF-1065632, CCF-1018984 and EECS-1029081. V. Anantharam was supported by the ARO MURI grant W911NF- 08-1-0233, Tools for the Analysis and Design of Complex Multi-Scale Networks, the NSF grant CNS-0910702 and EECS-1343398, the NSF Science & Technology Center grant CCF-0939370, Science of Information, Marvell Semiconductor Inc., and the U.C. Discovery program.

Appendix

Lemma 4 Let p and q be probability distributions on $\mathbb{N}$. Then

$$\frac{1}{4\ln 2}|p-q|_1^2 \leq \mathcal{J}(p,q) \leq \frac{1}{\ln 2}|p-q|_1 \ .$$

If, in addition, r is a probability distribution on $\mathbb{N}$, then

$$\mathcal{J}(p,q)+\mathcal{J}(q,r) \geq \mathcal{J}^2(p,r)\frac{\ln 2}{8}.$$

Proof The lower bound in the first statement follows from Pinsker's inequality on KL divergences,

$$D\left(p||\frac{p+q}{2}\right) \geq \frac{1}{2\ln 2}\frac{1}{4}|p-q|_1^2$$

and similarly for $D\big(q||\frac{p+q}{2}\big)$. See *e.g.,* Cover and Thomas (1991) for more details about Pinsker's inequality.

Since $\ln(1+z) \le z$ for all $z \ge 0$, the upper bound in the first statement follows as below:

$$\begin{aligned}\mathcal{J}(p,q)\ln 2 &\le \sum_{x:p(x)\ge q(x)} p(x)\left(\frac{p(x)-q(x)}{p(x)+q(x)}\right) + \sum_{x':q(x')\ge p(x')} q(x')\left(\frac{q(x')-p(x')}{p(x')+q(x')}\right)\\ &\le |p-q|_1.\end{aligned}$$

To prove the triangle-like inequality, note that

$$\begin{aligned}\mathcal{J}(p,q)+\mathcal{J}(q,r) &\ge \frac{1}{4\ln 2}\big(|p-q|_1^2+|q-r|_1^2\big)\\ &\ge \frac{1}{8\ln 2}(|p-q|_1+|q-r|_1)^2\\ &\ge \frac{1}{8\ln 2}(|p-r|_1)^2\\ &\ge \frac{\ln 2}{8}\mathcal{J}(p,r)^2,\end{aligned}$$

where the last inequality follows from the upper bound on $\mathcal{J}(p,r)$ already proved. □

Lemma 5 Let p and q be probability distributions on a countable set $\mathcal{A}$ with $\mathcal{J}(p,q) \le \epsilon$. Let p^N and q^N be distributions over $\mathcal{A}^N$ obtained by *i.i.d.* sampling from p and q respectively (the distribution induced by the product measure). For any $R_N \subset \mathcal{A}^N$ and $\alpha > 0$, if $p^N(R_N) \ge 1-\alpha$, then

$$q^N(R_N) \ge 1-\alpha-2N^3\sqrt{4\epsilon\ln 2}-\frac{1}{N}.$$

Proof Let

$$\mathcal{B}_1 = \left\{i \in \mathcal{A} : q(i) \le p(i)\left(1-\frac{1}{N^2}\right)\right\},$$

and let

$$\mathcal{B}_2 = \left\{i \in \mathcal{A} : p(i) \le q(i)\left(1-\frac{1}{N^2}\right)\right\},$$

If $\mathcal{J}(p,q) \le \epsilon$, then we have

$$\sqrt{\epsilon} \ge \sqrt{\mathcal{J}(p,q)} \ge \frac{|p-q|_1}{\sqrt{4\ln 2}}.$$

It can then be easily seen that

$$p(\mathcal{B}_1\cup\mathcal{B}_2) \le 2N^2\sqrt{4\epsilon\ln 2} \text{ and } q(\mathcal{B}_1\cup\mathcal{B}_2) \le 2N^2\sqrt{4\epsilon\ln 2} \tag{10}$$

because

$$|p-q|_1 \ge \sum_{x\in\mathcal{B}_1}(p(x)-q(x)) \ge \frac{p(\mathcal{B}_1)}{N^2} \ge \frac{q(\mathcal{B}_1)}{N^2}$$

and similarly

$$N^2|p-q|_1 \ge q(\mathcal{B}_2) \ge p(\mathcal{B}_2).$$

Let $S = \mathcal{A} - \mathcal{B}_1 \cup \mathcal{B}_2$. We have for all $x \in S$,

$$q(x) \geq p(x)\left(1 - \frac{1}{N^2}\right). \tag{11}$$

and from (10) we have $p(S) \geq 1 - 2N^2\sqrt{4\epsilon \ln 2}$. Now, we focus on the set $S_N \subset \mathcal{A}^N$ containing all length-N strings of symbols from S. Clearly

$$p(S_N) \geq 1 - 2N^3\sqrt{4\epsilon \ln 2}.$$

Thus we have

$$p(R_N \cap S_N) \geq 1 - 2N^3\sqrt{4\epsilon \ln 2} - \alpha.$$

From (11), for all $x^N \in S_N$,

$$q(x^N) \geq p(x^N)\left(1 - \frac{1}{N^2}\right)^N \geq p(x^N)\left(1 - \frac{1}{N}\right).$$

Therefore,

$$q(R_N) \geq q(R_N \cap S_N) \geq (1 - 2N^3\sqrt{4\epsilon \ln 2} - \alpha)\left(1 - \frac{1}{N}\right) \geq 1 - \alpha - 2N^3\sqrt{4\epsilon \ln 2} - \frac{1}{N}. \qquad \square$$

Lemma 6 Let $\epsilon_0 > 0$. If

$$|p_0 - q|_1 \leq \frac{\epsilon_0^2 (\ln 2)^2}{16} \ ,$$

then for all $p \in \mathcal{P}$ with $\mathcal{J}(p, p_0) \geq \epsilon_0$, we have

$$\mathcal{J}(p, q) \geq \frac{\epsilon_0^2 \ln 2}{16}.$$

Proof Since

$$|p_0 - q|_1 \leq \frac{\epsilon_0^2 (\ln 2)^2}{16},$$

Lemma 4 implies that

$$\mathcal{J}(p_0, q) \leq \frac{\epsilon_0^2 \ln 2}{16}.$$

Further, Lemma 4 then implies that

$$\mathcal{J}(p, q) + \frac{\epsilon_0^2 \ln 2}{16} \ \geq \ \mathcal{J}(p, q) + \mathcal{J}(p_0, q) \ \geq \ \frac{\mathcal{J}^2(p, p_0) \ln 2}{8} \ \geq \ \frac{\epsilon_0^2 \ln 2}{8},$$

where the last inequality follows since $\mathcal{J}(p, p_0) \geq \epsilon_0$. $\square$

Lemma 7 Let p be any probability distribution on $\mathbb{N}$. Let $\delta > 0$ and let $k \geq 2$ be an integer. Let X_1^n be a sequence generated *i.i.d.* with marginals p and let $q(X^n)$ be the empirical distribution of X_1^n. Then

$$p\big(|q(X^n) - p| > \delta \text{ and } 2F_q^{-1}(1 - \delta/6) \leq k\big) \leq (2^k - 2) \exp\left(-\frac{n\delta^2}{18}\right).$$

Remark There is a lemma that looks somewhat similar in Ho and Yeung (2010). The difference from Ho and Yeung (2010) is that the right side of the inequality above does *not* depend on p, and this property is crucial for its use here. □

Proof The starting point is the following result. Suppose p' is a probability distribution on $\mathbb{N}$ with finite support of size L. Then from Weissman et al. (2005), if we consider length n sequences,

$$p'(|q(X^n) - p'|_1 \leq t) \geq 1 - (2^L - 2)\exp\left(-\frac{nt^2}{2}\right). \tag{12}$$

Since $k \geq 2$, consider the distributions p' and q' with support $A = \{1, \ldots, k-1\} \cup \{-1\}$, obtained as

$$p'(i) = \begin{cases} p(i) & 1 \leq i < k \\ \sum_{j=k}^{\infty} p(j) & i = -1, \end{cases}$$

and similarly for q'.

From (12),

$$p'(|p' - q'|_1 > \delta/3) \leq (2^k - 2)\exp\left(-\frac{n\delta^2}{18}\right).$$

We will see that all sequences generated by p with empirical distributions q satisfying

$$|p - q|_1 > \delta \text{ and } 2F_q^{-1}(1 - \delta/6) \leq k$$

are now mapped into sequences generated by p' with empirical q' satisfying

$$|p' - q'|_1 > \delta/3 \text{ and } q'(-1) \leq \delta/3. \tag{13}$$

Thus, we will have

$$\begin{aligned} &p(|q(X^n) - p|_1 > \delta \text{ and } 2F_q^{-1}(1 - \delta/6) \leq k) \\ &\leq p'(|p' - q'|_1 > \delta/3 \text{ and } q'(-1) \leq \delta/3) \\ &\leq (2^k - 2)\exp\left(-\frac{n\delta^2}{18}\right). \end{aligned}$$

Finally we observe (13) as in Ho and Yeung (2010)

$$\begin{aligned} |p - q|_1 - \sum_{l=1}^{k-1} |p(l) - q(l)| \\ &\leq \sum_{j=k}^{\infty} (p(j) - q(j)) + 2\sum_{j=k}^{\infty} q(j) \\ &\leq |p'(-1) - q'(-1)| + 2\delta/3, \end{aligned}$$

where the last inequality above follows from (4). Since $p(l) = p'(l)$ and $q(l) = q'(l)$ for all $l = 1, \ldots, k-1$, we have

$$|p' - q'|_1 \geq |p - q|_1 - 2\delta/3.$$

If $|p - q|_1 \geq \delta$ in addition, $|p' - q'|_1 \geq \delta/3$. □

References

M. Asadi, R. Paravi, and N. Santhanam. Stationary and transition probabilities in slow mixing, long memory markov processes. *IEEE Transactions on Information Theory*, September 2014.

S. Asmussen and H. Albrecher. *Ruin Probabilities*. World Scientific Publishing Company, 2nd edition, 2010.

P. Billingsley. *Probability and Measure*. Wiley Interscience, 1995.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning and Games*. Cambridge University Press, 2006.

K.L. Chung. A note on the ergodic theorem of information theory. *Annals of Mathematical Statistics*, 32:612—614, 1961.

T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley and sons., 1991.

H. Cramer. Historical Review of Filip Lundberg's Work on Risk Theory. *Skandinavisk Aktuarietidskrift (Suppl.)*, 52:6–12, 1969. Reprinted in The Collected Works of Harald Cramér edited by Anders Martin-Löf, 2 volumes Springer 1994.

J. Dugundji. *Topology*. Allyn and Bacon Inc., Boston, 1970.

K. Englund and A. Martin-Löf. *Statisticians of the Centuries*, chapter Ernst Filip Oskar Lundberg, pages 308–311. New York: Springer, 2001.

B. Fittingoff. Universal methods of coding for the case of unknown statistics. In *Proceedings of the 5th Symposium on Information Theory*, pages 129—135. Moscow-Gorky, 1972.

P. Grunwald. *The Minimum Description Length Principle*. MIT Press, 2007.

S. Ho and R. Yeung. On information divergence measures and joint typicality. *IEEE Transactions on Information Theory*, 56(12):58935905, 2010.

J.C. Kieffer. A unified approach to weak universal source coding. *IEEE Transactions on Information Theory*, 24(6):674—682, November 1978.

D. McAllester. A pac-bayesian tutorial with a dropout bound. Available from arxiv doc:id 1307.2118, 2013.

J. Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information Theory*, 30(4):629—636, July 1984.

B. Ryabko. Compression based methods for non-parametric online prediction, regression, classification and density estimation. *Festschrift in Honor of Jorma Rissanen on the Occasion of his 75th Birthday*, pages 271–288, 2008.

N. Santhanam and V. Anantharam. Agnostic insurance tasks and their relation to compression. In *International Conference on Signal Processing and Communications (SPCOM)*, 2012.

Y.M. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):3—17, 1987.

T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. Weinberger. Universal discrete denoising: known channel. *IEEE Transactions on Information Theory*, 51(1):5–28, 2005. See also HP Labs Tech Report HPL-2003-29, Feb 2003.

Journal of Machine Learning Research 16 (2015) 2357-2375 Submitted 7/14; Revised 12/14; Published 10/15

Achievability of Asymptotic Minimax Regret by Horizon-Dependent and Horizon-Independent Strategies

Kazuho Watanabe WKAZUHO@CS.TUT.AC.JP
Department of Computer Science and Engineering
Toyohashi University of Technology
1-1, Hibarigaoka, Tempaku-cho, Toyohashi, 441-8580, Japan

Teemu Roos TEEMU.ROOS@CS.HELSINKI.FI
Helsinki Institute for Information Technology HIIT
Department of Computer Science, University of Helsinki
PO Box 68, FI-00014, Finland

Editor: Manfred Warmuth

Abstract

The normalized maximum likelihood distribution achieves minimax coding (log-loss) regret given a fixed sample size, or horizon, n. It generally requires that n be known in advance. Furthermore, extracting the sequential predictions from the normalized maximum likelihood distribution is computationally infeasible for most statistical models. Several computationally feasible alternative strategies have been devised. We characterize the achievability of asymptotic minimaxity by horizon-dependent and horizon-independent strategies. We prove that no horizon-independent strategy can be asymptotically minimax in the multinomial case. A weaker result is given in the general case subject to a condition on the horizon-dependence of the normalized maximum likelihood. Motivated by these negative results, we demonstrate that an easily implementable Bayes mixture based on a conjugate Dirichlet prior with a simple dependency on n achieves asymptotic minimaxity for all sequences, simplifying earlier similar proposals. Our numerical experiments for the Bernoulli model demonstrate improved finite-sample performance by a number of novel horizon-dependent and horizon-independent algorithms.

Keywords: on-line learning, prediction of individual sequences, normalized maximum likelihood, asymptotic minimax regret, Bayes mixture

1. Introduction

The normalized maximum likelihood (NML) distribution is derived as the optimal solution to the minimax problem that seeks to minimize the worst-case coding (log-loss) regret with fixed sample size n (Shtarkov, 1987). In this problem, any probability distribution can be converted into a sequential prediction strategy for predicting each symbol given an observed initial sequence, and *vice versa*. A minimax solution yields predictions that have the least possible regret, i.e., excess loss compared to the best model within a model class.

The important multinomial model, where each symbol takes one of $m > 1$ possible values, has a long history in the extensive literature on universal prediction of individual sequences especially in the Bernoulli case, $m = 2$ (see e.g. Laplace, 1795/1951; Krichevsky and Trofimov, 1981; Freund, 1996; Krichevsky, 1998; Merhav and Feder, 1998; Cesa-Bianchi

and Lugosi, 2001). A linear time algorithm for computing the NML probability of any individual sequence of full length n was given by Kontkanen and Myllymäki (2007). However, this still leaves two practical problems. First, given a distribution over sequences of length n, obtaining the marginal and conditional probabilities needed for predicting symbols before the last one requires evaluation of exponentially many terms. Second, the total length of the sequence, or the *horizon*, is not necessarily known in advance in so called online scenarios (see e.g. Freund, 1996; Azoury and Warmuth, 2001; Cesa-Bianchi and Lugosi, 2001). The predictions of the first $\tilde{n}$ symbols under the NML distribution depend on the horizon n in many models, including the multinomial. In fact, Bartlett et al. (2013) showed that NML is horizon-dependent in this sense in all one-dimensional exponential families with three exceptions (Gaussian, Gamma, and Tweedy). When this is the case, NML cannot be applied, and consequently, minimax optimality cannot be achieved without horizon-dependence. Similarly, in a somewhat different adversarial setting, Luo and Schapire (2014) show a negative result that applies to loss functions bounded within the interval $[0, 1]$.

Several easily implementable nearly minimax optimal strategies have been proposed (see Shtarkov, 1987; Xie and Barron, 2000; Takeuchi and Barron, 1997; Takimoto and Warmuth, 2000; Kotłowski and Grünwald, 2011; Grünwald, 2007, and references therein). For asymptotic minimax strategies, the worst-case total log-loss converges to that of the NML distribution as the sample size tends to infinity. This is not equivalent to the weaker condition that the average regret per symbol converges to zero. It is known, for instance, that neither the Laplace plus-one-rule that assigns probability $(k+1)/(n+m)$ to a symbol that has appeared k times in the first n observations, nor the Krichevsky-Trofimov plus-one-half-rule, $(k+1/2)/(n+m/2)$, which is also the Bayes procedure under the Jeffreys prior, are asymptotically minimax optimal over the full range of possible sequences (see Xie and Barron, 2000). Xie and Barron (2000) showed that a Bayes procedure defined by a modified Jeffreys prior, wherein additional mass is assigned to the boundaries of the parameter space, achieves asymptotic minimax optimality. Takeuchi and Barron (1997) studied an alternative technique for a more general model class. Both these strategies are horizon-dependent. An important open problem has been to determine whether a horizon-independent asymptotic minimax strategy for the multinomial case exists.

We investigate achievability of asymptotic minimaxity by horizon-dependent and horizon-independent strategies. Our main theorem (Theorem 2) answers the above open problem in the negative: no horizon-independent strategy can be asymptotic minimax for multinomial models. We give a weaker result that applies more generally under a condition on the horizon-dependence of NML. On the other hand, we show that an easily implementable horizon-dependent Bayes procedure defined by a simpler prior than the modified Jeffreys prior by Xie and Barron (2000) achieves asymptotic minimaxity. The proposed procedure assigns probability $(k+\alpha_n)/(n+m\alpha_n)$ to any outcome that has appeared k times in a sequence of length n, where m is the alphabet size and $\alpha_n = 1/2 - \ln 2/(2\ln n)$ is a prior mass assigned to each outcome. We also investigate the behavior of a generalization of the last-step minimax algorithm, which we call the k-last-step minimax algorithm and which is horizon-independent. Our numerical experiments (Section 5) demonstrate superior finite-sample performance by the proposed horizon-dependent and horizon-independent algorithms compared to existing approximate minimax algorithms.

2. Preliminaries

Consider a sequence $x^n = (x_1, \cdots, x_n)$ and a parametric model

$$p(x^n|\theta) = \prod_{i=1}^{n} p(x_i|\theta),$$

where $\theta = (\theta_1, \cdots, \theta_d)$ is a d-dimensional parameter. We focus on the case where each x_i is one of a finite alphabet of symbols and the maximum likelihood estimator

$$\hat{\theta}(x^n) = \operatorname*{argmax}_{\theta} \ln p(x^n|\theta)$$

can be computed.

The optimal solution to the minimax problem,

$$\min_{\overline{p}} \max_{x^n} \ln \frac{p(x^n|\hat{\theta}(x^n))}{\overline{p}(x^n)},$$

assuming that the solution exists, is given by

$$p^{(n)}_{\mathrm{NML}}(x^n) = \frac{p(x^n|\hat{\theta}(x^n))}{C_n}, \tag{1}$$

where $C_n = \sum_{x^n} p(x^n|\hat{\theta}(x^n))$ and is called the normalized maximum likelihood (NML) distribution (Shtarkov, 1987). For model classes where the above problem has no solution and the normalizing term C_n diverges, it may be possible to reach a solution by conditioning on some number of initial observations (see Liang and Barron, 2004; Grünwald, 2007). The regret of the NML distribution is equal to the minimax value $\ln C_n$ for all x^n. We mention that in addition to coding and prediction, the code length $-\ln p^{(n)}_{\mathrm{NML}}(x^n)$ can be used as a model selection criterion according to the minimum description length (MDL) principle (Rissanen, 1996); (see also Grünwald, 2007; Silander et al., 2010, and references therein).

In cases where the minimax optimal NML distribution cannot be applied (for reasons mentioned above), it can be approximated by another strategy, i.e., a sequence of distributions $(g^{(n)})_{n\in\mathbb{N}}$. A strategy is said to be *horizon-independent* if for all $1 \leq \tilde{n} < n$, the distribution $g^{(\tilde{n})}$ matches with the marginal distribution of $x^{\tilde{n}}$ obtained from $g^{(n)}$ by summing over all length n sequences that are obtained by concatenating $x^{\tilde{n}}$ with a continuation $x^n_{\tilde{n}+1} = (x_{\tilde{n}+1}, \cdots, x_n)$:

$$g^{(\tilde{n})}(x^{\tilde{n}}) = \sum_{x^n_{\tilde{n}+1}} g^{(n)}(x^n). \tag{2}$$

For horizon-independent strategies, we omit the horizon n in the notation and write $g(x^n) = g^{(n)}(x^n)$. This also implies that the ratio $g(x^n_{\tilde{n}+1}|x^{\tilde{n}}) = g(x^n)/g(x^{\tilde{n}})$ is a valid conditional probability distribution over the continuations $x^n_{\tilde{n}+1}$ assuming that $g(x^{\tilde{n}}) > 0$.[1]

1. Note that even if a strategy is based on *assuming* a fixed horizon (or an increasing sequence or horizons like in the so called doubling-trick, see Cesa-Bianchi et al., 1997), as long as the assumed horizon is independent of the true horizon, the strategy is horizon-independent.

A property of interest is *asymptotic* minimax optimality of g, which is defined by

$$\max_{x^n} \ln \frac{p(x^n|\hat{\theta}(x^n))}{g(x^n)} \leq \ln C_n + o(1), \tag{3}$$

where $o(1)$ is a term converging to zero as $n \to \infty$.

Hereafter, we focus mainly on the multinomial model with $x \in \{1, 2, \cdots, m\}$,

$$p(x|\theta) = \theta_x, \quad \sum_{j=1}^{m} \theta_j = 1, \tag{4}$$

extended to sequences by the i.i.d. assumption. The corresponding conjugate prior is the Dirichlet distribution. In the symmetric case where each outcome $x \in \{1, \ldots, m\}$ is treated equally, it takes the form

$$q(\theta|\alpha) = \frac{\Gamma(m\alpha)}{\Gamma(\alpha)^m} \prod_{j=1}^{m} \theta_j^{\alpha-1},$$

where $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ is the gamma function and $\alpha > 0$ is a hyperparameter. Probabilities of sequences under Bayes mixtures with Dirichlet priors can be obtained from

$$p_{B,\alpha}(x^n) = \int \prod_{i=1}^{n} p(x_i|\theta)q(\theta|\alpha)d\theta = \frac{\Gamma(m\alpha)}{\Gamma(\alpha)^m} \frac{\prod_{j=1}^{m} \Gamma(n_j + \alpha)}{\Gamma(n + m\alpha)}, \tag{5}$$

where n_j is the number of js in x^n. The Bayes mixture is horizon-dependent if α depends on n and horizon-independent otherwise.

The minimax regret is asymptotically given by Xie and Barron (2000),

$$\ln C_n = \frac{m-1}{2} \ln \frac{n}{2\pi} + \ln \frac{\Gamma(1/2)^m}{\Gamma(m/2)} + o(1). \tag{6}$$

3. (Un)achievability of Asymptotic Minimax Regret

We now give our main result, Theorem 2, showing that no horizon-independent asymptotic minimax strategy for the multinomial case exists. In the proof, we use the following lemma. The proof of the lemma is given in Appendix A.

Lemma 1 *Let*

$$f(x) = \ln \Gamma\left(x + \frac{1}{2}\right) - x \ln x + x - \frac{1}{2} \ln 2\pi,$$

for $x > 0$ *and* $f(0) = -\frac{\ln 2}{2}$. *Then for* $x \geq 0$,

$$-\frac{\ln 2}{2} \leq f(x) < 0 \tag{7}$$

and $\lim_{x\to\infty} f(x) = 0$.

Theorem 2 *For the multinomial model in (4), no horizon-independent strategy is asymptotic minimax.*

Proof Let g be an arbitrary horizon-independent strategy satisfying (2). First, by the properties of the Gamma function, we have $\ln\Gamma(n+\frac{m}{2}) = \ln\Gamma(n+\frac{1}{2}) + \frac{(m-1)}{2}\ln n + o(1)$. Applying this to (5) in the case of the Jeffreys mixture $p_{B,1/2}$ yields

$$\ln p_{B,1/2}(x^n) = \ln\frac{\Gamma(m/2)}{\Gamma(1/2)^m} + \sum_{j=1}^{m}\Big\{\ln\Gamma(n_j+1/2)\Big\} - \ln\Gamma(n+1/2) - \frac{m-1}{2}\ln n + o(1). \quad (8)$$

We thus have

$$\begin{aligned}\ln\frac{p^{(n)}_{\mathrm{NML}}(x^n)}{p_{B,1/2}(x^n)} &= \sum_{j=1}^{m}\left\{-\ln\Gamma\left(n_j+1/2\right) + n_j\ln n_j - n_j + \frac{1}{2}\ln 2\pi\right\} \\ &\quad + \ln\Gamma(n+1/2) - n\ln n + n - \frac{1}{2}\ln 2\pi + o(1) \\ &= -\sum_{j=1}^{m} f(n_j) + f(n) + o(1). \end{aligned} \quad (9)$$

By Lemma 1, for the sequence of all js (for any $j \in \{1,2,\cdots,m\}$),

$$\ln\frac{p^{(n)}_{\mathrm{NML}}(x^n)}{p_{B,1/2}(x^n)} \to \frac{m-1}{2}\ln 2 \quad (n\to\infty),$$

which means that the Jeffreys mixture is not asymptotically minimax. Hence, we can assume that g is not the Jeffreys mixture and pick $\tilde{n}$ and $x^{\tilde{n}}$ such that for some positive constant ε,

$$\ln\frac{p_{B,1/2}(x^{\tilde{n}})}{g(x^{\tilde{n}})} \geq \varepsilon. \quad (10)$$

By (9) and Lemma 1, we can find n_0 such that for all $n > n_0$ and all sequences x^n,

$$\ln\frac{p^{(n)}_{\mathrm{NML}}(x^n)}{p_{B,1/2}(x^n)} \geq -\frac{\varepsilon}{2}. \quad (11)$$

Then for all $n > \max\{\tilde{n}, n_0\}$, there exists a sequence x^n which is a continuation of the $x^{\tilde{n}}$ in (10), such that

$$\begin{aligned}\ln\frac{p^{(n)}_{\mathrm{NML}}(x^n)}{g(x^n)} &= \ln\frac{p^{(n)}_{\mathrm{NML}}(x^n)}{p_{B,1/2}(x^n)} + \ln\frac{p_{B,1/2}(x^n)}{g(x^n)} \\ &= \ln\frac{p^{(n)}_{\mathrm{NML}}(x^n)}{p_{B,1/2}(x^n)} + \ln\frac{p_{B,1/2}(x^n_{\tilde{n}+1}|x^{\tilde{n}})}{g(x^n_{\tilde{n}+1}|x^{\tilde{n}})} + \ln\frac{p_{B,1/2}(x^{\tilde{n}})}{g(x^{\tilde{n}})} \\ &\geq -\frac{\varepsilon}{2} + \varepsilon = \frac{\varepsilon}{2}, \end{aligned} \quad (12)$$

where the identity $\ln g(x^n) = \ln g(x^n_{\tilde{n}+1}|x^{\tilde{n}}) + \ln g(x^{\tilde{n}})$ implied by horizon-independence is used on the second row. The last inequality follows from (10), (11) and the fact that $g(x^n_{\tilde{n}+1}|x^{\tilde{n}})$ is a conditional probability distribution of $x^n_{\tilde{n}+1}$. Note that since (11) holds

for all continuations of $x^{\tilde{n}}$, it is sufficient that there exists one continuation for which $p_{B,1/2}(x_{\tilde{n}+1}^n|x^{\tilde{n}})/g(x_{\tilde{n}+1}^n|x^{\tilde{n}}) \geq 1$ holds on the second row of (12). ■

It will be interesting to study whether similar results as above can be obtained for other models than the multinomial. For models where the NML is horizon-dependent and the Jeffreys mixture satisfies the convergence to NML in the sense of (11), we can use the same proof technique to prove the non-achievability by horizon-independent strategies. Here we provide an alternative approach that leads to a weaker result, Theorem 3, showing that a slightly stronger notion of asymptotic minimaxity is unachievable under the following condition on the horizon-dependence of the NML distribution.

Assumption 1 *Suppose that for $\tilde{n}$ satisfying $\tilde{n} \to \infty$ and $\frac{\tilde{n}}{n} \to 0$ as $n \to \infty$ (e.g. $\tilde{n} = \sqrt{n}$), there exist a sequence $x^{\tilde{n}}$ and a unique constant $M > 0$ such that*

$$\ln \frac{p_{\mathrm{NML}}^{(\tilde{n})}(x^{\tilde{n}})}{\sum_{x_{\tilde{n}+1}^n} p_{\mathrm{NML}}^{(n)}(x^n)} \to M \quad (n \to \infty). \tag{13}$$

Assumption 1 means that the NML distribution changes over the sample size n by an amount that is characterized by M. The following theorem proves that under this assumption, a stronger notion of asymptotic minimaxity is never achieved simultaneously for the sample sizes $\tilde{n}$ and n by a strategy g that is independent of n.

Theorem 3 *Under Assumption 1, if a distribution g is horizon-independent, then it never satisfies*

$$\ln C_n - \underline{M} + o(1) \leq \ln \frac{p(x^n|\hat{\theta}(x^n))}{g(x^n)} \leq \ln C_n + o(1), \tag{14}$$

for all x^n and any $\underline{M} < M$, where M is the constant appearing in Assumption 1 and $o(1)$ is a term converging to zero uniformly on x^n as $n \to \infty$.

The proof is given in Appendix B.

The condition in (14) is stronger than the usual asymptotic minimax optimality in (3), where only the second inequality in (14) is required. Intuitively, this stronger notion of asymptotic minimaxity requires not only that for all sequences, the regret of the distribution g is asymptotically at most the minimax value, but also that for no sequence, the regret is asymptotically *less* than the minimax value by a margin characterized by $\underline{M}$. Note that non-asymptotically (without the $o(1)$ terms), the corresponding strong and weak minimax notions are equivalent.

The following additional result provides a way to assess the amount by which the NML distribution depends on the horizon in the multinomial model. At the same time, it evaluates the *conditional regret* of the NML distributions as studied by Rissanen and Roos (2007), Grünwald (2007), and Hedayati and Bartlett (2012).

Let l_j be the number of js in $x^{\tilde{n}}$ ($0 \leq l_j \leq \tilde{n}$, $\sum_{j=1}^m l_j = \tilde{n}$). It follows that

$$\ln \frac{p_{\mathrm{NML}}^{(\tilde{n})}(x^{\tilde{n}})}{\sum_{x_{\tilde{n}+1}^n} p_{\mathrm{NML}}^{(n)}(x^n)} = \ln \frac{\prod_{j=1}^m \left(\frac{l_j}{\tilde{n}}\right)^{l_j}}{\sum_{n_j \geq l_j} \binom{n-\tilde{n}}{n_j - l_j} \prod_{j=1}^m \left(\frac{n_j}{n}\right)^{n_j}} + \ln \frac{C_n}{C_{\tilde{n}}}, \tag{15}$$

where $\binom{n-\tilde{n}}{n_j-l_j} \equiv \binom{n-\tilde{n}}{n_1-l_1,\cdots,n_m-l_m}$ is the multinomial coefficient and $\sum_{n_j \geq l_j}$ denotes the summation over n_js satisfying $n_1 + \cdots + n_m = n$ and $n_j \geq l_j$ for $j = 1, 2, \cdots, m$. Lemma 4 evaluates

$$C_{n|x^{\tilde{n}}} \equiv \sum_{n_j \geq l_j} \binom{n-\tilde{n}}{n_j - l_j} \prod_{j=1}^{m} \left(\frac{n_j}{n}\right)^{n_j}$$

in (15). The proof is in Appendix C.[2]

Lemma 4 *$C_{n|x^{\tilde{n}}}$ is asymptotically evaluated as*

$$\ln C_{n|x^{\tilde{n}}} = \frac{m-1}{2} \ln \frac{n}{2\pi} + \ln \tilde{C}_{\frac{1}{2}} + o(1), \tag{16}$$

where $\tilde{C}_\alpha$ is defined for $\alpha > 0$ and $\{l_j\}_{j=1}^m$ as

$$\tilde{C}_\alpha = \frac{\prod_{j=1}^m \Gamma(l_j + \alpha)}{\Gamma(\tilde{n} + m\alpha)}. \tag{17}$$

Substituting (16) and (6) into (15), we have

$$\ln \frac{p^{(\tilde{n})}_{\mathrm{NML}}(x^{\tilde{n}})}{p^{(n)}_{\mathrm{NML}}(x^{\tilde{n}})} = -\frac{m-1}{2} \ln \frac{\tilde{n}}{2\pi} + \sum_{j=1}^{m} l_j \ln \frac{l_j}{\tilde{n}} - \ln \frac{\prod_{j=1}^m \Gamma(l_j + 1/2)}{\Gamma(\tilde{n} + m/2)} + o(1),$$

where $p^{(n)}_{\mathrm{NML}}(x^{\tilde{n}}) = \sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n)$. Applying Stirling's formula to $\ln \Gamma(\tilde{n} + m/2)$ expresses the right hand side as

$$-\sum_{j=1}^{m} f(l_j) + o(1),$$

where f is the function defined in Lemma 1.

To illustrate the degree to which the NML distribution depends on the horizon, take $l_1 = \tilde{n}$, $l_j = 0$ for $j = 2, \cdots, m$. By Lemma 1, we then have $\ln p^{(\tilde{n})}_{\mathrm{NML}}(x^{\tilde{n}}) - \ln p^{(n)}_{\mathrm{NML}}(x^{\tilde{n}}) = \frac{1}{2}(m-1)\ln 2 + o(1)$.

4. Asymptotic Minimax via Simpler Horizon-Dependence

We examine the asymptotic minimaxity of the Bayes mixture in (5). More specifically, we investigate the minimax optimal hyperparameter

$$\operatorname*{argmin}_{\alpha} \max_{x^n} \ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha}(x^n)} \tag{18}$$

2. For the Fisher information matrix $I(\theta)$ whose ijth element is given by $(I(\theta))_{ij} = -\sum_x p(x|\theta) \frac{\partial^2 \ln p(x|\theta)}{\partial\theta_i \partial\theta_j} = \delta_{i,j}/\theta_j$, the constant $\tilde{C}_{1/2}$ coincides with $\int \sqrt{|I(\theta)|} \prod_{j=1}^m \theta^{l_j} d\theta$. This proves that the asymptotic expression of the regret of the conditional NML (Grünwald, 2007, Equation (11.47), p.323) is valid for the multinomial model with the full parameter set rather than the restricted parameter set discussed by Grünwald (2007).

and show that it is asymptotically approximated by

$$\alpha_n = \frac{1}{2} - \frac{\ln 2}{2}\frac{1}{\ln n}. \tag{19}$$

As a function of $(n_1, \cdots, n_{m-1})$, the regret of $p_{B,\alpha}$ is

$$\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha}(x^n)} = \sum_{j=1}^{m} \{n_j \ln n_j - \ln \Gamma(n_j + \alpha)\} + \kappa \tag{20}$$

where $n_m = n - \sum_{j=1}^{m-1} n_j$ and κ denotes a constant that does not depend on $(n_1, \cdots, n_{m-1})$. We first prove the following lemma (Appendix D).

Lemma 5 *The possible worst-case sequences in (18) have l nonzero counts ($l = 1, 2, \cdots, m$), each of which is $\lfloor \frac{n}{l} \rfloor$ or $\lfloor \frac{n}{l} \rfloor + 1$ with all the other counts are zeros. Here $\lfloor \cdot \rfloor$ is the floor function, the largest integer not exceeding the argument.*

From this lemma, we focus on the regrets of the two extreme cases of x^n consisting of a single symbol repeated n times and x^n with a uniform number n/m of each symbol j. Let the regrets of these two cases be equal,

$$\Gamma(\alpha)^{m-1}\Gamma(n + \alpha) = \Gamma(n/m + \alpha)^m m^n. \tag{21}$$

Equating the regrets of these two cases also equates the regrets of $(n/l, \cdots, n/l, 0, \cdots, 0)$ for $1 \le l \le m$ up to $o(1)$ terms, which is verified by directly calculating the regrets. Note that equating the regrets of the m possible worst-case sequences leads to the least maximum regret. This is because the regrets at the m possible worst-case sequences are not equal, we can improve by reducing the regret at the actual worst-case sequence until it becomes equal to the other cases.

Taking logarithms, using Stirling's formula and ignoring diminishing terms in (21), we have

$$(m-1)\left(\alpha - \frac{1}{2}\right)\ln n - (m-1)\ln \Gamma(\alpha) - m\left(\alpha - \frac{1}{2}\right)\ln m + (m-1)\frac{\ln 2\pi}{2} = 0. \tag{22}$$

This implies that the optimal α is asymptotically given by

$$\alpha_n \simeq \frac{1}{2} - \frac{a}{\ln n}, \tag{23}$$

for some constant a. Substituting this back into (22) and solving it for a, we obtain (19).

We numerically calculated the optimal hyperparameter defined by (18) for the Bernoulli model ($m = 2$). Figure 1 shows the optimal α obtained numerically and its asymptotic approximation in (19). We see that the optimal hyperparameter is well approximated by α_n in (19) for large n. Note here the slow convergence speed, $O(1/\ln n)$ to the asymptotic value, $1/2$.

The next theorem shows the asymptotic minimaxity of α_n (the second inequality in (24)). We will examine the regret of α_n numerically in Section 5.1.

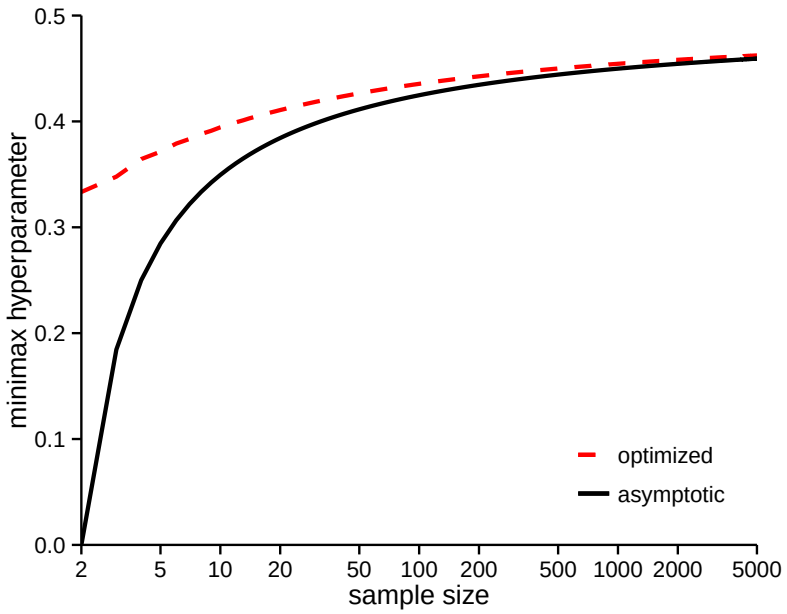


Figure 1: Minimax optimal hyperparameter α for sample size n

Theorem 6 *For the multinomial model in (4), the Bayes mixture defined by the prior* $\mathrm{Dir}(\alpha_n, \cdots, \alpha_n)$ *is asymptotic minimax and satisfies*

$$\ln C_n - M + o(1) \leq \ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha_n}(x^n)} \leq \ln C_n + o(1), \tag{24}$$

for all x^n *where* $M = (m-1)\ln 2/2$, *and* $\ln C_n$ *is the minimax regret evaluated asymptotically in (6).*

The proof is given in Appendix E.

5. Numerical Results

In this section, we numerically calculate the maximum regrets of several methods in the Bernoulli model ($m = 2$). The following two subsections respectively examine horizon-dependent algorithms based on Bayes mixtures with prior distributions depending on n and last-step minimax algorithms, which are horizon-independent.

5.1 Optimal Conjugate Prior and Modified Jeffreys Prior

We calculated the maximum regrets of the Bayes mixtures in (5) with the hyperparameter optimized by the golden section search and with its asymptotic approximation in (19). We also investigated the maximum regrets of Xie and Barron's modified Jeffreys prior which is proved to be asymptotic minimax (Xie and Barron, 2000). The modified Jeffreys prior is

defined by

$$q_{\mathrm{MJ}}^{(n)}(\theta) = \frac{\epsilon_n}{2}\left\{\delta\left(\theta - \frac{1}{n}\right) + \delta\left(\theta - 1 + \frac{1}{n}\right)\right\} + (1-\epsilon_n)b_{1/2}(\theta),$$

where δ is the Dirac's delta function and $b_{1/2}(\theta)$ is the density function of the beta distribution with hyperparameters 1/2, Beta$(1/2, 1/2)$, which is the Jeffreys prior for the Bernoulli model. We set $\epsilon_n = n^{-1/8}$ as proposed by Xie and Barron (2000) and also optimized ϵ_n by the golden section search so that the maximum regret

$$\max_{x^n} \ln \frac{p(x^n|\hat{\theta}(x^n))}{\int p(x^n|\theta) q_{\mathrm{MJ}}^{(n)}(\theta) d\theta}$$

is minimized.

Figure 2(a) shows the maximum regrets of these Bayes mixtures: asymptotic and optimized Beta refer to mixtures with Beta priors (Section 4), and modified Jeffreys methods refer to mixtures with a modified Jeffreys prior as discussed above. Also included for comparison is the maximum regret of the Jeffreys mixture (Krichevsky and Trofimov, 1981), which is not asymptotic minimax. To better show the differences, the regret of the NML distribution, $\ln C_n$, is subtracted from the maximum regret of each distribution.

We see that the maximum regrets of these distributions, except the one based on Jeffreys prior, decrease toward the regret of NML as n grows as implied by their asymptotic minimaxity. The modified Jeffreys prior with the optimized weight performs best of these strategies for this range of the sample size. For moderate and large sample sizes ($n > 100$), the asymptotic minimax hyperparameter, which can be easily evaluated by (19), performs almost as well as the optimized strategies which are not known analytically. Note that unlike the NML, Bayes mixtures provide the conditional probabilities $p(x_{\tilde{n}} \mid x_1, \ldots, x_{\tilde{n}-1})$ even if the prior depends on n. The time complexity for online prediction will be discussed in Section 5.3.

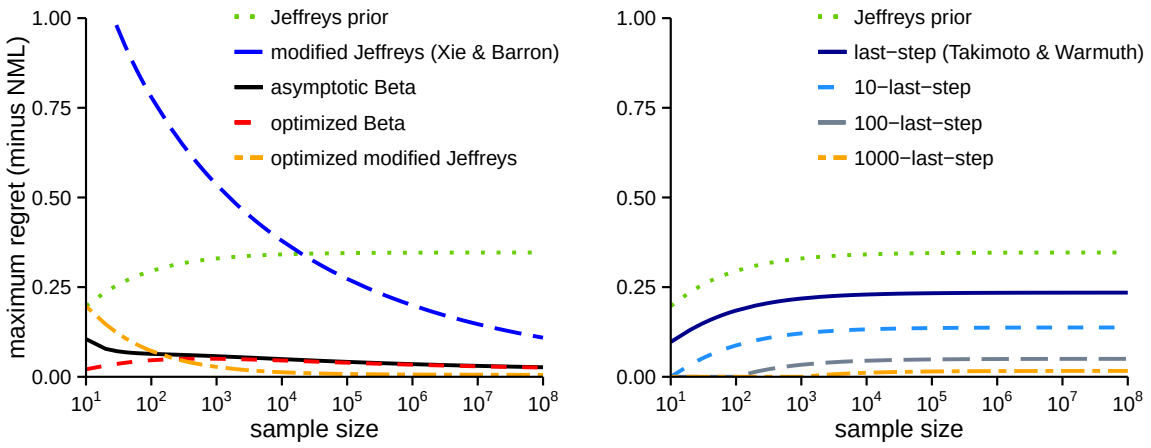
5.2 Last-Step Minimax Algorithms

The last-step minimax algorithm is an online prediction algorithm that is equivalent to the so called sequential normalized maximum likelihood method in the case of the multinomial model (Rissanen and Roos, 2007; Takimoto and Warmuth, 2000). A straightforward generalization, which we call the k-last-step minimax algorithm, normalizes $p(x^t|\hat{\theta}(x^t))$ over the last $k \geq 1$ steps to calculate the conditional distribution of $x_{t-k+1}^t = \{x_{t-k+1}, \cdots, x_t\}$,

$$p_{\mathrm{kLS}}(x_{t-k+1}^t|x^{t-k}) = \frac{p(x^t|\hat{\theta}(x^t))}{L_{t,k}},$$

where $L_{t,k} = \sum_{x_{t-k+1}^t} p(x^t|\hat{\theta}(x^t))$. Although this generalization was mentioned by Takimoto and Warmuth (2000), it was left as an open problem to examine how k affects the regret of the algorithm.

Our main result (Theorem 2) tells that k-last-step minimax algorithm with k independent of n is not asymptotic minimax. We numerically calculated the regret of the k-last-step minimax algorithm with $k = 1$, 10, 100 and 1000 for the sequence $x^n = 1010101010\cdots$ since



(a) Horizon-dependent algorithms

(b) Horizon-independent algorithms (lower bounds)

Figure 2: Maximum regret for sample size n. The regret of the NML distribution, $\ln C_n$, is subtracted from the maximum regret of each strategy. The first two algorithms (from the top) in each panel are from earlier work, while the remaining ones are novel.

it is infeasible to evaluate the maximum regret for large n. The regret for this particular sequence provides a lower bound for the maximum regret. Figure 2(b) shows the regret as a function of n together with the maximum regret of the Jeffreys mixture. The theoretical asymptotic regret for the Jeffreys mixture is $\frac{\ln 2}{2} \approx 0.34$ (Krichevsky and Trofimov, 1981), and the asymptotic bound for the 1-last-step minimax algorithm is slightly better, $\frac{1}{2}\left(1 - \ln \frac{\pi}{2}\right) \approx 0.27$ (Takimoto and Warmuth, 2000). We can see that although the regret decreases as k grows, it still increases as n grows and does not converge to that of the NML (zero in the figure).

5.3 Computational Complexity

As mentioned above, in the multinomial model, the NML probability of individual sequences of length n can be evaluated in linear time (Kontkanen and Myllymäki, 2007). However, for prediction purposes in online scenarios, we need to compute the predictive probabilities $p^{(n)}_{\mathrm{NML}}(x_t|x^{t-1})$ by summing over all continuations of x^t. Computing all the predictive probabilities up to n by this method takes the time complexity of $O(m^n)$. For all the other algorithms except NML, the complexity is $O(n)$ when m is considered fixed. More specifically, for Bayes mixtures, the complexity is $O(mn)$ and for k-laststep minimax algorithms, the complexity is $O(m^k n)$.

We mention that it was recently proposed that the computational complexity of the prediction strategy based on NML may be significantly reduced by representing the NML distribution as a Bayes-like mixture with a horizon-dependent prior (Barron et al., 2014). The authors show that for a parametric family with a finite-valued sufficient statistic, the

exact NML is achievable by a Bayes mixture with a signed discrete prior designed depending on the horizon n. The resulting prediction strategy may, however, require updating as many as $n/2+1$ weights on each prediction step even in the Bernoulli case, which leads to total time complexity of order n^2.

6. Conclusions

We characterized the achievability of asymptotic minimax coding regret in terms of horizon-dependency. The results have implications on probabilistic prediction, data compression, and model selection based on the MDL principle, all of which depend on predictive models or codes that achieve low logarithmic losses or short code-lengths. For multinomial models, which have been very extensively studied, our main result states that no horizon-independent strategy can be asymptotic minimax. A weaker result involving a stronger minimax notion is given for more general models. Future work can focus on obtaining precise results for different model classes where achievability of asymptotic minimaxity is presently unknown.

Our numerical experiments show that several easily implementable Bayes and other strategies are nearly optimal. In particular, a novel predictor based on a simple asymptotically optimal horizon-dependent Beta (or Dirichlet) prior, for which a closed form expression is readily available, offers a good trade-off between computational cost and worst-case regret. Overall, differences in the maximum regrets of many of the strategies under the Bernoulli model (Figure 2) are small (less than 1 nat). Such small differences may nevertheless be important from a practical point of view. For instance, it has been empirically observed that slight differences in the Dirichlet hyperparameter, leading to relatively small changes in the marginal probabilities, can be significant in Bayesian network structure learning (Silander et al., 2007). Furthermore, the differences are likely to be greater under multinomial ($m > 2$) and other models, which is another direction for future work.

Acknowledgments

The authors thank Andrew Barron and Nicolò Cesa-Bianchi for valuable comments they provided at the WITMSE workshop in Tokyo in 2013. We also thank the anonymous reviewers. This work was supported in part by the Academy of Finland (via the Center-of-Excellence COIN) and by JSPS grants 23700175 (15K16050) and 25120014. Part of this work was carried out when KW was visiting HIIT in Helsinki.

Appendix A. Proof of Lemma 1

Proof The function f is non-decreasing since $f'(x) = \psi(x+1/2) - \ln x \geq 0$ where $\psi(x) = (\ln \Gamma(x))'$ is the digamma function (Merkle, 1998). $\lim_{x\to\infty} f(x) = 0$ is derived from Stirling's formula,

$$\ln \Gamma(x) = \left(x - \frac{1}{2}\right) \ln x - x + \frac{1}{2} \ln(2\pi) + O\left(\frac{1}{x}\right).$$

It immediately follows from $f(0) = -\frac{\ln 2}{2}$ and this limit that $-\frac{\ln 2}{2} \leq f(x) < 0$ for $x \geq 0$. ■

Appendix B. Proof of Theorem 3

Proof Under Assumption 1, we suppose (14) holds for all sufficiently large n and derive contradiction. The inequalities in (14) are equivalent to

$$-\underline{M} + o(1) \le \ln \frac{p^{(n)}_{\mathrm{NML}}(x^n)}{g(x^n)} \le o(1).$$

For a horizon-independent strategy g we can expand the marginal probability $g(x^{\tilde{n}})$ in terms of the following sum and apply the above lower bound to obtain

$$\begin{aligned} g(x^{\tilde{n}}) &= \sum_{x^n_{\tilde{n}+1}} g(x^n) = \sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n) e^{-\ln \frac{p^{(n)}_{\mathrm{NML}}(x^n)}{g(x^n)}} \\ &\le e^{\underline{M}+o(1)} \sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n) \end{aligned} \tag{25}$$

for all $x^{\tilde{n}}$. Then we have

$$\begin{aligned} \max_{x^{\tilde{n}}} \ln \frac{p^{(\tilde{n})}_{\mathrm{NML}}(x^{\tilde{n}})}{g(x^{\tilde{n}})} &= \max_{x^{\tilde{n}}} \left\{ \ln \frac{p^{(\tilde{n})}_{\mathrm{NML}}(x^{\tilde{n}})}{\sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n)} + \ln \frac{\sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n)}{g(x^{\tilde{n}})} \right\} \\ &\ge \max_{x^{\tilde{n}}} \left\{ \ln \frac{p^{(\tilde{n})}_{\mathrm{NML}}(x^{\tilde{n}})}{\sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n)} \right\} - \underline{M} + o(1) \\ &\ge \epsilon + o(1), \end{aligned}$$

where $\epsilon = M - \underline{M} > 0$. The first inequality follows from (25) and the second inequality follows from Assumption 1, which implies $\max_{x^{\tilde{n}}} \ln \frac{p^{(\tilde{n})}_{\mathrm{NML}}(x^{\tilde{n}})}{\sum_{x^n_{\tilde{n}+1}} p^{(n)}_{\mathrm{NML}}(x^n)} \ge M + o(1)$. The above inequality contradicts the asymptotic minimax optimality in (14) with n replaced by $\tilde{n}$. ■

Appendix C. Proof of Lemma 4

Proof In order to prove Lemma 4, we modify and extend the proof in Xie and Barron (2000) for the asymptotic evaluation of $\ln C_n = \ln \sum_{x^n} p(x^n|\hat{\theta}(x^n))$ given by (6) to that of $\ln C_{n|x^{\tilde{n}}} = \ln \sum_{x^n_{\tilde{n}+1}} p(x^n|\hat{\theta}(x^n))$, which is conditioned on the first $\tilde{n}$ samples, $x^{\tilde{n}}$. More specifically, we will prove the following inequalities. Here, $p_{B,w}$ denotes the Bayes mixture defined by the prior $w(\theta)$, $p_{B,1/2}$ and p_{B,α_n} are those with the Dirichlet priors, $\mathrm{Dir}(1/2, \cdots, 1/2)$ (Jeffreys mixture) and $\mathrm{Dir}(\alpha_n, \cdots, \alpha_n)$ where $\alpha_n = \frac{1}{2} - \frac{\ln 2}{2} \frac{1}{\ln n}$ respectively.

$$\frac{m-1}{2} \ln \frac{n}{2\pi} + \tilde{C}_{\frac{1}{2}} + o(1) \le \sum_{x^n_{\tilde{n}+1}} p_{B,1/2}(x^n_{\tilde{n}+1}|x^{\tilde{n}}) \ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,1/2}(x^n_{\tilde{n}+1}|x^{\tilde{n}})} \tag{26}$$

$$
\begin{aligned}
&\leq \max_{w} \sum_{x_{\tilde{n}+1}^n} p_{B,w}(x_{\tilde{n}+1}^n|x^{\tilde{n}}) \ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,w}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} \\
&= \max_{w} \min_{\overline{p}} \sum_{x_{\tilde{n}+1}^n} p_{B,w}(x_{\tilde{n}+1}^n|x^{\tilde{n}}) \ln \frac{p(x^n|\hat{\theta}(x^n))}{\overline{p}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} \\
&\leq \min_{\overline{p}} \max_{x_{\tilde{n}+1}^n} \ln \frac{p(x^n|\hat{\theta}(x^n))}{\overline{p}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} \\
&= \ln \sum_{x_{\tilde{n}+1}^n} p(x^n|\hat{\theta}(x^n)) = \ln C_{n|x^{\tilde{n}}} \\
&\leq \max_{x_{\tilde{n}+1}^n} \ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha_n}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} \\
&\leq \frac{m-1}{2} \ln \frac{n}{2\pi} + \tilde{C}_{\frac{1}{2}} + o(1), \qquad (27)
\end{aligned}
$$

where the first equality follows from Gibbs' inequality, and the second equality as well as the second to last inequality follow from the minimax optimality of NML (Shtarkov, 1987). Let us move on to the proof of inequalities (26) and (27). The rest of the inequalities follow from the definitions and from the fact that maximin is no greater than minimax. To derive both inequalities, we evaluate $\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha}(x_{\tilde{n}+1}^n|x^{\tilde{n}})}$ for the Bayes mixture with the prior $\mathrm{Dir}(\alpha, \cdots, \alpha)$ asymptotically. It follows that

$$
\begin{aligned}
\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} &= \ln \frac{\prod_{j=1}^m \left(\frac{n_j}{n}\right)^{n_j}}{\frac{\Gamma(\tilde{n}+m\alpha)}{\Gamma(n+m\alpha)} \prod_{j=1}^m \frac{\Gamma(n_j+\alpha)}{\Gamma(l_j+\alpha)}} \\
&= \sum_{j=1}^m n_j \ln n_j - n \ln n - \sum_{j=1}^m \ln \Gamma(n_j+\alpha) + \ln \Gamma(n+m\alpha) + \ln \tilde{C}_\alpha \\
&= \sum_{j=1}^m \left\{ n_j \ln n_j - n_j - \ln \Gamma(n_j+\alpha) + \frac{1}{2} \ln(2\pi) \right\} \\
&\quad + \left(m\alpha - \frac{1}{2} \right) \ln n - (m-1)\frac{1}{2} \ln(2\pi) + \ln \tilde{C}_\alpha + o(1), \qquad (28)
\end{aligned}
$$

where $\tilde{C}_\alpha$ is defined in (17) and we applied Stirling's formula to $\ln \Gamma(n+m\alpha)$.

Substituting $\alpha = 1/2$ into (28), we have

$$
\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,1/2}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} = \sum_{j=1}^m \left(c_{n_j} + \frac{\ln 2}{2} \right) + \frac{m-1}{2} \ln \frac{n}{2\pi} + \ln \tilde{C}_{1/2} + o(1),
$$

where

$$
c_k = k \ln k - k - \ln \Gamma(k+1/2) + \frac{1}{2} \ln \pi, \qquad (29)
$$

for $k \geq 0$. Since from Lemma 1, $-\frac{\ln 2}{2} < c_k$,

$$
\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,1/2}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} > \frac{m-1}{2} \ln \frac{n}{2\pi} + \ln \tilde{C}_{1/2} + o(1),
$$

holds for all x^n, which proves the inequality (26).

Substituting $\alpha = \alpha_n = \frac{1}{2} - \frac{\ln 2}{2}\frac{1}{\ln n}$ into (28), we have

$$\begin{aligned}\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha_n}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} &= \sum_{j=1}^{m}\left\{n_j \ln n_j - n_j - \ln\Gamma(n_j+\alpha_n) + \frac{1}{2}\ln\pi\right\} \\ &\quad + \frac{m-1}{2}\ln\frac{n}{2\pi} + \ln\tilde{C}_{1/2} + o(1).\end{aligned}$$

Assuming that the first l n_js ($j = 1, \cdots, l$) are finite and the rest are large (tend to infinity as $n \to \infty$) and applying Stirling's formula to $\ln\Gamma(n_j + \alpha_n)$ ($j = l+1, \cdots, m$), we have

$$\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha_n}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} = \sum_{j=1}^{l} c_{n_j} + \sum_{j=l+1}^{m} d_{n_j} + \frac{m-1}{2}\ln\frac{n}{2\pi} + \ln\tilde{C}_{1/2} + o(1), \tag{30}$$

where c_k is defined in (29) and

$$d_k = \frac{\ln 2}{2}\left(\frac{\ln k}{\ln n} - 1\right)$$

for $1 < k \le n$. Since $c_k \le 0$ follows from Lemma 1 and $d_k \le 0$, we obtain

$$\ln \frac{p(x^n|\hat{\theta}(x^n))}{p_{B,\alpha_n}(x_{\tilde{n}+1}^n|x^{\tilde{n}})} \le \frac{m-1}{2}\ln\frac{n}{2\pi} + \ln\tilde{C}_{1/2} + o(1), \tag{31}$$

for all x^n, which proves the inequality (27). ■

Appendix D. Proof of Lemma 5

Proof The summation in (20) is decomposed into three parts,

$$\begin{aligned}&\{n_1 \ln n_1 - \ln\Gamma(n_1+\alpha)\} + \{(n'-n_1)\ln(n'-n_1) - \ln\Gamma(n'-n_1+\alpha)\} \\ &+ \sum_{j=2}^{m-1}\{n_j \ln n_j - \ln\Gamma(n_j+\alpha)\},\end{aligned}$$

where $n' = n - \sum_{j=2}^{m-1} n_j$. We analyze the regret of the multinomial case by reducing it to the binomial case since the summation in the above expression is constant with respect to n_1. Hence, we focus on the regret of the binomial case with sample size n',

$$R(z) = z\ln z - \ln\Gamma(z+\alpha) + (n'-z)\ln(n'-z) - \ln\Gamma(n'-z+\alpha),$$

as a function of $0 \le z \le \frac{n'}{2}$ because of the symmetry. We prove that the maximum of R is attained at the boundary ($z = 0$) or at the middle $z = \frac{n'}{2}$. We will use the following inequalities for $z \ge 0$,

$$\left(\Psi'(z)\right)^2 + \Psi^{(2)}(z) > 0, \tag{32}$$

and

$$2\left(-\Psi^{(2)}(z)\right)^{3/2} - \Psi^{(3)}(z) > 0, \tag{33}$$

which are directly obtained from Theorem 2.2 of Batir (2007).

The derivative of R is

$$R'(z) = h(z) - h(n' - z),$$

where

$$h(z) = \ln z - \Psi(z + \alpha).$$

We can prove that $h'(z) = \frac{1}{z} - \Psi'(z + \alpha)$ has at most one zero since (32) shows that the derivative of the function $z - \frac{1}{\Psi'(z+\alpha)}$ is positive, which implies that it is monotonically increasing from $-1/\Psi'(\alpha) < 0$ and hence has at most one zero coinciding with the zero of h'. Noting also that $\lim_{z\to 0} h(z) = -\infty$ and $\lim_{z\to\infty} h(z) = 0$, we see that there are the following two cases: (a) $h(z)$ is monotonically increasing in the interval $(0,\ n')$, and (b) $h(z)$ is unimodal with a unique maximum in $(0,\ n')$. In the case of (a), R' has no zero in the interval $(0,\ n'/2)$, which means that R is V-shaped, takes global minimum at $z = \frac{n'}{2}$, and has the maxima at the boundaries. In the case of (b), $R'(z) = 0$ has at most one solution in the interval $(0,\ n'/2)$, which is proved as follows.

The higher order derivatives of R are

$$\begin{aligned} R^{(2)}(z) &= h'(z) + h'(n' - z), \\ R^{(3)}(z) &= h^{(2)}(z) - h^{(2)}(n' - z), \end{aligned}$$

where $h^{(2)}(z) = -\frac{1}{z^2} - \Psi^{(2)}(z + \alpha)$. Let the unique zero of $h'(z)$ be z^* (if there is no zero, let $z^* = \infty$). If $z^* < \frac{n'}{2}$, since for $z^* \le z < n'/2$, $h'(z) \le 0$ and $h'(n' - z) \le 0$, we have $R^{(2)}(z) \le 0$, which means that $R'(z)$ is monotonically decreasing to $R'\left(\frac{n'}{2}\right) = 0$. That is, $R'(z) > 0$ for $z^* \le z < \frac{n'}{2}$. Hence, we focus on $z \le z^*$ and prove that $R'(z)$ is concave for $z \le z^*$, which, from $\lim_{z\to 0} R'(z) = -\infty$, means that $R'(z)$ has one zero if $R^{(2)}\left(\frac{n'}{2}\right) = 2h'\left(\frac{n'}{2}\right) < 0$, and $R'(z)$ has no zero otherwise.[3]

For $z \le z^*$, since $\frac{1}{z} > \Psi'(z + \alpha)$ holds, we have

$$h^{(2)}(z) = -\frac{1}{z^2} - \Psi^{(2)}(z + \alpha) < -\Psi'(z + \alpha)^2 - \Psi^{(2)}(z + \alpha) < 0, \tag{34}$$

from (32). Define $\tilde{h}(z) = z - \frac{1}{\sqrt{-\Psi^{(2)}(z+\alpha)}}$, for which $\tilde{h}(z) = 0$ is equivalent to $h^{(2)}(z) = 0$. Then $\tilde{h}(0) < 0$ and it follows from (33) that

$$\tilde{h}'(z) = 1 - \frac{\Psi^{(3)}(z + \alpha)}{2\left(-\Psi^{(2)}(z + \alpha)\right)^{3/2}} > 0,$$

which implies that $\tilde{h}(z)$ is monotonically increasing, and hence that $h^{(2)}(z) = 0$ has at most one solution. Let z^{**} be the unique zero of $h^{(2)}(z)$ (if there is no zero, let $z^{**} = \infty$). Noting

3. In case (b) where $h(z)$ is unimodal with a maximum in $(0,\ n')$, the condition that $h'\left(\frac{n'}{2}\right) \ge 0$ is equivalent to $z^* \ge \frac{n'}{2}$.

that $\lim_{z\to 0} h^{(2)}(z) = -\infty$, we see that $h^{(2)}(z) < 0$ for $z < z^{**}$ and $h^{(2)}(z) > 0$ for $z > z^{**}$. From (34), $z^* < z^{**}$ holds. For $z < z^{**}$, since $h^{(2)}(z) < 0$ implies that $-\frac{1}{z^2} < \Psi^{(2)}(z+\alpha)$, and hence $\frac{1}{z} > \sqrt{-\Psi^{(2)}(z+\alpha)}$ holds, it follows from (33) that

$$h^{(3)}(z) = \frac{2}{z^3} - \Psi^{(3)}(z+\alpha) > 2\left(-\Psi^{(2)}(z+\alpha)\right)^{3/2} - \Psi^{(3)}(z+\alpha) > 0.$$

This means that $h^{(2)}(z)$ is monotonically increasing for $z < z^{**}$. Therefore, $h^{(2)}(z)$ is negative and monotonically increasing for $z < z^{**}$, implying that $R^{(3)}(z)$ has no zero for $z \le z^{**}$ since $h^{(2)}(z) < h^{(2)}(n'-z)$, that is, $R^{(3)}(z) < 0$ holds. Thus $R'(z)$ is concave for $z \le z^* < z^{**}$, and hence $R'(z)$ has at most one zero between 0 and z^*.

Note that $\lim_{z\to 0} R'(z) = -\infty$ and $R'(n'/2) = 0$. If $R'(z) = 0$ has no solution in $(0,\ n'/2)$, that is, if $h'\left(\frac{n'}{2}\right) = \frac{2}{n'} - \Psi'\left(\frac{n'}{2}+\alpha\right) \ge 0$, the regret function looks similarly to the case of (a), and the maxima are attained at the boundaries. If $R'(z) = 0$ has a solution in $(0,\ n'/2)$, that is, if $\frac{2}{n'} - \Psi'\left(\frac{n'}{2}+\alpha\right) < 0$, R' changes its sign around the solution from negative to positive as z grows. In this case, R is W-shaped with possible maximum at the boundaries or at the middle.

We see that in any case, the maximum is always at the boundary or at the middle. Therefore, as a function of the count n_1, $R(n_1)$ is maximized at $n_1 = 0$ or at $n_1 = \lfloor \frac{n'}{2} \rfloor$ (or $n_1 = \lfloor \frac{n'}{2} \rfloor + 1$ if n' is odd). The same argument applies to optimizing n_j $(j = 2, 3, \cdots, m-1)$. Thus, if the counts are such that for any two indices, i and j, $n_i > n_j + 1 > 1$, then we can increase the regret either by replacing one of them by the sum, $n_i + n_j$ and the other one by zero or by replacing them by new values n_i' and n_j' such that $|n_i - n_j| \le 1$. This completes the proof of the lemma. ■

Appendix E. Proof of Theorem 6

Proof The proof of Lemma 4 itself applies to the case where $\tilde{n} = 0$ and $l_j = 0$ for $j = 1, \cdots, m$ as well. Since, in this case, $\tilde{C}_{1/2} = \ln \frac{\Gamma(1/2)^m}{\Gamma(m/2)}$, the inequality (31) in the proof gives the right inequality in (24).

Furthermore, in (30), we have

$$\sum_{j=1}^{l} c_{n_j} + \sum_{j=l+1}^{m} d_{n_j} > -(m-1)\frac{\ln 2}{2} + o(1). \tag{35}$$

This is because, from Lemma 1 and definition, $c_{n_j}, d_{n_j} > -\frac{\ln 2}{2}$ and for at least one of j, n_j is in the order of n since $\sum_{j=1}^{n} n_j = n$, which means that $d_{n_j} = o(1)$ for some j. Substituting (35) into (30), we obtain the left inequality in (24) with $M = \frac{1}{2}(m-1)\ln 2$. ■

References

K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43(3):211–246, 2001.

A. R. Barron, T. Roos, and K. Watanabe. Bayesian properties of normalized maximum likelihood and its fast computation. In *Proc. 2014 IEEE International Symposium on Information Theory*, pages 1667–1671, 2014.

P. Bartlett, P. Grünwald, P. Harremoës, F. Hedayati, and W. Kotłowski. Horizon-independent optimal prediction with log-loss in exponential families. In *JMLR: Workshop and Conference Proceedings: 26th Annual Conference on Learning Theory*, volume 30, pages 639–661, 2013.

N. Batir. On some properties of digamma and polygamma functions. *Journal of Mathematical Analysis and Applications*, 328(1):452–465, 2007.

N. Cesa-Bianchi and G. Lugosi. Worst-case bounds for the logarithmic loss of predictors. *Machine Learning*, 43(3):247–264, 2001.

N. Cesa-Bianchi, Y. Freund, D. Haussler, D. P. Helmbold, R. E. Schapire, and M. K. Warmuth. How to use expert advice. *Journal of the ACM*, 44(3):427–485, 1997.

Y. Freund. Predicting a binary sequence almost as well as the optimal biased coin. In *Proc. 9th Annual Conference on Computational Learning Theory*, pages 89–98, 1996.

P. D. Grünwald. *The Minimum Description Length Principle*. The MIT Press, 2007.

F. Hedayati and P. L. Bartlett. Exchangeability characterizes optimality of sequential normalized maximum likelihood and Bayesian prediction with Jeffreys prior. In *JMLR: Workshop and Conference Proceedings: 15th International Conference on Artificial Intelligence and Statistics*, volume 22, pages 504–510, 2012.

P. Kontkanen and P. Myllymäki. A linear-time algorithm for computing the multinomial stochastic complexity. *Information Processing Letters*, 103(6):227–233, 2007.

W. Kotłowski and P. D. Grünwald. Maximum likelihood vs. sequential normalized maximum likelihood in on-line density estimation. In *JMLR: Workshop and Conference Proceedings: 24th Annual Conference on Learning Theory*, volume 19, pages 457–476, 2011.

R. E. Krichevsky. Laplace's law of succession and universal encoding. *IEEE Trans. Information Theory*, 44(1):296–303, 1998.

R. E. Krichevsky and V. K. Trofimov. The performance of universal encoding. *IEEE Trans. Information Theory*, IT-27(2):199–207, 1981.

P. S. Laplace. *A Philosophical Essay on Probabilities*. Dover, New York, 1795/1951.

F. Liang and A. Barron. Exact minimax strategies for predictive density estimation, data compression, and model selection. *IEEE Trans. Informaton Theory*, 50:2708–2726, 2004.

H. Luo and R. Schapire. Towards minimax online learning with unknown time horizon. In *JMLR: Workshop and Conference Proceedings: 31st International Conference on Machine Learning*, volume 32, pages 226–234, 2014.

N. Merhav and M. Feder. Universal prediction. *IEEE Trans. Information Theory*, 44: 2124–2147, 1998.

M. Merkle. Conditions for convexity of a derivative and some applications to the Gamma function. *Aequationes Mathematicae*, 55:273–280, 1998.

J. Rissanen. Fisher information and stochastic complexity. *IEEE Trans. Information Theory*, IT-42(1):40–47, 1996.

J. Rissanen and T. Roos. Conditional NML universal models. In *Proc. 2007 Information Theory and Applications Workshop*, pages 337–341. IEEE Press, 2007.

Y. M. Shtarkov. Universal sequential coding of single messages. *Problems of Information Transmission*, 23(3):175–186, 1987.

T. Silander, P. Kontkanen, and P. Myllymäki. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In *Proc. 27th Conference on Uncertainty in Artificial Intelligence*, pages 360–367, 2007.

T. Silander, T. Roos, and P. Myllymäki. Learning locally minimax optimal Bayesian networks. *International Journal of Approximate Reasoning*, 51(5):544–557, 2010.

J. Takeuchi and A. R. Barron. Asymptotically minimax regret for exponential families. In *Proc. 20th Symposium on Information Theory and its Applications*, pages 665–668, 1997.

E. Takimoto and M. K. Warmuth. The last-step minimax algorithm. In *Algorithmic Learning Theory, Lecture Notes in Computer Science*, volume 1968, pages 279–290, 2000.

Q. Xie and A. R. Barron. Asymptotic minimax regret for data compression, gambling, and prediction. *IEEE Trans. Information Theory*, 46(2):431–445, 2000.

Journal of Machine Learning Research 16 (2015) 2377-2404 Submitted 8/13; Revised 1/15; Published 12/15

Multiclass Learnability and the ERM Principle

Amit Daniely AMIT.DANIELY@MAIL.HUJI.AC.IL
Dept. of Mathematics, The Hebrew University, Givat-Ram Campus, Jerusalem 91904, Israel

Sivan Sabato SIVAN.SABATO@MICROSOFT.COM
Dept. of Computer Science, Ben-Gurion University of the Negev, Beer Sheva 8410501, Israel

Shai Ben-David SHAI@CS.UWATERLOO.CA
David R. Cheriton School of Computer Science, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1

Shai Shalev-Shwartz SHAIS@CS.HUJI.AC.IL
School of Computer Science and Engineering, The Hebrew University, Givat-Ram Campus, Jerusalem 91904, Israel

Editor: Peter Auer

Abstract

We study the sample complexity of multiclass prediction in several learning settings. For the PAC setting our analysis reveals a surprising phenomenon: In sharp contrast to binary classification, we show that there exist multiclass hypothesis classes for which some Empirical Risk Minimizers (ERM learners) have lower sample complexity than others. Furthermore, there are classes that are learnable by some ERM learners, while other ERM learners will fail to learn them. We propose a principle for designing good ERM learners, and use this principle to prove tight bounds on the sample complexity of learning *symmetric* multiclass hypothesis classes—classes that are invariant under permutations of label names. We further provide a characterization of mistake and regret bounds for multiclass learning in the online setting and the bandit setting, using new generalizations of Littlestone's dimension.

Keywords: multiclass, sample complexity, ERM

1. Introduction

Multiclass prediction is the problem of classifying an object into one of several possible target classes. This task surfaces in many domains. Common practical examples include document categorization, object recognition in computer vision, and web advertisement.

The centrality of the multiclass learning problem has spurred the development of various approaches for tackling this task. Most of these approaches fall under the following general description: There is an instance domain $\mathcal{X}$ and a set of possible class labels $\mathcal{Y}$. The goal of the learner is to learn a mapping from instances to labels. The learner receives training examples, and outputs a predictor which belongs to some hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, where $\mathcal{Y}^{\mathcal{X}}$ is the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. We study the sample complexity of the task of learning $\mathcal{H}$, namely, how many random training examples are needed for learning an accurate predictor from $\mathcal{H}$. This question has been extensively studied and is quite well understood for the binary case (i.e, where $|\mathcal{Y}| = 2$). In contrast, as we shall see, existing theory of the multiclass case is less complete.

In the first part of the paper we consider multiclass learning in the classical PAC setting of Valiant (1984). Since the 1970's, following Vapnik and Chervonenkis's seminal work on binary classification (Vapnik and Chervonenkis, 1971), it was widely believed that excluding trivialities, if a problem is at all learnable, then uniform convergence holds, and the problem is also learnable by every Empirical Risk Minimizer (ERM learner). The equivalence between learnability and uniform convergence has been proved for binary classification and for regression problems (Kearns et al., 1994; Bartlett et al., 1996; Alon et al., 1997). Recently, Shalev-Shwartz et al. (2010) have shown that in the general setting of learning of Vapnik (1995), learnability is not equivalent to uniform convergence. Moreover, some learning problems are learnable, but not with every ERM. In particular, this was shown for an unsupervised learning problem in the class of stochastic convex learning problems. The conclusion in Shalev-Shwartz et al. (2010) is that the conditions for learnability in the general setting are significantly more complex than in supervised learning. In this work we show that even in multiclass learning, uniform convergence is not equivalent to learnability. We find this result surprising, since multiclass prediction is very similar to binary classification.

This result raises once more the question of determining the true sample complexity of multiclass learning, and the optimal learning algorithm in this setting. We provide conditions under which tight characterization of the sample complexity of a multiclass hypothesis class can be provided. Specifically, we consider the important case of hypothesis classes which are invariant to renaming of class labels. We term such classes *symmetric* hypothesis classes. We show that the sample complexity for symmetric classes is tightly characterized by a known combinatorial measure called the Natarajan dimension. We conjecture that this result holds for non-symmetric classes as well.

We further study multiclass sample complexity in other learning models. Overall, we consider the following categorization of learning models:

- Interaction with the data source (batch vs. online protocols): In the batch protocol, we assume that the training data is generated i.i.d. by some distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. The goal is to find, with a high probability over the training samples, a predictor h such that $\Pr_{(x,y)\sim\mathcal{D}}(h(x) \neq y)$ is as small as possible. In the online protocol we receive examples one by one, and are asked to predict the label of each given example on the fly. Our goal is to make as few prediction mistakes as possible in the worst case (see Littlestone 1987).

- The type of feedback (full information vs. bandits): In the full information setting, we receive the correct label of every example. In the bandit setting, the learner first sees an unlabeled example, and then outputs its prediction for the label. Then, a binary feedback is received, indicating only whether the prediction was correct or not, but not revealing the correct label in the case of a wrong guess (see for example Auer et al. 2003, 2002; Kakade et al. 2008).

The batch/full-information model is the standard PAC setting, while the online/full-information model is the usual online setting. The online/bandits model is the usual multiclass-bandits setting. We are not aware of a treatment of the batch/bandit model in previous works.

1.1 Paper Overview

After presenting formal definitions and notations in Section 2, we begin our investigation of multiclass sample complexity in the classical PAC learning setting. Previous results have provided upper and lower bounds on the sample complexity of multiclass learning in this setting when using any ERM algorithm. The lower bounds are controlled by the *Natarajan dimension*, a combinatorial measure which generalizes the VC dimension for the multiclass case, while the upper bounds are controlled by the *graph dimension*, which is another generalization of the VC dimension. The ratio between these two measures can be as large as $\Theta(\ln(k))$, where $k = |\mathcal{Y}|$ is the number of class labels. In Section 3 we survey known results, and also present a new improvement for the upper bound in the realizable case. All the bounds here are uniform, that is, they hold for all ERM learners.

These uniform bounds are the departure point of our research. Our goal is to find a combinatorial measure, similar to the VC-Dimension, that characterizes the sample complexity of a given class, up to logarithmic factors, *independent of the number of classes*. We delve into this challenge in Section 4. First, we show that no uniform bound on arbitrary ERM learners can tightly characterize the sample complexity: We describe a family of concept classes for which there exist 'good' ERM learners and 'bad' ERM learners, with a ratio of $\Theta(\ln(k))$ between their sample complexities. We further show that if k is infinite, then there are hypothesis classes that are learnable by some ERM learners but not by other ERM learners. Moreover, we show that for any hypothesis class, the sample complexity of the *worst* ERM learner in the realizable case is characterized by the graph dimension.

These results indicate that classical concepts which are commonly used to provide upper bounds for all ERM learners of some hypothesis class, such as the growth function, cannot lead to tight sample complexity characterization for the multiclass case. We thus propose algorithmic-dependent versions of these quantities, that allow bounding the sample complexity of specific ERM learners.

We consider three cases in which we show that the true sample complexity of multiclass learning in the PAC setting is fully characterized by the Natarajan dimension. The first case includes any ERM algorithm that does not use too many class labels, in a precise sense that we define via the new notion of *essential range* of an algorithm. In particular, the requirement is satisfied by any ERM learner which only predicts labels that appeared in the sample. The second case includes any ERM learner for symmetric hypothesis classes. The third case is the scenario where we have no prior knowledge on the different class labels, which we defined precisely in Section 4.3.

We conjecture that the upper bound obtained for symmetric classes holds for non-symmetric classes as well. Such a result cannot be implied by uniform convergence alone, since, by the results mentioned above, there always exist ERM learners with a sample complexity that is higher than this conjectured upper bound. It therefore follows that a proof of our conjecture will require the derivation of new learning rules. We hope that this would lead to new insights in other statistical learning problems as well.

In Section 5 we study multiclass learnability in the online model and in the bandit model. We introduce two generalizations of the Littlestone dimension, which characterize multiclass learnability in each of these models respectively. Our bounds are tight for the realizable case.

2. Problem Setting and Notation

Let $\mathcal{X}$ be a space, $\mathcal{Y}$ a discrete space[1] and $\mathcal{H}$ a class of functions from $\mathcal{X}$ to $\mathcal{Y}$. Denote $k = |\mathcal{Y}|$ (note that k can be infinite). For a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, the error of a function $f : \mathcal{X} \to \mathcal{Y}$ with respect to $\mathcal{D}$ is defined as $\mathrm{Err}(f) = \mathrm{Err}_{\mathcal{D}}(f) = \Pr_{(x,y)\sim\mathcal{D}}(f(x) \neq y)$. The best error achievable by $\mathcal{H}$ on $\mathcal{D}$, namely, $\mathrm{Err}_{\mathcal{D}}(\mathcal{H}) := \inf_{f\in\mathcal{H}} \mathrm{Err}_{\mathcal{D}}(f)$, is called the *approximation error* of $\mathcal{H}$ on $\mathcal{D}$.

In the PAC setting, a *learning algorithm* for a class $\mathcal{H}$ is a function, $\mathcal{A} : \cup_{n=0}^{\infty}(\mathcal{X}\times\mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}$. We denote a training sequence by $S_m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. An *ERM learner* for class $\mathcal{H}$ is a learning algorithm that for any sample S_m returns a function that minimizes the empirical error relative to any other function in $\mathcal{H}$. Formally, the empirical error of a function f on a sample S_m is

$$\underset{S_m}{\mathrm{Err}}(f) = \frac{1}{m}|\{i \in [m] : f(x_i) \neq y_i\}|.$$

A learning algorithm $\mathcal{A}$ of class $\mathcal{H}$ is an ERM learner if $\mathrm{Err}_{S_m}(\mathcal{A}(S_m)) = \min_{f\in\mathcal{H}} \mathrm{Err}_{S_m}(f)$.

The *agnostic sample complexity* of a learning algorithm $\mathcal{A}$ is the function $m^a_{\mathcal{A},\mathcal{H}}$ defined as follows: For every $\epsilon, \delta > 0$, $m^a_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$ is the minimal integer such that for every $m \geq m^a_{\mathcal{A},\mathcal{H}}(\epsilon, \delta)$ and every distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$,

$$\Pr_{S_m\sim\mathcal{D}^m}\left(\underset{\mathcal{D}}{\mathrm{Err}}(\mathcal{A}(S_m)) > \underset{\mathcal{D}}{\mathrm{Err}}(\mathcal{H}) + \epsilon\right) \leq \delta. \tag{1}$$

Here and in subsequent definitions, we omit the subscript $\mathcal{H}$ when it is clear from context. If there is no integer satisfying the inequality above, define $m^a_{\mathcal{A}}(\epsilon, \delta) = \infty$. $\mathcal{H}$ is learnable with $\mathcal{A}$ if for all ϵ and δ the agnostic sample complexity is finite. The agnostic sample complexity of a class $\mathcal{H}$ is

$$m^a_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) = \inf_{\mathcal{A}} m^a_{\mathcal{A},\mathcal{H}}(\epsilon, \delta) \ ,$$

where the infimum is taken over all learning algorithms for $\mathcal{H}$. The *agnostic ERM sample complexity* of $\mathcal{H}$ is the sample complexity that can be guaranteed for any ERM learner. It is defined by

$$m^a_{\mathrm{ERM},\mathcal{H}}(\epsilon, \delta) = \sup_{\mathcal{A}\in\mathrm{ERM}} m^a_{\mathcal{A},\mathcal{H}}(\epsilon, \delta) \ ,$$

where the supremum is taken over all ERM learners for $\mathcal{H}$. Note that always $m_{\mathrm{PAC}} \leq m_{\mathrm{ERM}}$.

We say that a distribution $\mathcal{D}$ is *realizable* by a hypothesis class $\mathcal{H}$ if there exists some $f \in \mathcal{H}$ such that $\mathrm{Err}_{\mathcal{D}}(f) = 0$. The *realizable sample complexity* of an algorithm $\mathcal{A}$ for a class $\mathcal{H}$, denoted $m^r_{\mathcal{A}}$, is the minimal integer such that for every $m \geq m^r_{\mathcal{A}}(\epsilon, \delta)$ and every distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ which is realizable by $\mathcal{H}$, Equation (1) holds. The realizable sample complexity of a class $\mathcal{H}$ is $m^r_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta) = \inf_{\mathcal{A}} m^r_{\mathcal{A}}(\epsilon, \delta)$, where the infimum is taken over all learning algorithms for $\mathcal{H}$. The realizable ERM sample complexity of a class $\mathcal{H}$ is $m^r_{\mathrm{ERM},\mathcal{H}}(\epsilon, \delta) = \sup_{\mathcal{A}\in\mathrm{ERM}} m^r_{\mathcal{A}}(\epsilon, \delta)$, where the supremum is taken over all ERM learners for $\mathcal{H}$.

Given a subset $S \subseteq \mathcal{X}$, we denote $\mathcal{H}|_S = \{f|_S : f \in \mathcal{H}\}$, where $f|_S$ is the restriction of f to S, namely, $f|_S : S \to \mathcal{Y}$ is such that for all $x \in S$, $f|_S(x) = f(x)$.

1. To avoid measurability issues, we assume that $\mathcal{X}$ and $\mathcal{Y}$ are countable.

3. Uniform Sample Complexity Bounds for ERM Learners

We first recall some known results regarding the sample complexity of multiclass learning. Recall the definition of the Vapnik-Chervonenkis dimension (Vapnik, 1995):

Definition 1 (VC dimension) *Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class. A subset $S \subseteq \mathcal{X}$ is* shattered *by $\mathcal{H}$ if $\mathcal{H}|_S = \{0,1\}^S$. The* VC-dimension *of $\mathcal{H}$, denoted $\mathrm{VC}(\mathcal{H})$, is the maximal cardinality of a subset $S \subseteq \mathcal{X}$ that is shattered by $\mathcal{H}$.*

The VC-dimension, a cornerstone in statistical learning theory, characterizes the sample complexity of learning *binary* hypothesis classes, as the following bounds suggest.

Theorem 2 (Vapnik, 1995 and Bartlett and Mendelson, 2002) *There are absolute constants $C_1, C_2 > 0$ such that for every $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$,*

$$C_1 \left(\frac{\mathrm{VC}(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon} \right) \leq m^r_{\mathrm{PAC}}(\epsilon, \delta) \leq m^r_{\mathrm{ERM}}(\epsilon, \delta) \leq C_2 \left(\frac{\mathrm{VC}(\mathcal{H}) \ln(\frac{1}{\epsilon}) + \ln(\frac{1}{\delta})}{\epsilon} \right),$$

and

$$C_1 \left(\frac{\mathrm{VC}(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2} \right) \leq m^a_{\mathrm{PAC}}(\epsilon, \delta) \leq m^a_{\mathrm{ERM}}(\epsilon, \delta) \leq C_2 \left(\frac{\mathrm{VC}(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2} \right).$$

One of the important implications of this result is that in binary classification, *all* ERM learners are as good, up to a multiplicative factor of $\ln(1/\epsilon)$.

It is natural to seek a generalization of the VC-dimension to hypothesis classes of non-binary functions. We recall two generalizations, both introduced by Natarajan (1989). In both generalizations, shattering of a set S is redefined by requiring that for any partition of S into T and $S \setminus T$, there exists a $g \in \mathcal{H}$ whose behavior on T differs from its behavior on $S \setminus T$. The two definitions are distinguished by their definition of "different behavior".

Definition 3 (Graph dimension) *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class and let $S \subseteq \mathcal{X}$. We say that $\mathcal{H}$* G-shatters *S if there exists an $f : S \to \mathcal{Y}$ such that for every $T \subseteq S$ there is a $g \in \mathcal{H}$ such that*

$$\forall x \in T,\ g(x) = f(x),\ \text{and}\ \forall x \in S \setminus T,\ g(x) \neq f(x).$$

The graph dimension *of $\mathcal{H}$, denoted $d_G(\mathcal{H})$, is the maximal cardinality of a set that is G-shattered by $\mathcal{H}$.*

Definition 4 (Natarajan dimension) *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class and let $S \subseteq \mathcal{X}$. We say that $\mathcal{H}$* N-shatters *S if there exist $f_1, f_2 : S \to \mathcal{Y}$ such that $\forall y \in S,\ f_1(y) \neq f_2(y)$, and for every $T \subseteq S$ there is a $g \in \mathcal{H}$ such that*

$$\forall x \in T,\ g(x) = f_1(x),\ \text{and}\ \forall x \in S \setminus T,\ g(x) = f_2(x).$$

The Natarajan dimension *of $\mathcal{H}$, denoted $d_N(\mathcal{H})$, is the maximal cardinality of a set that is N-shattered by $\mathcal{H}$.*

Both of these dimensions coincide with the VC-dimension for $k = 2$. Note also that we always have $d_N \leq d_G$. By reductions to and from the binary case, similarly to Natarajan (1989) and Ben-David et al. (1995) one can show the following result:

Theorem 5 *For the constants C_1, C_2 from Theorem 2, for every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ we have*

$$C_1 \left(\frac{d_N(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon} \right) \leq m^r_{\text{PAC}}(\epsilon, \delta) \leq m^r_{\text{ERM}}(\epsilon, \delta) \leq C_2 \left(\frac{d_G(\mathcal{H}) \ln(\frac{1}{\epsilon}) + \ln(\frac{1}{\delta})}{\epsilon} \right),$$

and

$$C_1 \left(\frac{d_N(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2} \right) \leq m^a_{\text{PAC}}(\epsilon, \delta) \leq m^a_{\text{ERM}}(\epsilon, \delta) \leq C_2 \left(\frac{d_G(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2} \right).$$

Proof (sketch) For the lower bound, let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class of Natarajan dimension d and Let $\mathcal{H}_d := \{0,1\}^{[d]}$. We claim that $m^r_{\text{PAC},\mathcal{H}_d} \leq m^r_{\text{PAC},\mathcal{H}}$, and similarly for the agnostic sample complexity, so the lower bounds are obtained by Theorem 2. Let $\mathcal{A}$ be a learning algorithm for $\mathcal{H}$. Consider the learning algorithm, $\bar{\mathcal{A}}$, for $\mathcal{H}_d$ defined as follows. Let $S = \{s_1, \ldots, s_d\} \subseteq X$ be a set and let f_0, f_1 be functions that witness the N-shattering of $\mathcal{H}$. Given a sample $((x_i, y_i))_{i=1}^m \subseteq [d] \times \{0,1\}$, let $g = \mathcal{A}((s_{x_i}, f_{y_i}(s_{x_i}))_{i=1}^m)$. $\bar{\mathcal{A}}$ returns $f : [d] \to \{0,1\}$ such that $f(i) = 1$ if and only if $g(s_i) = f_1(s_i)$. It is not hard to see that $m^r_{\bar{\mathcal{A}},\mathcal{H}_d} \leq m^r_{\mathcal{A},\mathcal{H}}$, thus $m^r_{\text{PAC},\mathcal{H}_d} \leq m^r_{\text{PAC},\mathcal{H}}$ and similarly for the agnostic case.

For the upper bound, let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class of graph dimension d. For every $f \in \mathcal{H}$ define $\bar{f} : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ by setting $\bar{f}(x, y) = 1$ if and only if $f(x) = y$ and let $\bar{\mathcal{H}} = \{\bar{f} : f \in \mathcal{H}\}$. It is not hard to see that $\text{VC}(\bar{\mathcal{H}}) = d_G(\mathcal{H})$. Let $\mathcal{A}$ be an ERM algorithm for $\mathcal{H}$. Let $\bar{\mathcal{A}}$ be an ERM algorithm for $\bar{\mathcal{H}}$ such that for a sample $(((x_i, z_i), y_i))_{i=1}^m \subseteq \mathcal{X} \times \mathcal{Y} \times \{0,1\}$, if for all i, $y_i = 1$, $\bar{\mathcal{A}}$ returns $\bar{f}$, where $f = \mathcal{A}((x_i, z_i)_{i=1}^m)$. It is easy to check that $\bar{\mathcal{A}}$ is consistent and therefore can be extended to an ERM learner for $\bar{\mathcal{H}}$, and that $m^r_{\mathcal{A},\mathcal{H}} \leq m^r_{\bar{\mathcal{A}},\bar{\mathcal{H}}}$. Thus $m^r_{\text{ERM},\mathcal{H}} \leq m^r_{\text{ERM},\bar{\mathcal{H}}}$. The analogous inequalities hold for the agnostic sample complexity as well. Thus the desired upper bounds follow from Theorem 2. ■

This theorem shows that the finiteness of the Natarajan dimension is a necessary condition for learnability, and the finiteness of the graph dimension is a sufficient condition for learnability. In Ben-David et al. (1995) it was proved that for every hypotheses class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$,

$$d_N(\mathcal{H}) \leq d_G(\mathcal{H}) \leq 4.67 \log_2(k) d_N(\mathcal{H}) \ . \tag{2}$$

It follows that if $k < \infty$ then the finiteness of the Natarajan dimension is both a necessary and a sufficient condition for learnability.[2] Incorporating Equation (2) into Theorem 5, it can be seen that the Natarajan dimension, as well as the graph dimension, characterize the sample complexity of $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ up to a multiplicative factor of $O(\ln(k) \ln(\frac{1}{\epsilon}))$. Precisely, the following result can be derived:

2. The result of Ben-David et al. (1995) in fact holds also for a rich family of generalizations of the VC dimension, of which the Graph dimension is one example.

Theorem 6 *There are constants C_1, C_2 such that, for every $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$,*

$$C_1\left(\frac{d_N(\mathcal{H})+\ln(\frac{1}{\delta})}{\epsilon}\right) \leq m^r_{\text{PAC}}(\epsilon,\delta) \leq m^r_{\text{ERM}}(\epsilon,\delta) \leq C_2\left(\frac{d_N(\mathcal{H})\ln(k)\cdot\ln(\frac{1}{\epsilon})+\ln(\frac{1}{\delta})}{\epsilon}\right),$$

and

$$C_1\left(\frac{d_N(\mathcal{H})+\ln(\frac{1}{\delta})}{\epsilon^2}\right) \leq m^a_{\text{PAC}}(\epsilon,\delta) \leq m^a_{\text{ERM}}(\epsilon,\delta) \leq C_2\left(\frac{d_N(\mathcal{H})\ln(k)+\ln(\frac{1}{\delta})}{\epsilon^2}\right).$$

3.1 An Improved Upper Bound for the Realizable Case

The following theorem provides a sample complexity upper bound which provides a tighter dependence on ϵ.

Theorem 7 *For every concept class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$,*

$$m^r_{\text{ERM}}(\epsilon,\delta) = O\left(\frac{d_N(\mathcal{H})\left(\ln(\frac{1}{\epsilon})+\ln(k)+\ln(d_N(\mathcal{H}))\right)+\ln(\frac{1}{\delta})}{\epsilon}\right).$$

The proof of this theorem is immediate given Theorem 13, which is provided in Section 4. We give the short proof of this theorem thereafter. While a proof for the Theorem can be established by a simple adaptation of previous techniques, we find it valuable to present this result here, as we could not find it in the literature.

4. PAC Sample Complexity with ERM Learners

In this section we study the sample complexity of multiclass ERM learners in the PAC setting. First, we show that unlike the binary case, in the multiclass setting different ERM learners can have very different sample complexities.

Example 1 (A Large Gap Between ERM Learners) *Let $\mathcal{X}$ be any finite or countable domain set. Let $\mathcal{P}_f(\mathcal{X})$ denote the collection of finite and co-finite subsets $A \subseteq \mathcal{X}$. We will take the label space to be $\mathcal{P}_f(\mathcal{X})$ together with a special label, denoted by $*$ (I.e. $\mathcal{Y} = \mathcal{P}_f(\mathcal{X}) \cup \{*\}$). For every $A \in \mathcal{P}_f(\mathcal{X})$, define $f_A : \mathcal{X} \to \mathcal{Y}$ by*

$$f_A(x) = \begin{cases} A & \text{if } x \in A \\ * & \text{otherwise,} \end{cases}$$

and consider the hypothesis class $\mathcal{H}_{\mathcal{X}} = \{f_A : A \in \mathcal{P}_f(\mathcal{X})\}$. It is not hard to see that $d_N(\mathcal{H}_{\mathcal{X}}) = 1$. On the other hand, if $\mathcal{X}$ is finite then $\mathcal{X}$ is G-shattered using the function $f_\emptyset$, therefore $d_G(\mathcal{H}_{\mathcal{X}}) = |\mathcal{X}|$. If $\mathcal{X}$ is infinite, then every finite subset of $\mathcal{X}$ is G-shattered, thus $d_G(\mathcal{H}_{\mathcal{X}}) = \infty$.

Consider two ERM algorithms for $\mathcal{H}_{\mathcal{X}}$, $\mathcal{A}_{\text{bad}}$ and $\mathcal{A}_{\text{good}}$, which satisfy the following properties. For $\mathcal{A}_{\text{bad}}$, whenever a sample of the form $S_m = \{(x_1,*),\ldots,(x_m,*)\}$ is observed, $\mathcal{A}_{\text{bad}}$ returns $f_{\{x_1,\ldots,x_m\}^c}$. Intuitively, while $\mathcal{A}_{\text{bad}}$ selects a hypothesis that minimizes the

empirical error, its choice for S_m seems to be sub-optimal. We will show later, based on Theorem 9, that the sample complexity of $\mathcal{A}_{\text{bad}}$ is $\Omega\left(\frac{|\mathcal{X}|+\ln(\frac{1}{\delta})}{\epsilon}\right)$.

For $\mathcal{A}_{\text{good}}$, we require that the algorithm only ever returns either $f_\emptyset$, or a hypothesis A such that the label A appeared in the sample—One can easily verify that there exists an ERM algorithm that satisfies this condition. Specifically, this means that for the sample $S_m = \{(x_1, *), \ldots, (x_m, *)\}$, $\mathcal{A}_{\text{good}}$ necessarily returns $f_\emptyset$. We have the following guarantee for $\mathcal{A}_{\text{good}}$:

Claim 1 *$m^r_{\mathcal{A}_{\text{good}},\mathcal{H}_\mathcal{X}}(\epsilon,\delta) \leq \frac{1}{\epsilon}\ln\frac{1}{\delta}$, and $m^a_{\mathcal{A}_{\text{good}},\mathcal{H}_\mathcal{X}}(\epsilon,\delta) \leq \frac{1}{\epsilon^2}\ln(\frac{1}{\epsilon})\ln\frac{1}{\delta}$.*

Proof *We prove the bound for the realizable case. The bound for the agnostic case will be immediate using Cor. 15, which we prove later.*

Let $\mathcal{D}$ be a distribution over $\mathcal{X}\times\mathcal{Y}$ and suppose that the correct labeling for $\mathcal{D}$ is f_A. Let m be the size of the sample. For any sample, $\mathcal{A}_{\text{good}}$ returns either $f_\emptyset$ or f_A. If it returns f_A then its error on $\mathcal{D}$ is zero. On the other hand, $\text{Err}_\mathcal{D}(f_\emptyset) = \Pr_{(X,Y)\sim\mathcal{D}}(X \in A)$. Thus, $\mathcal{A}_{\text{good}}$ returns a hypothesis with error ϵ or more only if $\Pr_{(X,Y)\sim\mathcal{D}}(X \in A) \geq \epsilon$ and all the m examples in the sample are from A^c. Assume $m \geq \frac{1}{\epsilon}\ln(\frac{1}{\delta})$, then the probability of the latter event is $(P(A^c))^m \leq (1-\epsilon)^m \leq e^{-\epsilon m} \leq \delta$. ■

This example shows that the gap between two different ERM learners can be as large as the gap between the Natarajan dimension and the graph dimension. By considering $\mathcal{H}_\mathcal{X}$ with an infinite $\mathcal{X}$, we conclude the following corollary.

Corollary 8 *There exist sets $\mathcal{X}$, $\mathcal{Y}$ and a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^\mathcal{X}$, such that $\mathcal{H}$ is learnable by some ERM learner but is not learnable by some other ERM learner.*

In Example 1, the bad ERM indeed requires as many examples as the graph dimension, while the good ERM requires only as many as the Natarajan dimension. Do such a 'bad' ERM and a 'good' ERM always exist? Our next result answers the question for the 'bad' ERM in the affirmative. Indeed, the graph dimension determines the learnability of $\mathcal{H}$ using the *worst* ERM learner.

Theorem 9 *There are constants $C_1, C_2 > 0$ such that the following holds. For every hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^\mathcal{X}$ of Graph dimension ≥ 2, there exists an ERM learner $\mathcal{A}_{\text{bad}}$ such that for every $\epsilon < \frac{1}{12}$ and $\delta < \frac{1}{100}$,*

$$C_1\left(\frac{d_G(\mathcal{H})+\ln(\frac{1}{\delta})}{\epsilon}\right) \leq m^r_{\mathcal{A}_{\text{bad}}}(\epsilon,\delta) \leq m^r_{\text{ERM}}(\epsilon,\delta) \leq C_2\left(\frac{d_G(\mathcal{H})\ln(\frac{1}{\epsilon})+\ln(\frac{1}{\delta})}{\epsilon}\right).$$

Proof The upper bound is simply a restatement of Theorem 5. It remains to prove that there exists an ERM learner, $\mathcal{A}_{\text{bad}}$, with $m^r_{\mathcal{A}_{bad}}(\epsilon,\delta) \geq C_1\left(\frac{d_G(\mathcal{H})+\ln(\frac{1}{\delta})}{\epsilon}\right)$.

First, assume that $d = d_G(\mathcal{H}) < \infty$. Let $S = \{x_0, \ldots, x_{d-1}\} \subseteq \mathcal{X}$ be a set which is G-Shattered by $\mathcal{H}$ using the function f_0. Let $\mathcal{A}_{\text{bad}}$ be an ERM learner with the following property. Upon seeing a sample $T \subseteq S$ which is consistent with f_0, $\mathcal{A}_{\text{bad}}$ returns a function

that coincides with f_0 on T and disagrees with f_0 on $S \setminus T$. Such a function exists since S is G-shattered using f_0.

Fix $\delta < \frac{1}{100}$ and $\epsilon < \frac{1}{12}$. Note that $1 - 2\epsilon \geq e^{-4\epsilon}$. Define a distribution on $\mathcal{X}$ by setting $\Pr(x_0) = 1 - 2\epsilon$ and for all $1 \leq i \leq d-1$, $\Pr(x_i) = \frac{2\epsilon}{d-1}$. Suppose that the correct hypothesis is f_0 and let $\{(X_i, f_0(X_i))\}_{i=1}^m$ be a sample. Clearly, the hypothesis returned by $\mathcal{A}_{\text{bad}}$ will err on all the examples from S which are not in the sample. By Chernoff's bound, if $m \leq \frac{d-1}{6\epsilon}$, then with probability at least $\frac{1}{100} \geq \delta$, the sample will include no more than $\frac{d-1}{2}$ examples from $S \setminus \{x_0\}$, so that the returned hypothesis will have error at least ϵ. To see that, define r.v. Y_i, $1 \leq i \leq m$ by setting $Y_i = 1$ if $X_i \neq x_0$ and 0 otherwise. By Chernoff's bound, if $r = \lfloor \frac{d-1}{6\epsilon} \rfloor$ then

$$\Pr\left(\sum_{i=1}^{m} Y_i \geq \frac{d-1}{2}\right) \leq \Pr\left(\sum_{i=1}^{r} Y_i \geq 3\epsilon k\right) \leq \exp\left(-\frac{\frac{1}{2}^2}{3} 2\epsilon r\right) < 0.99 \ .$$

Moreover, the probability that the sample includes only x_0 (and thus $\mathcal{A}_{\text{bad}}$ will return a hypothesis with error 2ϵ) is $(1-2\epsilon)^m \geq e^{-4\epsilon m}$, which is more than δ if $m \leq \frac{1}{4\epsilon}\ln(\frac{1}{\delta})$. We therefore obtain that

$$m^r_{\mathcal{A}_{\text{bad}}}(\epsilon, \delta) \geq \max\left\{\frac{d-1}{6\epsilon}, \frac{1}{2\epsilon}\ln(1/\delta)\right\} \geq \frac{d-1}{12\epsilon} + \frac{1}{4\epsilon}\ln(1/\delta) \ ,$$

as required.

If $d_G(\mathcal{H}) = \infty$, let S_n, $n = 2, 3, \ldots$ be a sequence of pairwise disjoint shattered sets such that $|S_n| = n$. For every n, suppose that f_0^n indicated that S_n is G-shattered. Let $\mathcal{A}_{\text{bad}}$ be an ERM learner with the following property. Upon seeing a sample $T \subseteq S_n$ labeled by f_0^n, $\mathcal{A}_{\text{bad}}$ returns a function that coincides with f_0^n on T and disagrees with f_0 on $S_n \setminus T$. Repeating the argument of the finite case for S_n instead of S shows that for every $\epsilon < \frac{1}{12}$ and $\delta < \frac{1}{100}$ it holds that $m_{\mathcal{A}_{\text{bad}}}(\epsilon, \delta) \geq C_1\left(\frac{n+\ln(\frac{1}{\delta})}{\epsilon}\right)$. Since it holds for every n, we conclude that $m^r_{\mathcal{A}_{\text{bad}}}(\epsilon, \delta) = \infty$. ■

To get the sample complexity lower bound for the ERM learner $\mathcal{A}_{\text{bad}}$ in Example 1, observe that this algorithm satisfies the specifications of a bad ERM algorithm from the proof above.

We conclude that for any multiclass learning problem there exists a 'bad' ERM learner. The existence of 'good' ERM learners turns out to be a more involved question. We conjecture that for every class there exists a 'good' ERM learner – that is, a learning algorithm whose realizable sample complexity is $\tilde{O}\left(\frac{d_N}{\epsilon}\right)$ (where the $\tilde{O}$ notation may hide poly-logarithmic factors of $\frac{1}{\epsilon}, d_N$ and $1/\delta$ but *not* of $|Y|$). As we describe in the rest of this section, in this work we prove this conjecture for several families of hypothesis classes.

What is the crucial feature that makes $\mathcal{A}_{\text{good}}$ better than $\mathcal{A}_{\text{bad}}$ in Example 1? For the realizable case, if the correct labeling is $f_A \in \mathcal{H}_{\mathcal{X}}$, then for *any* sample, $\mathcal{A}_{\text{good}}$ would return only one of at most two functions: either f_A or $f_\emptyset$. On the other hand, if the correct labeling is $f_\emptyset$, then $\mathcal{A}_{\text{bad}}$ might return *every* function in $\mathcal{H}_{\mathcal{X}}$. Thus, to return a hypothesis with error at most ϵ, $\mathcal{A}_{\text{good}}$ needs to reject at most one hypothesis, while $\mathcal{A}_{\text{bad}}$ might need to reject many more. Following this intuition, we propose the following rough principle: *A good ERM learner is one that, for every target hypothesis, considers a small number of hypotheses.*

We would like to use this intuition to design ERMs with a better sample complexity than the one that can be guaranteed for a general ERM as in Theorem 7. Classical sample complexity upper bounds that hold for all ERM learners hinge on the notion of a *growth function*, which counts the number of different hypotheses induced by the hypothesis class on a sample of a certain size. To bound the sample complexity of a specific ERM learner, we define algorithm-dependent variants of the concept of a growth function.

Definition 10 (Algorithm-dependent growth function) *Fix a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. Let $\mathcal{A}$ be a learning algorithm for $\mathcal{H}$. For $m > 0$ and a sample $S = ((x_i, y_i))_{i=1}^{2m}$ of size $2m$, let $\mathcal{X}_S = \{x_1, \ldots, x_{2m}\}$, and define*

$$F_{\mathcal{A}}(S) = \{\mathcal{A}(S')|_{\mathcal{X}_S} \mid S' \subseteq S,\ |S'| = m\}.$$

Let $R(\mathcal{H})$ be the set of samples which are consistent with $\mathcal{H}$, that is $S = ((x_i, f(x_i)))_{i=1}^{2m}$ for some $f \in \mathcal{H}$. Define the realizable algorithm-dependent growth function *of $\mathcal{A}$ by*

$$\Pi^r_{\mathcal{A}}(m) = \sup_{S \in R(\mathcal{H}), |S|=2m} |F_{\mathcal{A}}(S)|.$$

Define the agnostic algorithm-dependent growth function *of $\mathcal{A}$ for sample S by*

$$\Pi^a_{\mathcal{A}}(m) = \sup_{S \in (\mathcal{X}\times\mathcal{Y})^{2m}} |F_{\mathcal{A}}(S)|.$$

These definitions enable the use of a 'double sampling' argument, similarly to the one used with the classical growth function (see Anthony and Bartlett, 1999, chapter 4). This argument is captured by the following lemma.

Lemma 11 (The Double Sampling Lemma) *Let $\mathcal{A}$ be an ERM learner, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Denote $\epsilon = \mathrm{Err}_{\mathcal{D}}(\mathcal{A}(S_m)) - \mathrm{Err}_{\mathcal{D}}(\mathcal{H})$, and let $\delta \in (0, 1)$.*

1. *If $\mathcal{D}$ is realizable by $\mathcal{H}$ then with probability at least $1 - \delta$,*

$$\epsilon \leq 12 \ln(2\Pi^r_{\mathcal{A}}(m)/\delta)/m.$$

2. *For any $\mathcal{D}$, with probability at least $1 - \delta$,*

$$\epsilon \leq \sqrt{\frac{32 \ln((4\Pi^a_{\mathcal{A}}(m) + 4)/\delta)}{m}}.$$

Proof The proof idea of the this lemma is similar to the one of the 'double sampling' results of Anthony and Bartlett (1999) (see their Theorems 4.3 and 4.8).

For the first part of the claim, let $\mathcal{D}$ be a realizable distribution for $\mathcal{H}$. For $m \leq 8$, the claim trivially holds, therefore assume $m \geq 8$. Let $\nu = 12 \ln(2\Pi^r_{\mathcal{A}}(m)/\delta)/m$ and assume w.l.o.g. that $\nu \leq 1$.

Suppose that for some $S \in (\mathcal{X} \times \mathcal{Y})^m$, $\mathrm{Err}_{\mathcal{D}}(\mathcal{A}(S)) \geq \nu$. Let $T \in (\mathcal{X} \times \mathcal{Y})^m$ be another sample drawn from D^m, independently from S. We show that $Err_T(\mathcal{A}(S)) \geq \nu/2$ with probability at least $\frac{1}{2}$. For $\nu \leq \frac{1}{2}$, by Chernoff's bound, this holds with probability at least $1 - \exp(-m\nu/16)$, which is larger than $\frac{1}{2}$ by the definition of ν. For $\nu \geq \frac{1}{2}$, by Hoeffding's

inequality, this holds with probability at least $1-\exp(-m\nu^2/2) \geq 1-\exp(-m/8)$, which is larger than $\frac{1}{2}$, since $m \geq 8$. It follows that

$$\frac{1}{2} \Pr_{S\sim\mathcal{D}^m}(\operatorname*{Err}_{\mathcal{D}}(\mathcal{A}(S)) \geq \nu) \leq \Pr_{(S,T)\sim\mathcal{D}^{2m}}(\operatorname*{Err}_{T}(\mathcal{A}(S)) \geq \nu/2). \tag{3}$$

Let $Z = (z_1, \ldots, z_{2m}) \in R(\mathcal{H})$, and let $\sigma : [2m] \to [2m]$ be a permutation. We write Z^1_σ to mean $(z_{\sigma(1)}, \ldots, z_{\sigma(m)})$ and Z^2_σ to mean $(z_{\sigma(m+1)}, \ldots, z_{\sigma(2m)})$.

Similarly to Lemma 4.5 in Anthony and Bartlett (1999), for σ drawn uniformly from the set of permutations,

$$\begin{aligned}\Pr_{(S,T)\in\mathcal{D}^{2m}}(\operatorname*{Err}_{T}(\mathcal{A}(S)) \geq \nu/2) &= \mathop{\mathbb{E}}_{Z\sim\mathcal{D}^{2m}}(\Pr_{\sigma}(\operatorname*{Err}_{Z^2_\sigma}(\mathcal{A}(Z^1_\sigma)) \geq \nu/2)) \\ &\leq \sup_{Z\in R(\mathcal{H}),|Z|=2m} \Pr_{\sigma}(\operatorname*{Err}_{Z^2_\sigma}(\mathcal{A}(Z^1_\sigma)) \geq \nu/2).\end{aligned} \tag{4}$$

To bound the right hand side, note that since $\mathcal{A}$ is an ERM algorithm, for any fixed $Z \in R(\mathcal{H})$ and any σ, $\operatorname{Err}_{Z^1_\sigma}(\mathcal{A}(Z^1_\sigma)) = 0$. Thus

$$\Pr_{\sigma}(\operatorname*{Err}_{Z^2_\sigma}(\mathcal{A}(Z^1_\sigma)) \geq \nu/2) \leq \Pr_{\sigma}(\exists h \in F_{\mathcal{A}}(Z),\ \operatorname*{Err}_{Z^1_\sigma}(h) = 0 \text{ and } \operatorname*{Err}_{Z^2_\sigma}(h) \geq \nu/2).$$

For any fixed h, if the right hand side is not zero, then there exist at least $\nu m/2$ elements (x, y) in Z such that $h(x) \neq y$. In the latter case, the probability (over σ) that all such elements are in Z^2_σ is at most $2^{-\nu m/2}$. With a union bound over $h \in F_{\mathcal{A}}(Z)$, we conclude that for any Z,

$$\Pr_{\sigma}(\operatorname*{Err}_{Z^2_\sigma}(\mathcal{A}(Z^1_\sigma)) \geq \nu/2) \leq |F_{\mathcal{A}}(Z)|2^{-\nu m/2}.$$

Combining with Equation (4) gives

$$\Pr_{(S,T)\in\mathcal{D}^{2m}}(\operatorname*{Err}_{T}(\mathcal{A}(S)) \geq \nu/2) \leq \sup_{Z\in R(\mathcal{H})} |F_{\mathcal{A}}(Z)|2^{-\nu m/2} = \Pi^r_{\mathcal{A}}(m)2^{-\nu m/2}.$$

By Equation (3) and the definition of ν,

$$\Pr_{S\sim\mathcal{D}^m}(\operatorname*{Err}_{\mathcal{D}}(\mathcal{A}(S)) \geq \nu) \leq 2\Pi^r_{\mathcal{A}}(m)2^{-\nu m/2} \leq \delta.$$

This proves the first part of the claim.

For the second part of the claim, let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$. Denote $\epsilon^* = \operatorname{Err}_{\mathcal{D}}(\mathcal{H})$, and let $h^* \in \mathcal{H}$ such that $\operatorname{Err}_{\mathcal{D}}(h^*) = \epsilon^*$.

Let $\nu = \sqrt{\frac{32\ln((4\Pi^a_{\mathcal{A}}(m)+4)/\delta)}{m}}$. Suppose that for some $S \in (\mathcal{X} \times \mathcal{Y})^m$, $\operatorname{Err}_{\mathcal{D}}(\mathcal{A}(S)) \geq \epsilon^* + \nu$. Let $T \in (\mathcal{X} \times \mathcal{Y})^m$ be a random sample drawn from D^m independently from S. By Hoeffding's inequality, with probability at least $1-\exp(-m\nu^2/2)$, which is at least $\frac{1}{2}$ by the definition of ν^2, $\operatorname{Err}_T(\mathcal{A}(S)) \geq \epsilon^* + \nu/2$. It follows that

$$\frac{1}{2} \Pr_{S\sim\mathcal{D}^m}(\operatorname*{Err}_{\mathcal{D}}(\mathcal{A}(S)) \geq \epsilon^* + \nu) \leq \Pr_{(S,T)\sim\mathcal{D}^{2m}}(\operatorname*{Err}_{T}(\mathcal{A}(S)) \geq \epsilon^* + \nu/2). \tag{5}$$

Let $Z = (z_1, \ldots, z_{2m}) \in (\mathcal{X} \times \mathcal{Y})^{2m}$, and let $\sigma : [2m] \to [2m]$ be a permutation. Denote Z^1_σ and Z^2_σ as above.

Denote $\mathcal{Z} = \{Z \in (\mathcal{X} \times \mathcal{Y})^{2m} \mid \mathrm{Err}_Z(\mathcal{A}(Z^1_\sigma)) \leq \epsilon^* + \nu/8\}$. By lemma 4.5 in Anthony and Bartlett (1999) again, for σ drawn uniformly from the set of permutations,

$$\begin{aligned}\Pr_{(S,T)\in\mathcal{D}^{2m}}(\underset{T}{\mathrm{Err}}(\mathcal{A}(S)) \geq \epsilon^* + \nu/2) &= \mathop{\mathbb{E}}_{Z\sim\mathcal{D}^{2m}}(\Pr_\sigma(\underset{Z^2_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) \geq \epsilon^* + \nu/2)) \qquad (6)\\ &\leq \mathop{\mathbb{E}}_{Z\sim\mathcal{D}^{2m}}\left(\Pr_\sigma(\underset{Z^2_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) \geq \epsilon^* + \nu/2)\Big| Z \in \mathcal{Z}\right) + \Pr(Z \notin \mathcal{Z}).\end{aligned}$$

To bound the right hand side, first note that by Hoeffding's inequality, the second term is bounded by

$$\Pr(Z \notin \mathcal{Z}) \leq \exp(-\nu^2 m/16). \qquad (7)$$

For the first term, $\mathrm{Err}_{Z^2_\sigma}(\mathcal{A}(Z^1_\sigma)) \geq \epsilon^* + \nu/2$ implies that unless $\mathrm{Err}_{Z^1_\sigma}(\mathcal{A}(Z^1_\sigma)) > \epsilon^* + \nu/4$, necessarily $\mathrm{Err}_{Z^2_\sigma}(\mathcal{A}(Z^1_\sigma)) - \mathrm{Err}_{Z^1_\sigma}(\mathcal{A}(Z^1_\sigma)) \geq \nu/4$. Since $\mathcal{A}$ is an ERM algorithm, $\mathrm{Err}_{Z^1_\sigma}(\mathcal{A}(Z^1_\sigma)) > \epsilon^* + \nu/4$ only if also $\mathrm{Err}_{Z^1_\sigma}(h^*) > \epsilon^* + \nu/4$. Therefore, for any Z,

$$\begin{aligned}&\Pr_\sigma(\underset{Z^2_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) \geq \epsilon^* + \nu/2) \leq \\ &\quad \Pr_\sigma(\underset{Z^1_\sigma}{\mathrm{Err}}(h^*) > \epsilon^* + \nu/4) + \Pr_\sigma(\underset{Z^2_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) - \underset{Z^1_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) > \nu/4). \qquad (8)\end{aligned}$$

$\mathrm{Err}_{Z^1_\sigma}(h^*)$ is an average of m random variables of the form $\mathbb{I}[h^*(x_i) \neq y_i]$, that are sampled without replacement from the finite population Z, with population average $\mathrm{Err}_Z(h^*)$. For $Z \in \mathcal{Z}$, $\mathrm{Err}_Z(h^*) \leq \epsilon^* + \nu/8$. Therefore, by Hoeffding's inequality for sampling without replacements from a finite population (Hoeffding, 1963), for $Z \in \mathcal{Z}$,

$$\Pr_\sigma(\underset{Z^1_\sigma}{\mathrm{Err}}(h^*) > \epsilon^* + \nu/4) \leq \Pr_\sigma(\underset{Z^1_\sigma}{\mathrm{Err}}(h^*) - \underset{Z}{\mathrm{Err}}(h^*) > \nu/8) \leq \exp(-\nu^2 m/32). \qquad (9)$$

In addition, by the same inequality, and applying the union bound over $h \in F_\mathcal{A}(Z)$, for any Z

$$\begin{aligned}\Pr_\sigma(\underset{Z^2_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) - \underset{Z^1_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) > \nu/4) &\leq \Pr_\sigma(\exists h \in F_\mathcal{A}(Z), \underset{Z^2_\sigma}{\mathrm{Err}}(h) - \underset{Z^1_\sigma}{\mathrm{Err}}(h) > \nu/4)\\ &\leq \Pr_\sigma(\exists h \in F_\mathcal{A}(Z), \underset{Z^2_\sigma}{\mathrm{Err}}(h) - \underset{Z}{\mathrm{Err}}(h) > \nu/8) + \Pr_\sigma(\exists h \in F_\mathcal{A}(Z), \underset{Z^1_\sigma}{\mathrm{Err}}(h) - \underset{Z}{\mathrm{Err}}(h) > \nu/8)\\ &\leq 2\Pi^a_\mathcal{A}(m)\exp(-\nu^2 m/32). \qquad (10)\end{aligned}$$

Combined with Equation (8) and Equation (9), it follows that for $Z \in \mathcal{Z}$,

$$\Pr_\sigma(\underset{Z^2_\sigma}{\mathrm{Err}}(\mathcal{A}(Z^1_\sigma)) \geq \epsilon^* + \nu/2) \leq (2\Pi^a_\mathcal{A}(m) + 1)\exp(-\nu^2 m/32).$$

With Equation (5), Equation (6), and Equation (7), we conclude that

$$\Pr_{S\sim\mathcal{D}^m}(\underset{\mathcal{D}}{\mathrm{Err}}(\mathcal{A}(S)) \geq \epsilon^* + \nu) \leq (4\Pi^a_\mathcal{A}(m) + 4)\exp(-\nu^2 m/32) \equiv \delta.$$

The claim follows since $\epsilon = \mathrm{Err}_\mathcal{D}(\mathcal{A}(S)) - \epsilon^*$. ■

As we shall presently see, Lemma 11 can be used to provide better sample complexity bounds for some 'good' ERM learners.

4.1 Learning with a Small Essential Range

A key tool that we will use for providing better bounds is the notion of *essential range*, defined below. The essential range of an algorithm quantifies the number of different labels that can be emitted by the functions the algorithm might return for samples of a given size. In this definition we use the notion of the range of a function. Formally, for a function $f : \mathcal{X} \to \mathcal{Y}$, its range is the set of labels to which it maps $\mathcal{X}$, denoted by $\text{range}(f) = \{f(x) \mid x \in \mathcal{X}\}$.

Definition 12 (Essential range) *Let $\mathcal{A}$ be a learning algorithm for $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. The* realizable essential range *of $\mathcal{A}$ is the function $r^r_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$, defined as follows.*

$$r^r_{\mathcal{A}}(m) = \sup_{S \in R(\mathcal{H}), |S|=2m} \left| \cup_{S' \subset S,\, |S'|=m} \text{range}(\mathcal{A}(S')) \right| .$$

The agnostic essential range *of $\mathcal{A}$ is the function $r^a_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$, defined as follows.*

$$r^a_{\mathcal{A}}(m) = \sup_{S \subseteq \mathcal{X} \times \mathcal{Y}, |S|=2m} \left| \cup_{S' \subset S,\, |S'|=m} \text{range}(\mathcal{A}(S')) \right| .$$

Intuitively, an algorithm with a small essential range uses a smaller set of labels for any particular distribution, thus it enjoys better convergence guarantees. This is formally quantified in the following result.

Theorem 13 *Let $\mathcal{A}$ be an ERM learning algorithm for $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ with essential ranges $r^r_{\mathcal{A}}(m)$ and $r^a_{\mathcal{A}}(m)$. Denote $\epsilon = \text{Err}_{\mathcal{D}}(\mathcal{A}(S_m)) - \text{Err}_{\mathcal{D}}(\mathcal{H})$. Then,*

- *If $\mathcal{D}$ is realizable by $\mathcal{H}$ and $\delta < 0.1$ then with probability at least $1 - \delta$,*

$$\epsilon \leq O\left(\frac{d_N(\mathcal{H})(\ln(m) + \ln(r^r_{\mathcal{A}}(m))) + \ln(1/\delta)}{m} \right) .$$

- *For any probability distribution D, with probability at least $1 - \delta$,*

$$\epsilon \leq O\left(\sqrt{\frac{d_N(\mathcal{H})(\ln(m) + \ln(r^a_{\mathcal{A}}(m)) + \ln(1/\delta)}{m}} \right) .$$

To prove the realizable part of this theorem, we use the following combinatorial lemma by Natarajan:

Lemma 14 *(Natarajan, 1989) For every hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$, $|\mathcal{H}| \leq |\mathcal{X}|^{d_N(\mathcal{H})} |\mathcal{Y}|^{2d_N(\mathcal{H})}$.*

Proof [of Theorem 13] For the realizable sample complexity, the growth function can be bounded as follows. Let $S \in R(\mathcal{H})$ such that $|S| = 2m$, and consider the function class $F_{\mathcal{A}}(S)$ (see Definition 10). By definition, the domain of $F_{\mathcal{A}}(S)$ is $\mathcal{X}_S$ of size $2m$, and the range of $F_{\mathcal{A}}(S)$ is of size at most $r^r_{\mathcal{A}}(m)$. Lastly, the Natarajan dimension of $F_{\mathcal{A}}(S)$ is at most $d_N(\mathcal{H})$, since $F_{\mathcal{A}}(S) \subseteq \mathcal{H}|_S$.

Therefore, by Lemma 14, $|F_{\mathcal{A}}(S)| \leq (2m)^{d_N(\mathcal{H})} r^r_{\mathcal{A}}(m)^{2d_N(\mathcal{H})}$. Taking the supremum over all such S, we get

$$\Pi^r_{\mathcal{A}}(m) \leq (2m)^{d_N(\mathcal{H})} r^r_{\mathcal{A}}(m)^{2d_N(\mathcal{H})}.$$

The bound on ϵ follows from the first part of Lemma 11.

For the agnostic sample complexity, a similar argument shows that

$$\Pi^a_{\mathcal{A}}(m) \leq (2m)^{d_N(\mathcal{H})} r^a_{\mathcal{A}}(m)^{2d_N(\mathcal{H})},$$

and the bound on ϵ follows from the second part of Lemma 11. ■

Theorem 7, which provides an improved bound for the realizable case, now follows from the fact that the essential range is never more than k. But the essential range can also be much smaller than k. For example, the essential range of the algorithm from Example 1 is bounded by $2m+1$ (the $2m$ labels appearing in the sample together with the $*$ label). In fact, we can state a more general bound, for any algorithm which never 'invents' labels it did not observe in the sample.

Corollary 15 *Let $\mathcal{A}$ be an ERM learner for a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. Suppose that for every sample S, the function $\mathcal{A}(S)$ never outputs labels which have not appeared in S. Then*

$$m^r_{\mathcal{A}}(\epsilon,\delta) = O\left(\frac{d_N(\mathcal{H})(\ln(\frac{1}{\epsilon}) + \ln(d_N(\mathcal{H}))) + \ln(\frac{1}{\delta})}{\epsilon}\right),$$

and

$$m^a_{\mathcal{A}}(\epsilon,\delta) = O\left(\frac{d_N(\mathcal{H})(\ln(\frac{1}{\epsilon}) + \ln(d_N(\mathcal{H}))) + \ln(\frac{1}{\delta})}{\epsilon^2}\right).$$

This corollary is immediate from Theorem 13 by setting $r^r_{\mathcal{A}}(m) = r^a_{\mathcal{A}}(m) = 2m$.

From this corollary, we immediately get that every hypothesis class which admits such algorithms, and has a large gap between the Natarajan dimension and the graph dimension realizes a gap between the sample complexities of different ERM learners. Indeed, the graph dimension can even be unbounded, while the Natarajan dimension is finite and the problem is learnable. This is demonstrated by the following example.

Example 2 *Denote the ball in $\mathbb{R}^n$ with center z and radius r by $B_n(z,r) = \{x \mid \|x-z\| \leq r\}$. For a given ball $B = B_n(z,r)$ with $z \in \mathbb{R}^n$ and $r > 0$, let $h_B : \mathbb{R}^n \to \mathbb{R}^n \cup \{*\}$ be the function defined by $h_B(x) = z$ if $x \in B$ and $h_B(x) = *$ otherwise. Let h_* be a hypothesis that always returns $*$. Define the hypothesis class $\mathcal{H}_n$ of hypotheses from $\mathbb{R}^n$ to $\mathbb{R}^n \cup \{*\}$ by*

$$\mathcal{H}_n = \{h_B \mid \exists z \in \mathbb{R}^n, \infty \geq r > 0, \text{ such that } B = B_n(z,r)\} \cup \{h_*\}.$$

Relying on the fact that the VC dimension of balls in $\mathbb{R}^n$ is $n+1$, it is not hard to see that $d_G(\mathcal{H}_n) = n+1$. Also, it is easy to see that $d_N(\mathcal{H}_n) = 1$. It is not hard to see that there exists an ERM, $\mathcal{A}_{\text{good}}$, satisfying the requirements of Corollary 15. Thus,

$$m^r_{\mathcal{A}_{\text{good}}}(\epsilon,\delta) \leq O\left(\frac{\ln(1/\epsilon) + \ln(1/\delta)}{\epsilon}\right), \quad m^a_{\mathcal{A}_{\text{good}}}(\epsilon,\delta) \leq O\left(\frac{\ln(1/\delta)}{\epsilon^2}\right).$$

On the other hand, Theorem 9 implies that there exists a bad ERM learner, $\mathcal{A}_{\text{bad}}$ with

$$m^a_{\mathcal{A}_{\text{bad}}}(\epsilon,\delta) \geq m^r_{\mathcal{A}_{\text{bad}}}(\epsilon,\delta) \geq C_1\left(\frac{n + \ln(1/\delta)}{\epsilon}\right).$$

Our results so far show that whenever an ERM learner with a small essential range exists, the sample complexity of learning the multiclass problem can be improved over the worst ERM learner. In the next section we show that this is indeed the case for hypothesis classes which satisfy a natural condition of *symmetry*.

4.2 Learning with Symmetric Classes

We say that a hypothesis class $\mathcal{H}$ is symmetric if for any $f \in \mathcal{H}$ and any permutation $\phi : \mathcal{Y} \to \mathcal{Y}$ on labels we have that $\phi \circ f \in \mathcal{H}$ as well. Symmetric classes are a natural choice if there is no prior knowledge on properties of specific labels in $\mathcal{Y}$ (See also the discussion in Section 4.3.1 below). We now show that for symmetric classes, the Natarajan dimension characterizes the optimal sample complexity up to logarithmic factors. It follows that a finite Natarajan dimension is a necessary and sufficient condition for learnability of a symmetric class. We will make use of the following lemma, which provides a key observation on symmetric classes.

Lemma 16 *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a symmetric hypothesis class of Natarajan dimension d. Then any $h \in \mathcal{H}$ has a range of size at most $2d+1$.*

Proof If $k \leq 2d+1$ we are done. Thus assume that there are $2d+2$ distinct elements $y_1, \ldots, y_{2d+2} \in \mathcal{Y}$. Assume to the contrary that there is a hypothesis $h \in \mathcal{H}$ with a range of more than $2d+1$ values. Thus there is a set $S = \{x_1, \ldots, x_{d+1}\} \subseteq \mathcal{X}$ such that $h|_S$ has $d+1$ values in its range. Since $\mathcal{H}$ is symmetric, we can show that $\mathcal{H}$ N-shatters S as follows: Since $\mathcal{H}$ is symmetric, we can rename all the labels in the range of $h|_S$ as we please and get another function in $\mathcal{H}$. Thus there are two functions $f_1, f_2 \in \mathcal{H}$ such that for all $i \leq d+1$, $f_1(x_i) = y_i$ and $f_2(x_i) = y_{d+1+i}$. Now, let $S \subseteq T$. Since $\mathcal{H}$ is symmetric we can again rename the labels in the range of $h|_S$ to get a function $g \in \mathcal{H}$ such that $g(x) = f_1(x)$ for every $x \in T$ and $g(x) = f_2(x)$ for every $x \in S \setminus T$. Therefore the set S is shattered, thus the Natarajan dimension of $\mathcal{H}$ is at least $d+1$, contradicting the assumption. ■

First, we provide an upper bound on the sample complexity of ERM in the realizable case.

Theorem 17 *There are absolute constants C_1, C_2 such that for every symmetric hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$*

$$C_1 \left(\frac{d_N(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon} \right) \leq m^r_{\text{ERM}}(\epsilon, \delta) \leq C_2 \left(\frac{d_N(\mathcal{H}) \left(\ln(\frac{1}{\epsilon}) + \ln(d_N(\mathcal{H})) \right) + \ln(\frac{1}{\delta})}{\epsilon} \right)$$

Proof The lower bound is a restatement of Theorem 5. For the upper bound, first note that if $k \leq 4d_N(\mathcal{H}) + 2$ the upper bound trivially follows from Theorem 7. Thus assume $k > 4d_N(\mathcal{H}) + 2$. We define an ERM learner $\mathcal{A}$ with a small essential range, as required in Theorem 13: Fix a set $Z \subseteq \mathcal{Y}$ of size $|Z| = 2d_N(\mathcal{H}) + 1$. Assume an input sample $(x_1, f(x_1)), \ldots, (x_m, f(x_m))$, and denote the set of labels that appear in the sample by $L = \{f(x_i) \mid i \in [m]\}$. We require that $\mathcal{A}$ return a hypothesis which is consistent with the sample and has range in $L \cup Z$.

To see that such an ERM learner exists, observe that by Lemma 16, the range of f has at most $2d_N(\mathcal{H})+1$ distinct labels. Therefore, there is a set $R \subseteq \mathcal{Y}$ such that $|R| \leq 2d_N(\mathcal{H})+1$

and the range of f is $L \cup R$. Due to the symmetry of $\mathcal{H}$, we can rename the labels in R to labels in Z, and get another function $g \in \mathcal{H}$, that is consistent with the sample and has range in $L \cup Z$. This function can be returned by $\mathcal{A}$.

The range of $\mathcal{A}$ over all samples that are labeled by a fixed function $f \in \mathcal{H}$ is thus in the union of Z and the range of f. $|Z| \leq 2d_N(\mathcal{H}) + 1$ and by Lemma 16, the range of f is also at most $2d_N(\mathcal{H}) + 1$. Therefore the realizable essential range of $\mathcal{A}$ is at most $4d_N(\mathcal{H}) + 2$. The desired bound for the sample complexity of $\mathcal{A}$ thus follows from Theorem 13.

We now show that the same bound in fact holds for all ERM learners for $\mathcal{H}$. Suppose that $\mathcal{A}'$ is an ERM learner for which the bound does not hold. Then there is a function f and a distribution D over $\mathcal{X} \times \mathcal{Y}$ which is consistent with f, and there are m, ϵ and δ for which $m \geq m^r_{\mathcal{A}}(\epsilon, \delta)$, such that with probability greater than δ over samples S_m, $\mathrm{Err}_{\mathcal{D}}(\mathcal{A}'(S_m)) - \mathrm{Err}_{\mathcal{D}}(\mathcal{H}) > \epsilon$. Consider $\mathcal{A}$ as defined above, with a set Z that does not overlap with the range of f. For every sample S_m consistent with f, denote $\hat{f} = \mathcal{A}'(S_m)$, and let $\mathcal{A}$ return g which results from renaming the labels in $\hat{f}$ as follows: For any label that appeared in S_m, the same label is used in g. For any label that did not appear in S_m, a label from Z is used instead. Clearly, $\mathrm{Err}_{\mathcal{D}}(\mathcal{A}(S_m)) \geq \mathrm{Err}_{\mathcal{D}}(\mathcal{A}'(S_m))$. But this contradicts the upper bounds on $m^r_{\mathcal{A}}(\epsilon, \delta)$. We conclude that the upper bound holds for all ERM learners. ■

Second, we have the following upper bound for the agnostic case.

Theorem 18 *There are absolute constants C_1, C_2 such that for every symmetric hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$*

$$C_1 \left(\frac{d_N(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2} \right) \leq m^a_{\mathrm{ERM}}(\epsilon, \delta) \leq C_2 \left(\frac{d_N(\mathcal{H}) \ln(\min\{d_N(\mathcal{H}), k\}) + \ln(\frac{1}{\delta})}{\epsilon^2} \right),$$

Proof [3] The lower bound is a restatements of Theorem 6. For the upper bound, first note that if $k \leq 6d_N(\mathcal{H})$ then the upper bound follows from Theorem 6. Thus assume $k \geq 6d_N(\mathcal{H}) \geq 4d_N(\mathcal{H}) + 2$. Fix a set $Z \subseteq \mathcal{Y}$ of size $|Z| = 4d_N(\mathcal{H}) + 2$. Denote $\mathcal{H}' = \{f \in \mathcal{H} : f(\mathcal{X}) \subseteq Z\}$. By Lemma 16, the range of every function in $\mathcal{H}$ contains at most $\frac{|Z|}{2}$ elements. Thus, by symmetry, it is easy to see that $d_G(\mathcal{H}) = d_G(\mathcal{H}')$ and $d_N(\mathcal{H}) = d_N(\mathcal{H}')$. By equation (2) and the fact that the range of functions in $\mathcal{H}'$ is Z, we conclude that

$$\begin{aligned} d_G(\mathcal{H}) = d_G(\mathcal{H}') &= O(d_N(\mathcal{H}') \ln(|Z|)) \\ &= O(d_N(\mathcal{H}') \ln(\min\{d_N(\mathcal{H}'), k\}) = O(d_N(\mathcal{H}) \ln(d_N(\mathcal{H}))). \end{aligned}$$

Using Theorem 5 we obtain the desired upper bounds. ■

These results indicate that for symmetric classes, the sample complexity is determined by the Natarajan dimension up to logarithmic factors. Moreover, the ratio between the sample complexities of worst ERM and the best ERM in this case is also at most logarithmic in ϵ and the Natarajan dimension. We present the following open question:

Open question 19 *Are there symmetric classes such that there are two different ERM learners with a sample complexity ratio of $\Omega(\ln(d_N))$ between them?*

3. We note that this proof show that for symmetric classes $d_G = O(d_N \log(d_N))$. Hence, it can be adopted to give a simpler proof of Theorem 17, but with a multiplicative (rather than additive) factor of $\log\left(\frac{1}{\epsilon}\right)$.

4.3 Learning with No Prior Knowledge on Labels

Suppose we wish to learn some multiclass problem and have some hypothesis class that we wish to use for learning. The hypothesis class is defined using arbitrary label names, say $\mathcal{Y} = \{1, \ldots, k\} = [k]$. In many learning problems, we do not have any prior knowledge on a preferred mapping between these arbitrary label names and the actual real-world labels (e.g., names of topics of documents). Thus, any mapping between the real-world class labels and the arbitrary labels in $[k]$ is as reasonable as any other. We formalize the last assertion by *assuming that this mapping is chosen uniformly at random* [4]. In this section we show that in this scenario, when $k = \Omega(d_N(\mathcal{H}))$, it is likely that we will achieve poor classification accuracy.

Formally, let $\mathcal{H} \subset [k]^{\mathcal{X}}$ be a hypothesis class. Let $\mathcal{L}$ be the set of real-world labels, $|\mathcal{L}| = k$. A mapping of the label names $[k]$ to the true labels $\mathcal{L}$ is a bijection $\phi : [k] \to \mathcal{L}$. For such ϕ we let $\mathcal{H}_\phi = \{\phi \circ f : f \in \mathcal{H}\}$. [5]

The following theorem lower-bounds the approximation error when ϕ is chosen at random. The result holds for any distribution with fairly balanced label frequencies. Formally, we say that $\mathcal{D}$ over $\mathcal{X} \times \mathcal{L}$ is *balanced* if for any $l \in \mathcal{L}$, the probability that a random pair drawn from $\mathcal{D}$ has label l is at most $10/k$.

Theorem 20 *Fix $\alpha > 0$. There exist a constant $C_\alpha > 0$ such that for any $k > 0$, any hypothesis class $\mathcal{H} \subseteq [k]^{\mathcal{X}}$ such that $d_N(\mathcal{H}) \leq C_\alpha k$, and any balanced distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{L}$, with probability at least $1 - o(2^{-k})$ over the choice of ϕ, $\mathrm{Err}_{\mathcal{D}}(\mathcal{H}_\phi) \geq 1 - \alpha$.*

Remark 21 *Theorem 20 is tight, in the sense that a similar proposition cannot be obtained for all $d_N \leq f(k)$ for some $f(k) \in \omega(k)$. To see this, consider the class $\mathcal{H} = [k]^{[k]}$, for which $d_N(\mathcal{H}) = k$. For any ϕ, $\mathcal{H}_\phi = \mathcal{H}$. Thus, for any distribution such that $\mathrm{Err}_{\mathcal{D}}(\mathcal{H}) = 0$, we have $\mathrm{Err}_{\mathcal{D}}(\mathcal{H}_\phi) = 0$.*

To prove Theorem 20, we prove the following lemma, which provides a lower bound on the error of any hypothesis with a random bijection.

Lemma 22 *Let $h : \mathcal{X} \to [k]$ and let $\phi : [k] \to \mathcal{L}$ be a bijection chosen uniformly at random. Let $S = \{(x_1, l_1), \ldots, (x_m, l_m)\} \subseteq \mathcal{X} \times \mathcal{L}$. Denote, for $l \in \mathcal{L}$, $\hat{p}_l = \frac{|\{j : l_j = l\}|}{m}$. Fix $\alpha > 0$, and let $\gamma = \frac{\alpha^2}{\sum_{l \in \mathcal{L}} \hat{p}_l^2}$. Then*

$$\Pr[\operatorname*{Err}_S(\phi \circ h) < 1 - \alpha] \leq \left(\frac{8ke}{\gamma^2}\right)^{\frac{\gamma}{2}}.$$

Proof Denote $P = \sqrt{\sum_{l \in \mathcal{L}} \hat{p}_l^2}$. For a sample $S \subset \mathcal{X} \times \mathcal{L}$ and a function $f : \mathcal{X} \to \mathcal{L}$ denote $\mathrm{Gain}_S(f) = 1 - \mathrm{Err}_S(f)$. For $l \in \mathcal{L}$ denote $S_l = ((x_i, l_i))_{i : l_i = l}$. By Cauchy-Schwartz, we have

$$\operatorname*{Gain}_S(\phi \circ h) = \sum_{l \in \mathcal{L}} \hat{p}_l \cdot \operatorname*{Gain}_{S_l}(\phi \circ h) \leq P \cdot \sqrt{\sum_{l \in \mathcal{L}} \left(\operatorname*{Gain}_{S_l}(\phi \circ h)\right)^2}.$$

4. We note also that choosing this mapping at random is sometimes advocated for multiclass learning, e.g., for a filter tree Beygelzimer et al. (2007) and for an Error Correcting Output Code (Dietterich and Bakiri, 1995; Allwein et al., 2000).

5. Several notions, originally defined w.r.t. functions from $\mathcal{X}$ to $\mathcal{Y}$ (e.g. $\mathrm{Err}_{\mathcal{D}}(h)$), can be naturally extended to functions from $\mathcal{X}$ to $\mathcal{L}$. We will freely use these extensions.

Assume that $\mathrm{Err}_S(\phi \circ h) \leq 1 - \alpha$. Then

$$\sum_{l \in \mathcal{L}} \operatorname*{Gain}_{S_l}(\phi \circ h) \geq \sum_{l \in \mathcal{L}} \left(\operatorname*{Gain}_{S_l}(\phi \circ h) \right)^2 \geq \frac{\left(\mathrm{Gain}_S(\phi \circ h)\right)^2}{P^2} \geq \frac{\alpha^2}{P^2} = \gamma.$$

Note first that the left hand side is at most k, thus $\gamma \leq k$. Since for every $l \in \mathcal{L}$ it holds that $0 \leq \mathrm{Gain}_{S_l}(\phi \circ h) \leq 1$, we conclude that there are at least $n = \lceil \frac{\gamma}{2} \rceil$ labels $l \in \mathcal{L}$ such that

$$\operatorname*{Gain}_{S_l}(\phi \circ h) \geq \frac{\gamma}{2k} \ .$$

For a fixed set of n labels $l_1, \ldots, l_n \in \mathcal{L}$, the probability that $\forall i,\ \mathrm{Gain}_{S_{l_i}}(\phi \circ h) \geq \frac{\gamma}{2k}$ is at most

$$\prod_{i=1}^{n} \frac{2k}{(k+1-i)\gamma} \leq \left(\frac{2k}{(k+1-n)\gamma} \right)^n \ .$$

To see that, suppose that ϕ is sampled by first choosing the value of $\phi^{-1}(l_1)$ then $\phi^{-1}(l_2)$ and so on. For every l_i, there are at most $\frac{2k}{\gamma}$ values for $\phi^{-1}(l_i)$ for which $\mathrm{Gain}_{S_{l_i}}(\phi \circ h) \geq \frac{\gamma}{2k}$. Thus, after the values of $\phi^{-1}(l_1), \ldots, \phi^{-1}(l_{i-1})$ have been determined, the probability that $\phi^{-1}(l_i)$ is one of these values is at most $\frac{2k}{(k+1-i)\cdot\gamma}$.

It follows that the probability that $\mathrm{Gain}_{S_l}(\phi \circ h) \geq \frac{\gamma}{2k}$ for n different labels l is at most

$$\begin{aligned}
\binom{k}{n} \cdot \left(\frac{2k}{(k+1-n)\gamma} \right)^n &\leq \left(\frac{ek}{n} \right)^n \cdot \left(\frac{2k}{(k+1-n)\gamma} \right)^n \\
&\leq \left(\frac{2ke}{\gamma} \right)^n \cdot \left(\frac{2k}{(k-\gamma/2)\gamma} \right)^n \\
&\leq \left(\frac{8ke}{\gamma^2} \right)^n .
\end{aligned}$$

If $\frac{8ke}{\gamma^2} \geq 1$ then the bound in the statement of the lemma holds trivially. Otherwise, the bound follows since $n \geq \gamma/2$. ■

Proof [Proof of Theorem 20] Denote $p_l = \Pr_{(X,L)\sim\mathcal{D}}[L = l]$. Let $S = \{(x_1, l_1), \ldots, (x_m, l_m)\} \subseteq \mathcal{X} \times \mathcal{L}$ be an i.i.d. sample drawn according to $\mathcal{D}$. Denote $\hat{p}_l = \frac{|\{j : l_j = l\}|}{m}$.

For any fixed bijection ϕ, by Theorem 6, with probability $1 - \delta$ over the choice of S,

$$\operatorname*{Err}_{\mathcal{D}}(\mathcal{H}_\phi) \geq \inf_{h \in \mathcal{H}} \mathrm{Err}_S(\phi \circ h) - O\left(\sqrt{\frac{\ln(k) d_N(\mathcal{H}) + \ln(1/\delta)}{m}} \right).$$

Since there are less than k^k such bijections, we can apply the union bound to get that with probability $1 - \delta$ over the choice of S,

$$\forall \phi, \quad \operatorname*{Err}_{\mathcal{D}}(\mathcal{H}_\phi) \geq \inf_{h \in \mathcal{H}} \mathrm{Err}_S(\phi \circ h) - O\left(\sqrt{\frac{\ln(k) d_N(\mathcal{H}) + k\ln(k) + \ln(1/\delta)}{m}} \right).$$

Assume $k \geq C \cdot d_N(\mathcal{H})$ for some constant $C > 0$, and let $m = \Theta\left(\frac{k \cdot \ln(k)}{\alpha^2}\right)$ such that with probability at least 3/4,

$$\forall \phi, \quad \operatorname*{Err}_{\mathcal{D}}(\mathcal{H}_\phi) \geq \inf_{h \in \mathcal{H}} \operatorname{Err}_S(\phi \circ h) - \alpha/2. \tag{11}$$

We have

$$E[\sum_{l \in \mathcal{L}} \hat{p}_l^2] = 2\frac{1}{m^2} \sum_{l \in \mathcal{L}} \left(\binom{m}{2} p_l^2 + m p_l \right) \leq 2k \cdot \left(\frac{m(m-1)}{2m^2} \frac{100}{k^2} + \frac{10}{mk} \right) \leq \frac{120}{k}.$$

Thus, by Markov's inequality, with probability at least $\frac{1}{2}$ over the samples we have

$$\sum_{l \in \mathcal{L}} \hat{p}_l^2 < \frac{240}{k}. \tag{12}$$

Thus, with probability at least 1/4, both (12) and (11) hold. In particular, there exists a single sample S for which both (12) and (11) hold. Let us fix such an $S = \{(x_1, l_1), \ldots, (x_m, l_m)\}$.

Assume now that $\phi : \mathcal{Y} \to \mathcal{L}$ is sampled uniformly. For a fixed $h \in \mathcal{H}$ and for $\gamma = (\alpha/2)^2 / \sum_{l \in \mathcal{L}} \hat{p}_l^2 \geq k\alpha^2/960$, we have, by Lemma 22 that

$$\Pr_{\phi}\left[\operatorname{Err}_S(\phi \circ h) < 1 - \frac{\alpha}{2}\right] \leq \left(\frac{8ke}{\gamma^2}\right)^{\frac{\gamma}{2}} \leq (C_1 k \alpha^4)^{-C_2 k \alpha^2} := \eta,$$

for constants $C_1, C_2 > 0$. By Lemma 14, $|\mathcal{H}|_{\{x_1,\ldots,x_m\}}| \leq (m \cdot k)^{2d_N(\mathcal{H})}$. Thus, with probability $\geq 1 - (m \cdot k)^{2d} \cdot \eta$ over the choice of ϕ, $\inf_{h \in \mathcal{H}} \operatorname{Err}_S(\phi \circ h) \geq 1 - \frac{\alpha}{2}$ and by (11) also

$$\operatorname{Err}_{\mathcal{D}}(\mathcal{H}_\phi) \geq 1 - \alpha. \tag{13}$$

By our choice of m, and since $k \geq d_N(\mathcal{H})$, for some universal constant $C_1 \geq 1$, $m \leq C_1 \cdot \frac{k^2}{\alpha^2}$. Considering α a constant, we have, for some constants $C_i > 0$,

$$(m \cdot k)^{2d_N(\mathcal{H})} \cdot \eta \leq (C_3 k)^{6d_N(\mathcal{H})} \cdot (C_4 k)^{-C_5 k}.$$

By requiring that $k \geq 12 d_N(\mathcal{H})/C_5$, we get that the right hand side is at most $o(2^{-k})$.

■

4.3.1 Symmetrization

From Theorem 20 it follows that if there is no prior knowledge about the labels, and the label frequencies are balanced, we must use a class of Natarajan dimension $\Omega(k)$ to obtain reasonable approximation error. As we show next, in this case, there is almost no loss in the sample complexity if one instead uses the *symmetrization* of the class, obtained by considering all the possible label mappings $\phi : [k] \to \mathcal{L}$. Formally, let $\mathcal{H} \subset [k]^{\mathcal{X}}$ be some hypothesis class and let $\mathcal{L}$ be a set with $|\mathcal{L}| = k$. The symmetrization of $\mathcal{H}$ is the symmetric class

$$\mathcal{H}_{\text{sym}} = \{\phi \circ h \mid h \in \mathcal{H},\ \phi : [k] \to \mathcal{L} \text{ is a bijection}\}.$$

Lemma 23 *Let $\mathcal{H} \subseteq [k]^{\mathcal{X}}$ be a hypothesis class with Natarajan dimension d. Then*

$$d_N(\mathcal{H}_{\text{sym}}) = O(\max\{d\log(d), k\log(k)\}).$$

Proof Let $d_s = d_N(\mathcal{H}_{\text{sym}})$. Let $X \subset \mathcal{X}$ be a set of cardinality d_s that is N-shattered by $\mathcal{H}_{\text{sym}}$. By Lemma 14, $|\mathcal{H}|_X| \leq (d_s k^2)^d$. It follows that $|\mathcal{H}_{\text{sym}}|_X| \leq k!(d_s k^2)^d$. On the other hand, since $\mathcal{H}_{\text{sym}}$ N-shatters X, $|\mathcal{H}_{\text{sym}}|_X| \geq 2^{|X|} = 2^{d_s}$. It follows that $2^{d_s} \leq k!(d_s k^2)^d$. Taking logarithms we obtain that $d_s \leq k\log(k) + d(\ln(d_s) + 2\ln(k))$. The Lemma follows. ■

5. Other Learning Settings

In this section we consider the characterization of learnability in other learning settings: The online setting and the bandit setting.

5.1 The Online Model

Learning in the online model is conducted in a sequence of consecutive rounds. On each round $t = 1, 2, \ldots, T$, the environment presents a sample $x_t \in \mathcal{X}$, then the algorithm should predict a value $\hat{y}_t \in \mathcal{Y}$, and finally the environment reveals the correct value $y_t \in \mathcal{Y}$. The prediction at time t can be based only on the examples $x_1, \ldots, x_t$ and the previous outcomes $y_1, \ldots, y_{t-1}$. Our goal is to minimize the number of prediction mistakes in the worst case, where the number of mistakes on the first T rounds is $L_T = |\{t \in [T] : \hat{y}_t \neq y_t\}|$. Assume a hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$. In the realizable setting, we assume that for some function $f \in \mathcal{H}$ all the outcomes are evaluations of f, namely, $y_t = f(x_t)$.

Learning in the realizable online model has been studied by Littlestone (1987), who showed that a combinatorial measure, called the Littlestone dimension, characterizes the min-max optimal number of mistakes for *binary* hypotheses classes in the realizable case. We propose a generalization of the Littlestone dimension to multiclass hypotheses classes.

Consider a rooted tree T whose internal nodes are labeled by elements from $\mathcal{X}$ and whose edges are labeled by elements from $\mathcal{Y}$, such that the edges from a single parent to its child-nodes are each labeled with a different label. The tree T is *shattered* by $\mathcal{H}$ if, for every path from root to leaf which traverses the nodes $x_1, \ldots, x_k$, there is a function $f \in \mathcal{H}$ such that $f(x_i)$ is the label of the edge (x_i, x_{i+1}). We define the *Littlestone dimension* of a multiclass hypothesis class $\mathcal{H}$, denoted L-Dim($\mathcal{H}$), to be the maximal depth of a complete binary tree that is shattered by $\mathcal{H}$ (or ∞ if there are a shattered trees for arbitrarily large depth).

As we presently show, the number L-Dim($\mathcal{H}$) fully characterizes the worst-case mistake bound for the online model in the realizable setting. The upper bound is achieved using the following algorithm.

Algorithm: Standard Optimal Algorithm (SOA)
Initialization: $V_0 = \mathcal{H}$.
For $t = 1, 2 \ldots$,
receive x_t

for $y \in \mathcal{Y}$, let $V_t^{(y)} = \{f \in V_{t-1} : f(x_t) = y\}$
predict $\hat{y}_t \in \arg\max_y \text{L-Dim}(V_t^{(y)})$
receive true answer y_t
update $V_t = V_t^{(y_t)}$

Theorem 24 *The SOA algorithm makes at most* L-Dim($\mathcal{H}$) *mistakes on any realizable sequence. Furthermore, the worst-case number of mistakes of* any *deterministic online algorithm is at least* L-Dim($\mathcal{H}$)*. For any randomized online algorithm, the expected number of mistakes on the worst sequence is at least* $\frac{1}{2}$ L-Dim($\mathcal{H}$)*.*

Proof (sketch) First, we show that the *SOA* algorithm makes at most L-Dim($\mathcal{H}$) mistakes. The proof is a simple adaptation of the proof of the binary case (see Littlestone, 1987; Shalev-Shwartz, 2012). We note that for each t there is at most one $y \in \mathcal{Y}$ with L-Dim($V_t^{(y)}$) = L-Dim(V_t), and for the rest of the labels we have L-Dim($V_t^{(y)}$) < L-Dim(V_t) (otherwise, it is not hard to construct a tree of depth L-Dim(V_t) + 1, whose root is x_t, that is shattered by V_t). Thus, whenever the algorithm errs, the Littlestone dimension of V_t decreases by at least 1, so after L-Dim($\mathcal{H}$) mistakes, V_t is composed of a single function.

For the second part of the theorem, it is not hard to see that, given a shattered tree of depth L-Dim($\mathcal{H}$), the environment can force any deterministic online learning algorithm to make L-Dim($\mathcal{H}$) mistakes. Note also that allowing the algorithm to make randomized predictions cannot be too helpful. It is easy to see that given a shattered tree of depth L-Dim($\mathcal{H}$), the environment can enforce any randomized online learning algorithm to make at least L-Dim($\mathcal{H}$)/2 mistakes on average, by traversing the shattered tree, and providing at every round the label that the randomized algorithm is less likely to predict. ■

In the agnostic case, the sequence of outcomes, $y_1, \ldots, y_m$, is not necessarily consistent with some function $f \in \mathcal{H}$. Thus, one wishes to bound the *regret* of the algorithm, instead of its absolute number of mistakes. The regret is the difference between the number of mistakes made by the algorithm and the number of mistakes made by the best-matching function $f \in \mathcal{H}$. The agnostic case for classes of binary-output functions has been studied in Ben-David et al. (2009). It was shown that, as in the realizable case, the Littlestone dimension characterizes the optimal regret bound.

We show that the generalized Littlestone dimension characterizes the optimal regret bound for the multiclass case as well. The proof follows the paradigm of 'learning with expert advice' (see e.g. Cesa-Bianchi and Lugosi, 2006; Shalev-Shwartz, 2012), which we now briefly describe. Suppose that at each step, t, before the algorithm chooses its prediction, it observes N *advices* $(f_1^t, \ldots, f_N^t) \in \mathcal{Y}^N$, which can be used to determine its prediction. We think of f_i^t as the prediction made by the *expert* i at time t and denote the *loss* of the expert i at time T by $L_{i,T} = |\{t \in [T] : f_{i,t} \neq y_t\}|$. The goal here it to devise an algorithm that achieves a loss which is comparable with the loss of the best expert. Given T, the following algorithm (Cesa-Bianchi and Lugosi, 2006, chapter 2) achieves expected loss at most $\min_{i \in [N]} L_{i,T} + \sqrt{\frac{1}{2}\ln(N)T}$.

Algorithm: Learning with Expert Advice (LEA)
Parameters: Time horizon – T

Set $\eta = \sqrt{8\ln(N)/T}$
For $t = 1, 2 \ldots, T$
receive expert advices $(f_1^t, \ldots, f_N^t) \in \mathcal{Y}^N$
predict $\hat{y}_t = f_{i,t}$ with probability proportional to $\exp(-\eta L_{i,t-1})$
receive true answer y_t

We use this algorithm and its guarantee to prove the following theorem.

Theorem 25 *In the agnostic online multiclass setting, the expected loss of the optimal algorithm on the worst-case sequence is at most* $\min_{f\in\mathcal{H}} L_{f,T} + \sqrt{\frac{1}{2}\,\text{L-Dim}(\mathcal{H})T\log(Tk)}$.

Proof First, we construct an expert for every $f \in \mathcal{H}$, whose advice at time t is $f(x_t)$. Denote the loss of the expert corresponding to f at time t by $L_{f,t}$. Running the algorithm LEA with this set of experts yields an algorithm whose expected error is at most $\min_{f\in\mathcal{H}} L_{f,T} + \sqrt{\frac{1}{2}\ln(|\mathcal{H}|)T}$. Our goal now is to construct a more compact set of experts, which will allow us to bound the loss in terms of $\text{L-Dim}(\mathcal{H})$ instead of $\ln(|\mathcal{H}|)$.

Given time horizon T, let $A_T = \{A \subset [T] \mid |A| \leq \text{L-Dim}(\mathcal{H})\}$. For every $A \in A_T$ and $\phi : A \to \mathcal{Y}$, we define an expert $E_{A,\phi}$. The expert $E_{A,\phi}$ imitates the SOA algorithm when it errs exactly on the examples $\{x_t \mid t \in A\}$ and the true labels of these examples are determined by ϕ. Formally, the expert $E_{A,\phi}$ proceeds as follows:

Set $V_1 = \mathcal{H}$.
For $t = 1, 2 \ldots, T$
Receive x_t.
Set $l_t = \text{argmax}_{y\in\mathcal{Y}}\,\text{L-Dim}(\{f \in V_t : f(x_t) = y\})$.
If $t \in A$, Predict $\phi(t)$ and update $V_{t+1} = \{f \in V_t : f(x_t) = \phi(t)\}$.
If $t \notin A$, Predict l_t and update $V_{t+1} = \{f \in V_t : f(x_t) = l_t\}$.

The number of experts we constructed is $\sum_{j=0}^{\text{L-Dim}(\mathcal{H})} \binom{T}{j}(k-1)^j \leq (Tk)^{\text{L-Dim}(\mathcal{H})}$. Denote the number of mistakes made by the expert $E_{A,\phi}$ after T rounds by $L_{A,\phi,T}$. If we apply the LEA algorithm with the set of experts we have constructed, the resulting algorithm makes at most

$$\min_{A,\phi} L_{A,\phi,T} + \sqrt{\frac{1}{2}T\,\text{L-Dim}(\mathcal{H})\ln(Tk)}$$

mistakes. We claim that $\min_{A,\phi} L_{A,\phi,T} \leq \min_{f\in\mathcal{H}} L_{f,T}$: Let $f \in \mathcal{H}$. Denote by $A \subset [T]$ the set of rounds in which the SOA algorithm errs when running on the sequence $(x_1, f(x_1)), \ldots, (x_T, f(x_T))$ and define $\phi : A \to \mathcal{Y}$ by $\phi(t) = f(x_t)$. Since the SOA algorithm makes at most $\text{L-Dim}(\mathcal{H})$ mistakes, $|A| \leq \text{L-Dim}(\mathcal{H})$. It is not hard to see that the predictions of the expert $E_{A,\phi}$ coincide with the predictions of the expert E_f. Thus, $L_{A,\phi,T} = L_{f,T}$. ■

Adapting the proof of Lemma 14 from Ben-David et al. (2009), we conclude a corresponding lower bound:

Theorem 26 *In the agnostic online multiclass setting, the expected loss of every algorithm on the worst-case sequence is at least* $\min_{f\in\mathcal{H}} L_{f,T} + \sqrt{\frac{1}{8}\,\text{L-Dim}(\mathcal{H})T}$.

We leave as an open question to close the gap between the bounds in the above Theorems. Note that this gap is analogous to the sample complexity gap for ERM learners in the PAC setting, seen in Theorem 6.

5.2 The Bandit Setting

So far we have assumed that the label of each training example is fully revealed. In this section we deal with the bandit setting. In this setting, the learner does not get to see the correct label of a training example. Instead, the learner first receives an instance $x \in \mathcal{X}$, and should guess a label, $\hat{y}$. The learner then receives a binary response, which indicates only whether the guess was correct or not. If the guess is correct then the learner knows the identity of the correct label. If the guess is wrong, the learner only knows that $\hat{y}$ is not the correct label, and not the identity of the correct label.

5.2.1 Bandit vs. Full Information in the Batch Model

In this section we consider the bandit setting in the batch model. In this setting the sample is drawn i.i.d. as before, but the learner first observes only the instances $x_1, \ldots, x_m$. The learner then guesses a label for each of the instances, and receives a binary response indicating for each label whether it was the correct one.

Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class and let $k = |\mathcal{Y}|$. Our goal is to analyze the *realizable bandit sample complexity* of $\mathcal{H}$, which we denote by $m_b^r(\epsilon, \delta)$, and the *agnostic bandit sample complexity* of $\mathcal{H}$, which we denote by $m_b^a(\epsilon, \delta)$. The following theorem provides upper bounds on the sample complexities.

Theorem 27 *Let $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be a hypothesis class. Then,*

$$m_b^r(\epsilon,\delta) = O\left(k \cdot \frac{d_G(\mathcal{H}) \cdot \ln\left(\frac{1}{\epsilon}\right) + \ln(\frac{1}{\delta})}{\epsilon}\right) \text{ and } m_b^a(\epsilon,\delta) = O\left(k \cdot \frac{d_G(\mathcal{H}) + \ln(\frac{1}{\delta})}{\epsilon^2}\right) .$$

Proof Let $\mathcal{A}_f$ be a (full information) ERM learner for $\mathcal{H}$. Consider the following algorithm, denoted $\mathcal{A}_b$, for the bandit setting: Given a sample $(x_i, y_i)_{i=1}^m$, for each i the algorithm guesses a label $\hat{y}_i \in \mathcal{Y}$ drawn uniformly at random. Then the algorithm calls $\mathcal{A}_f$ with an input sample which consists only of the sample pairs for which the binary response indicated that the guess $\hat{y}_i$ was correct. Thus, the input sample is $\{(x_i, \hat{y}_i) \mid \hat{y}_i = y_i\}$. $\mathcal{A}_b$ then returns whatever hypothesis $\mathcal{A}_f$ returned.

We show that $m^r_{\mathcal{A}_b}(\epsilon,\delta) \leq 3k \cdot m^r_{\mathcal{A}_f}(\epsilon, \frac{\delta}{2}) + \frac{3}{2}\log\left(\frac{2}{\delta}\right) =: m'$ and similarly for the agnostic case, so that the theorem is implied by the bounds in the full information setting (Theorem 5). Indeed, suppose that m examples suffice for $\mathcal{A}_f$ to return a hypothesis with excess error at most ϵ, with probability at least $1 - \frac{\delta}{2}$. Let $(x_i, y_i)_{i=1}^{m'}$ be a sample for the bandit algorithm. By Chernoff's bound, with probability at least $1 - \frac{\delta}{2}$, $\mathcal{A}_b$ guesses correctly the label of at least m examples. Therefore $\mathcal{A}_f$ runs on a sample of at least this size. The sample that $\mathcal{A}_f$ receives is a conditionally i.i.d. sample, given the size of the sample, with

the same conditional distribution as the one the original sample was sampled from. Thus, with probability at least $1 - \frac{\delta}{2}$, $\mathcal{A}_f$ (and, consequently, $\mathcal{A}_b$) returns a hypothesis with excess error at most ϵ. ■

An interesting quantity to consider is the price of bandit information in the batch model: Let $\mathcal{H}$ be a hypotheses class, and define $\mathrm{PBI}_{\mathcal{H}}(\epsilon, \delta) = m^r_{b,\mathcal{H}}(\epsilon, \delta)/m^r_{\mathrm{PAC},\mathcal{H}}(\epsilon, \delta)$. By Theorems 27 and 6 and Equation 2 we see that, $\mathrm{PBI}(\epsilon, \delta) = O(\ln(\frac{1}{\epsilon})k\ln(k))$. This is essentially tight since it is not hard to see that if both $\mathcal{X}, \mathcal{Y}$ are finite and we let $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$, then $\mathrm{PBI}_{\mathcal{H}} = \Omega(k)$.

Using Theorems 27 and 5 and Equation 2 we can further conclude that, as in the full information case, the finiteness of the Natarajan dimension is necessary and sufficient for learnability in the bandit setting as well. However, the ratio between the upper bound due to Theorem 27 and the lower bound, due to Theorem 5, is $\Omega(\ln(k) \cdot k)$. It would be interesting to find a more tight characterization of the sample complexity in the bandit setting. This characterization cannot depend solely on the Natarajan dimension, or other quantities which are strongly related to it (such as the graph dimension or other notion of dimension defined in Ben-David et al. (1995)): For example, the classes $[k]^{[d]}$ and $[2]^{[d]}$ have the same Natarajan dimension, but their bandit sample complexity differs by a factor of $\Omega(k)$.

5.2.2 Bandit vs. Full Information in the Online Model

We now consider Bandits in the online learning model. We focus on the realizable case, in which the feedback provided to the learner is consistent with some function $f_0 \in \mathcal{H}$. We define a new notion of dimension of a class, that determines the sample complexity in this setting.

As in Section 5.1, consider a rooted tree T whose internal nodes are labeled by elements from $\mathcal{X}$ and whose edges are labeled by elements from $\mathcal{Y}$, such that the edges from a single parent to its child-nodes are each labeled with a different label. The tree T is *BL-shattered* by $\mathcal{H}$ if, for every path from root to leaf $x_1, \ldots, x_k$, there is a function $f \in \mathcal{H}$ such that for every i, $f(x_i)$ is *different* from the label of (x_i, x_{i+1}). The **Bandit-Littlestone dimension** of $\mathcal{H}$, denoted BL-dim($\mathcal{H}$), is the maximal depth of a complete k-ary tree that is BL-shattered by $\mathcal{H}$.

Theorem 28 *Let $\mathcal{H}$ be a hypothesis class with $L = \mathrm{BL\text{-}Dim}(\mathcal{H})$. Then every deterministic online bandit learning algorithm for $\mathcal{H}$ will make at least L mistakes in the worst case. Moreover, there is an online learning algorithm that makes at most L mistakes on every realizable sequence.*

Proof First, let T be a BL-shattered tree of depth L. We show that for every deterministic learning algorithm there is a sequence $x_1, \ldots, x_L$ and a labeling function $f_0 \in \mathcal{H}$ such that the algorithm makes L mistakes on this sequence. The sequence consists of the instances attached to nodes of T, when traversing the tree from the root to one of its leaves, such that the label of each edge (x_i, x_{i+1}) is equal to the algorithm's prediction $\hat{y}_i$. The labeling function $f_0 \in \mathcal{H}$ is one such that for all i, $f_0(x_i)$ is different from the label of edge (x_i, x_{i+1}). Such a function exists since T is BL-shattered, and the algorithm will clearly make L mistakes on this sequence.

Second, the following online learning algorithm makes at most L mistakes on any realizable input sequence.

Algorithm: Bandit Standard Optimal Algorithm (BSOA)
Initialization: $V_0 = \mathcal{H}$.
For $t = 1, 2 \ldots$,
Receive x_t
For $y \in \mathcal{Y}$, let $V_t^{(y)} = \{f \in V_{t-1} : f(x_t) \neq y\}$
Predict $\hat{y}_t \in \arg\min_y \text{BL-Dim}(V_t^{(y)})$
Receive an indication whether $\hat{y}_t = f(x_t)$
If the prediction is wrong, update $V_t = V_t^{(\hat{y}_t)}$.

To see that BSOA makes at most L mistakes, note that at each time t, there is at least one $V_t^{(y)}$ with $\text{BL-Dim}(V_t^{(y)}) < \text{BL-Dim}(V_{t-1})$. This can be seen by assuming to the contrary that this is not so, and concluding that if $\text{BL-Dim}(V_t^{(y)}) = \text{BL-Dim}(V_{t-1})$ for all $y \in [k]$, then one can construct a shattered tree of size $\text{BL-Dim}(V_{t-1}) + 1$ for V_{t-1}, thus reaching a contradiction.

Thus, whenever the algorithm errs, the dimension of V_t decreases by one. Thus, after L mistakes, the dimension is 0, which means that there is a single function that is consistent with the sample, so no more mistakes can occur. ■

The price of bandit information: Let $\text{PBI}(\mathcal{H}) = \text{BL-Dim}(\mathcal{H})/\,\text{L-Dim}(\mathcal{H})$ and fix $k \geq 2$. How large can $\text{PBI}(\mathcal{H})$ be when $\mathcal{H}$ is a class of functions from a domain $\mathcal{X}$ to a range $\mathcal{Y}$ of cardinality k? We refer the reader to Daniely and Helbertal (2013), where it is shown that $\text{PBI}(\mathcal{H}) \leq 4k\log(k)$. This bound is tight up to the logarithmic factor.

6. Discussion

We have shown in this work that even in the simple case of multiclass learning, different ERM learners for the same problem can have large gaps in their sample complexities. To put our results in a more general perspective, consider the *General Setting of Learning* introduced by Vapnik (1998). In this setting, a *learning problem* is a triplet $(\mathcal{H}, \mathcal{Z}, l)$, where $\mathcal{H}$ is a hypothesis class, $\mathcal{Z}$ is a data domain, and $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ is a loss function. We emphasize that $\mathcal{H}$ is not necessarily a class of functions but rather an abstract set of models. The goal of the learner is, given a sample $S \in \mathcal{Z}^m$, sampled from some (unknown) distribution $\mathcal{D}$ over $\mathcal{Z}$, to find a hypothesis $h \in \mathcal{H}$ that minimizes the *expected loss*, $l(h) = \mathbb{E}_{z\sim\mathcal{D}}[l(h, z)]$.

The general setting of learning encompasses multiclass learning as follows: given a hypotheses class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$, take $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and define $l : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ by $l(h, (x, y)) = 1[h(x) \neq y]$. However, the general learning setting encompasses many other problems as well, for instance:

- **Regression with the squared loss:** Here, $\mathcal{Z} = \mathbb{R}^n \times \mathbb{R}$, $\mathcal{H}$ is a set of real-valued functions over $\mathbb{R}^n$ and $l(h, (x, y)) = (h(x) - y)^2$.

- **k-means:** Here, $\mathcal{Z} = \mathbb{R}^n$, $\mathcal{H} = (\mathbb{R}^n)^k$ and, for $h = (c_1, \ldots, c_k) \in \mathcal{H}$ and $x \in \mathcal{Z}$, the loss is $l((c_1, \ldots, c_k), x) = \min_{j \in [k]} ||c_j - x||^2$.
- **Density estimation:** Here, $\mathcal{Z}$ is an arbitrary finite set, $\mathcal{H}$ is some set of probability density functions over $\mathcal{Z}$, and the loss function is the log loss, $l(p, x) = -\ln(p(x))$.

A learning problem is *learnable* in the general setting of learning if there exists a function $\mathcal{A} : \cup_{m=1}^{\infty} \mathcal{Z}^m \to \mathcal{H}$ such that for every $\epsilon > 0$ and $\delta > 0$ there exists an m such that for every distribution $\mathcal{D}$ over $\mathcal{Z}$,

$$\Pr_{S \sim \mathcal{Z}^m} \left(l(\mathcal{A}(S)) \geq \inf_{h \in \mathcal{H}} l(h) + \epsilon \right) < \delta$$

A learning problem *converges uniformly* if, for every $\epsilon > 0$,

$$\lim_{m \to \infty} \Pr_{S \sim \mathcal{Z}^m} \left(\sup_{h \in \mathcal{H}} |l(h) - l_S(h)| > \epsilon \right) = 0$$

where for $S = (z_1, \ldots, z_m) \in \mathcal{Z}^m$, $l_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h, z_i)$ is the empirical loss of h on the sample S. An easy observation is that uniform convergence implies learnability, and a classical result is that for binary classification and for regression (with absolute or squared loss), the inverse implication also holds. Thus, it was believed that excluding some trivialities, learnability is equivalent to uniform convergence. In Shalev-Shwartz et al. (2010) it is shown that for stochastic convex optimization, learnability does not imply uniform convergence, giving an evidence that the above belief might be misleading. Our results in this work can be seen as another step in this direction, as we have shown that even in multiclass classification – a simple, natural and popular generalization of binary classification, the above mentioned equivalence no longer holds.

We conclude with an open question. In view of our results in Section 4, the following conjecture suggests itself.

Conjecture 29 *There exists a constant C such that, for* every *hypothesis class $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$,*

$$m^r_{\text{PAC}}(\epsilon, \delta) \leq C \left(\frac{d_N(\mathcal{H}) \ln(\frac{1}{\epsilon}) + \ln(\frac{1}{\delta})}{\epsilon} \right)$$

In light of Theorem 9 and the fact that there are cases where $d_G \geq \log_2(k-1) d_N$, the conjecture can only be proved if this learning rate can be achieved by a learning algorithm that is not just an *arbitrary* ERM learner. So far, all the general upper bounds that we are aware of are valid for *any* ERM learner. Understanding how to select among ERM learners is fundamental as it teaches us what is the optimal way to learn. We hope that our examples from section 4 and our result for symmetric classes will lead to a better understanding of the optimal learning method.

Remark 30 *A subsequent paper (Daniely and Shalev-Shwartz, 2014) established several results that are highly related to the subject of this paper. First, they have shown that the ERM rule is suboptimal even for multiclass classification with linear classes. Second, they have shown that for some classes, an optimal learner must be improper – that is, it must have the ability to return a hypothesis that does not belong to the learnt class. Finally, they have show that the one-inclusion algorithm (Rubinstein et al., 2006) is optimal for multiclass classification. We note that Conjecture 29 is still open.*

Acknowledgments

We wish to thank Ohad Shamir for valuable comments. Amit Daniely is a recipient of the Google Europe Fellowship in Learning Theory, and this research is supported in part by this Google Fellowship. Sivan Sabato is partly supported by the ISRAEL SCIENCE FOUNDATION (grant No. 555/15).

References

E. L. Allwein, R.E. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141, 2000.

N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM (JACM)*, 44(4):615–631, 1997.

M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambirdge University Press, 1999.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

P. Auer, N. Cesa-Bianchi, Y. Freund, and R.E. Schapire. The nonstochastic multiarmed bandit problem. *SICOMP: SIAM Journal on Computing*, 32, 2003.

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

PL Bartlett, PM Long, and RC Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452, 1996.

S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. Long. Characterizations of learnability for classes of $\{0, \ldots, n\}$-valued functions. *Journal of Computer and System Sciences*, 50: 74–86, 1995.

S. Ben-David, D. Pal, , and S. Shalev-Shwartz. Agnostic online learning. In *COLT*, 2009.

A. Beygelzimer, J. Langford, and P. Ravikumar. Multiclass classification with filter trees. *Preprint, June*, 2007.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

A. Daniely and S. Shalev-Shwartz. Optimal learners to multiclass problems. In *COLT*, 2014.

Amit Daniely and Tom Helbertal. The price of bandit information in multiclass online classification. In *Conference on Learning Theory*, pages 93–104, 2013.

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, January 1995.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

S.M. Kakade, S. Shalev-Shwartz, and A. Tewari. Efficient bandit algorithms for online multiclass prediction. In *International Conference on Machine Learning*, 2008.

Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17:115–141, 1994.

N. Littlestone. Learning when irrelevant attributes abound. In *FOCS*, pages 68–77, October 1987.

B. K. Natarajan. On learning sets and functions. *Mach. Learn.*, 4:67–97, 1989.

Benjamin I Rubinstein, Peter L Bartlett, and J Hyam Rubinstein. Shifting, one-inclusion mistake bounds and tight multiclass expected risk bounds. In *Advances in Neural Information Processing Systems*, pages 1193–1200, 2006.

S. Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012.

S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *The Journal of Machine Learning Research*, 9999:2635–2670, 2010.

L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, XVI(2):264–280, 1971.

V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

Journal of Machine Learning Research 16 (2015) 2405-2436 Submitted 7/14; Revised 3/15; Published 12/15

Geometry and Expressive Power of Conditional Restricted Boltzmann Machines

Guido Montúfar MONTUFAR@MIS.MPG.DE
Max Planck Institute for Mathematics in the Sciences
04103 Leipzig, Germany

Nihat Ay NAY@MIS.MPG.DE
Max Planck Institute for Mathematics in the Sciences
04103 Leipzig, Germany

Department of Mathematics and Computer Science
Leipzig University
04009 Leipzig, Germany

Santa Fe Institute
Santa Fe, NM 87501, USA

Keyan Ghazi-Zahedi ZAHEDI@MIS.MPG.DE
Max Planck Institute for Mathematics in the Sciences
04103 Leipzig, Germany

Editor: Ruslan Salakhutdinov

Abstract

Conditional restricted Boltzmann machines are undirected stochastic neural networks with a layer of input and output units connected bipartitely to a layer of hidden units. These networks define models of conditional probability distributions on the states of the output units given the states of the input units, parameterized by interaction weights and biases. We address the representational power of these models, proving results on their ability to represent conditional Markov random fields and conditional distributions with restricted supports, the minimal size of universal approximators, the maximal model approximation errors, and on the dimension of the set of representable conditional distributions. We contribute new tools for investigating conditional probability models, which allow us to improve the results that can be derived from existing work on restricted Boltzmann machine probability models.

Keywords: conditional restricted Boltzmann machine, universal approximation, Kullback-Leibler approximation error, expected dimension

1. Introduction

Restricted Boltzmann Machines (RBMs) (Smolensky, 1986; Freund and Haussler, 1994) are generative probability models defined by undirected stochastic networks with bipartite interactions between visible and hidden units. These models are well-known in machine learning applications, where they are used to infer distributed representations of data and to train the layers of deep neural networks (Hinton et al., 2006; Bengio, 2009). The restricted connectivity of these networks allows to train them efficiently on the basis of cheap inference

and finite Gibbs sampling (Hinton, 2002, 2012), even when they are defined with many units and parameters. An RBM defines Gibbs-Boltzmann probability distributions over the observable states of the network, depending on the interaction weights and biases. An introduction is offered by Fischer and Igel (2012). The expressive power of these probability models has attracted much attention and has been studied in numerous papers, treating, in particular, their universal approximation properties (Younes, 1996; Le Roux and Bengio, 2008; Montúfar and Ay, 2011), approximation errors (Montúfar et al., 2011), efficiency of representation (Martens et al., 2013; Montúfar and Morton, 2015), and dimension (Cueto et al., 2010).

In certain applications, it is preferred to work with conditional probability distributions, instead of joint probability distributions. For example, in a classification task, the conditional distribution may be used to indicate a belief about the class of an input, without modeling the probability of observing that input; in sensorimotor control, it can describe a stochastic policy for choosing actions based on world observations; and in the context of information communication, to describe a channel. RBMs naturally define models of conditional probability distributions, called conditional restricted Boltzmann machines (CRBMs). These models inherit many of the nice properties of RBM probability models, such as the cheap inference and efficient training. Specifically, a CRBM is defined by clamping the states of an *input* subset of the visible units of an RBM. For each input state one obtains a conditioned distribution over the states of the *output* visible units. See Figure 1 for an illustration of this architecture. This kind of conditional models and slight variants thereof have seen success in many applications; for example, in classification (Larochelle and Bengio, 2008), collaborative filtering (Salakhutdinov et al., 2007), motion modeling (Taylor et al., 2007; Zeiler et al., 2009; Mnih et al., 2011; Sutskever and Hinton, 2007), and reinforcement learning (Sallans and Hinton, 2004).

So far, however, there is not much theoretical work addressing the expressive power of CRBMs. We note that it is relatively straightforward to obtain some results on the expressive power of CRBMs from the existing theoretical work on RBM probability models. Nevertheless, an accurate analysis requires to take into account the specificities of the conditional case. Formally, a CRBM is a collection of RBMs, with one RBM for each possible input value. These RBMs differ in the biases of the hidden units, as these are influenced by the input values. However, these hidden biases are not independent for all different inputs, and, moreover, the same interaction weights and biases of the visible units are shared for all different inputs. This sharing of parameters draws a substantial distinction of CRBM models from independent tuples of RBM models.

In this paper we address the representational power of CRBMs, contributing theoretical insights to the optimal number of hidden units. Our focus lies on the classes of conditional distributions that can possibly be represented by a CRBM with a fixed number of inputs and outputs, depending on the number of hidden units. Having said this, we do not discuss the problem of finding the optimal parameters that give rise to a desired conditional distribution (although our derivations include an algorithm that does this), nor problems related to incomplete knowledge of the target conditional distributions and generalization errors. A number of training methods for CRBMs have been discussed in the references listed above, depending on the concrete applications. The problems that we deal with here are the following: 1) are distinct parameters of the model mapped to distinct conditional

distributions; what is the smallest number of hidden units that suffices for obtaining a model that can 2) approximate any target conditional distribution arbitrarily well (a universal approximator); 3) approximate any target conditional distribution without exceeding a given error tolerance; 4) approximate selected classes of conditional distributions arbitrarily well? We provide non-trivial solutions to all of these problems. We focus on the case of binary units, but the main ideas extend to the case of discrete non-binary units.

This paper is organized as follows. Section 2 contains formal definitions and elementary properties of CRBMs. Section 3 investigates the geometry of CRBM models in three subsections. In Section 3.1 we study the dimension of the sets of conditional distributions represented by CRBMs and show that in most cases this is the dimension expected from counting parameters (Theorem 4). In Section 3.2 we address the universal approximation problem, deriving upper and lower bounds on the minimal number of hidden units that suffices for this purpose (Theorem 7). In Section 3.3 we analyze the maximal approximation errors of CRBMs (assuming optimal parameters) and derive an upper bound for the minimal number of hidden units that suffices to approximate every conditional distribution within a given error tolerance (Theorem 11). Section 4 investigates the expressive power of CRBMs in two subsections. In Section 4.1 we describe how CRBMs can represent natural families of conditional distributions that arise in Markov random fields (Theorem 14). In Section 4.2 we study the ability of CRBMs to approximate conditional distributions with restricted supports. This section addresses, especially, the approximation of deterministic conditional distributions (Theorem 21). In Section 5 we offer a discussion and an outlook. In order to present the main results in a concise way, we have deferred all proofs to the appendices. Nonetheless, we think that the proofs are interesting in their own right, and we have prepared them with a fair amount of detail.

2. Definitions

We will denote the set of probability distributions on $\{0,1\}^n$ by Δ_n. A probability distribution $p \in \Delta_n$ is a vector of 2^n non-negative entries $p(y)$, $y \in \{0,1\}^n$, adding to one, $\sum_{y\in\{0,1\}^n} p(y) = 1$. The set Δ_n is a (2^n-1)-dimensional simplex in $\mathbb{R}^{2^n}$.

We will denote the set of conditional distributions of a variable $y \in \{0,1\}^n$, given another variable $x \in \{0,1\}^k$, by $\Delta_{k,n}$. A conditional distribution $p(\cdot|\cdot) \in \Delta_{k,n}$ is a $2^k \times 2^n$ row-stochastic matrix with rows $p(\cdot|x) \in \Delta_n$, $x \in \{0,1\}^k$. The set $\Delta_{k,n}$ is a $2^k(2^n-1)$-dimensional polytope in $\mathbb{R}^{2^k\times 2^n}$. It can be regarded as the 2^k-fold Cartesian product $\Delta_{k,n} = \Delta_n \times \cdots \times \Delta_n$, where there is one probability simplex Δ_n for each possible input state $x \in \{0,1\}^k$. We will use the abbreviation $[N] := \{1,\ldots,N\}$, where N is a natural number.

Definition 1 The conditional restricted Boltzmann machine (CRBM) with k input units, n output units, and m hidden units, denoted $\mathrm{RBM}^k_{n,m}$, is the set of all conditional distributions in $\Delta_{k,n}$ that can be written as

$$p(y|x) = \frac{1}{Z(W,b,Vx+c)} \sum_{z\in\{0,1\}^m} \exp(z^\top Vx + z^\top Wy + b^\top y + c^\top z), \quad \forall y \in \{0,1\}^n, x \in \{0,1\}^k,$$

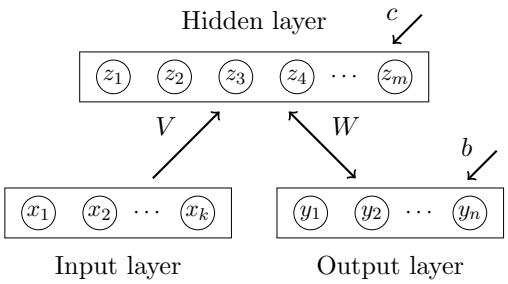


Figure 1: Architecture of a CRBM. An RBM is the special case with $k = 0$.

with normalization function

$$Z(W, b, Vx + c) = \sum_{y \in \{0,1\}^n} \sum_{z \in \{0,1\}^m} \exp(z^\top Vx + z^\top Wy + b^\top y + c^\top z), \quad \forall x \in \{0,1\}^k.$$

Here, x, y, and z are column state vectors of the k input units, n output units, and m hidden units, respectively, and $^\top$ denotes transposition. The parameters of this model are the matrices of interaction weights $V \in \mathbb{R}^{m \times k}$, $W \in \mathbb{R}^{m \times n}$ and the vectors of biases $b \in \mathbb{R}^n$, $c \in \mathbb{R}^m$. When there are no input units ($k = 0$), the model $\mathrm{RBM}^k_{n,m}$ reduces to the restricted Boltzmann machine probability model with n visible units and m hidden units, denoted $\mathrm{RBM}_{n,m}$.

We can view $\mathrm{RBM}^k_{n,m}$ as a collection of 2^k restricted Boltzmann machine probability models with shared parameters. For each input $x \in \{0,1\}^k$, the output distribution $p(\cdot|x)$ is the probability distribution represented by $\mathrm{RBM}_{n,m}$ for the parameters $W, b, (Vx + c)$. All $p(\cdot|x)$ have the same interaction weights W, the same biases b for the visible units, and differ only in the biases $(Vx + c)$ for the hidden units. The joint behavior of these distributions with shared parameters is not trivial.

The model $\mathrm{RBM}^k_{n,m}$ can also be regarded as representing block-wise normalized versions of the joint probability distributions represented by $\mathrm{RBM}_{n+k,m}$. Namely, a joint distribution $p \in \mathrm{RBM}_{n+k,m} \subseteq \Delta_{k+n}$ is an array with entries $p(x, y)$, $x \in \{0,1\}^k$, $y \in \{0,1\}^n$. Conditioning p on x is equivalent to considering the normalized x-th row $p(y|x) = p(x, y)/\sum_{y'} p(x, y')$, $y \in \{0,1\}^n$.

3. Geometry of Conditional Restricted Boltzmann Machines

In this section we investigate three basic questions about the geometry of CRBM models. First, what is the dimension of a CRBM model? Second, how many hidden units does a CRBM need in order to be able to approximate every conditional distribution arbitrarily well? Third, how accurate are the approximations of a CRBM, depending on the number of hidden units?

3.1 Dimension

The model $\mathrm{RBM}^k_{n,m}$ is defined by marginalizing out the hidden units of a graphical model. This implies that several choices of parameters may represent the same conditional distri-

butions. In turn, the dimension of the set of representable conditional distributions may be smaller than the number of model parameters, in principle.

When the dimension of $\mathrm{RBM}^k_{n,m}$ is equal to $\min\{(k+n)m+n+m, 2^k(2^n-1)\}$, which is the minimum of the number of parameters and the dimension of the ambient polytope of conditional distributions, the CRBM model is said to have the *expected dimension*. In this section we show that $\mathrm{RBM}^k_{n,m}$ has the expected dimension for most triplets (k,n,m). In particular, we show that this holds in all practical cases, where the number of hidden units m is smaller than exponential with respect to the number of visible units $k+n$.

The dimension of a parametric model is given by the maximum of the rank of the Jacobian of its parameterization (assuming mild differentiability conditions). Computing the rank of the Jacobian is not easy in general. A resort is to compute the rank only in the limit of large parameters, which corresponds to considering a piece-wise linearized version of the original model, called the *tropical model*. Cueto et al. (2010) used this approach to study the dimension of RBM probability models. Here we apply their ideas to address the dimension of CRBM conditional models.

The following functions from coding theory will be useful for phrasing the results:

Definition 2 Let $A(n,d)$ denote the cardinality of a largest subset of $\{0,1\}^n$ whose elements are at least Hamming distance d apart. Let $K(n,d)$ denote the smallest cardinality of a set such that every element of $\{0,1\}^n$ is at most Hamming distance d apart from that set.

Cueto et al. (2010) showed that $\dim(\mathrm{RBM}_{n,m}) = nm+n+m$ for $m+1 \leq A(n,3)$, and $\dim(\mathrm{RBM}_{n,m}) = 2^n-1$ for $m \geq K(n,1)$. It is known that $A(n,3) \geq 2^{n-\lceil \log_2(n+1)\rceil}$ and $K(n,1) \leq 2^{n-\lfloor \log_2(n+1)\rfloor}$. In turn, for most pairs (n,m) the probability model $\mathrm{RBM}_{n,m}$ has the expected dimension (although for many values of n there is a range of values of m where the results are inconclusive about this). Noting that $\dim(\mathrm{RBM}^k_{n,m}) \geq \dim(\mathrm{RBM}_{k+n,m}) - (2^k-1)$, these results on the dimension of RBM probability models directly imply following bounds on the dimension of CRBM models:

Proposition 3 *The conditional model* $\mathrm{RBM}^k_{n,m}$ *satisfies the following:*

- $\dim(\mathrm{RBM}^k_{n,m}) \geq (n+k)m+n+m+k-(2^k-1)$ *for* $m+1 \leq A(k+n,3)$.
- $\dim(\mathrm{RBM}^k_{n,m}) = 2^k(2^n-1)$ *for* $m \geq K(k+n,1)$.

This result shows that, when $m \geq K(k+n,1)$, the CRBM model has the maximum possible dimension, equal to the dimension of $\Delta_{k,n}$. In all other cases, however, the dimension bounds are too loose and do not allow us to conclude whether or not the CRBM model has the expected dimension. Hence we need to study the conditional model in more detail. We obtain the following result:

Theorem 4 *The conditional model* $\mathrm{RBM}^k_{n,m}$ *satisfies the following:*

- $\dim(\mathrm{RBM}^k_{n,m}) = (k+n)m+n+m$ *for* $m+1 \leq A(k+n,4)$.
- $\dim(\mathrm{RBM}^k_{n,m}) = 2^k(2^n-1)$ *for* $m \geq K(k+n,1)$.

We note the following practical version of the theorem, which results from inserting appropriate bounds on the functions A and K:

Corollary 5 *The conditional model* $\mathrm{RBM}^k_{n,m}$ *has the expected dimension in the following cases:*

- $\dim(\mathrm{RBM}^k_{n,m}) = (n+k)m + n + m$ *for* $m \leq 2^{(k+n)-\lfloor \log_2((k+n)^2-(k+n)+2)\rfloor}$.
- $\dim(\mathrm{RBM}^k_{n,m}) = 2^k(2^n-1)$ *for* $m \geq 2^{(k+n)-\lfloor \log_2(k+n+1)\rfloor}$.

These results show that, in all cases of practical interest, where m is less than exponential in $k+n$, the dimension of the CRBM model is indeed equal to the number of model parameters. In all these cases, almost every conditional distribution that can be represented by the model is represented by at most finitely many different choices of parameters. We should note that there is an interval of exponentially large values of m where the results remain inconclusive, namely the interval $A(k+n,4) \leq m < K(k+n,1)$. This is similar to the gap already mentioned above for RBM probability models and poses interesting theoretical problems (see also Montúfar and Morton, 2015).

On the other hand, the dimension alone is not very informative about the ability of a model to approximate target distributions. In particular, it may be that a high dimensional model covers only a tiny fraction of the set of all conditional distributions, or also that a low dimensional model can approximate any target conditional relatively well. We address the minimal dimension and number of parameters of a universal approximator in the next section. In the subsequent section we address the approximation errors depending on the number of parameters.

3.2 Universal Approximation

In this section we ask for the smallest number of hidden units m for which the model $\mathrm{RBM}^k_{n,m}$ can approximate every conditional distribution from $\Delta_{k,n}$ arbitrarily well.

Note that each conditional distribution $p(y|x)$ can be identified with the set of joint distributions of the form $r(x,y) = q(x)p(y|x)$, with strictly positive marginals $q(x)$. In particular, by fixing a marginal distribution, we obtain an identification of $\Delta_{k,n}$ with a subset of Δ_{k+n}. Figure 2 illustrates this identification in the case $n = k = 1$ and $q(0) = q(1) = \frac{1}{2}$.

This implies that universal approximators of joint probability distributions define universal approximators of conditional distributions. We know that $\mathrm{RBM}_{n+k,m}$ is a universal approximator whenever $m \geq \frac{1}{2}2^{k+n} - 1$ (see Montúfar and Ay, 2011), and therefore:

Proposition 6 *The model* $\mathrm{RBM}^k_{n,m}$ *can approximate every conditional distribution from* $\Delta_{k,n}$ *arbitrarily well whenever* $m \geq \frac{1}{2}2^{k+n} - 1$.

This improves previous results by Younes (1996) and van der Maaten (2011). On the other hand, since conditional models do not need to model the input distributions, in principle it is possible that $\mathrm{RBM}^k_{n,m}$ is a universal approximator even if $\mathrm{RBM}_{n+k,m}$ is not a universal approximator. In fact, we obtain the following improvement of Proposition 6, which does not follow from corresponding results for RBM probability models:

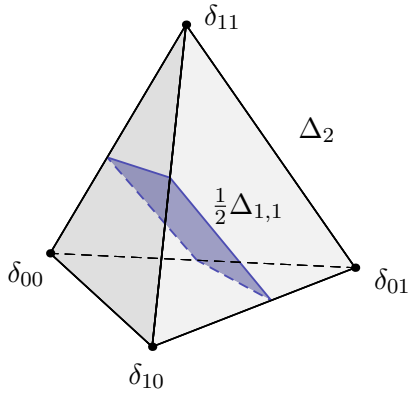


Figure 2: The polytope of conditional distributions $\Delta_{1,1}$ embedded in the simplex of probability distributions Δ_2.

Theorem 7 *The model* $\mathrm{RBM}^k_{n,m}$ *can approximate every conditional distribution from* $\Delta_{k,n}$ *arbitrarily well whenever*

$$m \geq \begin{cases} \frac{1}{2}2^k(2^n-1), & \text{if } k \geq 1 \\ \frac{3}{8}2^k(2^n-1)+1, & \text{if } k \geq 3 \\ \frac{1}{4}2^k(2^n-1+1/30), & \text{if } k \geq 21 \end{cases}.$$

In fact, the model $\mathrm{RBM}^k_{n,m}$ *can approximate every conditional distribution from* $\Delta_{k,n}$ *arbitrarily well whenever* $m \geq 2^k K(r)(2^n-1)+2^{S(r)}P(r)$*, where* r *is any natural number satisfying* $k \geq 1+\cdots+r =: S(r)$*, and* K *and* P *are functions (defined in Lemma 30 and Proposition 32) which tend to approximately* 0.2263 *and* 0.0269*, respectively, as* r *tends to infinity.*

We note the following weaker but practical version of Theorem 7:

Corollary 8 *Let* $k \geq 1$*. The model* $\mathrm{RBM}^k_{n,m}$ *can approximate every conditional distribution from* $\Delta_{k,n}$ *arbitrarily well whenever* $m \geq \frac{1}{2}2^k(2^n-1) = \frac{1}{2}2^{k+n} - \frac{1}{2}2^k$*.*

These results are significant, because they reduce the bounds following from universal approximation results for probability models by an additive term of order 2^k, which corresponds precisely to the order of parameters needed to model the input distributions.

As expected, the asymptotic behavior of the theorem's bound is exponential in the number of input and output units. This lies in the nature of the universal approximation property. A crude lower bound on the number of hidden units that suffices for universal approximation can be obtained by comparing the number of parameters of the model and the dimension of the conditional polytope:

Proposition 9 *If the model* $\mathrm{RBM}^k_{n,m}$ *can approximate every conditional distribution from* $\Delta_{k,n}$ *arbitrarily well, then necessarily* $m \geq \frac{1}{(n+k+1)}(2^k(2^n-1)-n)$*.*

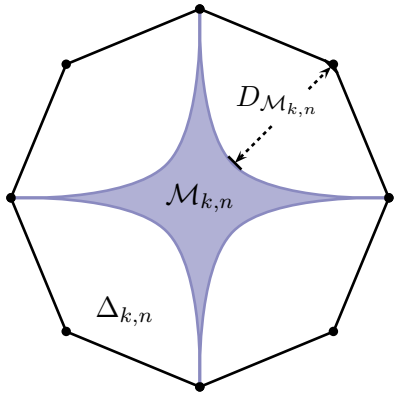


Figure 3: Schematic illustration of the maximal approximation error of a model of conditional distributions $\mathcal{M}_{k,n} \subseteq \Delta_{k,n}$.

The results presented above highlight the fact that CRBM universal approximation may be possible with a drastically smaller number of hidden units than RBM universal approximation, for the same number of visible units. However, even with these reductions the universal approximation property requires an enormous number of hidden units. In order to provide a more informative description of the approximation capabilities of CRBMs, in the next section we investigate how the maximal approximation error decreases as hidden units are added to the model.

3.3 Maximal Approximation Errors

From a practical perspective it is not necessary to approximate conditional distributions arbitrarily well, but fair approximations suffice. This can be especially important if the number of required hidden units grows disproportionately with the quality of the approximation. In this section we investigate the maximal approximation errors of CRBMs depending on the number of hidden units. Figure 3 gives a schematic illustration of the maximal approximation error of a conditional model.

The Kullback-Leibler divergence of two probability distributions p and q in Δ_{k+n} is given by

$$\begin{aligned} D(p\|q) :=& \sum_x \sum_y p(x)p(y|x) \log \frac{p(x)p(y|x)}{q(x)q(y|x)} \\ =& D(p_X\|q_X) + \sum_x p(x) D(p(\cdot|x)\|q(\cdot\|x)), \end{aligned}$$

where $p_X = \sum_{y\in\{0,1\}^n} p(x,y)$ denotes the marginal distribution over $x \in \{0,1\}^k$. The divergence of two conditional distributions $p(\cdot|\cdot)$ and $q(\cdot|\cdot)$ in $\Delta_{k,n}$ is given by

$$D(p(\cdot|\cdot)\|q(\cdot|\cdot)) := \sum_x u_X(x) D(p(\cdot|x)\|q(\cdot|x)),$$

where u_X denotes the uniform distribution over x. Even if the divergence between two joint distributions does not vanish, the divergence between their conditional distributions may vanish.

Consider a model $\mathcal{M}_{k+n} \subseteq \Delta_{k+n}$ of joint probability distributions and a corresponding model $\mathcal{M}_{k,n} \subseteq \Delta_{k,n}$ of conditional distributions. More precisely, $\mathcal{M}_{k,n}$ consists of all conditional distributions of the form $q(y|x) = q(x,y)/\sum_{y' \in \{0,1\}^n} q(x,y')$, for all $y \in \{0,1\}^n$ and $x \in \{0,1\}^k$, where $q(\cdot,\cdot)$ is any joint probability distribution from $\mathcal{M}_{k+n}$ satisfying $\sum_{y' \in \{0,1\}^n} q(x,y') > 0$, for all $x \in \{0,1\}^k$. The divergence from a conditional distribution $p(\cdot|\cdot) \in \Delta_{k,n}$ to the model $\mathcal{M}_{k,n}$ is given by

$$D(p(\cdot|\cdot)\|\mathcal{M}_{k,n}) := \inf_{q \in \mathcal{M}_{k,n}} D(p(\cdot|\cdot)\|q(\cdot|\cdot)) = \inf_{q \in \mathcal{M}_{k+n}} D(u_X p(\cdot|\cdot)\|q) - D(u_X\|q_X).$$

In turn, the maximum of the divergence from a conditional distribution to $\mathcal{M}_{k,n}$ satisfies

$$D_{\mathcal{M}_{k,n}} := \max_{p(\cdot|\cdot) \in \Delta_{k,n}} D(p(\cdot|\cdot)\|\mathcal{M}_{k,n}) \leq \max_{p \in \Delta_{k+n}} D(p\|\mathcal{M}_{k+n}) =: D_{\mathcal{M}_{k+n}}.$$

Hence we can bound the maximal divergence of a CRBM by the maximal divergence of an RBM (studied by Montúfar et al., 2011) and obtain the following:

Proposition 10 *If $m \leq 2^{(n+k)-1} - 1$, then the divergence from any conditional distribution $p(\cdot|\cdot) \in \Delta_{k,n}$ to the model $\mathrm{RBM}^k_{n,m}$ is bounded by*

$$D_{\mathrm{RBM}^k_{n,m}} \leq D_{\mathrm{RBM}_{k+n,m}} \leq (n+k) - \lfloor \log_2(m+1) \rfloor - \frac{m+1}{2^{\lfloor \log_2(m+1) \rfloor}}.$$

This proposition implies the universal approximation result from Proposition 6 as the special case with vanishing approximation error, but it does not imply Theorem 7 in the same way. Taking more specific properties of the conditional model into account, we can improve the proposition and obtain the following:

Theorem 11 *Let $l \in [n]$. The divergence from any conditional distribution in $\Delta_{k,n}$ to the model $\mathrm{RBM}^k_{n,m}$ is bounded from above by*

$$D_{\mathrm{RBM}^k_{n,m}} \leq n - l, \quad \text{whenever } m \geq \begin{cases} \frac{1}{2}2^k(2^l - 1), & \text{if } k \geq 1 \\ \frac{3}{8}2^k(2^l - 1) + 1, & \text{if } k \geq 3 \\ \frac{1}{4}2^k(2^l - 1 + 1/30), & \text{if } k \geq 21 \end{cases}.$$

In fact, the divergence from any conditional distribution in $\Delta_{k,n}$ to $\mathrm{RBM}^k_{n,m}$ is bounded from above by $D_{\mathrm{RBM}^k_{n,m}} \leq n - l$, where l is the largest integer with $m \geq 2^{k - S(r)}F(r)(2^l - 1) + R(r)$.

In plain terms, this theorem shows that the worst case approximation errors of CRBMs decrease at least with the logarithm of the number of hidden units. Given an error tolerance, we can use these bounds to find a sufficient number of hidden units that guarantees approximations within this error tolerance. Furthermore, the result implies the universal approximation result from Theorem 7 as the special case with vanishing approximation error. We note the following weaker but practical version of Theorem 11 (analogue to Corollary 8):

Corollary 12 *Let $k \geq 1$ and $l \in [n]$. The divergence from any conditional distribution in $\Delta_{k,n}$ to the model $\mathrm{RBM}^k_{n,m}$ is bounded from above by $D_{\mathrm{RBM}^k_{n,m}} \leq n - l$, whenever $m \geq \frac{1}{2}2^k(2^l - 1)$.*

In this section we have discussed the worst case approximation errors of CRBMs. On the other hand, in practice one is not interested in approximating all possible conditional distributions, but only special classes. One can expect that CRBMs can approximate certain classes of conditional distributions better than others. This is the subject of the next section.

4. Representation of Special Classes of Conditional Models

In this section we ask about the classes of conditional distributions that can be compactly represented by CRBMs and whether CRBMs can approximate interesting conditional distributions using only a moderate number of hidden units.

The first part of the question is about familiar classes of conditional distributions that can be expressed in terms of CRBMs, which in turn would allow us to compare CRBMs with other models and to develop a more intuitive picture of Definition 1.

The second part of the question clearly depends on the specific problem at hand. Nonetheless, some classes of conditional distributions may be considered generally interesting, as they contain solutions to all instances of certain classes of problems. An example is the class of deterministic conditional distributions, which suffices to solve any Markov decision problem in an optimal way.

4.1 Representation of Conditional Markov Random Fields

In this section we discuss the ability of CRBMs to represent conditional Markov random fields, depending on the number of hidden units that they have. The main idea is that each hidden unit of an RBM can be used to model the pure interaction of a group of visible units. This idea appeared in previous work by Younes (1996), in the context of universal approximation.

Definition 13 *Consider a simplicial complex I on $[N]$; that is, a collection of subsets of $[N] = \{1, \dots, N\}$ such that $A \in I$ implies $B \in I$ for all $B \subseteq A$ (in particular $\emptyset \in I$). The random field $\mathcal{E}_I \subseteq \Delta_N$ with interactions I is the set of probability distributions of the form*

$$p(x) = \frac{1}{Z} \exp\Big(\sum_{A \in I} \theta_A \prod_{i \in A} x_i \Big), \quad \text{for all } x = (x_1, \dots, x_N) \in \{0,1\}^N,$$

with normalization $Z = \sum_{x' \in \{0,1\}^N} \exp(\sum_{A \in I} \theta_A \prod_{i \in A} x'_i)$ and parameters $\theta_A \in \mathbb{R}$, $A \in I$.

Given a set S, we will denote the set of all subsets of S by 2^S. We obtain the following result:

Theorem 14 *Let I be a simplicial complex on $[k+n]$ and let $J = 2^{[k]} \cup \{\{k+1\}, \dots, \{k+n\}\}$. If $m \geq |I \setminus J|$, then the model $\mathrm{RBM}^k_{n,m}$ can represent every conditional distribution of $(x_{k+1}, \dots, x_{k+n})$, given $(x_1, \dots, x_k)$, that can be represented by $\mathcal{E}_I \subseteq \Delta_{k+n}$.*

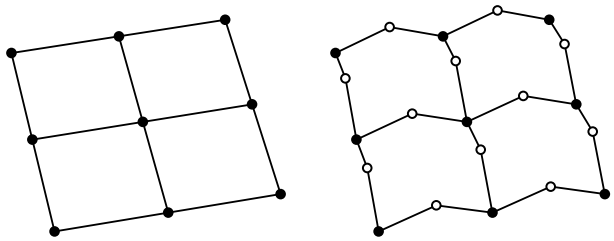


Figure 4: Example of a Markov random field and a corresponding RBM architecture that can represent it. Visible units are depicted in black and hidden units in white.

An interesting special case is when each output distribution can be chosen arbitrarily from a given Markov random field:

Corollary 15 *Let I be a simplicial complex on $[n]$ and for each $x \in \{0,1\}^n$ let p^x be some probability distribution from $\mathcal{E}_I \subseteq \Delta_n$. If $m \geq 2^k(|I| - 1) - |\{A \in I \colon |A| = 1\}|$, then the model $\mathrm{RBM}^k_{n,m}$ can represent the conditional distribution defined by $q(y|x) = p^x(y)$, for all $y \in \{0,1\}^n$, for all $x \in \{0,1\}^k$.*

We note the following direct implication for RBM probability models:

Corollary 16 *Let I be a simplicial complex on $[n]$. If $m \geq |\{A \in I \colon |A| > 1\}|$, then $\mathrm{RBM}_{n,m}$ can represent any probability distribution p from $\mathcal{E}_I$.*

Figure 4 illustrates a Markov random field and an RBM that can represent it.

4.2 Approximation of Conditional Distributions with Restricted Supports

In this section we continue the discussion about the classes of conditional distributions that can be represented by CRBMs, depending on the number of hidden units. Here we focus on a hierarchy of conditional distributions defined by the total number of input-output pairs with positive probability.

Definition 17 *For any k, n, and $0 \leq d \leq 2^k(2^n - 1)$, let $C_{k,n}(d) \subseteq \Delta_{k,n}$ denote the union of all d-dimensional faces of $\Delta_{k,n}$; that is, the set of conditional distributions that have a total of $2^k + d$ or fewer non-zero entries, $C_{k,n}(d) := \{p(\cdot|\cdot) \in \Delta_{k,n} \colon |\{(x,y) \colon p(y|x) > 0\}| \leq 2^k + d\}$.*

Note that $C_{k,n}(2^k(2^n - 1)) = \Delta_{k,n}$. The vertices (zero-dimensional faces) of $\Delta_{k,n}$ are the conditional distributions which assign positive probability to only one output, given each input, and are called *deterministic*. By Carathéodory's theorem, every element of $C_{k,n}(d)$ is a convex combination of $(d+1)$ or fewer deterministic conditional distributions.

The sets $C_{k,n}(d)$ arise naturally in the context of reinforcement learning and partially observable Markov decision processes (POMDPs). Namely, every finite POMDP has an associated effective dimension d, which is the dimension of the set of all state processes that can be generated by stationary stochastic policies. Montúfar et al. (2015) showed that

the policies represented by conditional distributions from the set $C_{k,n}(d)$ are sufficient to generate all the processes that can be generated by $\Delta_{k,n}$. In general, the effective dimension d is relative small, such that $C_{k,n}(d)$ is a much smaller policy search space than $\Delta_{k,n}$.

We have the following result:

Proposition 18 *If $m \geq 2^k + d - 1$, then the model $\mathrm{RBM}^k_{n,m}$ can approximate every element from $C_{k,n}(d)$ arbitrarily well.*

This result shows the intuitive fact that each hidden unit can be used to model the probability of an input-output pair. Since each conditional distribution has 2^k input-output probabilities that are completely determined by the other probabilities (due to normalization), it is interesting to ask whether the amount of hidden units indicated in Proposition 18 is strictly necessary. Further below, Theorem 21 will show that, indeed, hidden units are required for modeling the positions of the positive probability input-output pairs, even if their specific values do not need to be modeled.

We note that certain structures of positive probability input-output pairs can be modeled with fewer hidden units than stated in Proposition 18. An simple example is the following direct generalization of Corollary 8:

Proposition 19 *If d is divisible by 2^k and $m \geq d/2$, then the model $\mathrm{RBM}^k_{n,m}$ can approximate every element from $C_{k,n}(d)$ arbitrarily well, when the set of positive-probability outputs is the same for all inputs.*

In the following we will focus on deterministic conditional distributions. This is a particularly interesting and simple class of conditional distributions with restricted supports. It is well known that any finite Markov decision processes (MDPs) has an optimal policy defined by a stationary deterministic conditional distribution (see Bellman, 1957; Ross, 1983). Furthermore, Ay et al. (2013) showed that it is always possible to define simple two-dimensional manifolds that approximate all deterministic conditional distributions arbitrarily well.

Certain classes of conditional distributions (in particular deterministic conditionals) coming from feedforward networks can be approximated arbitrarily well by CRBMs. We use the following definitions. A *linear threshold unit* with inputs $x \in \{0,1\}^k$ is a function that outputs 1 when $\sum_j V_{ij} x_j + c_i > 0$, and outputs 0 otherwise. A *sigmoid belief unit* with inputs $z \in \{0,1\}^m$ is a stochastic function that outputs 1 with probability $p(y_i = 1|z) = \sigma(\sum_j W_{ij} z_j + b_i)$, where $\sigma(s) = \frac{1}{1+\exp(-s)}$, and outputs 0 with complementary probability.

Theorem 20 *The model $\mathrm{RBM}^k_{n,m}$ can approximate every conditional distribution arbitrarily well, which can be represented by a feedforward network with k input units, a hidden layer of m linear threshold units, and an output layer of n sigmoid belief units. In particular, the model $\mathrm{RBM}^k_{n,m}$ can approximate every deterministic conditional distribution from $\Delta_{k,n}$ arbitrarily well, which can be represented by a feedforward linear threshold network with k input, m hidden, and n output units.*

The representational power of feedforward linear threshold networks has been studied intensively in the literature. For example, Wenzel et al. (2000) showed that a feedforward linear threshold network with $k \geq 1$ input, m hidden, and $n = 1$ output units, can represent the following:

- Any Boolean function $f\colon \{0,1\}^k \to \{0,1\}$, when $m \geq 3 \cdot 2^{k-1-\lfloor \log_2(k+1) \rfloor}$; e.g., when $m \geq \frac{3}{k+2} 2^k$.
- The parity function $f_{\text{parity}}\colon \{0,1\}^k \to \{0,1\};\ x \mapsto \sum_i x_i \mod 2$, when $m \geq k$.
- The indicator function of any union of m linearly separable subsets of $\{0,1\}^k$.

Although CRBMs can approximate this rich class of deterministic conditional distributions arbitrarily well, the next result shows that the number of hidden units required for universal approximation of deterministic conditional distributions is rather large:

Theorem 21 *The model* $\text{RBM}^k_{n,m}$ *can approximate every deterministic conditional distribution from* $\Delta_{k,n}$ *arbitrarily well if* $m \geq \min\left\{2^k - 1, \frac{3n}{k+2} 2^k\right\}$ *and only if* $m \geq 2^{k/2} - \frac{(n+k)^2}{2n}$.

This theorem refines the statement of Proposition 18 in the special case $d = 0$. By this theorem, in order to approximate all deterministic conditional distributions arbitrarily well, a CRBM requires exponentially many hidden units, with respect to the number of input units.

5. Conclusion

This paper gives a theoretical description of the representational capabilities of conditional restricted Boltzmann machines (CRBMs) relating model complexity and model accuracy. CRBMs are based on the well studied restricted Boltzmann machine (RBM) probability models. We proved an extensive series of results that generalize recent theoretical work on the representational power of RBMs in a non-trivial way.

We studied the problem of parameter identifiability. We showed that every CRBM with up to exponentially many hidden units (in the number of input and output units) represent a set of conditional distributions of dimension equal to the number of model parameters. This implies that in all practical cases, CRBMs do not waste parameters, and, generically, only finitely many choices of the interaction weights and biases produce the same conditional distribution.

We addressed the classical problems of universal approximation and approximation quality. Our results show that a CRBM with m hidden units can approximate every conditional distribution of n output units, given k input units, without surpassing a Kullback-Leibler approximation error of the form $n - \log_2(m/2^{k-1} + 1)$ (assuming optimal parameters). Thus this model is a universal approximator whenever $m \geq \frac{1}{2} 2^k (2^n - 1)$. In fact we provided tighter bounds depending on k. For instance, if $k \geq 21$, then the universal approximation property is attained whenever $m \geq \frac{1}{4} 2^k (2^n - 29/30)$. Our proof is based on an upper bound for the complexity of an algorithm that packs Boolean cubes with sequences of non-overlapping stars, for which improvements may be possible. It is worth mentioning that the set of conditional distributions for which the approximation error is maximal may be very small. This is a largely open and difficult problem. We note that our results can be plugged into certain analytic integrals (see Montúfar and Rauh, 2014) to produce upper-bounds for the expectation value of the approximation error when approximating conditional distributions drawn from a product Dirichlet density on the polytope of all conditional distributions.

For future work it would be interesting to extend our (optimal-parameter) considerations by an analysis of the CRBM training complexity and the errors resulting from non-optimal parameter choices.

We also studied specific classes of conditional distributions that can be represented by CRBMs, depending on the number of hidden units. We showed that CRBMs can represent conditional Markov random fields by using each hidden unit to model the interaction of a group of visible variables. Furthermore, we showed that CRBMs can approximate all binary functions with k input bits and n output bits arbitrarily well if $m \geq 2^k - 1$ or $m \geq \frac{3n}{k+2} 2^k$ and only if $m \geq 2^{k/2} - (n+k)^2/2n$. In particular, this implies that there are exponentially many deterministic conditional distributions which can only be approximated arbitrarily well by a CRBM if the number of hidden units is exponential in the number of input units. This aligns with well known examples of functions that cannot be compactly represented by shallow feedforward networks, and reveals some of the intrinsic constraints of CRBM models that may prevent them from grossly over-fitting.

We think that the developed techniques can be used for studying other conditional probability models as well. In particular, for future work it would be interesting to compare the representational power of CRBMs and of combinations of CRBMs with feedforward nets (combined models of this kind include CRBMs with retroactive connections and recurrent temporal RBMs). Also, it would be interesting to apply our techniques to study stacks of CRBMs and other multilayer conditional models. Finally, although our analysis focuses on the case of binary units, the main ideas can be extended to the case of discrete non-binary units.

Acknowledgments

We are grateful to anonymous reviewers for helpful comments. We acknowledge support from the DFG Priority Program Autonomous Learning (DFG-SPP 1527). G. M. and K. G.-Z. would like to thank the Santa Fe Institute for hosting them during the initial work on this article.

Appendix A. Details on the Dimension

Proof of Proposition 3 Each joint distribution of x and y has the form $p(x,y) = p(x)p(y|x)$ and the set Δ_k of all marginals $p(x)$ has dimension $2^k - 1$. The items follow directly from the corresponding statements for the probability model (Cueto et al., 2010). ■

We will need two standard definitions from coding theory:

Definition 22 Let r and k be two natural numbers with $r \leq k$. A radius-r *Hamming ball* in $\{0,1\}^k$ is a set B consisting of a length-k binary vector, together with all other length-k binary vectors that are at most Hamming distance r apart from that vector; that is, $B = \{x \in \{0,1\}^k \colon d_H(x,z) \leq r\}$ for some $z \in \{0,1\}^k$, where $d_H(x,z) := |\{i \in [k] \colon x_i \neq z_i\}|$ denotes the Hamming distance between x and z. Here $[k] := \{1, \ldots, k\}$.

Definition 23 An r-dimensional *cylinder set* in $\{0,1\}^k$ is a set C of length-k binary vectors with arbitrary values in r coordinates and fixed values in the other coordinates; that is, $C = \{x \in \{0,1\}^k \colon x_i = z_i \text{ for all } i \in \Lambda\}$ for some $z \in \{0,1\}^k$ and some $\Lambda \subseteq [k]$ with $k - |\Lambda| = r$.

The geometric intuition is simple: a cylinder set corresponds to the vertices of a face of a unit cube. A radius-1 Hamming ball corresponds to the vertices of a corner of a unit cube. The vectors in a radius-1 Hamming ball are affinely independent. See Figure 5A for an illustration.

Proof of Theorem 4 The proof is based on the ideas developed by Cueto et al. (2010) for studying the RBM probability model. We prove a stronger (more technical) statement than the one given in the theorem: The set $\{0,1\}^{k+n}$ contains m disjoint radius-1 Hamming balls whose union does not contain any set of the form $[x] := \{(x,y) \in \{0,1\}^{k+n} \colon y \in \{0,1\}^n\}$ for $x \in \{0,1\}^k$, and whose complement has full affine rank, as a subset of $\mathbb{R}^{k+n}$.

We consider the Jacobian of $\mathrm{RBM}^k_{n,m}$ for the parameterization given in Definition 1. The dimension of $\mathrm{RBM}^k_{n,m}$ is the maximum rank of the Jacobian over all possible choices of $\theta = (W, V, b, c) \in \mathbb{R}^N$, $N = n + m + (n+k)m$. Let $h_\theta(v) := \operatorname{argmax}_{z \in \{0,1\}^m} p(z|v)$ denote the most likely hidden state of $\mathrm{RBM}_{k+n,m}$ given the visible state $v = (x,y)$, depending on the parameter θ. After a few direct algebraic manipulations, we find that the maximum rank of the Jacobian is bounded from below by the maximum over θ of the dimension of the column-span of the matrix $\mathcal{A}_\theta$ with rows

$$\left((1, x^\top, y^\top), (1, x^\top, y^\top) \otimes h_\theta(x,y)^\top\right), \quad \text{for all } (x,y) \in \{0,1\}^{k+n},$$

modulo vectors whose (x,y)-th entries are independent of y given x. Here $\otimes$ is the Kronecker product, which is defined by $(a_{ij})_{i,j} \otimes (b_{kl})_{k,l} = (a_{ij} b_{kl})_{ik,jl}$. The modulo operation has the effect of disregarding the input distribution $p(x)$ in the joint distribution $p(x,y) = p(x)p(y|x)$ represented by the RBM. For example, from the first block of $\mathcal{A}_\theta$ we can remove the columns that correspond to x, without affecting the mentioned column-span. Summarizing, the maximal column-rank of $\mathcal{A}_\theta$ modulo the vectors whose (x,y)-th entries are independent of y given x is a lower bound for the dimension of $\mathrm{RBM}^k_{n,m}$.

Note that $\mathcal{A}_\theta$ depends on θ in a discrete way: the parameter space $\mathbb{R}^N$ is partitioned in finitely many regions where $\mathcal{A}_\theta$ is constant. The piece-wise linear map thus emerging, with linear pieces represented by the $\mathcal{A}_\theta$, is the tropical CRBM morphism, and its image is the tropical CRBM model.

Each linear region of the tropical morphism corresponds to a function $h_\theta \colon \{0,1\}^{k+n} \to \{0,1\}^m$ taking visible state vectors to the most likely hidden state vectors. Geometrically, such an inference function corresponds to m slicings of the $(k+n)$-dimensional unit hypercube. Namely, every hidden unit divides the visible space $\{0,1\}^{k+n} \subset \mathbb{R}^{k+n}$ in two halfspaces, according to its preferred state.

Each of these m slicings defines a column block of the matrix $\mathcal{A}_\theta$. More precisely,

$$\mathcal{A}_\theta = (A, A_{C_1}, \cdots, A_{C_m}),$$

where A is the matrix with rows $(1, v_1, \ldots, v_{k+n})$ for all $v \in \{0,1\}^{k+n}$, and A_C is the same matrix, with rows multiplied by the indicator function of the set C of points v classified as positive by a linear classifier (slicing).

If we consider only linear classifiers that select rows of A corresponding to disjoint Hamming balls of radius one (that is, such that the C_i are disjoint radius-one Hamming balls), then the rank of $\mathcal{A}_\theta$ is equal to the number of such classifiers times $(n+k+1)$ (which is the rank of each block A_{C_i}), plus the rank of $A_{\{0,1\}^{k+n}\setminus\cup_{i\in[m]}C_i}$ (which is the remainder rank of the first block A). The column-rank modulo functions of x is equal to the rank minus $k+1$ (which is the dimension of the functions of x spanned by columns of A), minus at most the number of cylinder sets $[x] = \{(x,y)\colon y \in \{0,1\}^n\}$ for some $x \in \{0,1\}^k$ that are contained in $\cup_{i\in[m]}C_i$. This completes the proof of the claim.

The bound given in the first item is a consequence of the following observations. Each cylinder set $[x]$ contains 2^n points. If a given cylinder set $[x]$ intersects a radius-1 Hamming ball B but is not contained in it, then it also intersects the radius-2 Hamming sphere around B. Choosing the radius-1 Hamming ball slicings $C_1, \dots, C_m$ to have centers at least Hamming distance 4 apart, we can ensure that their union does not contain any cylinder set $[x]$.

The second item is by the second item of Proposition 3; when the probability model $\mathrm{RBM}_{n+k,m}$ is full dimensional, then $\mathrm{RBM}^k_{n,m}$ is full dimensional. ■

Proof of Corollary 5 For the maximal cardinality of distance-4 binary codes of length l it is known that $A(l,4) \geq 2^r$, where r is the largest integer with $2^r < \frac{2^l}{1+(l-1)+(l-1)(l-2)/2}$ (Gilbert, 1952; Varshamov, 1957), and so $A_2(l,4) \geq 2^{l-\lfloor \log_2(l^2-l+2)\rfloor}$. Furthermore, for the minimal size of radius-one covering codes of length l it is known that $K(l,1) \leq 2^{l-\lfloor \log_2(l+1)\rfloor}$ (Cueto et al., 2010). ■

Appendix B. Details on Universal Approximation

In the following two subsections we address the minimal sufficient and the necessary number of hidden units for universal approximation.

B.1 Sufficient Number of Hidden Units

This subsection contains the proof of Theorem 7 about the minimal size of CRBM universal approximators. The proof is constructive: given any target conditional distribution, it proceeds by adjusting the weights of the hidden units successively until obtaining the desired approximation. The idea of the proof is that each hidden unit can be used to model the probability of an output vector, for several different input vectors. The probability of a given output vector can be adjusted at will by a single hidden unit, jointly for several input vectors, when these input vectors are in general position. This comes at the cost of generating dependent output probabilities for all other inputs in the same affine space. The main difficulty of the proof lies in the construction of sequences of successively conflict-free groups of affinely independent inputs, and in estimating the shortest possible length of such sequences exhausting all possible inputs. The proof is composed of several lemmas and propositions. We start with a few definitions:

Definition 24 Given two probability distributions p and q on a finite set $\mathcal{X}$, the *Hadamard product* or renormalized entry-wise product $p*q$ is the probability distribution on $\mathcal{X}$ defined by $(p*q)(x) = p(x)q(x)/\sum_{x'} p(x')q(x')$ for all $x \in \mathcal{X}$. When building this product, we assume that the supports of p and q are not disjoint, such that the normalization term does not vanish.

The probability distributions that can be represented by RBMs can be described in terms of Hadamard products. Namely, for every probability distribution p that can be represented by $\mathrm{RBM}_{n,m}$, the model $\mathrm{RBM}_{n,m+1}$ with one additional hidden unit can represent precisely the probability distribution of the form $p' = p*q$, where $q = \lambda' r + (1-\lambda')s$ is a mixture, with $\lambda' \in [0,1]$, of two strictly positive product distributions $r(x) = \prod_{i\in[n]} r_i(x_i)$ and $s(x) = \prod_{i\in[n]} s_i(x_i)$. For clarity, the notations are $x = (x_1, \ldots, x_n) \in \{0,1\}^n$, $r, s \in \Delta_n$, and $r_i, s_i \in \Delta_1$ for all $i \in [n] = \{1, \ldots, n\}$. In other words, each additional hidden unit amounts to Hadamard-multiplying the distributions representable by an RBM with the distributions representable as mixtures of product distributions. The same result is obtained by considering only the Hadamard products with mixtures where r is equal to the uniform distribution. In this case, the distributions $p' = p*q$ are of the form $p' = \lambda p + (1-\lambda)p*s$, where s is any strictly positive product distribution and $\lambda = \frac{\lambda'}{\lambda' + 2^n(1-\lambda')\sum_x p(x)s(x)}$ is any weight in $[0,1]$.

Definition 25 A *probability sharing step* is a transformation taking a probability distribution p to $p' = \lambda p + (1-\lambda)p*s$, for some strictly positive product distribution s and some $\lambda \in [0,1]$.

In order to prove Theorem 7, for each $k \in \mathbb{N}$ and $n \in \mathbb{N}$ we want to find an $m_{k,n} \in \mathbb{N}$ such that: for any given strictly positive conditional distribution $q(\cdot|\cdot)$, there exists $p \in \mathrm{RBM}_{n+k,0}$ and $m_{k,n}$ probability sharing steps taking p to a strictly positive joint distribution p' with $p'(\cdot|\cdot) = q(\cdot|\cdot)$. The idea is that the starting distribution is represented by an RBM with no hidden units, and each sharing step is realized by adding a hidden unit to the RBM. In order to obtain these sequences of sharing steps, we will use the following technical lemma:

Lemma 26 *Let B be a radius-1 Hamming ball in $\{0,1\}^k$ and let C be a cylinder subset of $\{0,1\}^k$ containing the center of B. Let $\lambda^x \in (0,1)$ for all $x \in B \cap C$, let $\tilde{y} \in \{0,1\}^n$ and let $\delta_{\tilde{y}}$ denote the Dirac delta on $\{0,1\}^n$ assigning probability one to $\tilde{y}$. Let $p \in \Delta_{k+n}$ be a strictly positive probability distribution with conditionals $p(\cdot|x)$ and let*

$$p'(\cdot|x) := \begin{cases} \lambda^x p(\cdot|x) + (1-\lambda^x)\delta_{\tilde{y}}, & \text{for all } x \in B \cap C \\ p(\cdot|x), & \text{for all } x \in \{0,1\}^k \setminus C \end{cases}.$$

Then, for any $\epsilon > 0$, there is a probability sharing step taking p to a joint distribution p'' with conditionals satisfying $\sum_y |p''(y|x) - p'(y|x)| \leq \epsilon$ for all $x \in (B \cap C) \cup (\{0,1\}^k \setminus C)$.

Proof We define the sharing step $p' = \lambda p + (1-\lambda)p*s$ with a product distribution s supported on $C \times \{\tilde{y}\} \subseteq \{0,1\}^{k+n}$. Note that given any distribution q on C and a radius-1 Hamming ball B whose center is contained in C, there is a product distribution s on C such that $s|_{C\cap B} \propto q|_{C\cap B}$. In other words, the restriction of a product distribution s to a radius-1

Hamming ball B can be made proportional to any non-negative vector of length $|B|$. To see this, recall that a product distribution is a vector with entries $s(x) = \prod_{i\in[k]} s_i(x_i)$, $x = (x_1, \ldots, x_k) \in \{0,1\}^k$. Without loss of generality let B be centered at $(0, \ldots, 0)$; that is, $B = \{x \in \{0,1\}^k \colon \sum_{i\in[k]} x_i \leq 1\}$. The restriction of s to B is given by

$$\begin{aligned} s|_B &= \Big(\prod_{i\in[k]} s_i(0),\ s_1(1) \prod_{i\in[k]\setminus\{1\}} s_i(0),\ s_2(1) \prod_{i\in[k]\setminus\{2\}} s_i(0),\ \ldots,\ s_k(1) \prod_{i\in[k]\setminus\{k\}} s_i(0) \Big) \\ &= \Big(\prod_{i\in[k]} s_i(0),\ \frac{s_1(1)}{s_1(0)} \prod_{i\in[k]} s_i(0),\ \frac{s_2(1)}{s_2(0)} \prod_{i\in[k]} s_i(0),\ \ldots,\ \frac{s_k(1)}{s_k(0)} \prod_{i\in[k]} s_i(0) \Big) \\ &\propto \Big(1,\ \frac{s_1(1)}{s_1(0)},\ \frac{s_2(1)}{s_2(0)},\ \ldots,\ \frac{s_k(1)}{s_k(0)}\Big). \end{aligned}$$

Now, by choosing the factor distributions $s_i = (s_i(0), s_i(1)) \in \Delta_1$ appropriately, the vector $\big(\frac{s_1(1)}{s_1(0)}, \ldots, \frac{s_k(1)}{s_k(0)}\big)$ can be made arbitrary in $\mathbb{R}^k_+$. ■

We have the following two implications of Lemma 26:

Corollary 27 *For any $\epsilon > 0$ and $q(\cdot|x) \in \Delta_n$ for all $x \in B \cap C$, there is an $\epsilon' > 0$ such that, for any strictly positive joint distribution $p \in \Delta_{k+n}$ with conditionals satisfying $\sum_y |p(y|x) - \delta_0(y)| \leq \epsilon'$ for all $x \in B \cap C$, there are $2^n - 1$ sharing steps taking p to a joint distribution p'' with conditionals satisfying $\sum_y |p''(y|x) - p'(y|x)| \leq \epsilon$ for all $x \in (B \cap C) \cup (\{0,1\}^k \setminus C)$, where δ_0 is the Dirac delta on $\{0,1\}^n$ assigning probability one to the vector of zeros and*

$$p'(\cdot|x) := \begin{cases} q(\cdot|x), & \text{for all } x \in B \cap C \\ p(\cdot|x), & \text{for all } x \in \{0,1\}^k \setminus C \end{cases}.$$

Proof Consider any $x \in B \cap C$. We will show that the probability distribution $q(\cdot|x) \in \Delta_n$ can be written as the transformation of a Dirac delta by $2^n - 1$ sharing steps. Then the claim follows from Lemma 26. Let $\sigma \colon \{0,1\}^n \to \{0, \ldots, 2^n - 1\}$ be an enumeration of $\{0,1\}^n$. Let $p^{(0)}(y|x) = \delta_{\sigma^{-1}(0)}(y)$ be the starting distribution (the Dirac delta concentrated at the state $\tilde{y} \in \{0,1\}^n$ with $\sigma(\tilde{y}) = 0$) and let the t-th sharing step be defined by $p^{(t)}(y) = \lambda^x_{\sigma^{-1}(t)} p^{(t-1)}(y|x) + (1 - \lambda^x_{\sigma^{-1}(t)}) \delta_{\sigma^{-1}(t)}(y)$, for some weight $\lambda^x_{\sigma^{-1}(t)} \in [0,1]$. After $2^n - 1$ sharing steps, we obtain the distribution

$$p^{(2^n-1)}(y|x) = \sum_{\tilde{y}} \Big(\prod_{\tilde{y}' \colon \sigma(\tilde{y}') > \sigma(\tilde{y})} \lambda^x_{\tilde{y}'} \Big) (1 - \lambda^x_{\tilde{y}}) \delta_{\tilde{y}}(y), \quad \text{for all } y \in \{0,1\}^n,$$

whereby $\lambda^x_{\tilde{y}} := 0$ for $\sigma(\tilde{y}) = 0$. This distribution is equal to $q(\cdot|x)$ for the following choice of weights:

$$\lambda^x_{\tilde{y}} := 1 - \frac{q(\tilde{y}|x)}{1 - \sum_{\tilde{y}' \colon \sigma(\tilde{y}') > \sigma(\tilde{y})} q(\tilde{y}'|x)}, \quad \text{for all } \tilde{y} \in \{0,1\}^n.$$

It is easy to verify that these weights satisfy the condition $\lambda^x_{\tilde{y}} \in [0,1]$ for all $\tilde{y} \in \{0,1\}^n$, and $\lambda^x_{\tilde{y}} = 0$ for that $\tilde{y}$ with $\sigma(\tilde{y}) = 0$, independently of the specific choice of σ. ■

Note that this corollary does not make any statement about the rows $p''(\cdot|x)$ with $x \in C \setminus B$. When transforming the $(B \cap C)$-rows of p according to Lemma 26, the $(C \setminus B)$-rows get transformed as well, in a non-trivial dependent way. Fortunately, there is a sharing step that allows us to "reset" exactly certain rows to a desired point measure, without introducing new non-trivial dependencies:

Corollary 28 *For any $\epsilon > 0$, any cylinder set $C \subseteq \{0,1\}^k$, and any $\tilde{y} \in \{0,1\}^n$, any strictly positive joint distribution p can be transformed by a probability sharing step to a joint distribution p'' with conditionals satisfying $\sum_y |p''(y|x) - p'(y|x)| \leq \epsilon$ for all $x \in \{0,1\}^k$, where*

$$p'(\cdot|x) := \begin{cases} \delta_{\tilde{y}}, & \text{for all } x \in C \\ p(\cdot|x), & \text{for all } x \in \{0,1\}^k \setminus C \end{cases}.$$

Proof The sharing step can be defined as $p'' = \lambda p + (1-\lambda) p * s$ with s close to the uniform distribution on $C \times \{\tilde{y}\}$ and λ close to 0 (close enough depending on ϵ). ■

We will refer to a sharing step as described in Corollary 28 as a *reset* of the C-rows of p. Furthermore, we will denote by *star* the intersection of a radius-1 Hamming ball and a cylinder set containing the center of the ball. See Figure 5A.

With all the observations made above, we can construct an algorithm that generates an arbitrarily accurate approximation of any given conditional distribution by applying a sequence of sharing steps to any given strictly positive joint distribution. The details are given in Algorithm 1. The algorithm performs sequential sharing steps on a strictly positive joint distribution $p \in \Delta_{k+n}$ until the resulting distribution p' has a conditional distribution $p'(\cdot|\cdot)$ satisfying $\sum_y |p'(y|x) - q(y|x)| \leq \epsilon$ for all x.

In order to obtain a bound on the number m of hidden units for which $\mathrm{RBM}^k_{n,m}$ can approximate a given target conditional distribution arbitrarily well, we just need to evaluate the number of sharing steps run by Algorithm 1. For this purpose, we investigate the combinatorics of sharing step sequences and evaluate their worst case lengths. We can choose as starting distribution some $p \in \mathrm{RBM}_{n+k,0}$ with conditionals satisfying $\sum_y |p(y|x) - \delta_0(y)| \leq \epsilon'$ for all $x \in \{0,1\}^k$, for some $\epsilon' > 0$ small enough depending on the target conditional $q(\cdot|\cdot)$ and the targeted approximation accuracy ϵ.

Definition 29 A sequence of stars $B^1, \ldots, B^l$ packing $\{0,1\}^k$ with the property that the smallest cylinder set containing any of the stars in the sequence does not intersect any previous star in the sequence is called a *star packing sequence* for $\{0,1\}^k$.

The number of sharing steps run by Algorithm 1 is bounded from above by $(2^n - 1)$ times the length of a star packing sequence for the set of inputs $\{0,1\}^k$. Note that the choices of stars and the lengths of the possible star packing sequences are not unique. Figure 5B gives an example showing that starting a sequence with large stars is not necessarily the best strategy to produce a short sequence. The next lemma states that there is a class of star packing sequences of a certain length, depending on the size of the input space. Thereby, this lemma upper-bounds the worst case complexity of Algorithm 1.

Algorithm 1 Algorithmic illustration of the proof of Theorem 7.

Input: Strictly positive joint distribution p, target conditional distribution $q(\cdot|\cdot)$, and $\epsilon > 0$
Output: Transformation p' of the input p with $\sum_y |p'(y|x) - q(y|x)| \leq \epsilon$ for all x

Initialize $\mathcal{B} \leftarrow \emptyset$ {Here $\mathcal{B} \subseteq \{0,1\}^k$ denotes the set of inputs x that have been readily processed in the current iteration}
while $\mathcal{B} \not\supseteq \{0,1\}^k$ **do**
  Choose (disjoint) cylinder sets $C^1, \ldots, C^K$ packing $\{0,1\}^k \setminus \mathcal{B}$
  If needed, perform at most K sharing steps resetting the C^i rows of p for all $i \in [K]$, taking $p(\cdot|x)$ close to δ_0 for all $x \in C^i$ for all $i \in [K]$ and leaving all other rows close to their current values, according to Corollary 28
  for each $i \in [K]$ **do**
    Perform at most $2^n - 1$ sharing steps taking $p(\cdot|x)$ close to $q(\cdot|x)$ for all $x \in B^i$, where B^i is some star contained in C^i, and leaving the $(\{0,1\}^k \setminus C^i)$-rows close to their current values, according to Corollary 27
  end for
  $\mathcal{B} \leftarrow \mathcal{B} \cup (\cup_{i\in[K]} B^i)$
end while

Lemma 30 *Let $r \in \mathbb{N}$, $S(r) := 1 + 2 + \cdots + r$, $k \geq S(r)$, $f_i(z) := 2^{S(i-1)} + (2^i - (i+1))z$, and $F(r) := f_r(f_{r-1}(\cdots f_2(f_1)))$. There is a star packing sequence for $\{0,1\}^k$ of length $2^{k-S(r)}F(r)$. Furthermore, for this sequence, Algorithm 1 requires at most $R(r) := \prod_{i=2}^r (2^i - (i+1))$ resets.*

Proof The star packing sequence is constructed by the following procedure. In each step, we define a set of cylinder sets packing all sites of $\{0,1\}^k$ that have not been covered by stars so far, and include a sub-star of each of these cylinder sets in the sequence.

- As an initialization step, we split $\{0,1\}^k$ into $2^{k-S(r)}$ $S(r)$-dimensional cylinder sets, denoted $D^{(j_1)}$, $j_1 \in \{1, \ldots, 2^{k-S(r)}\}$.

- In the first step, for each j_1, the $S(r)$-dimensional cylinder set $D^{(j_1)}$ is packed by $2^{S(r-1)}$ r-dimensional cylinder sets $C^{(j_1),i}$, $i \in \{1, \ldots, 2^{S(r-1)}\}$. For each i, we define the star $B^{(j_1),i}$ as the radius-1 Hamming ball within $C^{(j_1),i}$ centered at the smallest element of $C^{(j_1),i}$ (with respect to the lexicographic order of $\{0,1\}^k$), and include it in the sequence.

- At this point, the sites in $D^{(j_1)}$ that have not yet been covered by stars is $D^{(j_1)} \setminus (\cup_i B^{(j_1),i})$. This set is split into $2^r - (r+1)$ $S(r-1)$-dimensional cylinder sets, which we denote by $D^{(j_1,j_2)}$, $j_2 \in \{1, \ldots, 2^r - (r+1)\}$.

- Note that $\cup_{j_1} D^{(j_1,j_2)}$ is a cylinder set, and hence, for each j_2, the $(\cup_{j_1} D^{(j_1,j_2)})$-rows of a conditional distribution being processed by Algorithm 1 can be jointly reset by one single sharing step to achieve $p'(\cdot|x) \approx \delta_0$ for all $x \in \cup_{j_1} D^{(j_1,j_2)}$.

- In the second step, for each j_2, the cylinder set $D^{(j_1,j_2)}$ is packed by $2^{S(r-2)}$ $(r-1)$-dimensional cylinder sets $C^{(j_1,j_2),i}$, $i \in \{1, \ldots, 2^{S(r-2)}\}$, and the corresponding stars are included in the sequence.

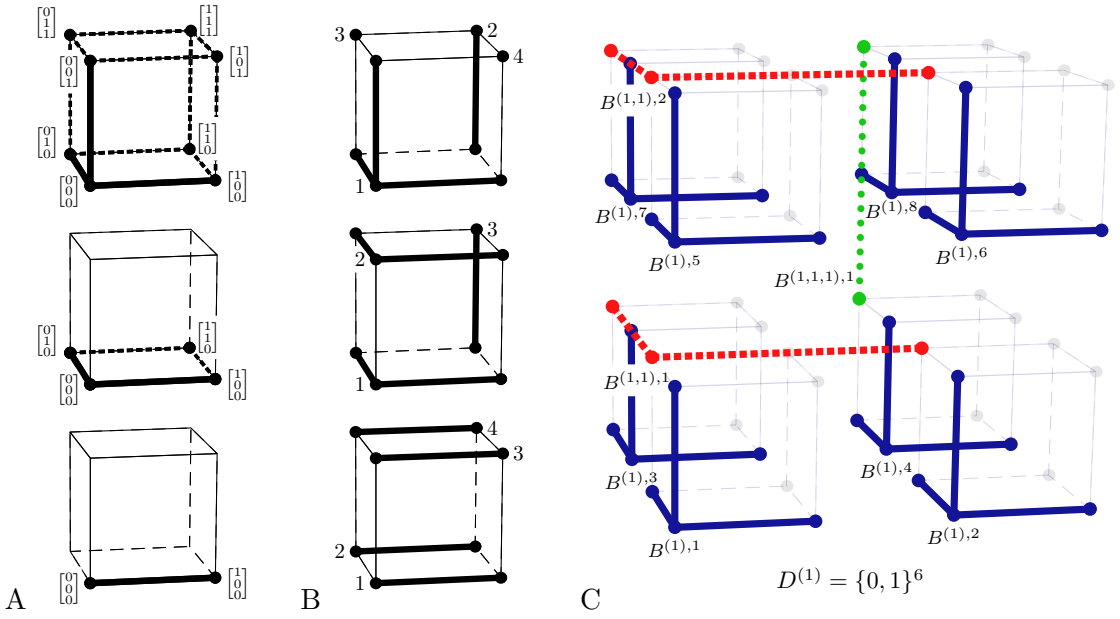


Figure 5: A) Examples of radius-1 Hamming balls in cylinder sets of dimension 3, 2, and 1. The cylinder sets are shown as bold vertices connected by dashed edges, and the nested Hamming balls (stars) as bold vertices connected by solid edges. B) Three examples of star packing sequences for $\{0,1\}^3$. C) Illustration of the star packing sequence constructed in Lemma 30 for $\{0,1\}^6$.

- The procedure is iterated until the r-th step. In this step, each $D^{(j_1,\dots,j_r)}$ is a 1-dimensional cylinder set and is packed by a single 1-dimensional cylinder set $C^{(j_1,\dots,j_r),1} = B^{(j_1,\dots,j_r),1}$. Hence, at this point, all of $\{0,1\}^k$ has been exhausted and the procedure terminates.

Summarizing, the procedure is initialized by creating the branches $D^{(j_1)}$, $j_1 \in [2^{k-S(r)}]$. In the first step, each branch $D^{(j_1)}$ produces $2^{S(r-1)}$ stars and splits into the branches $D^{(j_1,j_2)}$, $j_2 \in [2^r - (r+1)]$. More generally, in the i-th step, each branch $D^{(j_1,\dots,j_i)}$ produces $2^{S(r-i)}$ stars, and splits into the branches $D^{(j_1,\dots,j_i,j_{i+1})}$, $j_{i+1} \in [2^{r-(i-1)} - (r+1-(i-1))]$.

The total number of stars $D^{(j_1,\dots,j_r)}$ is given precisely by $2^{k-S(r)}$ times the value of the iterative function $F(r) = f_r(f_{r-1}(\cdots f_2(f_1)))$, whereby $f_1 = 1$. The total number of resets is given by the number of branches created from the first step on, which is precisely $R(r) = \prod_{i\in[r]}(2^i - (i+1))$.

Figure 5C offers an illustration of these star packing sequences. It shows the case $k = S(3) = 6$. In this case there is only one initial branch $D^{(1)} = \{0,1\}^6$. The stars $B^{(1),i}$, $i \in [2^{S(2)}] = [8]$ are shown in solid blue, $B^{(1,1),i}$, $i \in [2^{S(1)}] = [2]$ in dashed red, and $B^{(1,1,1),1}$ in dotted green. For clarity, only these stars are highlighted. The stars $B^{(1,j_2),i}$ and $B^{(1,j_2,1),1}$ resulting from split branches are similar to those highlighted. ■

With this, we obtain the general bound of the theorem:

	$m_{n,k}^{(r)} =$					
r	2^k	$2^{-S(r)}$	$F(r)$	(2^n-1)	$+$	$R(r)$
1	2^k	2^{-1}	1	(2^n-1)	$+$	0
2	2^k	2^{-3}	3	(2^n-1)	$+$	1
3	2^k	2^{-6}	20	(2^n-1)	$+$	4
4	2^k	2^{-10}	284	(2^n-1)	$+$	44
5	2^k	2^{-15}	8408	(2^n-1)	$+$	1144
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$
> 17	2^k	0.2263		(2^n-1)	$+$	$2^{S(r)}0.0269$

Table 1: Numerical evaluation of the bounds from Proposition 31. Each row evaluates the universal approximation bound $m_{n,k}^{(r)}$ for a value of r.

Proposition 31 (Theorem 7, general bound) *Let $k \geq S(r)$. The model $\mathrm{RBM}_{n,m}^k$ can approximate every conditional distribution from $\Delta_{k,n}$ arbitrarily well whenever $m \geq m_{k,n}^{(r)}$, where $m_{k,n}^{(r)} := 2^{k-S(r)}F(r)(2^n-1) + R(r)$.*

Proof This is in view of the complexity of Algorithm 1 for the sequence described in Lemma 30. ■

In order to make the universal approximation bound more comprehensible, in Table 1 we evaluated the sequence $m_{n,k}^{(r)}$ for $r = 1, 2, 3 \ldots$ and $k \geq S(r)$. Furthermore, the next proposition gives an explicit expression for the coefficients $2^{-S(r)}F(r)$ and $R(r)$ appearing in the bound. This yields the second part of Theorem 7. In general, the bound $m_{n,k}^{(r)}$ decreases with increasing r, except possibly for a few values of k when n is small. For a pair (k, n), any $m_{n,k}^{(r)}$ with $k \geq S(r)$ is a sufficient number of hidden units for obtaining a universal approximator.

Proposition 32 (Theorem 7, explicit bounds) *The function $K(r) := 2^{-S(r)}F(r)$ is bounded from below and above as $K(6)\prod_{i=7}^{r}\left(1-\frac{i-3}{2^i}\right) \leq K(r) \leq K(6)\prod_{i=7}^{r}\left(1-\frac{i-4}{2^i}\right)$ for all $r \geq 6$. Furthermore, $K(6) \approx 0.2442$ and $K(\infty) \approx 0.2263$. Moreover, $R(r) := \prod_{i=2}^{r}(2^i-(i+1)) = 2^{S(r)}P(r)$, where $P(r) := \frac{1}{2}\prod_{i=2}^{r}(1-\frac{(i+1)}{2^i})$, and $P(\infty) \approx 0.0269$.*

Proof From the definition of $S(r)$ and $F(r)$, we obtain that

$$K(r) = 2^{-r} + K(r-1)(1-2^{-r}(r+1)). \tag{1}$$

Note that $K(1) = \frac{1}{2}$, and that $K(r)$ decreases monotonically.

Now, note that if $K(r-1) \leq \frac{1}{c}$, then the left hand side of Equation (1) is bounded from below as $K(r) \geq K(r-1)(1-2^{-r}(r+1-c))$. For a given c, let r^c be the first r for which $K(r-1) \leq \frac{1}{c}$, assuming that such an r exists. Then

$$K(r) \geq K(r^c-1)\prod_{i=r^c}^{r}\left(1-\frac{i+1-c}{2^i}\right), \quad \text{for all } r \geq r^c. \tag{2}$$

Similarly, if $K(r) > \frac{1}{d}$ for all $r \geq r^b$, then

$$K(r) \leq K(r^b - 1) \prod_{i=r^b}^{r} \left(1 - \frac{i+1-b}{2^i}\right), \quad \text{for any } r \geq r^b.$$

Direct computations show that $K(6) \approx 0.2445 \leq \frac{1}{4}$. On the other hand, using the computational engine Wolfram—Alpha(access June 01, 2014) we obtain $\prod_{i=0}^{\infty}\left(1 - \frac{i-3}{2^i}\right) \approx 7.7413$. Plugging both terms into Equation (2) yields that $K(r)$ is always bounded from below by 0.2259.

Since $K(r)$ is never smaller than or equal to $\frac{1}{5}$, we obtain $K(r) \leq K(r'-1)\prod_{i=r'}^{r}\left(1 - \frac{i-4}{2^i}\right)$, for any r' and $r \geq r'$. Using $r' = 7$, the right hand side evaluates in the limit of large r to approximately 0.2293.

Numerical evaluation of $K(r)$ from Equation (1) for r up to one million (using Matlab R2013b) indicates that, indeed, $K(r)$ tends to approximately 0.2263 for large r. ■

We close this subsection with the remark that the proof strategy can be used not only to study universal approximation, but also approximability of selected classes of conditional distributions:

Remark 33 If we only want to model a restricted class of conditional distributions, then adapting Algorithm 1 to these restrictions may yield tighter bounds for the number of hidden units that suffices to represent these restricted conditionals. For example:

If we only want to model the target conditionals $q(\cdot|x)$ for the inputs x from a subset $\mathcal{S} \subseteq \{0,1\}^k$ and do not care about $q(\cdot|x)$ for $x \notin \mathcal{S}$, then in the algorithm we just need to replace $\{0,1\}^k$ by $\mathcal{S}$. In this case, a cylinder set packing of $\mathcal{S}\setminus\mathcal{B}$ is understood as a collection of disjoint cylinder sets $C^1, \ldots, C^K \subseteq \{0,1\}^k$ with $\cup_{i\in[K]} C^i \supseteq \mathcal{S}\setminus\mathcal{B}$ and $(\cup_{i\in[K]} C^i) \cap \mathcal{B} = \emptyset$.

Furthermore, if for some cylinder set C^i and a corresponding star $B^i \subseteq C^i$ the conditionals $q(\cdot|x)$ with $x \in B^i$ have a common support set $T \subseteq \{0,1\}^n$, then the C^i-rows of p can be reset to a distribution δ_y with $y \in T$, and only $|T| - 1$ sharing steps are needed to transform p to a distribution whose conditionals approximate $q(\cdot|x)$ for all $x \in B^i$ to any desired accuracy. In particular, for the class of target conditional distributions with $\operatorname{supp} q(\cdot|x) = T$ for all x, the term $2^n - 1$ in the complexity bound of Algorithm 1 is replaced by $|T| - 1$.

B.2 Necessary Number of Hidden Units

Proposition 9 follows from simple parameter counting arguments. In order to make this rigorous, first we make the observation that universal approximation of (conditional) probability distributions by Boltzmann machines or any other models based on exponential families, with or without hidden variables, requires the number of model parameters to be as large as the dimension of the set being approximated. We denote by $\Delta_{\mathcal{X},\mathcal{Y}}$ the set of conditionals with inputs form a finite set $\mathcal{X}$ and outputs from a finite set $\mathcal{Y}$. Accordingly, we denote by $\Delta_{\mathcal{Y}}$ the set of probability distributions on $\mathcal{Y}$.

Lemma 34 *Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ be some finite sets. Let $\mathcal{M} \subseteq \Delta_{\mathcal{X},\mathcal{Y}}$ be defined as the set of conditionals of the marginal $\mathcal{M}' \subseteq \Delta_{\mathcal{X}\times\mathcal{Y}}$ of an exponential family $\mathcal{E} \subseteq \Delta_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}$. If*

$\mathcal{M}$ is a universal approximator of conditionals from $\Delta_{\mathcal{X},\mathcal{Y}}$, then $\dim(\mathcal{E}) \geq \dim(\Delta_{\mathcal{X},\mathcal{Y}}) = |\mathcal{X}|(|\mathcal{Y}|-1)$.

The intuition of this lemma is that, for models defined by marginals of exponential families, the set of conditionals that can be approximated arbitrarily well is essentially equal to the set of conditionals that can be represented exactly, implying that there are no low-dimensional universal approximators of this type.

Proof of Lemma 34 We consider first the case of probability distributions; that is, the case with $|\mathcal{X}| = 1$ and $\mathcal{X} \times \mathcal{Y} \cong \mathcal{Y}$. Let $\mathcal{M}$ be the image of the exponential family $\mathcal{E}$ by a differentiable map f (for example, the marginal map). The closure $\overline{\mathcal{E}}$, which consists of all distributions that can be approximated arbitrarily well by $\mathcal{E}$, is a compact set. Since f is continuous, the image of $\overline{\mathcal{E}}$ is also compact, and $\overline{\mathcal{M}} = \overline{f(\mathcal{E})} = f(\overline{\mathcal{E}})$. The model $\mathcal{M}$ is a universal approximator if and only if $\overline{\mathcal{M}} = \Delta_{\mathcal{Y}}$. The set $\overline{\mathcal{E}}$ is a finite union of exponential families; one exponential family $\mathcal{E}_F$ for each possible support set F of distributions from $\overline{\mathcal{E}}$. When $\dim(\mathcal{E}) < \dim(\Delta_{\mathcal{Y}})$, each point of each $\mathcal{E}_F$ is a critical point of f (the Jacobian is not surjective at that point). By Sard's theorem, each $\mathcal{E}_F$ is mapped by f to a set of measure zero in $\Delta_{\mathcal{Y}}$. Hence the finite union $\cup_F f(\mathcal{E}_F) = f(\cup_F \mathcal{E}_F) = f(\overline{\mathcal{E}}) = \overline{\mathcal{M}}$ has measure zero in $\Delta_{\mathcal{Y}}$.

For the general case, with $|\mathcal{X}| \geq 1$, note that $\mathcal{M} \subseteq \Delta_{\mathcal{X},\mathcal{Y}}$ is a universal approximator if and only if the joint model $\Delta_{\mathcal{X}}\mathcal{M} = \{p(x)q(y|x)\colon p \in \Delta_{\mathcal{X}}, q \in \mathcal{M}\} \subseteq \Delta_{\mathcal{X}\times\mathcal{Y}}$ is a universal approximator. The latter is the marginal of the exponential family $\Delta_{\mathcal{X}} * \mathcal{E} = \{p * q\colon p \in \Delta_{\mathcal{X}}, q \in \mathcal{E}\} \subseteq \Delta_{\mathcal{X}\times\mathcal{Y}\times\mathcal{Z}}$. Hence the claim follows from the first part. ■

Proof of Proposition 9 If $\mathrm{RBM}^k_{n,m}$ is a universal approximator of conditionals from $\Delta_{k,n}$, then the model consisting of all probability distributions of the form $p(x,y) = \frac{1}{Z}\sum_z \exp(z^\top W y + z^\top V x + b^\top y + c^\top z + f(x))$ is a universal approximator of probability distributions from Δ_{k+n}. The latter is the marginal of an exponential family of dimension $mn + mk + n + m + 2^k - 1$. Thus, by Lemma 34, $m \geq \frac{2^{k+n}-2^k-n}{(n+k+1)}$. ■

Appendix C. Details on the Maximal Approximation Errors

Proof of Proposition 10 We have that $D_{\mathrm{RBM}^k_{n,m}} \leq \max_{p\in\Delta_{k+n}\colon p_X = u_X} D(p\|\,\mathrm{RBM}_{n+k,m})$. The right hand side is bounded by n, since the RBM model contains the uniform distribution. It is also bounded by the maximal divergence $D_{\mathrm{RBM}_{n+k,m}} \leq (n+k) - \lfloor \log_2(m+1)\rfloor - \frac{m+1}{2^{\lfloor \log_2(m+1)\rfloor}}$ (Montúfar et al., 2013). ■

In order to prove Theorem 11, we will upper bound the approximation errors of CRBMs by the approximation errors of submodels of CRBMs. First, we note the following:

Lemma 35 *The maximal divergence of a conditional model that is a Cartesian product of a probability model is bounded from above by the maximal divergence of that probability model: if $\mathcal{M} = \times_{x\in\{0,1\}^k}\mathcal{N} \subseteq \Delta_{k,n}$ for some $\mathcal{N} \subseteq \Delta_n$, then $D_{\mathcal{M}} \leq D_{\mathcal{N}}$.*

Proof For any $p \in \Delta_{k,n}$, we have

$$\begin{aligned} D(p\|\mathcal{M}) &= \inf_{q\in\mathcal{M}} \frac{1}{2^k}\sum_x D(p(\cdot|x)\|q(\cdot|x)) \\ &= \frac{1}{2^k}\sum_x \inf_{q(\cdot|x)\in\mathcal{N}} D(p(\cdot|x)\|q(\cdot|x)) \\ &\leq \frac{1}{2^k}\sum_x D_{\mathcal{N}} = D_{\mathcal{N}}. \end{aligned}$$

This proves the claim. ■

Definition 36 *Given a partition $\mathcal{Z} = \{\mathcal{Y}_1, \ldots, \mathcal{Y}_L\}$ of $\{0,1\}^n$, the* partition model $\mathcal{P}_{\mathcal{Z}} \subseteq \Delta_n$ *is the set of all probability distributions on $\{0,1\}^n$ with constant value on each partition block.*

The set $\{0,1\}^l$, $l \leq n$ naturally defines a partition of $\{0,1\}^n$ into cylinder sets $\{y \in \{0,1\}^n \colon y_{[l]} = z\}$ for all $z \in \{0,1\}^l$. The divergence from $\mathcal{P}_{\mathcal{Z}}$ is bounded from above by $D_{\mathcal{P}_{\mathcal{Z}}} \leq l - n$. Now, the model $\mathrm{RBM}^k_{n,m}$ can approximate certain products of partition models arbitrarily well:

Proposition 37 *Let $\mathcal{Z} = \{0,1\}^l$ with $l \leq n$. Let r be any integer with $k \geq S(r)$. The model $\mathrm{RBM}^k_{n,m}$ can approximate any conditional distribution from the product of partition models $\mathcal{P}^k_{\mathcal{Z}} := \mathcal{P}_{\mathcal{Z}} \times \cdots \times \mathcal{P}_{\mathcal{Z}}$ arbitrarily well whenever $m \geq 2^{k-S(r)}F(r)(|\mathcal{Z}| - 1) + R(r)$.*

Proof This is analogous to the proof of Proposition 19, with a few differences. Each element z of $\mathcal{Z}$ corresponds to a cylinder set $\{y \in \{0,1\}^n \colon y_{[l]} = z\}$ and the collection of cylinder sets for all $z \in \mathcal{Z}$ is a partition of $\{0,1\}^n$. Now we can run Algorithm 1 in a slightly different way, with sharing steps defined by $p' = \lambda p + (1-\lambda)u_z$, where u_z is the uniform distribution on the cylinder set corresponding to z. ■

Proof of Theorem 11 This follows directly from Lemma 35 and Proposition 37. ■

Appendix D. Details on the Representation of Conditional Distributions from Markov Random Fields

The proof of Theorem 14 is based on ideas from Younes (1996), who discussed the universal approximation property of Boltzmann machines. We will use the following:

Lemma 38 (Younes 1996, Lemma 1) *Let ϱ be a real number. Consider a fixed integer N and binary variables $x_1, \ldots, x_N$. There are real numbers w and b such that:*

- *If $\varrho \geq 0$,* $\log\left(1 + \exp(w(x_1 + \cdots + x_N) + b)\right) = \varrho \prod_i x_i + Q(x_1, \ldots, x_N)$.
- *If $\varrho \leq 0$,* $\log\left(1 + \exp(w(x_1 + \cdots + x_{N-1} - x_N) + b)\right) = \varrho \prod_i x_i + Q(x_1, \ldots, x_N)$.

Where Q is in each case a polynomial of degree less than $N-1$ in $x_1, \dots, x_N$.

The following is a generalization of another result from the same work:

Lemma 39 *Let I and J be two simplicial complexes on $[n]$ with $J \subseteq I$. If p is any distribution from $\mathcal{E}_I$ and $m \geq |\{A \in I \setminus J \colon |A| > 1\}|$, then there is a distribution $p' \in \mathcal{E}_J$, such that $p * p'$ is contained in* $\mathrm{RBM}_{n,m}$.

Proof The proof follows closely the arguments presented by Younes (1996, Lemma 2). Let $K = \{A \in I \setminus J \colon |A| > 1\}$. Consider an RBM with n visible units and $m = |K|$ hidden units. Consider a joint distribution $q(x,u) = \frac{1}{Z}\exp(H(x,u))$ of the fully observable RBM, defined as follows. We label the hidden units by subsets $A \in K$. For each $A \in K$, let $s(A)$ denote the largest element of A, and let

$$H(x,u) = \sum_{A \in K} u_A \left(w_A S_A^{\epsilon_A}(x_A) + b_A \right) + \sum_{s \in [n]} b_s x_s,$$

where

$$S_A^{\epsilon_A}(x_A) = \Big(\sum_{s \in A, s < s(A)} x_s \Big) + \epsilon_A x_{s(A)},$$

for some $\epsilon_A \in \{-1, +1\}$, $w_A, b_A, b_s \in \mathbb{R}$ that we will specify further below.

Denote the log probabilities of $p(x)$ and $p'(x)$ by

$$E(x) = \sum_{A \in I} \theta_A \prod_{i \in A} x_i \quad \text{and} \quad E'(x) = \sum_{A \in J} \vartheta_A \prod_{i \in A} x_i.$$

We obtain the desired equality $(p * p')(x) = \sum_u q(x,u)$ when

$$E(x) = \log \left(\sum_u \exp(H(x,u)) \right) - \sum_{A \in J} \vartheta_A \prod_{i \in A} x_i, \tag{3}$$

for some choice of ϑ_A, for $A \in J$, some choice of ϵ_A, w_A, b_A, for $A \in K$, and some choice of b_s, for $s \in [n]$. We have

$$\begin{aligned}
\log \left(\sum_u \exp(H(x,u)) \right) &= \log \left(\sum_u \exp \Big(\sum_A u_A (w_A S_A^{\epsilon_A}(x_A) + b_A) + \sum_{s \in [n]} b_s x_s \Big) \right) \\
&= \log \left(\Big(\sum_u \prod_A \exp(u_A (w_A S_A^{\epsilon_A}(x_A) + b_A)) \Big) \exp \Big(\sum_{s \in [n]} b_s x_s \Big) \right) \\
&= \log \left(\Big(\prod_A \sum_{u_A} \exp(u_A (w_A S_A^{\epsilon_A}(x_A) + b_A)) \Big) \exp \Big(\sum_{s \in [n]} b_s x_s \Big) \right) \\
&= \sum_A \log(1 + \exp(w_A S_A^{\epsilon_A}(x_A) + b_A)) + \sum_{s \in [n]} b_s x_s.
\end{aligned}$$

The terms $\phi_A^{\epsilon_A}(x_A) := \log\big(1+\exp(w_A S_A^{\epsilon_A}(x_A)+b_A)\big)$ are of the same form as the functions from Lemma 38. To solve Equation (3), we first apply Lemma 38 on $\phi_A^{\epsilon_A}$ to cancel the terms $\theta_A \prod_{i\in A} x_i$ of $E(x)$ for which A is a maximal element of $I\setminus J$ of cardinality more than one. This involves choosing appropriate $\epsilon_A \in \{-1,+1\}$, w_A and b_A, for the corresponding A. The remaining polynomial consists of terms with strictly smaller monomials. We apply lemma 38 repeatedly on this polynomial, until only monomials with $A \in J$ or $|A|=1$ remain. These terms are canceled with $\vartheta_A \prod_{i\in A} x_i$, $A\in J$, or with $b_s x_s$, $s\in[n]$. ■

Proof of Theorem 14 By Lemma 39, there is a $p' \in \mathcal{E}_J$, $J = 2^{[k]}$, such that $p * p'$ is in $\mathrm{RBM}_{k+n,m}$. Now, the conditionals distribution $(p*p')(y|x)$ of the last n units, given the first k units, are independent of p', since this is independent of y. ■

Proof of Corollary 15 The statement follows from Theorem 14, considering the simplicial complex $I = 2^{[k]} \times J$ and a joint probability distribution $p \in \mathcal{E}_I \subseteq \Delta_{k+n}$ with the desired conditionals $p(\cdot|x) = p^x$. ■

Appendix E. Details on the Approximation of Conditional Distributions with Restricted Supports

Proof of Proposition 18 This follows from the fact that $\mathrm{RBM}_{n+k,m}$ can approximate any probability distribution with support of cardinality $m+1$ arbitrarily well (Montúfar and Ay, 2011). ■

Proof of Proposition 19 This is analogous to the proof of Proposition 31. The complexity of Algorithm 1 as evaluated there does not depend on the specific structure of the support sets, but only on their cardinality, as long as they are the same for all x. ■

The following lemma states that a CRBM can compute all deterministic conditionals that can be computed by a feedforward linear threshold network with the same number of hidden units. Recall that the Heaviside step function, here denoted hs, maps a real number a to 0 if $a<0$, to $1/2$ if $a=0$, and to 1 if $a>0$. A linear threshold function with N input bits and M output bits is just a function of the form $\{0,1\}^N \to \{0,1\}^M$; $y \mapsto \mathrm{hs}(Wy+b)$ with a generic choice of $W\in\mathbb{R}^{M\times N}$ and $b\in\mathbb{R}^M$.

Lemma 40 *Consider a function $f\colon \{0,1\}^k \to \{0,1\}^n$. The model $\mathrm{RBM}_{n,m}^k$ can approximate the deterministic policy $p(y|x)=\delta_{f(x)}(y)$ arbitrarily well, whenever this can be represented by a feedforward linear threshold network with m hidden units; that is, when*

$$f(x) = \mathrm{hs}(W^\top(\mathrm{hs}(Vx+c))+b), \quad \textit{for all } x\in\{0,1\}^k,$$

for some generic choice of W,V,b,c.

Proof Consider the conditional distribution $p(\cdot|x)$. This is the visible marginal of $p(y,z|x) = \frac{1}{Z}\exp((Vx+c)^\top z + b^\top y + z^\top W y)$. Consider weights α and β, with α large enough, such that

$\operatorname{argmax}_z(\alpha Vx+\alpha c)^\top z = \operatorname{argmax}_z(\alpha Vx+\alpha c)^\top z + (\beta W^\top z+\beta b)^\top y$ for all $y \in \{0,1\}^n$. Note that for generic choices of V and c, the set $\operatorname{argmax}_z(\alpha V+\alpha c)^\top z$ consists of a single point $z^* = \operatorname{hs}(Vx+c)$. We have $\operatorname{argmax}_{(y,z)}(\alpha Vx+\alpha c)^\top z+(\beta W^\top z+\beta b)^\top y = (z^*, \operatorname{argmax}_y(\beta W^\top z^* + \beta b)^\top y)$. Here, again, for generic choices of V and b, the set $\operatorname{argmax}_y(\beta W^\top z^* + \beta b)^\top y$ consists of a single point $y^* = \operatorname{hs}(W^\top z^* + b)$. The joint distribution $p(y,z|x)$ with parameters $t\beta W, t\alpha V, t\beta b, t\alpha c$ tends to the point measure $\delta_{(y^*,z^*)}(y,z)$ as $t \to \infty$. In this case $p(y|x)$ tends to $\delta_{y^*}(y)$ as $t \to \infty$, where $y^* = \operatorname{hs}(W^\top z^* + b) = \operatorname{hs}(W^\top \operatorname{hs}(Vx + c) + b)$, for all $x \in \{0,1\}^k$. ■

Proof of Theorem 20 The second statement is precisely Lemma 40. For the more general statement the arguments are as follows. Note that the conditional distribution $p(y|z)$ of the output units, given the hidden units, is the same for a CRBM and for its feedforward network version. Furthermore, for each input x, the CRBM output distribution is $p(y|x) = \sum_z (q(z|x) * p(z))p(y|z)$, where

$$q(z|x) = \frac{\exp(z^\top Vx + c^\top z)}{\sum_{z'} \exp(z'^\top Vx + c^\top z')}$$

is the conditional distribution represented by the first layer,

$$p(y,z) = \frac{\exp(z^\top Wy + b^\top y)}{\sum_{y',z'} \exp(z'^\top Wy' + b^\top y')}$$

is the distribution represented by the RBM with parameters $W, b, 0$, and

$$q(z|x) * p(z) = \frac{q(z|x)p(z)}{\sum_{z'} q(z'|x)p(z')}, \quad \text{for all } z,$$

is the renormalized entry-wise product of the conditioned distribution $q(\cdot|x)$ and the RBM hidden marginal distribution

$$p(z) = \sum_y p(y,z).$$

Now, if q is deterministic, then $q(z|x) * p(z)$ is the same as $q(z|x)$, regardless of $p(z)$ (strictly positive). ■

The proof of Theorem 21 builds on the following lemma, which describes a combinatorial property of the deterministic policies that can be approximated arbitrarily well by CRBMs.

Lemma 41 *Consider a function $f\colon \{0,1\}^k \to \{0,1\}^n$. The model $\mathrm{RBM}^k_{n,m}$ can approximate the deterministic policy $p(y|x) = \delta_{f(x)}(y)$ arbitrarily well only if there is a choice of the model parameters W, V, b, c for which*

$$f(x) = \operatorname{hs}(W^\top \operatorname{hs}([W,V]\left[\begin{smallmatrix} f(x) \\ x \end{smallmatrix}\right] + c) + b), \quad \textit{for all } x \in \{0,1\}^k,$$

where the Heaviside function hs *is applied entry-wise to its argument.*

Proof Consider a choice of W, V, b, c. For each input state x, the conditional represented by $\mathrm{RBM}^k_{n,m}$ is equal to the mixture distribution $p(y|x) = \sum_z p(z|x)p(y|x,z)$, with mixture components $p(y|x,z) = p(y|z) \propto \exp((z^\top W + b^\top)y)$ and mixture weights $p(z|x) \propto \sum_{y'} \exp((z^\top W + b^\top)y' + z^\top(Vx+c))$ for all $z \in \{0,1\}^m$. The support of a mixture distribution is equal to the union of the supports of the mixture components with non-zero mixture weights. In the present case, if $\sum_y |p(y|x) - \delta_{f(x)}(y)| \leq \alpha$, then $\sum_y |p(y|x,z) - \delta_{f(x)}(y)| \leq \alpha/\epsilon$ for all z with $p(z|x) > \epsilon$, for any $\epsilon > 0$. Choosing α small enough, α/ϵ can be made arbitrarily small for any fixed $\epsilon > 0$. In this case, for every z with $p(z|x) > \epsilon$, necessarily

$$(z^\top W + b^\top)f(x) \gg (z^\top W + b^\top)y, \quad \text{for all } y \neq f(x), \tag{4}$$

and hence

$$\operatorname{sgn}(z^\top W + b^\top) = \operatorname{sgn}(f(x) - \tfrac{1}{2}).$$

Furthermore, the probability assigned by $p(z|x)$ to all z that do not satisfy Equation (4) has to be very close to zero (upper bounded by a function that decreases with α). The probability of z given x is given by

$$p(z|x) = \frac{1}{Z_{z|x}} \exp(z^\top(Vx+c)) \sum_{y'} \exp((z^\top W + b^\top)y').$$

In view of Equation (4), for all z with $p(z|x) > \epsilon$, if α is small enough, $p(z|x)$ is arbitrarily close to

$$\frac{1}{Z_{z|x}} \exp(z^\top(Vx+c)) \exp((z^\top W + b^\top)f(x)).$$

This holds, in particular, for every z that maximizes $p(z|x)$. Therefore,

$$\operatorname{argmax}_z p(z|x) = \operatorname{argmax}_z z^\top(Wf(x) + Vx + c).$$

Each of these z must satisfy Equation (4). This completes the proof. ■

Proof of Theorem 21 We start with the sufficient condition. The bound $2^k - 1$ follows directly from Proposition 18. For the second bound, note that any function $f\colon \{0,1\}^k \to \{0,1\}^n;\ x \mapsto y$ can be computed by a parallel composition of the functions $f_i\colon x \mapsto y_i$, for all $i \in [n]$. Hence the bound follows from Lemma 40 and the fact that a feedforward linear threshold network with $\frac{3}{k+2}2^k$ hidden units can compute any Boolean function.

We proceed with the necessary condition. Lemma 41 shows that each deterministic policy that can be approximated by $\mathrm{RBM}^k_{n,m}$ arbitrarily well corresponds to the y-coordinate fixed points of a map defined as the composition of two linear threshold functions $\{0,1\}^{k+n} \to \{0,1\}^m;\ (x,y) \mapsto \mathrm{hs}([W,V]\left[\begin{smallmatrix} y \\ x \end{smallmatrix}\right] + c)$ and $\{0,1\}^m \to \{0,1\}^n;\ z \mapsto \mathrm{hs}(W^\top z + b)$. In particular, we can upper bound the number of deterministic policies that can be approximated arbitrarily well by $\mathrm{RBM}^k_{n,m}$, by the total number of compositions of two linear threshold functions; one with $n+k$ inputs and m outputs and the other with m inputs and n outputs.

Let $\mathrm{LTF}(N,M)$ be the number of linear threshold functions with N inputs and M outputs. It is known that (Ojha, 2000; Wenzel et al., 2000)

$$\mathrm{LTF}(N,M) \leq 2^{N^2 M}.$$

The number of deterministic policies that can be approximated arbitrarily well by $\mathrm{RBM}^k_{n,m}$ is thus bounded above by $\mathrm{LTF}(n+k,m)\cdot\mathrm{LTF}(m,n)\leq 2^{m(n+k)^2+nm^2}$. The actual number may be smaller, in view of the fixed-point and shared parameter constraints. On the other hand, the number of deterministic policies in $\Delta_{k,n}$ is as large as $(2^n)^{2^k}=2^{n2^k}$. The claim follows from comparing these two numbers. ■

References

Nihat Ay, Guido Montúfar, and Johannes Rauh. Selection criteria for neuromanifolds of stochastic dynamics. In Y. Yamaguchi, editor, *Advances in Cognitive Neurodynamics (III)*, pages 147–154. Springer, 2013.

Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NY, 1957.

Yoshua Bengio. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, January 2009.

Maria A. Cueto, Jason Morton, and Bernd Sturmfels. Geometry of the restricted Boltzmann machine. In M. Viana and H. Wynn, editors, *Algebraic Methods in Statistics and Probability II, AMS Special Session*, volume 2. AMS, 2010.

Asja Fischer and Christian Igel. An introduction to restricted Boltzmann machines. In L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, editors, *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, volume 7441 of *Lecture Notes in Computer Science*, pages 14–36. Springer, 2012.

Yoav Freund and David Haussler. *Unsupervised Learning of Distributions of Binary Vectors Using Two Layer Networks*. Technical report. Computer Research Laboratory, University of California, Santa Cruz, 1994.

Edgar N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31:504–522, 1952.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Geoffrey E. Hinton. A practical guide to training restricted Boltzmann machines. In G. Montavon, G. Orr, and K.-R. Müller, editors, *Neural Networks: Tricks of the Trade*, volume 7700 of *Lecture Notes in Computer Science*, pages 599–619. Springer, 2012.

Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted Boltzmann machines. In W. Cohen, A. McCallum, and S. Roweis, editors, *Proceedings of the 25th International Conference on Machine Learning*, pages 536–543. ACM, 2008.

Nicolas Le Roux and Yoshua Bengio. Representational power of restricted Boltzmann machines and deep belief networks. *Neural Computation*, 20(6):1631–1649, 2008.

James Martens, Arkadev Chattopadhya, Toni Pitassi, and Richard Zemel. On the expressive power of restricted Boltzmann machines. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2877–2885. Curran Associates, Inc., 2013.

Volodymyr Mnih, Hugo Larochelle, and Geoffrey E. Hinton. Conditional restricted Boltzmann machines for structured output prediction. In F. Cozman and A. Pfeffer, editors, *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, pages 514–522. AUAI Press, 2011.

Guido Montúfar and Nihat Ay. Refinements of universal approximation results for deep belief networks and restricted Boltzmann machines. *Neural Computation*, 23(5):1306–1319, 2011.

Guido Montúfar and Jason Morton. Discrete restricted Boltzmann machines. *Journal of Machine Learning Research*, 16:653–672, 2015.

Guido Montúfar and Jason Morton. When does a mixture of products contain a product of mixtures? *SIAM Journal on Discrete Mathematics*, 29:321–347, 2015.

Guido Montúfar and Johannes Rauh. Scaling of model approximation errors and expected entropy distances. *Kybernetika*, 50(2):234–245, 2014.

Guido Montúfar, Johannes Rauh, and Nihat Ay. Expressive power and approximation errors of restricted Boltzmann machines. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 415–423. Curran Associates, Inc., 2011.

Guido Montúfar, Johannes Rauh, and Nihat Ay. Maximal information divergence from statistical models defined by neural networks. In F. Nielsen and F. Barbaresco, editors, *Geometric Science of Information*, volume 8085 of *Lecture Notes in Computer Science*, pages 759–766. Springer, 2013.

Guido Montúfar, Keyan Ghazi-Zahedi, and Nihat Ay. A theory of cheap control in embodied systems. *PLoS Comput Biol*, 11(9):e1004427, 09 2015.

Piyush C. Ojha. Enumeration of linear threshold functions from the lattice of hyperplane intersections. *IEEE Transactions on Neural Networks*, 11(4):839–850, Jul 2000.

Sheldon M. Ross. *Introduction to Stochastic Dynamic Programming*. Academic Press, Inc., Orlando, FL, USA, 1983.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey E. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, pages 791–798. ACM, 2007.

Brian Sallans and Geoffrey E. Hinton. Reinforcement learning with factored states and actions. *Journal of Machine Learning Research*, 5:1063–1088, 2004.

Paul Smolensky. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

Ilya Sutskever and Geoffrey E. Hinton. Learning multilevel distributed representations for high-dimensional sequences. In M. Meila and X. Shen, editors, *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pages 548–555. Journal of Machine Learning Research, 2007.

Graham W. Taylor, Geoffrey E. Hinton, and Sam T. Roweis. Modeling human motion using binary latent variables. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 1345–1352. MIT Press, 2007.

Laurens van der Maaten. Discriminative restricted Boltzmann machines are universal approximators for discrete data. Technical Report EWI-PRB TR 2011001, Delft University of Technology, 2011.

Rom R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akad. Nauk SSSR*, 117:739–741, 1957.

Walter Wenzel, Nihat Ay, and Frank Pasemann. Hyperplane arrangements separating arbitrary vertex classes in n-cubes. *Advances in Applied Mathematics*, 25(3):284–306, 2000.

Laurent Younes. Synchronous Boltzmann machines can be universal approximators. *Applied Mathematics Letters*, 9(3):109 – 113, 1996.

Matthew Zeiler, Graham Taylor, Niko Troje, and Geoffrey E. Hinton. Modeling pigeon behaviour using a conditional restricted Boltzmann machine. In *17th European Symposium on Artificial Neural Networks*, 2009.

Journal of Machine Learning Research 16 (2015) 2437-2457 Submitted 3/14; Revised 4/15; Published 12/15

From Dependency to Causality: A Machine Learning Approach

Gianluca Bontempi GBONTE@ULB.AC.BE
Maxime Flauder MAX.FLAUDER@GMAIL.COM
Machine Learning Group, Computer Science Department,
Interuniversity Institute of Bioinformatics in Brussels (IB)^2,
ULB, Université Libre de Bruxelles,
Brussels, Belgium

Editor: Isabelle Guyon and Alexander Statnikov

Abstract

The relationship between statistical dependency and causality lies at the heart of all statistical approaches to causal inference. Recent results in the ChaLearn cause-effect pair challenge have shown that causal directionality can be inferred with good accuracy also in Markov indistinguishable configurations thanks to data driven approaches. This paper proposes a supervised machine learning approach to infer the existence of a directed causal link between two variables in multivariate settings with $n > 2$ variables. The approach relies on the asymmetry of some conditional (in)dependence relations between the members of the Markov blankets of two variables causally connected. Our results show that supervised learning methods may be successfully used to extract causal information on the basis of asymmetric statistical descriptors also for $n > 2$ variate distributions.

Keywords: causal inference, information theory, machine learning

1. Introduction

The relationship between statistical dependency and causality lies at the heart of all statistical approaches to causal inference and can be summarized by two famous statements: *correlation (or more generally statistical association) does not imply causation* and *causation induces a statistical dependency between causes and effects (or more generally descendants)* (Reichenbach, 1956). In other terms it is well known that statistical dependency is a necessary yet not sufficient condition for causality. The unidirectional link between these two notions has been used by many formal approaches to causality to justify the adoption of statistical methods for detecting or inferring causal links from observational data. The most influential one is the Causal Bayesian Network approach, detailed in (Koller and Friedman, 2009) which relies on notions of independence and conditional independence to detect causal patterns in the data. Well known examples of related inference algorithms are the constraint-based methods like the PC algorithms (Spirtes et al., 2000) and IC (Pearl, 2000). These approaches are founded on probability theory and have been shown to be accurate in reconstructing causal patterns in many applications (Pourret et al., 2008), notably in bioinformatics (Friedman et al., 2000). At the same time they restrict the set of configurations which causal inference is applicable to. Such boundary is essentially determined by

the notion of *distinguishability* which defines the set of Markov equivalent configurations on the basis of conditional independence tests. Typical examples of indistinguishability are the two-variable setting and the completely connected triplet configuration (Guyon et al., 2007) where it is impossible to distinguish between cause and effects by means of conditional or unconditional independence tests.

If on one hand the notion of indistinguishability is probabilistically sound, on the other hand it should not prevent us from addressing interesting yet indistinguishable causal patterns. In fact, indistinguishability results rely on two main aspects: i) they refer only to specific features of dependency (notably conditional or unconditional independence) and ii) they state the conditions (e.g. faithfulness) under which it is possible to distinguish (or not) *with certainty* between configurations. Accordingly, indistinguishability results do not prevent the existence of statistical algorithms able to *reduce the uncertainty about the causal pattern* even in indistinguishable configurations. This has been made evident by the appearance in recent years of a series of approaches which tackle the cause-effect pair inference, like ANM (Additive Noise Model) (Hoyer et al., 2009), IGCI (Information Geometry Causality Inference) (Daniusis et al., 2010; Janzing et al., 2012), LiNGAM (Linear Non Gaussian Acyclic Model) (Shimizu et al., 2006) and the algorithms described in (Mooij et al., 2010) and (Statnikov et al., 2012)[1]. What is common to these approaches is that they use alternative statistical features of the data to detect causal patterns and reduce the uncertainty about their directionality. A further important step in this direction has been represented by the recent organization of the ChaLearn cause-effect pair challenge (Guyon, 2014). The good (and significantly better than random) accuracy obtained on the basis of observations of pairs of causally related (or unrelated) variables supports the idea that alternative strategies can be designed to infer with success (or at least significantly better than random) indistinguishable configurations.

It is worthy to remark that the best ranked approaches[2] in the ChaLearn competition share a common aspect: they infer from statistical features of the bivariate distribution the probability of the existence and then of the directionality of the causal link between two variables. The success of these approaches shows that the problem of causal inference can be successfully addressed as a supervised machine learning approach where the inputs are features describing the probabilistic dependency and the output is a class denoting the existence (or not) of a directed causal link. Once sufficient training data are made available, conventional feature selection algorithms (Guyon and Elisseeff, 2003) and classifiers can be used to return a prediction better than random.

The effectiveness of machine learning strategies in the case of pairs of variables encourages the extension of the strategy to configurations with a larger number of variables. In this paper we propose an original approach to learn from multivariate observations the probability that a variable is a direct cause of another. This task is undeniably more difficult because

- the number of parameters needed to describe a multivariate distribution increases rapidly (e.g. quadratically in the Gaussian case),

1. A more extended list of recent algorithms is available in `http://www.causality.inf.ethz.ch/cause-effect.php?page=help`.
2. We took part in the ChaLearn challenge and we ranked 8th in the final leader board.

- information about the existence of a causal link between two variables is returned also by the nature of the dependencies existing between the two variables and the remaining ones.

The second consideration is evident in the case of a collider configuration $\mathbf{z}_1 \rightarrow \mathbf{z}_2 \leftarrow \mathbf{z}_3$: in this case the dependency (or independency) between $\mathbf{z}_1$ and $\mathbf{z}_3$ tells us more about the link $\mathbf{z}_1 \rightarrow \mathbf{z}_2$ than the dependency between $\mathbf{z}_1$ and $\mathbf{z}_2$. This led us to develop a machine learning strategy (described in Section 2) where descriptors of the relation existing between members of the Markov blankets of two variables are used to learn the probability (i.e. a score) that a causal link exists between two variables. The approach relies on the asymmetry of some conditional (in)dependence relations between the members of the Markov blankets of two variables causally connected. The resulting algorithm (called D2C and described in Section 3) predicts the existence of a direct causal link between two variables in a multivariate setting by (i) creating a set of of features of the relationship based on asymmetric descriptors of the multivariate dependency and (ii) using a classifier to learn a mapping between the features and the presence of a causal link.

In Section 4 we report the results of a set of experiments assessing the accuracy of the D2C algorithm. Experimental results based on synthetic and published data show that the D2C approach is competitive and often outperforms state-of-the-art methods.

2. Learning the Relation between Dependency and Causality in a Configuration with $n > 2$ Variables.

This section presents an approach to learn, from a number of observations, the relationships existing between the n variate distribution of $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$ and the existence of a directed causal link between two variables $\mathbf{z}_i$ and $\mathbf{z}_j$, $1 \leq i \neq j \leq n$, in the case of no confounding, no selection bias and no feedback configurations. Several parameters may be estimated from data in order to represent the multivariate distribution of $\mathbf{Z}$, like the correlation or the partial correlation matrix. Some problems however arise in this case like: (i) these parameters are informative in case of Gaussian distributions only, (ii) identical (or close) causal configurations could be associated to very different parametric values, thus making difficult the learning of the mapping and (iii) different causal configurations may lead to identical (or close) parametric values.

In other terms it is more relevant to describe the distribution in structural terms (e.g. with notions of conditional dependence/independence) rather than in parametric terms. Two more aspects have to be taken into consideration. First since we want to use a learning approach to identify cause-effect relationships we need some quantitative features to describe the structure of the multivariate distribution. Second, since asymmetry is a distinguishing characteristic of a causal relationship, we expect that effective features should share the same asymmetric properties.

In this paper we will use information theory to represent and quantify the notions of (conditional) dependence and independence between variables and to derive a set of asymmetric features to reconstruct causality from dependency.

2.1 Notions of Information Theory

Let us consider three continuous random variables $\mathbf{z}_1$, $\mathbf{z}_2$ and $\mathbf{z}_3$ having a joint Lebesgue density[3]. Let us start by considering the relation between $\mathbf{z}_1$ and $\mathbf{z}_2$. The mutual information (Cover and Thomas, 1990) between $\mathbf{z}_1$ and $\mathbf{z}_2$ is defined in terms of their probabilistic density functions $p(z_1)$, $p(z_2)$ and $p(z_1, z_2)$ as

$$I(\mathbf{z}_1; \mathbf{z}_2) = \int \int \log \frac{p(z_1, z_2)}{p(z_1)p(z_2)} p(z_1, z_2) dz_1 dz_2 = H(\mathbf{z}_1) - H(\mathbf{z}_1|\mathbf{z}_2) \tag{1}$$

where H is the *entropy* and the convention $0 \log \frac{0}{0} = 0$ is adopted. This quantity measures the amount of stochastic dependence between $\mathbf{z}_1$ and $\mathbf{z}_2$ (Cover and Thomas, 1990). Note that, if $\mathbf{z}_1$ and $\mathbf{z}_2$ are Gaussian distributed the following relation holds

$$I(\mathbf{z}_1; \mathbf{z}_2) = -\frac{1}{2} \log(1 - \rho^2) \tag{2}$$

where ρ is the Pearson correlation coefficient between $\mathbf{z}_1$ and $\mathbf{z}_2$.

Let us now consider a third variable $\mathbf{z}_3$. The *conditional mutual information* (Cover and Thomas, 1990) between $\mathbf{z}_1$ and $\mathbf{z}_2$ once $\mathbf{z}_3$ is given is defined by

$$I(\mathbf{z}_1; \mathbf{z}_2|\mathbf{z}_3) = \int \int \int \log \frac{p(z_1, z_2|z_3)}{p(z_1|z_3)p(z_2|z_3)} p(z_1, z_2, z_3) dz_1 dz_2 dz_3 = \\ = H(\mathbf{z}_1|\mathbf{z}_3) - H(\mathbf{z}_1|\mathbf{z}_2, \mathbf{z}_3) \tag{3}$$

The conditional mutual information is null if and only if $\mathbf{z}_1$ and $\mathbf{z}_2$ are conditionally independent given $\mathbf{z}_3$.

A structural notion which can be described in terms of conditional mutual information is the notion of Markov Blanket (MB). The Markov Blanket of variable $\mathbf{z}_i$ in an n dimensional distribution is the smallest subset of variables belonging to $\mathbf{Z} \setminus \mathbf{z}_i$ (where $\setminus$ denotes the set difference operator) which makes $\mathbf{z}_i$ conditionally independent of all the remaining ones. In information theoretic terms let us consider a set $\mathbf{Z}$ of n random variables, a variable $\mathbf{z}_i$ and a subset $\mathbf{M}_i \subset \mathbf{Z} \setminus \mathbf{z}_i$. The subset $\mathbf{M}_i$ is said to be a *Markov blanket* of $\mathbf{z}_i$ if it is the minimal subset satisfying

$$I(\mathbf{z}_i; (\mathbf{Z} \setminus (\mathbf{M}_i \cup \mathbf{z}_i))|\mathbf{M}_i) = 0$$

Effective algorithms have been proposed in literature to infer a Markov Blanket from observed data (Tsamardinos et al., 2003b). Feature selection algorithms are also useful to construct a Markov blanket of a given target variable once they rely on notions of conditional independence to select relevant variables (Meyer and Bontempi, 2014).

2.2 Causality and Asymmetric Dependency Relationships

The notion of causality is central in science and also an intuitive notion of everyday life. The remarkable property of causality which distinguishes it from dependency is asymmetry.

In probabilistic terms a variable $\mathbf{z}_i$ is dependent on a variable $\mathbf{z}_j$ if the density of $\mathbf{z}_i$, conditional on the observation $\mathbf{z}_j = z_j$, is different from the marginal one

$$p(z_i|\mathbf{z}_j = z_j) \neq p(z_i)$$

3. Boldface denotes random variables.

In information theoretic terms the two variables are dependent if $I(\mathbf{z}_i;\mathbf{z}_j) = I(\mathbf{z}_j;\mathbf{z}_i) > 0$. This implies that dependency is *symmetric*. If $\mathbf{z}_i$ is dependent on $\mathbf{z}_j$, then $\mathbf{z}_j$ is dependent on $\mathbf{z}_i$ too as shown by

$$p(z_j|\mathbf{z}_i = z_i) \neq p(z_j)$$

The formal representation of the notion of causality demands an extension of the syntax of the probability calculus as done by Pearl (1995) with the introduction of the operator $\mathtt{do}$ which allows to distinguish the observation of a value of $\mathbf{z}_j$ (denoted by $\mathbf{z}_j = z_j$) from the manipulation of the variable $\mathbf{z}_j$ (denoted by $\mathtt{do}(\mathbf{z_j} = \mathbf{z_j})$). Once this extension is accepted we say that a variable $\mathbf{z}_j$ is a cause of a variable $\mathbf{z}_i$ (e.g. "diseases cause symptoms") if the distribution of $\mathbf{z}_i$ is different from the marginal one when we set the value $\mathbf{z}_j = z_j$

$$p(z_i|\mathtt{do}(\mathbf{z}_j = z_j)) \neq p(z_i)$$

but not vice versa (e.g. "symptoms do not cause disease")

$$p(z_j|\mathtt{do}(\mathbf{z}_i = z_i)) = p(z_j)$$

The extension of the probability notation made by Pearl allows to formalize the intuition that causality is *asymmetric*. Another notation which allows to represent causal expression is provided by graphical models or more specifically by Directed Acyclic Graphs (DAG) (Koller and Friedman, 2009). In this paper we will limit to consider causal relationships modeled by DAG, which proved to be convenient tools to understand and use the notion of causality. Furthermore we will make the assumption that the set of causal relationships existing between the variables of interest can be described by a Markov and faithful DAG (Pearl, 2000). This means that the DAG is an accurate map of dependencies and independencies of the represented distribution and that using the notion of *d-separation* it is possible to read from the graph if two sets of nodes are (in)dependent conditioned on a third.

The asymmetric nature of causality suggests that if we want to infer causal links from dependency we need to find some features (or descriptors) which describe the dependency and share with causality the property of asymmetry. Let us suppose that we are interested in predicting the existence of a directed causal link $\mathbf{z}_i \rightarrow \mathbf{z}_j$ where $\mathbf{z}_i$ and $\mathbf{z}_j$ are components of an observed n-dimensional vector $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_n]$.

We define as *dependency descriptor* of the ordered pair $\langle i, j\rangle$ a function $d(i, j)$ of the distribution of $\mathbf{Z}$ which depends on i and j. Example of dependency descriptors are the correlation $\rho(i, j)$ between $\mathbf{z}_i$ and $\mathbf{z}_j$, the mutual information $I(\mathbf{z}_i;\mathbf{z}_j)$ or the partial correlation between $\mathbf{z}_i$ and $\mathbf{z}_j$ given another variable $\mathbf{z}_k, i \neq j, j \neq k, i \neq k$.

We call a dependency descriptor *symmetric* if $d(i, j) = d(j, i)$ otherwise we call it *asymmetric*. Correlation and mutual information are symmetric descriptors since

$$d(i, j) = I(\mathbf{z}_i;\mathbf{z}_j) = I(\mathbf{z}_j;\mathbf{z}_i) = d(j, i)$$

Because of the asymmetric property of causality, if we want to maximize our chances to reconstruct causality from dependency we need to identify relevant asymmetric descriptors. In order to define useful asymmetric descriptors we have recourse to the Markov Blankets of the two variables $\mathbf{z}_i$ and $\mathbf{z}_j$.

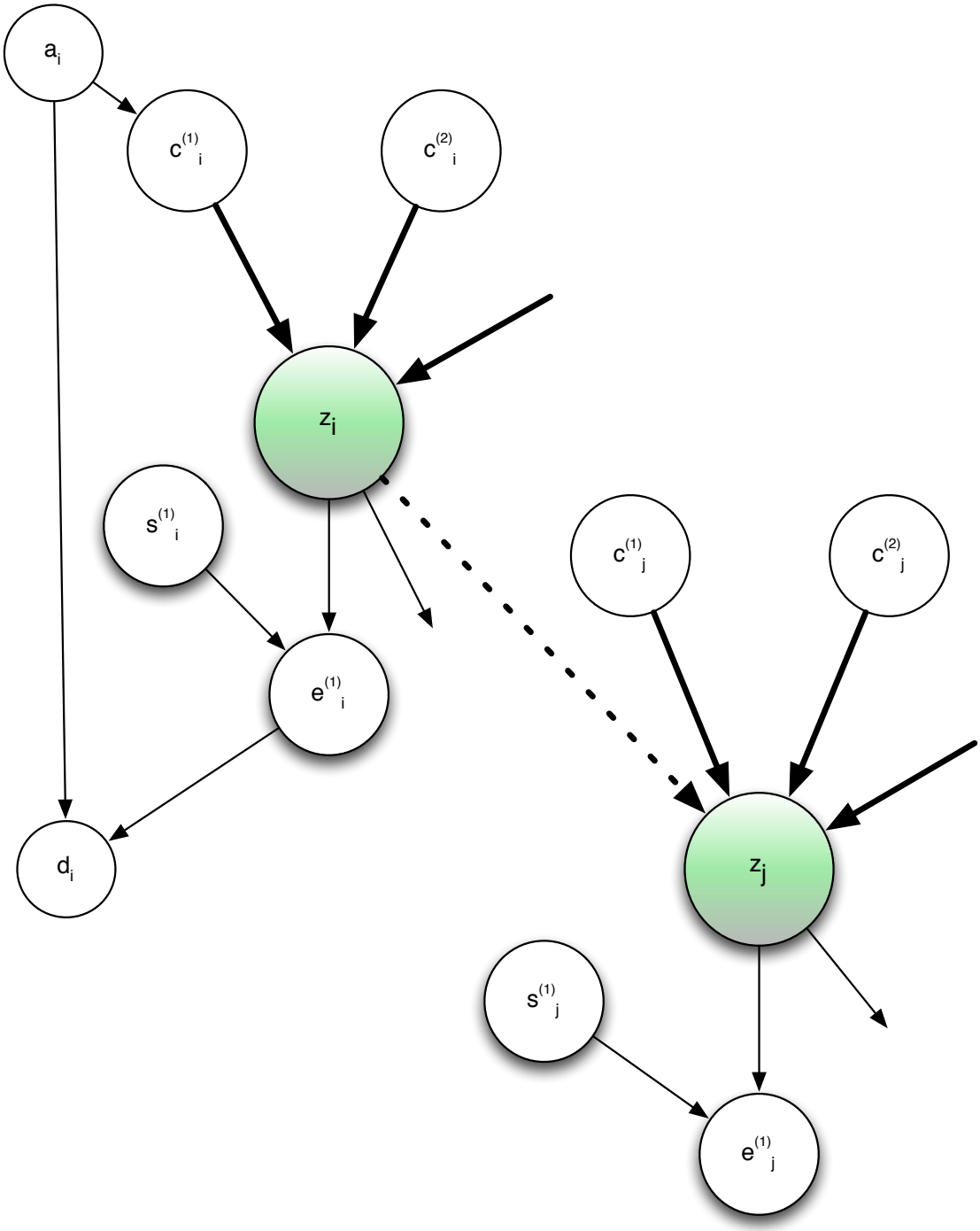


Figure 1: Two causally connected variables and their Markov Blankets.

Relation i, j	Relation j, i
$\forall k \quad \mathbf{z}_i \not\perp\!\!\!\perp \mathbf{c}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{z}_j \perp\!\!\!\perp \mathbf{c}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{e}_i^{(k)} \not\perp\!\!\!\perp \mathbf{c}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{e}_j^{(k)} \perp\!\!\!\perp \mathbf{c}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{c}_i^{(k)} \not\perp\!\!\!\perp \mathbf{c}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{c}_j^{(k)} \perp\!\!\!\perp \mathbf{c}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{z}_i \perp\!\!\!\perp \mathbf{c}_j^{(k)}$	$\forall k \quad \mathbf{z}_j \not\perp\!\!\!\perp \mathbf{c}_i^{(k)}$

Table 1: Asymmetric (un)conditional (in)dependance relationships between members of the Markov Blankets of $\mathbf{z}_i$ and $\mathbf{z}_j$ in Figure 1.

Let us consider for instance the portion of a DAG represented in Figure 1 where the variable $\mathbf{z}_i$ is a direct cause of $\mathbf{z}_j$. The figure shows also the Markov Blankets of the two variables (denoted M_i and M_j respectively) and their components, i.e. the direct causes (denoted by $\mathbf{c}$), the direct effects ($\mathbf{e}$) and the spouses ($\mathbf{s}$) (Pellet and Elisseeff, 2008).

In what follows we will make two assumptions: (i) the only path between the sets $\mathbf{z}_i \cup M_i$ and $\mathbf{z}_j \cup M_j$ is the edge $\mathbf{z}_i \to \mathbf{z}_j$ and (ii) there is no common ancestor of $\mathbf{z}_i$ ($\mathbf{z}_j$) and its spouses $\mathbf{s}_i$ ($\mathbf{s}_j$). We will discuss these assumptions at the end of the section. Given these assumptions and because of d-separation (Geiger et al., 1990), a number of asymmetric conditional (in)dependence relations holds between the members of M_i and M_j (Table 1). For instance (first line of Table 1), by conditioning on the effect $\mathbf{z}_j$ we create a dependence between $\mathbf{z}_i$ and the direct causes of $\mathbf{z}_j$ while by conditioning on the $\mathbf{z}_i$ we d-separate $\mathbf{z}_j$ and the direct causes of $\mathbf{z}_i$.

The relations in Table 1 can be used to define the following set of asymmetric descriptors,

$$d_1^{(k)}(i,j) = I(\mathbf{z}_i; \mathbf{c}_j^{(k)} | \mathbf{z}_j), \tag{4}$$
$$d_2^{(k)}(i,j) = I(\mathbf{e}_i^{(k)}; \mathbf{c}_j^{(k)} | \mathbf{z}_j), \tag{5}$$
$$d_3^{(k)}(i,j) = I(\mathbf{c}_i^{(k)}; \mathbf{c}_j^{(k)} | \mathbf{z}_j), \tag{6}$$
$$d_4^{(k)}(i,j) = I(\mathbf{z}_i; \mathbf{c}_j^{(k)}), \tag{7}$$

whose asymmetry is given by

$$d_1^{(k)}(i,j) = I(\mathbf{z}_i; \mathbf{c}_j^{(k)} | \mathbf{z}_j) > 0, \quad d_1^{(k)}(j,i) = I(\mathbf{z}_j; \mathbf{c}_i^{(k)} | \mathbf{z}_i) = 0, \tag{8}$$
$$d_2^{(k)}(i,j) = I(\mathbf{e}_i^{(k)}; \mathbf{c}_j^{(k)} | \mathbf{z}_j) > 0, \quad d_2^{(k)}(j,i) = I(\mathbf{e}_j^{(k)}; \mathbf{c}_i^{(k)} | \mathbf{z}_i) = 0, \tag{9}$$
$$d_3^{(k)}(i,j) = I(\mathbf{c}_i^{(k)}; \mathbf{c}_j^{(k)} | \mathbf{z}_j) > 0, \quad d_3^{(k)}(j,i) = I(\mathbf{c}_j^{(k)}; \mathbf{c}_i^{(k)} | \mathbf{z}_i) = 0, \tag{10}$$
$$d_4^{(k)}(i,j) = I(\mathbf{z}_i; \mathbf{c}_j^{(k)}) = 0, \quad d_4^{(k)}(j,i) = I(\mathbf{z}_j; \mathbf{c}_i^{(k)}) > 0. \tag{11}$$

Relation i, j	Relation j, i
$\forall k \quad \mathbf{z}_i \not\perp\!\!\!\perp \mathbf{e}_j^{(k)}$	$\forall k \quad \mathbf{z}_j \not\perp\!\!\!\perp \mathbf{e}_i^{(k)}$
$\forall k \quad \mathbf{z}_i \perp\!\!\!\perp \mathbf{s}_j^{(k)}$	$\forall k \quad \mathbf{z}_j \perp\!\!\!\perp \mathbf{s}_i^{(k)}$
$\forall k \quad \mathbf{z}_i \perp\!\!\!\perp \mathbf{e}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{z}_j \perp\!\!\!\perp \mathbf{e}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{z}_i \perp\!\!\!\perp \mathbf{s}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{z}_j \perp\!\!\!\perp \mathbf{s}_i^{(k)} \vert \mathbf{z}_i$
$\forall k \quad \mathbf{e}_i^{(k)} \perp\!\!\!\perp \mathbf{e}_j^{(k)} \vert \mathbf{z}_i$	$\forall k \quad \mathbf{e}_j^{(k)} \perp\!\!\!\perp \mathbf{e}_i^{(k)} \vert \mathbf{z}_j$
$\forall k \quad \mathbf{e}_i^{(k)} \perp\!\!\!\perp \mathbf{s}_j^{(k)} \vert \mathbf{z}_j$	$\forall k \quad \mathbf{e}_j^{(k)} \perp\!\!\!\perp \mathbf{s}_i^{(k)} \vert \mathbf{z}_i$

Table 2: Symmetric (un)conditional (in)dependance relationships between members of the Markov Blankets of $\mathbf{z}_i$ and $\mathbf{z}_j$ in Figure 1.

At the same time we can write a set of symmetric conditional (in)dependence relations (Table 2) and the equivalent formulations in terms of mutual information terms:

$$I(\mathbf{z}_j; \mathbf{e}_i^{(k)}) > 0, \tag{12}$$

$$I(\mathbf{z}_i; \mathbf{e}_j^{(k)}) > 0, \tag{13}$$

$$I(\mathbf{z}_j; \mathbf{s}_i^{(k)}) = I(\mathbf{z}_i; \mathbf{s}_j^{(k)}) = 0, \tag{14}$$

$$I(\mathbf{z}_i; \mathbf{e}_j^{(k)} | \mathbf{z}_j) = I(\mathbf{z}_j; \mathbf{e}_i^{(k)} | \mathbf{z}_i) = I(\mathbf{z}_i; \mathbf{s}_j^{(k)} | \mathbf{z}_j) = I(\mathbf{z}_j; \mathbf{s}_i^{(k)} | \mathbf{z}_i) = 0, \tag{15}$$

$$I(\mathbf{e}_j^{(k)}; \mathbf{e}_i^{(k)} | \mathbf{z}_i) = I(\mathbf{e}_i^{(k)}; \mathbf{e}_j^{(k)} | \mathbf{z}_j) = I(\mathbf{e}_i^{(k)}; \mathbf{s}_j^{(k)} | \mathbf{z}_j) = I(\mathbf{e}_j^{(k)}; \mathbf{s}_i^{(k)} | \mathbf{z}_i) = 0. \tag{16}$$

2.3 From Asymmetric Relationships to Distinct Distributions

The asymmetric properties of the four descriptors (4)-(7) is encouraging if we want to exploit dependency related features to infer causal properties from data. However, this optimism is undermined by the fact that all the descriptors require already the capability of distinguishing between the causes (i.e. the terms $\mathbf{c}$) and the effects (i.e. the terms $\mathbf{e}$) of the Markov Blanket of a given variable. Unfortunately this discriminating capability is what we are looking for!

In order to escape this circularity problem we consider two solutions. The first is to have recourse to a preliminary phase that prioritizes the components of the Markov Blanket and then use this result as starting point to detect asymmetries and then improve the classification of causal links. This is for instance feasible by using a filter selection algorithm, like mIMR (Bontempi and Meyer, 2010; Bontempi et al., 2011), which aims to prioritize the direct causes in the Markov Blanket by searching for pairs of variables with high relevance and low interaction.

The second solution is related to the fact that the asymmetry of the four descriptors induces a difference in the distributions of some information theoretic terms which do not require the distinction between causes and effects within the Markov Blanket. The consequence is that we can replace the descriptors (4)-(7) with other descriptors (denoted with the letter D) that can be actually estimated from data.

Let $\mathbf{m}^{(k)}$ denote a generic component of the Markov Blanket with no distinction between cause, effect or spouse. It follows that a population made of terms depending on $\mathbf{m}^{(k)}$ is a mixture of three subpopulations, the first made of causes, the second made of effects and the third of spouses, respectively. It follows that the distribution of the population is a *finite mixture* (McLaughlan, 2000) of three distributions, the first related to the causes, the second to the effects and the third to the spouses. Since the moments of the finite mixture are functions of the moments of each component, we can derive some properties of the resulting mixture from the properties of each component. For instance if we can show that all the subpopulations but one are identical (e.g. all the elements of the third subpopulation in the first mixture are larger than the elements of the analogous subpopulation in the second mixture), we can derive that the two mixture distributions are different.

Consider for instance the quantity $I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j)$ where $\mathbf{m}_j^{(k_j)}$, $k_j = 1, \ldots, K_j$ is a member of the set $M_j \setminus \mathbf{z}_i$. From (8) and (15) it follows that the mixture distribution associated to the populations $D_1(i,j) = \{I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j), k_j = 1, \ldots, K_j\}$ and $D_1(j,i) = \{I(\mathbf{z}_j; \mathbf{m}_i^{(k_i)} | \mathbf{z}_i), k_i = 1, \ldots, K_i\}$ are different since

$$\begin{cases} I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j) > I(\mathbf{z}_j; \mathbf{m}_i^{(k_i)} | \mathbf{z}_i), & \text{if } \mathbf{m}_j^{(k_j)} = \mathbf{c}_j^{(k_j)} \wedge \mathbf{m}_i^{(k_i)} = \mathbf{c}_i^{(k_i)} \\ I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j) = I(\mathbf{z}_j; \mathbf{m}_i^{(k_i)} | \mathbf{z}_i), & \text{else} \end{cases} \tag{17}$$

It follows that even if we are not able to distinguish between a cause $\mathbf{c}_j \in M_j$ and an effect $\mathbf{e}_j \in M_j$, we know that the distribution of the population $D_1(i,j)$ differs from the distribution of the population $D_1(j,i)$. We can therefore use the population $D_1(i,j)$ (or some of its moments) as descriptor of the causal dependency.

Similarly we can replace the descriptors (5), (6) with the distributions of the population $D_2(i,j) = \{I(\mathbf{m}_i^{(k_i)}; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j), k_j = 1, \ldots, K_j, k_i = 1, \ldots, K_i\}$. From (9), (10) and (16) we obtain that the distributions of the populations $D_2(i,j)$ and $D_2(j,i)$ are different.

If we make the additional assumption that $I(\mathbf{z}_j; \mathbf{e}_i^{(k)}) = I(\mathbf{z}_i; \mathbf{e}_j^{(k)}) > 0$ from (11) we obtain also that the distribution of the population $D_3(i,j) = \{I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)}), k_j = 1, \ldots, K_j\}$ is different from the one of $D_3(j,i) = \{I(\mathbf{z}_j; \mathbf{m}_i^{(k_i)}),\ k_i = 1, \ldots, K_i\}$.

The previous results are encouraging and show that though we are not able to distinguish between the different components of a Markov Blanket, we can notwithstanding compute some quantities (in this case distributions of populations) whose asymmetry is informative about the causal relationships $\mathbf{z}_i \to \mathbf{z}_j$.

As a consequence by measuring from observed data some statistics (e.g. quantiles) related to the distribution of these asymmetric descriptors, we may obtain some insight about the causal relationship between two variables. This idea is made explicit in the algorithm described in the following section.

Though these results rely on the two assumptions made before (i.e. single path and no common ancestors), two considerations are worthy to be made. First, the main goal of the approach is to shed light on the existence of dependency asymmetries also in multivariate contributions. Secondly we expect that the second layer (based on supervised learning) will eventually compensate for configurations not compliant with the assumptions and take advantage of complementarity or synergy of the descriptors in discriminating between causal configurations.

3. The D2C Algorithm

The rationale of the D2C algorithm is to predict the existence of a causal link between two variables in a multivariate setting by (i) creating a set of features of the relationship between the members of the Markov Blankets of the two variables and (ii) using a classifier (e.g. a Random Forest as in our experiments) to learn a mapping between the features and the presence of a causal link.

We use two sets of features to summarize the relation between the two Markov blankets: the first one accounts for the presence (or the position if the MB is obtained by ranking) of the terms of M_j in M_i and vice versa. For instance it is evident that if $\mathbf{z}_i$ is a cause of $\mathbf{z}_j$ we expect to find $\mathbf{z}_i$ highly ranked between the causal terms of M_j but $\mathbf{z}_j$ absent (or ranked low) among the causes of M_i. The second set of features is based on the results of the previous section and is obtained by summarizing the distributions of the asymmetric descriptors with a set of quantiles.

We propose then an algorithm (D2C) which for each pair of measured variables $\mathbf{z}_i$ and $\mathbf{z}_j$:

1. infers from data the two Markov Blankets (e.g. by using state-of-the-art approaches) M_i and M_j and the subsets $M_i \backslash \mathbf{z}_j = \{\mathbf{m}^{(k_i)}, k_i = 1, \ldots, K_i\}$ and $M_j \backslash \mathbf{z}_i = \{\mathbf{m}^{(k_j)}, k_j = 1, \ldots, K_j\}$. Most of the existing algorithms associate to the Markov Blanket a ranking such that the most strongly relevant variables are ranked before.

2. computes a set of (conditional) mutual information terms describing the dependency between $\mathbf{z}_i$ and $\mathbf{z}_j$

$$I = [I(\mathbf{z}_i; \mathbf{z}_j), I(\mathbf{z}_i; \mathbf{z}_j | \mathrm{M}_j \setminus \mathbf{z}_i), I(\mathbf{z}_i; \mathbf{z}_j | \mathrm{M}_i \setminus \mathbf{z}_j)] \tag{18}$$

3. computes the positions $P_i^{(k_i)}$ of the members $\mathbf{m}^{(k_i)}$ of $M_i \setminus \mathbf{z}_j$ in the ranking associated to $M_j \backslash \mathbf{z}_i$ and the positions $P_j^{(k_j)}$ of the terms $\mathbf{m}^{(k_j)}$ in the ranking associated to $M_i \backslash \mathbf{z}_j$. Note that in case of the absence of a term of M_i in M_j, the position is set to $K_j + 1$ (respectively $K_i + 1$).

4. computes the populations based on the asymmetric descriptors introduced in Section 2.3:

   (a) $D_1(i,j) = \{I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j), k_j = 1, \ldots, K_j\}$

   (b) $D_1(j,i) = \{I(\mathbf{z}_j; \mathbf{m}_i^{(k_i)} | \mathbf{z}_i), k_i = 1, \ldots, K_i\}$

   (c) $D_2(i,j) = \{I(\mathbf{m}_i^{(k_i)}; \mathbf{m}_j^{(k_j)} | \mathbf{z}_j), k_i = 1, \ldots, K_i, k_j = 1, \ldots, K_j\}$ and

   (d) $D_2(j,i) = \{I(\mathbf{m}_j^{(k_j)}; \mathbf{m}_i^{(k_i)} | \mathbf{z}_i), k_i = 1, \ldots, K_i, k_j = 1, \ldots, K_j\}$

   (e) $D_3(i,j) = \{I(\mathbf{z}_i; \mathbf{m}_j^{(k_j)}), k_j = 1, \ldots, K_j\}$,

   (f) $D_3(j,i) = \{I(\mathbf{z}_j, \mathbf{m}_i^{(k_i)}), k_i = 1, \ldots, K_i\}$

5. creates a vector of descriptors

$$x = [I, \mathcal{Q}(\hat{P}_i), \mathcal{Q}(\hat{P}_j), \mathcal{Q}(\hat{D}_1(i,j)), \mathcal{Q}(\hat{D}_1(j,i)), \\ \mathcal{Q}(\hat{D}_2(i,j)), \mathcal{Q}(\hat{D}_2(j,i)), \mathcal{Q}(\hat{D}_3(i,j)), \mathcal{Q}(\hat{D}_3(j,i))] \tag{19}$$

where $\hat{P}_i$ and $\hat{P}_j$ are the empirical distributions of the populations $\{P_i^{(k_i)}\}$ and $\{P_j^{(k_j)}\}$, $\hat{D}_h(i,j)$ denotes the empirical distribution of the corresponding population $D_h(i,j)$ and $\mathcal{Q}$ returns a set of sample quantiles of a distribution (in the experiments we set the quantiles to 0.1, 0.25, 0.5, 0.75, 0.9).

The vector x can be then derived from observational data and used to create a vector of descriptors to be used as inputs in a supervised learning paradigm.

The rationale of the algorithm is that the asymmetries between M_i and M_j (e.g. Table 1) induce an asymmetry on the distributions $\hat{P}$ and $\hat{D}$ and that the quantiles of those distributions provide information about the directionality of causal link ($\mathbf{z}_i \to \mathbf{z}_j$ or $\mathbf{z}_j \to \mathbf{z}_i$.) In other terms we expect that the distribution of these variables should return useful information about which is the cause and the effect. Note that these distributions would be more informative if we were able to rank the terms of the Markov Blankets by prioritizing the direct causes (i.e. the terms $\mathbf{c}_i$ and $\mathbf{c}_j$) since these terms play a major role in the asymmetries of Table 1. The D2C algorithm can then be improved by choosing an appropriate Markov Blanket selector algorithms, like the mIMR filter.

In the experiments (Section 4) we derive the information terms as difference between (conditional) entropy terms (see Equations 1 and 3) which are themselves estimated by a Lazy Learning regression algorithm (Bontempi et al., 1999) by making an assumption of Gaussian noise. Lazy Learning returns a leave-one-out estimation of conditional variance which can be easily transformed in entropy under the normal assumption (Cover and Thomas, 1990). The (conditional) mutual information terms are then obtained by using the relations (1) and (3).

3.1 Complexity Analysis

In this subsection we make a complexity analysis of the approach: first it is important to remark that since the D2C approach relies on a classifier, its learning phase can be time-consuming and dependent on the number of samples and dimension. However, this step is supposed to be performed only once and from the user perspective it is more relevant to consider the cost in the testing phase. Given two nodes for which a test of the existence of a causal link is required, three steps have to be performed:

1. computation of the Markov blankets of the two nodes. The information filters we used have a complexity $O(Cn^2)$ where C is the cost of the computation of mutual information (Meyer and Bontempi, 2014). In case of very large n this complexity may be bounded by having the filter preceded by a ranking algorithm with complexity $O(Cn)$. Such ranking may limit the number of features taken into consideration by the filters to $n' < n$ reducing then considerably the cost.

2. once a number K_i (K_j) of members of MB_i (MB_j) have been chosen, the rest of the procedure has a complexity related to the estimation of a number $O(K_iK_j)$ of descriptors. In this paper we used a local learning regression algorithm to estimate the conditional entropies terms. Given that each regression involves at most three terms, the complexity is essentially related linearly to the number N of samples

3. the last step consists in the computation of the Random Forest predictions on the test set. Since the RF has been already trained, the complexity of this step depends only on the number of trees and not on the dimensionality or number of samples.

For each test, the resulting complexity has then a cost of the order $O(Cn+Cn'^2+K_iK_jN)$. It is important to remark that an advantage of D2C is that, if we are interested in predicting the causal relation between two variables only, we are not forced to infer the entire adjacency matrix (as typically the case in constraint-based methods). This mean also that the computation of the entire matrix can be easily made parallel.

4. Experimental Validation

In this section the D2C (Section 3) algorithm is assessed in a set of synthetic experiments and published data sets.

4.1 Synthetic Data

This experimental session addresses the problem of inferring causal links from synthetic data generated for linear and non-linear DAG configurations of different sizes. All the variables are continuous, and the dependency between children and parents is modelled by the additive relationship

$$x_i = \sum_{j \in par(i)} f_{i,j}(x_j) + \epsilon_i, \qquad i = 1, \ldots, n \tag{20}$$

where the noise $\epsilon_i \sim N(0, \sigma_i)$ is Normal, $f_{i,j}(x) \in L(x)$ and three sets of continuous functions are considered:

- `linear`: $L(x) = \{f \mid f(x) = a_0 + a_1 x\}$
- `quadratic`: $L(x) = \{f \mid f(x) = a_0 + a_1 x + a_2 x^2\}$
- `sigmoid`: $L(x) = \{f \mid f(x) = \frac{1}{1+exp(a_0+a_1 x)}\}$

In order to assess the accuracy with respect to dimensionality, we considered three network sizes:

- `small`: number of nodes n is uniformly sampled in the interval $[20, 30]$,
- `medium`: number of nodes n is uniformly sampled in the interval $[100, 200]$,
- `large`: number of nodes n is uniformly sampled in the interval $[500, 1000]$,

The assessment procedure relies on the generation of a number of DAG structures[4] and the simulation, for each of them, of N (uniformly random in $[100, 500]$) node observations according to the dependency (20). In each data set we removed the observations of five percent of the variables in order to introduce unobserved variables.

4. We used the function `random_dag` from the R package gRbase (Dethlefsen and Højsgaard, 2005).

For each DAG, on the basis of its structure and the data set of observations, we collect a number of pairs $\langle x_d, y_d \rangle$, where x_d is the descriptor vector returned by (19) and y_d is the class denoting the existence (or not) of the causal link in the DAG topology.

Several sizes of training set are considered. The largest D2C training set is made of $D = 60000$ pairs $\langle x_d, y_d \rangle$ and is obtained by generating DAGs and storing for each of them the descriptors associated to at most 4 positives examples (i.e. a pair where the node z_i is a direct cause of z_j) and at most 6 negatives examples (i.e. a pair where the node z_i is not a direct cause of z_j). A Random Forest classifier is trained on the balanced data set: we use the implementation from the R package `randomForest` (Liaw and Wiener, 2002) with default setting.

The test set is obtained by considering a number of independently simulated DAGs. We consider 190 DAGs for the small and medium configurations and 90 for the large configuration. For each testing DAG we select 4 positives examples (i.e. a pair where the node z_i is a direct cause of z_j) and 6 negatives examples (i.e. a pair where the node z_i is not a direct cause of z_j). The predictive accuracy of the trained Random Forest classifier is then assessed on the test set.

The D2C approach is compared in terms of classification accuracy (Balanced Error Rate (BER)) to several state-of-the-art approaches:

- `ANM`: Additive Noise Model (Hoyer et al., 2009) using a Gaussian process with RBF kernel and the Hilbert-Schmidt Independence Criterion (pvalue=0.02)[5]
- `DAGL1`: DAG-Search score-based algorithm with potential parents selected with a L1 penalization (Schmidt et al., 2007)[6].
- `DAGSearch`: unrestricted DAG-Search score-based algorithm (multiple restart greedy hill-climbing, using edge additions, deletions, and reversals) (Schmidt et al., 2007)[6],
- `DAGSearchSparse`: DAG-Search score-based algorithm with potential parents restricted to the 10 most correlated features (Schmidt et al., 2007)[6],
- `gs`: Grow-Shrink constraint-based structure learning algorithm (Margaritis, 2003)[7],
- `hc`: hill-climbing score-based structure learning algorithm (Daly and Shen, 2007)[7],
- `iamb`: incremental association MB constraint-based structure learning algorithm (Tsamardinos et al., 2003b)[7],
- `mmhc`: max-min hill climbing hybrid structure learning algorithms (Tsamardinos et al., 2010)[7],
- `PC`: Estimate the equivalence class of a DAG using the PC algorithm[8] (this method was used only for the small size configuration (Figure 3) for computational time reasons)

5. The code is available in `https://staff.fnwi.uva.nl/j.m.mooij/code/additive-noise.tar.gz`.
6. The code is available in `http://www.cs.ubc.ca/~murphyk/Software/DAGlearn/`.
7. The code is available in the R package `bnlearn` (Scutari, 2010).
8. The code is available in the R package `pcalg` (Kalisch et al., 2012)

- `si.hiton.pc`: Semi-Interleaved HITON-PC local discovery structure learning algorithms (Tsamardinos et al., 2003a)[7],
- `tabu`: tabu search score-based structure learning algorithm[7].

The BER of six versions of the D2C method are compared to the BER of state-of-the-art methods in Figures 3 (small), Figure 4 (medium) and Figure 5 (large). The six versions of D2C are obtained by considering two types of training data (i.e. one based on linear dependency and one based on the same dependency used for testing) and three training set sizes (equal to 400, 3000 and 60000 respectively) Each subfigure corresponds to the three types of stochastic dependency (top: linear, middle: quadratic, bottom: sigmoid).

A series of considerations can be made on the basis of the experimental results:

- the n-variate approach D2C obtains competitive results with respect to several state-of-the-art techniques in the linear case,
- the improvement of D2C wrt state-of-the-art techniques (often based on linear assumptions) tends to increase when we move to more nonlinear configurations, In particular the accuracy of the D2C algorithm is able to generalize to DAG with different number of nodes and different distributions also when trained only on data observed for linear DAGs (see accuracy of D2Cx_{lin} in the second and third row of Figures 3, 4 and 5)
- the accuracy of the D2C approach improves by increasing the number of training examples,
- with a small number of examples (i.e. $N = 400$) it is already possible to learn a classifier D2C whose accuracy is competitive with state-of-the-art methods,
- the ANM approach is not able to return accurate information about causal dependency by taking into consideration only bivariate information,
- the analysis of the importance of the D2C descriptors (based on the Mean Decrease Accuracy of the Random Forest in Figure 2) shows that the most relevant variables in the vector (19) are the terms in I, D_1 and D_3.

The D2C code is available in the CRAN R package `D2C` (Bontempi et al., 2014).

4.2 Published Data

The second part of the assessment relies on the simulated and resimulated data sets proposed in Table 11 of (Aliferis et al., 2010). These 103 data sets were obtained by simulating data from known Bayesian networks and also by resimulation, where real data is used to elicit a causal network and then data is simulated from the obtained network. We split the 103 data sets in two portions: a training portion (made of 52 sets) and a second portion (made of 51 sets) for testing. This was done in order to assess the accuracy of two versions of the D2C algorithm: the first uses as training set only 40000 synthetic samples generated as in the previous section, the second includes in the training set also the 52 data sets of the training portion. The goal is to assess the generalization accuracy of the D2C algorithm with respect to DAG distributions never encountered before and not included in the training set.

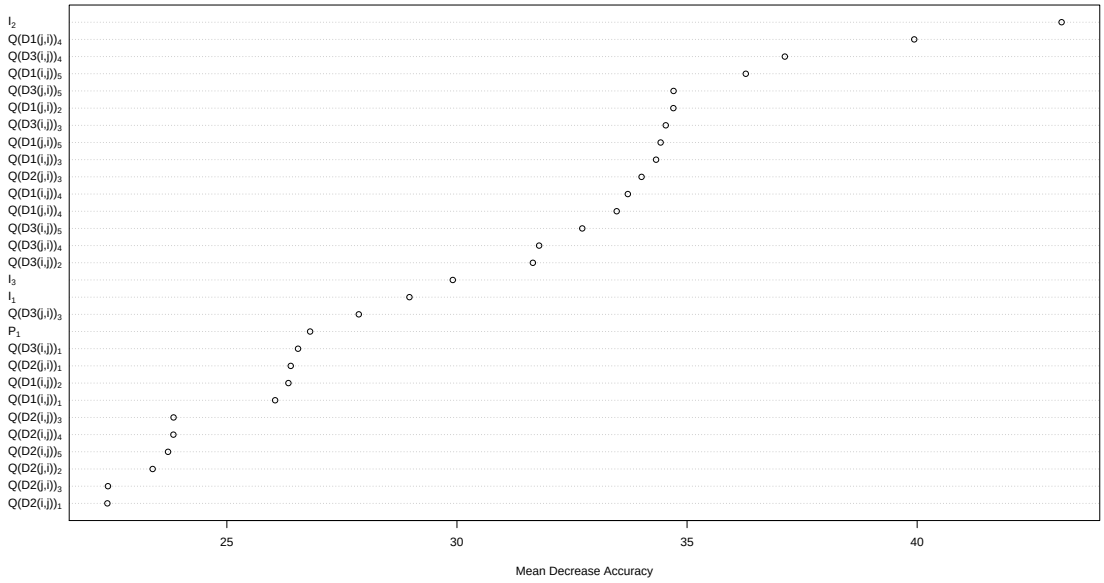


Figure 2: Importance of D2C features returned by the Random Forest mean decrease accuracy. I_i denotes the ith component of the descriptor vector (18) while $Q(Dx(i,j))_k$ denotes the kth quantile of the population of descriptor $Dx(i,j)$.

	GS	IAMB	IAMBnPC	interIAMBnPC	mRMR	mIMR
W-L	48-3 (32-0)	43-8 (21-0)	46-5 (26-0)	46-5 (25-0)	42-9 (17-0)	34-17 (12-0)

Table 3: D2C trained on synthetic data only: number of data sets for which D2C has an AUPRC (significantly (pval < 0.05)) higher/lower than the method in the column. W-L stands for Wins-Losses.

	GS	IAMB	IAMBnPC	interIAMBnPC	mRMR	mIMR
W-L	49-2 (36-0)	49-2 (27-0)	49-2 (32-0)	49-2 (32-0)	42-9 (17-0)	46-5 (19-1)

Table 4: D2C trained on synthetic data and 52 training data sets: number of data sets for which the D2C has an AUPRC (significantly (pval < 0.05)) higher/lower than the method in the column. W-L stands for Wins-Losses.

In this section we compare D2C to a set of algorithms implemented by the *Causal Explorer* software (Aliferis et al., 2003)[9]:

- `GS`: Grow/Shrink algorithm
- `IAMB`: Incremental Association-Based Markov Blanket
- `IAMBnPC`: IAMB with PC algorithm in the pruning phase
- `interIAMBnPC`: IAMB with PC algorithm in the interleaved pruning phase

and two filters based on information theory, mRMR (Peng et al., 2005) and mIMR (Bontempi and Meyer, 2010). The comparison is done as follows: for each data set and for each node (having at least a parent) the causal inference techniques return the ranking of the inferred parents. The ranking is assessed in terms of the average of Area Under the Precision Recall Curve (AUPRC) and a t-test is used to assess if the set of AUPRC values is significantly different between two methods. Note that the higher the AUPRC the more accurate is the inference method.

The summary of the paired comparisons is reported in Table 3 for the D2C algorithm trained on the synthetic data only and in Table 4 for the D2C algorithm trained on both synthetic data and the 52 training data sets.

It is worthy to remark that

- the D2C algorithm is extremely competitive and outperforms the other techniques taken into consideration,
- the D2C algorithm is able to generalize to DAG with different number of nodes and different distributions also when trained only on synthetic data simulated on linear DAGs,

9. Note that we use *Causal Explorer* here because, unlike `bnlearn` which estimates the entire adjacency matrix, it returns a ranking of the inferred causes for a given node.

- the D2C algorithm takes advantage from the availability of more training data and in particular of training data related to the causal inference task of interest, as shown by the improvement of the accuracy from Table 3 to Table 4,

- the two filters (mRMR and mIMR) algorithm appears to be the least inaccurate among the state-of-the-art algorithms,

- though the D2C is initialized with the results returned by the mIMR algorithm, it is able to improve its output and to significantly outperform it.

5. Conclusion

Two attitudes are common with respect to causal inference for observational data. The first is pessimistic and motivated by the consideration that *correlation (or dependency) does not imply causation*. The second is optimistic and driven by the fact that *causation implies correlation (or dependency)*. This paper belongs evidently to the second school of thought and relies on the confidence that causality leaves footprints in the form of stochastic dependency and that these footprints can be detected to retrieve causality from observational data. The results of the ChaLearn challenge and the preliminary results of this paper confirm the potential of machine learning approaches in predicting the existence of causality links on the basis of statistical descriptors of the dependency. We are convinced that this will open a new research direction where learning techniques may be used to reduce the degree of uncertainty about the existence of a causal relationships also in indistinguishable configurations which are typically not addressed by conditional independence approaches.

Further work will focus on 1) discovering additional features of multivariate distributions to improve the accuracy 2) addressing and assessing other related classification problems (e.g. predicting if a variable is an ancestor or descendant of a given one) 3) extending the work to partial ancestral graphs (Zhang, 2008) (e.g. exploiting the logical relations presented in Claassen and Heskes (2011)) extending the validation to real data sets and configurations with a still larger number of variables (e.g. network inference in bioinformatics).

Acknowledgments

This work was supported by the ARC project "Discovery of the molecular pathways regulating pancreatic beta cell dysfunction and apoptosis in diabetes using functional genomics and bioinformatics" funded by the Communauté Française de Belgique and the BridgeIRIS project funded by INNOVIRIS, Brussels Region. The authors wish to thank the editor and the anonymous reviewers for their insightful comments and remarks.

References

C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification. *Journal of Machine Learning Research*, 11:171–234, 2010.

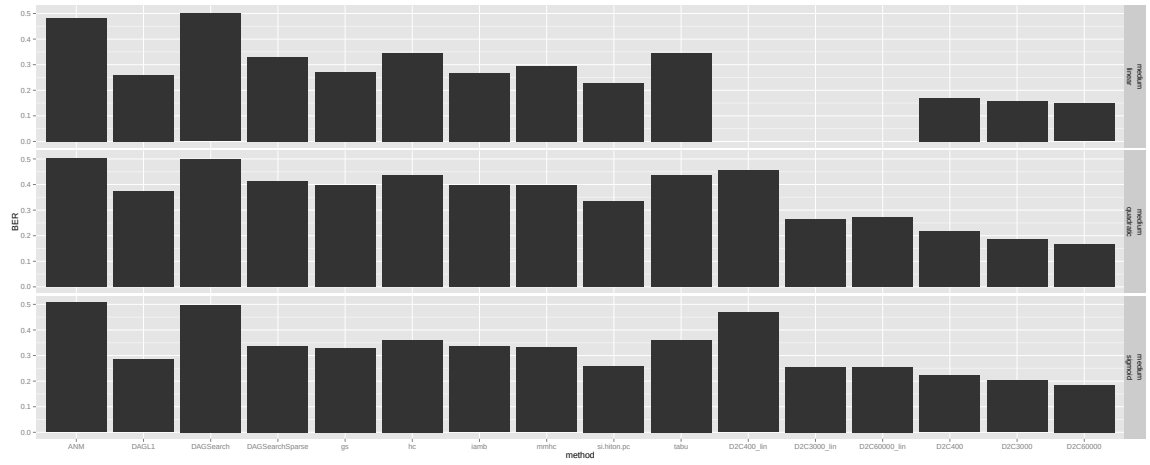


Figure 3: Balanced Error Rate of the different methods for small size DAGs and three types of dependency (top: linear, middle: quadratic, bottom: sigmoid). The notation D2Cx stands for D2C with a training set of size x and where training and test sets are based on DAGs with the same type of dependency. The notation D2Cx_lin stands for D2C with a training set of size x based on DAGs with linear dependency only.

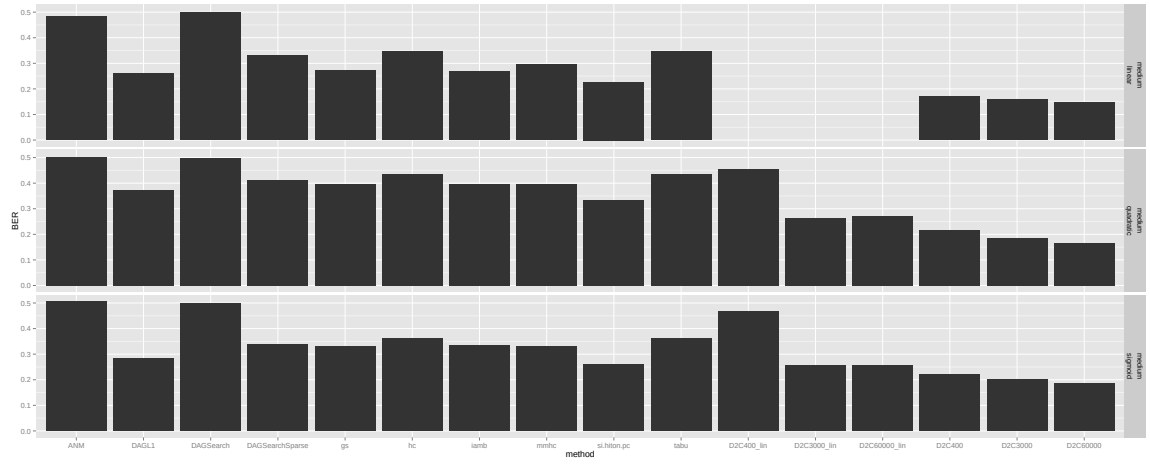


Figure 4: Balanced Error Rate of the different methods for medium size DAGs and three types of dependency (top: linear, middle: quadratic, bottom: sigmoid). The notation D2Cx stands for D2C with a training set of size x and where training and test sets are based on DAGs with the same type of dependency. The notation D2Cx_lin stands for D2C with a training set of size x based on DAGs with linear dependency only.

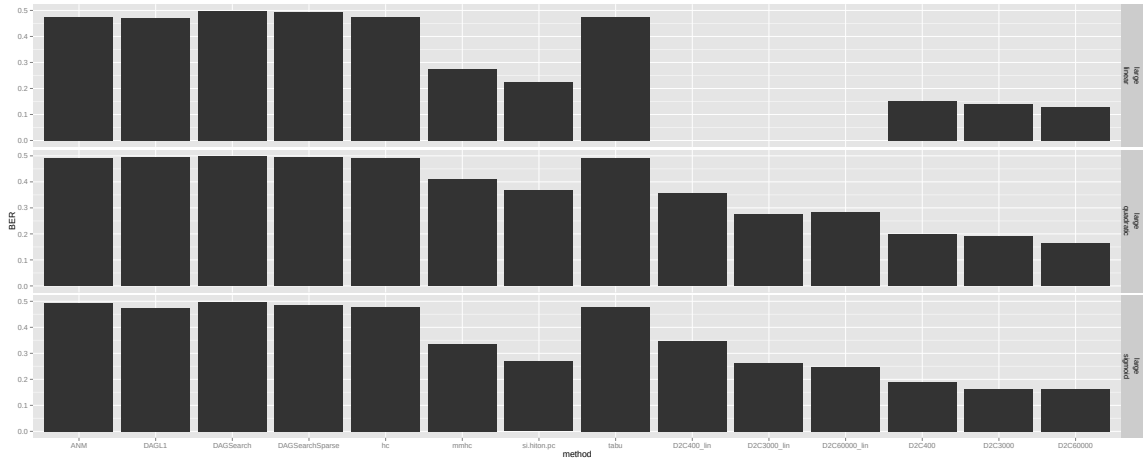


Figure 5: Balanced Error Rate of the different methods for large size DAGs and three types of dependency (top: linear, middle: quadratic, bottom: sigmoid). The notation D2Cx stands for D2C with a training set of size x and where training and test sets are based on DAGs with the same type of dependency. The notation D2Cx_lin stands for D2C with a training set of size x based on DAGs with linear dependency only.

C.F. Aliferis, I. Tsamardinos, and A. Statnikov. Causal explorer: A probabilistic network learning toolkit for biomedical discovery. In *Proceedings of METMBS*, 2003.

G. Bontempi and P.E. Meyer. Causal filter selection in microarray data. In *Proceedings of ICML*, 2010.

G. Bontempi, M. Birattari, and H. Bersini. Lazy learning for modeling and control design. *International Journal of Control*, 72(7/8):643–658, 1999.

G. Bontempi, B. Haibe-Kains, C. Desmedt, C. Sotiriou, and J. Quackenbush. Multiple-input multiple-output causal strategies for gene selection. *BMC Bioinformatics*, 12(1): 458, 2011.

G. Bontempi, C. Olsen, and M. Flauder. *D2C: Predicting Causal Direction from Dependency Features*, 2014. URL `http://CRAN.R-project.org/package=D2C`. R package version 1.1.

T. Claassen and T. Heskes. A logical characterization of constraint-based causal discovery. In *Proceedings of UAI*, 2011.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley, New York, 1990.

R. Daly and Q. Shen. Methods to accelerate the learning of bayesian network structures. In *Proceedings of the UK Workshop on Computational Intelligence*, 2007.

P. Daniusis, D. Janzing, J. Mooij, J. Zscheischler, B. Steudel, K. Zhang, and B. Schlkopf. Inferring deterministic causal relations. In *Proceedings of UAI*, pages 143–150, 2010.

C. Dethlefsen and S. Højsgaard. A common platform for graphical models in R: The gRbase package. *Journal of Statistical Software*, 14(17):1–12, 2005. URL `http://www.jstatsoft.org/v14/i17/`.

N. Friedman, M. Linial, I. Nachman, and Dana Pe'er. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7, 2000.

D. Geiger, T. Verma, and J. Pearl. Identifying independence in bayesian networks. *Networks*, 20, 1990.

I. Guyon. Results and analysis of the 2013 ChaLearn cause-effect pair challenge. In *Proceedings of NIPS 2013 Workshop on Causality: Large-scale Experiment Design and Inference of Causal Mechanisms*, 2014.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

I. Guyon, C. Aliferis, and A. Elisseeff. *Computational Methods of Feature Selection*, chapter Causal Feature Selection, pages 63–86. Chapman and Hall, 2007.

PO Hoyer, D. Janzing, J. Mooij, J. Peters, and B. Scholkopf. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems*, pages 689–696, 2009.

D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniusis, B. Steudel, and B. Scholkopf. Information-geometric approach to inferring causal directions. *Artificial Intelligence*, 2012.

M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11): 1–26, 2012. URL `http://www.jstatsoft.org/v47/i11/`.

D. Koller and N. Friedman. *Probabilistic Graphical Models*. The MIT Press, 2009.

A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL `http://CRAN.R-project.org/doc/Rnews/`.

D. Margaritis. *Learning Bayesian Network Model Structure from Data*. PhD thesis, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 2003.

G.J. McLaughlan. *Finite Mixture Models*. Wiley, 2000.

P.E. Meyer and G. Bontempi. *Biological Knowledge Discovery Handbook*, chapter Information-theoretic gene selection in expression data. IEEE Computer Society, 2014.

JM Mooij, O. Stegle, D. Janzing, K. Zhang, and B. Schlkopf. Probabilistic latent variable models for distinguishing between cause and effect. In *Advances in Neural Information Processing Systems*, 2010.

J. Pearl. Causal diagrams for empirical research. *Biometrika*, 82:669–710, 1995.

J. Pearl. *Causality: Models, Reasoning, and Inference.* Cambridge University Press, 2000.

J.P. Pellet and A. Elisseeff. Using markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9:1295–1342, 2008.

H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency,max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

O. Pourret, P. Nam, and B. Marcot. *Bayesian Networks: A Practical Guide to Applications.* Wiley, 2008.

H. Reichenbach. *The Direction of Time.* University of California Press, Berkeley, 1956.

M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using l1-regularization paths. In *Proceedings of AAAI*, 2007.

Marco Scutari. Learning bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010. URL `http://www.jstatsoft.org/v35/i03/`.

S. Shimizu, P.O. Hoyer, A. Hyvrinen, and A.J. Kerminen. A linear, non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7:2003–2030, 2006.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction and Search.* Springer Verlag, Berlin, 2000.

A. Statnikov, M. Henaff, N.I. Lytkin, and C. F. Aliferis. New methods for separating causes from effects in genomics data. *BMC Genomics*, 13(S22), 2012.

I. Tsamardinos, CF Aliferis, and A Statnikov. Time and sample efficient discovery of markov blankets and direct causal relations. In *Proceedings of KDD*, pages 673–678, 2003a.

I. Tsamardinos, C.F. Aliferis, and A. Statnikov. Algorithms for large scale markov blanket discovery. In *Proceedings of FLAIRS*, 2003b.

I. Tsamardinos, LE Brown, and CF Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, 2010.

J. Zhang. Causal reasoning with ancestral graphs. *Journal of Machine Learning Research*, 9:1437–1474, 2008.

Journal of Machine Learning Research 16 (2015) 2459-2463 Submitted 3/15; Revised 6/15; Published 12/15

The Libra Toolkit for Probabilistic Models

Daniel Lowd LOWD@CS.UOREGON.EDU
Amirmohammad Rooshenas PEDRAM@CS.UOREGON.EDU
Department of Computer and Information Science
University of Oregon
Eugene, OR 97403, USA

Editor: Antti Honkela

Abstract

The Libra Toolkit is a collection of algorithms for learning and inference with discrete probabilistic models, including Bayesian networks, Markov networks, dependency networks, and sum-product networks. Compared to other toolkits, Libra places a greater emphasis on learning the structure of tractable models in which exact inference is efficient. It also includes a variety of algorithms for learning graphical models in which inference is potentially intractable, and for performing exact and approximate inference. Libra is released under a 2-clause BSD license to encourage broad use in academia and industry.

Keywords: probabilistic graphical models, structure learning, inference

1. Introduction

The Libra Toolkit is a collection of algorithms for learning and inference with probabilistic models in discrete domains. What distinguishes Libra from other toolkits is the types of methods and models it supports. Libra includes a number of algorithms for *structure learning for tractable probabilistic models* in which exact inference can be done efficiently. Such models include sum-product networks (SPN), mixtures of trees (MT), and Bayesian and Markov networks with compact arithmetic circuits (AC). These learning algorithms are not available in any other open-source toolkit. Libra also supports *structure learning for graphical models*, such as Bayesian networks (BN), Markov networks (MN), and dependency networks (DN), in which inference is not necessarily tractable. Some of these methods are unique to Libra as well, such as using dependency networks to learn Markov networks. Libra provides a variety of exact and approximate inference algorithms for answering probabilistic queries in learned or manually specified models. Many of these are designed to exploit local structure, such as conjunctive feature functions or tree-structured conditional probability distributions.

The overall goal of Libra is to make these methods available to researchers, practitioners, and students for use in experiments, applications, and education. Each algorithm in Libra is implemented in a command-line program suitable for interactive use or scripting, with consistent options and file formats throughout the toolkit. Libra also supports the development of new algorithms through modular code organization, including shared libraries for different representations and file formats.

	Learning General Models	
○	BN structure with tree CPDs	(Chickering et al., 1997)
	DN structure with tree/boosted tree/LR CPDs	(Heckerman et al., 2000)
•	MN structure from DNs	(Lowd, 2012)
	MN parameters (pseudo-likelihood)	
	Learning Tractable Models	
•	Tractable BN/AC structure	(Lowd and Domingos, 2008)
•	Tractable MN/AC structure	(Lowd and Rooshenas, 2013)
•	Mixture of trees (MT)	(Meila and Jordan, 2000)
•	SPN structure (ID-SPN algorithm)	(Rooshenas and Lowd, 2014)
	Chow-Liu algorithm	(Chow and Liu, 1968)
•	AC parameters (maximum likelihood)	
	Approximate Inference	
	Gibbs sampling (BN,MN,•DN)	(Heckerman et al., 2000) (DN)
	Mean field (BN,MN,•DN)	(Lowd and Shamaei, 2011) (DN)
	Loopy belief propagation (BN,MN)	
	Max-product (BN,MN)	
	Iterated conditional modes (BN,MN,•DN)	
•	Variational optimization of ACs	(Lowd and Domingos, 2010)
	Exact Inference	
○	AC variable elimination (BN,MN)	(Chavira and Darwiche, 2007)
○	Marginal and MAP inference (AC,SPN,MT)	(Darwiche, 2003)

Table 1: Learning and inference algorithms implemented in Libra. Filled circles (•) indicate algorithms that are unique to Libra, and hollow circles (○) indicate algorithms with no other open-source implementation.

Libra is available under a modified (2-clause) BSD license, which allows modification and reuse in both academia and industry. Libra's source code and documentation can be found at `http://libra.cs.uoregon.edu`.

2. Functionality

Libra includes a variety of learning and inference algorithms, many of which are not available in any other open-source toolkit. See Table 1 for a brief overview.

Libra's command-line syntax is designed to be simple. For example, to learn a tractable BN, run the command: "`libra acbn -i train.data -mo model.bn -o model.ac`" where `train.data` is the input data, `model.bn` is the filename for saving the learned BN, and `model.ac` is the filename for the corresponding AC representation, which allows for efficient, exact inference. To compute exact conditional marginals in the learned model: "`libra acquery -m model.ac -ev test.ev -marg`". To compute approximate marginals in the BN with loopy belief propagation: "`libra bp -m model.bn -ev test.ev`". Additional command-line parameters can be used to specify other options, such as the priors and heuristics used by `acbn` or the maximum number of iterations for `bp`. These are just three of more than twenty commands included in Libra.

Libra supports a variety of file formats. For data instances, Libra uses comma separated values, where each value is a zero-based index indicating the discrete value of the corresponding variable. For evidence and query files, unknown or missing values are represented with the special value "`*`". For model files, Libra supports the XMOD representation from

	Representation		Inference		Learning	
Toolkit	Model Types	Factors	Exact	Approx.	Param.	Structure
Libra	BN,MN,DN,SPN,AC	Tree,Feature	ACVE	G,BP,MF	ML,PL	BN,...,AC
FastInf	BN,MN	Table	JT	Many	ML,EM	-
libDAI	BN,MN	Table	JT,E	Many	ML,EM	-
OpenGM2	BN,MN	Sparse	-	Many	-	-
Banjo	BN,DBN	Table	-	-	-	BN
BNT	BN,DBN,ID	LR,OR,NN	JT,VE,E	G,LW,BP	ML,EM	BN
Deal	BN	Table	-	-	-	BN
OpenMarkov	BN,MN,ID	Tree,ADD,OR	JT	LW	ML	BN,MN
SMILE	BN,DBN,ID	Table	JT	Sampling	ML,EM	BN
UnBBayes	BN,ID	Table	JT	G,LW	-	BN

Table 2: Comparison of Libra to several other probabilistic inference and learning toolkits.

the WinMine Toolkit, the Bayesian interchange format (BIF), and the simple representation from the UAI inference competition. Libra converts among these different formats using the provided `mconvert` utility, as well as to its own internal formats for BNs, MNs, and DNs (`.bn`, `.mn`, `.dn`). Libra has additional representations for ACs and SPNs (`.ac`, `.spn`). These formats are designed to be easy for humans to read and programs to parse.

Libra is implemented in OCaml. OCaml is a statically typed language that supports functional and imperative programming styles, compiles to native machine code on multiple platforms, and uses type inference and garbage collection to reduce programmer errors and effort. OCaml has a good foreign function interface, which Libra uses for linking to C libraries and a few memory-intensive subroutines. The code to Libra includes nine support libraries, which provide modules for input, output, and representation of different types of models, as well as commonly used algorithms and utility methods.

3. Comparison to Other Toolkits

In Table 2, we compare Libra to other toolkits in terms of representation, learning, and inference.

In terms of representation, Libra is the only open-source software package that supports ACs and one of a very small number that support DNs or SPNs. Libra does not currently support dynamic Bayesian networks (DBN) or influence diagrams (ID). For factors, Libra supports tables, trees, and arbitrary conjunctive feature functions. BNT (Murphy, 2001) and OpenMarkov (CISIAD, 2013) also support additional types of CPDs, such as logistic regression, noisy-OR, neural networks, and algebraic decision diagrams, but they only support tabular CPDs for structure learning. OpenGM2 (Andres et al., 2012) supports sparse factors, but iterates through all factor states during inference. Libra is unique in its ability to learn models with local structure and exploit that structure in inference.

For exact inference, the most common algorithms are junction tree (JT), enumeration (E), and variable elimination (VE). Libra provides ACVE (Chavira and Darwiche, 2007), which is similar to building a junction tree, but it can exploit structured factors to run inference in many high-treewidth models. For approximate inference, Libra provides Gibbs sampling (G), loopy belief propagation (BP), and mean field (MF), all of which are optimized for structured factors. A few learning toolkits offer likelihood weighting (LW) or

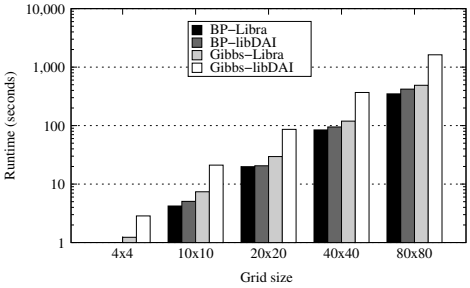


Figure 1: Running time of belief propagation and Gibbs sampling in Libra and libDAI, evaluated on grid-structured MNs of various sizes.

additional sampling algorithms for BNs. FastInf (Jaimovich et al., 2010), libDAI (Mooij, 2010), and OpenGM2 offer the most algorithms but only support tables.

For learning, Libra supports maximum likelihood (ML) parameter learning for BNs, ACs, and SPNs, and pseudo-likelihood (PL) optimization for MNs and DNs. Libra does not yet support expectation maximization (EM) for learning with missing values. Structure learning is one of Libra's greatest strengths. Most toolkits only provide algorithms for learning BNs with tabular CPDs or MNs using the PC algorithm (Spirtes et al., 1993). Libra includes methods for learning BNs, MNs, DNs, SPNs, and ACs, and all of its algorithms support learning with local structure.

In experiments on grid-structured MNs, Libra's implementations of BP and Gibbs sampling were at least as fast as libDAI, a popular C++ implementation of many inference algorithms. The accuracy of both toolkits was equivalent. Parameter settings, such as the number of iterations, were identical. See Figure 1 for more details.

4. Conclusion

The Libra Toolkit provides algorithms for learning and inference in a variety of probabilistic models, including BNs, MNs, DNs, SPNs, and ACs. Many of these algorithms are not available in any other open-source software. Libra's greatest strength is its support for tractable probabilistic models, for which very little other software exists. Libra makes it easy to use these state-of-the-art methods in experiments and applications, which we hope will accelerate the development and deployment of probabilistic methods.

Acknowledgments

The development of Libra was partially supported by ARO grant W911NF-08-1-0242, NSF grant IIS-1118050, NIH grant R01GM103309, and a Google Faculty Research Award. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, NIH, or the United States Government.

References

B. Andres, T. Beier, and J. H. Kappes. OpenGM: A C++ library for discrete graphical models. *ArXiv e-prints*, 2012. URL `http://arxiv.org/abs/1206.0111`.

M. Chavira and A. Darwiche. Compiling Bayesian networks using variable elimination. In *IJCAI*, pages 2443–2449, 2007.

D. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *UAI*, pages 80–89, 1997.

C. K. Chow and C. N Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

Research Center on Intelligent Decision-Support Systems (CISIAD). OpenMarkov 0.1.3. 2013. http://www.openmarkov.org.

A. Darwiche. A differential approach to inference in Bayesian networks. *JACM*, 50(3): 280–305, 2003.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *JMLR*, 1:49–75, 2000.

A. Jaimovich, O. Meshi, I. McGraw, and G. Elidan. FastInf: An efficient approximate inference library. *JMLR*, 11:1733–1736, 2010.

D. Lowd. Closed-form learning of Markov networks from dependency networks. In *UAI*, 2012.

D. Lowd and P. Domingos. Learning arithmetic circuits. In *UAI*, 2008.

D. Lowd and P. Domingos. Approximate inference by compilation to arithmetic circuits. In *NIPS*, 2010.

D. Lowd and A. Rooshenas. Learning Markov networks with arithmetic circuits. In *AISTATS*, 2013.

D. Lowd and A. Shamaei. Mean field inference in dependency networks: An empirical study. In *AAAI*, 2011.

M. Meila and M. Jordan. Learning with mixtures of trees. *JMLR*, 1:1–48, 2000.

Joris M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173, 2010.

K. Murphy. The Bayes net toolbox for MATLAB. *Computing Sci. and Statistics*, 33:2001.

A. Rooshenas and D. Lowd. Learning sum-product networks with direct and indirect interactions. In *ICML*, 2014.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. Springer, New York, NY, 1993.

Journal of Machine Learning Research 16 (2015) 2465-2500 Submitted 11/14; Revised 6/15; Published 12/15

Complexity of Equivalence and Learning for Multiplicity Tree Automata

Ines Marušić INES.MARUSIC@CS.OX.AC.UK
James Worrell JBW@CS.OX.AC.UK
Department of Computer Science
University of Oxford
Parks Road, Oxford OX1 3QD, UK

Editor: Alexander Clark

Abstract

We consider the query and computational complexity of learning multiplicity tree automata in Angluin's exact learning model. In this model, there is an oracle, called the Teacher, that can answer membership and equivalence queries posed by the Learner. Motivated by this feature, we first characterise the complexity of the equivalence problem for multiplicity tree automata, showing that it is logspace equivalent to polynomial identity testing.

We then move to query complexity, deriving lower bounds on the number of queries needed to learn multiplicity tree automata over both fixed and arbitrary fields. In the latter case, the bound is linear in the size of the target automaton. The best known upper bound on the query complexity over arbitrary fields derives from an algorithm of Habrard and Oncina (2006), in which the number of queries is proportional to the size of the target automaton and the size of a largest counterexample, represented as a tree, that is returned by the Teacher. However, a smallest counterexample tree may already be exponential in the size of the target automaton. Thus the above algorithm has query complexity exponentially larger than our lower bound, and does not run in time polynomial in the size of the target automaton.

We give a new learning algorithm for multiplicity tree automata in which counterexamples to equivalence queries are represented as DAGs. The query complexity of this algorithm is quadratic in the target automaton size and linear in the size of a largest counterexample. In particular, if the Teacher always returns DAG counterexamples of minimal size then the query complexity is quadratic in the target automaton size—almost matching the lower bound, and improving the best previously-known algorithm by an exponential factor.

Keywords: exact learning, query complexity, multiplicity tree automata, Hankel matrices, DAG representations of trees, polynomial identity testing

1. Introduction

Trees are a basic object in computer science and a natural model of hierarchical data, such as syntactic structures in natural language processing and XML data on the web. Trees arise across a broad range of applications, including natural text and speech processing, computer vision, bioinformatics, web information extraction, and social network analysis. Many of these applications require representing probability distributions over trees and more

general functions from trees into the real numbers. A broad class of such functions can be defined by *multiplicity tree automata*, a powerful algebraic model which strictly generalises probabilistic tree automata.

Multiplicity tree automata were introduced by Berstel and Reutenauer (1982) under the terminology of linear representations of tree series. They augment classical finite tree automata by assigning to each transition a value in a field. They also generalise multiplicity word automata, introduced by Schützenberger (1961), since words are a special case of trees. Multiplicity tree automata define many natural structural properties of trees and can be used to model probabilistic processes running on trees. Multiplicity word and tree automata have been applied to a wide variety of machine learning problems, including speech recognition, image processing, character recognition, and grammatical inference; see the paper of Balle and Mohri (2012) for references.

The task of learning automata from examples and queries has been extensively studied since the 1960s. Two notable results in this domain show the impossibility of efficiently learning deterministic finite automata from positive and negative examples alone. First, Gold (1978) showed that the problem of exactly identifying the smallest deterministic finite automaton consistent with a set of accepted and rejected words is NP-hard. Later, Kearns and Valiant (1994) showed that the concept class of regular languages is not efficiently PAC learnable using any polynomially-evaluable hypothesis class under standard cryptographic assumptions.

A significant positive result on learning regular languages was achieved by Angluin (1987), who considered a Learner that did not just passively receive data but that was also able to ask queries. Specifically, Angluin considered *membership queries*, in which the Learner asks an oracle whether a given word belongs to the target language, and *equivalence queries*, in which the Learner asks an oracle whether a hypothesis is correct, obtaining a counterexample if it is not. Subsequent research has sought to establish the learnability of many other hypothesis classes in the same setting, including classes of Boolean formulae, decision trees, context-free languages, and polynomials; see the book of Kearns and Vazirani (1994, Chapter 8) for more details and references.

In this paper we study the problem of learning multiplicity tree automata in the *exact learning model* of Angluin (1988), outlined above. Formally, in this model a Learner actively collects information about the target function from a Teacher through *membership queries*, which ask for the value of the function on a specific input, and *equivalence queries*, which suggest a hypothesis to which the Teacher provides a counterexample if one exists. A class of functions $\mathscr{C}$ is *exactly learnable* if there exists an exact learning algorithm such that for any function $f \in \mathscr{C}$, the Learner identifies f using polynomially many membership and equivalence queries in the size of a shortest representation of f and the size of a largest counterexample returned by the Teacher during the execution of the algorithm. The exact learning model is an important theoretical model of the learning process. It is well known that learnability in the exact learning model also implies learnability in the PAC model with membership queries (Valiant, 1985).

We are interested in questions of succinctness and computational efficiency, both from the point of view of the Teacher and the Learner. From the point of view of the Teacher, one of the main questions is checking *equivalence* of multiplicity tree automata, i.e., whether two multiplicity tree automata define the same function on trees. Seidl (1990) proved that

equivalence of multiplicity tree automata is decidable in polynomial time assuming unit-cost arithmetic, and in randomised polynomial time in the usual bit-cost model. No finer analysis of the complexity of this problem exists to date. In contrast, the complexity of equivalence for classical nondeterministic word and tree automata has been completely characterised: PSPACE-complete over words (Aho et al., 1974) and EXPTIME-complete over trees (Seidl, 1990).

Our first contribution, in Section 3, is to show that the equivalence problem for multiplicity tree automata is logspace equivalent to polynomial identity testing, i.e., the problem of deciding whether a polynomial given as an arithmetic circuit is zero. The latter problem is known to be solvable in randomised polynomial time (DeMillo and Lipton, 1978; Schwartz, 1980; Zippel, 1979), whereas solving it in deterministic polynomial time is a well-studied and longstanding open problem (see Arora and Barak, 2009).

Our second contribution, in Section 5, is to give lower bounds on the number of queries needed to learn multiplicity tree automata in the exact learning model, both for the case of an arbitrary and a fixed underlying field. The bound in the former case is linear in the automaton size. In the latter case, the bound is linear in the automaton size for alphabets of a fixed maximal rank. To the best of our knowledge, these are the first lower bounds on the query complexity of exactly learning multiplicity tree automata.

Habrard and Oncina (2006) give an algorithm for learning multiplicity tree automata in the exact learning model. Consider a target multiplicity tree automaton whose minimal representation A has n states. The algorithm of Habrard and Oncina, *op. cit.*, makes at most n equivalence queries and number of membership queries proportional to $|A| \cdot s$, where $|A|$ is the size of A and s is the size of a largest counterexample returned by the Teacher. Since this algorithm assumes that the Teacher returns counterexamples represented explicitly as trees, s can be exponential in $|A|$, even for a Teacher that returns counterexamples of minimal size (see Example 3). This observation reveals an exponential gap between the query complexity of the algorithm of Habrard and Oncina (2006) and our above-mentioned lower bound, which is only linear in $|A|$. Another consequence is that the worst-case time complexity of this algorithm is exponential in the size of the target automaton.

Given two inequivalent multiplicity tree automata with n states in total, the algorithm of Seidl (1990) produces a subtree-closed set of trees of cardinality at most n that contains a tree on which the automata differ. It follows that the counterexample contained in this set has at most n subtrees, and hence can be represented as a DAG with at most n vertices (see Section 3.2). Thus in the context of exact learning it is natural to consider a Teacher that can return succinctly-represented counterexamples, i.e., trees represented as DAGs.

DAGs have been used as succinct representations of trees in a number of domains, including classification problems (Sperduti and Starita, 1997) and query evaluation for XML (Buneman et al., 2003; Frick et al., 2003). Tree automata that run on DAG representations of finite trees were first introduced by Charatonik (1999) as extensions of ordinary tree automata, and were further studied by Anantharaman et al. (2005). The automata considered by Charatonik (1999) and Anantharaman et al. (2005) run on fully-compressed DAGs. Fila and Anantharaman (2006) extend this definition by introducing tree automata that run on DAGs that may be partially compressed. In this paper, we employ the latter framework in the context of learning multiplicity automata.

In Section 4, we present a new exact learning algorithm for multiplicity tree automata that achieves the same bound on the number of equivalence queries as the algorithm of Habrard and Oncina (2006), while using number of membership queries quadratic in the target automaton size and linear in the largest counterexample size, even when counterexamples are given as DAGs. Assuming that the Teacher provides minimal DAG representations of counterexamples, our algorithm therefore makes quadratically many queries in the target automaton size. This is exponentially fewer queries than the best previously-known algorithm (Habrard and Oncina, 2006) and quadratic in the above-mentioned lower bound. Furthermore, our algorithm performs a quadratic number of arithmetic operations in the size of the target automaton, and can be implemented in randomised polynomial time in the Turing model.

Like the algorithm of Habrard and Oncina (2006), our algorithm constructs a matricial representation of the target automaton, called the *Hankel matrix* (Carlyle and Paz, 1971; Fliess, 1974). However on receiving a counterexample tree z, the former algorithm adds a new column to the Hankel matrix for every suffix of z, while our algorithm adds (at most) one new row for each subtree of z. Crucially the number of suffixes may be exponential in the size of a DAG representation of z, whereas the number of subtrees is only linear in the size of a DAG representation.

An extended abstract (Marušić and Worrell, 2014) of this work appeared in the proceedings of MFCS 2014. The current paper contains full proofs of all results reported there, the formal definition of multiplicity tree automata running on DAGs, and a refined complexity analysis of the learning algorithm.

1.1 Related Work

One of the earliest results about the exact learning model was the proof of Angluin (1987) that deterministic finite automata are learnable. This result was generalised by Drewes and Högberg (2007) to show exact learnability of deterministic finite (bottom-up) tree automata, generalising also a result of Sakakibara (1990) on the exact learnability of context-free grammars from their structural descriptions[1].

The learning algorithm of Drewes and Högberg (2007) was generalised by Maletti (2007) to show that deterministic weighted tree automata over a (commutative) semifield are exactly learnable, generalising also an earlier result of Drewes and Vogler (2007) which was restricted to the class of deterministic *all-accepting* (i.e., every final weight is non zero) weighted tree automata. Recently, a unifying framework for exact learning of deterministic weighted tree automata over a semifield has been proposed (Drewes et al., 2011). Specifically, Drewes et al., *op. cit.*, introduce the notion of *abstract observation tables*, an abstract data type for learning deterministic weighted tree automata in the exact learning model, and show that every correct implementation of abstract observation tables yields a correct learning algorithm.

Exact learnability of nondeterministic weighted automata over a field (here called *multiplicity automata*) has also been extensively studied. Beimel et al. (2000) show that multiplicity word automata can be learned efficiently, and apply this to learn various classes of DNF formulae and polynomials. These results were generalised by Klivans and Shpilka

1. *Structural descriptions* of a context-free grammar are unlabelled derivation trees of the grammar.

(2006) to show exact learnability of restricted algebraic branching programs and noncommutative set-multilinear arithmetic formulae. Bisht et al. (2006) give an almost tight (up to a *log* factor) lower bound on the number of queries made by any exact learning algorithm for the class of multiplicity word automata.

An exact learning algorithm for a class of nondeterministic tree automata, namely *residual finite* tree automata, is given by Kasprzik (2013). The latter paper identifies the size of counterexamples as a hidden exponential factor in the complexity of the learning algorithm, observing in particular that a smallest counterexample can have exponential size in the number of states of the target automaton. Such a phenomenon does not prevent the class of tree automata from being exactly learnable since in the exact learning model the complexity measure takes into account the size of a largest counterexample. However, this does raise the question of developing a learning algorithm whose complexity would be polynomial in the size of succinctly-represented counterexamples, which is one of the motivations for the present work.

Denis and Habrard (2007) consider the problem of learning probability distributions over trees that are recognised by a multiplicity tree automaton from samples drawn independently according to the target distribution. They give an inference algorithm that exactly identifies such recognisable probability distributions in the limit with probability one (with respect to the randomly-drawn examples). Most closely related to the topic of the present paper is the work of Habrard and Oncina (2006), who give an algorithm for learning multiplicity tree automata in the exact learning model, as discussed above.

A variety of spectral methods have been employed for learning multiplicity word and tree automata (Bailly et al., 2009; Balle and Mohri, 2012; Denis et al., 2014; Gybels et al., 2014). This line of research originates in earlier work of Hsu et al. (2012) that gives a spectral learning algorithm (based on singular value decomposition) for hidden Markov models. Particularly close to the present paper is the work of Bailly et al. (2010), which learns probability distributions over trees that are recognised by some multiplicity tree automaton. Their approach lies within a passive learning framework in which one is given a sample of trees independently drawn according to a target distribution, and the aim is to infer a multiplicity tree automaton that approximates the target. As in our approach, the notion of a Hankel matrix plays a central role in the algorithm of Bailly et al. (2010). There the Hankel matrix is called an *observation matrix*, and it encodes an empirical distribution on trees obtained by sampling from the target distribution. Bailly et al., *op. cit.*, apply principal component analysis in order to identify a low-dimensional approximation of the vector space spanned by the residuals of the target probability distribution. From this approximation they build an automaton whose associated tree series approximates the target distribution. They moreover obtain bounds on the estimation error of the output tree series with respect to the target distribution in terms of the sample size and the desired confidence.

In contrast to the above-described approach of Bailly et al. (2010), in our work the target dimension (i.e., number of states) is not part of the input since our aim is to learn a *minimal* multiplicity tree automaton that exactly represents the target tree series. Moreover, in the present paper the entries of the Hankel matrix are determined by active queries rather than passive observations, and the learning process continues until we know a sufficient number of entries to be able to exactly construct a representation of the target.

2. Preliminaries

Let $\mathbb{N}$ and $\mathbb{N}_0$ denote the set of all positive and nonnegative integers, respectively. Let $n \in \mathbb{N}$. We write $[n]$ for the set $\{1, 2, \ldots, n\}$ and I_n for the identity matrix of order n. For every $i \in [n]$, we write e_i for the i^{th} n-dimensional coordinate row vector. For any n-dimensional vector v, we write v_i for its i^{th} entry.

For any matrix A, we write A_i for its i^{th} row, A^j for its j^{th} column, and $A_{i,j}$ for its $(i,j)^{\text{th}}$ entry. Given nonempty subsets I and J of the rows and columns of A, respectively, we write $A_{I,J}$ for the submatrix $(A_{i,j})_{i \in I, j \in J}$ of A. For singletons, we write simply $A_{i,J} := A_{\{i\},J}$ and $A_{I,j} := A_{I,\{j\}}$. We also consider matrices whose rows and columns are indexed by tuples of natural numbers ordered lexicographically.

Given a set V, we denote by V^* the set of all finite ordered tuples of elements from V. For any subset $S \subseteq V$, the *characteristic function* of S (relative to V) is the function $\chi_S : V \to \{0,1\}$ such that $\chi_S(x) = 1$ if $x \in S$, and $\chi_S(x) = 0$ otherwise.

2.1 Kronecker Product

Let A be a matrix of dimension $m_1 \times n_1$ and B a matrix of dimension $m_2 \times n_2$. The *Kronecker product* of A by B, written as $A \otimes B$, is a matrix of dimension $m_1 m_2 \times n_1 n_2$ where $(A \otimes B)_{(i_1,i_2),(j_1,j_2)} = A_{i_1,j_1} \cdot B_{i_2,j_2}$ for every $i_1 \in [m_1]$, $i_2 \in [m_2]$, $j_1 \in [n_1]$, $j_2 \in [n_2]$.

The Kronecker product is bilinear, associative, and has the following *mixed-product property*: For any matrices A, B, C, D such that products $A \cdot C$ and $B \cdot D$ are defined, it holds that $(A \otimes B) \cdot (C \otimes D) = (A \cdot C) \otimes (B \cdot D)$.

Let $k \in \mathbb{N}$ and $A_1, \ldots, A_k$ be matrices such that for every $l \in [k]$, matrix A_l has n_l rows. It can easily be shown using induction on k that for every $(i_1, \ldots, i_k) \in [n_1] \times \cdots \times [n_k]$, it holds that

$$(A_1 \otimes \cdots \otimes A_k)_{(i_1,\ldots,i_k)} = (A_1)_{i_1} \otimes \cdots \otimes (A_k)_{i_k}. \tag{1}$$

We write $\bigotimes_{l=1}^{k} A_l := A_1 \otimes \cdots \otimes A_k$.

For every $k \in \mathbb{N}_0$ we define the *k-fold Kronecker power* of a matrix A, written as $A^{\otimes k}$, inductively by $A^{\otimes 0} = I_1$ and $A^{\otimes k} = A^{\otimes(k-1)} \otimes A$ for $k \geq 1$.

Let $k \in \mathbb{N}_0$. For any square matrices A and B, we have

$$(A \otimes B)^k = A^k \otimes B^k. \tag{2}$$

For any matrices $A_1, \ldots, A_k$ and $B_1, \ldots, B_k$ where product $A_l \cdot B_l$ is defined for every $l \in [k]$, we have

$$(A_1 \otimes \cdots \otimes A_k) \cdot (B_1 \otimes \cdots \otimes B_k) = (A_1 \cdot B_1) \otimes \cdots \otimes (A_k \cdot B_k). \tag{3}$$

Equations (2) and (3) follow easily from the mixed-product property by induction on k.

2.2 Finite Trees

A *ranked alphabet* is a tuple (Σ, rk) where Σ is a nonempty finite set of symbols and $rk : \Sigma \to \mathbb{N}_0$ is a function. Ranked alphabet (Σ, rk) is often written Σ for short. For every

$k \in \mathbb{N}_0$, we define the set of all *k-ary* symbols $\Sigma_k := rk^{-1}(\{k\})$. If $\sigma \in \Sigma_k$ then we say that σ has *rank* (or *arity*) k. We say that Σ has *rank* m if $m = max\{rk(\sigma) : \sigma \in \Sigma\}$.

The set of Σ-*trees* (*trees* for short), written as T_Σ, is the smallest set T satisfying the following two conditions: (i) $\Sigma_0 \subseteq T$; and (ii) if $k \geq 1$, $\sigma \in \Sigma_k$, $t_1, \ldots, t_k \in T$ then $\sigma(t_1, \ldots, t_k) \in T$. Given a Σ-tree t, a *subtree* of t is a Σ-tree consisting of a node in t and all of its descendants in t. The set of all subtrees of t is denoted by $Sub(t)$.

Let Σ be a ranked alphabet and $\mathbb{F}$ be a field. A *tree series* over Σ with coefficients in $\mathbb{F}$ is a function $f : T_\Sigma \to \mathbb{F}$. For every $t \in T_\Sigma$, we call $f(t)$ the *coefficient* of t in f. The set of all tree series over Σ with coefficients in $\mathbb{F}$ is denoted by $\mathbb{F}\langle\langle T_\Sigma \rangle\rangle$.

We define the tree series $height, size, \#_\sigma \in \mathbb{Q}\langle\langle T_\Sigma \rangle\rangle$ where $\sigma \in \Sigma$, as follows: (i) if $t \in \Sigma_0$ then $height(t) = 0$, $size(t) = 1$, $\#_\sigma(t) = \chi_{\{t=\sigma\}}$; and (ii) if $t = a(t_1, \ldots, t_k)$ where $k \geq 1$, $a \in \Sigma_k$, $t_1, \ldots, t_k \in T_\Sigma$ then $height(t) = 1 + max_{i \in [k]} height(t_i)$, $size(t) = 1 + \sum_{i \in [k]} size(t_i)$, $\#_\sigma(t) = \chi_{\{a=\sigma\}} + \sum_{i \in [k]} \#_\sigma(t_i)$, respectively. For every $n \in \mathbb{N}_0$, we define the sets $T_\Sigma^{<n} := \{t \in T_\Sigma : height(t) < n\}$, $T_\Sigma^n := \{t \in T_\Sigma : height(t) = n\}$, and $T_\Sigma^{\leq n} := T_\Sigma^{<n} \cup T_\Sigma^n$.

Let $\Box$ be a nullary symbol not contained in Σ. The set C_Σ of Σ-*contexts* (*contexts* for short) is the set of all $(\{\Box\} \cup \Sigma)$-trees in which $\Box$ occurs exactly once. The *concatenation* of $c \in C_\Sigma$ and $t \in T_\Sigma \dot{\cup} C_\Sigma$, written as $c[t]$, is the tree obtained by substituting t for $\Box$ in c. Intuitively, the $\Box$-labelled leaf of c acts as a variable in that substituting a Σ-tree (respectively, Σ-context) t for that variable yields a new Σ-tree (Σ-context) $c[t]$.

A *suffix* of a Σ-tree t is a Σ-context c such that $t = c[t']$ for some Σ-tree t'. The *Hankel matrix* of a tree series $f \in \mathbb{F}\langle\langle T_\Sigma \rangle\rangle$ is the matrix $H : T_\Sigma \times C_\Sigma \to \mathbb{F}$ such that $H_{t,c} = f(c[t])$ for every $t \in T_\Sigma$ and $c \in C_\Sigma$.

2.3 Multiplicity Tree Automata

Let $\mathbb{F}$ be a field. An $\mathbb{F}$-*multiplicity tree automaton* ($\mathbb{F}$-*MTA*) is a quadruple $A = (n, \Sigma, \mu, \gamma)$ which consists of the *dimension* $n \in \mathbb{N}_0$ representing the number of states, a ranked alphabet Σ, a family of *transition matrices* $\mu = \{\mu(\sigma) : \sigma \in \Sigma\}$, where $\mu(\sigma) \in \mathbb{F}^{n^{rk(\sigma)} \times n}$, and the *final weight vector* $\gamma \in \mathbb{F}^{n \times 1}$. The *size* of the automaton A, written as $|A|$, is defined as

$$|A| := \sum_{\sigma \in \Sigma} n^{rk(\sigma)+1} + n.$$

That is, the size of A is the total number of entries in all transition matrices and the final weight vector.[2]

Example 1 *Let $\Sigma = \{0, 1, +, \times, -\}$ be a ranked alphabet where $0, 1$ are nullary symbols and $+, \times, -$ are binary symbols. We define an $\mathbb{F}$-multiplicity tree automaton $A = (2, \Sigma, \mu, \gamma)$ as follows. Automaton A has two states, q_1 and q_2, and has final weight vector $\gamma = \begin{bmatrix} 0 & 1 \end{bmatrix}^\top$. This means that states q_1 and q_2 have final weights $\gamma_1 = 0$ and $\gamma_2 = 1$, respectively. Given a symbol $\sigma \in \Sigma$ of rank k, the transition matrix $\mu(\sigma)$ has dimension $2^k \times 2$ and stores the weights of transitions from each k-tuple of origin states to each destination state. Let the*

2. We measure size assuming explicit rather than sparse representations of the transition matrices and final weight vector because minimal automata are only unique up to change of basis (see Theorem 4).

transition matrices of A be $\mu(0) = \begin{bmatrix}1 & 0\end{bmatrix}$, $\mu(1) = \begin{bmatrix}1 & 1\end{bmatrix}$,

$$\mu(+) = \begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 1\\ 0 & 0\end{bmatrix},\ \mu(-) = \begin{bmatrix}1 & 0\\ 0 & -1\\ 0 & 1\\ 0 & 0\end{bmatrix},\ \text{and}\ \mu(\times) = \begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}.$$

Entry $\mu(1)_2 = 1$ means that there is a transition $1 \xrightarrow{1} q_2$ with weight 1 into state q_2 on reading symbol 1. Similarly, entry $\mu(+)_{(2,1),2} = 1$ means that there is a transition $+(q_2, q_1) \xrightarrow{1} q_2$ with weight 1 from pair of states (q_2, q_1) into state q_2 on reading symbol $+$.

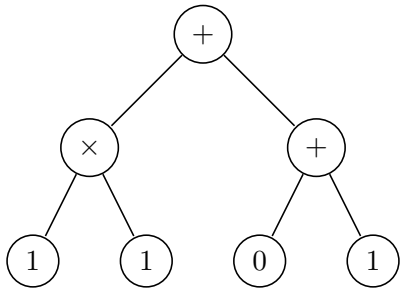
We extend μ from Σ to T_Σ by defining

$$\mu(\sigma(t_1, \ldots, t_k)) := (\mu(t_1) \otimes \cdots \otimes \mu(t_k)) \cdot \mu(\sigma)$$

for every $\sigma \in \Sigma_k$ and $t_1, \ldots, t_k \in T_\Sigma$. The tree series $\|A\| \in \mathbb{F}\langle\langle T_\Sigma \rangle\rangle$ *recognised* by A is defined by $\|A\|(t) = \mu(t) \cdot \gamma$ for every $t \in T_\Sigma$. Note that a 0-dimensional multiplicity tree automaton necessarily recognises a zero tree series. Two automata A_1, A_2 are said to be *equivalent* if $\|A_1\| \equiv \|A_2\|$.

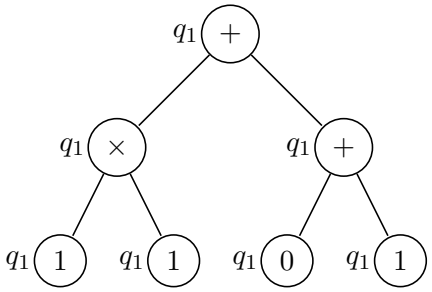
We further extend μ from T_Σ to C_Σ by treating $\Box$ as a unary symbol and defining $\mu(\Box) := I_n$. This allows to define $\mu(c) \in \mathbb{F}^{n\times n}$ for every $c = \sigma(t_1, \ldots, t_k) \in C_\Sigma$ inductively by writing $\mu(c) := (\mu(t_1) \otimes \cdots \otimes \mu(t_k)) \cdot \mu(\sigma)$. It can easily be shown that $\mu(c[t]) = \mu(t) \cdot \mu(c)$ for every $t \in T_\Sigma$ and $c \in C_\Sigma$.

Example 2 *Let us consider the computation of* $\mathbb{F}$-MTA $A = (2, \Sigma, \mu, \gamma)$ *from Example 1 on the following Σ-tree t:*

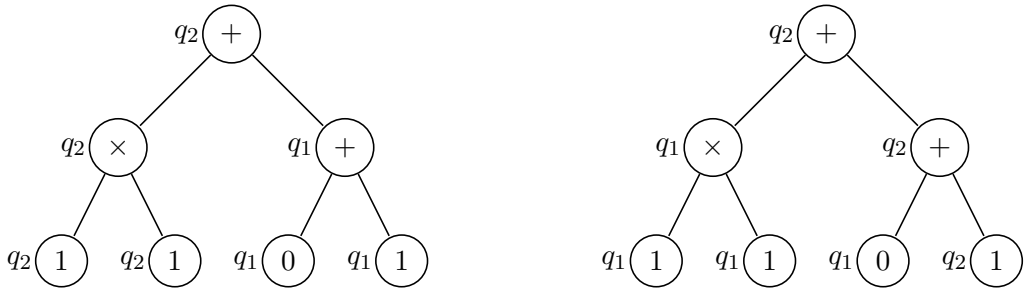


The transition matrices define bottom-up runs of A on t. Intuitively, a run on t corresponds to multiple copies of automaton A walking along t from leaves to the root. Every such run has a weight in $\mathbb{F}$ which is defined as the product of the weights of all transitions taken.

On tree t, automaton A has one nonzero-weight run ending in state q_1, as follows:



Moreover, automaton A has two nonzero-weight runs ending in state q_2, as follows:



Each of the above three runs has weight 1*. Therefore, the total weight of all runs of automaton A on tree t in which the root is labelled q_1 is* 1*, and the total weight of all runs in which the root is labelled q_2 is* 2*. Indeed, algebraically, by definition of μ we have that*

$$
\begin{aligned}
\mu(t) &= (\mu(\times(1,1)) \otimes \mu(+(0,1))) \cdot \mu(+) \\
&= (((\mu(1) \otimes \mu(1)) \cdot \mu(\times)) \otimes ((\mu(0) \otimes \mu(1)) \cdot \mu(+))) \cdot \mu(+) \\
&= \left(\left(\left(\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right) \otimes \left(\left(\begin{bmatrix} 1 & 0 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\
&= \left(\left(\begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \right) \otimes \left(\begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \right) \right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \\
&= \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 1 \end{bmatrix} \right) \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 2 \end{bmatrix}.
\end{aligned}
$$

Finally, the weight $\|A\|(t)$ of tree t is the sum of the weights of all runs on t, where the weight of each run is multiplied by the final weight of its root label. Algebraically, we have

$$\|A\|(t) = \mu(t) \cdot \gamma = \begin{bmatrix} 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \end{bmatrix}^\top = 2.$$

Let $A_1 = (n_1, \Sigma, \mu_1, \gamma_1)$ and $A_2 = (n_2, \Sigma, \mu_2, \gamma_2)$ be two $\mathbb{F}$-multiplicity tree automata. The *product* of A_1 by A_2, written as $A_1 \times A_2$, is the $\mathbb{F}$-multiplicity tree automaton (n, Σ, μ, γ) where:

- $n = n_1 \cdot n_2$;
- If $\sigma \in \Sigma_k$ then $\mu(\sigma) = P_k \cdot (\mu_1(\sigma) \otimes \mu_2(\sigma))$ where P_k is a permutation matrix of order $(n_1 \cdot n_2)^k$ uniquely defined (see Remark 1 below) by

$$(u_1 \otimes \cdots \otimes u_k) \otimes (v_1 \otimes \cdots \otimes v_k) = ((u_1 \otimes v_1) \otimes \cdots \otimes (u_k \otimes v_k)) \cdot P_k \tag{4}$$

  for all $u_1, \ldots, u_k \in \mathbb{F}^{1 \times n_1}$ and $v_1, \ldots, v_k \in \mathbb{F}^{1 \times n_2}$;

- $\gamma = \gamma_1 \otimes \gamma_2$.

Remark 1 *We argue that for every $k \in \mathbb{N}_0$ such that k is the rank of a symbol in Σ, matrix P_k is well-defined by Equation (4). In order to do this, it suffices to show that P_k is well-defined on a set of basis vectors of $\mathbb{F}^{1\times n_1}$ and $\mathbb{F}^{1\times n_2}$ and then extend linearly. To that end, let $(e^1_i)_{i\in[n_1]}$ and $(e^2_j)_{j\in[n_2]}$ be bases of $\mathbb{F}^{1\times n_1}$ and $\mathbb{F}^{1\times n_2}$, respectively. Let us define sets of vectors*

$$E_1 := \{(e^1_{i_1} \otimes \cdots \otimes e^1_{i_k}) \otimes (e^2_{j_1} \otimes \cdots \otimes e^2_{j_k}) : i_1, \ldots, i_k \in [n_1], j_1, \ldots, j_k \in [n_2]\}$$

and

$$E_2 := \{(e^1_{i_1} \otimes e^2_{j_1}) \otimes \cdots \otimes (e^1_{i_k} \otimes e^2_{j_k}) : i_1, \ldots, i_k \in [n_1], j_1, \ldots, j_k \in [n_2]\}.$$

Then, E_1 and E_2 are two bases of the vector space $\mathbb{F}^{1\times n_1 n_2}$. Therefore, P_k is well-defined as an invertible matrix mapping basis E_1 to basis E_2.

Essentially the same product construction as in the proof of the first part of the following proposition is given by Berstel and Reutenauer (1982, Proposition 5.1) using the terminology of linear representations of tree series rather than multiplicity tree automata.

Proposition 2 *Let A_1 and A_2 be $\mathbb{F}$-multiplicity tree automata over a ranked alphabet Σ. Then, for every $t \in T_\Sigma$ it holds that $\|A_1 \times A_2\|(t) = \|A_1\|(t) \cdot \|A_2\|(t)$. Furthermore, in case $\mathbb{F} = \mathbb{Q}$, automaton $A_1 \times A_2$ can be computed from A_1 and A_2 in logarithmic space.*

Proof Let $A_1 = (n_1, \Sigma, \mu_1, \gamma_1)$, $A_2 = (n_2, \Sigma, \mu_2, \gamma_2)$, and $A_1 \times A_2 = (n, \Sigma, \mu, \gamma)$. First we show that for any $t \in T_\Sigma$,

$$\mu(t) = \mu_1(t) \otimes \mu_2(t). \tag{5}$$

We prove that Equation (5) holds for all $t \in T_\Sigma$ using induction on $\mathit{height}(t)$. The base case $t = \sigma \in \Sigma_0$ holds immediately by definition since $P_0 = I_1$. For the induction step, let $h \in \mathbb{N}_0$ and assume that Equation (5) holds for every $t \in T_\Sigma^{\leq h}$. Take any $t \in T_\Sigma^{h+1}$. Then $t = \sigma(t_1, \ldots, t_k)$ for some $k \geq 1$, $\sigma \in \Sigma_k$, and $t_1, \ldots, t_k \in T_\Sigma^{\leq h}$. By induction hypothesis, Equation (4), and the mixed-product property of Kronecker product we now have

$$\begin{aligned}
\mu(t) &= (\mu(t_1) \otimes \cdots \otimes \mu(t_k)) \cdot \mu(\sigma) \\
&= ((\mu_1(t_1) \otimes \mu_2(t_1)) \otimes \cdots \otimes (\mu_1(t_k) \otimes \mu_2(t_k))) \cdot P_k \cdot (\mu_1(\sigma) \otimes \mu_2(\sigma)) \\
&= ((\mu_1(t_1) \otimes \cdots \otimes \mu_1(t_k)) \otimes (\mu_2(t_1) \otimes \cdots \otimes \mu_2(t_k))) \cdot (\mu_1(\sigma) \otimes \mu_2(\sigma)) \\
&= ((\mu_1(t_1) \otimes \cdots \otimes \mu_1(t_k)) \cdot \mu_1(\sigma)) \otimes ((\mu_2(t_1) \otimes \cdots \otimes \mu_2(t_k)) \cdot \mu_2(\sigma)) \\
&= \mu_1(t) \otimes \mu_2(t).
\end{aligned}$$

This completes the proof of Equation (5) by induction. For every $t \in T_\Sigma$, we now have

$$\begin{aligned}
\|A_1 \times A_2\|(t) &= \mu(t) \cdot \gamma = (\mu_1(t) \otimes \mu_2(t)) \cdot (\gamma_1 \otimes \gamma_2) \\
&= (\mu_1(t) \cdot \gamma_1) \otimes (\mu_2(t) \cdot \gamma_2) = \|A_1\|(t) \otimes \|A_2\|(t) = \|A_1\|(t) \cdot \|A_2\|(t).
\end{aligned}$$

Automaton $A_1 \times A_2$ can be computed by a Turing machine which scans the transition matrices and the final weight vectors of A_1 and A_2, and then writes down the entries of the transition matrices and the final weight vector of their product $A_1 \times A_2$ onto the output tape. This computation requires maintaining only a constant number of counters to store the indices of transition matrices, which takes logarithmic space in the representation of automata A_1 and A_2. Hence, the Turing machine computing $A_1 \times A_2$ uses logarithmic space in the work tape. ■

A tree series f is called *recognisable* if it is recognised by some multiplicity tree automaton; such an automaton is called an *MTA-representation* of f. An MTA-representation of f that has the smallest dimension is called *minimal*. The set of all recognisable tree series in $\mathbb{F}\langle\langle T_\Sigma \rangle\rangle$ is denoted by $\mathrm{Rec}(\Sigma, \mathbb{F})$.

The following result was first shown by Bozapalidis and Louscou-Bozapalidou (1983); an essentially equivalent result was later shown by Habrard and Oncina (2006).

Theorem 3 (Bozapalidis and Louscou-Bozapalidou, 1983) *Let Σ be a ranked alphabet and $\mathbb{F}$ be a field. Let $f \in \mathbb{F}\langle\langle T_\Sigma \rangle\rangle$ and let H be the Hankel matrix of f. It holds that $f \in Rec(\Sigma, \mathbb{F})$ if and only if H has finite rank over $\mathbb{F}$. In case $f \in Rec(\Sigma, \mathbb{F})$, the dimension of a minimal MTA-representation of f equals the rank of H.*

The following result by Bozapalidis and Alexandrakis (1989, Proposition 4) states that for any recognisable tree series, its minimal MTA-representation is unique up to change of basis.

Theorem 4 (Bozapalidis and Alexandrakis, 1989) *Let Σ be a ranked alphabet and $\mathbb{F}$ be a field. Let $f \in Rec(\Sigma, \mathbb{F})$ and let r be the rank (over $\mathbb{F}$) of the Hankel matrix of f. Let $A_1 = (r, \Sigma, \mu_1, \gamma_1)$ be an MTA-representation of f. Given an $\mathbb{F}$-multiplicity tree automaton $A_2 = (r, \Sigma, \mu_2, \gamma_2)$, it holds that A_2 recognises f if and only if there exists an invertible matrix $U \in \mathbb{F}^{r \times r}$ such that $\gamma_2 = U \cdot \gamma_1$ and $\mu_2(\sigma) = U^{\otimes rk(\sigma)} \cdot \mu_1(\sigma) \cdot U^{-1}$ for every $\sigma \in \Sigma$.*

2.4 DAG Representations of Finite Trees

A *directed multigraph* consists of a set of nodes V and a multiset of directed edges $E \subseteq V \times V$. We say that a directed multigraph is *acyclic* if the underlying directed graph has no cycles; we say it is *ordered* if a linear order on the successors of each node is assumed. A directed multigraph is *rooted* if there is a distinguished *root* node v such that all other nodes are reachable from v.

Let Σ be a ranked alphabet. A *DAG representation of a Σ-tree* (Σ-*DAG* or *DAG* for short) is a rooted acyclic ordered directed multigraph whose nodes are labelled with symbols from Σ such that the outdegree of each node is equal to the rank of the symbol it is labelled with. Formally a Σ-DAG consists of a set of nodes V, for each node $v \in V$ a list of successors $succ(v) \in V^*$, and a node labelling $\lambda : V \to \Sigma$ where for each node $v \in V$ it holds that $\lambda(v) \in \Sigma_{|succ(v)|}$. Note that Σ-trees are a subclass of Σ-DAGs.

Let G be a Σ-DAG. The *size* of G, denoted by $size(G)$, is the number of nodes in G. The *height* of G, denoted by $height(G)$, is the length of a longest directed path in G. For any node v in G, the *sub-DAG* of G *rooted at* v, denoted by $G|_v$, is the Σ-DAG consisting

of the node v and all of its descendants in G. Clearly, if v is the root of G then $G|_v = G$. The set $\{G|_v : v \text{ is a node in } G\}$ of all the sub-DAGs of G is denoted by $Sub(G)$.

For any Σ-DAG G, we define its *unfolding* into a Σ-tree, denoted by $unfold(G)$, inductively as follows: If the root of G is labelled with a symbol σ and has the list of successors $v_1, \ldots, v_k$, then

$$unfold(G) = \sigma(unfold(G|_{v_1}), \ldots, unfold(G|_{v_k})).$$

The next proposition follows easily from the definition.

Proposition 5 *If G is a Σ-DAG, then $Sub(unfold(G)) = unfold[Sub(G)]$.*

Because a context has exactly one occurrence of symbol $\Box$, any *DAG representation of a Σ-context* is a $(\{\Box\} \cup \Sigma)$-DAG that has a unique path from the root to the (unique) $\Box$-labelled node. The *concatenation* of a DAG K, representing a Σ-context, and a Σ-DAG G is the Σ-DAG, denoted by $K[G]$, obtained by substituting the root of G for $\Box$ in K.

Proposition 6 *Let K be a DAG representation of a Σ-context, and let G be a Σ-DAG. Then, $unfold(K[G]) = unfold(K)[unfold(G)]$.*

Proof The proof is by induction on $height(K)$. For the base case, let $height(K) = 0$. Then, we have that $K = \Box$ and therefore $unfold(\Box[G]) = unfold(G) = unfold(\Box)[unfold(G)]$ for any Σ-DAG G.

For the induction step, let $h \in \mathbb{N}_0$ and assume that the result holds if $height(K) \leq h$. Let K be a DAG representation of a Σ-context such that $height(K) = h + 1$. Let the root of K have label σ and list of successors $v_1, \ldots, v_k$. By definition, there is a unique path in K going from the root to the $\Box$-labelled node. Without loss of generality, we can assume that the $\Box$-labelled node is a successor of v_1. Take an arbitrary Σ-DAG G. Since $height(K|_{v_1}) \leq h$, we have by the induction hypothesis that

$$\begin{aligned} unfold(K[G]) &= \sigma(unfold(K|_{v_1}[G]), unfold(K|_{v_2}), \ldots, unfold(K|_{v_k})) \\ &= \sigma(unfold(K|_{v_1})[unfold(G)], unfold(K|_{v_2}), \ldots, unfold(K|_{v_k})) \\ &= \sigma(unfold(K|_{v_1}), unfold(K|_{v_2}), \ldots, unfold(K|_{v_k}))[unfold(G)] \\ &= unfold(K)[unfold(G)]. \end{aligned}$$

This completes the proof by induction. ∎

2.5 Multiplicity Tree Automata on DAGs

In this section, we introduce the notion of a multiplicity tree automaton running on DAGs. To the best of our knowledge, this notion has not been studied before.

Let $\mathbb{F}$ be a field, and $A = (n, \Sigma, \mu, \gamma)$ be an $\mathbb{F}$-multiplicity tree automaton. The computation of automaton A on a Σ-DAG $G = (V, E)$ is defined as follows: A *run* of A on G is a mapping $\rho : Sub(G) \to \mathbb{F}^n$ such that for every node $v \in V$, if v is labelled with σ and has the list of successors $succ(v) = v_1, \ldots, v_k$ then

$$\rho(G|_v) = (\rho(G|_{v_1}) \otimes \cdots \otimes \rho(G|_{v_k})) \cdot \mu(\sigma).$$

Automaton A assigns to G a *weight* $\|A\|(G) \in \mathbb{F}$ where $\|A\|(G) = \rho(G) \cdot \gamma$.

In the following proposition, we show that the weight assigned by a multiplicity tree automaton to a DAG is equal to the weight assigned to its tree unfolding.

Proposition 7 *Let $\mathbb{F}$ be a field, and $A = (n, \Sigma, \mu, \gamma)$ be an $\mathbb{F}$-multiplicity tree automaton. For any Σ-DAG G, it holds that $\rho(G) = \mu(\mathit{unfold}(G))$ and $\|A\|(G) = \|A\|(\mathit{unfold}(G))$.*

Proof Let V be the set of nodes of G. First we show that for every $v \in V$,

$$\rho(G|_v) = \mu(\mathit{unfold}(G|_v)). \tag{6}$$

The proof is by induction on $\mathit{height}(G|_v)$. For the base case, let $\mathit{height}(G|_v) = 0$. This implies that $G|_v = \sigma \in \Sigma_0$. Therefore, by definition we have that

$$\rho(G|_v) = \mu(\sigma) = \mu(\mathit{unfold}(\sigma)) = \mu(\mathit{unfold}(G|_v)).$$

For the induction step, let $h \in \mathbb{N}_0$ and assume that Equation (6) holds for every $v \in V$ such that $\mathit{height}(G|_v) \leq h$. Take any $v \in V$ such that $\mathit{height}(G|_v) = h + 1$. Let the root of $G|_v$ be labelled with a symbol σ and have list of successors $\mathit{succ}(v) = v_1, \ldots, v_k$. Then for every $j \in [k]$, we have that $\mathit{height}(G|_{v_j}) \leq h$ and thus $\rho(G|_{v_j}) = \mu(\mathit{unfold}(G|_{v_j}))$ holds by the induction hypothesis. This implies that

$$\begin{aligned}
\rho(G|_v) &= (\rho(G|_{v_1}) \otimes \cdots \otimes \rho(G|_{v_k})) \cdot \mu(\sigma) \\
&= (\mu(\mathit{unfold}(G|_{v_1})) \otimes \cdots \otimes \mu(\mathit{unfold}(G|_{v_k}))) \cdot \mu(\sigma) \\
&= \mu(\sigma(\mathit{unfold}(G|_{v_1}), \ldots, \mathit{unfold}(G|_{v_k}))) \\
&= \mu(\mathit{unfold}(G|_v)),
\end{aligned}$$

which completes the proof of Equation (6) for all $v \in V$ by induction.

Taking v to be the root of G, we get from Equation (6) that $\rho(G) = \mu(\mathit{unfold}(G))$. Therefore, $\|A\|(G) = \rho(G) \cdot \gamma = \mu(\mathit{unfold}(G)) \cdot \gamma = \|A\|(\mathit{unfold}(G))$. ■

Example 3 *Let $\Sigma = \{\sigma_0, \sigma_2\}$ be a ranked alphabet such that $rk(\sigma_0) = 0$ and $rk(\sigma_2) = 2$. Take any $n \in \mathbb{N}$. Let t_n, depicted in Figure 1, be the perfect binary Σ-tree of height $n - 1$. Note that $\mathit{size}(t_n) = O(2^n)$. Define an $\mathbb{F}$-MTA $A = (n, \Sigma, \mu, e_1)$ such that $\mu(\sigma_0) = e_n \in \mathbb{F}^{1 \times n}$ and $\mu(\sigma_2) \in \mathbb{F}^{n^2 \times n}$ where $\mu(\sigma_2)_{(i+1,i+1),i} = 1$ for every $i \in [n-1]$, and all other entries of $\mu(\sigma_2)$ are zero. It is easy to see that $\|A\|(t_n) = 1$ and $\|A\|(t) = 0$ for every $t \in T_\Sigma \setminus \{t_n\}$.*

Let B be the 0-dimensional $\mathbb{F}$-MTA over Σ (so that $\|B\| \equiv 0$). Suppose we were to check whether automata A and B are equivalent. Then the only counterexample to their equivalence, namely the tree t_n, has size $O(2^n)$. Note, however, that t_n has an exponentially more succinct DAG representation G_n, given in Figure 2.

2.6 Arithmetic Circuits

An *arithmetic circuit* is a finite acyclic vertex-labelled directed multigraph whose vertices, called *gates*, have indegree 0 or 2. Vertices of indegree 0 are called *input gates* and are labelled with a constant 0 or 1, or a variable from the set $\{x_i : i \in \mathbb{N}\}$. Vertices of indegree

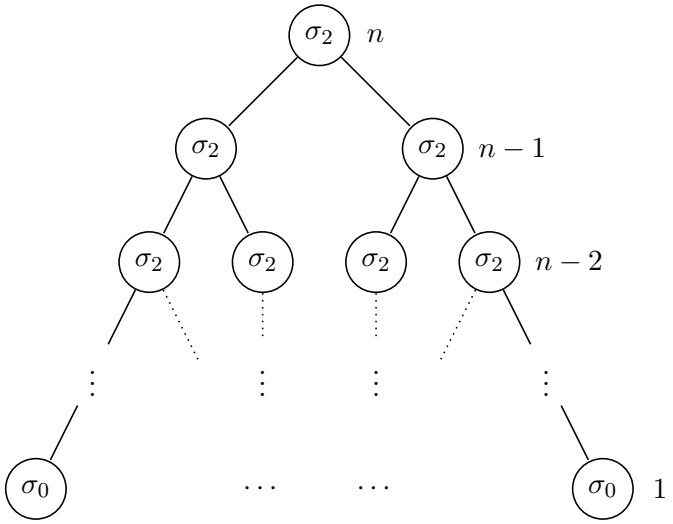


Figure 1: Tree t_n

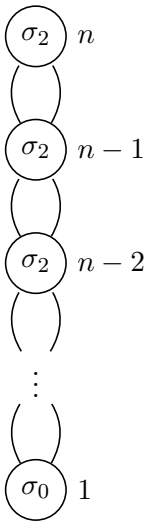


Figure 2: DAG G_n

2 are called *internal gates* and are labelled with an arithmetic operation $+$, $\times$, or $-$. We assume that there is a unique gate with outdegree 0 called the *output gate*. An arithmetic circuit is called *variable-free* if all input gates are labelled with 0 or 1.

Given two gates u and v of an arithmetic circuit C, we call u a *child* of v if (u, v) is a directed edge in C. The *size* of C is the number of gates in C. The *height* of a gate v in C, written as $height(v)$, is the length of a longest directed path from an input gate to v. The *height* of C is the maximal height of a gate in C.

An arithmetic circuit C computes a polynomial over the integers as follows: An input gate of C labelled with $\alpha \in \{0, 1\} \cup \{x_i : i \in \mathbb{N}\}$ computes the polynomial α. An internal gate of C labelled with $* \in \{+, \times, -\}$ computes the polynomial $p_1 * p_2$ where p_1 and p_2 are the polynomials computed by its children. For any gate v in C, we write f_v for the polynomial computed by v. The *output* of C, written as f_C, is the polynomial computed by the output gate of C. The *arithmetic circuit identity testing (**ACIT**)* problem asks whether the output of a given arithmetic circuit is equal to the zero polynomial.

Remark 8 *Any variable-free arithmetic circuit C can be seen as a Σ-DAG with the ranked alphabet $\Sigma = \{0, 1, +, \times, -\}$ where $0, 1$ are nullary symbols and $+, \times, -$ are binary symbols. Let $A = (2, \Sigma, \mu, \gamma)$ be the multiplicity tree automaton from Example 1. Then, for any gate v in C it holds that $\mu(C|_v) = \begin{bmatrix} 1 & f_v \end{bmatrix}$, where $C|_v$ is the sub-DAG of C rooted at v and f_v is the number computed at gate v. (This result can be easily proved using induction on $height(C|_v)$.) In particular, when v is the output gate of C we get that*

$$\|A\|(C) = \mu(C) \cdot \gamma = \begin{bmatrix} 1 & f_C \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \end{bmatrix}^\top = f_C.$$

Hence, automaton A evaluates the circuit C.

2.7 The Learning Model

In this paper we work with the *exact learning model* of Angluin (1988): Let f be a *target function*. A *Learner* (*learning algorithm*) may, in each step, propose a hypothesis function h by making an *equivalence query* to a *Teacher*. If h is equivalent to f, then the Teacher returns YES and the Learner succeeds and halts. Otherwise, the Teacher returns NO with a *counterexample*, which is an assignment x such that $h(x) \neq f(x)$. Moreover, the Learner may query the Teacher for the value of the function f on a particular assignment x by making a *membership query* on x. The Teacher returns the value $f(x)$ to such a query.

We say that a class of functions $\mathscr{C}$ is *exactly learnable* if there is a Learner that for any target function $f \in \mathscr{C}$, outputs a hypothesis $h \in \mathscr{C}$ such that $h(x) = f(x)$ for all assignments x, and does so in time polynomial in the size of a shortest representation of f and the size of a largest counterexample returned by the Teacher. We moreover say that the class $\mathscr{C}$ is *exactly learnable in (randomised) polynomial time* if the learning algorithm can be implemented to run in (randomised) polynomial time in the Turing model.

3. Equivalence Queries

In the exact learning model, one of the principal algorithmic questions from the point of view of the Teacher is the computational complexity of equivalence testing. In this section we characterise the computational complexity of equivalence testing for multiplicity tree automata, showing that this problem is logspace equivalent to polynomial identity testing. The latter is a well-studied problem for which there are numerous randomised polynomial-time algorithms, with the existence of a deterministic polynomial-time algorithm being a longstanding open problem. Moreover in this section, we explain why it is natural to expect the Teacher to return succinct DAG counterexamples in the case of inequivalence.

3.1 Computational Complexity of MTA Equivalence

A key algorithmic component of the exact learning framework is checking the equivalence of the hypothesis and the target function: a task for the Teacher rather than the Learner. The existence of efficient algorithms to perform such equivalence checks is important for several applications of the exact learning framework (see, e.g., Feng et al., 2011). With this in mind, in this subsection we characterise the computational complexity of the equivalence problem for $\mathbb{Q}$-multiplicity tree automata. Here we specialise the weight field to be $\mathbb{Q}$ since we want to work within the classical Turing model of computation. Parts of this section also exploit the fact that $\mathbb{Q}$ is an ordered field.

Our main result is:

Theorem 9 *The equivalence problem for $\mathbb{Q}$-multiplicity tree automata is logspace interreducible with* ***ACIT****.*

A related result, characterising equivalence of probabilistic visibly pushdown automata on words in terms of polynomial identity testing, was shown by Kiefer et al. (2013). On several occasions in this section, we will implicitly make use of the fact that a composition of two logspace reductions is again a logspace reduction (Arora and Barak, 2009, Lemma 4.17).

3.1.1 From MTA Equivalence to **ACIT**

First, we present a logspace reduction from the equivalence problem for $\mathbb{Q}$-MTAs to **ACIT**. We start with the following lemma.

Lemma 10 *Given an integer $n \in \mathbb{N}$ and a $\mathbb{Q}$-multiplicity tree automaton A over a ranked alphabet Σ, one can compute, in logarithmic space in $|A|$ and n, a variable-free arithmetic circuit that has output $\sum_{t \in T_\Sigma^{\leq n}} \|A\|(t)$.*

Proof Let $A = (r, \Sigma, \mu, \gamma)$, and let m be the rank of Σ. By definition, it holds that

$$\sum_{t \in T_\Sigma^{\leq n}} \|A\|(t) = \left(\sum_{t \in T_\Sigma^{\leq n}} \mu(t) \right) \cdot \gamma. \tag{7}$$

We have $\sum_{t \in T_\Sigma^{\leq 1}} \mu(t) = \sum_{\sigma \in \Sigma_0} \mu(\sigma)$. Furthermore for every $i \in \mathbb{N}$, it holds that

$$T_\Sigma^{\leq i+1} = \{\sigma(t_1, \ldots, t_k) : k \in \{0, \ldots, m\}, \sigma \in \Sigma_k, t_1, \ldots, t_k \in T_\Sigma^{\leq i}\}$$

and thus by bilinearity of Kronecker product,

$$\begin{aligned}
\sum_{t \in T_\Sigma^{\leq i+1}} \mu(t) &= \sum_{k=0}^{m} \sum_{\sigma \in \Sigma_k} \sum_{t_1 \in T_\Sigma^{\leq i}} \cdots \sum_{t_k \in T_\Sigma^{\leq i}} (\mu(t_1) \otimes \cdots \otimes \mu(t_k)) \cdot \mu(\sigma) \\
&= \sum_{k=0}^{m} \sum_{\sigma \in \Sigma_k} \left(\left(\sum_{t_1 \in T_\Sigma^{\leq i}} \mu(t_1) \right) \otimes \cdots \otimes \left(\sum_{t_k \in T_\Sigma^{\leq i}} \mu(t_k) \right) \right) \cdot \mu(\sigma) \\
&= \sum_{k=0}^{m} \left(\sum_{t \in T_\Sigma^{\leq i}} \mu(t) \right)^{\otimes k} \sum_{\sigma \in \Sigma_k} \mu(\sigma).
\end{aligned} \tag{8}$$

In the following we define a variable-free arithmetic circuit Φ that has output $\sum_{t \in T_\Sigma^{\leq n}} \|A\|(t)$. First, let us denote $G(i) := \sum_{t \in T_\Sigma^{\leq i}} \mu(t)$ for every $i \in \mathbb{N}$. Then by Equation (8) we have $G(i+1) = \sum_{k=0}^{m} G(i)^{\otimes k} \cdot S(k)$ where $S(k) := \sum_{\sigma \in \Sigma_k} \mu(\sigma)$ for every $k \in \{0, \ldots, m\}$. In coordinate notation, for every $j \in [r]$ we have by Equation (1) that

$$G(i+1)_j = \sum_{k=0}^{m} \sum_{(l_1, \ldots, l_k) \in [r]^k} \prod_{a=1}^{k} G(i)_{l_a} \cdot S(k)_{(l_1, \ldots, l_k), j}. \tag{9}$$

We present Φ as a straight-line program, with built-in constants

$$\{\mu^\sigma_{(l_1, \ldots, l_k), j}, \gamma_j : k \in \{0, \ldots, m\}, \sigma \in \Sigma_k, (l_1, \ldots, l_k) \in [r]^k, j \in [r]\}$$

representing the entries of the transition matrices and the final weight vector of A, internal variables $\{s^k_{(l_1, \ldots, l_k), j} : k \in \{0, \ldots, m\}, (l_1, \ldots, l_k) \in [r]^k, j \in [r]\}$ and $\{g_{i,j} : i \in [n], j \in [r]\}$ evaluating the entries of matrices $S(k)$ and vectors $G(i)$ respectively, and the final internal variable f computing the value of Φ.

1. For $j \in [r]$ do $g_{1,j} \leftarrow \sum_{\sigma \in \Sigma_0} \mu_j^\sigma$
2. For $k \in \{0, \ldots, m\}$, $(l_1, \ldots, l_k) \in [r]^k$, $j \in [r]$ do $s^k_{(l_1,\ldots,l_k),j} \leftarrow \sum_{\sigma \in \Sigma_k} \mu^\sigma_{(l_1,\ldots,l_k),j}$
3. For $i = 1$ to $n - 1$ do
   3.1. For $k \in \{0, \ldots, m\}$, $(l_1, \ldots, l_k) \in [r]^k$, $j \in [r]$ do
   $$h^{i,k}_{(l_1,\ldots,l_k),j} \leftarrow \prod_{a=1}^{k} g_{i,l_a} \cdot s^k_{(l_1,\ldots,l_k),j}$$
   3.2. For $j \in [r]$ do
   $$g_{i+1,j} \leftarrow \sum_{k=0}^{m} \sum_{(l_1,\ldots,l_k) \in [r]^k} h^{i,k}_{(l_1,\ldots,l_k),j}$$
4. For $j \in [r]$ do $f_j \leftarrow g_{n,j} \cdot \gamma_j$
5. $f \leftarrow \sum_{j \in [r]} f_j$.

Table 1: Straight-line program Φ

Formally, the straight-line program Φ is given in Table 1. Here the statements are given in indexed-sum and indexed-product notation, which can easily be expanded in terms of the corresponding binary operations. It follows from Equations (7) and (9) that the output of Φ is $G(n) \cdot \gamma = \sum_{t \in T_\Sigma^{\leq n}} \|A\|(t)$.

The input gates of Φ are labelled with rational numbers. By separately encoding numerators and denominators, we can in logarithmic space reduce Φ to an arithmetic circuit where all input gates are labelled with integers. Moreover, without loss of generality we can assume that every input gate of Φ is labelled with 0 or 1. Any other integer label given in binary can be encoded as an arithmetic circuit.

Recalling that a composition of two logspace reductions is again a logspace reduction, we conclude that the entire computation takes logarithmic space in $|A|$ and n. ■

Before presenting the reduction in Proposition 12, we recall the following characterisation (Seidl, 1990, Theorem 4.2) of equivalence of two multiplicity tree automata over an arbitrary field.

Proposition 11 (Seidl, 1990) *Suppose A and B are multiplicity tree automata of dimension n_1 and n_2, respectively, and over a ranked alphabet Σ. Then, A and B are equivalent if and only if $\|A\|(t) = \|B\|(t)$ for every $t \in T_\Sigma^{<n_1+n_2}$.*

We now turn to the reduction:

Proposition 12 *The equivalence problem for $\mathbb{Q}$-multiplicity tree automata is logspace reducible to **ACIT**.*

Proof Let A and B be $\mathbb{Q}$-multiplicity tree automata over a ranked alphabet Σ, and let n be the sum of their dimensions. Proposition 2 implies that

$$\begin{aligned}\sum_{t\in T_\Sigma^{<n}} (\|A\|(t) - \|B\|(t))^2 &= \sum_{t\in T_\Sigma^{<n}} \left(\|A\|(t)^2 + \|B\|(t)^2 - 2\|A\|(t)\|B\|(t)\right) \\ &= \sum_{t\in T_\Sigma^{<n}} \left(\|A\times A\|(t) + \|B\times B\|(t) - 2\|A\times B\|(t)\right).\end{aligned}$$

Thus by Proposition 11, automata A and B are equivalent if and only if

$$\sum_{t\in T_\Sigma^{<n}} \|A\times A\|(t) + \sum_{t\in T_\Sigma^{<n}} \|B\times B\|(t) - 2\sum_{t\in T_\Sigma^{<n}} \|A\times B\|(t) = 0. \tag{10}$$

We know from Proposition 2 that automata $A\times A$, $B\times B$, and $A\times B$ can be computed in logarithmic space. Thus by Lemma 10 one can compute, in logarithmic space in $|A|$ and $|B|$, variable-free arithmetic circuits that have outputs $\sum_{t\in T_\Sigma^{<n}} \|A\times A\|(t)$, $\sum_{t\in T_\Sigma^{<n}} \|B\times B\|(t)$, and $\sum_{t\in T_\Sigma^{<n}} \|A\times B\|(t)$ respectively. Using Equation (10), we can now easily construct a variable-free arithmetic circuit that has output 0 if and only if A and B are equivalent. ■

3.1.2 From **ACIT** to MTA Equivalence

We now present a converse reduction: from **ACIT** to the equivalence problem for $\mathbb{Q}$-MTAs.

Allender et al. (2009, Proposition 2.2) give a logspace reduction of the general **ACIT** problem to the special case of **ACIT** for variable-free circuits. The latter can, by representing arbitrary integers as differences of two nonnegative integers, be reformulated as the problem of deciding whether two variable-free arithmetic circuits with only $+$ and $\times$-internal gates compute the same number. With this result at hand, we turn to the reduction:

Proposition 13 ***ACIT** is logspace reducible to the equivalence problem for $\mathbb{Q}$-multiplicity tree automata.*

Proof Let C_1 and C_2 be two variable-free arithmetic circuits whose internal gates are labelled with $+$ or $\times$. By padding with extra gates, without loss of generality we can assume that in each circuit the children of a height-i gate both have height $i-1$, $+$-gates have even height, $\times$-gates have odd height, and the output gate has an even height h.

In the following we define two $\mathbb{Q}$-MTAs, A_1 and A_2, that are equivalent if and only if circuits C_1 and C_2 have the same output. Automata A_1 and A_2 are both defined over a

ranked alphabet $\Sigma = \{\sigma_0, \sigma_1, \sigma_2\}$ where σ_0 is a nullary, σ_1 is a unary, and σ_2 is a binary symbol. Intuitively, automata A_1 and A_2 both recognise the common 'tree unfolding' of circuits C_1 and C_2.

We now derive A_1 from C_1; A_2 is analogously derived from C_2. Let $\{v_1, \ldots, v_r\}$ be the set of gates of C_1 where v_r is the output gate. Automaton A_1 has a state q_i for every gate v_i of C_1. Formally, $A_1 = (r, \Sigma, \mu, e_r^\top)$ where for every $i \in [r]$:

- If v_i is an input gate with label 1 then $\mu(\sigma_0)_i = 1$, otherwise $\mu(\sigma_0)_i = 0$.
- If v_i is a $+$-gate with children v_{j_1} and v_{j_2} then $\mu(\sigma_1)_{j_1,i} = \mu(\sigma_1)_{j_2,i} = 1$ if $j_1 \neq j_2$, $\mu(\sigma_1)_{j_1,i} = 2$ if $j_1 = j_2$, and $\mu(\sigma_1)_{l,i} = 0$ for every $l \notin \{j_1, j_2\}$. If v_i is an input gate or a $\times$-gate then $\mu(\sigma_1)^i = 0_{r\times 1}$.
- If v_i is a $\times$-gate with children v_{j_1} and v_{j_2} then $\mu(\sigma_2)_{(j_1,j_2),i} = 1$, and $\mu(\sigma_2)_{(l_1,l_2),i} = 0$ for every $(l_1, l_2) \neq (j_1, j_2)$. If v_i is an input gate or a $+$-gate then $\mu(\sigma_2)^i = 0_{r^2\times 1}$.

We define a sequence of trees $(t_n)_{n\in\mathbb{N}_0} \subseteq T_\Sigma$ by $t_0 = \sigma_0$, $t_{n+1} = \sigma_1(t_n)$ for n odd, and $t_{n+1} = \sigma_2(t_n, t_n)$ for n even. In the following we show that $\|A_1\|(t_h) = f_{C_1}$. For every gate v of C_1, by assumption it holds that all paths from v to the output gate have equal length. We now prove that for every $i \in [r]$,

$$\mu(t_{h_i})_i = f_{v_i} \tag{11}$$

where $h_i := \mathit{height}(v_i)$. The proof uses induction on $h_i \in \{0, \ldots, h\}$. For the base case, let $h_i = 0$. Then, v_i is an input gate and thus by definition of automaton A_1 we have

$$\mu(t_{h_i})_i = \mu(t_0)_i = \mu(\sigma_0)_i = f_{v_i}.$$

For the induction step, let $n \in [h]$ and assume that Equation (11) holds for every gate v_i of height less than n. Take an arbitrary gate v_i of C_1 such that $h_i = n$. Let gates v_{j_1} and v_{j_2} be the children of v_i. Then $h_{j_1} = h_{j_2} = h_i - 1 = n - 1$ by assumption. The induction hypothesis now implies that $\mu(t_{h_i-1})_{j_1} = f_{v_{j_1}}$ and $\mu(t_{h_i-1})_{j_2} = f_{v_{j_2}}$. Depending on the label of v_i, there are two possible cases as follows:

(i) If v_i is a $+$-gate, then h_i is even and thus by definition of A_1 we have

$$\begin{aligned}\mu(t_{h_i})_i &= \mu(\sigma_1(t_{h_i-1}))_i = \mu(t_{h_i-1}) \cdot \mu(\sigma_1)^i \\ &= \mu(t_{h_i-1})_{j_1} + \mu(t_{h_i-1})_{j_2} = f_{v_{j_1}} + f_{v_{j_2}} = f_{v_i}.\end{aligned}$$

(ii) If v_i is a $\times$-gate, then h_i is odd and thus by definition of A_1 and Equation (1) we have

$$\begin{aligned}\mu(t_{h_i})_i &= \mu(\sigma_2(t_{h_i-1}, t_{h_i-1}))_i = \mu(t_{h_i-1})^{\otimes 2} \cdot \mu(\sigma_2)^i \\ &= \mu(t_{h_i-1})_{j_1} \cdot \mu(t_{h_i-1})_{j_2} = f_{v_{j_1}} \cdot f_{v_{j_2}} = f_{v_i}.\end{aligned}$$

This completes the proof of Equation (11) by induction. Now for the output gate v_r of C_1, we get from Equation (11) that $\mu(t_h)_r = f_{v_r}$ since $h_r = h$. Therefore,

$$\|A_1\|(t_h) = \mu(t_h) \cdot e_r^\top = \mu(t_h)_r = f_{v_r} = f_{C_1}.$$

Analogously, it holds that $\|A_2\|(t_h) = f_{C_2}$. It is moreover clear by construction that $\|A_1\|(t) = 0$ and $\|A_2\|(t) = 0$ for every $t \in T_\Sigma \setminus \{t_h\}$. Therefore, automata A_1 and A_2 are equivalent if and only if arithmetic circuits C_1 and C_2 have the same output. ■

Propositions 12 and 13 together imply Theorem 9. On a positive note, it should be remarked that there are numerous efficient randomised algorithms for **ACIT**. Indeed, it was already known that there is a randomised polynomial-time algorithm for equivalence of multiplicity tree automata (Seidl, 1990). On the other hand, we have shown that obtaining a deterministic polynomial-time algorithm for multiplicity tree automaton equivalence would imply also a deterministic polynomial-time algorithm for **ACIT**.

3.2 DAG Counterexamples

In the exact learning model, when answering an equivalence query the Teacher not only checks equivalence but also provides a counterexample in case of inequivalence. As mentioned before, there is a randomised polynomial-time algorithm for checking MTA equivalence (Seidl, 1990). In this subsection, we explain why a Teacher using this algorithm would naturally give succinct DAG counterexamples.

Although the paper of Seidl (1990) does not mention counterexamples, they can be easily extracted from the algorithm presented therein. Indeed the correctness proof of the algorithm shows, *inter alia*, that for any two inequivalent MTAs $A_1 = (n_1, \Sigma, \mu_1, \gamma_1)$ and $A_2 = (n_2, \Sigma, \mu_2, \gamma_2)$, there exists a tree t such that $\|A_1\|(t) \neq \|A_2\|(t)$ and t can be represented by a DAG with at most $n_1 + n_2$ vertices. To see this, we now briefly describe the main idea behind the procedure: Given MTAs A_1 and A_2 as above, a prefix-closed set of trees $S \subseteq T_\Sigma$ is maintained such that $\{\begin{bmatrix}\mu_1(t) & \mu_2(t)\end{bmatrix} : t \in S\}$ is a linearly independent set of vectors. Note that since this set of vectors lies in $\mathbb{F}^{n_1+n_2}$, it necessarily holds that $|S| \leq n_1 + n_2$. The algorithm terminates when

$$\mathit{span}\left\{\begin{bmatrix}\mu_1(t) & \mu_2(t)\end{bmatrix} : t \in S\right\} = \mathit{span}\left\{\begin{bmatrix}\mu_1(t) & \mu_2(t)\end{bmatrix} : t \in T_\Sigma\right\}$$

and reports that A_1 and A_2 are inequivalent just in case a tree $t \in S$ is found such that

$$\begin{bmatrix}\mu_1(t) & \mu_2(t)\end{bmatrix} \cdot \begin{bmatrix}\gamma_1 \\ -\gamma_2\end{bmatrix} \neq 0,$$

i.e., $\|A_1\|(t) \neq \|A_2\|(t)$. Such a tree t, if one exists, has at most $n_1 + n_2$ subtrees and thus has a DAG representation of size at most $n_1 + n_2$. As we have seen in Example 3, the number of vertices of tree t may be exponential in $n_1 + n_2$, thus it is very natural that a Teacher that resolves equivalence queries using the algorithm of Seidl (1990) would return counterexamples represented succinctly as DAGs.

4. The Learning Algorithm

In this section, we give an exact learning algorithm for multiplicity tree automata. Our algorithm is polynomial in the size of a minimal automaton equivalent to the target and the size of a largest counterexample given as a DAG. As seen in Example 3, DAG counterexamples can be exponentially more succinct than tree counterexamples. Therefore, achieving a polynomial bound in the context of DAG representations is a more exacting criterion.

Over an arbitrary field $\mathbb{F}$, the algorithm can be seen as running on a Blum-Shub-Smale machine that can write and read field elements to and from its memory at unit cost and that can also perform arithmetic operations and equality tests on field elements at unit cost (see Arora and Barak, 2009). Over $\mathbb{Q}$, the algorithm can be implemented in randomised polynomial time by representing rationals as arithmetic circuits and using a coRP algorithm for equality testing of such circuits (see Allender et al., 2009).

This section is organised as follows: In Section 4.1 we present the learning algorithm. In Section 4.2 we prove correctness on trees, and then argue in Section 4.3 that the algorithm can be faithfully implemented using a DAG representation of trees. Finally, in Section 4.4 we give a complexity analysis of the algorithm assuming the DAG representation.

4.1 The Algorithm

Let $f \in \mathrm{Rec}(\Sigma, \mathbb{F})$ be the target function. The algorithm learns an MTA-representation of f using its Hankel matrix H, which has finite rank over $\mathbb{F}$ by Theorem 3.

The algorithm iteratively constructs a full row-rank submatrix of the Hankel matrix H. At each stage, the algorithm maintains the following data:

- An integer $n \in \mathbb{N}$.
- A set of n 'rows' $X = \{t_1, \ldots, t_n\} \subseteq T_\Sigma$.
- A finite set of 'columns' $Y \subseteq C_\Sigma$ such that $\Box \in Y$.
- A submatrix $H_{X,Y}$ of H that has full row rank.

These data determine a *hypothesis automaton* A of dimension n, whose states correspond to the rows of $H_{X,Y}$, with the i^{th} row corresponding to the state reached after reading tree t_i. The Learner makes an equivalence query on the hypothesis A. In case the Teacher answers NO, the Learner receives a counterexample z. The Learner then parses z bottom-up to find a minimal subtree of z that is also a counterexample, and uses this subtree to augment the row set X and the column set Y in a way that increases the rank of the submatrix $H_{X,Y}$.

Formally, the algorithm LMTA is given in Table 2. Here for any k-ary symbol $\sigma \in \Sigma$ we define $\sigma(X, \ldots, X) := \{\sigma(t_{i_1}, \ldots, t_{i_k}) : (i_1, \ldots, i_k) \in [n]^k\}$.

Algorithm LMTA follows a classical scheme: it generalises the procedure of Beimel et al. (2000) by working with a more general notion of a Hankel matrix that is appropriate for tree series. Moreover, LMTA differs from the procedure of Habrard and Oncina (2006) in the way counterexamples are treated and the hypothesis automaton updated; we provide more details on this point at the end of this section.

4.2 Correctness Proof

In this subsection, we prove the correctness of the exact learning algorithm LMTA. Specifically, we show that, given a target $f \in \mathrm{Rec}(\Sigma, \mathbb{F})$, algorithm LMTA outputs a minimal MTA-representation of f after at most $\mathit{rank}(H)$ iterations of the main loop.

The correctness proof naturally breaks down into several lemmas. First, we show that matrix $H_{X,Y}$ has full row rank.

Algorithm LMTA

Target: $f \in \mathrm{Rec}(\Sigma, \mathbb{F})$, where Σ has rank m and $\mathbb{F}$ is a field

1. Make an equivalence query on the 0-dimensional $\mathbb{F}$-MTA over Σ.
   If the answer is YES then **output** the 0-dimensional $\mathbb{F}$-MTA over Σ and halt.
   Otherwise the answer is NO and z is a counterexample. Initialise:
   $n \leftarrow 1$, $t_n \leftarrow z$, $X \leftarrow \{t_n\}$, $Y \leftarrow \{\Box\}$.

2. 2.1. For every $k \in \{0, \ldots, m\}$, $\sigma \in \Sigma_k$, and $(i_1, \ldots, i_k) \in [n]^k$:
   If $H_{\sigma(t_{i_1}, \ldots, t_{i_k}),Y}$ is not a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$ then
   $n \leftarrow n+1$, $t_n \leftarrow \sigma(t_{i_1}, \ldots, t_{i_k})$, $X \leftarrow X \cup \{t_n\}$.

   2.2. Define an $\mathbb{F}$-MTA $A = (n, \Sigma, \mu, \gamma)$ as follows:

   - $\gamma = H_{X,\Box}$.
   - For every $k \in \{0, \ldots, m\}$ and $\sigma \in \Sigma_k$:
     Define matrix $\mu(\sigma) \in \mathbb{F}^{n^k \times n}$ by the equation

$$\mu(\sigma) \cdot H_{X,Y} = H_{\sigma(X,\ldots,X),Y}. \tag{12}$$

3. 3.1. Make an equivalence query on A.
   If the answer is YES then **output** A and halt.
   Otherwise the answer is NO and z is a counterexample. Searching bottom-up, find a subtree $\sigma(\tau_1, \ldots, \tau_k)$ of z that satisfies the following two conditions:

   (i) For every $j \in [k]$, $H_{\tau_j,Y} = \mu(\tau_j) \cdot H_{X,Y}$.
   (ii) For some $c \in Y$, $H_{\sigma(\tau_1,\ldots,\tau_k),c} \neq \mu(\sigma(\tau_1, \ldots, \tau_k)) \cdot H_{X,c}$.

   3.2. For every $j \in [k]$ and $(i_1, \ldots, i_{j-1}) \in [n]^{j-1}$:
   $Y \leftarrow Y \cup \{c[\sigma(t_{i_1}, \ldots, t_{i_{j-1}}, \Box, \tau_{j+1}, \ldots, \tau_k)]\}$.

   3.3. For every $j \in [k]$:
   If $H_{\tau_j,Y}$ is not a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$ then
   $n \leftarrow n+1$, $t_n \leftarrow \tau_j$, $X \leftarrow X \cup \{t_n\}$.

   3.4. Go to 2.

Table 2: Exact learning algorithm LMTA for the class of multiplicity tree automata

Lemma 14 *Linear independence of the set of vectors $\{H_{t_1,Y}, \ldots, H_{t_n,Y}\}$ is an invariant of the loop consisting of Step 2 and Step 3.*

Proof We argue inductively on the number of iterations of the loop. The base case $n = 1$ clearly holds since $f(z) \neq 0$.

For the induction step, suppose that the set $\{H_{t_1,Y}, \ldots, H_{t_n,Y}\}$ is linearly independent at the start of an iteration of the loop. If a tree $t \in T_\Sigma$ is added to X during Step 2.1, then $H_{t,Y}$ is not a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$, and therefore $\{H_{t_1,Y}, \ldots, H_{t_n,Y}, H_{t,Y}\}$ is a linearly independent set of vectors. Hence, the set $\{H_{t_1,Y}, \ldots, H_{t_n,Y}\}$ is linearly independent at the start of Step 3.

Unless the algorithm halts in Step 3.1, it proceeds to Step 3.2 where the set of columns Y is increased, which clearly preserves linear independence of vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}$. If a tree τ_j is added to X in Step 3.3, then $H_{\tau_j,Y}$ is not a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$ which implies that the vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}, H_{\tau_j,Y}$ are linearly independent. Hence, the set $\{H_{t_1,Y}, \ldots, H_{t_n,Y}\}$ is linearly independent at the start of the next iteration of the loop. This completes the induction step. ■

Secondly, we show that Step 2.2 of LMTA can always be performed.

Lemma 15 *Whenever Step 2.2 starts, for every $k \in \{0, \ldots, m\}$ and $\sigma \in \Sigma_k$ there exists a unique matrix $\mu(\sigma) \in \mathbb{F}^{n^k \times n}$ satisfying Equation (12).*

Proof Take any $(i_1, \ldots, i_k) \in [n]^k$. Step 2.1 ensures that $H_{\sigma(t_{i_1},\ldots,t_{i_k}),Y}$ can be represented as a linear combination of vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}$. This representation is unique since $H_{t_1,Y}, \ldots, H_{t_n,Y}$ are linearly independent vectors by Lemma 14. Row $\mu(\sigma)_{(i_1,\ldots,i_k)} \in \mathbb{F}^{1\times n}$ is, therefore, uniquely defined by the equation $\mu(\sigma)_{(i_1,\ldots,i_k)} \cdot H_{X,Y} = H_{\sigma(t_{i_1},\ldots,t_{i_k}),Y}$. ■

Thirdly, we show that Step 3.1 of LMTA can always be performed.

Lemma 16 *Suppose that upon making an equivalence query on A in Step 3.1, the Learner receives the answer NO and a counterexample z. Then, there exists a subtree $\sigma(\tau_1, \ldots, \tau_k)$ of z that satisfies the following two conditions:*

(i) For every $j \in [k]$, $H_{\tau_j,Y} = \mu(\tau_j) \cdot H_{X,Y}$.

(ii) For some $c \in Y$, $H_{\sigma(\tau_1,\ldots,\tau_k),c} \neq \mu(\sigma(\tau_1, \ldots, \tau_k)) \cdot H_{X,c}$.

Proof Towards a contradiction, assume that there exists no subtree $\sigma(\tau_1, \ldots, \tau_k)$ of z that satisfies conditions (i) and (ii). We claim that then for every subtree τ of z, it holds that

$$H_{\tau,Y} = \mu(\tau) \cdot H_{X,Y}. \tag{13}$$

In the following we prove this claim using induction on $height(\tau)$. The base case $\tau \in \Sigma_0$ follows immediately from Equation (12). For the induction step, let $0 \leq h < height(z)$ and assume that Equation (13) holds for every subtree $\tau \in T_\Sigma^{\leq h}$ of z. Take an arbitrary subtree $\tau \in T_\Sigma^{h+1}$ of z. Then $\tau = \sigma(\tau_1, \ldots, \tau_k)$ for some $k \in [m]$, $\sigma \in \Sigma_k$, and $\tau_1, \ldots, \tau_k \in T_\Sigma^{\leq h}$, where $\tau_1, \ldots, \tau_k$ are subtrees of z. The induction hypothesis implies that $H_{\tau_j,Y} = \mu(\tau_j) \cdot H_{X,Y}$ holds for every $j \in [k]$. Hence, subtree τ satisfies condition (i). By assumption, no subtree of z satisfies both conditions (i) and (ii). Thus τ does not satisfy condition (ii), i.e., it holds that $H_{\tau,Y} = \mu(\tau) \cdot H_{X,Y}$. This completes the proof by induction.

Equation (13) for $\tau = z$ gives $H_{z,Y} = \mu(z) \cdot H_{X,Y}$. Since $\Box \in Y$, this in particular implies that

$$f(z) = H_{z,\Box} = \mu(z) \cdot H_{X,\Box} = \mu(z) \cdot \gamma = \|A\|(z),$$

which yields a contradiction since z is a counterexample for the hypothesis A. ■

Finally, we show that the row set X is augmented with at least one element in each iteration of the main loop.

Lemma 17 *Every complete iteration of the Step 2 - 3 loop strictly increases the cardinality of the row set X.*

Proof It suffices to show that in Step 3.3 at least one of the trees $\tau_1, \ldots, \tau_k$ is added to X. By Lemma 14, at the start of Step 3.2 vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}$ are linearly independent. Thus by condition (i) of Step 3.1, for every $j \in [k]$ it holds that

$$H_{\tau_j,Y} = \mu(\tau_j) \cdot H_{X,Y} \tag{14}$$

and, moreover, Equation (14) is the unique representation of vector $H_{\tau_j,Y}$ as a linear combination of vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}$. Clearly, vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}$ remain linearly independent when Step 3.2 ends.

Towards a contradiction, assume that in Step 3.3 none of the trees $\tau_1, \ldots, \tau_k$ is added to X. This means that for every $j \in [k]$, vector $H_{\tau_j,Y}$ can be represented as a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$. The latter representation is unique, since vectors $H_{t_1,Y}, \ldots, H_{t_n,Y}$ are linearly independent, and is given by Equation (14). By condition (ii) of Step 3.1 and Equations (12) and (1), we now have that

$$\begin{aligned} H_{\sigma(\tau_1,\ldots,\tau_k),c} &\neq \mu(\sigma(\tau_1, \ldots, \tau_k)) \cdot H_{X,c} \\ &= (\mu(\tau_1) \otimes \cdots \otimes \mu(\tau_k)) \cdot \mu(\sigma) \cdot H_{X,c} \\ &= (\mu(\tau_1) \otimes \ldots \otimes \mu(\tau_k)) \cdot H_{\sigma(X,\ldots,X),c} \\ &= \sum_{(i_1,\ldots,i_k)\in[n]^k} \left(\prod_{j=1}^{k} \mu(\tau_j)_{i_j} \right) \cdot H_{\sigma(t_{i_1},\ldots,t_{i_k}),c}. \end{aligned} \tag{15}$$

By Step 3.2, it holds that $c[\sigma(t_{i_1}, \ldots, t_{i_{j-1}}, \Box, \tau_{j+1}, \ldots, \tau_k)] \in Y$ for every $j \in [k]$ and every $(i_1, \ldots, i_{j-1}) \in [n]^{j-1}$. Thus by Equation (14) for $j = k$, we have

$$\begin{aligned} &\sum_{(i_1,\ldots,i_k)\in[n]^k} \left(\prod_{j=1}^{k} \mu(\tau_j)_{i_j} \right) \cdot H_{\sigma(t_{i_1},\ldots,t_{i_k}),c} \\ &= \sum_{(i_1,\ldots,i_{k-1})\in[n]^{k-1}} \left(\prod_{j=1}^{k-1} \mu(\tau_j)_{i_j} \right) \cdot \sum_{i\in[n]} \mu(\tau_k)_i \cdot H_{t_i,c[\sigma(t_{i_1},\ldots,t_{i_{k-1}},\Box)]} \\ &= \sum_{(i_1,\ldots,i_{k-1})\in[n]^{k-1}} \left(\prod_{j=1}^{k-1} \mu(\tau_j)_{i_j} \right) \cdot \mu(\tau_k) \cdot H_{X,c[\sigma(t_{i_1},\ldots,t_{i_{k-1}},\Box)]} \\ &= \sum_{(i_1,\ldots,i_{k-1})\in[n]^{k-1}} \left(\prod_{j=1}^{k-1} \mu(\tau_j)_{i_j} \right) \cdot H_{\tau_k,c[\sigma(t_{i_1},\ldots,t_{i_{k-1}},\Box)]}. \end{aligned} \tag{16}$$

Proceeding inductively as above and applying Equation (14) for every $j \in \{k-1, \ldots, 1\}$, we get that the expression of (16) is equal to $H_{\tau_1,c[\sigma(\Box,\tau_2,\ldots,\tau_k)]}$. However, this contradicts Equation (15). The result follows. ■

Putting together Lemmas 14 - 17, we conclude the following:

Proposition 18 *Let Σ be a ranked alphabet and $\mathbb{F}$ be a field. Let $f \in Rec(\Sigma, \mathbb{F})$, let H be the Hankel matrix of f, and let r be the rank (over $\mathbb{F}$) of H. On target f, algorithm* LMTA *outputs a minimal MTA-representation of f after at most r iterations of the loop consisting of Step 2 and Step 3.*

Proof Lemmas 15 and 16 show that every step of algorithm LMTA can be performed.

Theorem 3 implies that r is finite. From Lemma 14 we know that, whenever Step 2 starts, matrix $H_{X,Y}$ has full row rank and thus $n = |X| \leq r$. Lemma 17 implies that n increases by at least one in each iteration of the Step 2 - 3 loop. Therefore, the number of iterations of the loop is at most r.

The proof follows by observing that LMTA halts only upon receiving the answer YES to an equivalence query. ■

4.3 Succinct Representations

In this subsection, we explain how algorithm LMTA can be correctly implemented using a DAG representation of trees. In particular, we assume that membership queries are made on Σ-DAGs, that the counterexamples are given as Σ-DAGs, the elements of X are Σ-DAGs, and the elements of Y are DAG representations of Σ-contexts, i.e., $(\{\Box\} \cup \Sigma)$-DAGs.

As shown in Section 2.5, multiplicity tree automata can run directly on DAGs and, moreover, they assign equal weight to a DAG and to its tree unfolding. Crucially also, as explained in the proof of Theorem 19, Step 3.1 can be run directly on a DAG representation of the counterexample, without unfolding. Specifically, Step 3.1 involves multiple executions of the hypothesis automaton on trees. By Proposition 7, we can faithfully carry out these executions on DAG representations of trees. Step 3.1 also involves considering all the subtrees of a given counterexample. However, by Proposition 5, this is equivalent to looking at all the sub-DAGs of a DAG representation of the counterexample.

At various points in the algorithm, we take $c \in Y$, $t \in X$ and compute their concatenation $c[t]$ in order to determine the corresponding entry $H_{t,c}$ of the Hankel matrix by making a membership query. Proposition 6 implies that this can be done faithfully using DAG representations of Σ-trees and Σ-contexts.

4.4 Complexity Analysis

In this subsection, we give a query and computational complexity analysis of our algorithm and compare it to the best previously-known exact learning algorithm for multiplicity tree automata (Habrard and Oncina, 2006) showing in particular an exponential improvement on the query complexity and the running time in the worst case.

Theorem 19 *Let $f \in Rec(\Sigma, \mathbb{F})$ where Σ has rank m and $\mathbb{F}$ is a field. Let A be a minimal MTA-representation of f, and let r be the dimension of A. Then, f is learnable by the algorithm* LMTA*, making $r + 1$ equivalence queries, $|A|^2 + |A| \cdot s$ membership queries, and $O(|A|^2 + |A| \cdot r \cdot s)$ arithmetic operations, where s denotes the size of a largest counterexample z, represented as a DAG, that is obtained during the execution of the algorithm.*

Proof Let H be the Hankel matrix of f. Note that, by Theorem 3, the rank of H is equal to r. Proposition 18 implies that on target f, algorithm LMTA outputs a minimal MTA-representation of f after at most r iterations of the Step 2 - 3 loop, thereby making at most $r+1$ equivalence queries.

From Lemma 14 we know that matrix $H_{X,Y}$ has full row rank, which implies that $|X| \leq r$. As for the cardinality of the column set Y, at the end of Step 1 we have $|Y| = 1$. Furthermore, in each iteration of Step 3.2 the number of columns added to Y is at most

$$\sum_{j=1}^{k} n^{j-1} \leq \sum_{j=1}^{k} r^{j-1} = \frac{r^k - 1}{r-1} \leq \frac{r^m - 1}{r-1},$$

where k and n are as defined in Step 3.2. Since the number of iterations of Step 3.2 is at most $r-1$, we have $|Y| \leq r^m$.

The number of membership queries made in Step 2 over the whole algorithm is

$$\left(\sum_{\sigma \in \Sigma} |\sigma(X, \ldots, X)| + |X| \right) \cdot |Y|$$

because the Learner needs to ask for the values of the entries of matrices $H_{X,Y}$ and $H_{\sigma(X,\ldots,X),Y}$ for every $\sigma \in \Sigma$.

To analyse the number of membership queries made in Step 3, we now detail the procedure by which an appropriate sub-DAG of the counterexample z is found in Step 3.1. By Lemma 16, there exists a sub-DAG τ of z such that $H_{\tau,Y} \neq \mu(\tau) \cdot H_{X,Y}$. Thus given a counterexample z in Step 3.1, the procedure for finding a required sub-DAG of z is as follows: Check if $H_{\tau,Y} = \mu(\tau) \cdot H_{X,Y}$ for every sub-DAG τ of z in a nondecreasing order of height; stop when a sub-DAG τ is found such that $H_{\tau,Y} \neq \mu(\tau) \cdot H_{X,Y}$.

In each iteration of Step 3, the Learner makes $size(z) \cdot |Y| \leq s \cdot |Y|$ membership queries because, for every sub-DAG τ of z, the Learner needs to ask for the values of the entries of vector $H_{\tau,Y}$. All together, the number of membership queries made during the execution of the algorithm is at most

$$\begin{aligned} &\left(\sum_{\sigma \in \Sigma} |\sigma(X, \ldots, X)| + |X| \right) \cdot |Y| + (r-1) \cdot s \cdot |Y| \\ &\leq \left(\sum_{\sigma \in \Sigma} r^{rk(\sigma)} + r \right) \cdot r^m + (r-1) \cdot s \cdot r^m \leq |A|^2 + |A| \cdot s. \end{aligned}$$

As for the arithmetic complexity, in Step 2.1 one can determine if a vector $H_{\sigma(t_{i_1},\ldots,t_{i_k}),Y}$ is a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$ via Gaussian elimination using $O(n^2 \cdot |Y|)$ arithmetic operations (see Cohen, 1993, Section 2.3). Analogously, in Step 3.3 one can determine if $H_{\tau_j,Y}$ is a linear combination of $H_{t_1,Y}, \ldots, H_{t_n,Y}$ via Gaussian elimination using $O(n^2 \cdot |Y|)$ arithmetic operations. Since $|X| \leq r$ and $|Y| \leq r^m$, all together Step 2.1 and Step 3.3 require at most $O(|A|^2)$ arithmetic operations.

Lemma 15 implies that in each iteration of Step 2.2, for every $\sigma \in \Sigma$ there exists a unique matrix $\mu(\sigma) \in \mathbb{F}^{n^{rk(\sigma)} \times n}$ that satisfies Equation (12). To perform an iteration of Step 2.2,

we first put matrix $H_{X,Y}$ in echelon form and then, for each $\sigma \in \Sigma$, solve Equation (12) for $\mu(\sigma)$ by back substitution. It follows from standard complexity bounds on the conversion of matrices to echelon form (Cohen, 1993, Section 2.3) that the total operation count for Step 2.2 can be bounded above by $O(|A|^2)$.

Finally, let us consider the arithmetic complexity of Step 3.1. In every iteration, for each sub-DAG τ of the counterexample z the Learner needs to compute the vector $\mu(\tau)$ and the product $\mu(\tau) \cdot H_{X,Y}$. Note that $\mu(\tau)$ can be computed bottom-up from the sub-DAGs of τ. Since z has at most s sub-DAGs, Step 3.1 requires at most $O(|A| \cdot r \cdot s)$ arithmetic operations. All together, the algorithm requires at most $O(|A|^2 + |A| \cdot r \cdot s)$ arithmetic operations. ■

Algorithm LMTA can be used to show that over $\mathbb{Q}$, multiplicity tree automata are exactly learnable in randomised polynomial time. The key idea is to represent numbers as arithmetic circuits. In executing LMTA, the Learner need only perform arithmetic operations on circuits (addition, subtraction, multiplication, and division), which can be done in constant time, and equality testing, which can be done in coRP (see Arora and Barak, 2009). These suffice for all the operations detailed in the proof of Theorem 19; in particular they suffice for Gaussian elimination, which can be used to implement the linear-independence checks in LMTA.

The complexity of algorithm LMTA should be compared to the complexity of the algorithm of Habrard and Oncina (2006), which learns multiplicity tree automata by making $r+1$ equivalence queries, $|A| \cdot s$ membership queries, and a number of arithmetic operations polynomial in $|A|$ and s, where s is the size of a largest counterexample given as a tree. Note that the algorithm of Habrard and Oncina (2006) cannot be straightforwardly adapted to work directly with DAG representations of trees since when given a counterexample z, every suffix of z is added to the set of columns. However, the tree unfolding of a DAG can have exponentially many different suffixes in the size of the DAG. For example, the DAG in Figure 2 has size n, and its tree unfolding, shown in Figure 1, has $O(2^n)$ different suffixes.

5. Lower Bounds on Query Complexity of Learning MTA

In this section, we study lower bounds on the query complexity of learning multiplicity tree automata in the exact learning model. Our results generalise the corresponding lower bounds for learning multiplicity word automata by Bisht et al. (2006), and make no assumption about the computational model of the learning algorithm.

First, we give a lower bound on the total number of queries required by an exact learning algorithm that works over any field, which is the situation of our algorithm in Section 4. Note that when we say that an algorithm works over any field, we mean that it just uses field arithmetic, equality testing, and the ability to store and communicate field elements to the Teacher, and its correctness depends only on these operations satisfying the field axioms.

Theorem 20 *Any exact learning algorithm that learns the class of multiplicity tree automata of dimension at most r, over a ranked alphabet (Σ, rk) and any field, must make at least $\sum_{\sigma \in \Sigma} r^{rk(\sigma)+1} - r^2$ queries.*

Proof Take an arbitrary exact learning algorithm L that learns the class of multiplicity tree automata of dimension at most r, over a ranked alphabet (Σ, rk) and over any field.

Let $\mathbb{F}$ be any field. Let $\mathbb{K} := \mathbb{F}(\{z^\sigma_{i,j} : \sigma \in \Sigma, i \in [r^{rk(\sigma)}], j \in [r]\})$ be an extension field of $\mathbb{F}$, where the set $\{z^\sigma_{i,j} : \sigma \in \Sigma, i \in [r^{rk(\sigma)}], j \in [r]\}$ is algebraically independent over $\mathbb{F}$. We define a 'generic' $\mathbb{K}$-multiplicity tree automaton $A := (r, \Sigma, \mu, \gamma)$ where $\gamma = e_1^\top \in \mathbb{F}^{r\times 1}$ and $\mu(\sigma) = [z^\sigma_{i,j}]_{i,j} \in \mathbb{K}^{r^{rk(\sigma)}\times r}$ for every $\sigma \in \Sigma$. We define a tree series $f := \|A\|$. Observe that every r-dimensional $\mathbb{F}$-MTA over Σ can be obtained from A by substituting values from the field $\mathbb{F}$ for the variables $z^\sigma_{i,j}$. Thus if the Hankel matrix of f had rank less than r, then every r-dimensional $\mathbb{F}$-MTA over Σ would have Hankel matrix of rank less than r. Therefore, the Hankel matrix of f has rank r.

We run algorithm L on the target function f. By assumption, the output of L is an MTA $A' = (r, \Sigma, \mu', \gamma')$ such that $\|A'\| \equiv f$. Let n be the number of queries made by L on target f. Let $t_1, \ldots, t_n \in T_\Sigma$ be the trees on which L either made a membership query, or which were received as the counterexample to an equivalence query. Then for every $l \in [n]$, there exists a multivariate polynomial $p_l \in \mathbb{F}[(z^\sigma_{i,j})_{i,j,\sigma}]$ such that $f(t_l) = p_l$.

Note that both A and A' are minimal MTA-representations of f. Thus by Theorem 4, there exists an invertible matrix $U \in \mathbb{K}^{r\times r}$ such that $\gamma = U\cdot\gamma'$ and $\mu(\sigma) = U^{\otimes rk(\sigma)}\cdot\mu'(\sigma)\cdot U^{-1}$ for every $\sigma \in \Sigma$. This implies that the entries of matrices $\mu(\sigma)$, $\sigma \in \Sigma$, lie in an extension of $\mathbb{F}$ generated by the entries of U and $\{p_l : l \in [n]\}$, i.e., by at most $r^2 + n$ elements. But since the entries of matrices $\mu(\sigma)$, $\sigma \in \Sigma$, form an algebraically independent set over $\mathbb{F}$, the total number $\sum_{\sigma\in\Sigma} r^{rk(\sigma)+1}$ of such entries is at most $r^2 + n$. Therefore, the number of queries n is at least $\sum_{\sigma\in\Sigma} r^{rk(\sigma)+1} - r^2$. ■

One may wonder whether a learning algorithm could do better over a specific field $\mathbb{F}$ by exploiting particular features of that field such as having zero characteristic, being ordered, or being algebraically closed. In this setting, we have the following lower bound.

Theorem 21 *Let $\mathbb{F}$ be a fixed but arbitrary field. Any exact learning algorithm that learns the class of $\mathbb{F}$-multiplicity tree automata of dimension at most r, over a ranked alphabet (Σ, rk) that has rank m and contains at least one unary symbol, must make number of queries at least*

$$\frac{1}{2^{m+1}} \cdot \left(\sum_{\sigma\in\Sigma} r^{rk(\sigma)+1} - r^2 - r \right).$$

Proof Without loss of generality, we can assume that r is even and can, therefore, define a natural number $n := r/2$. Let L be an exact learning algorithm for the class of $\mathbb{F}$-multiplicity tree automata of dimension at most r, over a ranked alphabet (Σ, rk) of rank m such that $rk^{-1}(\{1\}) \neq \emptyset$. We will identify a class of functions $\mathcal{C}$ such that L has to make at least $\sum_{\sigma\in\Sigma} n^{rk(\sigma)+1} - n^2 - n$ queries to distinguish between the members of $\mathcal{C}$.

Let $\sigma_0, \sigma_1 \in \Sigma$ be a nullary and a unary symbol, respectively. Let $P \in \mathbb{F}^{n\times n}$ be the permutation matrix corresponding to the cycle $(1, 2, \ldots, n)$. Define $\mathcal{A}$ to be the set of all $\mathbb{F}$-multiplicity tree automata $(2n, \Sigma, \mu, \gamma)$ where:

- $\mu(\sigma_0) = \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes e_1$ and $\mu(\sigma_1) = I_2 \otimes P$;

- For each k-ary symbol $\sigma \in \Sigma \setminus \{\sigma_0, \sigma_1\}$, there exists $B(\sigma) \in \mathbb{F}^{n^k \times n}$ such that

$$\mu(\sigma) = \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\begin{bmatrix} I_n \\ -I_n \end{bmatrix}^{\otimes k} \cdot B(\sigma) \right);$$

- $\gamma = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top \otimes e_1^\top$.

We define a set of recognisable tree series $\mathcal{C} := \{\|A\| : A \in \mathcal{A}\}$.

In Lemma 22 we state some properties of the functions in $\mathcal{C}$. Specifically, we show that the coefficient of a tree $t \in T_\Sigma$ in any series $f \in \mathcal{C}$ fundamentally depends on whether t has zero, one, or at least two nodes whose label is not σ_0 or σ_1. Here for every $i \in \mathbb{N}_0$ and $t \in T_\Sigma$, we use $\sigma_1^i(t)$ to denote the tree $\underbrace{\sigma_1(\sigma_1(\ldots \sigma_1}_{i}(t)\ldots))$.

Lemma 22 *The following properties hold for every $f \in \mathcal{C}$ and $t \in T_\Sigma$:*

(i) If $t = \sigma_1^j(\sigma_0)$ where $j \in \{0, 1, \ldots, n-1\}$, then $f(\sigma_0) = 1$ and $f(\sigma_1^j(\sigma_0)) = 0$ for $j > 0$.

(ii) If $t = \sigma_1^j(\sigma(\sigma_1^{i_1}(\sigma_0), \ldots, \sigma_1^{i_k}(\sigma_0)))$ where $k \in \{0, 1, ..., m\}$, $\sigma \in \Sigma_k \setminus \{\sigma_0, \sigma_1\}$, and $j, i_1, \ldots, i_k \in \{0, 1, \ldots, n-1\}$, then $f(t) = B(\sigma)_{(1+i_1,\ldots,1+i_k),(1+n-j) \bmod n}$.

(iii) If $\sum_{\sigma \in \Sigma \setminus \{\sigma_0,\sigma_1\}} \#_\sigma(t) \geq 2$, then $f(t) = 0$.

Proof Let $A = (2n, \Sigma, \mu, \gamma) \in \mathcal{A}$ be such that $\|A\| \equiv f$. First, we prove property *(i)*. Using Equation (2) and the mixed-product property of Kronecker product, we get that

$$\mu(\sigma_1^j(\sigma_0)) \quad = \mu(\sigma_0) \cdot \mu(\sigma_1)^j = (\begin{bmatrix} 1 & 0 \end{bmatrix} \otimes e_1) \cdot (I_2 \otimes P^j) = \begin{bmatrix} 1 & 0 \end{bmatrix} \otimes e_1 P^j \tag{17}$$

and therefore

$$\begin{aligned} f(\sigma_1^j(\sigma_0)) = \mu(\sigma_1^j(\sigma_0)) \cdot \gamma &= (\begin{bmatrix} 1 & 0 \end{bmatrix} \otimes e_1 P^j) \cdot (\begin{bmatrix} 1 & 0 \end{bmatrix}^\top \otimes e_1^\top) \\ &= (\begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \end{bmatrix}^\top) \otimes (e_1 P^j \cdot e_1^\top) = e_{j+1} \cdot e_1^\top. \end{aligned} \tag{18}$$

If $j = 0$ then the expression of (18) is equal to 1, otherwise the expression of (18) is equal to 0. This completes the proof of property *(i)*.

Next, we prove property *(ii)*. By the mixed-product property of Kronecker product and Equations (2), (3), and (17), we have

$$\begin{aligned} &\mu(\sigma_1^j(\sigma(\sigma_1^{i_1}(\sigma_0), \ldots, \sigma_1^{i_k}(\sigma_0)))) \\ &= \left(\bigotimes_{l=1}^{k} \mu(\sigma_1^{i_l}(\sigma_0)) \right) \cdot \mu(\sigma) \cdot \mu(\sigma_1)^j \\ &= \left(\begin{bmatrix} 1 \end{bmatrix} \otimes \bigotimes_{l=1}^{k} \mu(\sigma_1^{i_l}(\sigma_0)) \right) \cdot \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\begin{bmatrix} I_n \\ -I_n \end{bmatrix}^{\otimes k} \cdot B(\sigma) \right) \right) \cdot (I_2 \otimes P)^j \\ &= \left((\begin{bmatrix} 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 \end{bmatrix}) \otimes \left(\bigotimes_{l=1}^{k} \mu(\sigma_1^{i_l}(\sigma_0)) \cdot \begin{bmatrix} I_n \\ -I_n \end{bmatrix}^{\otimes k} \cdot B(\sigma) \right) \right) \cdot (I_2 \otimes P^j) \end{aligned}$$

$$
\begin{aligned}
&= \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\bigotimes_{l=1}^{k} \left(\left(\begin{bmatrix} 1 & 0 \end{bmatrix} \otimes e_1 P^{i_l} \right) \cdot \begin{bmatrix} I_n \\ -I_n \end{bmatrix} \right) \cdot B(\sigma) \right) \right) \cdot (I_2 \otimes P^j) \\
&= \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\bigotimes_{l=1}^{k} e_1 P^{i_l} \cdot B(\sigma) \right) \right) \cdot (I_2 \otimes P^j) \\
&= \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \cdot I_2 \right) \otimes \left(\bigotimes_{l=1}^{k} e_{1+i_l} \cdot B(\sigma) \cdot P^j \right) \\
&= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes (B(\sigma)_{(1+i_1,\ldots,1+i_k)} \cdot P^j) \qquad (19)
\end{aligned}
$$

and therefore, using the fact that $P^n = I_n$, we get that

$$
\begin{aligned}
f(\sigma_1^j(\sigma(\sigma_1^{i_1}(\sigma_0), \ldots, \sigma_1^{i_k}(\sigma_0)))) &= \mu(\sigma_1^j(\sigma(\sigma_1^{i_1}(\sigma_0), \ldots, \sigma_1^{i_k}(\sigma_0)))) \cdot \gamma \\
&= (\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes (B(\sigma)_{(1+i_1,\ldots,1+i_k)} \cdot P^j)) \cdot (\begin{bmatrix} 1 & 0 \end{bmatrix}^\top \otimes e_1^\top) \\
&= (\begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \end{bmatrix}^\top) \otimes (B(\sigma)_{(1+i_1,\ldots,1+i_k)} \cdot P^j \cdot e_1^\top) \\
&= B(\sigma)_{(1+i_1,\ldots,1+i_k)} \cdot (e_1 P^{n-j})^\top \\
&= B(\sigma)_{(1+i_1,\ldots,1+i_k),(1+n-j) \bmod n}.
\end{aligned}
$$

Finally, we prove property *(iii)*. If $\sum_{\sigma \in \Sigma \setminus \{\sigma_0,\sigma_1\}} \#_\sigma(t) \geq 2$, then there exists a subtree $\sigma'(t_1, \ldots, t_k)$ of t where $k \geq 1$, $\sigma' \in \Sigma_k \setminus \{\sigma_1\}$, and $\sum_{\sigma \in \Sigma \setminus \{\sigma_0,\sigma_1\}} \#_\sigma(t_i) = 1$ for some $i \in [k]$. It follows from Equation (19) that $\mu(t_i) = \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \alpha$ for some $\alpha \in \mathbb{F}^{1 \times n}$. By the mixed-product property of Kronecker product and Equation (3), we have

$$
\begin{aligned}
\mu(\sigma'(t_1, \ldots, t_k)) &= \left(\bigotimes_{j=1}^{k} \mu(t_j) \right) \cdot \left(\begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\begin{bmatrix} I_n \\ -I_n \end{bmatrix}^{\otimes k} \cdot B(\sigma') \right) \right) \\
&= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\bigotimes_{j=1}^{k} \mu(t_j) \cdot \begin{bmatrix} I_n \\ -I_n \end{bmatrix}^{\otimes k} \cdot B(\sigma') \right) \\
&= \begin{bmatrix} 1 & 1 \end{bmatrix} \otimes \left(\bigotimes_{j=1}^{k} \left(\mu(t_j) \cdot \begin{bmatrix} I_n \\ -I_n \end{bmatrix} \right) \cdot B(\sigma') \right) = 0_{1 \times 2n}
\end{aligned}
$$

where the last equality holds because

$$
\mu(t_i) \cdot \begin{bmatrix} I_n \\ -I_n \end{bmatrix} = \begin{bmatrix} \alpha & \alpha \end{bmatrix} \cdot \begin{bmatrix} I_n \\ -I_n \end{bmatrix} = 0_{1 \times n}.
$$

Since $\sigma'(t_1, \ldots, t_k)$ is a subtree of t, we now have that $\mu(t) = 0_{1 \times 2n}$ and thus $f(t) = 0$. ■

Remark 23 *As $P^n = I_n$, we have $\mu(\sigma_1)^n = I_{2n}$. Thus for every $f \in \mathcal{C}$, $k \in \{0, 1, ..., m\}$, $\sigma \in \Sigma_k \setminus \{\sigma_0, \sigma_1\}$, and $j, i_1, \ldots, i_k \in \mathbb{N}_0$, it holds that $f(\sigma_1^j(\sigma_0)) = f(\sigma_1^{j \bmod n}(\sigma_0))$ and*

$$
f(\sigma_1^j(\sigma(\sigma_1^{i_1}(\sigma_0), \ldots, \sigma_1^{i_k}(\sigma_0)))) = f(\sigma_1^{j \bmod n}(\sigma(\sigma_1^{i_1 \bmod n}(\sigma_0), \ldots, \sigma_1^{i_k \bmod n}(\sigma_0)))).
$$

Returning to the proof of Theorem 21, let us run the learning algorithm L on a target $f \in \mathcal{C}$. Lemma 22 (i) and Remark 23 imply that when L makes a membership query on $t = \sigma_1^j(\sigma_0)$ where $j \in \mathbb{N}_0$, the Teacher returns 1 if $j \bmod n = 0$ and returns 0 otherwise. Furthermore, by Lemma 22 (iii), when L makes a membership query on $t \in T_\Sigma$ such that $\sum_{\sigma \in \Sigma \setminus \{\sigma_0, \sigma_1\}} \#_\sigma(t) \geq 2$, the Teacher returns 0. In these cases, L does not gain any new information about f since every function in $\mathcal{C}$ is consistent with the values returned by the Teacher.

When L makes a membership query on a tree $t = \sigma_1^j(\sigma(\sigma_1^{i_1}(\sigma_0), \ldots, \sigma_1^{i_k}(\sigma_0)))$, where $k \in \{0, 1, ..., m\}$, $\sigma \in \Sigma_k \setminus \{\sigma_0, \sigma_1\}$, and $j, i_1, \ldots, i_k \in \mathbb{N}_0$, the Teacher returns an arbitrary number from the field $\mathbb{F}$ if the value $f(t)$ is not already known from an earlier query. It follows from Lemma 22 (ii) and Remark 23 that L thereby learns the entry

$$B(\sigma)_{(1+(i_1 \bmod n), ..., 1+(i_k \bmod n)), (1+n-j) \bmod n}.$$

When L makes an equivalence query on a hypothesis $h \in \mathcal{C}$, the Teacher finds some entry $B(\sigma)_{(i_1,...,i_k),j}$ that L does not already know from previous queries and returns the tree $\sigma_1^{1+n-j}(\sigma(\sigma_1^{i_1-1}(\sigma_0), \ldots, \sigma_1^{i_k-1}(\sigma_0)))$ as the counterexample.

With each query, the Learner L learns at most one entry of $B(\sigma)$ where $\sigma \in \Sigma \setminus \{\sigma_0, \sigma_1\}$. The number of queries made by L on target f is, therefore, at least the total number of entries of matrices $B(\sigma)$ for all $\sigma \in \Sigma \setminus \{\sigma_0, \sigma_1\}$. The latter number is equal to

$$\begin{aligned} \sum_{\sigma \in \Sigma \setminus \{\sigma_0, \sigma_1\}} n^{rk(\sigma)+1} &\geq \frac{1}{2^{m+1}} \cdot \sum_{\sigma \in \Sigma \setminus \{\sigma_0, \sigma_1\}} r^{rk(\sigma)+1} \\ &= \frac{1}{2^{m+1}} \cdot \left(\sum_{\sigma \in \Sigma} r^{rk(\sigma)+1} - r^2 - r \right). \end{aligned}$$

This completes the proof. ■

The lower bounds of Theorem 20 and Theorem 21 are both linear in the target automaton size. Note that when the alphabet rank is fixed, the lower bound for learning over a fixed field (Theorem 21) is the same, up to a constant factor, as for learning over an arbitrary field (Theorem 20).

Assuming a Teacher that represents counterexamples as succinctly as possible (see Section 3.2 for details), the upper bound of algorithm LMTA from Theorem 19 is quadratic in the target automaton size and, therefore, also quadratic in the lower bound of Theorem 20.

6. Conclusions and Future Work

In this work, we have characterised the query and computational complexity of learning multiplicity tree automata in the exact learning model. We gave the first-known lower bound on the number of queries needed by any exact learning algorithm to learn a target recognisable tree series. This bound is linear in the size of a smallest multiplicity tree automaton recognising the series. We also gave a new learning algorithm whose query complexity is quadratic in the size of a smallest automaton recognising the target tree series and linear in the size of a largest DAG counterexample provided by the Teacher. With

regard to computational complexity, we show that the problem of deciding equivalence of multiplicity tree automata is logspace equivalent to polynomial identity testing.

The algebraic theory of recognisable word series, notably the connection to finite-rank Hankel matrices, generalises naturally to recognisable tree series and underlies many of the approaches to learning tree automata, including the present paper (see Section 1.1 for more details). In the case of trees, however, the issue of succinctness of automaton and counterexample representations comes to the fore. As we have noted, the smallest counterexample to the equivalence of two tree automata may be exponential in their total size. Therefore, in order to obtain even a polynomial query complexity, our learning algorithm works with a succinct representation of trees in terms of DAGs. The assumption of a Teacher that provides succinct DAG counterexamples is reasonable in light of the fact that the algorithm of Seidl (1990) for deciding equivalence of multiplicity tree automata can easily be modified to produce DAG counterexamples of minimal size in case of inequivalence.

The issue of succinctness of automaton representations seems to be more subtle and has not been addressed in the present paper. Here we have used the standard definition of automaton size, in which an automaton with n states and maximum alphabet rank m necessarily has size at least n^{m+1}. Adopting a sparse encoding of the transition matrices may result in an exponentially more succinct automaton representation. However, it seems a difficult problem to efficiently learn an automaton of minimal size under a sparse representation of transition matrices. In this regard, note that two different MTAs recognising the same tree series, both with a minimal number of states, can have considerably different sizes under a sparse representation since minimal MTAs are only unique up to change of basis.

One route to obtaining succinct automaton representations in the case of alphabets of unbounded rank is to use the encoding of unranked alphabets into binary alphabets presented by Comon et al. (2007) and Bailly et al. (2010). Such an encoding would potentially allow to use our learning algorithm to learn recognisable tree series over an arbitrary alphabet Σ (including even unranked alphabets) while maintaining hypothesis automaton and Hankel matrix over a binary alphabet. Note though that if the algorithm were required to present its hypotheses to the Teacher as automata over the original alphabet Σ, then it would need to translate automata over the binary encoding to corresponding automata over Σ—potentially leading to an exponential blow-up.

With regard to applications of tree-automaton learning algorithms to other problems, we recall that Beimel et al. (2000) apply their exact learning algorithm for multiplicity word automata to show exact learnability of certain classes of polynomials over both finite and infinite fields. Beimel et al. (2000) also prove the learnability of disjoint DNF formulae (i.e., DNF formulae in which each assignment satisfies at most one term) and, more generally, disjoint unions of geometric boxes over finite domains.

The learning framework considered in this paper concerns multiplicity tree automata, which are strictly more expressive than multiplicity word automata. Moreover, our result on the computational complexity of equivalence testing for multiplicity tree automata shows that, through equivalence queries, the Learner essentially has an oracle for polynomial identity testing. Thus a natural direction for future work is to seek to apply our algorithm to derive new results on exact learning of other concept classes, such as propositional formulae and polynomials (both in the commutative and noncommutative cases). In this direction, we

plan to examine the relationship of our work with that of Klivans and Shpilka (2006) on exact learning of algebraic branching programs and arithmetic circuits and formulae. The latter paper relies on rank bounds for Hankel matrices of polynomials in noncommuting variables, obtained by considering a generalised notion of partial derivative. Here we would like to determine whether the extra expressiveness of tree series can be used to show learnability of more general classes of formulae and circuits than have hitherto been handled using learnability of word series.

Sakakibara (1990) showed that context-free grammars (CFGs) can be learned efficiently from their structural descriptions in the exact learning model, using structural membership queries and structural equivalence queries. Specifically, Sakakibara, *op. cit.*, notes that the set of structural descriptions of a context-free grammar constitutes a rational tree language, and thereby reduces the problem of learning a context-free grammar from its structural descriptions to the problem of learning a tree automaton. Given the important role of weighted and probabilistic CFGs across a range of applications including linguistics, a natural next step would be to apply our algorithm to learn weighted CFGs. The idea is to reduce the problem of learning a weighted context-free grammar using structural membership queries and structural equivalence queries to the problem of learning a multiplicity tree automaton in the exact learning model. The basis for applying our algorithm in this setting is the fact that the tree series that maps unlabelled derivation trees to their total weights under a given weighted context-free grammar is recognisable.

Acknowledgments

The authors would like to thank Michael Benedikt for stimulating discussions and helpful advice. We would also like to thank the referees for providing detailed and constructive reports. Both authors gratefully acknowledge the support of the EPSRC.

References

A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, Reading, MA, 1974.

E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009.

S. Anantharaman, P. Narendran, and M. Rusinowitch. Closure properties and decision problems of DAG automata. *Information Processing Letters*, 94(5):231–240, 2005.

D. Angluin. Learning regular sets from queries and counterexamples. *Information and Computation*, 75(2):87–106, 1987.

D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, NY, 2009.

R. Bailly, F. Denis, and L. Ralaivola. Grammatical inference as a principal component analysis problem. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 33–40, 2009.

R. Bailly, A. Habrard, and F. Denis. A spectral approach for probabilistic grammatical inference on trees. In *Proceedings of the 21st International Conference on Algorithmic Learning Theory (ALT)*, volume 6331 of *LNCS*, pages 74–88, 2010.

B. Balle and M. Mohri. Spectral learning of general weighted automata via constrained matrix completion. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2168–2176, 2012.

A. Beimel, F. Bergadano, N. H. Bshouty, E. Kushilevitz, and S. Varricchio. Learning functions represented as multiplicity automata. *Journal of the ACM*, 47(3):506–530, 2000.

J. Berstel and C. Reutenauer. Recognizable formal power series on trees. *Theoretical Computer Science*, 18(2):115–148, 1982.

L. Bisht, N. H. Bshouty, and H. Mazzawi. On optimal learning algorithms for multiplicity automata. In *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*, volume 4005 of *LNCS*, pages 184–198. Springer, 2006.

S. Bozapalidis and A. Alexandrakis. Représentations matricielles des séries d'arbre reconnaissables. *Informatique Théorique et Applications (ITA)*, 23(4):449–459, 1989.

S. Bozapalidis and O. Louscou-Bozapalidou. The rank of a formal tree power series. *Theoretical Computer Science*, 27(1):211–215, 1983.

P. Buneman, M. Grohe, and C. Koch. Path queries on compressed XML. In *Proceedings of the 29th International Conference on Very Large Data Bases (VLDB)*, pages 141–152, 2003.

J. W. Carlyle and A. Paz. Realizations by stochastic finite automata. *Journal of Computer and System Sciences*, 5(1):26–40, 1971.

W. Charatonik. Automata on DAG representations of finite trees. Research Report MPI-I-1999-2-001, Max-Planck-Institut für Informatik, Saarbrücken, 1999.

H. Cohen. *A Course in Computational Algebraic Number Theory*. Springer-Verlag, Berlin, 1993.

H. Comon, M. Dauchet, R. Gilleron, C. Löding, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available at `http://tata.gforge.inria.fr/`, 2007.

R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978.

F. Denis and A. Habrard. Learning rational stochastic tree languages. In *Proceedings of the 18th International Conference on Algorithmic Learning Theory (ALT)*, volume 4754 of *LNAI*, pages 242–256, 2007.

F. Denis, M. Gybels, and A. Habrard. Dimension-free concentration bounds on Hankel matrices for spectral learning. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 449–457, 2014.

F. Drewes and J. Högberg. Query learning of regular tree languages: How to avoid dead states. *Theory of Computing Systems*, 40(2):163–185, 2007.

F. Drewes and H. Vogler. Learning deterministically recognizable tree series. *Journal of Automata, Languages and Combinatorics*, 12(3):332–354, 2007.

F. Drewes, J. Högberg, and A. Maletti. MAT learners for tree series: an abstract data type and two realizations. *Acta Informatica*, 48(3):165–189, 2011.

L. Feng, T. Han, M. Z. Kwiatkowska, and D. Parker. Learning-based compositional verification for synchronous probabilistic systems. In *Proceedings of the 9th International Symposium on Automated Technology for Verification and Analysis (ATVA)*, pages 511–521, 2011.

B. Fila and S. Anantharaman. Automata for analyzing and querying compressed documents. Research Report RR-2006-03, Laboratoire d'Informatique Fondamentale d'Orléans (LIFO), Université d'Orléans, France, 2006.

M. Fliess. Matrices de Hankel. *Journal de Mathématiques Pures et Appliquées*, 53(9): 197–222, 1974.

M. Frick, M. Grohe, and C. Koch. Query evaluation on compressed trees (extended abstract). In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 188–197, 2003.

E. M. Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

M. Gybels, F. Denis, and A. Habrard. Some improvements of the spectral learning approach for probabilistic grammatical inference. In *Proceedings of the 12th International Conference on Grammatical Inference (ICGI)*, pages 64–78, 2014.

A. Habrard and J. Oncina. Learning multiplicity tree automata. In *Proceedings of the 8th International Colloquium on Grammatical Inference: Algorithms and Applications (ICGI)*, volume 4201 of *LNCS*, pages 268–280. Springer, 2006.

D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

A. Kasprzik. Four one-shot learners for regular tree languages and their polynomial characterizability. *Theoretical Computer Science*, 485:85–106, 2013.

M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, 1994.

S. Kiefer, A. Murawski, J. Ouaknine, B. Wachter, and J. Worrell. On the complexity of equivalence and minimisation for $\mathbb{Q}$-weighted automata. *Logical Methods in Computer Science*, 9(1), 2013.

A. R. Klivans and A. Shpilka. Learning restricted models of arithmetic circuits. *Theory of Computing*, 2(1):185–206, 2006.

A. Maletti. Learning deterministically recognizable tree series - revisited. In *Proceedings of the 2nd International Conference on Algebraic Informatics (CAI)*, volume 4728 of *LNCS*, pages 218–235, 2007.

I. Marušić and J. Worrell. Complexity of equivalence and learning for multiplicity tree automata. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS), Part I*, volume 8634 of *LNCS*, pages 414–425. Springer, 2014.

Y. Sakakibara. Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, 76(2-3):223–242, 1990.

M. P. Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2–3):245–270, 1961.

J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980.

H. Seidl. Deciding equivalence of finite tree automata. *SIAM Journal on Computing*, 19 (3):424–437, 1990.

A. Sperduti and A. Starita. Supervised neural networks for the classification of structures. *IEEE Transactions on Neural Networks*, 8(3):714–735, 1997.

L. G. Valiant. Learning disjunctions of conjunctions. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI), Volume 1*, pages 560–566. Morgan Kaufmann, 1985.

R. E. Zippel. Probabilistic algorithms for sparse polynominals. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM)*, volume 72 of *LNCS*, pages 216–226. Springer, 1979.

Journal of Machine Learning Research 16 (2015) 2501-2542 Submitted 2/15; Revised 7/15; Published 12/15

Bayesian Nonparametric Covariance Regression

Emily B. Fox EBFOX@STAT.WASHINGTON.EDU
Department of Statistics
University of Washington
Seattle, WA 98195-4322, USA

David B. Dunson DUNSON@STAT.DUKE.EDU
Department of Statistical Science
Duke University
Durham, NC 27708-0251, USA

Editor: Edoardo M. Airoldi

Abstract

Capturing predictor-dependent correlations amongst the elements of a multivariate response vector is fundamental to numerous applied domains, including neuroscience, epidemiology, and finance. Although there is a rich literature on methods for allowing the variance in a univariate regression model to vary with predictors, relatively little has been done in the multivariate case. As a motivating example, we consider the Google Flu Trends data set, which provides indirect measurements of influenza incidence at a large set of locations over time (our predictor). To accurately characterize temporally evolving influenza incidence across regions, it is important to develop statistical methods for a time-varying covariance matrix. Importantly, the locations provide a redundant set of measurements and do not yield a sparse nor static spatial dependence structure. We propose to reduce dimensionality and induce a flexible Bayesian nonparametric covariance regression model by relating these location-specific trajectories to a lower-dimensional subspace through a latent factor model with predictor-dependent factor loadings. These loadings are in terms of a collection of basis functions that vary nonparametrically over the predictor space. Such low-rank approximations are in contrast to sparse precision assumptions, and are appropriate in a wide range of applications. Our formulation aims to address three challenges: scaling to large p domains, coping with missing values, and allowing an irregular grid of observations. The model is shown to be highly flexible, while leading to a computationally feasible implementation via Gibbs sampling. The ability to scale to large p domains and cope with missing values is fundamental in analyzing the Google Flu Trends data.

Keywords: covariance regression, dictionary learning, Gaussian process, latent factor model, nonparametric Bayes, time series

1. Introduction

Spurred by the increasing prevalence of high-dimensional data sets and the computational capacity to analyze them, capturing heteroscedasticity in multivariate processes has become a growing focus in many applied domains. For example, within the field of financial time series modeling, capturing the time-varying *volatility* and *co-volatility* of a collection of risky assets is key in devising a portfolio management scheme. Likewise, the spatial statistics community is often faced with multivariate measurements (e.g., temperature, precipitation,

etc.) recorded at a large collection of locations, necessitating methodology to model the strong spatial (and spatio-temporal) variations in correlations. Within neuroscience, there is interest in analyzing the time-varying coactivation patterns in brain activity, referred to as *functional connectivity*.

As a motivating example, we focus on the problem of modeling the changing correlations in flu activity amongst a large collection of regions in the United States as a function of time. The *Google Flu Trends* data set (available at `http://www.google.org/flutrends/`) provides estimates of flu activity in 183 regions on a weekly basis. The regions consist of the U.S. national level, 50 states, 10 regions, and 122 cities. A common strategy for modeling such data are Markov random fields (cf. Mugglin et al., 2002) (and relatedly, the kriging exploratory flu analysis of Sakai et al. (2004).) However, in addition to assuming (temporal) homoscedasticity, a limitation of such approaches is the typical reliance on a locally defined neighborhood structure that does not directly capture potential long-range dependencies (e.g., between New York and California.) Indeed, influenza spread can occur rapidly between non-contiguous regions (e.g., by air travel (Brownstein et al., 2006).) From exploratory data analysis, we find that the flu data does not yield a sparse graphical model structure. Instead, the redundancy between time series (e.g., Los Angeles and California) is naturally modeled via *low-rank approximations* that embed the observed trajectories in a low-dimensional subspace. Beyond its dimensionality, another challenge posed by this data set is the extent of missing data. For example, 25% of regions do not report data in the first year. The existing influenza modeling approaches described above rely on imputing such missing values, which we aim to avoid. The data attributes presented by the Google Flu Trends data set—redundancy in high dimensions, changing correlations, missing observations—are common to many applications.

In general terms, let $\boldsymbol{y} = (y_1, \ldots, y_p)' \in \Re^p$ denote a multivariate response and $\boldsymbol{x} = (x_1, \ldots, x_q)' \in \mathcal{X} \subset \Re^q$ an arbitrary multivariate predictor (e.g., *time*, *space*, etc.). In the flu analysis, p is the number of regions and $q = 1$ with $\boldsymbol{x}$ representing a scalar time index. A typical focus is on capturing the conditional mean $\mathrm{E}(\boldsymbol{y}|\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{x})$, assuming a *homoscedastic* model with conditional covariance $\mathrm{cov}(\boldsymbol{y}|\boldsymbol{x}) = \Sigma$. Recall that this covariance matrix captures key correlations between the elements of the response vector (e.g., flu activity in the various regions). In our exploratory analysis of the flu data in Appendix G, the residuals from a smoothing spline fit indicate that a model of i.i.d. errors across time is inappropriate for this data. In such cases, an assumption of homoscedasticity can have significant ramifications on inferences (e.g., predictive accuracy) as we demonstrate in Sections 4 and 5.2.2. It is possible to decrease residual correlation through a more intricate mean model, but the complexities of doing so motivate us to instead turn to modeling the conditional covariance. In particular, our focus is on developing Bayesian methods that allow not only $\mathrm{E}(\boldsymbol{y}|\boldsymbol{x}) = \boldsymbol{\mu}(\boldsymbol{x})$ but also $\mathrm{cov}(\boldsymbol{y}|\boldsymbol{x}) = \Sigma(\boldsymbol{x})$ to change flexibly with $\boldsymbol{x} \in \mathcal{X}$.

Classical strategies for estimating $\Sigma(\boldsymbol{x})$ rely on standard regression methods applied to the elements of the log or Cholesky decomposition of $\Sigma(\boldsymbol{x})$ or $\Sigma(\boldsymbol{x})^{-1}$ (Chiu et al., 1996; Pourahmadi, 1999; Leng et al., 2010; Zhang and Leng, 2012). This involves fitting $p(p+1)/2$ separate regression models, and hence these methods are ill-suited to high-dimensional applications due to the curse of dimensionality. Hoff and Niu (2012) instead proposed modeling $\Sigma(\boldsymbol{x})$ as a quadratic function of $\boldsymbol{x}$ plus a baseline positive definite matrix. The mapping from predictors to covariance assumes a parametric form, thus limiting the

model's expressivity. A nonparametric Nadaraya-Watson kernel estimator was proposed by Yin et al. (2010). Their approach is only appropriate for random $\boldsymbol{x}$ (i.e., not time series) and the kernel is required to be symmetric with a single bandwidth for all elements of $\Sigma(\boldsymbol{x})$. The result is a kernel estimator that may not be locally adaptive. For time series, heteroscedastic modeling has a long history (Chib et al., 2009), with the main approaches being multivariate generalized autoregressive conditional heteroscedasticity (GARCH) (Engle, 2002) (limited to applications with $p \leq 5$), multivariate stochastic volatility models (Harvey et al., 1994), and Wishart processes (Philipov and Glickman, 2006a,b; Gouriéroux et al., 2009). Central to the cited volatility models are assumptions of (i) Markov dynamics, limiting the ability to capture long-range dependencies, (ii) observation times that are equally spaced with no missing values, (iii) challenges in model fitting, and (iv) limited theory to justify flexibility.

We instead propose a Bayesian nonparametric approach to simultaneously modeling $\boldsymbol{\mu}(\boldsymbol{x})$ and $\Sigma(\boldsymbol{x})$. Using low-rank approximations as a parsimonious modeling technique when p is not small, we consider latent factor models with *predictor-dependent factor loadings*. In particular, we characterize the loadings as a sparse combination of unknown basis functions, with Gaussian processes providing a convenient prior for basis elements varying nonparametrically over $\mathcal{X}$. The induced covariance is then a regularized quadratic function of these basis elements. The proposed approach is provably flexible and admits a latent variable representation with simple conjugate posterior updates, which facilitates tractable posterior computation in moderate to high dimensions. In addition to being able to state theoretical properties of our proposed prior—such as large support integral to a Bayesian nonparametric approach—the proposed methodology has numerous practical advantages over previous covariance regression frameworks:

1. *Scaling to high dimensions in the presence of limited data* (via structured latent factor models)
2. *Handling irregular grid of observations* (via continuous functions as basis elements)
3. *Tractable computations* (via simple conjugate posterior updates)
4. *Coping with ignorable missing data* (no data imputation required)
5. *Robustness to outlying observations* (via sharing information in the latent basis).

Importantly, our framework enables analytic marginalization of missing data from the complete data likelihood, and without introducing extra dependencies amongst the remaining variables. The benefits of this analytic marginalization are two-fold: (1) we do not spend computational resources imputing the missing values, and (2) compared to the otherwise dramatically increased Markov chain Monte Carlo (MCMC) state space that includes the missing values, we can improve convergence and mixing rates through marginalization (Liu et al., 1994). Combined with the model's flexible sharing of information via the latent basis functions, we are able to handle data sets with substantial missing data, such as in the flu application of Section 5. Finally, the Google Flu Trends estimates are based on user search queries, and as such are susceptible to the types of malicious attacks that Google regularly guards against in other domains. Our model is well-geared for handling some forms of these situations: the inherent redundancy and borrowing of information across locations provides robustness to limited amounts of inaccurate estimates. Note that these inaccurate estimates may not be malicious in nature, but instead represent outliers arising from unusual spurs in search activity and poorly calibrated models (Cook et al., 2011). As long as these errors do not form systematic or stochastic trends or explosive processes (Fuller, 2009), our model

appears to be robust, as we demonstrate in Section 5.2.3. This is in contrast to approaches that look at rates in individual locations (e.g., Dukić et al. (2012)).

An earlier version of this work appeared in a technical report (Fox and Dunson, 2011); the current version provides significant additions including revised proofs, an extended model presentation, new experiments on the Google Flu Trends data, and an extensive model assessment. The recent work of Durante et al. (2014) builds on our framework and has shown great promise, but with a focus on time series applications and without handling missing data or scaling to large p domains.

The paper is organized as follows. In Section 2, we describe our proposed Bayesian nonparametric covariance regression model and analyze the theoretical properties of the model. Section 3 details the Gibbs sampling steps involved in our posterior computations. Finally, a number of simulation studies are examined in Section 4, with an application to the Google Flu Trends data set presented in Section 5.

2. Covariance Regression Priors

In this section, we consider the specific form for our Bayesian nonparametric covariance regression. Section 2.1 examines our assumed covariance structuring whereas Section 2.2 details our prior specification for the various model components.

2.1 Model Specification

We focus on a multivariate Gaussian nonparametric mean-covariance regression model

$$\boldsymbol{y}_i = \boldsymbol{\mu}(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim \mathrm{N}_p(0, \Sigma(\boldsymbol{x}_i)), \quad i = 1, \ldots, n, \tag{1}$$

with $\boldsymbol{x}_i \in \mathcal{X}$, $\mathcal{X}$ a compact subset of $\Re^q$, and the $\boldsymbol{\epsilon}_i$s independent. We focus on $\boldsymbol{x}$ non-random. In the flu application, $q = 1$ with $\{x_1, \ldots, x_n\}$ a set of week indices and $\boldsymbol{y}_i = \log \boldsymbol{r}_i$, the vector of log Google-estimated ILI rates in the 183 regions ($p = 183$) at time x_i. To cope with large p, we take model (1) to be induced through the factor model

$$\boldsymbol{y}_i = \Lambda(\boldsymbol{x}_i)\boldsymbol{\eta}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\eta}_i \sim \mathrm{N}_k(\boldsymbol{\psi}(\boldsymbol{x}_i), I_k), \quad \boldsymbol{\epsilon}_i \sim \mathrm{N}_p(0, \Sigma_0) \tag{2}$$

where $\Lambda(\boldsymbol{x})$ is a $p \times k$ *factor loadings matrix* specific to predictor value $\boldsymbol{x}$, $\boldsymbol{\eta}_i = (\eta_{i1}, \ldots, \eta_{ik})'$ are *latent factors* associated with observation $\boldsymbol{y}_i$, and $\Sigma_0 = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_p^2)$.

A latent factor model harnesses a lower-dimensional description of the observations, assuming $k \ll p$. $\boldsymbol{\psi}(\boldsymbol{x})$ captures the evolution of the latent factors whereas $\Lambda(\boldsymbol{x})$ dictates a low-rank evolution to the conditional covariance of the response vector. In particular, marginalizing out $\boldsymbol{\eta}_i$, the mean and covariance regression models are expressed as

$$\boldsymbol{\mu}(\boldsymbol{x}) = \Lambda(\boldsymbol{x})\boldsymbol{\psi}(\boldsymbol{x}), \quad \Sigma(\boldsymbol{x}) = \Lambda(\boldsymbol{x})\Lambda(\boldsymbol{x})' + \Sigma_0. \tag{3}$$

To make this concrete, in our flu application, $\boldsymbol{\eta}_i$ captures a small latent set of flu responses (not necessarily standard ILI rates) at week i, $\boldsymbol{\psi}(x)$ the evolution of these latent responses, and $\Lambda(x_i)$ a low-rank description of the spatial correlations at week i. The motivation for modeling the mean as in (3) arises from a desire to have a parsimonious model in large p domains. This is in contrast to, for example, a model $\boldsymbol{y}_i = \Lambda(\boldsymbol{x}_i)\boldsymbol{\eta}_i + \boldsymbol{\mu}(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i$ where $\boldsymbol{\eta}_i \sim N_k(0, I_k)$ and $\boldsymbol{\mu}(\boldsymbol{x}_i)$ is a p-dimensional mean regression.

Before specifying our priors for each of the components in (2), we first place our formulation within the context of dynamic latent factor models.

2.1.1 Relationship to Dynamic Latent Factor Models

A standard latent factor model characterizes independent observations $\boldsymbol{y}_i$ via independent latent factors $\boldsymbol{\eta}_i$:

$$\boldsymbol{y}_i = \Lambda\boldsymbol{\eta}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\eta}_i \sim N_k(0, I_k), \quad \boldsymbol{\epsilon}_i \sim N_p(0, \Sigma_0). \tag{4}$$

Marginalizing the latent factors $\boldsymbol{\eta}_i$ yields $\boldsymbol{y}_i \sim N_p(0, \Sigma)$ with $\Sigma = \Lambda\Lambda' + \Sigma_0$. The ideas of latent factor analysis have also been applied to the time-series domain by assuming a latent factor *process*. Such *dynamic latent factor models* have a rich history. Typically, the dynamics of the latent factors are assumed to follow a simple Markov evolution with a time-invariant parameterization (West, 2003; Lopes et al., 2008):

$$\begin{aligned} \boldsymbol{\eta}_i &= \Gamma\boldsymbol{\eta}_{i-1} + \boldsymbol{\nu}_i, \quad \boldsymbol{\nu}_i \sim N_k(0, I_k) \\ \boldsymbol{y}_i &= \Lambda\boldsymbol{\eta}_i + \boldsymbol{\epsilon}_i, \quad \boldsymbol{\epsilon}_i \sim N_p(0, \Sigma_0), \end{aligned} \tag{5}$$

where $\Gamma \in \Re^{k\times k}$ is the dynamic matrix for the latent factor evolution. Assuming a stationary process on $\boldsymbol{\eta}_i$, then $\boldsymbol{y}_i \sim N_p(0, \Sigma)$ with $\Sigma = \Lambda\Sigma_\eta\Lambda' + \Sigma_0$. Here, Σ_η denotes the marginal covariance of $\boldsymbol{\eta}_i$. If we restrict our attention to cases in which $\boldsymbol{x}_i$ is a discrete time index, as in our flu application, then our proposed model of (2) can be related to the class of dynamic latent factor models as follows. The latent factor evolution is governed by $\boldsymbol{\psi}$ rather than a standard linear autoregression: $\boldsymbol{\eta}_i = \boldsymbol{\psi}(x_i) + \boldsymbol{\nu}_i$, $\boldsymbol{\nu}_i \sim N_k(0, I_k)$. In Section 2.2, we specify $\boldsymbol{\psi}$ via Gaussian processes, providing a nonparametric evolution in *continuous* time. Importantly, the factor loadings matrix $\Lambda(x)$ also evolves in time: $\boldsymbol{y}_i = \Lambda(x_i)\boldsymbol{\eta}_i + \boldsymbol{\epsilon}_i$ with conditional covariance $\Sigma(x) = \Lambda(x)\Lambda(x)' + \Sigma_0$. Again, this analogy relies on assuming $\boldsymbol{x}$ represents time. The formulation of (2) is proposed for general predictors $\boldsymbol{x} \in \mathcal{X}$.

2.2 Prior specification

To capture the evolution of $\boldsymbol{\psi}(\boldsymbol{x})$ and $\Lambda(\boldsymbol{x})$, we use Gaussian processes as a set of basis functions. We first briefly review Gaussian processes and then describe how this basis is used in our model.

2.2.1 Gaussian Processes

A Gaussian process provides a distribution over real-valued functions $f : \mathcal{X} \to \Re$, with the property that the function evaluated at any finite collection of points is jointly Gaussian. The Gaussian process, denoted $\mathrm{GP}(m, c)$, is uniquely defined by its *mean function* m and *covariance kernel* c. In particular, $f \sim \mathrm{GP}(m, c)$ if and only if for all n and $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$,

$$p(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)) \sim N_n(\boldsymbol{\mu}, K), \tag{6}$$

with $\boldsymbol{\mu} = [m(\boldsymbol{x}_1), \ldots, m(\boldsymbol{x}_n)]$ and K the $n \times n$ *Gram matrix* with entries $K_{ij} = c(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The properties (e.g., continuity, smoothness, periodicity, etc.) of functions drawn from a

given Gaussian process are determined by the covariance kernel. One example leading to smooth functions is the squared exponential, or *Gaussian*, kernel:

$$c(\boldsymbol{x}, \boldsymbol{x}') = d \exp(-\kappa ||\boldsymbol{x} - \boldsymbol{x}'||_2^2), \tag{7}$$

where d is a *scale* hyperparameter and κ the *bandwidth*, which determines the extent of the correlation in f over $\mathcal{X}$. See Rasmussen and Williams (2006) for further details.

2.2.2 Latent Factor Mean Process

Letting $\boldsymbol{\psi}(\boldsymbol{x}) = \{\psi_1(\boldsymbol{x}), \ldots, \psi_k(\boldsymbol{x})\}$, we specify independent Gaussian process priors for each ψ_h as a convenient and flexible choice. In particular, $\psi_h \sim \text{GP}(0, c_\psi)$ with $c_\psi(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\kappa_\psi ||\boldsymbol{x} - \boldsymbol{x}'||_2^2)$ a squared exponential covariance kernel. We assume unit variance for reasons of identifiability seen in (3) through the multiplication of the latent factors with $\Lambda(\boldsymbol{x})$.

2.2.3 Idiosyncratic Noise

We choose independent inverse gamma priors for the diagonal elements of Σ_0 by letting $\sigma_j^{-2} \sim \text{Ga}(a_\sigma, b_\sigma)$. The off-diagonal elements are deterministically set to zero.

2.2.4 Factor Matrix Process

Specifying a prior for $\Lambda(\boldsymbol{x})$ is more challenging, as naive approaches, such as independent Gaussian process priors for each element of the $p \times k$ matrix, may have poor performance in large p application domains even for small k. Likewise, the computational demands for considering $p \times k$ Gaussian processes can be prohibitive depending on the choice of p, k, n (see Section 3). Instead, we take the factor loadings to be a weighted combination of a much smaller set of basis elements ξ_{lh},

$$\Lambda(\boldsymbol{x}) = \Theta \xi(\boldsymbol{x}), \quad \Theta \in \Re^{p \times L}, \quad \xi(\boldsymbol{x}) = \{\xi_{lh}(\boldsymbol{x}), l = 1, \ldots, L, h = 1, \ldots, k\}, \tag{8}$$

where Θ is a matrix of coefficients that maps the $L \times k$ array of basis functions $\xi(\boldsymbol{x})$ to the predictor-dependent loadings matrix $\Lambda(\boldsymbol{x})$. Typically, $k \ll p$ and $L \ll p$. Again, k defines the factor dimension (i.e., assumed subspace that captures the statistical variability) whereas L controls the size of the basis for any fixed choice of k. We once again choose independent Gaussian process priors $\xi_{lh} \sim \text{GP}(0, c)$, with $c(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\kappa ||\boldsymbol{x} - \boldsymbol{x}'||_2^2)$ a squared exponential covariance kernel. The choice of unit variance Gaussian processes again arises for reasons of identifiability, but now with the multiplication with Θ.

To allow for an adaptive choice of the basis size, we in theory let $L \to \infty$ and employ the shrinkage prior of Bhattacharya and Dunson (2011) for Θ,

$$\theta_{jl} \sim \text{N}(0, \phi_{jl}^{-1} \tau_l^{-1}), \quad \phi_{jl} \sim \text{Ga}(\gamma/2, \gamma/2), \quad \tau_l = \prod_{h=1}^{l} \delta_h, \tag{9}$$

with ϕ_{jl} a local precision specific in element j, l, and τ_l a column-specific multiplier, which is assigned a multiplicative gamma process prior to favor increasing shrinkage of elements in later columns by letting $\delta_1 \sim \text{Ga}(a_1, 1)$ and $\delta_h \sim \text{Ga}(a_2, 1)$, $h \geq 2$, with $a_2 > 1$. If a column

of Θ is shrunk towards zero, the corresponding row of the basis $\xi(\boldsymbol{x})$ has insignificant effect in defining $\{\boldsymbol{\mu}(\boldsymbol{x}), \Sigma(\boldsymbol{x})\}$. Our chosen prior specification increasingly shrinks columns with column index, effectively truncating Θ. That is, despite an arbitrarily large L, the effective dimension of the basis is much smaller, providing our desired dimensionality reduction. In practice, of course, a finite truncation $\bar{L}$ is chosen. See Appendix E for a discussion on other possible decompositions of $\Lambda(\boldsymbol{x})$ and prior specifications.

At any point $\boldsymbol{x} \in \mathcal{X}$, the different $\xi_{lk}(\boldsymbol{x})$s are independently Gaussian distributed, and hence $\xi(\boldsymbol{x})\xi(\boldsymbol{x})'$ is Wishart distributed. Conditioned on Θ, $\Theta\xi(\boldsymbol{x})\xi(\boldsymbol{x})'\Theta'$ is also Wishart distributed and, as $\boldsymbol{x}$ varies, follows the matrix-variate Wishart process of Gelfand et al. (2004) with Wilson and Ghahramani (2011) recently considering a related specification. However, these alternative specifications do not have the dimensionality reduction structure, which is key to the performance of our approach in moderate to high dimensions. Furthermore, they do not provide the theoretical statements of large support we show in Section 2.4 nor a framework for coping with missing data. Marginalizing over the prior for Θ, one obtains a type of adaptively scaled mixture of Wishart processes that has fundamentally different behavior than the Wishart. Our prior is also somewhat related to the spatial dynamic factor model of Lopes et al. (2008), though their focus is on space-time dependence in univariate observations. Finally, following our early technical report version of this paper (Fox and Dunson, 2011), Fosdick and Hoff (2014) examine factor-structured separable covariance models for general M-array data. Considering a 2-array of space and time, the model assumes a spatial structure $\Lambda_s\Lambda_s' + \Sigma_{0,s}$ and temporal structure $\Lambda_t\Lambda_t' + \Sigma_{0,t}$. That is, the model is low rank in both space and time. In contrast, our covariance decomposition at any predictor $\boldsymbol{x}$ assumes the factor structure $\Lambda(\boldsymbol{x})\Lambda(\boldsymbol{x})' + \Lambda_0$ for the response vector (e.g., indexed by spatial location); however, the dependence between predictors $\boldsymbol{x}$ and $\boldsymbol{x}'$ (e.g., across time) is described via a stochastic *process*.

2.2.5 Identifiability

The factorizations for $\boldsymbol{\mu}(\boldsymbol{x})$ and $\Sigma(\boldsymbol{x})$ are not unique but instead we obtain a many-to-one specification. It is not necessary to enforce identifiability constraints, as our focus is on inducing a prior for $\boldsymbol{\mu}(\boldsymbol{x})$ and $\Sigma(\boldsymbol{x})$ that favors an effectively low-dimensional representation without constraining the possible changes in the mean and covariance with predictors beyond minimal regularity conditions.

2.3 Parsimony of Covariance Decomposition

Through the chosen covariance decomposition $\Sigma(\boldsymbol{x}) = \Theta\xi(\boldsymbol{x})\xi(\boldsymbol{x})'\Theta' + \Sigma_0$ specified in (3) and (8), we have transformed the problem of modeling $p(p+1)/2$ predictor dependent elements to one of modeling $p \times (L+1)$ non-predictor dependent elements (comprising Θ and Σ_0) plus $L \times k$ predictor dependent elements (comprising $\xi(\cdot)$). A substantial reduction in parameterization occurs when $k \ll p$ and $L \ll p$. Such an assumption is appropriate in modeling a large class of covariance regressions $\Sigma(\boldsymbol{x})$ that arise when analyzing real data.

For arbitrary Θ and $\xi(\cdot)$, this parameterization still scales poorly to large data sets. It is only through the implied regularization effect of our chosen prior specification that a parsimonious model arises (even for L large, as previously discussed). Specifically, the continuity of the latent Gaussian process basis elements $\xi_{\ell k}$ combined with the shrinkage

properties of the prior on Θ forms a flexible, adaptive hierarchical structure that borrows information and can collapse on an effectively lower dimensional structure.

Another important aspect of the covariance decomposition is the implied *transfer of knowledge* property that allows us to cope with substantial missing or corrupted data. Let x_m correspond to a point in the predictor space at which the jth response component y_{mj} is missing or corrupted. In our model, the estimates of $\Sigma_{j\cdot}(\boldsymbol{x}_m) = \Theta_{j\cdot}\xi(\boldsymbol{x}_m)\xi(\boldsymbol{x}_m)'\Theta' + \Sigma_{0j\cdot}$ are improved by the fact that (i) the rows $\Theta_{j\cdot}$ are informed by all available observations y_{ij} at predictor locations $\boldsymbol{x}_i \neq \boldsymbol{x}_m$, and (ii) the latent basis functions $\xi(\boldsymbol{x}_m)$ are informed by the available response components y_{mk}, $k \neq j$, at the predictor location $\boldsymbol{x}_m$ and at nearby locations via the continuity of the basis functions. By employing a small collection of latent basis elements with non-predictor-dependent weights, our model better copes with limited data and is more robust to corrupted values than one in which the elements of $\Sigma(\boldsymbol{x})$ are modeled independently.

2.4 Properties

Our proposed Bayesian nonparametric covariance regression framework of Section 2 yields various important theoretical properties, such as large prior support and stationarity, which we examine here.

2.4.1 Large Support

We induce a prior $\{\Sigma(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\} \sim \Pi_\Sigma$ through priors Π_ξ, Π_Θ and Π_{Σ_0} for ξ, Θ and Σ_0, respectively. In this section, we explore the properties of the induced prior Π_Σ. Most fundamentally, we establish that this prior has large support in Theorem 2. Large support implies that the prior can generate a covariance regression function $\Sigma : \mathcal{X} \to \mathcal{P}_p^+$ arbitrarily close to any continuous function $\Sigma^* : \mathcal{X} \to \mathcal{P}_p^+$, with $\mathcal{P}_p^+$ the space of $p \times p$ positive semidefinite matrices. Such a support property is the defining feature of a Bayesian nonparametric approach and cannot simply be assumed. Often, seemingly flexible models can have quite restricted support due to hidden constraints in the model and not to real prior knowledge that certain values are implausible. The proofs associated with the theoretical statements made in this section can be found in Appendix A.

We start by introducing a notion of *k-decomposability* of a covariance regression function $\Sigma(\boldsymbol{x})$.

Definition 1 $\Sigma : \mathcal{X} \to \mathcal{P}_p^+$ *is said to be* **k-*decomposable*** *if* $\Sigma(\boldsymbol{x}) = \Lambda(\boldsymbol{x})\Lambda(\boldsymbol{x})' + \Sigma_0$ *for* $\Lambda(\boldsymbol{x}) \in \Re^{p\times k}$, $\Sigma_0 \in \mathcal{X}_{\Sigma_0}$, *and for all* $\boldsymbol{x} \in \mathcal{X}$.

In Appendix A, we show that such a decomposition always exists for k sufficiently large. Now assume our model $\Sigma(\boldsymbol{x}) = \Theta\xi(\boldsymbol{x})\xi(\boldsymbol{x})'\Theta' + \Sigma_0$ with priors Π_ξ and Π_{Σ_0} as specified in Section 2.2. For Π_Θ, we aim to make our statement of prior support as general as possible and thus simply assume that Π_Θ satisfies the following two conditions. The proof that Assumptions 2.1 and 2.2 are satisfied by our shrinkage prior (9) is provided in Appendix A.

Assumption 2.1 Π_Θ *is such that* $\sum_\ell E(|\theta_{j\ell}|) < \infty$, *ensuring that the prior for* Θ *shrinks the elements towards zero fast enough as* $\ell \to \infty$.

Assumption 2.2 Π_Θ *is such that* $\Pi_\Theta\,(rank(\Theta) = p) > 0$. *That is, there is positive prior probability of* Θ *being full rank.*

Our main result on prior support now follows.

Theorem 2 *Let Π_Σ denote the induced prior on $\{\Sigma(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\}$ based on $\Pi_\xi \otimes \Pi_\Theta \otimes \Pi_{\Sigma_0}$, with Π_Θ satisfying Assumptions 2.1 and 2.2. Assume $\mathcal{X}$ is compact. Then, for all continuous functions $\Sigma^* : \mathcal{X} \to \mathcal{P}_p^+$ that are k^*-decomposable and for all $\epsilon > 0$ and $k \geq k^*$,*

$$\Pi_\Sigma \left(\sup_{\boldsymbol{x} \in \mathcal{X}} ||\Sigma(\boldsymbol{x}) - \Sigma^*(\boldsymbol{x})||_2 < \epsilon \right) > 0.$$

Informally, Theorem 2 states that there is positive prior probability of random covariance regressions $\Sigma(\boldsymbol{x})$ that stay within an L_2 ϵ-ball of any specified continuous $\Sigma^*(\boldsymbol{x})$ everywhere over the predictor space $\mathcal{X}$. Intuitively, the support on continuous covariance functions $\Sigma^*(\boldsymbol{x})$ arises from the continuity of the Gaussian process basis functions. However, since we are mixing over infinitely many such basis functions, we need the mixing weights specified by Θ to tend towards zero, and to do so "fast enough"—this is where Assumption 2.1 becomes important. We also rely on the large support of Π_Σ at any point $\boldsymbol{x}_0 \in \mathcal{X}$. Combining the large support of the Wishart distribution for $\Theta\xi(\boldsymbol{x}_0)\xi(\boldsymbol{x}_0)'\Theta'$ (Θ fixed) with that of the gamma distribution on the inverse elements of Σ_0 provides the desired large support of the induced prior Π_Σ at each predictor location $\boldsymbol{x}_0$.

Remark 3 *Our theory holds for $L \to \infty$ (an arbitrarily large set of latent basis functions); however, our large support result only relies on choosing $L = p$. Assuming Σ^* is k^*-decomposable with $k^* \ll p$ such that we can select $k \ll p$, this still represents a reduction in parameterization relative to a full model necessitating $p \times p$ basis functions. The reliance on $L = p$ is to be able to capture* any *k^*-decomposable Σ^*. We can further introduce a concept of* **L-decomposability** *where $\Sigma(\boldsymbol{x}) = \Lambda(\boldsymbol{x})\Lambda(\boldsymbol{x})' + \Sigma_0$ with $\Lambda(\boldsymbol{x}) = \Theta\xi(\boldsymbol{x})$ for $\Theta \in \Re^{p \times L}$ and for all $\boldsymbol{x} \in \mathcal{X}$, which represents a second factor assumption. Assuming $L \ll p$ is likely reasonable for large p. Then for a (k^*, L^*)-decomposable Σ^*, and choosing $k > k^*$ and $L > L^*$ (rather than relying on $L = p$), the theory of large support follows straightforwardly.*

Even when selecting $L = p$, due to our shrinkage prior for Θ of Section 2.2.4, we find in practice that many columns tend to be shrunk to zero *a posteriori* such that choosing a truncation $\bar{L} \ll p$ suffices. See Sections 4 and 5.2.4.

2.4.2 Moments and Stationarity

To better understand the relationship between our hyperparameter settings and resulting covariance regressions, it is useful to analyze the moments of $\{\Sigma(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\} \sim \Pi_\Sigma$. Lemma 4 provides the prior mean and Lemma 5 the covariance between elements of $\Sigma(\boldsymbol{x})$ and $\Sigma(\boldsymbol{x}')$. As the distance between $\boldsymbol{x}$ and $\boldsymbol{x}'$ increases, the correlation decreases at a rate depending on the Gaussian process covariance kernel $c(\boldsymbol{x}, \boldsymbol{x}')$.

Lemma 4 *Let μ_σ denote the mean of σ_j^2, $j = 1, \ldots, p$. Then,*

$$E[\Sigma(\boldsymbol{x})] = diag\left(k \sum_\ell \phi_{1\ell}^{-1} \tau_\ell^{-1} + \mu_\sigma, \ldots, k \sum_\ell \phi_{p\ell}^{-1} \tau_\ell^{-1} + \mu_\sigma \right).$$

That is, the expected covariance at $\boldsymbol{x}$ is diagonal with expected variance elements depending on our latent dimension k.

Lemma 5 *Let σ^2_σ denote the variance of σ^2_j, $j = 1, \ldots, p$. Then,*

$$cov(\Sigma_{ij}(\boldsymbol{x}), \Sigma_{ij}(\boldsymbol{x}')) = \begin{cases} k\, c(\boldsymbol{x}, \boldsymbol{x}') \left(5 \sum_\ell \phi_{i\ell}^{-2} \tau_\ell^{-2} + (\sum_\ell \phi_{i\ell}^{-1} \tau_\ell^{-1})^2\right) + \sigma^2_\sigma & i = j, \\ k\, c(\boldsymbol{x}, \boldsymbol{x}') \left(\sum_\ell \phi_{i\ell}^{-1} \phi_{j\ell}^{-1} \tau_\ell^{-2} + \sum_\ell \phi_{i\ell}^{-1} \tau_\ell^{-1} \sum_{\ell'} \phi_{j\ell'}^{-1} \tau_{\ell'}^{-1}\right) & i \neq j. \end{cases} \tag{10}$$

For $\Sigma_{ij}(\boldsymbol{x})$ and $\Sigma_{uv}(\boldsymbol{x}')$ with $i \neq u$ or $j \neq v$, $cov(\Sigma_{ij}(\boldsymbol{x}), \Sigma_{uv}(\boldsymbol{x}')) = 0$.

Here, we see how our Gaussian process covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$ controls the dependence over $\mathcal{X}$ in an interpretable, linear fashion.

From Lemma 5, the autocorrelation $ACF(\boldsymbol{x}) = \text{corr}(\Sigma_{ij}(0), \Sigma_{ij}(\boldsymbol{x}))$ is simply specified by $c(0, \boldsymbol{x})$. When we choose a Gaussian process kernel $c(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\kappa ||\boldsymbol{x} - \boldsymbol{x}'||_2^2)$, we have

$$ACF(\boldsymbol{x}) = \exp(-\kappa ||\boldsymbol{x}||_2^2). \tag{11}$$

Thus, the length-scale parameter κ directly determines the shape of the autocorrelation function. This property aids in the selection of κ via a data-driven mechanism (i.e., a quasi-empirical Bayes approach), as outlined in Appendix C. One can also consider selecting κ using methods akin to those proposed by Higdon et al. (2008); Paulo (2005).

Finally, Lemma 6 shows that the stochastic process Σ has stationarity properties, an often desirable property of a covariance process specification since Σ itself captures heteroscedasticity in the observation process.

Lemma 6 *The process $\{\Sigma(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}\} \sim \Pi_\Sigma$ is first-order stationary in that for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathcal{X}$, $\Pi_\Sigma(\Sigma(\boldsymbol{x})) = \Pi_\Sigma(\Sigma(\boldsymbol{x}'))$. Furthermore, assuming a stationary covariance function $c(\boldsymbol{x}, \boldsymbol{x}')$, the process is wide sense stationary: $cov(\Sigma_{ij}(\boldsymbol{x}), \Sigma_{uv}(\boldsymbol{x}'))$ solely depends upon $||\boldsymbol{x} - \boldsymbol{x}'||$.*

3. Posterior Computation via Gibbs Sampling

Based on a fixed truncation level $\bar{L}$ and a latent factor dimension $\bar{k}$, we propose a Gibbs sampler for posterior computation. For the model of Section 2, the full joint probability is given by $p_{obs} \cdot p_{params} \cdot p_{hypers}$ where

$$\begin{aligned} p_{obs} &= \prod_{i=1}^n \left[p(\boldsymbol{y}_i \mid \Theta, \xi, \boldsymbol{\eta}_i, \Sigma_0) \prod_{k=1}^{\bar{k}} p(\eta_{ik} \mid \psi_k) \right] \quad p_{hypers} = \prod_{\ell=1}^{\bar{L}} \left[p(\tau_\ell) \prod_{j=1}^p p(\phi_{j\ell}) \right] \\ p_{params} &= \prod_{k=1}^{\bar{k}} \left[p(\psi_k) \prod_{\ell=1}^{\bar{L}} p(\xi_{\ell k}) \right] \prod_{j=1}^p \left[p(\sigma_j^2) \prod_{\ell=1}^{\bar{L}} p(\theta_{j\ell} \mid \phi_{j\ell}, \tau_\ell) \right] \end{aligned} \tag{12}$$

The resulting sampler is outlined in Steps 1-5 below. Step 1 is derived in Appendix B. In this section, we equivalently represent the latent factor process of (2) as $\boldsymbol{\eta}_i = \boldsymbol{\psi}(\boldsymbol{x}_i) + \boldsymbol{\nu}_i$, with $\boldsymbol{\nu}_i \sim \text{N}_k(0, I_k)$.

Step 1. Update each basis function $\xi_{\ell m}$ from the conditional posterior given $\{y_i\}$, Θ, $\{\eta_i\}$, Σ_0. We can rewrite the observation model for the jth component of the ith response as $y_{ij} = \sum_{m=1}^{\bar{k}} \eta_{im} \sum_{\ell=1}^{\bar{L}} \theta_{j\ell}\xi_{\ell m}(x_i) + \epsilon_{ij}$. Conditioning on $\xi^{-\ell m} = \{\xi_{rs}, r \neq \ell, s \neq m\}$,

$$\begin{pmatrix} \xi_{\ell m}(\boldsymbol{x}_1) \\ \vdots \\ \xi_{\ell m}(\boldsymbol{x}_n) \end{pmatrix} \mid \{\boldsymbol{y}_i\}, \{\boldsymbol{\eta}_i\}, \Theta, \xi^{-\ell m}, \Sigma_0 \sim \mathrm{N}_n \left(\tilde{\Sigma}_\xi \begin{pmatrix} \eta_{1m} \sum_{j=1}^p \theta_{j\ell}\sigma_j^{-2}\tilde{y}_{1j} \\ \vdots \\ \eta_{nm} \sum_{j=1}^p \theta_{j\ell}\sigma_j^{-2}\tilde{y}_{nj} \end{pmatrix}, \tilde{\Sigma}_\xi \right),$$

where $\tilde{y}_{ij} = y_{ij} - \sum_{(r,s)\neq(\ell,m)} \theta_{jr}\xi_{rs}(\boldsymbol{x}_i)$ and, taking K to be the Gaussian process covariance matrix with $K_{ij} = c(\boldsymbol{x}_i, \boldsymbol{x}_j)$,

$$\tilde{\Sigma}_\xi^{-1} = K^{-1} + \mathrm{diag}\left(\eta_{1m}^2 \sum_{j=1}^p \theta_{j\ell}^2\sigma_j^{-2}, \ldots, \eta_{nm}^2 \sum_{j=1}^p \theta_{j\ell}^2\sigma_j^{-2} \right).$$

Step 2. Sample each latent factor mean function ψ_l. Letting $\Omega_i = \Theta\xi(\boldsymbol{x}_i)$, we have $\boldsymbol{y}_i = \Omega_i\boldsymbol{\psi}(\boldsymbol{x}_i) + \Omega_i\boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i$. Marginalizing out $\boldsymbol{\nu}_i$, $\boldsymbol{y}_i = \Omega_i\boldsymbol{\psi}(\boldsymbol{x}_i) + \boldsymbol{\omega}_i$ with $\boldsymbol{\omega}_i \sim \mathrm{N}(0, \tilde{\Sigma}_i = \Omega_i\Omega_i' + \Sigma_0)$. Assuming $\psi_\ell \sim \mathrm{GP}(0, c)$, the posterior of ψ_ℓ follows analogously to that of $\xi_{\ell m}$ resulting in

$$\begin{pmatrix} \psi_l(\boldsymbol{x}_1) \\ \vdots \\ \psi_l(\boldsymbol{x}_n) \end{pmatrix} \mid \{\boldsymbol{y}_i\}, \{\boldsymbol{\eta}_i\}, \boldsymbol{\psi}^{-l}, \Theta, \xi, \Sigma_0 \sim \mathrm{N}_n \left(\tilde{\Sigma}_\psi \begin{pmatrix} \Omega_{1l}'\tilde{\Sigma}_1^{-1}\tilde{\boldsymbol{y}}_1^{-l} \\ \vdots \\ \Omega_{nl}'\tilde{\Sigma}_n^{-1}\tilde{\boldsymbol{y}}_n^{-l} \end{pmatrix}, \tilde{\Sigma}_\psi \right),$$

where $\tilde{\boldsymbol{y}}_i^{-l} = \boldsymbol{y}_i - \sum_{(r\neq l)} \Omega_{ir}\boldsymbol{\psi}_r(\boldsymbol{x}_i)$, Ω_{il} is the lth column vector of Ω_i, and

$$\tilde{\Sigma}_\psi^{-1} = K^{-1} + \mathrm{diag}\left(\Omega_{1l}'\tilde{\Sigma}_1^{-1}\Omega_{1l}, \ldots, \Omega_{nl}'\tilde{\Sigma}_n^{-1}\Omega_{nl} \right).$$

Step 3. Sample $\boldsymbol{\nu}_i$. Defining $\tilde{\boldsymbol{y}}_i = \boldsymbol{y}_i - \Omega_i\boldsymbol{\psi}(\boldsymbol{x}_i)$ such that $\tilde{\boldsymbol{y}}_i = \Omega_i\boldsymbol{\nu}_i + \boldsymbol{\epsilon}_i$, we draw $\boldsymbol{\nu}_i$ given $\tilde{\boldsymbol{y}}_i, \boldsymbol{\psi}(\boldsymbol{x}_i), \Theta, \xi(\boldsymbol{x}_i), \Sigma_0$ from the conditional posterior,

$$\mathrm{N}_{\bar{k}} \left(\left\{ I + \xi(\boldsymbol{x}_i)'\Theta'\Sigma_0^{-1}\Theta\xi(\boldsymbol{x}_i) \right\}^{-1} \xi(\boldsymbol{x}_i)'\Theta'\Sigma_0^{-1}\tilde{\boldsymbol{y}}_i, \left\{ I + \xi(\boldsymbol{x}_i)'\Theta'\Sigma_0^{-1}\Theta\xi(\boldsymbol{x}_i) \right\}^{-1} \right).$$

Step 4. Sample σ_j^2. Letting $\theta_{j\cdot}$ denote the jth row vector of Θ, we draw

$$\sigma_j^{-2} \mid \{\boldsymbol{y}_i\}, \{\boldsymbol{\eta}_i\}, \Theta, \xi \sim \mathrm{Ga}\left(a_\sigma + \frac{n}{2}, b_\sigma + \frac{1}{2}\sum_{i=1}^n (y_{ij} - \theta_{j\cdot}\xi(\boldsymbol{x}_i)\boldsymbol{\eta}_i)^2 \right).$$

Step 5. Sample $\theta_{j\cdot}$. The conditional posterior on the row vectors of Θ is

$$\theta_{j\cdot} \mid \{\boldsymbol{y}_i\}, \{\boldsymbol{\eta}_i\}, \xi, \phi, \tau \sim \mathrm{N}_{\bar{L}} \left(\tilde{\Sigma}_\theta\tilde{\boldsymbol{\eta}}'\sigma_j^{-2}(y_{1j}, \ldots, y_{nj})', \tilde{\Sigma}_\theta \right),$$

where $\tilde{\boldsymbol{\eta}} = \{\xi(\boldsymbol{x}_1)\boldsymbol{\eta}_1, \ldots, \xi(\boldsymbol{x}_n)\boldsymbol{\eta}_n\}'$ and $\tilde{\Sigma}_\theta^{-1} = \sigma_j^{-2}\tilde{\boldsymbol{\eta}}'\tilde{\boldsymbol{\eta}} + \mathrm{diag}(\phi_{j1}\tau_1, \ldots, \phi_{j\bar{L}}\tau_{\bar{L}})$.

Step 6. Finally, for the hyperparameters in the shrinkage prior for Θ, we have

$$\phi_{jl} \mid \theta_{jl}, \tau_l \sim \mathrm{Ga}\left(2, \frac{\gamma + \tau_l\theta_{jl}^2}{2} \right)$$

$$\delta_1 \mid \Theta, \tau^{(-1)} \sim \text{Ga}\left(a_1 + \frac{p\bar{L}}{2}, 1 + \frac{1}{2}\sum_{l=1}^{\bar{L}} \tau_l^{(-1)} \sum_{j=1}^{p} \phi_{j\ell}\theta_{jl}^2\right)$$

$$\delta_h \mid \Theta, \tau^{(-h)} \sim \text{Ga}\left(a_2 + \frac{p(\bar{L}-h+1)}{2}, 1 + \frac{1}{2}\sum_{l=1}^{\bar{L}} \tau_l^{(-h)} \sum_{j=1}^{p} \phi_{jl}\theta_{jl}^2\right),$$

where $\tau_l^{(-h)} = \prod_{t=1,t\neq h}^{l} \delta_t$ for $h = 1, \ldots, p$.

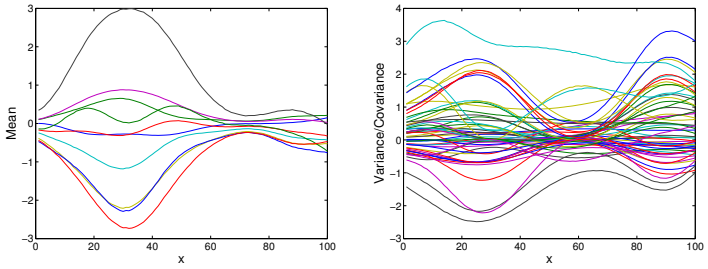
Each of the above steps is straightforward to implement involving sampling from standard distributions. We have observed good rates of convergence and mixing in our considered applications (see Section 5). As with other models involving Gaussian processes, computational bottlenecks can arise as n increases due to $O(n^3)$ matrix computation. Standard computational approaches can be used for dealing with this problem, as discussed in Section 6. We find inferences to be somewhat robust to the Gaussian process covariance parameter κ due the quadratic mixing over the basis functions. In the applications described below, we estimate κ from the data as an empirical Bayes approach, with details in Appendix C.

4. Simulation Example

We assess the performance of the proposed approach in terms of both covariance estimation and predictive performance. In Case 1 we simulated from the proposed model, while in Case 2 we simulated from a parametric model. In Case 1, we let $\mathcal{X} = \{1, \ldots, 100\}$, $p = 10$, $L = 5$, $k = 4$, $a_1 = a_2 = 10$, $\gamma = 3$, $a_\sigma = 1$, $b_\sigma = 0.1$ and $\kappa_\psi = \kappa = 10$ in the Gaussian process after scaling $\mathcal{X}$ to $(0, 1]$ with an additional nugget of $1e^{-5}I_n$ added to K. Figure 1 displays the resulting values of the elements of $\boldsymbol{\mu}(x)$ and $\Sigma(x)$. For inference, we use truncation levels $\bar{k} = \bar{L} = 10$, which we found to be sufficiently large from the fact that the last few columns of the posterior samples of Θ were consistently shrunk close to 0. We set $a_1 = a_2 = 2$, $\gamma = 3$, and placed a $\text{Ga}(1, 0.1)$ prior on the precision parameters σ_j^{-2}. The length-scale parameter κ was set from the data according to the heuristic described in Appendix C, and was determined to be 10 (after rounding). Details on initialization are available in Appendix D. We simulated 10,000 Gibbs iterations, discarded the first 5,000 and saved every 10th iteration.

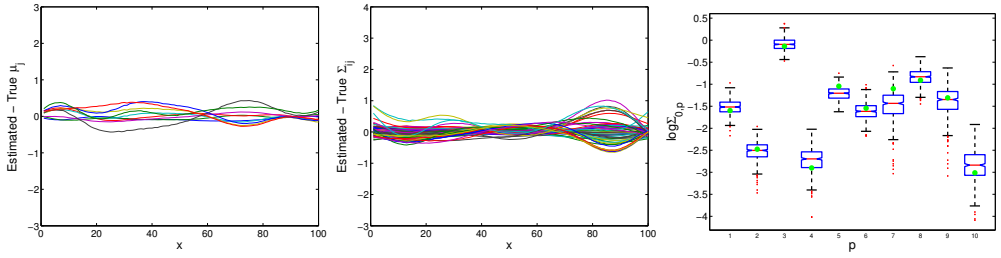
The residuals between the true and posterior mean over all components are displayed in Figure 2(a) and (b). Figure 2(c) compares the posterior samples of the elements σ_j^2 of the residual covariance Σ_0 to the true values. In Figure 3 we display a select set of plots of the true and posterior mean of components of $\boldsymbol{\mu}(x)$ and $\Sigma(x)$, along with the 95% highest posterior density intervals computed pointwise. From Figures 2 and 3, we see that we are clearly able to capture heteroscedasticity in combination with a nonparametric mean regression. The true values of the mean and covariance components are all contained within the 95% highest posterior density intervals, with these intervals typically narrow.

For the same simulated data set, we assessed predictive performance compared to homoscedastic models $\boldsymbol{y} \sim \text{N}_p(\boldsymbol{\mu}(x), \Sigma)$, with $\mu_j(x)$ either arising as independent $\text{GP}(0, c)$ draws or through a latent factor regression model with $\boldsymbol{\mu}(x) = \Theta\xi(x)\boldsymbol{\psi}(x)$ just as in the heteroscedastic formulation; in both cases, Σ was assigned an inverse-Wishart prior. By



(a) (b)

Figure 1: Plot of each component of the (a) true mean vector $\boldsymbol{\mu}(x)$ and (b) true covariance matrix $\Sigma(x)$ over $\mathcal{X} = \{1, \ldots, 100\}$.



(a) (b) (c)

Figure 2: Differences between each component of the true and posterior mean of (a) the mean $\boldsymbol{\mu}(x)$, and (b) covariance $\Sigma(x)$. The y-axis scale matches that of Figure 1. (c) Box plot of posterior samples of $\log(\sigma_j^2)$ for $j = 1, \ldots, p$ compared to the true value (green).

comparing to this latter homoscedastic model, we can directly analyze the benefits of our heteroscedastic model since both share exactly the same mean regression formulation. To generate a hold out sample, we removed 48 of the 1,000 observations by deleting observations y_{ij} with probability p_i, where p_i was chosen to vary with x_i to slightly favor removal in regions with more concentrated conditional response distributions.

We first calculated the average Kullback-Leibler divergence between the estimated and true predictive distribution of the missing elements y_{ij} given the observed elements of y_i. The average values were 0.341, 0.291 and 0.122 for the homoscedastic mean regression, homoscedastic latent factor mean regression and heteroscedastic latent factor mean regression, respectively. In this scenario, the missing observations y_{ij} are imputed as an additional step in the MCMC computations.[1] The results clearly indicate that our Bayesian nonparametric covariance regression model provides more accurate predictive distributions. We also observed improvements in estimating the mean $\boldsymbol{\mu}(x)$ for the heteroscedastic approach.

1. Note that it is not necessary to impute the missing y_{ij} within our proposed Bayesian covariance regression model because of the conditional independencies at each Gibbs step. In Section 5, we simply sample based only on actual observations. Here, however, we impute in order to directly compare our performance to the homoscedastic models.

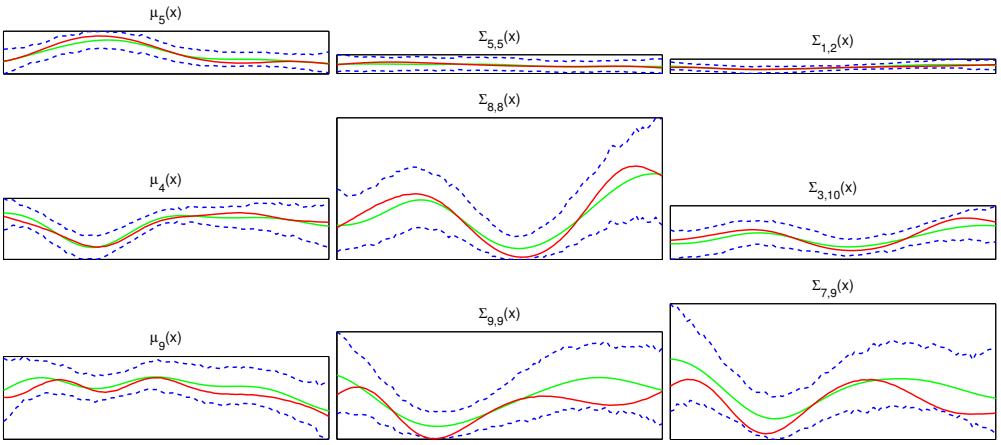


Figure 3: Plots of truth (red) and posterior mean (green) for select components of the mean $\mu_p(x)$ (*left*), variances $\Sigma_{pp}(x)$ (*middle*), and covariances $\Sigma_{pq}(x)$ (*right*). The point-wise 95% highest posterior density intervals are shown in blue. The top row represents the component with the lowest $L2$ error between the truth and posterior mean. Likewise, the middle row represents median $L2$ error and the bottom row the worst $L2$ error. The size of the box indicates the relative magnitudes of each component.

In Case 2, we generated 30 replicates from a 30-dimensional parametric heteroscedastic model with $y \sim \mathrm{N}_p(0, \Sigma(x))$ and $\mathcal{X} = \{1, \ldots, 500\}$. To generate $\Sigma(x)$, we chose a set of 5 evenly spaced knots x_k and generated $S(x_k) \sim \mathrm{N}(0, \Sigma_s)$, with $\Sigma_s = \sum_{j=1}^{30} s_j s_j'$ and $s_j \sim \mathrm{N}((-29, -27, \ldots, 27, 29)', I_{30})$. The covariance is constructed as $\Sigma(x) = \alpha \tilde{S}(x)\tilde{S}(x)' + \Sigma_0$, $x = 1, \ldots, 500$, where $\tilde{S}(x)$ is a spline fit to the $S(x_k)$ and Σ_0 is a diagonal matrix with a $\mathrm{N}(0, 1)$ truncated to be positive on its diagonal elements. The constant α is chosen to scale the maximum value of $\alpha \tilde{S}(x)\tilde{S}(x)'$ to 1.

Our hyperparameters and initialization scheme are as in Case 1, but we use truncation levels $\bar{k} = \bar{L} = 5$ based on an initial analysis with $\bar{k} = \bar{L} = 17$. A posterior mean estimate of $\Sigma(x)$ is displayed in Figure 4(c). Compare to the true $\Sigma(x)$ shown in Figure 4(a). Figure 4(b) shows the mean and 95% highest posterior density intervals of the log Frobenius norm $\log ||\Sigma^{(\tau,m)}(x) - \Sigma(x)||_2$ over Gibbs iterations τ and replicates $m = 1, \ldots, 30$. The average (un-logged) norm error over $\mathcal{X}$ is around 3, which is equivalent to each element of the inferred $\Sigma^{(\tau,m)}(x)$ deviating from the true $\Sigma(x)$ by 0.1. Since the covariance elements are approximately in the range of $[-1, 1]$ and the variances in $[0, 3]$, these values indicate good estimation performance. We compare to a Wishart matrix discounting approach of Prado and West (2010), which is commonly used in stochastic volatility modeling. Details on our implementation are included in Appendix F. From Figures 4(b) and (d), Wishart discounting has substantially worse performance, with estimation error particularly large at high xs due to accumulation of errors in forward filtering.

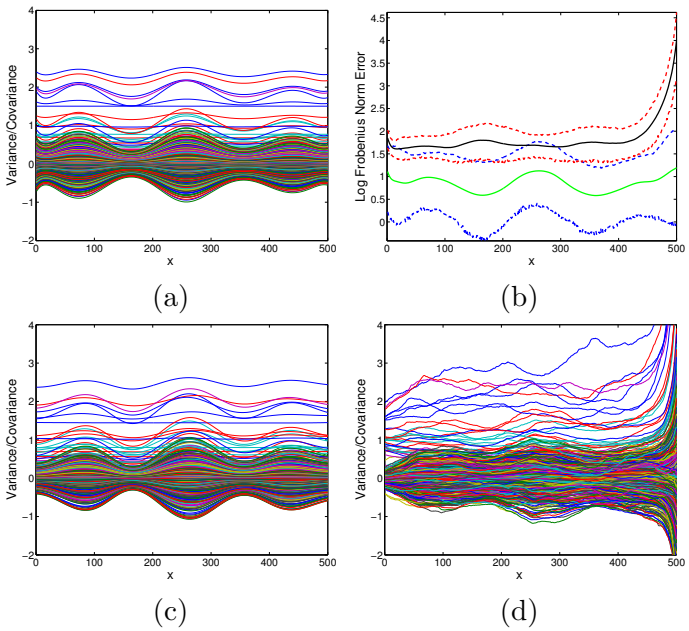


Figure 4: (a) Plot of each component of the true $\Sigma(x)$ over $\mathcal{X} = \{1, \ldots, 500\}$; (c) corresponding posterior means for our proposed approach; and (d) results for a Wishart discounting method, with (c)–(d) based on a single simulation replicate. (b) Mean and 95% highest posterior density intervals of the log Frobenius norm, $\log ||\Sigma^{(\tau,m)}(x) - \Sigma(x)||_2$, for the proposed approach (blue and green) and Wishart discounting (red and black). Results are aggregated over 100 posterior samples and replicates $m = 1, \ldots, 30$.

5. Analysis of Spatio-temporal Trends in Flu

We now turn to our analysis of the Google Flu Trends data, described in detail in Section 5.1. Our focus is on applying our Bayesian nonparametric covariance regression model to capture the heteroscedasticity noted in the exploratory analysis of Appendix G. We also examine how our modeling approach is robust to (i) inaccuracies in the mean model, (ii) missing data, and (iii) outlying estimates.

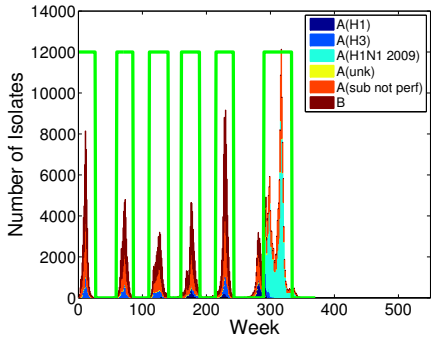
Surveillance of influenza has been of growing interest following a series of pandemic scares (e.g., SARS and avian flu) and the 2009 H1N1 ("swine flu") pandemic. Although influenza pandemics have a long history, a convergence of factors—such as the rapid rate by which geographically distant cases of influenza can spread worldwide—have increased the current public interest in influenza surveillance. A number of papers have recently analyzed the temporal (Martínez-Beneito et al., 2008) and spatio-temporal dynamics of influenza transmission (Stark et al., 2012; Dukić et al., 2012; Hooten et al., 2010; Sakai et al., 2004; Viboud et al., 2004; Mugglin et al., 2002). These approaches focus on data from a modest number of locations, and make restrictive assumptions about the spatial dependence structure, which itself may evolve temporally. Our focus is on addressing these limitations.

For example, Dukić et al. (2012) also examine portions of the Google Flu Trends data, but with the goal of on-line tracking of influenza rates on either a national, state, or regional level. Specifically, they employ a state-space model with particle learning. Our goal differs considerably. We aim to jointly analyze the full 183-dimensional data, as opposed to univariate modeling. Through such joint modeling, we can uncover important spatial dependencies lost when analyzing components of the data individually. Such spatial information can be key in predicting influenza rates based on partial observations from select regions or in retrospectively imputing missing data. Additionally, the inherent redundancy and borrowing of information across locations provided by our model should lead to robustness to inaccuracies of flu estimates caused by malicious attacks to the Google infrastructure or unaccounted for sudden spikes in web searches (see Section 5.2.3). Hooten et al. (2010) consider the temporal dynamics of the state-level Google estimates, building on a susceptible–infected–recovered (SIR) model to capture the complexities of intra- and inter-state dynamics of flu dispersal. Such a model aims to captures the intricate mechanistic structure of flu transmission, whereas our goals are focused primarily on fit using metrics such as predictive performance, with an eye towards scalability and robustness. Our exploratory data analysis of Appendix G shows that even with a very flexible and well-fit mean model, temporally changing spatial structure persist in the residuals motivating a heteroscedastic approach. In Section 4, we demonstrated that actually modeling such heteroscedasticity can improve predictive performance. Here, we show that the model of Section 2 can effectively capture such time-varying correlations in region-specific Google-estimated ILI rates, even when considering 183 regions jointly and in the presence of significant missing data.

5.1 Influenza Monitoring and Google Flu Trends

The surveillance of rates of influenza-like illness (ILI) within the United States is coordinated by the Centers for Disease Control and Prevention (CDC), which consolidates data from a large network of diagnostic laboratories, hospitals, clinics, individual healthcare providers, and state health departments. The CDC produces weekly reports (`http://www.cdc.gov/flu/weekly/`) for 10 geographic regions and a U.S. aggregate rate. A plot of the number of isolates tested positive by the WHO and NREVSS from September 28, 2003 to October 24, 2010 is shown in Figure 5 (left). From these data and the CDC weekly flu reports, we defined a set of six events (Events A-F) corresponding to the 2003-2004, 2004-2005, 2005-2006, 2006-2007, 2007-2008, and 2009-2010 flu seasons, respectively. See the specific dates listed in Figure 5 (right). The 2003-2004 flu season began earlier than normal, and coincided with a flu vaccination shortage in many states. Additionally, the CDC found that the vaccination was "not effective or had very low effectiveness" (CDC, 2004). Finally, the 2009-2010 flu season coincides with the emergence of the 2009 H1N1 ("swine flu") subtype in the U.S..

To aid in a more rapid response to influenza activity, researchers at Google devised a model in collaboration with the CDC based on Google user search queries that is meant to be predictive of CDC ILI rates, measured as cases per 100,000 physician visits (Ginsberg et al., 2008). The *Google Flu Trends* methodology was devised based on a two-stage procedure: (i) a massive variable selection procedure was used to select a subset of search queries, and (ii) using these queries as the explanatory variable, region-independent univariate linear



Event	***S***tart Date	***E***nd Date
A	Sept. 28, 2003	Mar. 21, 2004
B	Nov. 14, 2004	May 8, 2005
C	Nov. 6, 2005	May 28, 2006
D	Oct. 22, 2006	May 6, 2007
E	Nov. 4, 2007	May 11, 2008
F	Apr. 12, 2009	Feb. 7, 2010

Figure 5: *Left:* Number of isolates of Influenza A and B tested positive by the WHO and NREVSS over the period of September 29, 2003 to May 23, 2010, with Influenza A broken down into various subtypes. The green line indicates the time periods determined to be flu events. *Right:* Corresponding date ranges for flu events A-F.

models were fit to the weekly CDC ILI rates from 2003-2007. The fitted models are then used for making estimates in any region based on the ILI-related query rates from that region. A key advantage of the Google data is that the ILI rate predictions are available 1 to 2 weeks before the CDC weekly reports are published. Additionally, a user's IP address is typically connected with a specific geographic area and can thus provide information at a finer scale than the 10-regional and U.S. aggregate reporting provided by the CDC.

There has, however, been significant recent debate about the accuracy of the Google Flu Trend estimates (Butler, 2013; Lazer et al., 2014; Harris, 2014). For this paper, we take a backseat in this discussion and simply use this data set to demonstrate the potential impact of our methods in this domain. Revised Google-estimated ILI rates could likewise be used in our framework, as could other recent sources of rapid ILI estimates, e.g., using Twitter data (Lamb et al., 2013; Achrekar et al., 2012) or platforms that incorporate user-contributed reported cases (e.g., `https://flunearyou.org`). Regardless, as we demonstrate in Section 5.2.3, our formulation provides some robustness to inaccurate estimates.

5.1.1 Data Description and Key Features

We analyze the Google Flu Trends data—produced on a weekly basis—from September 28, 2003 through October 24, 2010, totaling 370 weeks. These data provide ILI estimates in 183 regions, consisting of the U.S. national level, 50 states, 10 U.S. Department of Health & Human Services surveillance regions, and 122 cities. For our modeling, we take our observation vectors $\boldsymbol{y}_i = (y_{i1}, \ldots, y_{ip})$ to be the log of the Google-estimated ILI rates in the $p = 183$ regions at week i. We denote the untransformed rates by $\boldsymbol{r}_i = (r_{i1}, \ldots, r_{ip})$. Our predictor x_i is simply a discrete time index indicating the current week ($x_i = i$, $i = 1, \ldots, 370$).

Since the Google model fits regions independently, it is not the case that city counts add to regional counts which add to state counts, and so on. That is, the dimensions of $\boldsymbol{y}_i$ are not deterministic functions of each other. There is, however, inherent redundancy (e.g., between

the estimated ILI rates for California and Los Angeles) that is naturally accommodated by a latent factor approach. Another important note is that there is substantial missing data with entire blocks of observations unavailable (as opposed to certain weeks sporadically being omitted). At the beginning of the examined time frame only 114 of the 183 regions were reporting. By the end of Year 1, there were 130 regions. These numbers increased to 173, 178, 180, and 183 by the end of Years 2, 3, 4, and 5, respectively.

5.2 Analysis via Bayesian Nonparametric Covariance Regression

We apply our Bayesian nonparametric covariance regression model as follows: $\log \boldsymbol{r}_i \sim N(\boldsymbol{\mu}(x_i), \Sigma(x_i))$. Recall that $\boldsymbol{r}_i$ simply stacks all region-specific measurements r_{ij} into a 183-dimensional vector for each week x_i. The spatial conditional correlation structure at week x_i is then captured by the covariance $\Sigma(x_i) = \Theta\xi(x_i)\xi(x_i)'\Theta' + \Sigma_0$ and the mean by $\boldsymbol{\mu}(x_i) = \Theta\xi(x_i)\boldsymbol{\psi}(x_i)$. Temporal changes are implicitly modeled through the proposed mean-covariance regression framework that allows for continuous variations in $\{\boldsymbol{\mu}(x_i), \Sigma(x_i)\}$ via our Gaussian-process-based formulation. As such, we can also examine $\{\boldsymbol{\mu}(x), \Sigma(x)\}$ for unobserved time points $x \in \mathcal{X}$ occurring between the weekly measurements.

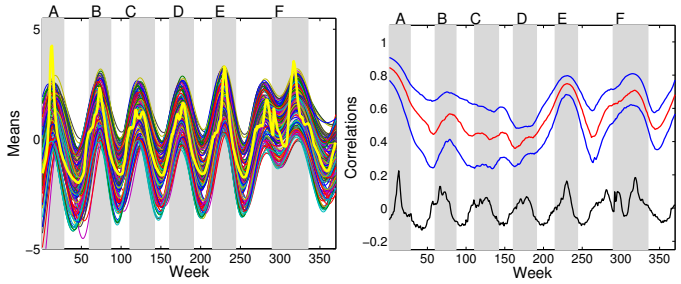
We emphasize that our model does not explicitly encode any spatial structure between the regions (comprising the dimensions of the response vector $\boldsymbol{y}_i$), which is in contrast to many spatial and spatio-temporal models that build in a notion of neighborhood structure. This is motivated both by the fact that, as we see in the correlation maps of the exploratory data analysis in Appendix G, the definition of "neighborhood" is not necessarily straightforward to encode using Euclidean distance since geographically distant regions might have significant correlation[2]. Likewise, this structure need not remain fixed across time. Finally, the full set of 183 regions—comprised of cities, states, regions, and the U.S. national level—represents a type of multiresolution spatial description of flu activity. Although multiresolution-based spatial structures could be imposed based on known relationships, the inherent redundancy of these observations in this task is very well accommodated by a latent factor model. As we have shown, such a structure is very simple to work with computationally and enables our ability to straightforwardly cope with missing data without imputing these values. We could consider a model that combines latent factor and neighborhood based approaches, leading to low-rank plus sparse precision forms for the covariance. This is a topic that has received considerable recent attention (Chandrasekaran et al., 2012). We leave this as a direction of future research.

Details on our model and MCMC setup are provided in Section 5.2.4.

5.2.1 Qualitative Assessment

We begin by producing correlation map snapshots similar to those of the exploratory data analysis in Appendix G, but here with an ability to examine instantaneous correlations that utilize (i) all 183 regions jointly and (ii) the entire time course. In contrast, the analysis of Appendix G reduces dimensionality to state-level, aggregates data amongst flu versus non-flu events to cope with data scarcity, and discards data prior to Event B due to significant missing values. The results presented in Figures 6 and 7 clearly demonstrate that

2. Perhaps this effect arises from air travel (Brownstein et al., 2006), which was found to be a statistically significant driver in the state-level model of Hooten et al. (2010).



(a) (b)

Figure 6: (a) Plot of posterior means of the nonparametric mean function $\mu_j(x)$ for each of the 183 Google Flu Trends regions. The thick yellow line indicates the empirical mean of the log Google-estimated ILI rates, $\log r_{ij}$, across regions j. (b) For New York, the 25th, 50th, and 75th quantiles of correlation with the 182 other regions based on the posterior mean estimate of $\Sigma(x)$. The black line is a scaled version of the log Google-estimated United States ILI rate. The shaded gray regions indicate the time periods determined to be flu events (see Figure 5).

we are able to capture temporal changes in the spatial correlations of the Google Flu Trends data, even in the presence of substantial missing information. In Figure 6(b), we plot the posterior mean of the 183 components of $\boldsymbol{\mu}(x)$, showing trends that follow the empirical mean Google-estimated ILI rate. Although this mean model provides a slightly worse fit than the smoothing splines, our quantitative assessment of Section 5.2.2 demonstrates that modeling heteroscedasticity allows for a well-calibrated joint model. That is, we are robust to our simple choice for the mean regression function. (We note that more complicated mean models could be used within this framework, but this analysis demonstrates the flexibility of joint mean-covariance modeling.) For New York, in Figure 6(c) we plot the 25th, 50th, and 75th quantiles of correlation with the 182 other states and regions based on the posterior mean estimate of $\Sigma(x)$. From this plot, we immediately notice the time-varying correlations.

The specific time-varying geographic structure of the inferred correlations is displayed in Figure 7. Qualitatively, we see changes in the residual structure not just between flu and non-flu periods as in Appendix G, but also between flu events. In the more mild 2005-2006 season, we see much more local correlation structure than the more severe 2007-2008 season (which still maintains stronger regional than distant correlations.) The November 2009 H1N1 event displays overall regional correlation structure and values similar to the 2007-2008 season, but with key geographic areas that are less correlated. The 2006-2007 season is rather typical, with correlation maps very similar to those of the exploratory data analysis in Figure 12. Note that some geographically distant states, such as New York and California, are often highly correlated. Interestingly, the strong local spatial correlation structure for South Dakota in February 2006 has been inferred before any data are available for that state. Actually, no data are available for South Dakota from September 2003 to November 2006. Despite this missing data, the inferred correlation structures over these years are fairly consistent with those of neighboring states and change in manners similar to the flu-to-non-flu changes inferred after data for South Dakota are available. (See the movies

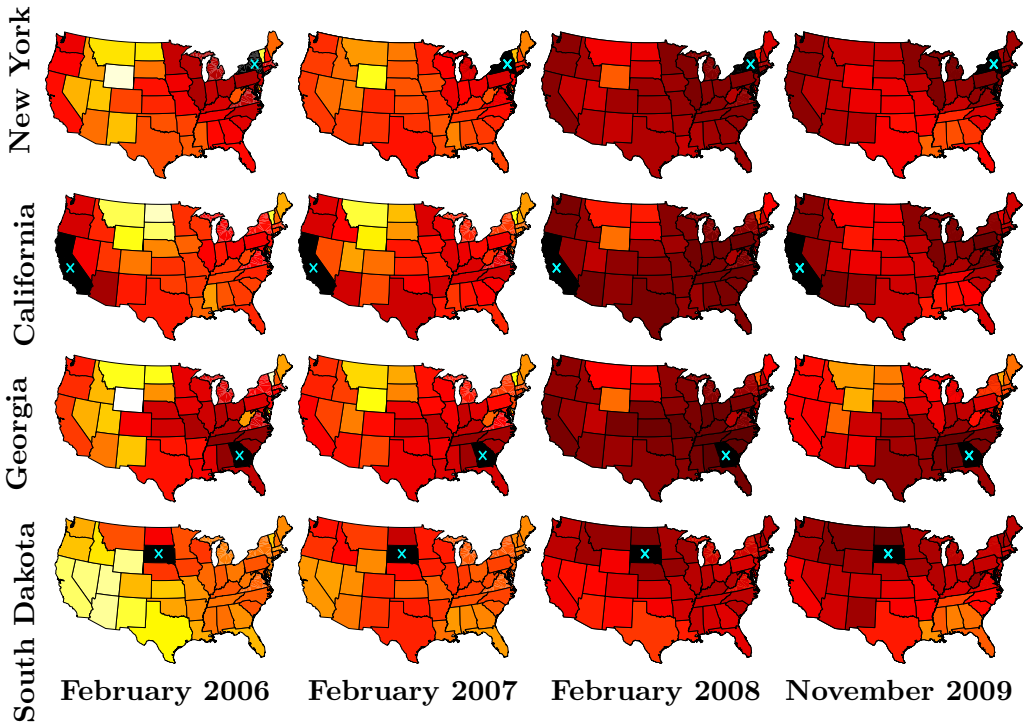


Figure 7: For the states in Figure 12 and each of four key dates (February 2006 of Event C, February 2007 of Event D, February 2008 of Event E, and November 2009 of Event F), correlation maps based on the posterior mean estimate of $\Sigma(x)$ using samples $[5000 : 10 : 10000]$ from 10 chains. The color scale is exactly the same as in Figure 12. The plots indicate spatial structure captured by $\Sigma(x)$, and that these spatial dependencies change over time. Note that no geographic information was included in our model.

provided in the Online Appendix.) This is enabled by the transfer of knowledge property described in Section 2.3. In particular, the row of Θ corresponding to South Dakota is informed by all of South Dakota's available data while the latent GP basis elements $\xi_{\ell k}$ are informed by all of the other regions' data, in addition to assumed continuity of $\xi_{\ell k}$ which shares information across time.

Comparing the maps of Figure 7 to those of the sample-based estimates in Figure 12, we see much of the same correlation structure, which at a high level validates our findings. Since the sample-based estimates aggregate data over Events B-F (containing those displayed in Figure 7), they tend to represent a time-average of the event-specific correlation structure we uncovered. Note that due to the dimensionality of the data set, the sample-based estimates are based solely on state-level measurements and thus are unable to harness the richness (and crucial redundancy) provided by the other regional reporting agencies. The high-dimensionality and missing data structure of this data set also limit our ability to compare to alternative methods such as those cited in Section 1—none yield results directly comparable to the full analysis we have provided here. Instead, they are either limited to examination of the small subset of data for which all observations are present and/or

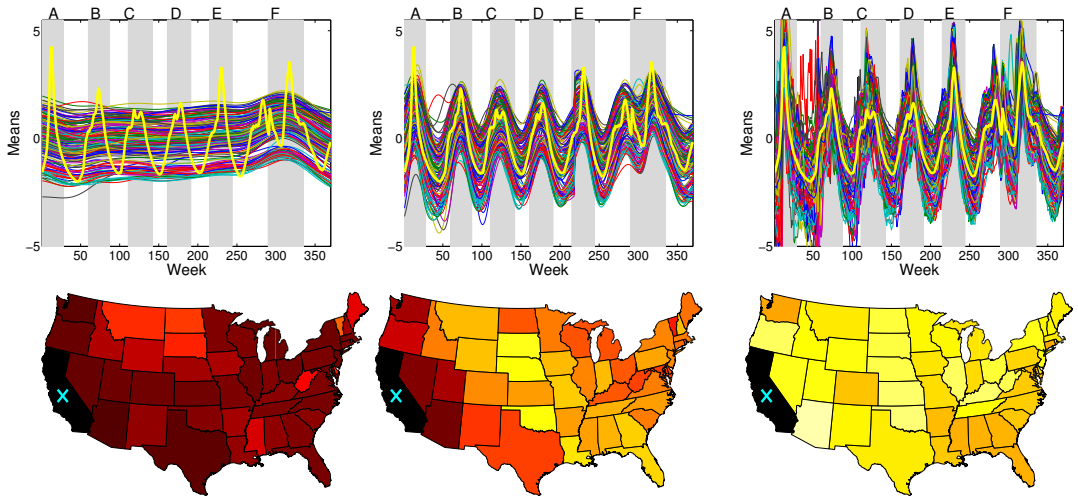


Figure 8: *Top:* Based on the nonparametric Nadaraya-Watson kernel estimator of Yin et al. (2010) using three different bandwidth settings ($(\kappa/2)^{-1/2} = 0.07, 0.02, 0.0008$), plots of the nonparametric mean estimate $\hat{\mu}_j(x)$ for each of the 183 regions, as in Figure 6(b). The estimate is based on averaging samples $[500 : 10 : 1000]$ from a stochastic EM chain that iterated between imputing missing values and computing the kernel estimate. Note that in the rightmost panel, the y-axis is truncated and the estimates in Event A actually extend to above 12. *Bottom:* Associated plots of correlations between California and all other states during February 2006 based on the nonparametric Nadaraya-Watson kernel estimator of the covariance function $\hat{\Sigma}(x)$. The color scale is exactly the same as in Figures 12 and 7.

a lower-dimensional selection (or projection) of observations. For example, the common GARCH models cannot handle missing data and are limited to typically no more than 5 dimensions. On the other hand, our proposed algorithm can readily utilize all information available to model the heteroscedasticity present here.

In an attempt to make a comparison, we propose a stochastic EM algorithm (Diebolt and Ip, 1995) for handling missing data within the framework of the nonparametric Nadaraya-Watson kernel estimator of Yin et al. (2010). Details are provided in Appendix H. The results based on a Gaussian kernel, as employed in Yin et al. (2010), are summarized in Figure 8. We examine three settings for the kernel bandwidth parameter: one (0.0008) based on the cross validation technique proposed in Yin et al. (2010) using the last portion of the time series without any missing values, one (0.02) tuned to match the smoothness of the mean function estimated from the Bayesian nonparametric method proposed herein (see Figure 6(b)), and one large setting (0.07) that leads to substantial sharing of information, but over-smooths the mean. The leave-one-out cross validation method leads to a very small bandwidth because of the specific temporal structure of the data (intuitively, the best estimate of a missing flu rate is achieved by averaging the nearest neighbors in time). However, this setting leads to poor predictive performance in the presence of consecutive

missing values. In Figure 8(a), we see the unreasonably large mean values estimated at the beginning of the time series when there is substantial missing data. The smoothness of the mean function using a bandwidth of 0.02 captures the global changes in flu activity, leaving the covariance to explain the residual correlations in the observations, better matching our goals. However, the covariance, as visualized through the California correlation map of February 2006 (Figure 8(middle)), lacks key geographic structure such as the strong correlation between California and New York. This correlation is present during other flu events, and is unlikely to be truly missing from this event. Instead, the failure to capture this and other correlations is likely due to the increased uncertainty from the substantial early missing data and lack of global sharing of information. Using a much larger bandwidth of 0.07 necessarily leads to more sharing of information, and results in the presence of these correlations. The resulting over-smooth mean function, however, does not capture global flu variations. On the other hand, our Bayesian nonparametric method is able to maintain a local description of the data while sharing information across the entire time series, thus ameliorating sensitivity to missing data.

5.2.2 Model Calibration

The plots of Section 5.2.1 qualitatively demonstrate that we are able to capture time-varying changes in the spatial conditional correlation structure of the (log) Google-estimated ILI rates. Despite not encoding spatial structure in our latent-factor-based model, we note that some local geographic structure has emerged, while still allowing for long-range correlations and temporal changes in this structure. We now turn to a quantitative assessment of the fit and robustness of our model. To this end, we examine posterior predictive intervals of randomly heldout data. More specifically, from the available observations (omitting the significant number of truly missing observations), we randomly held out 10% of the values uniformly across time and regions. We then simulated from our Gibbs sampler treating these values as missing data and analytically marginalizing them from the complete data likelihood, just as we do for the truly missing values. Based on each of our MCMC samples, we form $\boldsymbol{\mu}(x)$ and $\Sigma(x)$ for each $x = 1, \ldots, 370$ and compute the predictive distribution for the heldout data given any available *state-level* observations at week x (i.e., we condition on a subset of observed regions, ignoring non-state-level measurements). Averaging over MCMC samples, we then form 95% posterior predictive intervals and associated coverage rates for each x. We run this experiment of randomly holding out 10% of the observed data twice.

As a comparison, we consider an artificially generated homoscedastic model where we simply form $\hat{\Sigma} = \sum_{i=1}^{370} \Sigma(x_i)$ for each of our MCMC samples. In this case, both models have exactly the same mean regression, $\boldsymbol{\mu}(x)$. Likewise, the underlying $\boldsymbol{\mu}(x)$ and $\Sigma(x)$ (and thus $\hat{\Sigma}$) were all informed using the same low-dimensional embedding of the observations, harnessing the previously described benefits of such a latent factor approach. The only difference is whether we consider the week-specific covariance, $\Sigma(x)$, or instead its mean, $\hat{\Sigma}$, in forming our predictive intervals for week x. Considering the two experiments separately, we find that 94.4% and 94.6% of the heldout observations were covered by our 95% posterior predictive intervals, respectively. This result indicates that our joint mean-covariance regression model is well-calibrated and robust to the rather simple mean model. In compar-

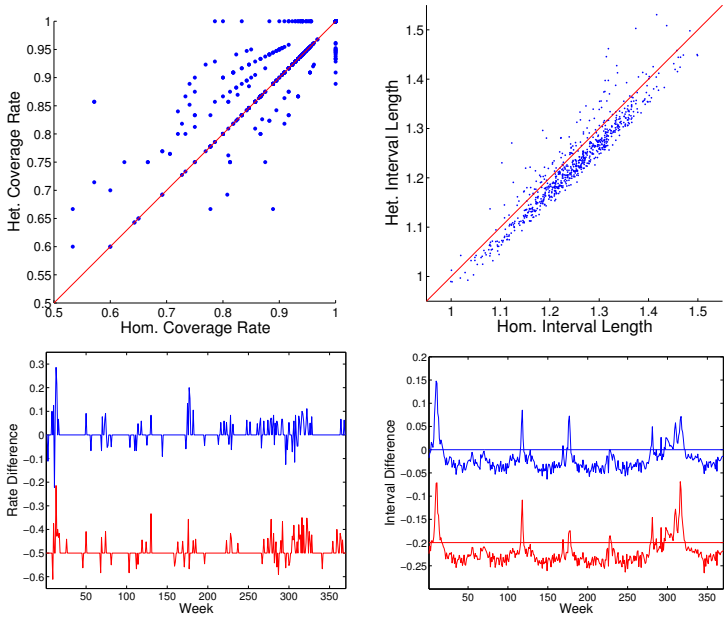


Figure 9: Comparison of posterior predictive intervals using our heteroscedastic model versus a homoscedastic model. *Top:* Scatter plots of week-specific coverage rates and interval lengths, aggregated over two experiments. *Bottom:* Differences in coverage rates and interval lengths by week, separating the experiments via an offset for clarity.

ison, the artificially generated homoscedastic model had coverage rates of 93.7% and 93.8%, respectively. Importantly, the better calibrated coverage rates of our heteroscedastic model came from shorter predictive intervals with average lengths of 1.2272 and 1.2268 compared to 1.2469 and 1.2475 for the homoscedastic model.

Figure 9 explores the differences between these posterior predictive intervals on a week-by-week basis. In Figure 9 (top) we see that a majority of the week-specific intervals have higher coverage rates and shorter interval lengths (i.e., most coverage rate comparisons are on or above the $x - y$ line whereas most interval length comparisons are below this line of equal performance). Time courses of the rates and interval lengths are shown separately for the two experiments in Figure 9 (bottom), where the temporal patterns in these differences become clear. There are stretches of weeks with identical coverage rates, leading to the similarity in overall rates for the two methods, though with the heteroscedastic approach using shorter intervals. The difference of going from overall rates of roughly 93.7% to 94.5% is attributed to certain bursts of time where capturing heteroscedasticity is really key. These time points can sometimes be attributed to the heteroscedastic approach providing wider intervals.

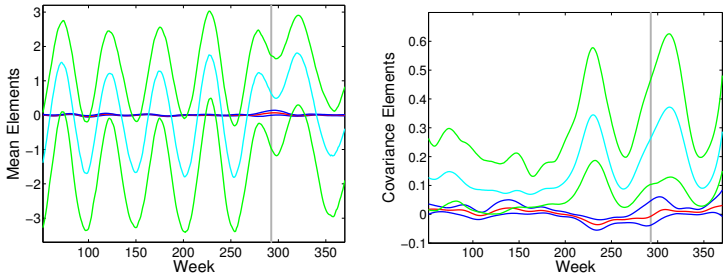


Figure 10: Comparison of posterior mean estimates of $\boldsymbol{\mu}(x)$ and $\Sigma(x)$ using the full data versus using the data with the outlying weeks of April 26, 2009 and May 3, 2009 removed. Denote the resulting estimates by $\{\hat{\boldsymbol{\mu}}(x), \hat{\Sigma}(x)\}$ and $\{\hat{\boldsymbol{\mu}}^{out}(x), \hat{\Sigma}^{out}(x)\}$, respectively. The two outlying weeks are highlighted by the gray shaded region. The blue lines indicate 0.05 and 0.95 pointwise-quantiles of the components (*left*) $\hat{\mu}_j(x) - \hat{\mu}_j^{out}(x)$ and (*right*) $\hat{\Sigma}_{ij}(x) - \hat{\Sigma}_{ij}^{out}(x)$. The red line is the median. The green and cyan lines are the corresponding 0.05, 0.5, and 0.95 quantiles of the full data $\hat{\mu}_j(x)$ and $\hat{\Sigma}_{ij}(x)$ shown for scale. We omit the first year with significant missing data to hone in on the smaller scale variability in subsequent years. No quantiles (blue/green lines) overlapped in this first year for the covariance elements, and trends were of the same scale for the mean elements.

5.2.3 Sensitivity to Outliers

As discussed in Section 5.1, there has been some debate about the accuracy of the Google Flu Trends estimates. In Cook et al. (2011), the two weeks of April 26, 2009 and May 3, 2009 were highlighted as having inflated Google estimates based on the significant media attention spurred by the H1N1 virus. In theory, our mean-covariance regression model has two defenses against such outliers. The first is due to the implicit shrinkage and regularization achieved through our use of a small set of latent basis functions. The second is the fact that an outlier at time x only has limited impact on inferences at time points x' that are "far" from x. This is formalized by in Lemma 5, where we see that the covariance between $\Sigma_{ij}(x)$ and $\Sigma_{uv}(x')$ decays with $(x - x')^2$ (for univariate x and our squared exponential kernel $c(x, x')$).

To empirically examine the impact of these outliers on our inferences, we removed all of the data from the weeks of April 26, 2009 and May 3, 2009 and simulated from our Gibbs sampler treating these data as missing. In Figure 10, we examine the differences between the posterior mean estimates of $\boldsymbol{\mu}(x)$ and $\Sigma(x)$ using the full data and this data set with the outlying weeks removed. We see that the most significant differences in our estimates are localized in time around these removed outlying weeks, as we would expect. Likewise, the sheer magnitude of these differences (which of course are larger for the harder-to-estimate covariance process) are quite small relative to the scale of the parameters in our model. These results demonstrate a robustness to outliers and allude to a robustness to certain types of malicious attacks.

5.2.4 MCMC Details, Sensitivity Analysis, and Convergence Diagnostics

We simulated 5 chains each for 10,000 MCMC iterations, discarded the first 5,000 for burn-in, and thinned the chains by examining every 10 samples. Each chain was initialized with parameters sampled from the prior. The hyperparameters were set as in the simulation study, except with larger truncation levels $\bar{L} = 10$ and $\bar{k} = 20$ and with the Gaussian process length-scale hyperparameter set to $\kappa_\psi = \kappa = 100$ to account for the time scale (weeks) and the rate at which ILI incidences change. By examining posterior samples of Θ, we found that the chosen truncation level was sufficiently large. To assess convergence, we performed the modified Gelman-Rubin diagnostic of Brooks and Gelman (1998) on the MCMC samples of the variance terms $\Sigma_{jj}(x_i)$. We also performed hyperparameter sensitivity, letting $\kappa_\psi = \kappa = 200$ to induce less temporal correlation and using a larger truncation level of $\bar{L} = \bar{k} = 20$ with less stringent shrinkage hyperparameters $a_1 = a_2 = 2$ (instead of $a_1 = a_2 = 10$). The results were essentially identical to those presented. Note that after taking the log transformation, the data were preprocessed by removing the empirical mean of each region and scaling the entire data set by one over the largest variance of any of the 183 time series.

5.2.5 Computational Complexity

Each of our chains of 10,000 Gibbs iterations based on a naive implementation in MATLAB (R2010b) took approximately 12 hours on a machine with four Intel Xeon X5550 Quad-Core 2.67GHz processors and 48 GB of RAM. For a sense of scaling of computations, the $p = 10$, $n = 100$ simulation study of Section 4 took 10 minutes for 10,000 Gibbs iterations while the $p = 30$, $n = 500$ scenario of took 3 hours for 10,000 Gibbs iterations. In terms of memory and storage, our method only requires maintaining samples of a $p \times L$ matrix Θ, the p elements of Σ_0, and an $L \times k \times q \times n$ matrix for the basis functions $\xi(x)$. (Compare to maintaining the $p \times p \times q \times n$ dimensional matrix for the Nadaraya-Watson estimates of $\Sigma(x)$ in the stochastic EM algorithm to which we compared.)

6. Discussion

In this paper, we have presented a Bayesian nonparametric approach to covariance regression which allows an unknown $p \times p$ dimensional covariance matrix $\Sigma(\boldsymbol{x})$ to vary flexibly over $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X}$ is some arbitrary (potentially multivariate) predictor space. As a concrete example, we considered multivariate heteroscedastic modeling of the Google Flu Trends data set, where p represents the 183 regions and x a weekly index. Key to this analysis is our model's ability to (i) scale to the full 183 regions (large p) and (ii) cope with the significant missing data without relying on imputing these values. Inherent to both of these capabilities is our predictor-dependent latent factor model that enables efficient sharing of information in a low-dimensional subspace. The factor loadings are based on a quadratic mixing over a collection of basis elements, assumed herein to be Gaussian process random functions, defined over $\mathcal{X}$. The Gaussian processes define a continuous evolution over $\mathcal{X}$ (e.g., time in the flu analysis), allowing us cope with an irregular grid of observations. Our proposed methodology also yields computationally tractable algorithms for posterior inference via fully conjugate Gibbs updates—this is crucial in our being able to analyze high-dimensional

multivariate data sets. In our Google Flu Trends analysis, we demonstrated the scalability, calibration, and robustness of our formulation.

There are many possible extensions of the proposed covariance regression framework. The most immediate are those that fall into the categories of (i) avoiding the multivariate Gaussian assumption, and (ii) scaling to data sets with larger numbers of observations.

In terms of (i), a natural possibility is to embed the proposed model within a richer hierarchical framework. For example, a promising approach is to use a Gaussian copula while allowing the marginal distributions to be unknown as in Hoff (2007). One can also use more flexible distributions for the latent variables and residuals, such as mixtures of Gaussians. Additionally, it would be trivial to extend our framework to accommodate multivariate categorical responses, or joint categorical and continuous responses, by employing the latent variable probit model of Albert and Chib (1993).

In terms of (ii), our sampler relies on $\bar{L} \times \bar{k}$ draws from an n-dimensional Gaussian (i.e., posterior draws of our Gaussian process random basis functions). For very large n, this becomes infeasible in practice since computations are, in general, $O(n^3)$. Standard tools for scaling up Gaussian process computation to large data sets, such as covariance tapering (Kaufman et al., 2008; Du et al., 2009) and the predictive process (Banerjee et al., 2008), can be applied directly in our context. Additionally, one might consider using the integrated nested Laplace approximations of Rue et al. (2009) for computations. One could also consider replacing the chosen basis elements with a basis expansion, wavelets, or simply autoregressive (i.e., band-limited) Gaussian processes. Including flat basis elements allows the model to collapse on homoscedasticity, enabling testing for heteroscedasticity.

It is also interesting to consider extensions that harness a known, predictor-independent structured covariance. One approach is to assume a *low rank plus sparse* model (instead of our low rank plus diagonal) in which the residuals have a sparse conditional dependence structure. For example, in the Google flu application the residuals could be modeled via a Markov random field to capture static local spatial dependencies while the low-rank portion captures time variation about this nominal structure. One could similarly extend to unknown sparse structures. Such formulations might allow for fewer latent factor dimensions.

There are also a number of interesting theoretical directions related to showing posterior consistency and rates of convergence including in cases in which the dimension p increases with sample size n.

Acknowledgments

The authors would like to thank Surya Tokdar for helpful discussions on the proof of prior support for the proposed covariance regression formulation. This work was supported in part by NSF Award 0903022 and DARPA Grant FA9550-12-1-0406 negotiated by AFOSR.

Appendix A: Proofs of Theorems and Lemmas

In Lemma 7, we show that our proposed factorization of $\Sigma(x)$ in (3) and (8) is sufficiently flexible to characterize any $\{\Sigma(x), x \in \mathcal{X}\}$ for sufficiently large k. Let $\mathcal{X}_\xi$ denote the space of all possible $L \times k$ arrays of $\mathcal{X} \to \Re$ functions, $\mathcal{X}_{\Sigma_0}$ all $p \times p$ diagonal matrices with non-negative entries, and $\mathcal{X}_\Theta$ all $p \times L$ real-valued matrices such that $\Theta\Theta'$ has finite elements. Recall that our modeling specification considers $L \to \infty$.

Lemma 7 *Given $\Sigma : \mathcal{X} \to \mathcal{P}_p^+$, for sufficiently large k there exists $\{\xi(\cdot), \Theta, \Sigma_0\} \in \mathcal{X}_\xi \otimes \mathcal{X}_\Theta \otimes \mathcal{X}_{\Sigma_0}$ such that $\Sigma(x) = \Theta\xi(x)\xi(x)'\Theta' + \Sigma_0$, for all $x \in \mathcal{X}$.*
Proof *Assume without loss of generality that $\Sigma_0 = 0_{p\times p}$ and take $k \geq p$. Consider*

$$\Theta = [I_p \; 0_{p\times 1} \; 0_{p\times 1} \; \ldots], \quad \xi(x) = \begin{pmatrix} chol(\Sigma(x)) & 0_{p\times k-p} \\ 0_{1\times p} & 0_{1\times k-p} \\ 0_{1\times p} & 0_{1\times k-p} \\ \vdots & \vdots \end{pmatrix}. \tag{13}$$

Then, $\Sigma(x) = \Theta\xi(x)\xi(x)'\Theta^T$ for all $x \in \mathcal{X}$. ■

Proof [Proof of Theorem 2] Since $\mathcal{X}$ is compact, for every $\epsilon_0 > 0$ there exists an open covering of ϵ_0-balls $B_{\epsilon_0}(x_0) = \{x : ||x - x_0||_2 < \epsilon_0\}$ with a finite subcover such that $\bigcup_{x_0\in\mathcal{X}_0} B_{\epsilon_0}(x_0) \supset \mathcal{X}$, where $|\mathcal{X}_0| = n$. Then,

$$\Pi_\Sigma\left(\sup_{x\in\mathcal{X}} ||\Sigma(x) - \Sigma^*(x)||_2 < \epsilon\right) = \Pi_\Sigma\left(\max_{x_0\in\mathcal{X}_0} \sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma(x) - \Sigma^*(x)||_2 < \epsilon\right). \tag{14}$$

Define $Z(x_0) = \sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma(x) - \Sigma^*(x)||_2$. Since

$$\Pi_\Sigma\left(\max_{x_0\in\mathcal{X}_,} Z(x_0) < \epsilon\right) > 0 \iff \Pi_\Sigma\left(Z(x_0) < \epsilon\right) > 0, \, \forall x_0 \in \mathcal{X}_0, \tag{15}$$

we only need to look at each ϵ_0-ball independently, which we do as follows.

$$\begin{aligned}
&\Pi_\Sigma\left(\sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma(x) - \Sigma^*(x)||_2 < \epsilon\right) \\
&\quad \geq \Pi_\Sigma\Bigg(\sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma^*(x_0) - \Sigma^*(x)||_2 + \sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma(x_0) - \Sigma(x)||_2 \\
&\qquad\qquad + ||\Sigma(x_0) - \Sigma^*(x_0)||_2 < \epsilon\Bigg) \\
&\quad \geq \Pi_\Sigma\left(\sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma^*(x_0) - \Sigma^*(x)||_2 < \epsilon/3\right) \\
&\qquad \cdot \Pi_\Sigma\left(\sup_{x\in B_{\epsilon_0}(x_0)} ||\Sigma(x_0) - \Sigma(x)||_2 < \epsilon/3\right) \Pi_\Sigma\left(||\Sigma(x_0) - \Sigma^*(x_0)||_2 < \epsilon/3\right)
\end{aligned} \tag{16}$$

where the first inequality comes from repeated uses of the triangle inequality, and the second inequality follows from the fact that each of these terms is an independent event. We evaluate each of these terms in turn. The first follows directly from the assumed continuity of $\Sigma^*(\cdot)$. The second will follow from a statement of (almost sure) continuity of $\Sigma(\cdot)$ that arises from the (almost sure) continuity of the $\xi_{\ell k}(\cdot) \sim \mathrm{GP}(0, c)$ and the shrinkage prior on $\theta_{\ell k}$ (i.e., $\theta_{\ell k} \to 0$ almost surely as $\ell \to \infty$, and does so "fast enough".) Finally, the third will follow from the support of the conditionally Wishart prior on $\Sigma(x_0)$ at every fixed $x_0 \in \mathcal{X}$.

Based on the continuity of $\Sigma^*(\cdot)$, for all $\epsilon/3 > 0$ there exists an $\epsilon_{0,1} > 0$ such that

$$||\Sigma^*(x_0) - \Sigma^*(x)||_2 < \epsilon/3, \quad \forall ||x - x_0||_2 < \epsilon_{0,1}. \tag{17}$$

Therefore, $\Pi_\Sigma \left(\sup_{x \in B_{\epsilon_{0,1}}(x_0)} ||\Sigma^*(x_0) - \Sigma^*(x)||_2 < \epsilon/3 \right) = 1$.

Based on Theorem 8, each element of $\Lambda(\cdot) \triangleq \Theta\xi(\cdot)$ is almost surely continuous on $\mathcal{X}$ assuming k finite. Letting $g_{jk}(x) = [\Lambda(x)]_{jk}$,

$$[\Lambda(x)\Lambda(x)']_{ij} = \sum_{m=1}^{k} g_{im}(x) g_{jm}(x), \quad \forall x \in \mathcal{X}. \tag{18}$$

Eq. (18) represents a finite sum over pairwise products of almost surely continuous functions, and thus results in a matrix $\Lambda(x)\Lambda(x)'$ with elements that are almost surely continuous on $\mathcal{X}$. Therefore, $\Sigma(x) = \Lambda(x)\Lambda(x)' + \Sigma_0 = \Theta\xi(x)\xi(x)'\Theta' + \Sigma_0$ is almost surely continuous on $\mathcal{X}$. We can then conclude that for all $\epsilon/3 > 0$ there exists an $\epsilon_{0,2} > 0$ such that

$$\Pi_\Sigma \left(\sup_{x \in B_{\epsilon_{0,2}}(x_0)} ||\Sigma(x_0) - \Sigma(x)||_2 < \epsilon/3 \right) = 1. \tag{19}$$

To examine the third term, we first note that

$$\begin{aligned} &\Pi_\Sigma \left(||\Sigma(x_0) - \Sigma^*(x_0)||_2 < \epsilon/3 \right) \\ &\qquad = \Pi_\Sigma \left(||\Theta\xi(x_0)\xi(x_0)'\Theta' + \Sigma_0 - \Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'} - \Sigma_0^*||_2 < \epsilon/3 \right), \end{aligned} \tag{20}$$

where $\{\xi^*(x_0), \Theta^*, \Sigma_0^*\}$ is any element of $\mathcal{X}_\xi \otimes \mathcal{X}_\Theta \otimes \mathcal{X}_{\Sigma_0}$ such that $\Sigma^*(x_0) = \Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'} + \Sigma_0^*$ with $\Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'}$ having rank k^*. Such a factorization exists by the assumption of Σ^* being k^*-decomposable. If $k^* = p$, Lemma 7 states that such a decomposition exists for *any* Σ^*. We can then bound this prior probability by

$$\begin{aligned} &\Pi_\Sigma \left(||\Sigma(x_0) - \Sigma^*(x_0)||_2 < \epsilon/3 \right) \\ &\qquad \geq \Pi_\Sigma \left(||\Theta\xi(x_0)\xi(x_0)'\Theta' - \Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'}||_2 < \epsilon/6 \right) \\ &\qquad\qquad \Pi_{\Sigma_0} \left(||\Sigma_0 - \Sigma_0^*||_2 < \epsilon/6 \right) \\ &\qquad \geq \Pi_\Sigma \left(||\Theta\xi(x_0)\xi(x_0)'\Theta' - \Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'}||_2 < \epsilon/6 \right) \\ &\qquad\qquad \Pi_{\Sigma_0} \left(||\Sigma_0 - \Sigma_0^*||_\infty < \epsilon/(6\sqrt{p}) \right), \end{aligned} \tag{21}$$

where the first inequality follows from the triangle inequality, and the second from the fact that for all $A \in \Re^{p \times p}$, $||A||_2 \leq \sqrt{p}||A||_\infty$, with the sup-norm defined as $||A||_\infty =$

$\max_{1\leq i\leq p}\sum_{i=1}^{p}|a_{ij}|$. Since $\Sigma_0 = diag(\sigma_1^2,\ldots,\sigma_p^2)$ with $\sigma_i^2 \overset{i.i.d.}{\sim} \text{Ga}(a_\sigma, b_\sigma)$, the support of the gamma prior implies that

$$\Pi_{\Sigma_0}\left(||\Sigma_0 - \Sigma_0^*||_\infty < \epsilon/(6\sqrt{p})\right) = \Pi_{\Sigma_0}\left(\max_{1\leq i\leq p}|\sigma_i^2 - \sigma_i^{*2}| < \epsilon/(6\sqrt{\pi})\right) > 0. \tag{22}$$

Recalling that $[\xi(x_0)]_{\ell k} = \xi_{\ell k}(x_0)$ with $\xi_{\ell k}(x_0) \overset{i.i.d.}{\sim} \mathcal{N}(0,1)$ and taking Θ a real matrix with $\text{rank}(\Theta) = p$,

$$\Theta\xi(x_0)\xi(x_0)'\Theta' \mid \Theta \sim \text{W}(k, \Theta\Theta'). \tag{23}$$

By Assumption 2.2, there is positive probability under Π_Θ on the set of Θ such that $\text{rank}(\Theta) = p$. Since $\Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'}$ is an arbitrary symmetric positive semidefinite matrix in $\Re^{p\times p}$ with rank $k \geq k^*$, and based on the support of the Wishart distribution,

$$\Pi_\Sigma\left(||\Theta\xi(x_0)\xi(x_0)'\Theta' - \Theta^*\xi^*(x_0)\xi^*(x_0)'\Theta^{*'}||_2 < \epsilon/6\right) > 0. \tag{24}$$

We thus conclude that $\Pi_\Sigma\left(||\Sigma(x_0) - \Sigma^*(x_0)||_2 < \epsilon/3\right) > 0$.

For every $\Sigma^*(\cdot)$ and $\epsilon > 0$, let $\epsilon_0 = \min(\epsilon_{0,1}, \epsilon_{0,2})$ with $\epsilon_{0,1}$ and $\epsilon_{0,2}$ defined as above. Then, combining the positivity results of each of the three terms in Eq. (16) completes the proof. ■

Theorem 8 *Assuming $\mathcal{X}$ compact, for every finite k and $L \to \infty$ (or L finite), $\Lambda(\cdot) = \Theta\xi(\cdot)$ is almost surely continuous on $\mathcal{X}$.*

Proof [Proof of Theorem 8] We can represent each element of $\Lambda(\cdot)$ as follows:

$$\begin{aligned}[\Lambda(\cdot)]_{jk} &= \lim_{L\to\infty}\left[\begin{bmatrix}\theta_{11} & \theta_{12} & \ldots & \theta_{1L}\\ \theta_{21} & \theta_{22} & \ldots & \theta_{2L}\\ \vdots & \vdots & \ddots & \vdots\\ \theta_{p1} & \theta_{p2} & \ldots & \theta_{pL}\end{bmatrix}\begin{bmatrix}\xi_{11}(\cdot) & \xi_{12}(\cdot) & \ldots & \xi_{1k}(\cdot)\\ \xi_{21}(\cdot) & \xi_{22}(\cdot) & \ldots & \xi_{2k}(\cdot)\\ \vdots & \vdots & \ddots & \vdots\\ \xi_{L1}(\cdot) & \xi_{L2}(\cdot) & \ldots & \xi_{Lk}(\cdot)\end{bmatrix}\right]_{jk}\\ &= \sum_{\ell=1}^{\infty}\theta_{j\ell}\xi_{\ell k}(\cdot).\end{aligned} \tag{25}$$

If $\xi_{\ell k}(x)$ is continuous for all ℓ, k and $s_n(x) = \sum_{\ell=1}^{n}\theta_{j\ell}\xi_{\ell k}(x)$ uniformly converges almost surely to some $g_{jk}(x)$, then $g_{jk}(x)$ is almost surely continuous. That is, if for all $\epsilon > 0$ there exists an N such that for all $n \geq N$

$$\Pr\left(\sup_{x\in\mathcal{X}}|g_{jk}(x) - s_n(x)| < \epsilon\right) = 1, \tag{26}$$

then $s_n(x)$ converges uniformly almost surely to $g_{jk}(x)$ and we can conclude that $g_{jk}(x)$ is continuous based on the definition of $s_n(x)$. To show almost sure uniform convergence, it is sufficient to show that there exists an M_n with $\sum_{n=1}^{\infty}M_n$ almost surely convergent and

$$\sup_{x\in\mathcal{X}}|\theta_{jn}\xi_{nk}(x)| \leq M_n. \tag{27}$$

Let $c_{nk} = \sup_{x \in \mathcal{X}} |\xi_{nk}(x)|$. Then,

$$\sup_{x \in \mathcal{X}} |\theta_{jn} \xi_{nk}(x)| \leq |\theta_{jn}| c_{nk}. \tag{28}$$

Since $\xi_{nk}(\cdot) \overset{i.i.d.}{\sim} \text{GP}(0, c)$ and $\mathcal{X}$ is compact, $c_{nk} < \infty$ and $E[c_{nk}] = \bar{c}$ with $\bar{c}$ finite. Defining $M_n = |\theta_{jn}| c_{nk}$,

$$\begin{aligned} E_{\Theta,c}\left[\sum_{n=1}^{\infty} M_n\right] &= E_{\Theta}\left[E_{c|\Theta}\left[\sum_{n=1}^{\infty} |\theta_{jn}| c_{nk} \mid \Theta\right]\right] = E_{\Theta}\left[\sum_{n=1}^{\infty} |\theta_{jn}| \bar{c}\right] \\ &= \bar{c} \sum_{n=1}^{\infty} E_{\Theta}\left[|\theta_{jn}|\right], \end{aligned} \tag{29}$$

where the last equality follows from Fubini's theorem. Based on Assumption 2.1, we conclude that $E[\sum_{n=1}^{\infty} M_n] < \infty$ which implies that $\sum_{n=1}^{\infty} M_n$ converges almost surely. ■

Lemma 9 *Assuming the prior specification of expression (9) with $a_2 > 2$ and $\gamma > 2$, the rows of Θ are absolutely summable in expectation: $\sum_{\ell} E(|\theta_{j\ell}|) < \infty$, satisfying Assumption 2.1.*

Proof [Proof of Lemma 9] Recall that $\theta_{j\ell} \sim \mathcal{N}(0, \phi_{j\ell}^{-1} \tau_{\ell}^{-1})$ with $\phi_{j\ell} \sim \text{Ga}(\gamma/2, \gamma/2)$ and $\tau_{\ell} = \prod_{h=1}^{\ell} \delta_h$ for $\delta_1 \sim \text{Ga}(a_1, 1)$, $\delta_h \sim \text{Ga}(a_2, 1)$. Using the fact that if $x \sim \mathcal{N}(0, \sigma^2)$ then $E[|x|] = \sigma\sqrt{2/\pi}$ and if $y \sim \text{Ga}(a, b)$ then $1/y \sim \text{Inv-Ga}(a, 1/b)$ with $E[1/y] = 1/(b \cdot (a-1))$, we derive that

$$\begin{aligned} \sum_{\ell=1}^{\infty} E_{\theta}[|\theta_{j\ell}|] &= \sum_{\ell=1}^{\infty} E_{\phi,\tau}[E_{\theta|\phi,\tau}[|\theta_{j\ell}| \mid \phi_{j\ell}, \tau_{\ell}]] = \sqrt{\frac{2}{\pi}} \sum_{\ell=1}^{\infty} E_{\phi,\tau}[\phi_{j\ell}^{-1} \tau_{\ell}^{-1}] \\ &= \sqrt{\frac{2}{\pi}} \sum_{\ell=1}^{\infty} E_{\phi}[\phi_{j\ell}^{-1}] E_{\tau}[\tau_{\ell}^{-1}] = \frac{4}{\gamma(\gamma-2)} \sqrt{\frac{2}{\pi}} \sum_{\ell=1}^{\infty} E_{\delta}\left[\prod_{h=1}^{\ell} \frac{1}{\delta_h}\right] \\ &= \frac{1}{a_1 - 1} \frac{4}{\gamma(\gamma-2)} \sqrt{\frac{2}{\pi}} \sum_{\ell=1}^{\infty} \left(\frac{1}{a_2 - 1}\right)^{\ell-1}. \end{aligned} \tag{30}$$

When $a_2 > 2$ and $\gamma > 2$, we conclude that $\sum_{\ell} E[|\theta_{j\ell}|] < \infty$. ■

Proof [Proof of Lemma 4] Recall that $\Sigma(x) = \Theta \xi(x) \xi(x)' \Theta' + \Sigma_0$ with $\Sigma_0 = \text{diag}(\sigma_1^2, \ldots, \sigma_p^2)$. The elements of the respective matrices are independently distributed as $\theta_{i\ell} \sim \mathcal{N}(0, \phi_{i\ell}^{-1} \tau_{\ell}^{-1})$, $\xi_{\ell k}(\cdot) \sim \text{GP}(0, c)$, and $\sigma_i^{-2} \sim \text{Gamma}(a_\sigma, b_\sigma)$. Let μ_σ and σ_σ^2 represent the mean and variance of the implied inverse gamma prior on σ_i^2, respectively. In all of the following, we first condition on Θ and then use iterated expectations to find the marginal moments.

The expected covariance matrix at any predictor location x is simply derived as

$$\begin{aligned} E[\Sigma(x)] &= E[E[\Sigma(x) \mid \Theta]] = E[E[\Theta \xi(x) \xi(x)' \Theta' \mid \Theta]] + \mu_\sigma I_p = k E[\Theta \Theta'] + \mu_\sigma I_p \\ &= \text{diag}(k \sum_{\ell} \phi_{1\ell}^{-1} \tau_{\ell}^{-1} + \mu_\sigma, \ldots, k \sum_{\ell} \phi_{p\ell}^{-1} \tau_{\ell}^{-1} + \mu_\sigma). \end{aligned}$$

Here, we have used the fact that conditioned on Θ, $\Theta\xi(x)\xi(x)'\Theta'$ is Wishart distributed with mean $k\Theta\Theta'$ and

$$E[\Theta\Theta']_{ij} = \sum_{\ell}\sum_{\ell'} E[\theta_{i\ell}\theta_{j\ell'}] = \sum_{\ell} E[\theta_{i\ell}^2]\delta_{ij} = \sum_{\ell} \text{var}(\theta_{i\ell})\delta_{ij} = \sum_{\ell} \phi_{i\ell}^{-1}\tau_{\ell}^{-1}\delta_{ij}.$$

■

Proof [Proof of Lemma 5] One can use the conditionally Wishart distribution of $\Theta\xi(x)\xi(x)'\Theta'$ to derive $\text{cov}(\Sigma_{ij}(x), \Sigma_{uv}(x))$. Specifically, let $S = \Theta\xi(x)\xi(x)'\Theta'$. Then $S = \sum_{n=1}^{k} z^{(n)}z^{(n)'}$ with $z^{(n)} \mid \Theta \sim \mathcal{N}(0, \Theta\Theta')$ independently for each n. Then, using standard Gaussian second and fourth moment results,

$$\begin{aligned}
\text{cov}(\Sigma_{ij}(x), \Sigma_{uv}(x) \mid \Theta) &= \text{cov}(S_{ij}, S_{uv} \mid \Theta) + \sigma_\sigma^2\delta_{ijuv} \\
&= \sum_{n=1}^{k} E[z_i^{(n)}z_j^{(n)}z_u^{(n)}z_v^{(n)} \mid \Theta] - E[z_i^{(n)}z_j^{(n)} \mid \Theta]E[z_u^{(n)}z_v^{(n)} \mid \Theta] + \sigma_\sigma^2\delta_{ijuv} \\
&= k((\Theta\Theta')_{iu}(\Theta\Theta')_{jv} + (\Theta\Theta')_{iv}(\Theta\Theta')_{ju}) + \sigma_\sigma^2\delta_{ijuv}.
\end{aligned}$$

Here, $\delta_{ijuv} = 1$ if $i = j = u = v$ and is 0 otherwise. Taking the expectation with respect to Θ yields $\text{cov}(\Sigma_{ij}(x), \Sigma_{uv}(x))$. However, instead of looking at one slice of the predictor space, we are interested in how the correlation between elements of the covariance matrix changes with predictors. Thus, we work directly with the latent Gaussian processes to derive $\text{cov}(\Sigma_{ij}(x), \Sigma_{uv}(x'))$. Let

$$g_{in}(x) = \sum_{\ell} \theta_{i\ell}\xi_{\ell n}(x), \tag{31}$$

implying that $g_{in}(x)$ is independent of all $g_{im}(x')$ for any $m \neq n$ and all $x' \in \mathcal{X}$. Since each $\xi_{\ell n}(\cdot)$ is distributed according to a zero mean Gaussian process, $g_{in}(x)$ is zero mean. Using this definition, we condition on Θ (which is dropped in the derivations for notational simplicity) and write

$$\begin{aligned}
\text{cov}(\Sigma_{ij}(x), \Sigma_{uv}(x') \mid \Theta) &= \sum_{n=1}^{k} \text{cov}(g_{in}(x)g_{jn}(x), g_{un}(x'), g_{vn}(x')) + \sigma_\sigma^2\delta_{ijuv} \\
&= \sum_{n=1}^{k} E[g_{in}(x)g_{jn}(x)g_{un}(x'), g_{vn}(x')] \\
&\quad - E[g_{in}(x)g_{jn}(x)]E[g_{un}(x'), g_{vn}(x')] + \sigma_\sigma^2\delta_{ijuv}
\end{aligned}$$

We replace each $g_{kn}(x)$ by the form in Eq. (31), summing over different dummy indices for each. Using the fact that $\xi_{\ell n}(x)$ is independent of $\xi_{\ell' n}(x')$ for any $\ell \neq \ell'$ and that each $\xi_{\ell n}(x)$ is zero mean, all cross terms in the resulting products cancel if a $\xi_{\ell n}(x)$ arising from one $g_{kn}(x)$ does not share an index ℓ with at least one other $\xi_{\ell n}(x)$ arising from another

$g_{pn}(x)$. Thus,

$$\begin{aligned}\text{cov}(\Sigma_{ij}(x),\Sigma_{uv}(x') \mid \Theta) = &\sum_{n=1}^{k}\sum_{\ell}\theta_{i\ell}\theta_{j\ell}\theta_{u\ell}\theta_{v\ell}E[\xi_{\ell n}^2(x)\xi_{\ell n}^2(x')] \\ &+\sum_{\ell}\theta_{i\ell}\theta_{u\ell}E[\xi_{\ell n}(x)\xi_{\ell n}(x')]\sum_{\ell'\neq\ell}\theta_{j\ell'}\theta_{v\ell'}E[\xi_{\ell' n}(x)\xi_{\ell' n}(x')] \\ &+\sum_{\ell}\theta_{i\ell}\theta_{j\ell}E[\xi_{\ell n}^2(x)]\sum_{\ell'\neq\ell}\theta_{u\ell'}\theta_{v\ell'}E[\xi_{\ell' n}^2(x')] \\ &-\sum_{\ell}\theta_{i\ell}\theta_{j\ell}E[\xi_{\ell n}^2(x)]\sum_{\ell'}\theta_{u\ell'}\theta_{v\ell'}E[\xi_{\ell' n}^2(x')] + \sigma_\sigma^2\delta_{ijuv}\end{aligned}$$

The Gaussian process moments are given by

$$\begin{aligned}E[\xi_{\ell n}^2(x)] &= 1 \\ E[\xi_{\ell n}(x)\xi_{\ell n}(x')] &= E[E[\xi_{\ell n}(x) \mid \xi_{\ell n}(x')]\xi_{\ell n}(x')] = c(x,x')E[\xi_{\ell n}^2(x')] = c(x,x') \\ E[\xi_{\ell n}^2(x)\xi_{\ell n}^2(x')] &= E[E[\xi_{\ell n}^2(x) \mid \xi_{\ell n}(x')]\xi_{\ell n}^2(x')] \\ &= E[\{(E[\xi_{\ell n}(x) \mid \xi_{\ell n}(x')])^2 + \text{var}(\xi_{\ell n}(x) \mid \xi_{\ell n}(x'))\}\xi_{\ell n}^2(x')] \\ &= c^2(x,x')E[\xi_{\ell n}^4(x')] + (1-c^2(x,x'))E[\xi_{\ell n}^2(x')] = 2c^2(x,x') + 1,\end{aligned}$$

from which we derive that

$$\begin{aligned}&\text{cov}(\Sigma_{ij}(x),\Sigma_{uv}(x') \mid \Theta) \\ &= k\Bigg\{(2c^2(x,x')+1)\sum_{\ell}\theta_{i\ell}\theta_{j\ell}\theta_{u\ell}\theta_{v\ell} + c^2(x,x')\sum_{\ell}\theta_{i\ell}\theta_{u\ell}\sum_{\ell'\neq\ell}\theta_{j\ell'}\theta_{v\ell'} \\ &\qquad + \sum_{\ell}\theta_{i\ell}\theta_{j\ell}\sum_{\ell'\neq\ell}\theta_{u\ell'}\theta_{v\ell'} - \sum_{\ell}\theta_{i\ell}\theta_{j\ell}\sum_{\ell'}\theta_{u\ell'}\theta_{v\ell'}\Bigg\} + \sigma_\sigma^2\delta_{ijuv} \\ &= kc^2(x,x')\left\{\sum_{\ell}\theta_{i\ell}\theta_{j\ell}\theta_{u\ell}\theta_{v\ell} + \sum_{\ell}\theta_{i\ell}\theta_{u\ell}\sum_{\ell'}\theta_{j\ell'}\theta_{v\ell'}\right\} + \sigma_\sigma^2\delta_{ijuv}.\end{aligned}$$

An iterated expectation with respect to Θ yields the following results. When $i \neq u$ or $j \neq v$, the independence between $\theta_{i\ell}$ (or $\theta_{j\ell}$) and the set of other $\theta_{k\ell}$ implies that $\text{cov}(\Sigma_{ij}(x),\Sigma_{uv}(x')) = 0$. When $i = u$ and $j = v$, but $i \neq j$,

$$\begin{aligned}\text{cov}(\Sigma_{ij}(x),\Sigma_{ij}(x')) &= kc^2(x,x')\left\{\sum_{\ell}E[\theta_{i\ell}^2]E[\theta_{j\ell}^2] + \sum_{\ell}E[\theta_{i\ell}^2]\sum_{\ell'}E[\theta_{j\ell'}^2]\right\} \\ &= kc^2(x,x')\left\{\sum_{\ell}\phi_{i\ell}^{-1}\phi_{j\ell}^{-1}\tau_{\ell}^{-2} + \sum_{\ell}\phi_{i\ell}^{-1}\tau_{\ell}^{-1}\sum_{\ell'}\phi_{j\ell'}^{-1}\tau_{\ell'}^{-1}\right\}.\end{aligned}$$

Finally, when $i = j = u = v$,

$$\begin{aligned}\text{cov}(\Sigma_{ii}(x), \Sigma_{ii}(x')) &= kc^2(x,x') \left\{ 2\sum_\ell E[\theta_{i\ell}^4] + \sum_\ell E[\theta_{i\ell}^2] \sum_{\ell' \neq \ell} E[\theta_{i\ell'}^2] \right\} + \sigma_\sigma^2 \\ &= kc^2(x,x') \left\{ 6\sum_\ell \phi_{i\ell}^{-2}\tau_\ell^{-2} + \sum_\ell \phi_{i\ell}^{-1}\tau_\ell^{-1} \sum_{\ell' \neq \ell} \phi_{i\ell'}^{-1}\tau_{\ell'}^{-1} \right\} + \sigma_\sigma^2 \\ &= kc^2(x,x') \left\{ 5\sum_\ell \phi_{i\ell}^{-2}\tau_\ell^{-2} + (\sum_\ell \phi_{i\ell}^{-1}\tau_\ell^{-1})^2 \right\} + \sigma_\sigma^2.\end{aligned}$$

■

Proof [Proof of Lemma 6] The first-order stationarity follows immediately from the stationarity of the Gaussian process dictionary elements $\xi_{\ell k}$ and recalling that $\Sigma(x) = \Theta\xi(x)\xi(x)'\Theta' + \Sigma_0$. Assuming a Gaussian process kernel $c(x,x')$ that solely depends upon the distance between x and x', Lemma 5 implies that the defined process is wide sense stationary. ■

Appendix B: Derivation of Gibbs Sampler

In this Appendix, we derive the conditional distribution for sampling the Gaussian process dictionary elements. Combining Eq. (1) and Eq. (8), we have that

$$y_i = \Theta \begin{bmatrix} \xi_{11}(x_i) & \xi_{12}(x_i) & \dots & \xi_{1k}(x_i) \\ \xi_{21}(x_i) & \xi_{22}(x_i) & \dots & \xi_{2k}(x_i) \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{L1}(x_i) & \xi_{L2}(x_i) & \dots & \xi_{Lk}(x_i) \end{bmatrix} \eta_i + \epsilon_i = \Theta \begin{bmatrix} \sum_{m=1}^k \xi_{1m}(x_i)\eta_{im} \\ \vdots \\ \sum_{m=1}^k \xi_{Lm}(x_i)\eta_{Lm} \end{bmatrix} + \epsilon_i \tag{32}$$

implying that

$$y_{ij} = \sum_{\ell=1}^L \sum_{m=1}^k \theta_{j\ell}\eta_{im}\xi_{\ell m}(x_i) + \epsilon_{ij}. \tag{33}$$

Conditioning on $\xi(\cdot)^{-\ell m}$, we rewrite Eq. (32) as

$$y_i = \eta_{im} \begin{bmatrix} \theta_{1\ell} \\ \vdots \\ \theta_{p\ell} \end{bmatrix} \xi_{\ell m}(x_i) + \tilde{\epsilon}_i, \quad \tilde{\epsilon}_i \sim \mathcal{N}\left(\mu_{\ell m}(x_i) \triangleq \begin{bmatrix} \sum_{(r,s)\neq(\ell,m)} \theta_{1r}\eta_{is}\xi_{rs}(x_i) \\ \vdots \\ \sum_{(r,s)\neq(\ell,m)} \theta_{pr}\xi_{rs}(x_i) \end{bmatrix}, \Sigma_0 \right). \tag{34}$$

Let $\theta_{\cdot\ell} = \begin{bmatrix} \theta_{1\ell} & \dots & \theta_{p\ell} \end{bmatrix}'$. Then,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} \eta_{1m}\theta_{\cdot\ell} & 0 & \dots & 0 \\ 0 & \eta_{2m}\theta_{\cdot\ell} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \eta_{nm}\theta_{\cdot\ell} \end{bmatrix} \begin{bmatrix} \xi_{\ell m}(x_1) \\ \xi_{\ell m}(x_2) \\ \vdots \\ \xi_{\ell m}(x_n) \end{bmatrix} + \begin{bmatrix} \tilde{\epsilon}_1 \\ \tilde{\epsilon}_2 \\ \vdots \\ \tilde{\epsilon}_n \end{bmatrix} \tag{35}$$

Defining $A_{\ell m} = \text{diag}(\eta_{1m}\theta_{\cdot\ell}, \ldots, \eta_{nm}\theta_{\cdot\ell})$, our Gaussian process prior on the dictionary elements $\xi_{\ell m}(\cdot)$ implies the following conditional posterior

$$\begin{bmatrix} \xi_{\ell m}(x_1) \\ \xi_{\ell m}(x_2) \\ \vdots \\ \xi_{\ell m}(x_n) \end{bmatrix} \mid \{y_i\}, \Theta, \eta, \xi(\cdot), \Sigma_0 \sim \mathcal{N}\left(\tilde{\Sigma} A_{\ell m}' \begin{bmatrix} \Sigma_0^{-1} & & \\ & \ddots & \\ & & \Sigma_0^{-1} \end{bmatrix} \begin{bmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{bmatrix}, \tilde{\Sigma} \right)$$

$$= \mathcal{N}\left(\tilde{\Sigma} \begin{bmatrix} \eta_{1m} \sum_{j=1}^p \theta_{j\ell} \sigma_j^{-2} \tilde{y}_{1j} \\ \vdots \\ \eta_{nm} \sum_{j=1}^p \theta_{j\ell} \sigma_j^{-2} \tilde{y}_{nj} \end{bmatrix}, \tilde{\Sigma} \right), \tag{36}$$

where $\tilde{y}_i = y_i - \mu_{\ell m}(x_i)$ and, taking K to be the matrix of correlations $K_{ij} = c(x_i, x_j)$ defined by the Gaussian process parameter κ,

$$\tilde{\Sigma}^{-1} = K^{-1} + A_{\ell m}' \begin{bmatrix} \Sigma_0^{-1} & & \\ & \ddots & \\ & & \Sigma_0^{-1} \end{bmatrix} A_{\ell m}$$

$$= K^{-1} + \text{diag}\left(\eta_{1m}^2 \sum_{j=1}^p \theta_{j\ell}^2 \sigma_j^{-2}, \ldots, \eta_{nm}^2 \sum_{j=1}^p \theta_{j\ell}^2 \sigma_j^{-2} \right). \tag{37}$$

Appendix C: Hyperparameter Sampling and Empirical Bayes

One can also consider sampling the Gaussian process length-scale hyperparameter κ. Due to the linear-Gaussianity of the proposed covariance regression model, we can analytically marginalize the latent Gaussian process random functions in considering the posterior of κ. Taking $\mu(x) = 0$ for simplicity, our posterior is based on marginalizing the Gaussian processes random vectors $\underline{\xi}_{\ell m} = [\xi_{\ell m}(x_1) \ \ldots \ \xi_{\ell m}(x_n)]'$. Noting that

$$\begin{bmatrix} y_1' & y_2' & \ldots & y_n' \end{bmatrix}' = \sum_{\ell m} \left[\text{diag}(\eta_{\cdot m}) \otimes \theta_{\cdot \ell} \right] \underline{\xi}_{\ell m} + \begin{bmatrix} \epsilon_1' & \epsilon_2' & \ldots & \epsilon_n' \end{bmatrix}', \tag{38}$$

and letting K_κ denote the Gaussian process covariance matrix based on a length-scale κ,

$$\begin{bmatrix} y_1' & \cdots & y_n' \end{bmatrix}' \mid \kappa, \Theta, \eta, \Sigma_0 \sim$$

$$\text{N}_{np}\left(\sum_{\ell, m} \left[\text{diag}(\eta_{\cdot m}) \otimes \theta_{\cdot \ell} \right] K_\kappa \left[\text{diag}(\eta_{\cdot m}) \otimes \theta_{\cdot \ell} \right]' + I_n \otimes \Sigma_0 \right). \tag{39}$$

We can then Gibbs sample κ based on a fixed grid and prior $p(\kappa)$ on this grid. Note, however, that computation of the likelihood specified in Eq. (39) requires evaluation of an np-dimensional Gaussian for each value κ specified in the grid. For large p scenarios, or when there are many observations y_i, this may be computationally infeasible. In such cases, a naive alternative is to iterate between sampling ξ given K_κ and K_κ given ξ. However, this can lead to extremely slow mixing. Alternatively, one can consider employing the recent Gaussian process hyperparameter slice sampler of Adams and Murray (2011).

In general, because of the quadratic mixing over Gaussian process dictionary elements, our model is relatively robust to the choice of the length-scale parameter and the computational burden imposed by sampling κ is typically unwarranted. Instead, one can preselect a value for κ using a data-driven heuristic, which leads to a quasi-empirical Bayes approach. Lemma 5 implies that the autocorrelation $ACF(x) = \text{corr}(\Sigma_{ij}(0), \Sigma_{ij}(x))$ is simply specified by $c(0, x)$. As given by Eq. (11), when we choose a Gaussian process kernel $c(x, x') = \exp(-\kappa||x - x'||_2^2)$, we have $ACF(x) = \exp(-\kappa||x||_2^2)$. Thus, we see that the length-scale parameter κ directly determines the shape of the autocorrelation function. If one can devise a procedure for estimating the autocorrelation function from the data, one can set κ accordingly. We propose the following, most easily implemented for scalar predictor spaces $\mathcal{X}$, but also feasible (in theory) for multivariate $\mathcal{X}$.

1. For a set of evenly spaced knots $x_k \in \mathcal{X}$, compute the sample covariance $\hat{\Sigma}(x_k)$ from a local bin of data. If the bin contains fewer than p observations, add a small diagonal component to ensure positive definiteness.
2. Compute the Cholesky decomposition $C(x_k) = chol(\hat{\Sigma}(x_k))$.
3. Fit a spline through the elements of the computed $C(x_k)$. Denote the spline fit of the Cholesky by $\tilde{C}(x)$ for each $x \in \mathcal{X}$
4. For $i = 1, \ldots, n$, compute a point-by-point estimate of $\Sigma(x_i)$ from the splines: $\Sigma(x_i) = \tilde{C}(x_i)\tilde{C}(x_i)'$.
5. Compute the autocorrelation function of each element $\Sigma_{ij}(x)$ of this kernel-estimated $\Sigma(x)$.
6. According to $-\log(ACF(x)) = \kappa||x||_2^2$, choose κ to best fit the most correlated $\Sigma_{ij}(x)$ (since less correlated components can be captured via weightings of dictionary elements with stronger correlation.)

Appendix D: Initialization of Gibbs Sampler

Experimentally we found that our sampler was fairly insensitive to initialization (after a short burn-in period) and one can just initialize each of Θ, ξ, Σ_0, η_i, and the shrinkage parameters $\phi_{j\ell}$ and δ_h from their respective priors. However, in certain scenarios, the following more intricate initialization can improve mixing rates. The predictor-independent parameters Θ and Σ_0 are sampled from their respective priors (first sampling the shrinkage parameters $\phi_{j\ell}$ and δ_h from their priors). The variables η_i and $\xi(x_i)$ are set via a data-driven initialization scheme in which an estimate of $\Sigma(x_i)$ for $i = 1, \ldots, n$ is formed using Steps 1-4 outlined above. Then, $\Theta\xi(x_i)$ is taken to be a k^*-dimensional low-rank approximation to the Cholesky of the estimates of $\Sigma(x_i)$. The latent factors η_i are sampled from their posterior using this data-driven estimate of $\Theta\xi(x_i)$. Similarly, the $\xi(x_i)$ are initially taken to be spline fits of the pseudo-inverse of the low-rank Cholesky at the knot locations and the sampled Θ. We then iterate a couple of times between sampling: (i) ξ given $\{y_i\}$, Θ, Σ_0, and the data-driven estimates of η, ξ; (ii) Θ given $\{y_i\}$, Σ_0, η, and the sampled ξ; (iii) Σ_0 given $\{y_i\}$, Θ, η, and ξ; and (iv) determining a new data-driven approximation to ξ based on the newly sampled Θ.

Appendix E: Other Prior Specifications

Intuitively, the chosen prior on Θ flexibly shrinks the columns of this matrix towards zero as the column index increases, implying that the effect of dictionary elements $\xi_{\ell k}(x)$ on the induced covariance matrix $\Sigma(x)$ decreases with row index ℓ. Harnessing this idea, one can extend the framework to allow for variable-smoothness dictionary elements by introducing row-dependent bandwidth parameters κ_ℓ. For example, one could encourage increasingly *bumpy* dictionary elements $\xi_{\ell k}(\cdot)$ for large ℓ in order to capture multiple resolutions of smoothness in the covariance regression. The prior on Θ would then encourage the smoother dictionary elements to be more prominent in forming $\Sigma(x)$, with the bumpier elements being more heavily regularized. One could, of course, also consider other dictionary element specifications such as based on basis expansions or with a finite autoregressive (band-limited covariance) structure. Such specifications could ameliorate some of the computational burden associated with Gaussian processes, but might induce different prior support for the covariance regression.

Likewise, just as we employed a shrinkage prior on Θ to be more robust to the choice of $\bar{L}$, one could similarly cope with $\bar{k}$ by considering an augmented formulation in which

$$\Lambda(x) = \Theta\xi(x)\Gamma, \tag{40}$$

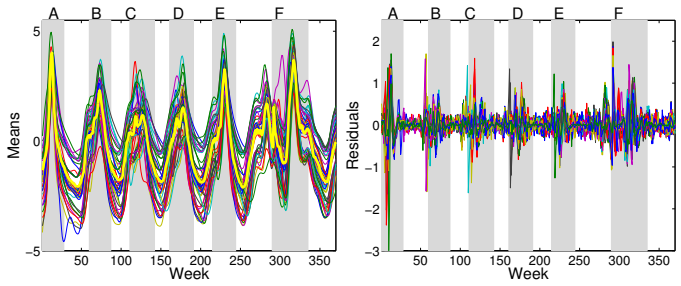
where $\Gamma = \text{diag}(\gamma_1, \ldots, \gamma_k)$ is a diagonal matrix of parameters that shrink the columns of $\xi(x)$ towards zero. One can take these shrinkage parameters to be distributed as

$$\gamma_i \sim \text{N}(0, \omega_i^{-1}), \quad \omega_i = \prod_{h=1}^{i} \zeta_h, \quad \zeta_1 \sim \text{Ga}(a_3, 1), \quad \zeta_h \sim \text{Ga}(a_4, 1) \quad h = 2, \ldots, k. \tag{41}$$

For $a_4 > 1$, such a model shrinks the γ_i values towards zero for large indices i just as in the shrinkage prior on Θ. Computations in this augmented model are a straightforward extension of the developed Gibbs sampler.

Appendix F: Simulation Studies

For the simulation studies of Case 2, the Wishart matrix discounting method to which we compared is given as follows, with details in Section 10.4.2 of Prado and West (2010). Let $\Phi_t = \Sigma_t^{-1}$. The Wishart matrix discounting model assumes $\Sigma_t^{-1} \mid y_{1:t-1}, \beta \sim W(\beta h_{t-1}, (\beta D_{t-1})^{-1})$, with $D_t = \beta D_{t-1} + y_t y_t'$ and $h_t = \beta h_{t-1} + 1$, such that $E[\Sigma_t^{-1} \mid y_{1:t-1}] = E[\Sigma_{t-1}^{-1} \mid y_{1:t-1}] = h_{t-1} D_{t-1}^{-1}$, but with certainty discounted by a factor determined by β. The update with observation y_t is conjugate, maintaining a Wishart posterior on Σ_t^{-1}. A limitation of this construction is that it constrains $h_t > p-1$ (or h_t integral) implying that $\beta > (p-2)/(p-1)$. We set $h_0 = 40$ and $\beta = 1 - 1/h_0$ such that $h_t = 40$ for all t and ran the forward filtering backward sampling (FFBS) algorithm outlined in Prado and West (2010), generating 100 independent samples. Increasing h_t can mitigate the large errors for high xs seen in Figure 4(b) and (d), but shrinks the model towards homoscedasticity. In general, the formulation is sensitive to the choice of h_t, and in high-dimensional problems this degree of freedom is forced to take large (or integral) values.



(a) (b)

Figure 11: (a) Plot of smoothing spline fits $\hat{f}_j(x)$ for each of the 183 Google Flu Trends regions. The thick yellow line indicates the empirical mean of the log Google-estimated ILI rates, $\log y_{ij}$, across regions j. (b) Residuals $\log r_{ij} - \hat{f}_j(x_i)$. The shaded gray regions indicate the flu events of Figure 5.

Appendix G: Exploratory Data Analysis

To examine the spatial correlation structure of the Google-estimated ILI rates and how these correlations vary across time, we performed the following exploratory data analysis. First, we consider a log transform of our rate data and a model

$$\log r_{ij} = f_j(x_i) + \epsilon_{ij}, \quad i = 1, \ldots, 370, \; j = 1, \ldots, 183, \tag{42}$$

with $f_j(\cdot)$ taken to be a region-specific smoothing spline. These spline fits are shown along with the residuals ϵ_{ij} in Figure 11. We then examine the spatial correlations of these residuals in Figure 12. We omit data prior to Event B because of the extent of missing values. Due to the dimensionality of the data (183 dimensions) and limited number of observations (157 event and 127 non-event observations), we simply consider state-level observations plus District of Columbia (resulting in 51 dimensions) and then aggregate the data over Events B-F to create a "flu event" maximum likelihood (ML) estimate, $\hat{\Sigma}^{flu}$. We likewise examine an aggregation of data between events to form a "non-event" estimate $\hat{\Sigma}^{nonflu}$.

From Figure 12, we see that the correlations between regions is much lower during non-event periods than event periods. For event periods, there is clear spatial correlation, defined both locally and with long-range dependencies. Note that because of the dimensionality and limited data, these exploratory methods cannot handle the full set of regions nor examine smoothly varying correlations. Instead, the plots simply provide insight into the geographic structure of the correlations and the fact that this structure is clearly different within and outside of flu event periods. As such, an i.i.d. model for $\boldsymbol{\epsilon}_i$ is inappropriate, motivating our heteroscedastic model of Section 2. In Section 5, we analyze how our proposed covariance regression model enables analysis of the changing extent and intensity of the correlations as a function of time. The method allows us to harness all of the available data, both across regions and time.

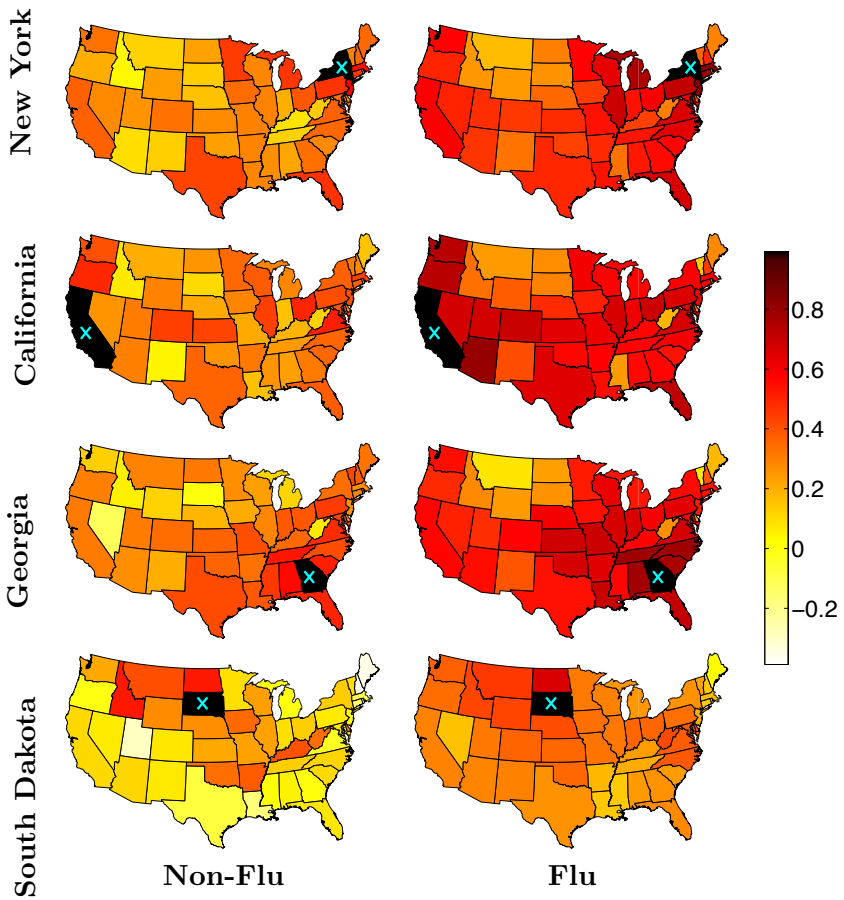


Figure 12: For each of four geographically distinct states (New York, California, Georgia, and South Dakota), plots of correlations between the state and all other states based on the sample covariance estimate from state-level data. The estimates are for data aggregated over non-event periods following event B (left) and event periods B-F (right). The data are taken to be the residuals of smoothing spline estimates fit independently for each region using log ILI rates. Event A was omitted due to an insufficient number of states reporting. Note that South Dakota is missing 58 of the 157 event B-F observations.

Appendix H: Details on Nadaraya-Watson Approach

Our proposed stochastic EM algorithm for the nonparametric Nadaraya-Watson kernel estimator iterates between (i) sampling missing values from the predictive distribution associated with the current kernel estimates of the mean and covariance functions and the available data, and (ii) computing the kernel-estimate of the mean and covariance functions using the available data and imputed missing values. We initialize by pooling all available data to form a static mean and covariance estimate from which the missing values are initially sampled. Due to the high-dimensionality compared to the limited bandwidth, we add a diagonal element $1e^{-6}I_p$ to the estimate $\hat{\Sigma}(x)$ to ensure positive definiteness.

References

H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu. Twitter improves seasonal influenza prediction. In *HEALTHINF*, pages 61–70, 2012.

R.P. Adams and I. Murray. Slice sampling covariance hyperparameters of latent Gaussian models. In *Advances in Neural Information Processing Systems*, volume 23, 2011.

J.H. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

S. Banerjee, A.E. Gelfand, A.O. Finley, and H. Sang. Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society, Series B*, 70(4):825–848, 2008.

A. Bhattacharya and D.B. Dunson. Sparse Bayesian infinite factor models. *Biometrika*, 98 (2):291–306, 2011.

S.P. Brooks and A. Gelman. General methods for monitoring convergence of iterative simulations. *Journal of Computational and Graphical Statistics*, 7:434–455, 1998.

J. S. Brownstein, C. J. Wolfe, and K. D. Mandl. Empirical evidence for the effect of airline travel on inter-regional influenza spread in the United States. *PLoS Medicine*, 3(10):e401, 2006.

D. Butler. When Google got flu wrong: US outbreak foxes a leading web-based method for tracking seasonal. *Nature*, 494:155–156, 2013.

CDC. Influenza vaccine effectiveness studies, January 2004. URL `http://www.cdc.gov/media/pressrel/fs040115.htm`.

V. Chandrasekaran, P. A. Parrilo, and A. S. Willsky. Latent variable graphical model selection via convex optimization. *Annals of Statistics*, 40(4):1935–1967, 2012.

S. Chib, Y. Omori, and M. Asai. Multivariate stochastic volatility. *Handbook of Financial Time Series*, pages 365–400, 2009.

T.Y.M. Chiu, T. Leonard, and K.W. Tsui. The matrix-logarithmic covariance model. *Journal of the American Statistical Association*, 91(433):198–210, 1996.

S. Cook, C. Conrad, A. L. Fowlkes, and M. H. Mohebbi. Assessing Google Flu Trends performance in the United States during the 2009 influenza virus A (H1N1) pandemic. *PLoS ONE*, 6(8):e23610, 2011.

J. Diebolt and E.H.S. Ip. *Markov Chain Monte Carlo in Practice*, chapter Stochastic EM: methods and application, pages 259–273. Chapman & Hall, 1995.

J. Du, H. Zhang, and V.S. Mandrekarm. Fixed-domain asymptotic properties of tapered maximum likelihood estimators. *the Annals of Statistics*, 37(6A):3330–3361, 2009.

V. Dukić, H.F. Lopes, and N.G. Polson. Tracking epidemics with Google Flu Trends data and a state-space SEIR model. *Journal of the American Statistical Assocaition*, 107(500): 1410–1426, 2012.

D. Durante, B. Scarpa, and D. B. Dunson. Locally adaptive factor processes for multivariate time series. *The Journal of Machine Learning Research*, 15(1):1493–1522, 2014.

R. Engle. New frontiers for ARCH models. *Journal of Applied Econometrics*, 17(5):425–446, 2002.

B. K. Fosdick and P. D. Hoff. Separable factor analysis with applications to mortality data. *Annals of Applied Statistics*, 8(1):120–147, 2014.

E. B. Fox and D. B. Dunson. Bayesian nonparametric covariance regression. *arXiv preprint arXiv:1101.2017*, 2011.

W. A. Fuller. *Introduction to Statistical Time Series*, volume 428. John Wiley & Sons, 2009.

A.E. Gelfand, A.M. Schmidt, S. Banerjee, and C.F. Sirmans. Nonstationary multivariate process modeling through spatially varying coregionalization. *Test*, 13(2):263–312, 2004.

J. Ginsberg, M.H. Mohebbi, R.S. Patel, L. Brammer, M.S. Smolinski, and L. Brilliant. Detecting influenza epidemics using search engine query data. *Nature*, 457(7232):1012–1014, 2008.

C. Gouriéroux, J. Jasiak, and R. Sufana. The Wishart autoregressive process of multivariate stochastic volatility. *Journal of Econometrics*, 150(2):167–181, 2009.

R. Harris. Google's flu tracker suffers from sniffles. *http://www.npr.org/blogs/health/2014/03/13/289802934/googles-flu-tracker-suffers-from-sniffles*, March 2014.

A.C. Harvey, E. Ruiz, and N. Shephard. Multivariate stochastic variance models. *Review of Economic Studies*, 61:247–264, 1994.

D. Higdon, C. Nakhleh, J. Gattiker, and B. Williams. A Bayesian calibration approach to the thermal problem. *Computer Methods in Applied Mechanics and Engineering*, 197 (29):2431–2441, 2008.

P. D. Hoff and X. Niu. A covariance regression model. *Statistica Sinica*, 22:729–753, 2012.

P.D. Hoff. Extending the rank likelihood for semiparametric copula estimation. *Annals of Applied Statistics*, 1(1):265–283, 2007.

M. B. Hooten, J. Anderson, and L. A. Waller. Assessing North American influenza dynamics with a statistical SIRS model. *Spatial and Spatio-temporal Epidemiology*, 1(2):177–185, 2010.

C.G. Kaufman, M.J. Schervish, and D.W. Nychka. Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103 (484):1545–1555, 2008.

A. Lamb, M. J. Paul, and M. Dredze. Separating fact from fear: Tracking flu infections on Twitter. In *Proceedings of NAACL-HLT*, pages 789–795, 2013.

D. Lazer, R. Kennedy, G. King, and A. Vespignani. The parable of Google Flu: Traps in big data analysis. *Science*, 343(6176):1203–1205, 2014.

C. Leng, W. Zhang, and J. Pan. Semiparametric mean-covariance regression analysis for longitudinal data. *Journal of the American Statistical Association*, 105(489):181–193, 2010.

J.S. Liu, W.H. Wong, and A. Kong. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81(1): 27–40, 1994.

H.F. Lopes, E. Salazar, and D. Gamerman. Spatial dynamic factor analysis. *Bayesian Analysis*, 3(4):759–792, 2008.

M.A. Martínez-Beneito, D. Conesa, A. López-Quílez, and A. López-Maside. Bayesian Markov switching models for the early detection of influenza epidemics. *Statistics in Medicine*, 27(22):4455–4468, 2008.

A. S. Mugglin, N. Cressie, and I. Gemmell. Hierarchical statistical modelling of influenza epidemic dynamics in space and time. *Statistics in Medicine*, 21(18):2703–2721, 2002.

R. Paulo. Default priors for Gaussian processes. *Annals of Statistics*, pages 556–582, 2005.

A. Philipov and M.E. Glickman. Multivariate stochastic volatility via Wishart processes. *Journal of Business & Economic Statistics*, 24(3):313–328, 2006a.

A. Philipov and M.E. Glickman. Factor multivariate stochastic volatility via Wishart processes. *Econometric Reviews*, 25(2-3):311–334, 2006b.

M. Pourahmadi. Joint mean-covariance models with applications to longitudinal data: Unconstrained parameterisation. *Biometrika*, 86(3):677–690, 1999.

R. Prado and M. West. *Time Series: Modeling, Computation, and Inference*. Chapman & Hall / CRC, Boca Raton, FL, 2010.

C. E. Rasmussen and C. K. .I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society, Series B*, 71(2):319–392, 2009.

T. Sakai, H. Suzuki, A. Sasaki, R. Saito, N. Tanabe, and K. Taniguchi. Geographic and temporal trends in influenzalike illness, Japan, 1992-1999. *Emerging Infectious Diseases*, 10(10):1822–1826, 2004.

J. H. Stark, R. Sharma, S. Ostroff, D. A. T. Cummings, B. Ermentrout, S. Stebbins, D. S. Burke, and S. R. Wisniewski. Local spatial and temporal processes of influenza in Pennsylvania, USA: 2003–2009. *PLoS ONE*, 7(3):e34245, 2012.

C. Viboud, P. Y. Boëlle, K. Pakdaman, F. Carrat, A. J. Valleron, A. Flahault, et al. Influenza epidemics in the United States, France, and Australia, 1972-1997. *Emerging Infectious Diseases*, 10(1):32–39, 2004.

M. West. Bayesian factor regression models in the "large p, small n" paradigm. *Bayesian Statistics*, 7:723–732, 2003.

A. G. Wilson and Z. Ghahramani. Generalized Wishart processes. In *Uncertainty in Artificial Intelligence*, 2011.

J. Yin, Z. Geng, R. Li, and H. Wang. Nonparametric covariance model. *Statistica Sinica*, 20:469–479, 2010.

W. Zhang and C. Leng. A moving average Cholesky factor model in covariance modelling for longitudinal data. *Biometrika*, 99(1):141–150, 2012.

Journal of Machine Learning Research 16 (2015) 2543-2588 Submitted 8/14; Revised 5/15; Published 12/15

A General Framework for Fast Stagewise Algorithms

Ryan J. Tibshirani RYANTIBS@STAT.CMU.EDU
Departments of Statistics and Machine Learning
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Editor: Bin Yu

Abstract

Forward stagewise regression follows a very simple strategy for constructing a sequence of sparse regression estimates: it starts with all coefficients equal to zero, and iteratively updates the coefficient (by a small amount ϵ) of the variable that achieves the maximal absolute inner product with the current residual. This procedure has an interesting connection to the lasso: under some conditions, it is known that the sequence of forward stagewise estimates exactly coincides with the lasso path, as the step size ϵ goes to zero. Furthermore, essentially the same equivalence holds outside of least squares regression, with the minimization of a differentiable convex loss function subject to an ℓ_1 norm constraint (the stagewise algorithm now updates the coefficient corresponding to the maximal absolute component of the gradient).

Even when they do not match their ℓ_1-constrained analogues, stagewise estimates provide a useful approximation, and are computationally appealing. Their success in sparse modeling motivates the question: can a simple, effective strategy like forward stagewise be applied more broadly in other regularization settings, beyond the ℓ_1 norm and sparsity? The current paper is an attempt to do just this. We present a general framework for stagewise estimation, which yields fast algorithms for problems such as group-structured learning, matrix completion, image denoising, and more.

Keywords: forward stagewise regression, lasso, ϵ-boosting, regularization paths

1. Introduction

In a regression setting, let $y \in \mathbb{R}^n$ denote an outcome vector and $X \in \mathbb{R}^{n \times p}$ a matrix of predictor variables, with columns $X_1, \ldots X_p \in \mathbb{R}^n$. For modeling y as a linear function of X, we begin by considering (among the many possible candidates for sparse estimation tools) a simple method: *forward stagewise regression.* In plain words, forward stagewise regression produces a sequence of coefficient estimates $\beta^{(k)}$, $k = 0, 1, 2, \ldots$, by iteratively decreasing the maximal absolute inner product of a variable with the current residual, each time by only a small amount. A more precise description of the algorithm is as follows.

Algorithm 1 (Forward stagewise regression)

Fix $\epsilon > 0$, initialize $\beta^{(0)} = 0$, and repeat for $k = 1, 2, 3, \ldots$,

$$\beta^{(k)} = \beta^{(k-1)} + \epsilon \cdot \mathrm{sign}\big(X_i^T (y - X\beta^{(k-1)})\big) \cdot e_i, \tag{1}$$

$$\textit{where } i \in \operatorname*{argmax}_{j=1,\ldots p} |X_j^T (y - X\beta^{(k-1)})|. \tag{2}$$

In the above, $\epsilon > 0$ is a small fixed constant (e.g., $\epsilon = 0.01$), commonly referred to as the step size or learning rate; e_i denotes the ith standard basis vector in $\mathbb{R}^p$; and the element notation in (2) emphasizes that the maximizing index i need not be unique. The basic idea behind the forward stagewise updates (1), (2) is highly intuitive: at each iteration we greedily select the variable i that has the largest absolute inner product (or correlation, for standardized variables) with the residual, and we add $s_i\epsilon$ to its coefficient, where s_i is the sign of this inner product. Accordingly, the fitted values undergo the update:

$$X\beta^{(k)} = X\beta^{(k-1)} + \epsilon \cdot s_i X_i.$$

Such greediness, in selecting variable i, is counterbalanced by the small step size $\epsilon > 0$; instead of increasing the coefficient of X_i by a (possibly) large amount in the fitted model, forward stagewise only increases it by ϵ, which "slows down" the learning process. As a result, it typically requires many iterations to produce estimates of reasonable interest with forward stagewise regression, e.g., it could easily take thousands of iterations to reach a model with only tens of active variables (we use "active" here to refer to variables that are assigned nonzero coefficients). See the left panel of Figure 1 for a small example.

This "slow learning" property is a key difference between forward stagewise regression and the closely-named *forward stepwise regression* procedure: at each iteration, the latter algorithm chooses a variable in a similar manner to that in (2)[1], but once it does so, it updates the fitted model by regressing y on all variables selected thus far. While both are greedy algorithms, the stepwise procedure is much greedier; after k iterations, it produces a model with exactly k active variables. Forward stagewise and forward stepwise are old techniques (some classic references for stepwise regression methods are Efroymson, 1966 and Draper and Smith, 1966, but there could have been earlier relevant work). According to Hastie et al. (2009), forward stagewise was historically dismissed by statisticians as being "inefficient" and hence less useful than methods like forward or backward stepwise. This is perhaps understandable, if we keep in mind the limited computational resources of the time. From a modern perspective, however, we now appreciate that "slow learning" is a form of regularization and can present considerable benefits in terms of the generalization error of the fitted models—this is seen not only in regression, but across variety of settings. Furthermore, by modern standards, forward stagewise is computationally cheap: to trace out a path of regularized estimates, we repeat very simple iterations, each one requiring (at most) p inner products, computations that could be trivially parallelized.

The revival of interest in stagewise regression began with the work of Efron et al. (2004), where the authors derived a surprising connection between the sequence of forward stagewise estimates and the solution path of the *lasso* (Tibshirani, 1996),

$$\hat{\beta}(t) = \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \ \frac{1}{2}\|y - X\beta\|_2^2 \ \text{ subject to } \ \|\beta\|_1 \leq t, \tag{3}$$

over the regularization parameter $t \geq 0$. The relationship between stagewise and the lasso will be reviewed in Section 2.1 in detail, but the two panels in Figure 1 tell the essence

1. If A denotes the active set at the end of iteration $k-1$, then at iteration k forward stepwise chooses the variable i such that the sum of squared errors from regressing y onto the variables in $A \cup \{i\}$ is smallest. This is equivalent to choosing i such that $|\widetilde{X}_i^T(y - X\beta^{(k-1)})|$ is largest, where $\beta^{(k-1)}$ denote the coefficients from regressing y on the variables in A, and $\widetilde{X}_i$ is the residual from regressing X_i on the variables in A.

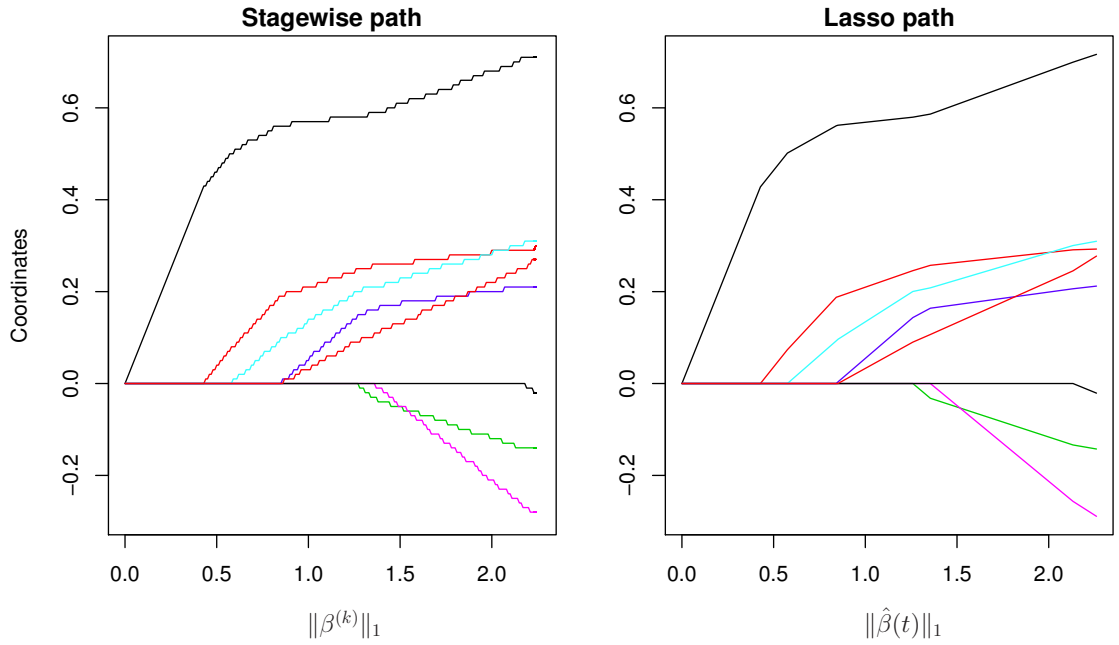


Figure 1: A simple example using the prostate cancer data from Hastie et al. (2009), where the log PSA score of $n = 67$ men with prostate cancer is modeled as a linear function of $p = 8$ biological predictors. The left panel shows the forward stagewise regression estimates $\beta^{(k)} \in \mathbb{R}^8$, $k = 1, 2, 3, \ldots$, with the 8 coordinates plotted in different colors. The stagewise algorithm was run with $\epsilon = 0.01$ for 250 iterations, and the x-axis here gives the ℓ_1 norm of the estimates across iterations. The right panel shows the lasso solution path, also parametrized by the ℓ_1 norm of the estimate. The similarity between the stagewise and lasso paths is visually striking; for small enough ϵ, they appear identical. This is not a coincidence and has been rigorously studied by Efron et al. (2004), and other authors; in Section 2.1 we provide an intuitive explanation for this phenomenon.

of the story. The stagewise paths, on the left, appear to be jagged versions of their lasso counterparts, on the right. Indeed, as the step size ϵ is made smaller, this jaggedness becomes less noticeable, and eventually the two sets of paths appear exactly the same. This is not a coincidence, and under some conditions (on the problem instance in consideration), it is known that the stagewise path converges to the lasso path, as $\epsilon \to 0$. Interestingly, when these conditions do not hold, stagewise estimates can deviate substantially from lasso solutions, and yet in such situations the former estimates can still perform competitively with the latter, say, in terms of test error (or really any other standard error metric). This is an important point, and it supports the use of stagewise regression as a general tool for regularized estimation.

1.1 Summary of Our Contributions

This paper departs from the lasso setting and considers the generic convex problem

$$\hat{x}(t) \in \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) \;\; \text{subject to} \;\; g(x) \leq t, \tag{4}$$

where $f, g : \mathbb{R}^n \to \mathbb{R}$ are convex functions, and f is differentiable. Motivated by forward stagewise regression and its connection to the lasso, our main contribution is the following *general stagewise algorithm* for producing an approximate solution path of (4), as the regularization parameter t varies over $[t_0, \infty)$.

Algorithm 2 (General stagewise procedure)

Fix $\epsilon > 0$ and $t_0 \in \mathbb{R}$. Initialize $x^{(0)} = \hat{x}(t_0)$, a solution in (4) at $t = t_0$. Repeat, for $k = 1, 2, 3, \ldots$,

$$x^{(k)} = x^{(k-1)} + \Delta, \tag{5}$$

$$\text{where } \Delta \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \langle \nabla f(x^{(k-1)}), z \rangle \text{ subject to } g(z) \leq \epsilon. \tag{6}$$

The intuition behind the general stagewise algorithm can be seen right away: at each iteration, we update the current iterate in a direction that minimizes the inner product with the gradient of f (evaluated at the current iterate), but simultaneously restrict this direction to be small under g. By applying these updates repeatedly, we implicitly adjust the trade-off between minimizing f and g, and hence one can imagine that the kth iterate $x^{(k)}$ approximately solves (4) with $t = g(x^{(k)})$. In Figure 2, we show a few simple examples of the general stagewise paths implemented for various different choices of loss functions f and regularizing functions g.

In the next section, we develop further intuition and motivation for the general stagewise procedure, and we tie in forward stagewise regression as a special case. The rest of this article is then dedicated to the implementation and analysis of stagewise algorithms: Section 3 derives the specific form of the stagewise updates (5), (6) for various problem setups, Section 4 conducts large-scale empirical evaluations of stagewise estimates, Section 5 presents some theory on suboptimality, and Section 6 concludes with a discussion.

Throughout, our arguments and examples are centered around three points, summarized below.

1. *Simple, fast estimation procedures.* The general framework for stagewise estimation in Algorithm 2 leads to simple and efficient stagewise procedures for group-structured regularization problems (e.g., the group lasso, multitask learning), trace norm regularization problems (e.g., matrix completion), quadratic regularization problem problems (e.g., nonparametric smoothing), and (some) generalized lasso problems (e.g., image denoising). For such problems, the proposed stagewise procedures are often competitive with existing commonly-used algorithms in terms of efficiency, and are generally much simpler.

2. *Similar to actual solution paths, but more stable.* In many examples, the computed stagewise path is highly similar to the actual solution path of the corresponding convex regularization problem in (4)—typically, this happens when the components of the actual solution change "slowly" with the regularization parameter t. In many others, even though it shares gross characteristics of the actual solution path, the stagewise path is different—typically, this happens when the components of the actual solution change "rapidly" with t, and the stagewise component paths are much more stable.

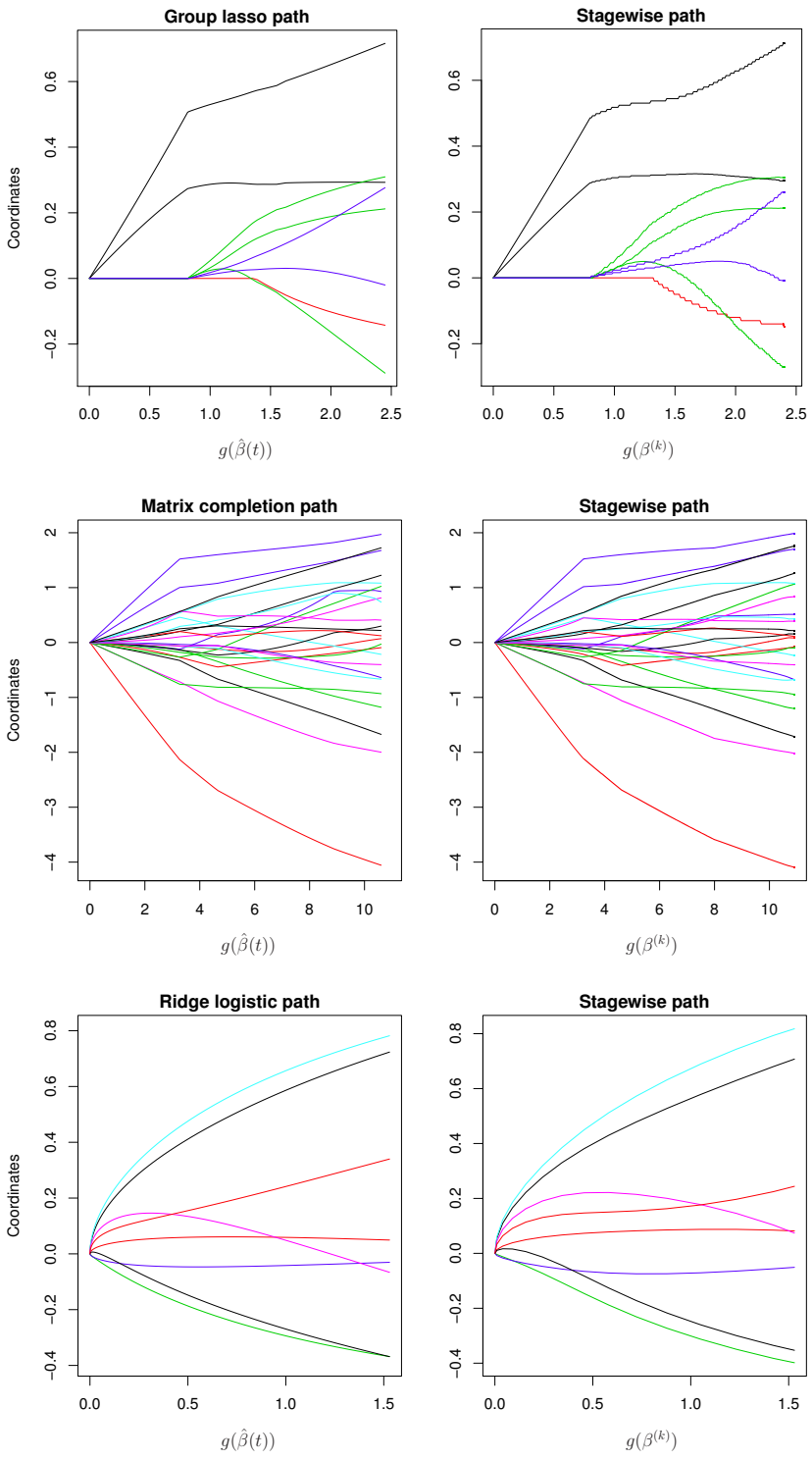


Figure 2: Examples comparing the actual solution paths (left column) to the stagewise paths (right column) across various problem contexts, using the prostate cancer data set. The first row considers a group lasso model on the prostate data (where the groups were somewhat arbitrarily chosen based on the predictor types); the second row considers a matrix completion task, on a partially observed submatrix of the full predictor matrix; the third row considers a logistic regression model with ridge regularization (the outcome being the indicator of log PSA > 1). In each case, the stagewise estimates were very easy to compute; Sections 3.1, 3.3, and 3.4 discuss these problem settings in detail.

3. *Competitive statistical performance.* Across essentially all cases, even those in which its constructed path is not close to the actual solution path, the stagewise algorithm performs favorably from a statistical point of view. That is, stagewise estimates are comparable to solutions in (4) with respect to relevant error metrics, across various problem settings. This suggests that stagewise estimates deserved to be studied on their own, regardless of their proximity to solutions in (4).

The third point above, on the favorable statistical properties of stagewise estimates, is based on empirical arguments, rather than theoretical ones. Statistical theory for stagewise estimates is an important topic for future work.

2. Properties of the General Stagewise Framework

For motivation and background, we cover the connection between stagewise regression and the lasso in more detail, and then rewrite the stagewise regression updates in a form that naturally suggests the general stagewise proposal of this paper. Following this, we discuss properties of the general stagewise framework, and related work.

2.1 Motivation: Stagewise Regression and the Lasso

The lasso estimator is a popular tool for sparse estimation in the regression setting. Displayed in (3), we assume for simplicity that the lasso solution $\hat{\beta}(t)$ in (3) is unique, which holds under very weak conditions on X.[2] Recall that the parameter t controls the level of sparsity in the estimate $\hat{\beta}(t)$: when $t = 0$, we have $\hat{\beta}(0) = 0$, and as t increases, select components of $\hat{\beta}(t)$ become nonzero, corresponding to variables entering the lasso model (nonzero components of $\hat{\beta}(t)$ can also become zero, corresponding to variables leaving the model). The solution path $\hat{\beta}(t)$, $t \in [0, \infty)$ is continuous and piecewise linear as a function of t, and for a large enough value of t, the path culminates in a least squares estimate of y on X.

The right panel of Figure 1 shows an example of the lasso path, which, as we discussed earlier, appears quite similar to the stagewise path on the left. This is explained by the seminal work of Efron et al. (2004), who describe two algorithms (actually three, but the third is unimportant for our purposes): one for explicitly constructing the lasso path $\hat{\beta}(t)$ as a continuous, piecewise linear function of the regularization parameter $t \in [0, \infty)$, and another for computing the limiting stagewise regression paths as $\epsilon \to 0$. One of the (many) consequences of their work is the following: if each component of the lasso solution path $\hat{\beta}(t)$ is a monotone function of t, then these two algorithms coincide, and therefore so do the stagewise and lasso paths (in the limit as $\epsilon \to 0$). Note that the lasso paths for the data example in Figure 1 are indeed monotone, and hence the theory confirms the observed convergence of stagewise and lasso estimates in this example.

The lasso has undergone intense study as a regularized regression estimator, and its statistical properties (e.g., its generalization error, or its ability to detect a truly relevant set of variables) are more or less well-understood at this point. Many of these properties cast

2. For example, it suffices to assume that X has columns in general position, see Tibshirani (2013). Note that here we are only claiming uniqueness for all parameter values $t < t^*$, where t^* is the smallest ℓ_1 norm of a least squares solution of y on X.

the lasso in a favorable light. Therefore, the equivalence between the (limiting) stagewise and lasso paths lends credibility to forward stagewise as a regularized regression procedure: for a small step size ϵ, we know that the forward stagewise estimates will be close to lasso estimates, at least when the individual coordinate paths are monotone. At a high level, it is actually somewhat remarkable that such a simple algorithm, Algorithm 1, can produce estimates that can stand alongside those defined by the (relatively) sophisticated optimization problem in (3). There are now several interesting points to raise.

- *The nonmonotone case.* In practice, the components of the lasso path are rarely monotone. How do the stagewise and lasso paths compare in such cases? A precise theoretical answer is not known, but empirically, these paths can be quite different. In particular, for problems in which the predictors $X_1, \ldots X_p$ are correlated, the lasso coordinate paths can be very wiggly (as variables can enter and leave the model repeatedly), while the stagewise paths are often very stable; see, e.g., Hastie et al. (2007). In support of these empirical findings, the latter authors derived a local characterization of the lasso and forward stagewise paths: they show that at any point along the path, the lasso estimate decreases the sum of squares loss function at an optimal rate with respect to the increase in ℓ_1 norm, and the (limiting) forward stagewise estimate decreases the loss function at an optimal rate with respect to the increase in ℓ_1 arc length. Loosely speaking, since the ℓ_1 arc length accounts for the entire history of the path up until the current point, the (limiting) stagewise algorithm is less "willing" to produce wiggly estimates.

  Despite these differences, stagewise estimates tend to perform competitively with lasso estimates in terms of test error, and this is true even with highly correlated predictor variables, when the stagewise and lasso paths are very different (such statements are based on simulations, and not theory; see Hastie et al., 2007; Knudsen, 2013). This is a critical point, as it suggests that stagewise should be considered as an effective tool for regularized estimation, apart from any link to a convex problem. We return to this idea throughout the paper.

- *General convex loss functions.* Fortunately, the stagewise method extends to sparse modeling in other settings, beyond Gaussian regression. Let $f : \mathbb{R}^p \to \mathbb{R}$ be a differentiable convex loss function, e.g., $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ for the regression setting. Beginning again with $\beta^{(0)} = 0$, the analogy of the stagewise steps in (1), (2) for the present general setting are

$$\beta^{(k)} = \beta^{(k-1)} - \epsilon \cdot \mathrm{sign}\big(\nabla_i f(\beta^{(k-1)})\big) \cdot e_i, \tag{7}$$

$$\text{where } i \in \operatorname*{argmax}_{j=1,\ldots p} |\nabla_j f(\beta^{(k-1)})|. \tag{8}$$

  That is, at each iteration we update $\beta^{(k)}$ in the direction opposite to the largest component of the gradient (largest in absolute value). Note that this reduces to the usual update rules (1), (2) when $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$. Rosset et al. (2004) studied the stagewise routine (7), (8), and its connection to the ℓ_1-constrained estimate

$$\hat{\beta}(t) = \operatorname*{argmin}_{\beta \in \mathbb{R}^p} f(\beta) \;\text{ subject to }\; \|\beta\|_1 \leq t. \tag{9}$$

Similar to the result for lasso regression, these authors prove that if the solution $\hat{\beta}(t)$ in (9) has monotone coordinate paths, then under mild conditions[3] on f, the stagewise paths given by (7), (8) converge to the path $\hat{\beta}(t)$ as $\epsilon \to 0$. This covers, e.g., the cases of logistic regression and Poisson regression losses, with predictor variables X in general position. The same general message, as in the linear regression setting, applies here: compared to the relatively complex optimization problem (9), the stagewise algorithm (7), (8) is very simple. The most (or really, the only) advanced part of each iteration is the computation of the gradient $\nabla f(\beta^{(k-1)})$; in the logistic or Poisson regression settings, the components of $\nabla f(\beta^{(k-1)})$ are given by

$$\nabla_j f(\beta^{(k-1)}) = X_j^T \big(y - \mu(\beta^{(k-1)})\big), \quad j = 1, \ldots p,$$

where $y \in \mathbb{R}^n$ is the outcome and $\mu(\beta^{(k-1)}) \in \mathbb{R}^n$ has components

$$\mu_i(\beta^{(k-1)}) = \begin{cases} 1/[1 + \exp(-(X\beta^{(k-1)})_i)] & \text{for logistic regression} \\ \exp((X\beta^{(k-1)})_i) & \text{for Poisson regression} \end{cases}, \quad i = 1, \ldots n.$$

Its precise connection to the ℓ_1-constrained optimization problem (9) for monotone paths is encouraging, but even outside of this case, the simple and efficient stagewise algorithm (7), (8) produces regularized estimates deserving of attention in their own right.

- *Forward-backward stagewise.* Zhao and Yu (2007) examined a novel modification of forward stagewise, under a general loss function f: at each iteration, their proposal takes a backward step (i.e., moves a component of $\beta^{(k)}$ towards zero) if this would decrease the loss function by a sufficient amount ξ; otherwise it takes a forward step as usual. The authors prove that, as long as the parameter ξ used for the backward steps scales as $\xi = o(\epsilon)$, the path from this forward-backward stagewise algorithm converges to the solution path in (9) as $\epsilon \to 0$. The important distinction here is that their result does not assume monotonicity of the coordinate paths in (9). (It does, however, assume that the loss function f is strongly convex—in the linear regression setting, $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$, this is equivalent to assuming that $X \in \mathbb{R}^{n \times p}$ has linearly independent predictors, which requires $n \geq p$).[4] The forward-backward stagewise algorithm hence provides another way to view the connection between (the usual) forward stagewise steps (7), (8) and the ℓ_1-regularized optimization problem (9): the forward stagewise path is an approximation to the solution path in (9) given by skipping the requisite backward steps needed to correct for nonmonotonicities.

Clearly, there has been some fairly extensive work connecting the stagewise estimates (1), (2) and the lasso estimate (3), or more generally, the stagewise estimates (7), (8) and

3. Essentially, Rosset et al. (2004) assume that conditions on f that imply a unique solution in (9), and allow for a second order Taylor expansion of f. Such conditions are that $f(\beta) = h(X\beta)$, with h twice differentiable and strictly convex, and X having columns in general position.
4. It is also worth pointing out that the type of convergence considered by Zhao and Yu (2007) is stronger than that considered by Efron et al. (2004) and Rosset et al. (2004). The former authors prove that, under suitable conditions, the entire stagewise path converges globally to the lasso solution path; the latter authors only prove a local type of convergence, that has to do with the limiting stagewise and lasso directions at any fixed point along the path.

the ℓ_1-constrained estimate (9). Still, however, this connection seems mysterious. Both methods produce a regularization path, with a fully sparse model on one end, and a fully dense model on the other—but beyond this basic degree of similarity, why should we expect the stagewise path (7), (8) and the ℓ_1 regularization path (9) to be so closely related? The work referenced above gives a mathematical treatment of this question, and we feel, does not provide much intuition. In fact, there is a simple interpretation of the forward stagewise algorithm that explains its connection to the lasso problem, seen next.

2.2 A New Perspective on Forward Stagewise Regression

We start by rewriting the steps (7), (8) for the stagewise algorithm, under a general loss f, as

$$\begin{gathered}\beta^{(k)} = \beta^{(k-1)} + \Delta, \\ \text{where } \Delta = -\epsilon \cdot \operatorname{sign}\big(\nabla_i f(\beta^{(k-1)})\big) \cdot e_i, \\ \text{and } |\nabla_i f(\beta^{(k-1)})| = \|\nabla f(\beta^{(k-1)})\|_\infty.\end{gathered}$$

As $\nabla_i f(\beta^{(k-1)})$ is maximal in absolute value among all components of the gradient, the quantity $\operatorname{sign}(\nabla_i f(\beta^{(k-1)})) \cdot e_i$ is a subgradient of the ℓ_∞ norm evaluated at $\nabla f(\beta^{(k-1)})$:

$$\Delta \in -\epsilon \cdot \Big(\partial \|x\|_\infty \Big|_{x=\nabla f(\beta^{(k-1)})}\Big).$$

Using the duality between the ℓ_∞ and ℓ_1 norms,

$$\Delta \in -\epsilon \cdot \Big(\operatorname*{argmax}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z\rangle \;\text{ subject to }\; \|z\|_1 \le 1\Big),$$

or equivalently,

$$\Delta \in \operatorname*{argmin}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z\rangle \;\text{ subject to }\; \|z\|_1 \le \epsilon.$$

(Above, as before, the element notation emphasizes that the maximizer or minimizer is not necessarily unique.) Hence the forward stagewise steps (7), (8) satisfy

$$\beta^{(k)} = \beta^{(k-1)} + \Delta, \tag{10}$$

$$\text{where } \Delta \in \operatorname*{argmin}_{z \in \mathbb{R}^p} \langle \nabla f(\beta^{(k-1)}), z\rangle \;\text{ subject to }\; \|z\|_1 \le \epsilon. \tag{11}$$

Written in this form, the stagewise algorithm exhibits a natural connection to the ℓ_1-regularized optimization problem (9). At each iteration, forward stagewise moves in a direction that minimizes the inner product with the gradient of f, among all directions constrained to have a small ℓ_1 norm; therefore, the sequence of stagewise estimates balance (small) decreases in the loss function f with (small) increases in the ℓ_1 norm, just like the solution path in (9), as the regularization parameter t increases. This intuitive perspective aside, the representation (10), (11) for the forward stagewise estimates is important because it inspires an analogous approach for general convex regularization problems. This was already presented in Algorithm 2, and next we discuss it further.

2.3 Basic Properties of the General Stagewise Procedure

Recall the general minimization problem in (4), where we assume that the loss function f is convex and differentiable, and the regularizer g is convex. It can now be seen that the steps (5), (6) in the general stagewise procedure in Algorithm 2 are directly motivated by the forward stagewise steps, as expressed in (10), (11). The explanation is similar to that given above: as we repeat the steps of the algorithm, the iterates are constructed to decrease the loss function f (by following its negative gradient) at the cost of a small increase in the regularizer g. In this sense, the stagewise algorithm navigates the trade-off between minimizing f and g, and produces an approximate regularization path for (4), i.e., the kth iterate $x^{(k)}$ approximately solves problem (4) with $t = g(x^{(k)})$.

From our work at the end of the last subsection, it is clear that forward stagewise regression (7), (8), or equivalently (10), (11), is a special case of the general stagewise procedure, applied to the ℓ_1-regularized problem (9). Moreover, the general stagewise procedure can be applied in many other settings, well beyond ℓ_1 regularization, as we show in the next section. Before presenting these applications, we now make several basic remarks.

- *Initialization and termination.* In many cases, initializing the algorithm is easy: if $g(x) = 0$ implies $x = 0$ (e.g., this is true when g is a norm), then we can start the stagewise procedure at $t_0 = 0$ and $x^{(0)} = 0$. In terms of a stopping criterion, a general strategy for (approximately) tracing a full solution path is to stop the algorithm when $g(x^{(k)})$ does not change very much between successive iterations. If instead the algorithm has been terminated upon reaching some maximum number of iterations or some maximum value of $g(x^{(k)})$, and more iterations are desired, then the algorithm can surely be restarted from the last reached iterate $x^{(k)}$.

- *First-order justification.* If g satisfies the triangle inequality (again, e.g., it would as a norm), then the increase in the value of g between successive iterates is bounded by ϵ:

$$g(x^{(k)}) \leq g(x^{(k-1)}) + g(\Delta) \leq g(x^{(k-1)}) + \epsilon.$$

  Furthermore, we can give a basic (and heuristic) justification of the stagewise steps (5), (6). Consider the minimization problem (4) at the parameter $t = g(x^{(k-1)}) + \epsilon$; we can write this as

$$\hat{x}(t) \in \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) - f(x^{(k-1)}) \;\text{ subject to }\; g(x) - g(x^{(k-1)}) \leq \epsilon,$$

  and then reparametrize as

$$\hat{x}(t) = x^{(k-1)} + \Delta^*, \tag{12}$$

$$\Delta^* \in \operatorname*{argmin}_{z \in \mathbb{R}^n} f(x^{(k-1)} + z) - f(x^{(k-1)}) \;\text{ subject to }\; g(x^{(k-1)} + z) - g(x^{(k-1)}) \leq \epsilon. \tag{13}$$

  We now modify the problem (13) in two ways: first, we replace the objective function in (13) with its first-order (linear) Taylor approximation around $x^{(k-1)}$,

$$\langle \nabla f(x^{(k-1)}), z \rangle \approx f(x^{(k-1)} + z) - f(x^{(k-1)}), \tag{14}$$

and second, we shrink the constraint set in (13) to

$$\{z \in \mathbb{R}^n : g(z) \leq \epsilon\} \subseteq \{z \in \mathbb{R}^n : g(x^{(k-1)} + z) - g(x^{(k-1)}) \leq \epsilon\},$$

since, as noted earlier, any element of the left-hand side above is an element of the right-hand side by the triangle inequality. These two modifications define a different update direction

$$\Delta \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \langle \nabla f(x^{(k-1)}), z \rangle \;\text{ subject to }\; g(z) \leq \epsilon,$$

which is exactly the direction (6) in the general stagewise procedure. Hence the stagewise algorithm chooses Δ as above, rather than choosing the actual direction Δ^* in (13), to perform an update step from $x^{(k-1)}$. This update results in a feasible point $x^{(k)} = x^{(k-1)} + \Delta$ for the problem (4) at $t = g^{(k-1)} + \epsilon$; of course, the point $x^{(k)}$ is not necessarily optimal, but as ϵ gets smaller, the first-order Taylor approximation in (14) becomes tighter, so one would imagine that the point $x^{(k)}$ becomes closer to optimal.

- *Dual update form.* If g is a norm, then the update direction defined in (6) can be expressed more succinctly in terms of the dual norm $g^*(x) = \max_{g(z) \leq 1} x^T z$. We write

$$\begin{aligned} \Delta &\in -\epsilon \cdot \Big(\operatorname*{argmax}_{z \in \mathbb{R}^n} \langle \nabla f(x^{(k-1)}), z \rangle \;\text{ subject to }\; g(z) \leq 1 \Big) \\ &= -\epsilon \cdot \partial g^* \big(\nabla f(x^{(k-1)}) \big), \end{aligned} \tag{15}$$

  i.e., the direction Δ is $-\epsilon$ times a subgradient of the dual norm g^* evaluated at $\nabla f(x^{(k-1)})$. This is a useful observation, since many norms admit a known dual norm with known subgradients; we will see examples of this in the coming section.

- *Invariance around ∇f.* The level of difficulty associated with computing the update direction, i.e., in solving problem (6), depends entirely on g and not on f at all (assuming that ∇f can be readily computed). We can think of Δ as an operator on $\mathbb{R}^n$:

$$\Delta(x) \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \langle x, z \rangle \;\text{ subject to }\; g(z) \leq \epsilon. \tag{16}$$

  This operator $\Delta(\cdot)$ is often called the *linear minimization oracle* associated with the function g, in the optimization literature. At each input x, it returns a minimizer of the problem in (16). Provided that $\Delta(\cdot)$ can be expressed in closed-form—which is fortuitously the case for many common statistical optimization problems, as we will see in the sections that follow—the stagewise update step (5) simply evaluates this operator at $\nabla f(x^{(k-1)})$, and adds the result to $x^{(k-1)}$:

$$x^{(k)} = x^{(k-1)} + \Delta\big(\nabla f(x^{(k-1)})\big).$$

  An analogy can be drawn here to the proximal operator in proximal gradient descent, used for minimizing the composite function $f+g$, where f is smooth but g is (possibly) nonsmooth. The proximal operator is defined entirely in terms of g, and as long as it can be expressed analytically, the generalized gradient update for $x^{(k)}$ simply uses the output of this operator at $\nabla f(x^{(k-1)})$.

- *Unbounded stagewise steps.* Suppose that g is a seminorm, i.e., it satisfies $g(ax) = |a|g(x)$ for $a \in \mathbb{R}$, and $g(x+y) \leq g(x) + g(y)$, but g can have a nontrivial null space, $N_g = \{x \in \mathbb{R}^n : g(x) = 0\}$. In this case, the stagewise update step in (5) can be unbounded; in particular, if

$$\langle \nabla f(x^{(k)}), z \rangle \neq 0 \quad \text{for some } z \in N_g, \tag{17}$$

then we can drive $\langle \nabla f(x^{(k)}), z \rangle \to -\infty$ along a sequence with $g(z) = 0$, and so the stagewise update step would be clearly undefined. Fortunately, a simple modification of the general stagewise algorithm can account for this problem. Since we are assuming that g is a seminorm, the set N_g is a linear subspace. To initialize the general stagewise algorithm at say $t_0 = 0$, therefore, we solve the linearly constrained optimization problem

$$x^{(0)} \in \operatorname*{argmin}_{x \in N_g} f(x).$$

In subsequent stagewise steps, we then restrict the updates to lie in the subspace orthogonal to N_g. That is, to be explicit, we replace (5) (6) in Algorithm 2 with

$$x^{(k)} = x^{(k-1)} + \Delta, \tag{18}$$

$$\text{where } \Delta \in \operatorname*{argmin}_{z \in N_g^\perp} \langle \nabla f(x^{(k-1)}), z \rangle \ \text{ subject to } \ g(z) \leq \epsilon, \tag{19}$$

where $N_g^\perp$ denotes the orthocomplement of N_g. We will see this modification, e.g., put to use for the quadratic regularizer $g(\beta) = \beta^T Q \beta$, where Q is positive semidefinite and singular.

Some readers may wonder why we are working with the constrained problem (4), and not

$$\hat{x}(\lambda) \in \operatorname*{argmin}_{x \in \mathbb{R}^n} f(x) + \lambda g(x), \tag{20}$$

where $\lambda \geq 0$ is now the regularization parameter, and is called the Lagrange multiplier associated with g. It is probably more common in the current statistics and machine learning literature for optimization problems to be expressed in the Lagrange form (20), rather than the constrained form (4). The solution paths of (4) and (20) (given by varying t and λ in their respective problems) are not necessarily equal for general convex functions f and g; however, they are equal under very mild assumptions[5], which hold for all of the examples visited in this paper. Therefore, there is not an important difference in terms of studying (4) versus (20). We choose to focus on (4) as we feel that the intuition for stagewise algorithms is easier to see with this formulation.

2.4 Related Work

There is a lot of work related to the proposal of this paper. Readers familiar with optimization will likely identify the general stagewise procedure, in Algorithm 2, as a particular

5. For example, it is enough to assume that $g \geq 0$, and that for all parameters $t, \lambda \geq 0$, the solution sets of (4), (20) are nonempty.

type of (normalized) *steepest descent*. Steepest descent is an iterative algorithm for minimizing a smooth convex function f, in which we update the current iterate in a direction that minimizes the inner product with the gradient of f (evaluated at the current iterate), among all vectors constrained to have norm $\|\cdot\|$ bounded by 1 (e.g., see Boyd and Vandenberghe, 2004); the step size for the update can be chosen in any one of the usual ways for descent methods. Note that gradient descent is simply a special case of steepest descent with $\|\cdot\| = \|\cdot\|_2$ (modulo normalizing factors). Meanwhile, the general stagewise algorithm is just steepest descent with $\|\cdot\| = g(\cdot)$, and a constant step size ϵ. It is important to point out that our interest in the general stagewise procedure is different from typical interest in steepest descent. In the classic usage of steepest descent, we seek to minimize a differentiable convex function f; our choice of norm $\|\cdot\|$ affects the speed with which we can find such a minimizer, but under weak conditions, any choice of norm will eventually bring us to a minimizer nonetheless. In the general stagewise algorithm, we are not really interested in the final minimizer itself, but rather, the path traversed in order to get to this minimizer. The stagewise path is composed of iterates that have interesting statistical properties, given by gradually balancing f and g; choosing different functions g will lead to generically different paths. Focusing on the path, instead of its endpoint, may seem strange to a researcher in optimization, but it is quite natural for researchers in statistics and machine learning.

Another method related to our general stagewise proposal is the *Frank-Wolfe algorithm* (Frank and Wolfe, 1956), used to minimize a differentiable convex function f over a convex set C. Similar to (projected) gradient descent, which iteratively minimizes local quadratic approximations of f over C, the Frank-Wolfe algorithm iteratively minimizes local linear approximations of f over C. In a recent paper, Jaggi (2013) shed light on Frank-Wolfe as an efficient, scalable algorithm for modern machine learning problems. For a single value of the regularization parameter t, the Frank-Wolfe algorithm can be used to solve problem (4), taking as the constraint set $C = \{x : g(x) \leq t\}$; the Frank-Wolfe steps here look very similar to the general stagewise steps (5), (6), but an important distinction is that the iterates from Frank-Wolfe result in a single estimate, rather than each iterate constituting its own estimate along the regularization path, as in the general stagewise procedure. This connection deserves more discussion, see Online Appendix A.1. Other well-known methods based on local linearization are *cutting-plane methods* (Kelley, 1960) and *bundle methods* (Hiriart-Urruty and Lemarechal, 1993). Teo et al. (2007) present a general bundle method for regularized risk minimization that is particularly relevant to our proposal (see also Teo et al., 2010); this is similar to the Frank-Wolfe approach in that it solves the problem (4) at a fixed value of the parameter t (one difference is that its local linearization steps are based on the entire history of previous iterates, instead of just the single last iterate). For brevity, we do not conduct a detailed comparison between their bundle method and our general stagewise procedure, though we believe it would be interesting to do so.

Yet another class of methods that are highly relevant to our proposal are *boosting* procedures. Boosting algorithms are iterative in form, and we typically think of them as tracing out a sequence of estimates, just like our general stagewise algorithm (and unlike the iterative algorithms described above, e.g., steepest descent and Frank-Wolfe, which we tend to think of as culminating in a single estimate). The literature on boosting is vast; see, e.g., Hastie et al. (2009) or Buhlmann and Yu (2010) for a nice review. Among boosting

methods, *gradient boosting* (Friedman, 2001) most closely parallels forward stagewise fitting. Consider a setup in which our weak learners are the individual predictor variables $X_1, \ldots X_p$, and the loss function is $L(X\beta) = f(\beta)$. The gradient boosting updates, using a shrinkage factor ϵ, are most commonly expressed in terms of the fitted values, as in

$$X\beta^{(k)} = X\beta^{(k-1)} + \epsilon \cdot \alpha_i X_i, \tag{21}$$

$$\text{where } \alpha_i \in \operatorname*{argmin}_{\alpha \in \mathbb{R}} L(X\beta^{(k-1)} + \alpha X_i), \tag{22}$$

$$\text{and } i \in \operatorname*{argmin}_{j=1,\ldots p} \left(\min_{\alpha \in \mathbb{R}} \| - \nabla L(X\beta^{(k-1)}) - \alpha X_j \|_2^2 \right). \tag{23}$$

The step (23) selects the weak learner X_i that best matches the negative gradient, $-\nabla L(X\beta^{(k-1)})$, in a least squares sense; the step (22) chooses the coefficient α_i of X_i via line search. If we assume that the variables have been scaled to have unit norm, $\|X_j\|_2 = 1$ for $j = 1, \ldots p$, then it is easy to see that (23) is equivalent to

$$i \in \operatorname*{argmax}_{j=1,\ldots p} |X_j^T \nabla L(X\beta^{(k-1)})| = \operatorname*{argmax}_{j=1,\ldots p} |\nabla_j f(\beta^{(k-1)})|,$$

which is exactly the same selection criterion used by forward stagewise under the loss function f, as expressed in (8). Therefore, at a given iteration, gradient boosting and forward stagewise choose the next variable i in the same manner, and only differ in their choice of the coefficient of X_i in the constructed additive model. The gradient boosting update in (21) adds $\epsilon \cdot \alpha_i X_i$ to the current model, where α_i is chosen by line search in (22); meanwhile, the forward stagewise update in (7) can be expressed as

$$X\beta^{(k)} = X\beta^{(k-1)} + \epsilon \cdot s_i X_i, \tag{24}$$

where $s_i = -\text{sign}(\nabla_i f(\beta^{(k-1)})$, a simple choice of coefficient compared to α_i. Because α_i is chosen by minimizing the loss function along the direction defined by X_i (anchored at $X\beta^{(k-1)}$), gradient boosting is even more greedy than forward stagewise, but practically there is not a big difference between the two, especially when ϵ is small. In fact, the distinction between (21) and (24) is slight enough that several authors refer to forward stagewise as a boosting procedure, e.g., Rosset et al. (2004), Zhao and Yu (2007), and Buhlmann and Yu (2010) refer to forward stagewise as *ϵ-boosting*.

The tie between boosting and forward stagewise suggests that we might be able to look at our general stagewise proposal through the lens of boosting, as well. Above we compared boosting and forward stagewise for the problem of sparse estimation; in this problem, deciding on the universe of weak learners for gradient boosting is more or less straightforward, as we can use the variables $X_1, \ldots X_p$ themselves (or, e.g., smooth marginal transformations of these variables for sparse nonparametric estimation). This works because each iteration of gradient boosting adds a single weak learner to the fitted model, so the model is sparse in the early stages of the algorithm, and becomes increasingly dense as the algorithm proceeds. However, for more complex problems (beyond sparse estimation), specifying a universe of weak learners is not as straightforward. Consider, e.g., matrix completion or image denoising—what kind of weak learners would be appropriate here? At a broad level, our general stagewise procedure offers a prescription for a class of weak learners

based on the regularizer g, through the definition of Δ in (6). Such weak learners seem intuitively reasonable in various problem settings: they end up being groups of variables for group-structured estimation problems (see Section 3.1), rank 1 matrices for matrix completion (Section 3.3), and pixel contrasts for image denoising (Section 3.5). This may lead to an interesting perspective on gradient boosting with an arbitrary regularization scheme, though we do not explore it further.

Finally, the form of the update Δ in (6) sets our work apart from other general path tracing procedures. Zhao and Yu (2007) and Friedman (2008) propose approximate path following methods for optimization problems whose regularizers extend beyond the ℓ_1 norm, but their algorithms only update one component of the estimate at a time (which corresponds to using individual variables as weak learners, in the boosting perspective); on the other hand, our general stagewise procedure specifically adapts its updates to the regularizer of concern g. We note that, in certain special cases (i.e., for certain regularizers g), our proposed algorithm bears similarities to existing algorithms in the literature: for ridge regularization, our proposal is similar to gradient-directed path following, as studied in Friedman and Popescu (2004) and Ramsay (2005), and for ℓ_1/ℓ_2 multitask learning, our stagewise algorithm is similar to the block-wise path following method of Obozinski et al. (2010).

3. Applications of the General Stagewise Framework

In each subsection below, we walk through the application of the stagewise framework to a particular type of regularizer.

3.1 Group-structured Regularization

We begin by considering the group-structured regularization problem

$$\hat{\beta}(t) \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} f(\beta) \;\text{ subject to }\; \sum_{j=1}^{G} w_j \|\beta_{\mathcal{I}_j}\|_2 \leq t, \tag{25}$$

where the index set $\{1, \ldots p\}$ has been partitioned into G groups $\mathcal{I}_1, \ldots \mathcal{I}_G$, $\beta_{\mathcal{I}_j} \in \mathbb{R}^{p_j}$ denotes the components of $\beta \in \mathbb{R}^p$ for the jth group, and $w_1, \ldots w_G \geq 0$ are fixed weights. The loss f is kept as a generic differentiable convex function—this is because, as explained in Section 2.3, the stagewise updates are invariant around ∇f, in terms of their computational form.

Note that the *group lasso* problem (Bakin, 1999; Yuan and Lin, 2006) is a special case of (25). In the typical group lasso regression setup, we observe an outcome $y \in \mathbb{R}^n$ and predictors $X \in \mathbb{R}^{n \times p}$, and the predictor variables admit some natural grouping $\mathcal{I}_1, \ldots \mathcal{I}_G$. To perform group-wise variable selection, one can use the group lasso estimator, defined as in (25) with

$$f(\beta) = \frac{1}{2}\Big\|y - \sum_{j=1}^{G} X_{\mathcal{I}_j}\beta_{\mathcal{I}_j}\Big\|_2^2 \quad \text{and} \quad w_j = \sqrt{p_j},\;\; j = 1, \ldots G,$$

where $X_{\mathcal{I}_j} \in \mathbb{R}^{n \times p_j}$ is the predictor matrix for group j, and $p_j = |\mathcal{I}_j|$ is the size of the group j. The same idea clearly applies outside of the linear regression setting (e.g., see Meier et al., 2008 for a study of the group lasso regularization in logistic regression).

A related yet distinct problem is that of *multitask learning*. In this setting we consider not one but multiple learning problems, or tasks, and we want to select a common set of variables that are important across all tasks. A popular estimator for this purpose is based on ℓ_1/ℓ_2 regularization (Argyriou et al., 2006; Obozinski et al., 2010), and also fits into the framework (25): the loss function f becomes the sum of the losses across the tasks, and the groups $\mathcal{I}_1, \ldots \mathcal{I}_G$ collect the coefficients corresponding to the same variables across tasks. For example, in multitask linear regression, we write $y^{(i)} \in \mathbb{R}^n$ for the outcome, $X^{(i)} \in \mathbb{R}^{n \times m}$ for the predictors, and $\beta^{(i)}$ the coefficients for the ith task, $i = 1, \ldots r$. We form a global coefficient vector $\beta = (\beta^{(1)}, \ldots \beta^{(m)}) \in \mathbb{R}^p$, where $p = m \cdot r$, and form groups $\mathcal{I}_1, \ldots \mathcal{I}_m$, where $\mathcal{I}_j$ collects the coefficients of predictor variable j across the tasks. The ℓ_1/ℓ_2 regularized multitask learning estimator is then defined as in (25) with

$$f(\beta) = \frac{1}{2} \sum_{i=1}^{r} \|y^{(i)} - X^{(i)} \beta^{(i)}\|_2^2 \quad \text{and} \quad w_j = 1, \;\; j = 1, \ldots m,$$

where the default is to set all of the weights to 1, in the lack of any prior information about variable importance (note that the groups $\mathcal{I}_1, \ldots \mathcal{I}_m$ are all the same size here).

The general stagewise algorithm, Algorithm 2, does not make any distinction between cases such as the group lasso and multitask learning problems; it only requires f to be a convex and smooth function. To initialize the algorithm for the group regularized problem (25), we can take $t_0 = 0$ and $\beta^{(0)} = 0$. The next lemma shows how to calculate the appropriate update direction Δ in (6).

Lemma 1 *For $g(\beta) = \sum_{j=1}^{G} w_j \|\beta_{\mathcal{I}_j}\|_2$, the general stagewise procedure in Algorithm 2 repeats the updates $\beta^{(k)} = \beta^{(k-1)} + \Delta$, where Δ can be computed as follows: first find i such that*

$$\frac{\|(\nabla f)_{\mathcal{I}_i}\|_2}{w_i} = \max_{j=1,\ldots G} \frac{\|(\nabla f)_{\mathcal{I}_j}\|_2}{w_j}, \tag{26}$$

where we abbreviate $\nabla f = \nabla f(\beta^{(k-1)})$, then let

$$\Delta_{\mathcal{I}_j} = 0 \quad \textit{for all } j \neq i, \tag{27}$$

$$\Delta_{\mathcal{I}_i} = \frac{-\epsilon \cdot (\nabla f)_{\mathcal{I}_i}}{w_i \|(\nabla f)_{\mathcal{I}_i}\|_2}. \tag{28}$$

We omit the proof; it follows straight from the KKT conditions for (6), with g as defined in the lemma. Computation of Δ in (26), (27), (28) is very cheap, and requires $O(p)$ operations. To rephrase: at the kth iteration, we simply find the group i such that the corresponding block of the gradient $\nabla f(\beta^{(k-1)})$ has the largest ℓ_2 norm (after scaling appropriately by the weights). We then move the coefficients for group i in a direction opposite to this gradient value; for all other groups, we leave their coefficients untouched (note that, if a group has not been visited by past update steps, then this means leaving its coefficients identically equal to zero). The outputs of the stagewise algorithm therefore

match our intuition about the role of the constraint in (25)—for some select groups, all coefficients are set to nonzero values, and for other groups, all coefficients are set to zero. That the actual solution in (25) satisfies this intuitive property can be verified by examining its own KKT conditions.

Looking back at Figure 2, the first row compares the exact solution and stagewise paths for a group lasso regression problem. The stagewise path was computed using 300 steps with $\epsilon = 0.01$, and shows strong similarities to the exact group lasso path. In other problem instances, say, when the predictors across different groups are highly correlated, the group lasso coefficient paths can behave wildly with t, and yet the stagewise paths can appear much less wild and more stable. Later, in Section 4, we consider larger examples and give more thorough empirical comparisons.

3.2 Group-structured Regularization with Arbitrary Norms

Several authors have considered group-based regularization using the ℓ_∞ norm in place of the usual ℓ_2 norm (e.g., see Turlach et al., 2005 for such an approach in multitask learning). To accommodate this and other general group-structured regularization approaches, we consider the problem

$$\hat{\beta}(t) \in \operatorname*{argmin}_{\beta \in \mathbb{R}^p} f(\beta) \;\text{ subject to }\; \sum_{j=1}^{G} w_j h_j(\beta_{\mathcal{I}_j}) \leq t, \tag{29}$$

where each h_j is an arbitrary norm. Let h_j^* denote the dual norm of h_j; e.g., if $h_j(x) = \|x\|_{q_j}$, then $h_j^*(x) = \|x\|_{r_j}$, where $1/q_j + 1/r_j = 1$. Similar to the result in Lemma 1, the stagewise updates for problem (29) take a simple group-based form.

Lemma 2 *For $g(\beta) = \sum_{j=1}^{G} w_j h_j(\beta_{\mathcal{I}_j})$, the general stagewise procedure in Algorithm 2 repeats the updates $\beta^{(k)} = \beta^{(k-1)} + \Delta$, where Δ can be computed as follows: first find i such that*

$$\frac{h_i^*\big((\nabla f)_{\mathcal{I}_i}\big)}{w_i} = \max_{j=1,\ldots G} \frac{h_j^*\big((\nabla f)_{\mathcal{I}_j}\big)}{w_j},$$

where we abbreviate $\nabla f = \nabla f(\beta^{(k-1)})$, then let

$$\begin{aligned} \Delta_{\mathcal{I}_j} &= 0 \quad \textit{for all } j \neq i, \\ \Delta_{\mathcal{I}_i} &\in -\frac{\epsilon}{w_i} \cdot \partial h_i^*\big((\nabla f)_{\mathcal{I}_i}\big). \end{aligned}$$

Again we omit the proof; it follows from the KKT conditions for (6). Indeed, Lemma 2 covers Lemma 1 as a special case, recalling that the ℓ_2 norm is self-dual. Also, recalling that the ℓ_∞ and ℓ_1 norms are dual, Lemma 2 says that the stagewise algorithm for $g(\beta) = \sum_{j=1}^{G} w_j \|\beta_{\mathcal{I}_j}\|_\infty$ first finds i such that

$$\frac{\|(\nabla f)_{\mathcal{I}_i}\|_1}{w_i} = \max_{j=1,\ldots G} \frac{\|(\nabla f)_{\mathcal{I}_j}\|_1}{w_j},$$

and then defines the update direction Δ by

$$\Delta_{\mathcal{I}_j} = 0 \quad \text{for all } j \neq i,$$
$$\Delta_\ell = -\frac{\epsilon}{w_i} \cdot \begin{cases} 0 & \text{for } \ell \in \mathcal{I}_i,\ (\nabla f)_\ell = 0 \\ \text{sign}\big((\nabla f)_\ell\big) & \text{for } \ell \in \mathcal{I}_i,\ (\nabla f)_\ell \neq 0. \end{cases}$$

More broadly, Lemma 2 provides a general prescription for deriving the stagewise updates for regularizers that are block-wise sums of norms, as long as we can compute subgradients of the dual norms. For example, the norms in consideration could be a mix of ℓ_p norms, matrix norms, etc.

3.3 Trace Norm Regularization

Consider a class of optimization problems over matrices,

$$\hat{B}(t) \in \underset{B \in \mathbb{R}^{m \times n}}{\text{argmin}}\ f(B) \ \text{ subject to } \ \|B\|_* \leq t, \tag{30}$$

where $\|B\|_*$ denotes the trace norm (also called the nuclear norm) of a matrix B, i.e., the sum of its singular values. Perhaps the most well-known example of trace norm regularization comes from the problem of *matrix completion* (e.g., see Candes and Recht, 2009; Candes and Tao, 2010; Mazumder et al., 2010). Here the setup is that we only partially observe entries of a matrix $Y \in \mathbb{R}^{m \times n}$—say, we observe all entries $(i, j) \in \Omega$—and we seek to estimate the missing entries. A natural estimator for this purpose (studied by, e.g., Mazumder et al., 2010) is defined as in (30) with

$$f(B) = \frac{1}{2} \sum_{(i,j) \in \Omega} (Y_{ij} - B_{ij})^2.$$

The trace norm also appears in interesting examples beyond matrix completion. For example, Chen and Ye (2014) consider regularization with the trace norm in multiple nonparametric regression, and Harchaoui et al. (2012) consider it in large-scale image classification.

The general stagewise algorithm applied to the trace norm regularization problem (30) can be initialized with $t_0 = 0$ and $B^{(0)} = 0$, and the update direction in (6) is now simple and efficient.

Lemma 3 *For $g(B) = \|B\|_*$, the general stagewise procedure in Algorithm 2 repeats the updates $\beta^{(k)} = \beta^{(k-1)} + \Delta$, where*

$$\Delta = -\epsilon \cdot uv^T, \tag{31}$$

with u, v being leading left and right singular vectors, respectively, of $\nabla f(B^{(k-1)})$.

The proof relies on the fact that the dual of the trace norm $g(B) = \|B\|_*$ is the spectral norm $g^*(B) = \|B\|_2$, and then invokes the representation (15) for stagewise estimates. For the stagewise update direction (31), we need to compute the leading left and right singular vectors u, v of the $m \times n$ matrix $\nabla f(B^{(k-1)})$—these are the left and right singular vectors corresponding to the top singular value of $\nabla f(B^{(k-1)})$. Assuming that $\nabla f(B^{(k-1)})$ has a distinct largest singular value, this can be done, e.g., using the power method: letting

$A = \nabla f(B^{(k-1)})$, we first run the power method on the $m \times m$ matrix AA^T, or the $n \times n$ matrix A^TA, depending on whichever is smaller. This gives us either u or v; to recover the other, we then simply use matrix multiplication: $v = A^Tu/\|A^Tu\|_2$ or $u = Av/\|Av\|_2$. The power method is especially efficient if $A = \nabla f(B^{(k-1)})$ is sparse (each iteration being faster), or has a large spectral gap (fewer iterations required until convergence). Of course, alternatives to the power method can be used for computing the leading singular vectors of $\nabla f(B^{(k-1)})$, such as methods based on inverse iterations, Rayleigh quotients, or QR iterations; see, e.g., Golub and Van Loan (1996).

In the second row of Figure 2, the exact and stagewise paths for are shown matrix completion problem, where the stagewise paths were computed using 500 steps with $\epsilon = 0.05$. While the two sets of paths appear fairly similar, we note that it is harder to judge the degree of similarity between the two in the matrix completion context. Here, the coordinate paths correspond to entries in the estimated matrix $\hat{B}$, and their roles are not as clear as they are in, say, in a regression setting, where the coordinate paths correspond to the coefficients of individual variables. In other words, it is difficult to interpret the slight differences between the exact and stagewise paths in the second row of Figure 2, which present themselves as the trace norm grows large. Therefore, to get a sense for the effect of these differences, we might compare the mean squared error curves generated by the exact and stagewise estimates. This is done in depth in Section 4.

3.4 Quadratic Regularization

Consider problems of the form

$$\hat{\beta}(t) \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; f(\beta) \;\; \text{subject to} \;\; \beta^T Q \beta \leq t, \tag{32}$$

where $Q \succeq 0$, a positive semidefinite matrix. The quadratic regularizer in (32) encompasses several common statistical tasks. When $Q = I$, the regularization term $\beta^T\beta = \|\beta\|_2^2$ is well-known as *ridge* (Hoerl and Kennard, 1970), or *Tikhonov* regularization (Tikhonov, 1943). This regularizer shrinks the components of the solution $\hat{\beta}$ towards zero. In a (generalized) linear model setting with many predictor variables, such shrinkage helps control the variance of the estimated coefficients. Beyond this simple ridge case, roughness regularization in nonparametric regression often fits into the form (32), with Q not just the identity. For example, *smoothing splines* (Wahba, 1990; Green and Silverman, 1994) and *P-splines* (Eilers and Marx, 1996) can both be expressed as in (32). To see this, suppose that $y_1, \ldots y_n \in \mathbb{R}$ are observed across input points $x_1, \ldots x_n \in \mathbb{R}$, and let $b_1, \ldots b_p$ denote the B-spline basis (of, say, cubic order) with knots at locations $z_1, \ldots z_p \in \mathbb{R}$. Smoothing splines use the inputs as knots, $z_1 = x_1, \ldots z_p = x_n$ (so that $p = n$); P-splines typically employ a (much) smaller number of knots across the range of $x_1, \ldots x_n \in \mathbb{R}$. Both estimators solve problem (32), with a loss function $f(\beta) = \frac{1}{2}\|y - B\beta\|_2^2$, and $B \in \mathbb{R}^{n \times p}$ having entries $B_{ij} = b_j(x_i)$, but the two use a different definition for Q: its entries are given by $Q_{ij} = \int b_i''(x) b_j''(x)\, dx$ in the case of smoothing splines, while $Q = D^TD$ in the case of P-splines, where D is the discrete difference operator of a given (fixed) integral order. Both estimators can be extended to the logistic or Poisson regression settings, just by setting f to be the logistic or Poisson loss, with natural parameter $\eta = B\beta$ (Green and Silverman, 1994; Eilers and Marx, 1996).

When Q is positive definite, the general stagewise algorithm, applied to (32), can be initialized with $t_0 = 0$ and $\beta^{(0)} = 0$. The update direction Δ in (6) is described by the following lemma.

Lemma 4 *For $g(\beta) = \beta^T Q \beta$, with Q a positive definite matrix, the general stagewise procedure in Algorithm 2 repeats the updates $\beta^{(k)} = \beta^{(k-1)} + \Delta$, where*

$$\Delta = -\sqrt{\epsilon} \cdot \frac{Q^{-1} \nabla f}{\sqrt{(\nabla f)^T Q^{-1} \nabla f}}, \tag{33}$$

and ∇f is an abbreviation for $\nabla f(\beta^{(k-1)})$.

The proof follows by checking the KKT conditions for (6). When $Q = I$, the update step (33) of the general stagewise procedure for quadratic regularization is computationally trivial, reducing to

$$\Delta = -\sqrt{\epsilon} \cdot \frac{\nabla f}{\|\nabla f\|_2}.$$

This yields fast, simple updates for ridge regularized estimators. For a general matrix Q, computing the update direction in (33) boils down to solving the linear equation

$$Qv = \nabla f(\beta^{(k-1)}) \tag{34}$$

in v. This is expensive for an arbitrary, dense Q; a single solve of the linear system (34) generally requires $O(p^3)$ operations. Of course, since the systems across all iterations involve the same linear operator Q, we could initially compute a Cholesky decomposition of Q (or a related factorization), requiring $O(p^3)$ operations, and then use this factorization to solve (34) at each iteration, requiring only $O(p^2)$ operations. While certainly more efficient than the naive strategy of separately solving each instance of (34), this is still not entirely desirable for large problems.

On the other hand, for several cases in which Q is structured or sparse, the linear system (34) can be solved efficiently. For example, if Q is banded with bandwidth d, then we can solve (34) in $O(pd^2)$ operations (actually, an initial Cholesky decomposition takes $O(pd^2)$ operations, and each successive solve with this decomposition then takes $O(pd)$ operations).

Importantly, the matrix Q is banded in both the smoothing spline and P-spline regularization cases: for smoothing splines, Q is banded because the B-spline basis functions have local support; for P-splines, Q is banded because the discrete difference operator is. However, some care must be taken in applying the stagewise updates in these cases, as Q is singular, i.e., positive semidefinite but not strictly positive definite. The stagewise algorithm needs to be modified, albeit only slightly, to deal with this issue—this modification was discussed in (18), (19) in Section 2.3, and here we summarize the implications for problem (32). First we compute the initial iterate to lie in null(Q), the null space of Q,

$$\beta^{(0)} \in \operatorname*{argmin}_{\beta \in \text{null}(Q)} f(\beta). \tag{35}$$

For, e.g., P-splines with $Q = D^T D$, and D the discrete difference operator of order k, the space null(Q) is k-dimensional and contains (the evaluations of) all polynomial functions

of order $k-1$. The stagewise algorithm is then initialized at such a point $\beta^{(0)}$ in (35), and $t_0 = 0$. For future iterations, note that when $\nabla f(\beta^{(k)})$ has a nontrivial projection onto $\text{null}(Q)$, the stagewise update in (6) is undefined, since $\langle \nabla f(\beta^{(k)}), z \rangle$ can be made arbitrarily small along a direction z such that $z^T Q z = 0$. Therefore, we must further constrain the stagewise update to lie in the orthocomplement $\text{null}(Q)^\perp = \text{row}(Q)$, the row space of Q, as in

$$\Delta \in \operatorname*{argmin}_{z \in \text{row}(Q)} \langle \nabla f(\beta^{(k-1)}), z \rangle \;\; \text{subject to} \;\; z^T Q z \leq \epsilon.$$

It is not hard to check that, instead of (33), the update now becomes

$$\Delta = -\sqrt{\epsilon} \cdot \frac{Q^+ \nabla f}{\sqrt{(\nabla f)^T Q^+ \nabla f}}, \tag{36}$$

with Q^+ denoting the (Moore-Penrose) generalized inverse of Q.

From a computational perspective, the stagewise update in (36) for the rank deficient case does not represent more much work than that in (33) for the full rank case. With P-splines, e.g., we have $Q = D^T D$ where $D \in \mathbb{R}^{(n-k) \times n}$ is a banded matrix of full row rank. A short calculation shows that in this case

$$(D^T D)^+ = D^T (D D^T)^{-2} D,$$

i.e., applying Q^+ is computationally equivalent to two banded linear system solves and two banded matrix multiplications. Hence one stagewise update for P-spline regularization problems takes $O(p)$ operations (the bandwidth of D is a constant, $d = k+1$), excluding computation of the gradient.

The third row of Figure 2 shows an example of logistic regression with ridge regularization, and displays the grossly similar exact solution and stagewise paths. Notably, the stagewise path here was constructed using only *15 steps*, with an effective step size $\sqrt{\epsilon} = 0.1$. This is a surprisingly small number of steps, especially compared to the numbers needed by stagewise in the examples (both small and large) from other regularization settings covered in this paper. As far as we can tell, this rough scaling appears to hold for ridge regularization problems in general—for such problems, the stagewise algorithm can be run with relatively large step sizes for small numbers of steps, and it will still produce statistically appealing paths. Unfortunately, this trend does not persist uniformly across all quadratic regularization problems; it seems that the ridge case ($Q = I$) is really a special one.

For a second example, we consider P-spline regularization, using both continuous and binomial outcomes. The left panel of Figure 3 displays an array of stagewise estimates, computed under P-spline regularization and a Gaussian regression loss. We generated $n = 100$ noisy observations $y_1, \ldots y_{100}$ from an underlying sinusoidal curve, sampled at input locations $x_1, \ldots x_{100}$ drawn uniformly over $[0, 1]$. The P-splines were defined using 30 equally spaced knots across $[0, 1]$, and the stagewise algorithm was run for 300 steps with $\sqrt{\epsilon} = 0.005$. The figure shows the spline approximations delivered by the stagewise estimates (from every 15th step along the path, for visibility) and the true sinusoidal curve overlayed as a thick dotted black line. We note that in this particular setting, the stagewise algorithm is not so interesting computationally, because each update step solves a banded linear system, and yet the exact solution can itself be computed at the same cost, at any regularization

parameter value. The example is instead meant to portray that the stagewise algorithm can produce smooth and visually reasonable estimates of the underlying curve.

The right panel of Figure 3 displays an analogous example using $n = 100$ binary observations, $y_1, \ldots y_{100}$, generated according to the probabilities $p_i^* = 1/(1 + e^{-\mu(x_i)})$, $i = 1, \ldots 100$, where the inputs $x_1, \ldots x_{100}$ were sampled uniformly from $[0, 1]$, and μ is a smooth function. The probability curve $p^*(x) = 1/(1 + e^{-\mu(x)})$ is drawn as a thick dotted black line. We ran the stagewise algorithm under a logistic loss, with $\sqrt{\epsilon} = 0.005$, and for 300 steps; the figure plots the probability curves associated with the stagewise estimates (from every 15th step along the path, for visibility). Again, we can see that the fitted curves are smooth and visually reasonable. Computationally, the difficulty of the stagewise algorithm in this logistic setting is essentially the same as that in the previous Gaussian setting; all that changes is the computation of the gradient, which is an easy task. The exact solution, however, is more difficult to compute in this setting than the previous, and requires the use of iterative algorithm like Newton's method. This kind of computational invariance around the loss function, recall, is an advantage of the stagewise framework.

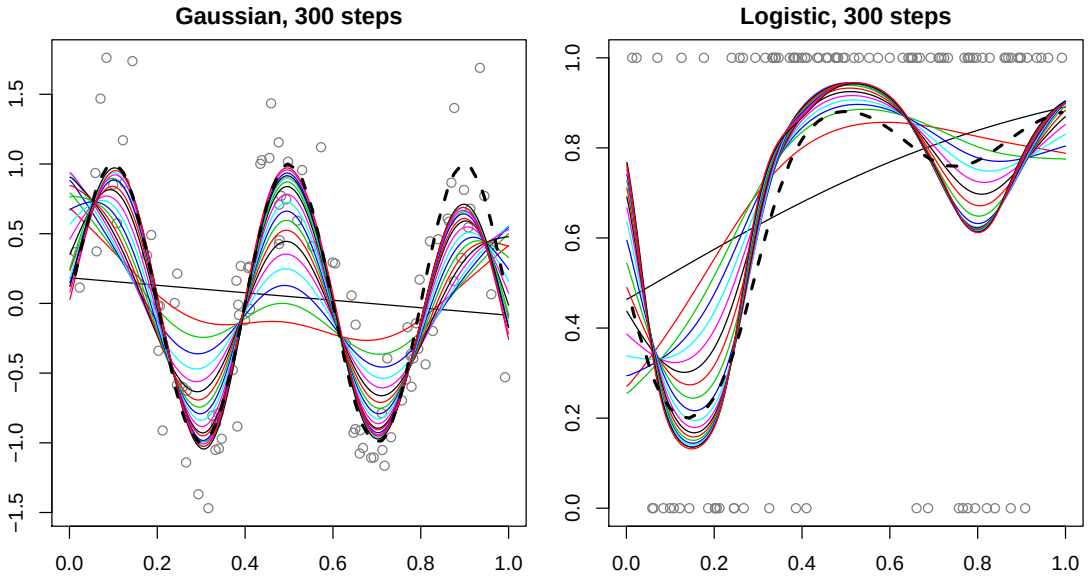


Figure 3: Snapshots of the stagewise path for P-spline regularization problems, with continuous data in the left panel, and binary data in the right panel. In both examples, we use $n = 100$ points, and the true data generating curve is displayed as a thick dotted black line. The colored curves show the stagewise estimates over the first 300 path steps (plotted are every 15th estimate, for visibility).

3.5 Generalized Lasso Regularization

In this last application, we study generalized ℓ_1 regularization problems,

$$\hat{\beta}(t) \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; f(\beta) \;\; \text{subject to} \;\; \|D\beta\|_1 \leq t, \tag{37}$$

where D is a given matrix (it need not be square). The regularization term above is also called *generalized lasso* regularization, since it includes lasso regularization as a special case, with $D = I$, but also covers a number of other regularization forms (Tibshirani and Taylor, 2011). For example, *fused lasso* regularization is encompassed by (37), with D chosen to be the edge incidence matrix of some graph G, having nodes $V = \{1, \dots p\}$ and edges $E = \{e_1, \dots e_m\}$. In the special case of the chain graph, wherein $E = \{\{1,2\}, \{2,3\}, \dots \{p-1, p\}\}$, we have

$$D = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 \\ \vdots & & & & & \\ 0 & 0 & 0 & \dots & -1 & 1 \end{bmatrix},$$

so that $\|D\beta\|_1 = \sum_{j=1}^{p-1} |\beta_j - \beta_{j+1}|$. This regularization term encourages the ordered components of β to be piecewise constant, and problem (37) with this particular choice of D is usually called the *1-dimensional fused lasso* in the statistics literature (Tibshirani et al., 2005), or *1-dimensional total variation denoising* in signal processing (Rudin et al., 1992). In general, the edge incidence matrix $D \in \mathbb{R}^{m \times p}$ has rows corresponding to edges in E, and its ℓth row is

$$D_\ell = (0, \dots \underset{\uparrow}{-1}_{i}, \dots \underset{\uparrow}{1}_{j}, \dots 0) \in \mathbb{R}^p,$$

provided that the ℓth edge is $e_\ell = \{i, j\}$. Hence $\|D\beta\|_1 = \sum_{\{i,j\} \in E} |\beta_i - \beta_j|$, a regularization term that encourages the components of β to be piecewise constant with respect to the structure defined by the graph G. Higher degrees of smoothness can be regularized in this framework as well, using *trend filtering* methods; see Kim et al. (2009) or Tibshirani (2014) for the 1-dimensional case, and Wang et al. (2015) for the more general case over arbitrary graphs.

Unfortunately the stagewise update in (6), under the regularizer $g(\beta) = \|D\beta\|_1$, is not computationally tractable. Computing this update is the same as solving a linear program, absent of any special structure in the presence of a generic matrix D. But we can make progress by studying the generalized lasso from the perspective of convex duality. Our jumping point for the dual is actually the Lagrange form of problem (37), namely

$$\hat{\beta}(\lambda) \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \; f(\beta) + \lambda \|D\beta\|_1, \tag{38}$$

with $\lambda \geq 0$ now being the regularization parameter. The switch from (37) to (38) is justified because the two parametrizations admit identical solution paths. Following standard arguments in convex analysis, the dual problem of (38) can be written as

$$\hat{u}(\lambda) \in \underset{u \in \mathbb{R}^m}{\operatorname{argmin}} \; f^*(-D^T u) \;\; \text{subject to} \;\; \|u\|_\infty \leq \lambda, \tag{39}$$

with f^* denoting the convex conjugate of f. The primal and dual solutions satisfy the relationship

$$\nabla f(\hat{\beta}(\lambda)) + D^T \hat{u}(\lambda) = 0. \tag{40}$$

The general strategy is now to apply the stagewise algorithm to the dual problem (39) to produce an approximate dual solution path, and then convert this into an approximate

primal solution path via (40). The stagewise procedure for (39) can be initialized with $\lambda_0 = 0$ and $u^{(0)} = 0$, and the form of the updates is described next. We assume that the conjugate function f^* is differentiable, which holds if f is strictly convex.

Lemma 5 *Applied to the problem* (39)*, the general stagewise procedure in Algorithm 2 repeats the updates* $u^{(k)} = u^{(k-1)} + \Delta$*, where*

$$\Delta_i = -\epsilon \cdot \begin{cases} 1 & \big[D\nabla f^*(-D^T u^{(k-1)})\big]_i < 0 \\ -1 & \big[D\nabla f^*(-D^T u^{(k-1)})\big]_i > 0 \\ 0 & \big[D\nabla f^*(-D^T u^{(k-1)})\big]_i = 0 \end{cases} \quad \text{for } i = 1, \ldots m. \tag{41}$$

The proof follows from the duality of the ℓ_∞ and ℓ_1 norms, and the alternative representation in (15) for stagewise updates. Computation of Δ in (41), aside from evaluating the gradient ∇f^*, reduces to two matrix multiplications: one by D and one by D^T. In many cases (e.g., fused lasso and trend filtering problems), the matrix D is sparse, which makes this update step very cheap. To reiterate the dual strategy: we compute the dual estimates $u^{(k)}$, $k = 1, 2, 3, \ldots$ using the stagewise updates outlined above, and we compute primal estimates $\beta^{(k)}$, $k = 1, 2, 3, \ldots$ by solving for $\beta^{(k)}$ in the stationarity condition

$$\nabla f(\beta^{(k)}) + D^T u^{(k)} = 0, \tag{42}$$

for each k. The kth dual iterate $u^{(k)}$ is viewed as an approximate solution in (39) at $\lambda = \|u^{(k)}\|_\infty$, and the kth primal iterate $\beta^{(k)}$ an approximate solution in (37) at $t = \|D\beta^{(k)}\|_1$.

As pointed out by a referee of this paper, there is a key relationship between f and its conjugate f^* that simplifies the update direction in (41) considerably. At step k, observe that

$$\nabla f^*(-D^T u^{(k-1)}) = \nabla f^*(\nabla f(\beta^{(k-1)})) = \beta^{(k-1)}.$$

The first equality comes from the primal-dual relationship (40) at step $k-1$, and the second is due to the fact that $x = \nabla f^*(z) \iff z = \nabla f(x)$. As a result, the dual update $u^{(k)} = u^{(k-1)} + \Delta$ with Δ as in (41) can be written more succinctly as

$$u^{(k)} = u^{(k-1)} - \epsilon \cdot \mathrm{sign}(D\beta^{(k-1)}), \tag{43}$$

where $\mathrm{sign}(\cdot)$ is to be interpreted componentwise (with the convention $\mathrm{sign}(0) = 0$). Therefore, one can think of the dual stagewise strategy as alternating between computing a dual estimate $u^{(k)}$ as in (43), and computing a primal estimate $\beta^{(k)}$ by solving (42).

We note that, since the stagewise algorithm is being run through the dual, the estimates $\beta^{(k)}$, $k = 1, 2, 3, \ldots$ for generalized lasso problems differ from those in the other stagewise implementations encountered thus far, in that $\beta^{(k)}$, $k = 1, 2, 3, \ldots$ correspond to approximate solutions at *increasing* levels of regularization, as k increases. That is, the stagewise algorithm for problem (37) begins at the unregularized end of the path and iterates towards the fully regularized end, which is opposite to its usual direction.

A special case worth noting is that of Gaussian signal approximator problems, where the loss is $f(\beta) = \frac{1}{2}\|y - \beta\|_2^2$. For such problems, the primal-dual relationship in (42) reduces to

$$\beta^{(k)} = y - D^T u^{(k)},$$

for each k. This means that the initialization $u^{(0)} = 0$ and $\lambda_0 = 0$ in the dual is the same as $\beta^{(0)} = y$ and $t_0 = \|Dy\|_1$ in the primal. Furthermore, it means that the dual updates in (43) lead to primal updates that can be expressed directly as

$$\beta^{(k)} = \beta^{(k-1)} - \epsilon \cdot D^T \text{sign}(D\beta^{(k-1)}). \tag{44}$$

From the pure primal perspective, therefore, the stagewise algorithm begins with the trivial unregularized estimate $\beta^{(0)} = y$, and to fit subsequent estimates in (44), it iteratively shrinks along directions opposite to the active rows of D. That is, if $D_\ell \beta^{(k-1)} > 0$ (where D_ℓ is the ℓth row of D), then the algorithm adds D_ℓ^T to $\beta^{(k-1)}$ in forming $\beta^{(k)}$, which shrinks $D_\ell \beta^{(k)}$ towards zero, as $D_\ell D_\ell^T > 0$ (recall that D_ℓ is a row vector). The case $D_\ell \beta^{(k-1)} < 0$ is similar. If $D_\ell \beta^{(k-1)} = 0$, then no shrinkage is applied along D_ℓ.

This story can be made more concrete for fused lasso problems, where D is the edge incidence matrix of a graph: here the update in (44) evaluates the differences across neighboring components of $\beta^{(k-1)}$, and for any nonzero difference, it shrinks the associated components towards each other to build $\beta^{(k)}$. The level of shrinkage is uniform across all active differences, as any two neighboring components move a constant amount ϵ towards each other.[6] This is a simple and natural iterative procedure for fitting piecewise constant estimates over graphs. For small examples using 1d and 2d grid graphs, see Online Appendix A.3.

4. Large-scale Examples and Practical Considerations

We compare the proposed general stagewise procedure to various alternatives, with respect to both computational and statistical performance, across the three of the four major regularization settings seen so far. The fourth setting is moved to Online Appendix A.4 for reasons of space. The current section specifically investigates large examples, at least relative to the small examples presented in Sections 1–3. Of course, one can surely find room to criticize our comparisons, e.g., with respect to a different tuning of the algorithm that computes exact solutions, a coarser grid of regularization parameter values over which it computes solutions, a different choice of algorithm completely, etc. We have tried to conduct fair comparisons in each problem setting, but we recognize that perfectly fair and exhaustive comparisons are near impossible. The message that we hope to convey is not that the stagewise algorithm is computationally superior to other algorithms in the problems we consider, but rather, that the stagewise algorithm is computationally competitive with the others, yet it is very simple, and capable of producing estimates of high statistical quality.

4.1 Group Lasso Regression

Overview. We examine two simulated high-dimensional group lasso regression problems. To compute group lasso solution paths, we used the `SGL` R package, available on the CRAN repository. This package implements a block coordinate descent algorithm for solving the group lasso problem, where each block update itself applies accelerated proximal gradient

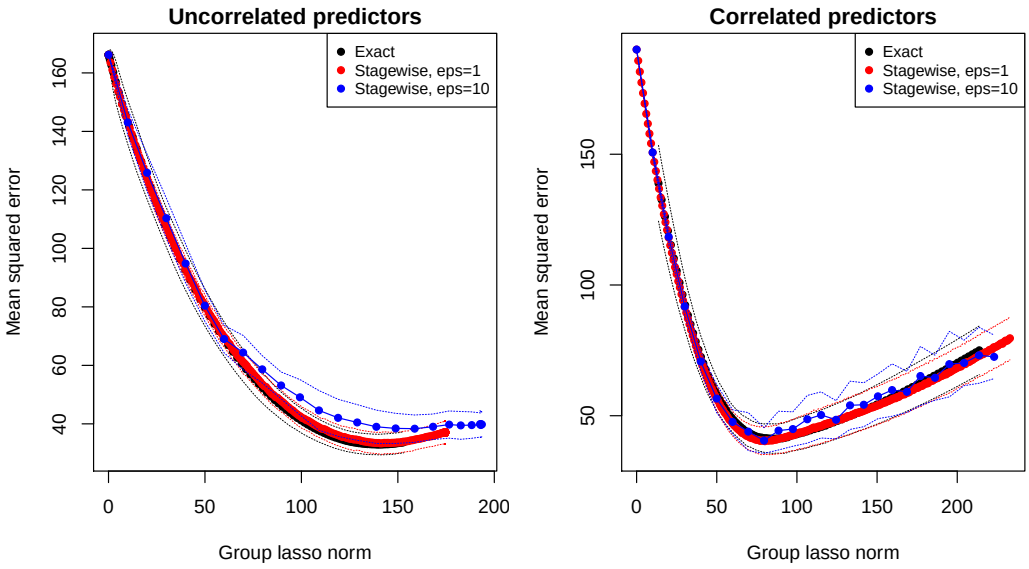
6. This is assuming that D is the edge incidence matrix of an unweighted graph; with edge weights, the rows of D scale accordingly, and so the effective amounts of shrinkage in the stagewise algorithm scale accordingly too.

descent (Simon et al., 2013). This idea is not complicated, but an efficient implementation of this algorithm requires care and attention to detail, such as backtracking line search for the proximal gradient step sizes. The stagewise algorithm, on the other hand, is very simple—in C++, the implementation is only about 50 lines of code. Refer to Section 3.1 for a description of the stagewise update steps. The algorithmics of the `SGL` package are also written in C++.

Examples and Comparisons. In both problem setups, we used $n = 200$ observations, $p = 4000$ predictors, and $G = 100$ equal-sized groups (of size 40). The true coefficient vector $\beta^* \in \mathbb{R}^{4000}$ was defined to be group sparse, supported on only 4 groups, and the nonzero components were drawn independently from $N(0, 1)$. We generated observations $y \in \mathbb{R}^{200}$ by adding independent $N(0, \tau^2)$ noise to $X\beta^*$, where the predictor matrix $X \in \mathbb{R}^{200\times 4000}$ and noise level τ were chosen under two different setups. In the first, the entries of X were drawn independently from $N(0, 1)$, so that the predictors were uncorrelated (in the population); we also let $\tau = 6$. In the second, each row of X was drawn independently from a $N(0, \Sigma)$ distribution, where Σ had a block correlation structure. The covariance matrix Σ was defined so that each predictor variable had unit (population) variance, but (population) correlation $\rho = 0.85$ with 99 other predictors, each from a different group. Further, in this second setup, we used an elevated noise level $\tau = 10$.

Figure 4 shows a comparison of the group lasso and stagewise paths, from both computational and statistical perspectives. We fit group lasso solutions over 100 regularization parameter values (the `SGL` package started at the regularized end, and used warm starts). We also ran the stagewise algorithm in two modes: for 250 steps with $\epsilon = 1$, and for 25 steps with $\epsilon = 10$. The top row of Figure 4 asserts that, in both the uncorrelated and correlated problem setups, the mean squared errors of the stagewise fits $X\beta^{(k)}$ to the underlying mean $X\beta^*$ are quite competitive with those of the exact fits $X\hat{\beta}(t)$. In both plots, the red and black error curves, corresponding to the stagewise fits with $\epsilon = 1$ and the exact fits, respectively, lie directly on top of each other. It took less than 1 second to compute these stagewise fits, in either problem setup; meanwhile, it took about 10 times this long to compute the group lasso fits in the uncorrelated setup, and 100 times this long in the correlated setup. The stagewise algorithm with $\epsilon = 10$ took less than 0.1 seconds to compute a total of 25 estimates, and offers a slightly degraded but still surprisingly competitive mean squared error curve, in both the correlated and uncorrelated problem setups. Exact timings can be found in the middle row of Figure 4. The error curves and timings were all averaged over 10 draws of observations y from the uncorrelated or correlated simulation models (for fixed X, β^*); the timings were made on a desktop personal computer.

Though the exact and stagewise component paths typically appear quite similar in the uncorrelated problem setup, the same is not true for the correlated setup. The bottom row of Figure 4 displays an example of the two sets of component paths for one simulated draw of observations, under the correlated predictor model. The component paths of the group lasso solution, on the left, vary wildly with the regularization parameter; the stagewise paths, on the right, are much more stable. It is interesting to see that such different estimates can yield similar mean squared errors (as, recall, shown in the top row of Figure 4) but this is the nature of using correlated predictors in a regression problem.



Algorithm timings		
Method	Uncorrelated case	Correlated case
Exact: coordinate descent, 100 solutions	9.08 (1.06)	78.64 (17.92)
Stagewise: $\epsilon = 1$, 250 estimates	0.93 (0.00)	0.94 (0.01)
Stagewise: $\epsilon = 10$, 25 estimates	0.09 (0.00)	0.10 (0.01)
Frank-Wolfe: within 1% of criterion value	67.73 (10.37)	92.91 (8.37)
Frank-Wolfe: within 1% of mean squared error	1.30 (0.56)	13.17 (26.26)

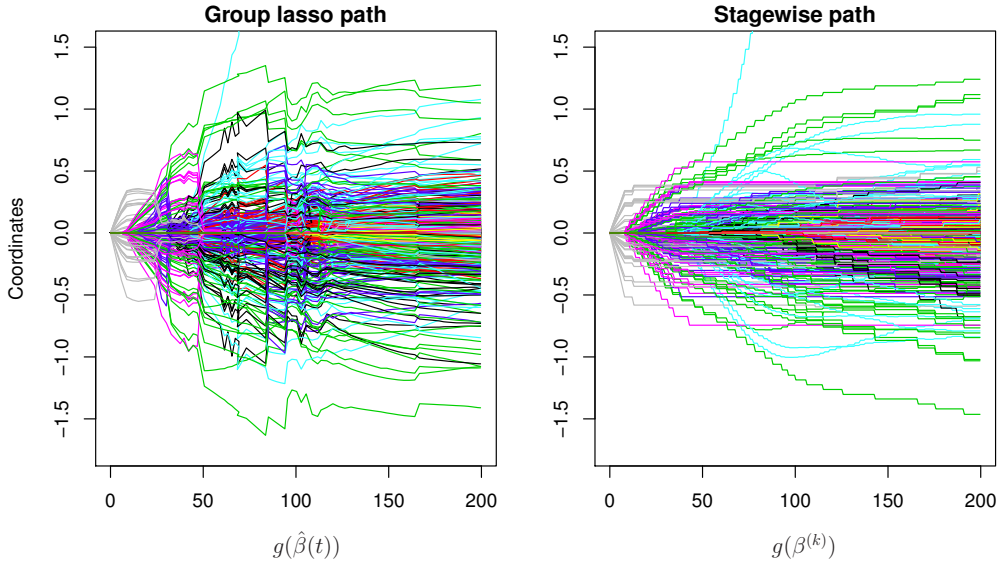


Figure 4: Statistical and computational comparisons between group lasso solutions and corresponding estimates produced by the stagewise approach, when $n = 200$, $p = 4000$. The top row shows that stagewise estimates can achieve competitive mean squared errors to that of group lasso solutions, as computed by coordinate descent, under two different setups for the predictors in group lasso regression: uncorrelated and block correlated. (The curves were averaged over 10 simulations, with standard deviations denoted by dotted lines.) The middle table reports runtimes in seconds (averaged over 10 simulations, with standard deviations in parentheses) for the various algorithms considered, and shows that the stagewise algorithm represents a computationally attractive alternative to the `SGL` coordinate descent approach and the Frank-Wolfe algorithm. Lastly, the bottom row contrasts the group lasso and stagewise component paths, for one draw from the correlated predictors setup.

Frank-Wolfe. We include a comparison to the Frank-Wolfe algorithm for computing group lasso solutions, across the same 100 regularization parameter values considered by the coordinate descent method. Recall that the updates from Frank-Wolfe share the same computational underpinnings as the stagewise ones, but are combined in a different manner; refer to Online Appendix A.1 for details. We implemented the Frank-Wolfe method for group lasso regression in C++, which starts at the largest regularization parameter value, and uses warm starts along the parameter sequence. The middle row of Figure 4 reports the Frank-Wolfe timings, averaged over 10 draws from the uncorrelated and correlated simulation models. We considered two schemes for termination of the algorithm, at each regularization parameter value t: the first terminates when

$$\|y - X\tilde{\beta}(t)\|_2^2 \leq 1.01 \cdot \|y - X\hat{\beta}(t)\|_2^2, \tag{45}$$

where $\tilde{\beta}(t)$ is the Frank-Wolfe iterate at t, and $\hat{\beta}(t)$ is the computed coordinate descent solution at t; the second terminates when

$$\|X\beta^* - X\tilde{\beta}(t)\|_2^2 \leq 1.01 \cdot \max\Big\{\|X\beta^* - X\hat{\beta}(t)\|_2^2, \|X\beta^* - X\beta^{(k_t)}\|_2^2\Big\}, \tag{46}$$

where $\beta^{(k_t)}$ is the imputed stagewise estimate at the parameter value t (computed by linear interpolation of the appropriate neighboring stagewise estimates). In other words, the first rule (45) stops when the Frank-Wolfe iterate is within 1% of the criterion value achieved by the coordinate descent solution, and the second rule (46) stops when the Frank-Wolfe iterate is within 1% of the mean squared error of either of the coordinate descent or stagewise fits. Using the first rule, the Frank-Wolfe algorithm took about 68 seconds to compute 100 solutions in the uncorrelated problem setup, and 93 seconds in the correlated problem setup. In terms of the total iteration count, this meant 18,627 Frank-Wolfe iterations in the uncorrelated case, and 25,579 in the correlated case; these numbers are meaningful, because, recall, one Frank-Wolfe iteration is (essentially) computationally equivalent to one stagewise iteration. We can see that Frank-Wolfe struggles here to compute solutions that match the accuracy of coordinate descent solutions, especially for large values of t—in fact, when we changed the factor of 1.01 to 1 in the stopping rule (45), the Frank-Wolfe algorithm converged far, far more slowly. (For this part, the coordinate descent solutions themselves were only computed to moderate accuracy; we used the default convergence threshold in the `SGL` package.) The results are more optimistic under the second stopping rule. Under this rule, the Frank-Wolfe algorithm ran in just over 1 second (274 iterations) in the uncorrelated setup, and about 13 seconds (3592 iterations) in the correlated setup. But this stopping rule represents an idealistic situation for Frank-Wolfe, and moreover, it cannot be realistically applied in practice, since it relies on the underlying mean $X\beta^*$.

4.2 Matrix Completion

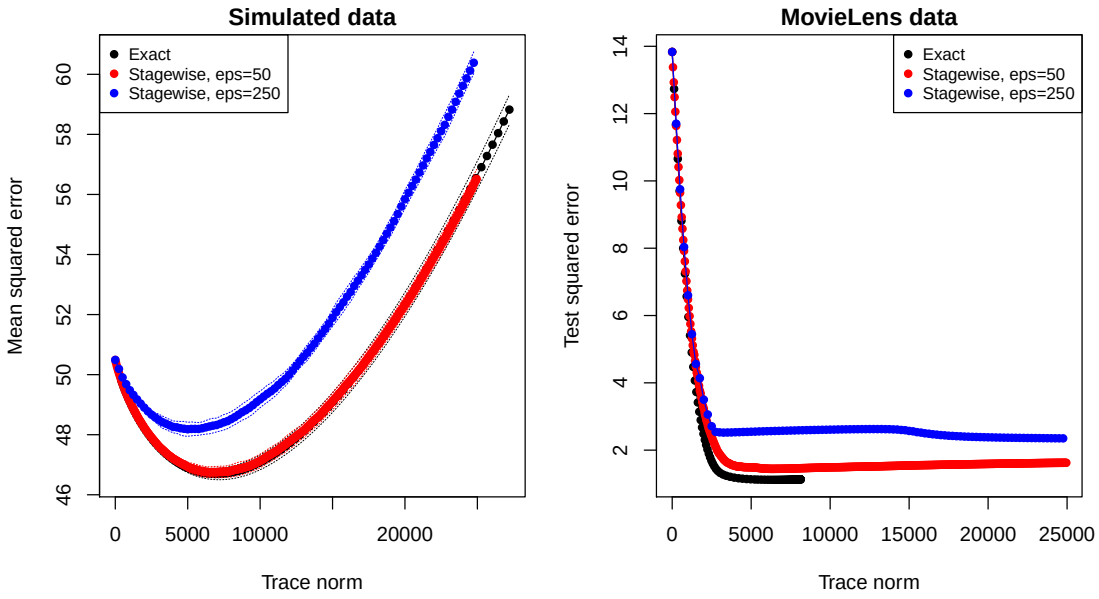
Overview. We consider two matrix completion examples, one simulated and one using real data. To compute solutions of the matrix completion problem, under trace norm regularization, we used the `softImpute` R package from CRAN, which implements proximal gradient descent (Mazumder et al., 2010). The proximal operator here requires a truncated singular value decomposition (SVD) of a matrix the same dimensions as the input (partially

observed) matrix Y. SVD calculations are generally very expensive, but for this problem a partial SVD can be efficiently computed with clever schemes based on bidiagonalization or alternating least squares. The `softImpute` package uses the latter scheme to compute a truncated SVD, and though this does provide a substantial improvement over the naive method of computing a full SVD, it is still far from cheap. The partial SVD computation via alternating least squares scales roughly quadratically with the rank of the sought solution, and this must be repeated for every iteration taken by the algorithm until convergence.

In comparison, the stagewise steps for the matrix completion problem require only the top left and right singular vectors of a matrix the same size as the input Y. Refer back to Section 3.3 for an explanation. To emphasize the differences between the two methods: the proximal gradient descent algorithm of `softImpute`, at each regularization parameter value t of interest, must iteratively compute a partial SVD until converging on the desired solution; the stagewise algorithm computes a single pair of left and right singular vectors, to form one estimate at one parameter value t, and then moves on to the next value of t. For the following examples, we used a simple R implementation of the stagewise algorithm; the computational core of the `softImpute` package is also written in R.

Examples and Comparisons. In the first example, we simulated an underlying low-rank matrix $B^* \in \mathbb{R}^{500\times 500}$, of rank 50, by letting $B^* = UU^T$, where $U \in \mathbb{R}^{500\times 50}$ had independent $N(0,1)$ entries. We then added $N(0,20)$ noise, and discarded 40% of the entries, to form the input matrix $Y \in \mathbb{R}^{500\times 500}$ (so that Y was 60% observed). We ran `softImpute` at 100 regularization parameter values (starting at the regularized end, and using warm starts), and we ran two different versions of the stagewise algorithm: one with $\epsilon = 50$, for 500 steps, and one with $\epsilon = 250$, for 100 steps. The left plot in Figure 5 shows the mean squared error curves of the resulting estimates, averaged over 10 draws of the input matrix Y from the above prescription (with B^* fixed). We can see that the stagewise estimates, with $\epsilon = 50$, trace out an essentially identical mean squared error curve to that from the exact solutions. We can also see that, curiously, the larger step size $\epsilon = 250$ leads to suboptimal performance in stagewise estimation, as measured by mean squared error. This is unlike the previous group lasso setting, in which a larger step size still yielded basically the same performance (albeit slightly noisier mean squared error curves).

The proximal gradient descent method implemented by `softImpute` in this simulated example took an average of 206 iterations to compute 100 solutions across 100 values of the regularization parameter (averaged over the 10 repetitions of the observation matrix Y). This means an average of just 2.06 iterations per solution—quite rapid convergence behavior for a first-order method like proximal gradient descent. (Note: we used the default convergence threshold for `softImpute`, which is only moderately small.) The stagewise algorithms, using step sizes $\epsilon = 50$ and $\epsilon = 250$, ran for 500 and 100 iterations, respectively. As explained, the two types of iterations here are different in nature. Each iteration of proximal gradient descent computes a truncated SVD, which is of roughly quadratic complexity in the rank of current solution, and therefore becomes more expensive as we progress down the regularization path; each stagewise iteration computes a single pair of left and right singular vectors, which has the same cost throughout the path, independent of the rank of the current estimate. The bottom row of Figure 5 is a table containing the running times of these two methods (averaged over 10 draws of Y, and recorded on a desktop computer). We



Algorithm timings		
Method	Simulated data	MovieLens data
Exact: proximal gradient, 100 solutions	60.20 (1.45)	334.67
Stagewise: $\epsilon = 50$, 500 estimates	92.92 (2.42)	107.66
Stagewise: $\epsilon = 250$, 100 estimates	18.26 (0.98)	21.22
Frank-Wolfe: within 1% of criterion value	989.77 (19.88)	-
Frank-Wolfe: within 1% of mean squared error	154.06 (10.76)	-

Figure 5: Comparisons between exact and stagewise estimates for matrix completion problems. The top left plot shows mean squared error curves for a simulated example of a 40% observed, 500×500 input matrix, and the right shows the same for the MovieLens data, where the input is 6% observed and 943×1682. (The error curves in the left plot were averaged over 10 repetitions, and standard deviations are drawn as dotted lines.) The stagewise estimates with $\epsilon = 50$ are competitive in both cases. The bottom table gives the runtimes of `softImpute` proximal gradient descent, stagewise, and the Frank-Wolfe algorithm. (Timings for the simulated case were averaged over 10 repetitions, with standard deviations in parentheses; Frank-Wolfe was not run on the MovieLens example.)

see that proximal gradient descent spent an average of about 60 seconds to compute 100 solutions, i.e., 0.6 seconds per solution. The stagewise algorithm with $\epsilon = 50$ took an average of about 93 seconds for 500 steps, and the algorithm with $\epsilon = 250$ an average of 18 seconds for 100 steps, with both translate into about 0.18 seconds per estimate. The speedy time of 0.6 seconds per estimate of `softImpute` is explained by two factors: fast iterations (using the impressive, custom alternating least squares routine developed by the package authors to compute partial SVDs), and few iterations needed per solution (recall, only an average of 2.06 per solution in this example). The 0.18 seconds per stagewise iteration reflects the runtime of computing leading left and right singular vectors with R's standard `svd` function, as our implementation somewhat naively does (it does not take advantage of sparsity in any

way). This naive stagewise implementation works just fine for moderate matrix sizes, as in the current example. But for larger matrix sizes (and higher levels of missingness), we see significant improvements when we use a more specialized routine for computing the top singular vectors. We also see a bigger separation in the costs per estimate with stagewise and proximal gradient descent. This is discussed next.

The second example is based on the MovieLens data set (collected by the GroupLens Research Project at the University of Minnesota, see `http://grouplens.org/datasets/movielens/`). We examined a subset of the full data set, with 100,000 ratings from $m = 943$ users on $n = 1682$ movies (hence the input matrix $Y \in \mathbb{R}^{943\times 1682}$ was approximately 6% observed). We used an 80%/20% split of these ratings for training and testing, respectively; i.e., we computed matrix completion estimates using the first 80% of the ratings, and evaluated test errors on the held out 20% of the ratings. For the estimates, we ran `softImpute` over 100 values of the regularization parameter (starting at the regularized end, using warm starts), and stagewise with $\epsilon = 50$ for 500 steps, as well as with $\epsilon = 250$ for 100 steps. The right plot of Figure 5 shows the test error curves from each of these methods. The stagewise estimates computed with $\epsilon = 50$ and the exact solutions perform quite similarly, with the exact solutions having a slight advantage as the trace norm exceeds about 2500. The stagewise error curve when $\epsilon = 250$ begins by dropping off strongly just like the other two curves, but then it flattens out too early, while the other two continue descending. (We note that, for step sizes larger than $\epsilon = 250$, the test error curve stops decreasing even earlier, and for step sizes smaller than $\epsilon = 50$, the error curve reaches a slightly lower minimum, in line with that of the exact solution. This type of behavior is reminiscent of boosting algorithms.)

In terms of computation, the proximal gradient descent algorithm used a total of 1220 iterations to compute 100 solutions in the MovieLens example, or an average of 122 iterations per solution. This is much more than the 2.06 seconds per iteration as in the previous simulated example, and it explains the longer total runtime of about 335 seconds, i.e., the longer total time of 33.5 seconds per solution. The stagewise algorithms ran, by construction, for 500 and 100 steps and took about 108 and 21 seconds, respectively, i.e., an average of 0.21 seconds per estimate. To compute the leading left and right singular vectors in each stagewise step here, we used the `rARPACK` R package from CRAN, which accommodates sparse matrices. This was highly beneficial because the gradient $\nabla f(B^{(k-1)})$ at each stagewise step was very sparse (about 6% entries of its were nonzero, since Y was about 6% observed).

Frank-Wolfe. We now compare the Frank-Wolfe algorithm for computing matrix completion solutions, over the same 100 regularization parameter values used by `softImpute`. Each Frank-Wolfe iteration computes a single pair of left and right top singular vectors, just like stagewise iterations; see Online Appendix A.1 for a general description of the Frank-Wolfe method (or Jaggi and Sulovsky, 2010 for a study of Frank-Wolfe for trace norm regularization problems in particular). We implemented the Frank-Wolfe algorithm for matrix completion in R, which starts at the regularized end of the path, and uses warm starts at each regularization parameter value. The timings for the Frank-Wolfe method, run on the simulated example, are given in the table in Figure 5 (we did not run it on the MovieLens example). As before, in the group lasso setting, we considered two different stopping rules

for Frank-Wolfe, applied at each regularization parameter value t: the first stops when the achieved criterion value is within 1% of that achieved by the proximal gradient descent approach in `softImpute`, and the second stops when the achieved mean squared error is within 1% of either that of `softImpute` or stagewise. In either case, we cap the maximum number of iterations at 100, at each parameter value t.

Under the first stopping rule, the Frank-Wolfe algorithm required an average of 5847 iterations to compute 100 solutions (averaged over 10 draws of the input matrix Y); furthermore, this total was calculated under the limit of 100 maximum iterations per solution, and the algorithm met this limit at each one of the largest 50 regularization parameter values t. Recall that each one of these Frank-Wolfe iterations is computationally equivalent to a stagewise iteration. Accordingly, 500 steps of the stagewise algorithm, with $\epsilon = 50$, ran in about an order of magnitude less time—93 seconds versus 990 seconds. The message is that the Frank-Wolfe algorithm experiences serious difficulty in producing solutions at a level of accuracy close to that of proximal gradient descent, especially for lower levels of regularization. Using the second stopping rule, Frank-Wolfe ran much faster, and computed 100 solutions in about 997 iterations, or 154 seconds. However, there are two important points to stress. First, this rule is not generally available in practice, as it depends on performance measured with respect to the true matrix B^*. Second, the termination behavior under this rule is actually somewhat misleading, because once the mean squared error curve begins to rise (in the left plot of Figure 5, after about $t = 7000$ in trace norm), the second rule will always cause Frank-Wolfe with warm starts to trivially terminate in 1 iteration. Indeed, in the simulated data example, the Frank-Wolfe algorithm using this rule took about 22 iterations per solution before $t = 7000$, and trivially 1 iteration per solution after this point.

4.3 Image Denoising

Overview. We study the image denoising problem, cast as a generalized lasso problem with Gaussian signal approximator loss, and 2d fused lasso or 2d total variation regularization (meaning that the underlying graph is a 2d grid). To compute exact solutions of this problem, we applied a direct (noniterative) algorithm of Chambolle and Darbon (2009), that reduces this problem to sequence of maximum flow problems. The "parametric" maximum flow approach taken by these authors is both very elegant and highly specialized. To the best of our knowledge, their algorithm is one of the fastest existing algorithms for 2d fused lasso problems (more generally, fused lasso problems over graphs). For the simulations in this section we relied on a fast C++ implementation provided by the authors (see `http://www.cmap.polytechnique.fr/~antonin/software/`), which totals close to 1000 lines of code. The stagewise algorithm is almost trivially simple in comparison, as our own C++ implementation requires only about 50 lines of code. For the 2d fused lasso regularizer, the stagewise update steps reduce to sparse matrix multiplications; refer to Section 3.5 for details.

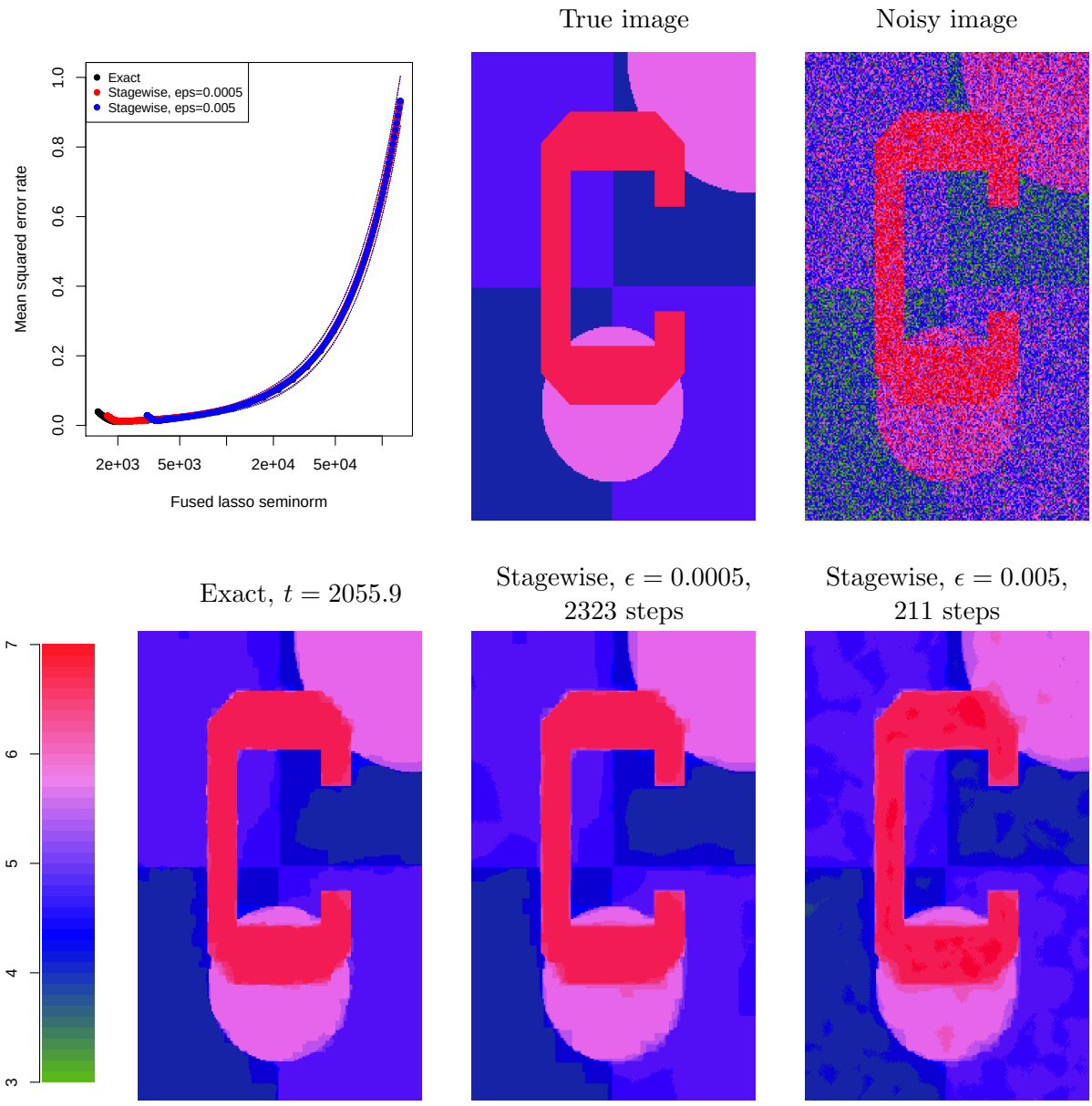
Examples and Comparisons. We inspect two image denoising examples. For the first, we constructed a 300×200 image to have piecewise constant levels, and added independent $N(0, 1)$ noise to the level of each pixel. Both this true underlying image and its noisy version are displayed in Figure 6. We then ran the parametric max flow approach of Chambolle and Darbon (2009), to compute exact 2d fused lasso solutions, at 100 values of the regularization

parameter. (This algorithm is direct and does not take warm starts, so each instance was solved separately.) We also ran the stagewise method in two modes: for 6000 steps with $\epsilon = 0.0005$, and for 500 steps with $\epsilon = 0.005$. The mean squared error curves for each method are shown in the top left corner of Figure 6, and timings are given in the bottom table. (All results here have been averaged over 10 draws of the noisy image, and the timings were recorded on a desktop computer.) We can see that the stagewise estimates, both with $\epsilon = 0.0005$ and $\epsilon = 0.005$, perform comparably to the exact solutions in terms of mean squared error, though the estimates under the smaller step size fare slightly better towards the more regularized end of the path. The 6000 stagewise estimates using $\epsilon = 0.0005$ took about 15 seconds to compute, and the 500 stagewise estimates using $\epsilon = 0.005$ took roughly 1.5 seconds. The max flow approach required an average of about 110 seconds to compute 100 solutions, with the majority of computation time spent on solutions at higher levels of regularization (which, here, correspond to lower mean squared errors). Finally, the estimate from each method that minimized mean squared error is also plotted in Figure 6; all look very similar and do a visually reasonable job of recovering the underlying image. That the stagewise approach can deliver such high-quality denoised images with simple, cheap iterations is both fortuitous and surprising.

The second example considers the stagewise algorithm for a larger-scale image denoising task, based on a real 640×480 image, of Lake Pukaki in front of Mount Cook, New Zealand. We worked with each color channel—red, green, blue—separately, and the pixel values were scaled to lie between 0 and 1. For each of these three images, we added independent $N(0, 0.5)$ noise to the pixel values, and ran the stagewise algorithm with $\epsilon = 0.005$ for 650 steps. We chose this number of steps because the achieved mean squared error (averaged over the three color channels) roughly began to rise after this point. We then recombined the three denoised images—on the red, green, blue color channels—to form a single image. See Figure 7. Visually, the reconstructed image is remarkably close to the original one, especially considering the input noisy image on which it is computed. The stagewise algorithm took a total of around 21 seconds to produce this result; recall, though, that in this time it actually produced $650 \times 3 = 1950$ fused lasso estimates (650 steps in three different image denoising tasks, one for each color).

4.4 Choice of Step Size

We discuss a main practical issue when running the stagewise algorithm: choice of the step size ϵ. Of course, when ϵ is too small, the algorithm is less efficient, and when ϵ is too large, the stagewise estimates can fail to span the full regularization path (or a sizeable portion of it). Our heuristic suggestion therefore is to start with a large step size ϵ, and plot the progress of the achieved loss $f(x^{(k)})$ and regularizer $g(x^{(k)})$ function values across steps $k = 1, 2, 3, \ldots$ of the algorithm. With a proper choice of ϵ, note that we should see $f(x^{(k)})$ monotone decreasing with k, as well as $g(x^{(k)})$ monotone increasing with k (this is true of $f(\hat{x}(t))$ and $g(\hat{x}(t))$ as we increase the regularization parameter t, in the exact solution computation). If ϵ is too large, then it seems to be the tendency in practice that the achieved values $f(x^{(k)})$ and $g(x^{(k)})$, $k = 1, 2, 3, \ldots$ stop their monotone progress at some point, and alternate back and forth. Figure 8 illustrates this behavior. Once encountered,



Algorithm timings	
Method	Runtime
Exact: maximum flow, 100 solutions	109.04 (6.21)
Stagewise: $\epsilon = 0.0025$, 6000 estimates	15.11 (0.18)
Stagewise: $\epsilon = 0.25$, 500 estimates	1.26 (0.02)

Figure 6: Comparisons between exact 2d fused lasso solutions and stagewise estimates on a synthetic image denoising example. The true underlying 300×200 image is displayed in the middle of the top row. (A color scale is applied for visualization purposes, see the left end of the bottom row.) Over 10 noisy perturbations of this underlying image, with one such example shown in the right plot of the top row, we compare averaged mean squared errors of the exact solutions and stagewise estimates, in the left plot of the top row. Average timings for these methods are given in the bottom table. (Standard deviations are denoted by dotted lines in the error plots, and are in parentheses in the table.) The stagewise estimates have competitive mean squared errors and are fast to compute. The bottom row of plots shows the optimal image (i.e., that minimizing mean squared error) from each method, based on the single noisy image in the top right.

Original image:

Noisy version:

Stagewise, $\epsilon = 0.001$, 650 steps:

(computed in 21.34 seconds)

Figure 7: A more realistic image denoising example using stagewise. We began with a 640×480 photograph of Lake Pukaki and Mount Cook, in New Zealand, shown at the top. Working with each color channel separately, we added noise to form the middle noisy image, and ran the stagewise algorithm to eventually yield the bottom image, a nice reconstruction.

an appropriate response would be decrease ϵ (say, halve it), and continue the stagewise algorithm from the last step before this alternating pattern surfaced.

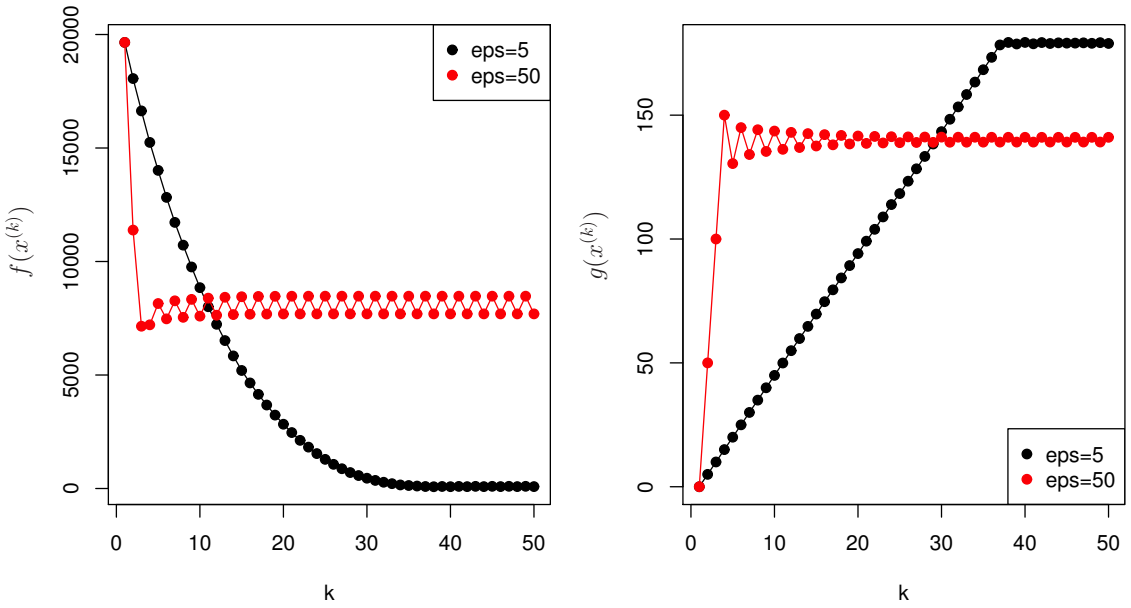


Figure 8: An example displaying a common tendency of stagewise estimates under a choice of step size ϵ that is too large. We used the group lasso regression data setup from Figure 4 (uncorrelated case). Both the achieved loss $f(x^{(k)})$ (left plot) and regularizer $g(x^{(k)})$ (right plot) function values should be monotonic across steps $k = 1, 2, 3, \ldots$. We see that for the larger step size $\epsilon = 50$ (in red), progress halts and an alternating pattern begins, with both sequences; for the smaller step size $\epsilon = 5$ (in black), progress continues all the way until the end of the path.

The heuristic guideline above attempts to produce the largest step size ϵ that still produces an expansive regularization path of stagewise estimates. This ignores the subtlety that a larger choice ϵ may offer suboptimal statistical performance, even if the corresponding estimates span the full path. This was seen in some examples of Section 4 (e.g., matrix completion, in Figure 5), but not in others (e.g., group lasso regression, in Figure 4). The issue of tuning ϵ for optimal statistical performance is more complex and problem dependent. Although it is clearly important, we do not study this task in the current paper. We mention the (somewhat obvious) point that strategies like cross-validation (if applicable, in the given problem setting) could be helpful here.

5. Suboptimality Bounds for Stagewise Estimates

This section focuses on theoretical suboptimality guarantees for the general stagewise algorithm, and proposes a new shrunken variant of the stagewise method.

5.1 General Stagewise Suboptimality

We present a suboptimality bound for estimates produced by the general stagewise algorithm, restricting our attention to a norm regularizer g. The following result makes use of the dual norm g^* of g which, recall, is defined as $g^*(x) = \max_{g(z) \leq 1} x^T z$. Its proof is based on recursively tracking a duality gap for the general problem (4), and is deferred until Online Appendix A.5.

Theorem 1 *Consider the general problem* (4)*, assuming that f is differentiable and convex, and g is a norm. Assume also that ∇f is Lipschitz with respect to the pair g^*, g with constant L, i.e.,*

$$g^*\big(\nabla f(x) - \nabla f(y)\big) \leq L \cdot g(x-y), \quad \text{all } x, y.$$

Fix a regularization parameter value t of interest, and consider running the general stagewise algorithm, Algorithm 2, from $x^{(0)} = \hat{x}(t_0)$, a solution in (4) *at a parameter value $t_0 \leq t$. Suppose that we run the algorithm for k steps, with step size ϵ, such that $t_k = t_0 + k\epsilon = t$. The resulting stagewise estimate $x^{(k)}$ satisfies*

$$f(x^{(k)}) - f(\hat{x}(t)) \leq L(t^2 - t_0^2) + L(t - t_0)\epsilon.$$

Therefore, if we consider the limiting stagewise estimate at the parameter value t, denoted by $\tilde{x}(t)$, as the step size $\epsilon \to 0$, then such an estimate satisfies

$$f(\tilde{x}(t)) - f(\hat{x}(t)) \leq L(t^2 - t_0^2).$$

Remark 1. In the theorem, the kth stagewise estimate $x^{(k)}$ is taken to be an approximate solution at the static regularization parameter value $t_k = t_0 + k\epsilon$, not at the dynamic value $t_k = g(x^{(k)})$, as we have been considering so far. It is easy to see that with the static choice $t_k = t_0 + k\epsilon$, we have $g(x^{(k)}) \leq t_k$, so that $x^{(k)}$ is still feasible at the parameter t_k. Furthermore, this choice simplifies the analysis, and would also simplify running the algorithm in practice (when g is expensive to compute, e.g., in the trace norm setting).

Remark 2. The assumptions that f is differentiable and that its gradient ∇f is Lipschitz continuous are fairly standard in the analysis of optimization algorithms; usually the Lipschitz assumption is made with respect to a prespecified pair of primal and dual norms, but here instead we rely on the pair naturally suggested by the problem (4), namely, g, g^*. For example, in the least squares setting, $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$, with an arbitrary norm g as the regularizer, the Lipschitz constant of ∇f is

$$L = \max_{u \neq 0} \frac{g^*(X^T X u)}{g(u)},$$

which we might write as $L = \|X^T X\|_{g,g^*}$ in the spirit of matrix norms.

Remark 3. The theorem can be extended to the case when g is a seminorm regularizer. As written, the Lipschitz constant L would be infinite if g has a nontrivial null space N_g that overlaps with ∇f, as made precise in (17). However, we could g^* redefine as

$$g^*(x) = \max_{z \in N_g^\perp,\, g(z) \leq 1} x^T z,$$

and one can then check that, under the same conditions, the proof of Theorem 1 goes through just as before, but now the bounds apply to the modified stagewise estimates in (18), (19).

5.2 Shrunken Stagewise Framework

For reasons that will become apparent, we introduce a shrunken version of the stagewise estimates.

Algorithm 3 (Shrunken stagewise procedure)

Fix $\epsilon > 0$, $\alpha \in (0,1)$, $t_0 \in \mathbb{R}$. Set $x^{(0)} = \hat{x}(t_0)$, a solution in (4) at $t = t_0$. Repeat, for $k = 1, 2, 3, \ldots$,

$$x^{(k)} = \alpha x^{(k-1)} + \Delta, \tag{47}$$

$$\textit{where } \Delta \in \operatorname*{argmin}_{z \in \mathbb{R}^n} \langle \nabla f(x^{(k-1)}), z \rangle \;\text{ subject to }\; g(z) \leq \epsilon. \tag{48}$$

The only difference between Algorithm 3 and the existing stagewise proposal in Algorithm 2 is that the update step in (47) shrinks the current iterate $x^{(k-1)}$ by a constant amount $\alpha < 1$, before adding the direction Δ. Note that in the case of unbounded stagewise updates, we would replace (48) by the subspace constrained version (19), as explained in Section 2.3.

Before we give examples or theory, we motivate the study of the shrunken stagewise algorithm from a conceptual point of view. It helps to think about lasso regression in particular, with $f(\beta) = \frac{1}{2}\|y - X\beta\|_2^2$ and $g(\beta) = \|\beta\|_1$. Recall that in this case, the general stagewise procedure reduces to classical forward stagewise regression, in Algorithm 1. A step k, forward stagewise updates the component i of the estimate $\beta^{(k-1)}$ such that the variable X_i has the largest absolute inner product with the residual $y - X\beta^{(k-1)}$; further, it moves $\beta_i^{(k-1)}$ in a direction given by the sign of this inner product. It is intuitively clear why such a procedure generally yields monotone component paths: if X_i has a large positive inner product with the residual, and we add a small amount ϵ to the ith coefficient, then in the next step, X_i will still have a large positive inner product with the residual. This inner product will have been slightly decremented due to the change in ith coefficient, but we will continue to increment the ith coefficient by ϵ (decrement the ith inner product) until another variable attains a comparable inner product with the residual. In other words, the ith component path computed by forward stagewise will increase monotonically, and eventually flatten out.

So how does nonmonoticity occur in stagewise paths? Keeping with the above thought experiment, in order for the ith coefficient path to decrease at some point, the variable X_i must achieve a *negative* inner product with the residual, and this must be largest in magnitude compared to the inner products from all other variables. Given that X_i had a large positive inner product with the residual in previous iterations, this seems highly unlikely, especially in a high-dimensional setting with many variables in total. But we know from many examples that the components of the exact lasso solution path can exhibit many nonmonotonicities, even very early on in the regularization path, and even in high-dimensional settings. To recover the exact path with a stagewise-like algorithm, therefore, some change needs to be made to counteract the momentum gathered over successive updates. Zhao and Yu (2007) do just this, as discussed in the introduction, by adding an explicit backward step to the stagewise routine in which coefficients are driven towards zero as long as this decreases the loss by a significant amount.

An arguably simpler way to achieve a roughly similar effect is to shrink all coefficients towards zero at each step. This is what is done by the shrunken stagewise method, in Algorithm 3, via the parameter $\alpha < 1$. In shrunken stagewise for lasso regression, the importance of each variable wanes over steps of the algorithm. Thus, in the absence of attention from the stagewise update mechanism, a coefficient path slides towards zero, instead of leveling off; for a coefficient path to depart from zero, or even remain at a constant level, it must regain the attention of the update mechanism by repeatedly achieving the maximal absolute inner product. This actually represents a fairly different philosophy from the pure stagewise approach (with $\alpha = 1$) and the two can be crudely contrasted as follows: pure stagewise keeps coefficients at constant levels, unless there is good reason to move them away from zero; shrunken stagewise drives coefficients to zero, unless there is good reason to keep them on their current trajectories.

We give a small example of shrunken stagewise applied to lasso regression, with $n = 20$ observations and $p = 10$ variables. The rows of the predictor matrix $X \in \mathbb{R}^{20\times 10}$ were drawn independently from a Gaussian distribution with mean zero, and a covariance matrix having unit diagonals and constant off-diagonals $\rho = 0.8$. The underlying coefficient vector $\beta^* \in \mathbb{R}^{10}$ had dense support, with all entries drawn from $N(0, 1)$, and the observations y were formed by adding independent $N(0, 1)$ noise to $X\beta^*$. Figure 9 shows the exact lasso solution path on the left panel, the stagewise path in the middle panel, and the shrunkage stagewise path on the right. We can see that, at various points, components of the exact lasso path become nonmonotone, and as expected, the corresponding the stagewise component paths ignore this trend and level out. The shrunken stagewise component paths pick up on the nonmonotonicities and actually mimick the exact ones quite closely. We note that the stagewise and shrunken stagewise algorithms were not run here for efficiency, but were run at fine resolution to reveal their limiting behaviors; both used a small step size $\epsilon = 0.0001$, and the latter used a shrinkage factor $\alpha = \epsilon/10$. The two required 100,000 and 500,000 steps, respectively.

To be upfront, we remark that the shrunken stagewise method is *not a computationally efficient approach*, and we do not advocate its use in practice. The stagewise algorithm in the above example could have been run, e.g., with $\epsilon = 0.01$ and for 100 steps, and this would have yielded a sequence of estimates with effectively the same pattern. But to capture the nonmonotonicities present in the exact solution path, larger step sizes do not suffice for shrunken stagewise, and the algorithm needs to be run with $\epsilon = 0.0001$ and for 500,000 steps—this is clearly not desirable for such a small example with $n = 20$ and $p = 10$, and it does not bode well for scalability. We will see in what follows that the shrunken stagewise estimates provide a bridge between pure stagewise estimates and exact solutions in the general convex regularization problem (4). Hence we view the shrunken stagewise estimates as interesting and worthwhile *because* they provides this connection.

The main reason we choose to study the shrinkage strategy in Algorithm 3, as opposed to, say, backward steps, is that the shrinkage approach applies outside of the lasso regularization setting; as far as we can tell, there is no natural analog of backwards steps beyond the sparse setting. In fact, in the general problem setup, the shrinkage factor α in Algorithm 3 somewhat roughly mirrors what is done by Frank-Wolfe (this is really a different strategy, but still, it is one that computes exact solutions; compare equations (48) and (2) from Online Appendix A.1). A general interpretation of the shrinkage operation in (48) is that

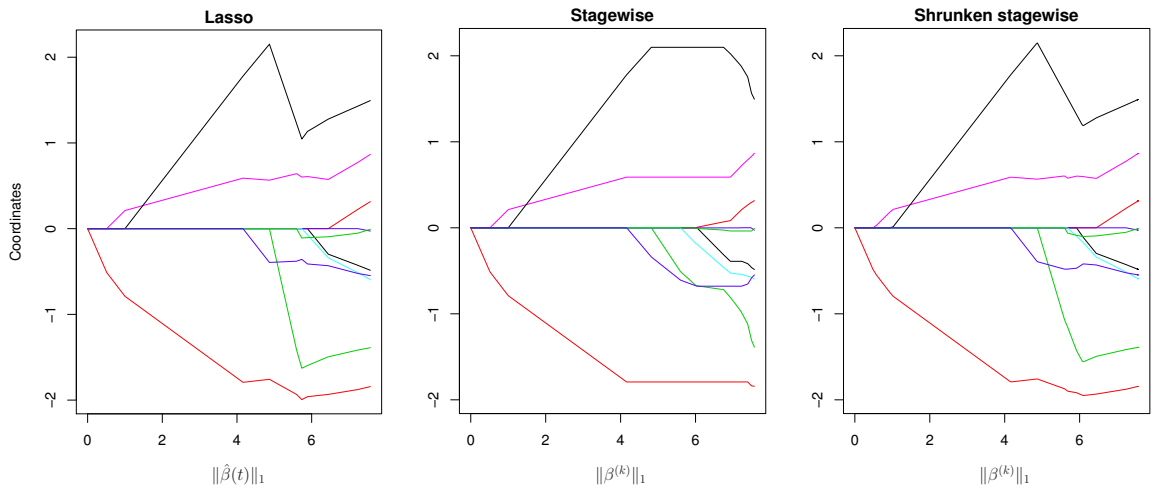


Figure 9: Exact, stagewise, and shrunken stagewise paths for a small lasso regression problem with $n = 20$ observations, and $p = 10$ correlated predictors. When components of the lasso solution path become nonmonotone (e.g., top black path, and bottom red path), the corresponding stagewise ones are more stable and remain at a constant level, but shrunken stagewise matches the nonmonotonicities.

it lessens the dependence of the stagewise estimates on the computed history, i.e., decreases the stability of the computed stagewise component paths, and implicitly allows for more weight to be placed on the local update directions. Empirical examples with, e.g., group lasso regression or matrix completion confirm that shrunken stagewise estimates can be tuned to track the exact solution path even when the pure stagewise path deviates from it. We do not examine these cases here but instead turn to theoretical development.

5.3 Shrunken Stagewise Suboptimality

As in Section 5.1, we assume that g is a norm, and write g^* for its dual norm. We also consider the kth shrunken stagewise estimate $x^{(k)}$ as an approximate solution in the general problem (4) at a static value of the regularization parameter, defined recursively as $t_k = \alpha t_{k-1} + \epsilon$. A straightforward inductive argument shows that $g(x^{(k)}) \leq t_k$, i.e., the estimate $x^{(k)}$ is feasible for the problem (4) at $t = t_k$. Under this setup, the same limiting suboptimality bound as in Theorem 1 can be established for the shrunken stagewise estimates. For the sake of space, we do not present this result. Instead we show that, under additional conditions, the shrunken stagewise estimates overcome the stability inherent to stagewise, and achieve the idealized behavior suggested by Figure 9, i.e., they converge to exact solutions along the path. See Online Appendix A.6 for the proof.

Theorem 2 *Consider the general problem* (4)*. Assume, as in Theorem 1, that the loss function f is differentiable and convex, the regularizer g is a norm, and ∇f is Lipschitz with respect to g^*, g, having Lipschitz constant L. Fix a parameter value t, and consider running the shrunken stagewise algorithm, Algorithm 3, from $x^{(0)} = \hat{x}(t_0)$, a solution in* (4) *at a parameter value $t_0 \leq t$. Consider the limiting estimate $\tilde{x}(t)$ at the parameter value t,*

as both $\epsilon \to 0$ *and* $\alpha \to 1$. *Suppose that*

$$\frac{1-\alpha}{\epsilon} \to 0 \quad \textit{and} \quad \frac{1-\alpha}{\epsilon^2} \to \infty.$$

Let $k = k(\epsilon, \alpha)$ *denote the number of steps taken by the shrunken stagewise algorithm to reach the parameter value* $t_k = t$; *note that* $k \to \infty$ *as* $\epsilon \to 0$, $\alpha \to 1$. *Define the effective Lagrange parameters* $\lambda_i = g^*(\nabla f(x^{(i)}))$, $i = 1, \ldots k$, *and assume that these parameters exhibit a weak type of decay:*

$$\begin{aligned} &\lambda_i/t_i \geq CL, \quad i = 1, \ldots r-1, \\ &\lambda_r/t_r \leq \frac{(C+1)\theta^2 - 2}{2} L, \end{aligned} \tag{49}$$

for some $r < k$, *with* $r/k \to \theta \in (0,1)$, *and some constant* C. *Then the limiting shrunken stagewise estimate* $\tilde{x}(t)$ *at the parameter value* t, *as* $\epsilon \to 0$ *and* $\alpha \to 1$, *satisfies*

$$f(\tilde{x}(t)) = f(\hat{x}(t)),$$

i.e., $\tilde{x}(t)$ *is a solution in* (4) *at the parameter value* t.

Remark 1. The result above can be extended to the case when g is a seminorm. We simply need to redefine g^* and the updates in order to accommodate the possibly nontrivial null space N_g of g, as discussed in the third remark following Theorem 1.

Remark 2. The assumption in (49) of Theorem 2 stands out as technical assumption that is hard to interpret. This condition is used in the proof to control a term in the duality gap expansion that involves differences of $g^*(\nabla f(x^{(i)}))$ across successive iterations $i, i+1$. The theorem refers to such a quantity, $\lambda_i = g^*(\nabla f(x^{(i)}))$, as the "effective Lagrange parameter" at $x^{(i)}$. To explain this, consider the stationarity condition for the problem (4),

$$\nabla f(x) + \lambda v = 0,$$

where $v \in \partial g(x) = \operatorname{argmax}_{g^*(z) \leq 1} x^T z$. This implies that $\nabla f(x) = -\lambda v$, or $g^*(\nabla f(x)) = \lambda g^*(v) = \lambda$, which gives an expression for the Lagrange parameter associated with a solution of the constrained problem (4). As $x^{(i)}$ is not a solution, but an approximate one, we call $\lambda_i = g^*(\nabla f(x^{(i)}))$ its effective Lagrange parameter.

The condition (49) says that until some number of steps r along the path, the ratio of effective Lagrange parameters λ_i to bound parameters t_i must not be too small, and then at step r it must not be too large. This is a formulation of a type of weak decay of λ_i/t_i, $i = 1, 2, 3, \ldots$. It is not intuitively clear to us when (i.e., in what kinds of problems) we should expect this condition to be satisfied. We can, however, inspect it empirically. For the example lasso problem in Figure 9 (where, recall, the shrunken stagewise path appears to approach the exact solution path), we plot the ratio λ_i/t_i, $i = 1, 2, 3, \ldots$ in Figure 10. This ratio displays a sharp decay across steps of the algorithm, and so, at least empirically, the assumption (49) seems reasonable. We suspect that in general, the two hard bounds in (49) can be replaced by a more natural decay condition, and furthermore, there are characterizable problem classes with sharp decays of the Lagrange to bound parameter ratios. These are topics for future work.

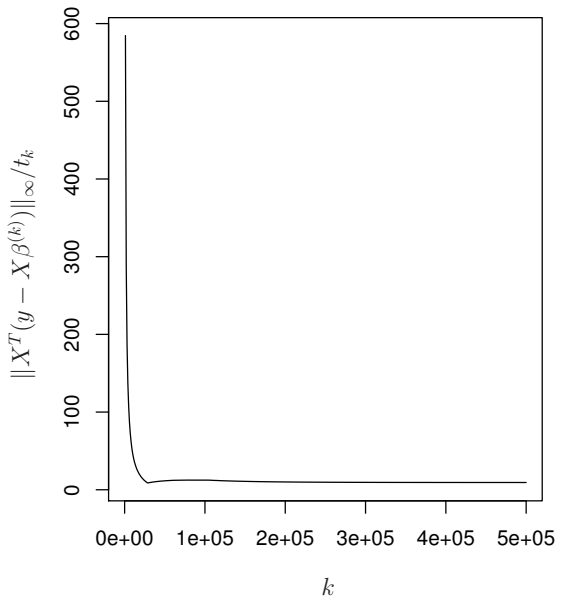


Figure 10: A plot of $\lambda_k/t_k = \|X^T(y - X\beta^{(k)})\|_\infty/t_k$ across steps k of the shrunken stagewise algorithm, for the lasso data set of Figure 9. This decay roughly verifies the condition (49) of Theorem 2, needed to ensure the convergence of shrunken stagewise estimates to exact solutions.

6. Discussion

We presented a framework for computing incremental stagewise paths in a general regularized estimation setting, defined by minimizing a differentiable convex loss function subject to a convex constraint. The stagewise estimates are explicitly and efficiently computable for a wide variety of problems, and they provide an approximate solution path for the underlying convex problem of interest, but exhibit generally more stability as the regularization parameter changes. In some situations this approximation (i.e., the discrepancy between stagewise estimates and solutions) appears empirically to be quite tight, and in others it does not. All in all, however, we have found that the stagewise estimates essentially always offer competitive statistical performance (as measured, e.g., by test error) with that of exact solutions. This suggests that they should be a point of study, even apart from their ability to approximate solution paths of convex problems, and a rigorous (theoretical) characterization of the statistical properties of stagewise estimates is an important direction to pursue in the future. There are many other potential topics for future work, as alluded to throughout the paper. It is our hope that other researchers will take an interest too, and that this paper marks the beginning of a deeper understanding of stagewise capabilities.

Acknowledgements

This paper was inspired by an attempt to explain the intuitive connection between forward stagewise regression and the lasso, in preparing lectures for a graduate class on optimization

at Carnegie Mellon University. We thank co-teacher Geoff Gordon and the students of this class for early motivating conversations. We also thank Rob Tibshirani, Jerry Friedman, Jonathan Taylor, Jacob Bien, and Lester Mackey for their helpful feedback. We are grateful to Jacob Bien for his understanding and patience throughout our (unusually slow) writing process, and to Lester Mackey for enlightening discussion on the Frank-Wolfe connection. Lastly, we would like to thank the editors and referees who reviewed this paper, as they provided extremely helpful and constructive reports. We gratefully acknowledge the funding support from NSF grant DMS-1309174.

References

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. *Advances in Neural Information Processing Systems*, 19, 2006.

Sergey Bakin. *Adaptive Regression and Model Selection in Data Mining Problems.* PhD thesis, School of Mathematical Sciences, Australian National University, 1999.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, 2004.

Peter Buhlmann and Bin Yu. Boosting. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):69–74, 2010.

Emmanuel J. Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Emmanuel J. Candes and Terence Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

Antonin Chambolle and Jerome Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84:288–307, 2009.

Jianhui Chen and Jieping Ye. Sparse trace norm regularization. *Computational Statistics*, 29(3–4):623–629, 2014.

Norman Draper and Henry Smith. *Applied Regression Analysis.* Wiley, New York, 1966.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32(2):407–499, 2004.

M. Efroymson. Stepwise regression—a backward and forward look. *Eastern Regional Meetings of the Institute of Mathematical Statistics*, 1966.

Paul Eilers and Brian Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121, 1996.

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 32(1–2):95–110, 1956.

Jerome Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1190–1232, 2001.

Jerome Friedman. Fast sparse regression and classification. Unpublished manuscript, `http://www-stat.stanford.edu/~jhf/ftp/GPSpub.pdf`, 2008.

Jerome Friedman and Bogdan Popescu. Gradient directed regularization. Unpublished manuscript, `http://www-stat.stanford.edu/~jhf/ftp/pathlite.pdf`, 2004.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 1996. Third edition.

Peter Green and Bernard Silverman. *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*. Chapman & Hall/CRC Press, Boca Raton, 1994.

Zaid Harchaoui, Matthijs Douze, Mattis Paulin, Miroslav Dudik, and Jerome Malick. Large-scale image classification with trace-norm regularization. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3386–3393, 2012.

Trevor Hastie, Jonathan Taylor, Robert Tibshirani, and Guenther Walther. Forward stagewise regression and the monotone lasso. *Electronic Journal of Statistics*, 1:1–29, 2007.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Springer, New York, 2009. Second edition.

Jean-Baptiste Hiriart-Urruty and Claude Lemarechal. *Convex Analysis and Minimization Algorithms*. Springer, Berlin, 1993. Two volumes.

Arthur Hoerl and Robert Kennard. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. *Proceedings of the International Conference on Machine Learning*, 30, 2013.

Martin Jaggi and Marek Sulovsky. A simple algorithm for nuclear norm regularized problems. *Proceedings of the International Conference on Machine Learning*, 27, 2010.

J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2):339–360, 2009.

Eliot Knudsen. *Stagewise Regression: Competing with the State of the Art via an Efficient, Simple, Iterative Algorithm*. Undergraduate honors thesis, Department of Statistics, Carnegie Mellon University, 2013.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11:2287–2322, 2010.

Lukas Meier, Sara van de Geer, and Peter Buhlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B*, 70(1):53–71, 2008.

Guillame Obozinski, Ben Taskar, and Michael Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2): 231–252, 2010.

James Ramsay. Parameter flows. Unpublished manuscript, 2005.

Saharon Rosset, Ji Zhu, and Trevor Hastie. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5:941–973, 2004.

Leonid I. Rudin, Stanley Osher, and Emad Faterni. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259–268, 1992.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse group lasso. *Journal of Computational and Graphical Statistics*, 22(2), 2013.

Choon Hui Teo, Quoc Le, Alex Smola, and S. V. N. Vishwanathan. A scalable modular convex solver for regularized risk minimization. *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 13, 2007.

Choon Hui Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc Le. Bundle methods for regularized risk minimization. *Journal of Machine Learning Research*, 11:311–365, 2010.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288, 1996.

Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67 (1):91–108, 2005.

Ryan J. Tibshirani. The lasso problem and uniqueness. *Electronic Journal of Statistics*, 7: 1456–1490, 2013.

Ryan J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *Annals of Statistics*, 42(1):285–323, 2014.

Ryan J. Tibshirani and Jonathan Taylor. The solution path of the generalized lasso. *Annals of Statistics*, 39(3):1335–1371, 2011.

Andrey Tikhonov. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39 (5):195–198, 1943.

Berwin Turlach, William Venables, and Stephen Wright. Simultaneous variable selection. *Technometrics*, 47(3):349–363, 2005.

Grace Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, Philadelphia, 1990.

Yu-Xiang Wang, James Sharpnack, Alex Smola, and Ryan J. Tibshirani. Trend filtering on graphs. *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 18:1042–1050, 2015.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B*, 68(1):49–67, 2006.

Peng Zhao and Bin Yu. Stagewise lasso. *Journal of Machine Learning Research*, 8:2701–2726, 2007.

Journal of Machine Learning Research 16 (2015) 2589-2609 Submitted 9/14; Revised 3/15; Published 12/15

Counting and Exploring Sizes of Markov Equivalence Classes of Directed Acyclic Graphs

Yangbo He HEYB@PKU.EDU.CN
Jinzhu Jia JZJIA@MATH.PKU.EDU.CN
LMAM, School of Mathematical Sciences, LMEQF, and Center for Statistical Science
Peking University, Beijing 100871, China

Bin Yu BINYU@STAT.BERKELEY.EDU
Departments of Statistics and EECS
University of California at Berkeley, Berkeley, CA 94720

Editor: Isabelle Guyon and Alexander Statnikov

Abstract

When learning a directed acyclic graph (DAG) model via observational data, one generally cannot identify the underlying DAG, but can potentially obtain a Markov equivalence class. The size (the number of DAGs) of a Markov equivalence class is crucial to infer causal effects or to learn the exact causal DAG via further interventions. Given a set of Markov equivalence classes, the distribution of their sizes is a key consideration in developing learning methods. However, counting the size of an equivalence class with many vertices is usually computationally infeasible, and the existing literature reports the size distributions only for equivalence classes with ten or fewer vertices.

In this paper, we develop a method to compute the size of a Markov equivalence class. We first show that there are five types of Markov equivalence classes whose sizes can be formulated as five functions of the number of vertices respectively. Then we introduce a new concept of a rooted sub-class. The graph representations of rooted subclasses of a Markov equivalence class are used to partition this class recursively until the sizes of all rooted subclasses can be computed via the five functions. The proposed size counting is efficient for Markov equivalence classes of sparse DAGs with hundreds of vertices. Finally, we explore the size and edge distributions of Markov equivalence classes and find experimentally that, in general, (1) most Markov equivalence classes are half completed and their average sizes are small, and (2) the sizes of sparse classes grow approximately exponentially with the numbers of vertices.

Keywords: directed acyclic graphs, Markov equivalence class, size distribution, causality

1. Introduction

Graphical models based on directed acyclic graphs (DAGs) are commonly used to derive the dependent or causal relationships in many fields such as sociology, epidemiology, and biology (Finegold and Drton, 2011; Friedman, 2004; Heckerman et al., 1999; Jansen et al., 2003; Maathuis et al., 2009). A DAG can be used to represent causal relationships of variables, where the directed edges connect the causes and their direct effects. In general, observational data is not sufficient to distinguish the underlying DAG from its statistically

equivalent DAGs; however, it is possible to learn the Markov equivalence class that contains these equivalent DAGs (Pearl, 2000; Spirtes et al., 2001). This has led to many works that try to learn a Markov equivalence class or to learn causality based on a given Markov equivalence class from observational or experimental data (Castelo and Perlman, 2004; Chickering, 2002; He and Geng, 2008; Maathuis et al., 2009; Perlman, 2001).

The size of a Markov equivalence class is the number of DAGs in the class. This size has been used in papers to design causal learning approaches or to evaluate the "complexity" of a Markov equivalence class in causal learning. For example, He and Geng (2008) proposes several criteria, all of which are defined on the sizes of Markov equivalence classes, to minimize the number of interventions; this minimization makes helpful but expensive interventions more efficient. Based on observational data, Maathuis et al. (2009) introduces a method to estimate the average causal effects of the covariates on the response by considering the DAGs in the equivalence class; the size of the class determines the complexity of the estimation. Chickering (2002) shows that causal structure search in the space of Markov equivalence class models could be substantially more efficient than that in the space of DAG models if most sizes of Markov equivalence classes are large.

The size of a small Markov equivalence class is usually counted via traversal methods that list all DAGs in the Markov equivalence class (Gillispie and Perlman, 2002). However, if the class is large, it may be infeasible to list all DAGs. For example, as we will show later in our experiments, the size of a Markov equivalence class with 50 vertices and 250 edges could be greater than 10^{24}. To our knowledge, there are no efficient methods to compute the size of a large Markov equivalence class; approximate proxies, such as the number of vertices and the number of spanning trees related to the class, have been used instead of the exact size in the literature (Chickering, 2002; He and Geng, 2008; Meganck et al., 2006).

Computing the size of a Markov equivalence class is the focus of this article. We first discuss Markov equivalence classes whose sizes can be calculated just through the numbers of vertices and edges. Five explicit formulas are given to obtain the sizes for five types of Markov equivalence classes respectively. Then, we introduce rooted sub-classes of a Markov equivalence class and discuss the graphical representations of these sub-classes. Finally, for a general Markov equivalence class, we introduce a counting method by recursively partitioning the Markov equivalence class into smaller rooted sub-classes until all rooted sub-classes can be counted with the five explicit formulas.

Next, we also report new results about the size and edge distributions of Markov equivalence classes for sparse graphs with hundreds of vertices. By using the proposed size counting method in this paper and an MCMC sampling method recently developed by He et al. (2013a,b), we experimentally explore the size distributions of Markov equivalence classes with large numbers of vertices and different levels of edge sparsity. In the literature, the size distributions are studied in detail just for Markov equivalence classes with up to 10 vertices by traversal methods (Gillispie and Perlman, 2002).

The rest of the paper is arranged as follows. In Section 2, we provide a brief review of the concept of a Markov equivalence class. In Section 3, we propose efficient algorithms to calculate the size of a Markov equivalence class. In Section 4, we study the sizes of Markov equivalence classes experimentally. We conclude in Section 5 and finally present all proofs in the Appendix.

2. Markov Equivalence Class

A graph $\mathcal{G}$ consists of a vertex set V and an edge set E. A graph is directed (undirected) if all of its edges are directed (undirected). A sequence of edges that connect distinct vertices in V, say $\{v_1, \cdots, v_k\}$, is called a path from v_1 to v_k if either $v_i \to v_{i+1}$ or $v_i - v_{i+1}$ is in E for $i = 1, \cdots, k-1$. A path is *partially directed* if at least one edge in the path is directed. A path is directed (undirected) if all edges are directed (undirected). A *cycle* is a path from a vertex to itself.

A *directed acyclic graph* (DAG) $\mathcal{D}$ is a directed graph without any directed cycle. Let V be the vertex set of $\mathcal{D}$ and τ be a subset of V. The induced subgraph $\mathcal{D}_\tau$ of $\mathcal{D}$ over τ, is defined to be the graph whose vertex set is τ and whose edge set contains all of those edges of $\mathcal{D}$ with two end points in τ. A *v-structure* is a three-vertex induced subgraph of $\mathcal{D}$ like $v_1 \to v_2 \leftarrow v_3$. A graph is called a *chain graph* if it contains no partially directed cycles. The isolated undirected subgraphs of the chain graph after removing all directed edges are the chain components of the chain graph. A *chord* of a cycle is an edge that joins two nonadjacent vertices in the cycle. An undirected graph is *chordal* if every cycle with four or more vertices has a chord.

A graphical model is a probabilistic model for which a DAG denotes the conditional independencies between random variables. *A Markov equivalence class* is a set of DAGs that encode the same set of conditional independencies. Let the *skeleton* of an arbitrary graph $\mathcal{G}$ be the undirected graph with the same vertices and edges as $\mathcal{G}$, regardless of their directions. Verma and Pearl (1990) proves that two DAGs are *Markov equivalent* if and only if they have the same skeleton and the same v-structures. Moreover, Andersson et al. (1997) shows that a Markov equivalence class can be represented uniquely by an *essential graph*.

Definition 1 (Essential graph) *The* essential graph *of a DAG $\mathcal{D}$, denoted as $\mathcal{C}$, is a graph that has the same skeleton as $\mathcal{D}$, and an edge is directed in $\mathcal{C}$ if and only if it has the same orientation in every equivalent DAG of $\mathcal{D}$.*

It can be seen that the essential graph $\mathcal{C}$ of a DAG $\mathcal{D}$ has the same skeleton as $\mathcal{D}$ and keeps the v-structures of $\mathcal{D}$. Andersson et al. (1997) also introduces some properties of an essential graph.

Lemma 2 (Andersson et al. (1997)) *Let $\mathcal{C}$ be an essential graph of $\mathcal{D}$. Then $\mathcal{C}$ is a chain graph, and each chain component $\mathcal{C}_\tau$ of $\mathcal{C}$ is an undirected and connected chordal graph, where τ is the vertex set of the chain component $\mathcal{C}_\tau$.*

Let SizeMEC($\mathcal{C}$) denote the size of the Markov equivalence class represented by $\mathcal{C}$ (size of $\mathcal{C}$ for short). Clearly, SizeMEC($\mathcal{C}$) = 1 if $\mathcal{C}$ is a DAG; otherwise $\mathcal{C}$ may contain more than one chain component, denoted by $\mathcal{C}_{\tau_1}, \cdots, \mathcal{C}_{\tau_k}$. From Lemma 2, each chain component is an undirected and connected chordal graph (UCCG for short); and any UCCG is an essential graph that represents a Markov equivalence class (Andersson et al., 1997). We can calculate the size of $\mathcal{C}$ by counting the DAGs in Markov equivalence classes represented by its chain components using the following equation (Gillispie and Perlman, 2002; He and Geng, 2008):

$$\text{SizeMEC}(\mathcal{C}) = \prod_{i=1}^{k} \text{SizeMEC}(\mathcal{C}_{\tau_i}). \tag{1}$$

To count the size of Markov equivalence class represented by a UCCG, we can generate all equivalent DAGs in the class. However, when the number of vertices in the UCCG is large, the number of DAGs in the corresponding Markov equivalence class may be huge, and the traversal method proves to be infeasible to count the size. This paper tries to solve this counting problem for Markov equivalence classes of DAGs with hundred of vertices.

3. The Size of Markov Equivalence Class

In order to obtain the size of a Markov equivalence class, it is sufficient to compute the size of Markov equivalence classes represented by undirected and connected chordal graph (UCCGs) according to Lemma 2 and Equation (1). In Section 3.1, we discuss Markov equivalence classes represented by UCCGs whose sizes are functions of the number of vertices. Then in Section 3.2.1, we provide a method to partition a Markov equivalence class into smaller subclasses. Using these methods, finally in Section 3.2.2, we propose a recursive approach to calculate the size of a general Markov equivalence class.

3.1 Size of Markov Equivalence Class Determined by the Number of Vertices

Let $\mathcal{U}_{p,n}$ be an undirected and connected chordal graph (UCCG) with p vertices and n edges. Clearly, the inequality $p-1 \leq n \leq p(p-1)/2$ holds for any UCCG $\mathcal{U}_{p,n}$. When $\mathcal{U}_{p,n}$ is a tree, $n = p-1$ and when $\mathcal{U}_{p,n}$ is a completed graph, $n = p(p-1)/2$. Given p and n, in some special cases, the size of a UCCG $\mathcal{U}_{p,n}$ is completely determined by p. For example, it is well known that a Markov equivalence class represented by a completed UCCG with p vertices contains $p!$ DAGs. Besides the Markov equivalence classes represented by completed UCCGs, there are five types of UCCGs whose sizes are also functions of p. We present them as follows.

Theorem 3 *Let $\mathcal{U}_{p,n}$ be a UCCG with p vertices and n edges. In the following five cases, the size of the Markov equivalence class represented by $\mathcal{U}_{p,n}$ is determined by p.*

1. *If $n = p-1$, we have SizeMEC($\mathcal{U}_{p,n}$) $= p$.*
2. *If $n = p$, we have SizeMEC($\mathcal{U}_{p,n}$) $= 2p$.*
3. *If $n = p(p-1)/2 - 2$, we have* SizeMEC($\mathcal{U}_{p,n}$) $= (p^2 - p - 4)(p-3)!$.
4. *If $n = p(p-1)/2 - 1$, we have* SizeMEC($\mathcal{U}_{p,n}$) $= 2(p-1)! - (p-2)!$.
5. *If $n = p(p-1)/2$, we have SizeMEC($\mathcal{U}_{p,n}$) $= p!$.*

For the UCCGs other than the above five cases, it seems that the sizes of the corresponding Markov equivalence classes cannot be completely determined by the numbers of vertices and edges; the sizes of these Markov equivalence classes may depend on the exact essential graphs. Below, we display several classes of this kind for $n = p+1$ or $n = p(p-1)/2 - 3$ in Example 1.

Example 1. Figure 1 displays four UCCGs. Both $\mathcal{U}_{5,6}$ and $\mathcal{U}'_{5,6}$ have 6 edges, and both $\mathcal{U}_{5,7}$ and $\mathcal{U}'_{5,7}$ have 7 edges. We have that SizeMEC($\mathcal{U}_{5,6}$) $= 13$, SizeMEC($\mathcal{U}'_{5,6}$) $= 12$, SizeMEC($\mathcal{U}_{5,7}$) $= 14$ and SizeMEC($\mathcal{U}'_{5,7}$) $= 30$. Clearly, in these cases, the sizes of Markov equivalence classes are not completely determined by the numbers of vertices and edges.

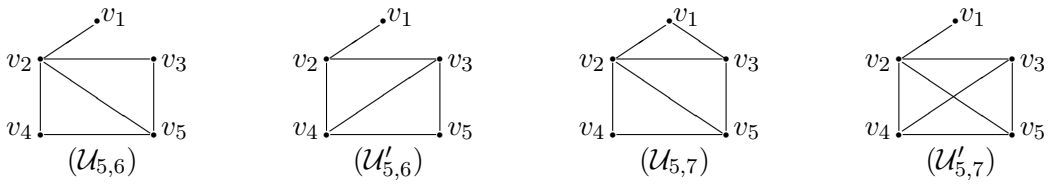


Figure 1: Examples that UCCGs with the same number of edges have different sizes.

3.2 Size of a General Markov Equivalence Class

In this section, we introduce a general method to count the size of a Markov equivalence class. We have shown in Theorem 3 that there are five types of Markov equivalence classes whose sizes can be calculated with five formulas respectively. For one any other Markov equivalence class, we will show in this section that it can be partitioned recursively into smaller subclasses until the sizes of all subclasses can be calculated with the five formulas above. We first introduce the partition method and the graph representation of each subclass in Section 3.2.1. Then provide a size counting algorithm for one arbitrary Markov equivalence class in Section 3.2.2. The proofs of all results in this section can be found in the Appendix.

3.2.1 Methods to Partition a Markov Equivalence Class

Let $\mathcal{U}$ be a UCCG, τ be the vertex set of $\mathcal{U}$ and let $\mathcal{D}$ be a DAG in the equivalence class represented by $\mathcal{U}$. A vertex v is a *root* of $\mathcal{D}$ if all directed edges adjacent to v are out of v, and $\mathcal{D}$ is *v-rooted* if v is a root of $\mathcal{D}$. To count DAGs in the class represented by $\mathcal{U}$, below, we show that all DAGs can be divided into different groups according to the roots of the DAGs and then we calculate the numbers of the DAGs in these groups separately. Each group is called as a rooted sub-class defined as follows.

Definition 4 (*a rooted sub-class*) *Let $\mathcal{U}$ be a UCCG over τ and $v \in \tau$. We define the v-rooted sub-class of $\mathcal{U}$ as the set of all v-rooted DAGs in the Markov equivalence class represented by $\mathcal{U}$.*

The following theorem provides a partition of a Markov equivalence class represented by a UCCG and the proof can be found in Appendix.

Theorem 5 (a rooted partition) *Let $\mathcal{U}$ be a UCCG over $\tau = \{v_1, \cdots, v_p\}$. For any $i \in \{1, \cdots, p\}$, the v_i-rooted sub-class is not empty and this set of p sub-classes forms a disjoint partition of the set of all DAGs represented by $\mathcal{U}$.*

Below we describe an efficient graph representation of v-rooted sub-class. One reason to this graph representation is that for any $v \in \tau$, the number of DAGs in the v-rooted sub-class might be extremely huge and it is computationally infeasible to list all v-rooted DAGs in this sub-class. Using all DAGs in which v is a root, we construct a rooted essential graph in Definition 6.

Definition 6 (rooted essential graph) *Let $\mathcal{U}$ be a UCCG. The v-rooted essential graph of $\mathcal{U}$, denoted by $\mathcal{U}^{(v)}$, is a graph that has the same skeleton as $\mathcal{U}$, and an edge is directed in $\mathcal{U}^{(v)}$ if and only if it has the same orientation in every v-rooted DAG of $\mathcal{U}$.*

From Definition 6, a rooted essential graph has more directed edges than the essential graph $\mathcal{U}$ since the root introduces some directed edges. Algorithm 3 in Appendix shows how to generate the v-rooted essential graph of a UCCG $\mathcal{U}$. We display the properties of a rooted essential graph in Theorem 7 and the proof can be found in Appendix.

Theorem 7 *Let $\mathcal{U}$ be a UCCG and $\mathcal{U}^{(v)}$ be a v-rooted essential graph of $\mathcal{U}$ defined in Definition 6. The following three properties hold for $\mathcal{U}^{(v)}$:*

1. *$\mathcal{U}^{(v)}$ is a chain graph,*
2. *every chain component $\mathcal{U}^{(v)}_{\tau'}$ of $\mathcal{U}^{(v)}$ is chordal, and*
3. *the configuration $v_1 \to v_2 - v_3$ does not occur as an induced subgraph of $\mathcal{U}^{(v)}$.*

Moreover, there is a one-to-one correspondence between v-rooted sub-classes and v-rooted essential graphs, so $\mathcal{U}^{(v)}$ can be used to represent uniquely the v-rooted sub-class of $\mathcal{U}$.

From Theorem 7, we see that the number of DAGs in a v-rooted essential graph $\mathcal{U}^{(v)}$ can be calculated by Equation (1) which holds for any essential graph. To use Equation (1), we have to generate all chain components of $\mathcal{U}^{(v)}$. Below we introduce an algorithm called *ChainCom*($\mathcal{U}$, v) in Algorithm 1 to generate $\mathcal{U}^{(v)}$ and all of its chain components.

Algorithm 1: *ChainCom*$(\mathcal{U}, v)$

Input: $\mathcal{U}$, a UCCG; v, a vertex of $\mathcal{U}$.
Output: $v-$rooted essential graph of $\mathcal{U}$ and all of its chain components.

```
Set A = {v}, B = τ \ v, G = U and O = ∅
while B is not empty do
    Set T = {w : w in B and adjacent to A} ;
    Orient all edges between A and T as c → t in G, where c ∈ A, t ∈ T;
    repeat
        for each edge y − z in the vertex-induced subgraph G_T do
            if x → y − z in G and x and z are not adjacent in G then
                Orient y − z to y → z in G
    until no more undirected edges in the vertex-induced subgraph G_T can be
    oriented;
    Set A = T and B = B \ T;
    Append all isolated undirected graphs in G_T to O;
return G and O
```

We show that Algorithm 1 can generate rooted essential graph and the chain components of this essential graph correctly in the following theorem.

Theorem 8 *Let $\mathcal{U}$ be a UCCG and let v be a vertex in $\mathcal{U}$. Let $\mathcal{O}$ and $\mathcal{G}$ be the outputs of Algorithm 1 given $\mathcal{U}$ and v. Then $\mathcal{G}$ is the v-rooted essential graph $\mathcal{G} = \mathcal{U}^{(v)}$ of $\mathcal{U}$ and $\mathcal{O}$ is the set of all chain components of $\mathcal{U}^{(v)}$.*

The following example displays rooted essential graphs of a UCCG and illustrates how to implement Algorithm 1 to construct a rooted essential graph and how to generate all DAGS in the corresponding rooted sub-classes.

Example 2. Figure 2 displays an undirected chordal graph $\mathcal{U}$ and its rooted essential graphs. There are five rooted essential graphs $\{\mathcal{U}^{(v_i)}\}_{i=1,\cdots,5}$. We need to construct only $\mathcal{U}^{(v_1)}$, $\mathcal{U}^{(v_2)}$ and $\mathcal{U}^{(v_3)}$ since $\mathcal{U}^{(v_4)}$ and $\mathcal{U}^{(v_5)}$ are symmetrical to $\mathcal{U}^{(v_1)}$ and $\mathcal{U}^{(v_3)}$ respectively. Clearly, they satisfy the conditions shown in Theorem 7. Given $\mathcal{U}$ in Figure 2, $\mathcal{U}^{(v_1)}$ is constructed according to Algorithm 1 as follows: (1) set $T = \{v_2, v_3\}$ in which vertices are adjacent to v_1, orient $v_1 - v_2, v_1 - v_3$ to $v_1 \to v_2, v_1 \to v_3$ respectively; (2) set $T = \{v_4, v_5\}$ in which vertices are adjacent to $\{v_2, v_3\}$, orient $v_2 - v_4, v_2 - v_5, v_3 - v_5$ to $v_2 \to v_4, v_2 \to v_5, v_3 \to v_5$ respectively; (3) orient $v_5 - v_4$ to $v_5 \to v_4$ because $v_3 \to v_5 - v_4$ occurs but v_3 and v_4 are not adjacent. By orientating the undirected edges of the chain components of a rooted essential graph with the constraint that no new v-structures and directed cycle occur, we can generate all DAGS in the corresponding sub-class (He and Geng, 2008; Meek, 1995; Verma, 1992). For example, consider $\mathcal{U}^{(v_1)}$ in Figure 2, we get two v_1-rooted DAGs by orienting $v_2 - v_3$ to $v_2 \to v_3$ or $v_2 \leftarrow v_3$.

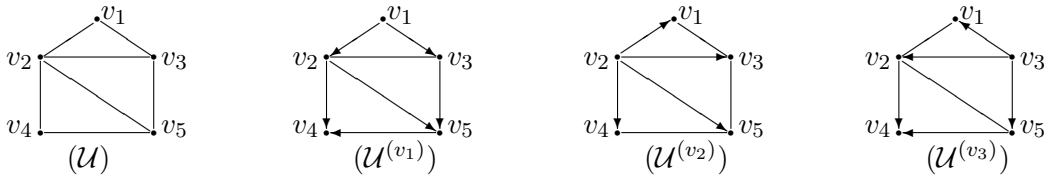


Figure 2: An undirected chordal graph $\mathcal{U}$ and its rooted essential graphs: $\mathcal{U}^{(v_1)}$, $\mathcal{U}^{(v_2)}$, and $\mathcal{U}^{(v_3)}$.

Now we can partition a Markov equivalence class represented by a UCCG into disjoint sub-classes, each of which can be represented by a rooted essential graph. In the next section, we will show how to recursively implement these partitions until the sizes of the subclasses or their essential graphs can be calculated with the five formulas in Theorem 3.

3.2.2 Calculating the Size of a Markov Equivalence Class

Let $\mathcal{U}$ be an undirected and connected chordal graph (UCCG) over τ. For any $v \in \tau$, SizeMEC($\mathcal{U}^{(v)}$) denotes the number of DAGs in v-rooted sub-class of $\mathcal{U}$. According to Theorem 5, the size of $\mathcal{U}$ can be calculated via the following corollary.

Corollary 9 *Let $\mathcal{U}$ be a UCCG over $\tau = \{v_i\}_{i=1,\cdots,p}$. We have* SizeMEC($\mathcal{U}^{(v_i)}$) > 1 *for* $i = 1, \cdots, p$ *and*

$$\text{SizeMEC}(\mathcal{U}) = \sum_{i=1}^{p} \text{SizeMEC}(\mathcal{U}^{(v_i)}). \tag{2}$$

This corollary shows that the size of Markov equivalence class represented by $\mathcal{U}$ can be calculated via the sizes of smaller sub-classes represented by $\{\mathcal{U}^{(v_i)}\}_{i=1,\cdots,p}$. The following example illustrates how to calculate the size of $\mathcal{U}$ in Figure 2.

Example 3. Consider again the undirected chordal graph $\mathcal{U}$ in Figure 2, SizeMEC($\mathcal{U}$) can be calculated as $\sum_{i=1}^{5}$ SizeMEC($\mathcal{U}^{(v_i)}$) according to Corollary 9. The sizes of the five subclasses represented by $\mathcal{U}^{(v_1)}, \cdots, \mathcal{U}^{(v_5)}$ are $2, 4, 3, 2, 3$ respectively. Therefore, we can get that SizeMEC($\mathcal{U}$) $= 2 + 4 + 3 + 2 + 3 = 14$.

According to Theorem 7, for any $i \in \{1, \cdots, p\}$, the v_i-rooted essential graph $\mathcal{U}^{(v_i)}$ is a chain graph. If $\mathcal{U}^{(v_i)}$ is not directed, each of their isolated undirected subgraphs is a UCCG. Recall that we can calculate the size of a Markov equivalence class through its chain components using Equation (1), similarly, we can calculate the size of v_i-rooted sub-class of $\mathcal{U}$ with its isolated UCCGs as follows.

Corollary 10 *Let $\mathcal{U}^{(v_i)}$ be a v_i-rooted equivalent sub-class of $\mathcal{U}$ defined in Definition 6 and $\{\mathcal{U}_{\tau_j}^{(v_i)}\}_{j=1,\cdots,l}$ be the isolated undirected chordal sub-graphs of $\mathcal{U}^{(v_i)}$ over the vertex set τ_j for $j = 1, \cdots, l$. We have*

$$\text{SizeMEC}(\mathcal{U}^{(v_i)}) = \prod_{j=1}^{l} \text{SizeMEC}(\mathcal{U}_{\tau_j}^{(v_i)}). \tag{3}$$

Since $\{\mathcal{U}_{\tau_j}^{(v_i)}\}_{j=1,\cdots,l}$ are UCCGs according to Theorem 7, SizeMEC($\mathcal{U}_{\tau_j}^{(v_i)}$) can be calculated again via Equation (2) in Corollary 9 recursively. In this iterative approach, Equation (2) and Equation (3) are used alternately to calculate the sizes of equivalence classes represented by an undirected essential graph and a rooted essential graph.

Now in Algorithm 2 we present an enumeration to give *SizeMEC*($\mathcal{U}$). Corollary 11 shows that the enumeration returns the size correctly. For any essential graph $\mathcal{C}$, we can calculate the size of Markov equivalence class represented by $\mathcal{C}$ according to Equation (1) and Algorithm 2.

Algorithm 2: *SizeMEC*($\mathcal{U}$)

Input: $\mathcal{U}$: a UCCG.
Output: the size of Markov equivalence classes represented by $\mathcal{U}$

Let p and n be the numbers of vertices and edges in $\mathcal{U}$;
switch n **do**
  case $p - 1$ **return** p;
  case p **return** $2p$;
  case $p(p-1)/2 - 2$ **return** $(p^2 - p - 4)(p-3)!$;
  case $p(p-1)/2 - 1$ **return** $2(p-1)! - (p-2)!$;
  case $p(p-1)/2$ **return** $p!$;
for $j \leftarrow 1$ **to** p **do**
  $\{\mathcal{U}_1, \cdots, \mathcal{U}_{l_j}\} \leftarrow$ *ChainCom*($\mathcal{U}, v_j$);
  $s_j \leftarrow \prod_{i=1}^{l_j}$ *SizeMEC*($\mathcal{U}_i$)
return $\sum_{i=1}^{p} s_i$

Corollary 11 *Let $\mathcal{U}$ be a UCCG and SizeMEC(·) be the function defined in Algorithm 2. The function SizeMEC($\mathcal{U}$) returns the size of Markov equivalence class represented by $\mathcal{U}$.*

The complexity of calculating SizeMEC($\mathcal{U}$) via Algorithm 2 depends on the number of times this recursive function is called. Our experiments in the next section show that when the number of vertices in $\mathcal{U}$ is small, or when the number is large but $\mathcal{U}$ is sparse, our proposed approach is efficient. However, when $\mathcal{U}$ is large and dense, the proposed approach may be computational infeasible since calculating SizeMEC($\mathcal{U}$) via Algorithm 2 may require a very deep recursion. In the worst case, the time complexity of Algorithm 2 might be $O(p!)$. For example, it might be extremely time-consuming to count SizeMEC($\mathcal{U}$) via Algorithm 2 when U is a UCCG with large p vertices and $p(p-1)/2-3$ edges. Fortunately, according to the experimental results in He et al. (2013a), the undirected and connected chordal subgraphs in sparse essential graphs with hundreds of vertices are mostly small. This implies that our approach may be valuable for size counting in most situations of causal learning based on sparse graphical models.

In the next section, we demonstrate our approach experimentally and explore the size and edge distributions of Markov equivalence classes in sparse graphical models.

4. Experimental Results

We conduct experiments to evaluate the proposed size counting algorithms in Section 4.1, and then to study sizes of Markov equivalence classes in Section 4.2. The main contributions of these experiments are as follows.

1. Our proposed approach can calculate the size of classes represented by a UCCG with a few vertices ($p < 15$) in seconds on a laptop of 2.7GHz and 8G RAM. When the number of vertices is large, our approach is also efficient for the graphs with a sparsity constraint.

2. For the essential graphs with a sparsity constraint, the sizes of the corresponding Markov equivalence classes are nearly exponential in p. This explains the result in Chickering (2002) that causal structure search in the space of Markov equivalence class models could be substantially more efficient than the search in the space of DAG models for learning sparse graphical models.

3. In the set of all Markov equivalence classes of DAGs with p vertices, most graphs are half-completed (nearly $p^2/4$ edges exist) and the Markov equivalent classes represented by these graphs have small average sizes. This is the reason why all Markov equivalence classes have small average sizes (approximately 3.7 reported by Gillispie and Perlman (2002)) even though sparse Markov equivalence classes are huge.

4.1 Calculating the Size of Classes Represented by UCCGs

In this section, we experimentally study the time complexity of our proposed counting algorithms for the UCCGs with a small p or with a large p but also with a sparsity constraint. All experiments are run on a laptop with Intel 2.7GHz and 8G RAM. Note that the chain components are mostly small from sparse Markov equivalence classes with hundreds of vertices (He et al., 2013a). The experimental results show that the proposed method is efficient to count the sizes of sparse Markov equivalence classes with hundreds of vertices.

Let $\mathbb{U}_p^{n*}$ be the set of Markov equivalence classes with p vertices and n edges. The graphs in $\mathbb{U}_p^{n*}$ are sparse if n is a small multiple of p. We generate random choral graphs in $\mathbb{U}_p^{n*}$ as follows. First, we construct a tree by connecting two vertices (one is sampled from the connected vertices and the other from the isolated vertices) sequentially until all p vertices are connected. Then we randomly insert an edge such that the resulting graph is chordal, repeatedly until the number of edges reaches n. Repeating this procedure N times, we obtain N samples from $\mathbb{U}_p^{i*}$ for each $i \in [p-1, n]$.

We first consider the undirected chordal graphs with 5 to 13 vertices. Our experiments on $\mathbb{U}_p^{n*}$ for any $n < p(p-1)/2-3$ show that it is most time-consuming to calculate the size of UCCGs when $n = p(p-1)/2-3$. Based on the samples from $\mathbb{U}_p^{n*}$ where $n = p(p-1)/2-3$, we report in Table 1 the the maximum, the minimum and the average of the sizes of Markov equivalence classes and the time to count them. We see that the size is increasing exponentially in p and the proposed size-counting algorithm is computationally efficient for undirected chordal graphs with a few vertices.

p		5	6	7	8	9	10	11	12	13
Size	Min	14	60	312	1920	1.36e4	1.11e5	1.00e6	1.02e7	1.12e8
	Mean	22	104	658	4508	3.27e4	2.90e5	2.96e6	2.92e7	3.57e8
	Max	30	144	828	5616	4.39e4	3.89e5	3.84e6	4.19e7	4.99e8
Time (sec.)	Min	0	0	1.0e-3	5.0e-3	2.8e-2	1.7e-1	1.3	10.6	95
	Mean	1.3e-4	4.3e-4	1.5e-3	6.8e-3	3.6e-2	2.2e-1	1.6	13.6	140
	Max	1.0e-3	1.0e-3	4.0e-3	1.3e-2	9.6e-2	6.4e-1	5.1	53.5	476

Table 1: The size of Markov equivalence class and the time to calculate it via Algorithm 2 based on 10^5 samples from $\mathbb{U}_p^{n*}$, where p ranges from 5 to 13 and $n = p(p-1)/2-3$ (the worst case for classes with p vertices).

We also study the sets $\mathbb{U}_p^{n*}$ that contain UCCGs with tens of vertices. The number of vertices p is set to $15, 20, \cdots, 100$ and the edge constraint m is set to rp where r is the ratio of m to p. For each p, we consider four ratios: 2, 3, 4 and 5. The undirected chordal graphs in $\mathbb{U}_p^{rp*}$ are sparse since $r \leq 5$. Based on 10^5 samples, we report the average size and time in Table 2. We can see that when $r \leq 4$, the algorithm just takes a few seconds even when the sizes are very huge; when the chordal graphs become denser ($r > 4$), the algorithm takes more time.

Here we have to point out that the choral graphs generated in this experiment might not be uniformly distributed in the space of chordal graphs and that the averages in Table 1 and Table 2 are not accurate estimations of expectations of sizes and time.

4.2 Size and Edge Distributions of Markov Equivalence Classes

In this section, we focus on the size and edge distributions of Markov equivalence classes of directed acyclic graphs. First, we generate a Markov chain on Markov equivalence classes of interest and simultaneously obtain the stationary distribution of the chain according to the methods in He et al. (2013a,b). Then, based on the stationary distribution of the chain, we reweigh the samples from the chain and further use them to calculate the distribution

r	p	15	20	30	40	50	60	70	80	90	100
2	Size	7363	6.98e4	4.74e6	6.94e8	1.9e10	1.2e12	1.2e14	1.5e15	1.8e17	2.6e19
3		3.0e5	3.3e6	1.1e10	7.1e12	4.4e15	8.6e18	1.3e21	6.1e23	1.4e27	9.1e27
4		2.7e6	5.4e8	6.7e12	2.8e16	3.5e19	5.9e22	5.8e25	1.3e29	1.3e38	1.5e34
5		4.9e7	6.7e9	8.3e14	5.4e18	1.1e24	2.8e26	2.3e30	4.8e33	5.6e40	3.8e40
2	Time (sec.)	3.2e-3	5.7e-3	1.2e-2	2.3e-2	0.028	0.037	0.059	0.074	0.090	0.15
3		1.7e-2	3.8e-2	8.8e-2	0.15	0.17	0.27	0.42	0.53	0.75	0.86
4		0.19	0.43	0.72	1.37	1.51	2.16	3.35	3.64	6.14	9.03
5		2.89	7.07	7.91	17.49	50.43	82.99	90.37	95.54	127.25	213

Table 2: The average size of Markov equivalence classes and average counting time via Algorithm 2 are reported based on 10^5 samples from $\mathbb{U}_p^{pr*}$, where p ranges from 15 to 100.

of Markov equivalence classes of interest. In Section 4.2.1, we study the size and edge distributions of Markov equivalence classes with tens of vertices, and in Section 4.2.2, we provide the size distributions of Markov equivalence classes with hundred of vertices under sparsity constraints.

4.2.1 Size and Edge Distribution of Markov Equivalence Classes

In this section, we discuss the distributions of Markov equivalence classes on their sizes and number of edges. We use "size distribution" for the distribution on sizes of Markov equivalence classes, and "edge distribution" for the distribution on the number of edges. First, we consider the number of edges of Markov equivalence classes with p vertices for $10 \leq p < 20$. Then, we focus on the size and edge distribution of Markov equivalence classes with 20 vertices. Finally, we explore the size distributions of Markov equivalence classes with different numbers of edges to show how size distributions change with increasing numbers of edges.

The numbers of edges in the Markov equivalence classes with p vertices range from 0 to $p(p-1)/2$. Based on a Markov chain with length of 10^6 for each p, we display in Table 3 the modes and 99% intervals of edge distributions of Markov equivalence classes with p vertices for $10 \leq p < 20$. The mode is the number that appears with the maximum probability, 99% interval is the shortest interval that covers more than 99% of Markov equivalence classes. The ratios that measure the fraction of 99% interval to $p(p-1)/2+1$ are also given. For example, consider the edge distribution of Markov equivalence classes with 10 vertices; we see that 99% of Markov equivalence classes have between 17 and 32 edges. The ratio is $16/46 \approx 0.348$, where the number 16 is the length of the 99% interval $[17, 32]$ and 46 is the length of the edge distribution's support $[0, 45]$. From the 99% intervals and the corresponding ratios, we see that the numbers of edges of Markov equivalence classes are sharply distributed around $p^2/4$, and these distributions become sharper with increasing of p. This result is reasonable since the number of skeletons of essential graphs with k edges is $\binom{p(p-1)/2}{k}$, and the k-combination reaches maximum around $k = p^2/4$.

In Figure 3, we display the proportions of Markov equivalence classes with 20 vertices according to their sizes and the number of edges. Two scaled marginal distributions in the

p	mode	99% interval	ratio	p	mode	99% interval	ratio
10	25	[17,32]	0.348	15	56	[44,68]	0.236
11	30	[22,39]	0.321	16	64	[51,77]	0.223
12	36	[26,45]	0.299	17	73	[59,87]	0.216
13	42	[32,53]	0.278	18	81	[66,96]	0.201
14	49	[38,60]	0.25	19	91	[75,106]	0.180

Table 3: The edge distributions of Markov equivalence classes with p vertices for $10 \leq p < 20$. The mode is the number that appears with the maximum probability, the 99% interval covers more than 99% of Markov equivalence classes, ratio is the fraction of the length of the 99% interval to the length of the support of edge distribution.

planes are also shown. The black dashed line is the size distribution and the black solid line is the edge distribution of Markov equivalence classes. According to the marginal size distribution, we see that most of the Markov equivalence classes with 20 vertices have small sizes. For example, 26.89% of Markov equivalence classes are of size one; the proportion of Markov equivalence classes with size ≤ 10 is greater than 95%. We also see that the marginal edge distribution of Markov equivalences is concentrated around $100(= 20^2/4)$. The proportion of Markov equivalence classes with 20 vertices and 100 edges is nearly 6%.

To study how the size distribution changes with the number of edges, we consider Markov equivalence classes with 100 vertices and n edges for different n.

Figure 4 displays the size distribution of Markov equivalence classes with 100 vertices and n edges for $n = 10, 50, 100, 200, 400, 600, 1000, 1500, 2000$ and 2500, respectively. We see that the sizes of Markov equivalence classes are very small when the number of edges is close to $p^2/4 = 2500$. For example, when $n \in (1000, 2500)$, the median of the sizes is no more than 4. These results shed light on why the Markov equivalence classes have a small average size (approximately 3.7 reported by Gillispie and Perlman (2002)).

4.2.2 Size Distributions of Markov Equivalence Classes with Sparsity Constraints

We study Markov equivalence classes with p vertices and n vertices. The number of vertices p is set to $100, 200, 500$ or 1000 and the maximum number of edges n is set to rp where r is the ratio of n to p. For each p, we consider four ratios: 1.2, 1.5, 3 and 5. The essential graphs with p vertices and rp edges are sparse since $r \leq 5$. In each simulation, given p and r, a Markov chain with length of 10^6 Markov equivalence classes is generated.

There are sixteen distributions, each of which is calculated with 10^6 essential graphs. We plot the four distributions for $r = 1.2$ in the main window, and the other 12 distributions for $r = 1.5, 3, 5$ in three sub-windows, respectively. In each distribution, the 95% quantiles and 99% quantiles are marked with diamonds and circles, respectively. We see that the sizes of equivalence classes are extremely large. The medians of size distributions are connected by a dashed line in Figure 5. It seems that there is a linear relationship between the logarithm of size and the number of vertices p. In other words, the size grows exponentially with p.

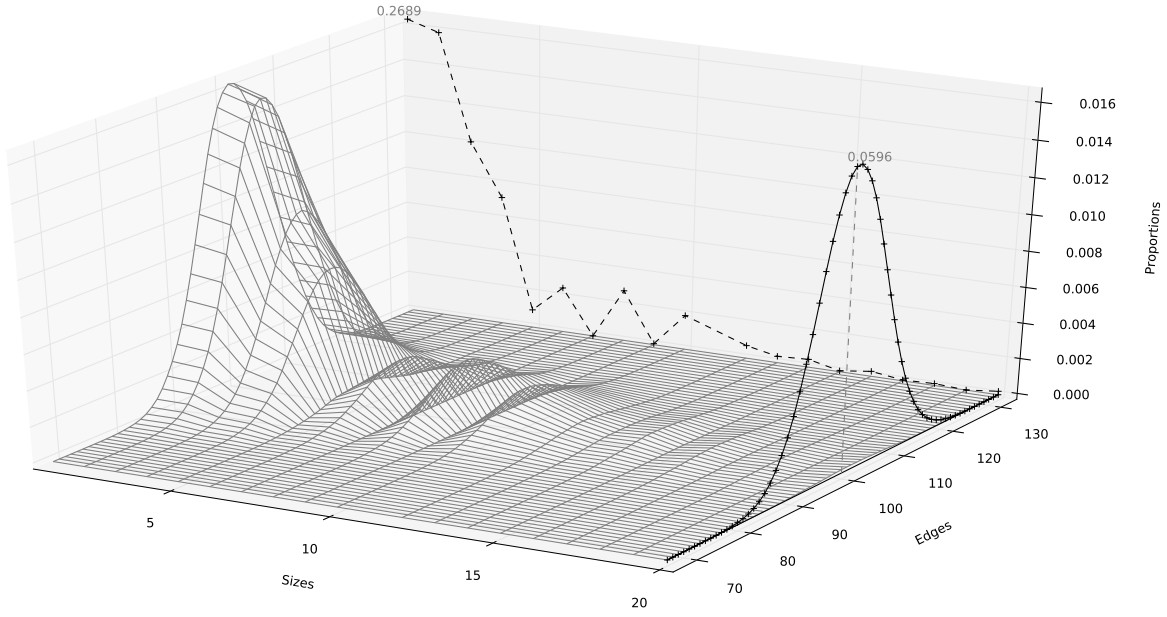


Figure 3: The surface displays the distribution of the Markov equivalence classes with 20 vertices. Two rescaled marginal distributions are shown in the planes. The black dashed line is the size distribution and the black solid line is the edge distribution of Markov equivalence classes.

These results suggest that, to learn directed graphical models, a searching among Markov equivalence classes might be more efficient than that among DAGs since the number of Markov equivalence classes is much less than the number of DAGs when the graphs of interest are sparse.

5. Conclusions and Discussions

In this paper, we propose a method to calculate the sizes of Markov equivalence classes. A rooted sub-class of a Markov equivalence class is introduced and the graph representation of this sub-class, called rooted essential graph, is proposed. We can partition a Markov equivalence class into smaller rooted sub-classes recursively until the sizes of all sub-classes can be obtained via five closed-form formulas. Then we explore the size and edge distributions of Markov equivalence classes. We study experimentally how size distribution changes with the number of edges and report the size distributions of Markov equivalence classes with hundreds of vertices under sparsity constraints. We find that the essential graphs with around $p^2/4$ edges dominate in the set of all essential graphs with p vertices and the corresponding Markov equivalence classes have small sizes. This results in a small average size of all Markov equivalence classes with p vertices. For the sparse essential graphs with p vertices, we find that the sizes of the corresponding Markov equivalence classes are super-exponential in p.

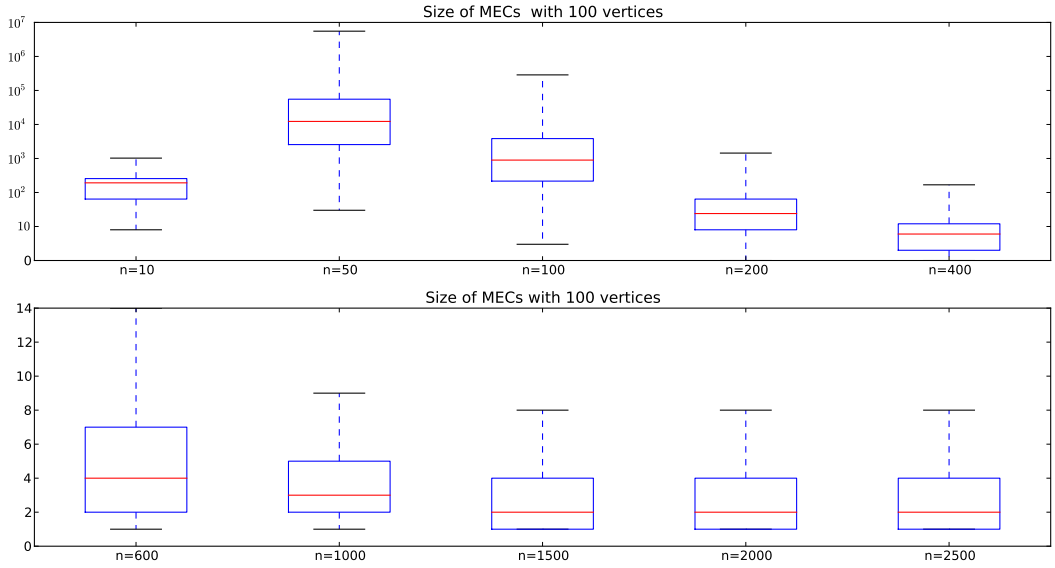


Figure 4: The size distributions of Markov equivalence classes with p vertices and n edges, where $n = 10, 50, 100, 200, 400, 600, 1000, 1500, 2000$ and 2500, respectively.

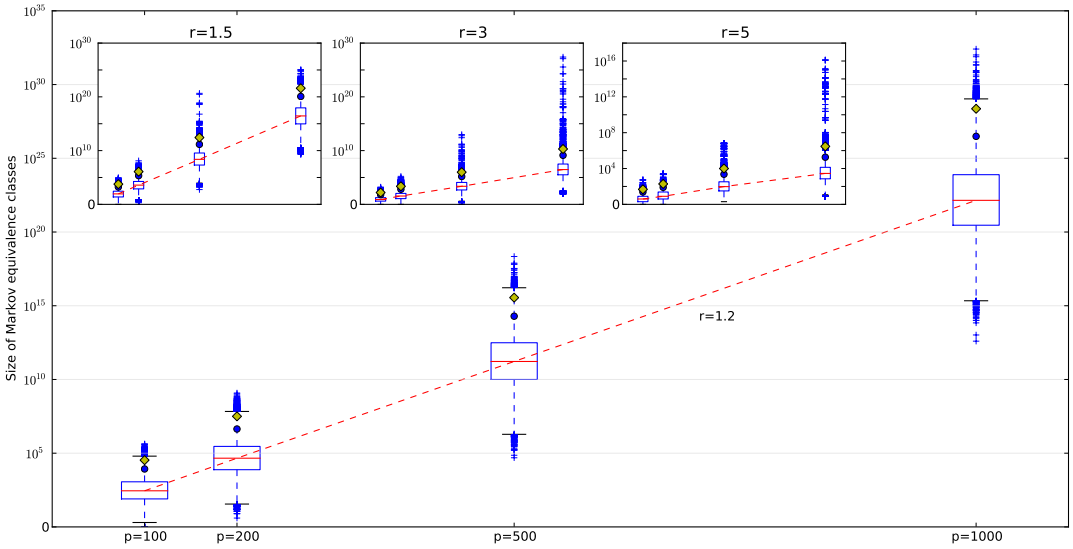


Figure 5: Size distributions of Markov equivalence classes with p vertices and at most rp edges. The lines in the boxes and the two circles above the boxes indicate the medians, the 95th, and the 99th percentiles respectively.

To calculate the sizes of Markov equivalence classes, we provide five closed-form formulas for UCCGs with p vertices and $n = p-1$, p, $p(p-1)/2-2$, $p(p-1)/2-1$, and $p(p-1)/2$ edges respectively. As shown in Example 1, for other cases, say $n = p(p-1)/2-3$, the

size of a Markov equivalence class is no longer determined by the number of vertices p; it depends on the structure of the corresponding UCCG and our proposed method might be inefficient when p is large. For these cases, it is of interest to develop more efficient algorithm, or formulas, to calculate the size of a general Markov equivalence class in the future work.

Moreover, we use python to implement algorithms and experiments in this paper and the python package can be found at `pypi.python.org/pypi/MarkovEquClasses`.

Acknowledgments

We are very grateful to Adam Bloniarz for his comments that significantly improved the presentation of our manuscript. We also thank the editors and the reviewers for their helpful comments and suggestions. This work was supported partially by NSFC (11101008, 11101005, 71271211), DPHEC-20110001120113, US NSF grants DMS-1107000, CDS&E-MSS 1228246, ARO grant W911NF-11-1-0114, AFOSR Grant FA 9550-14-0016, and the Center for Science of Information (CSoI, a US NSF Science and Technology Center) under grant agreement CCF-0939370.

Appendix A. Proofs of Results

In this section, we provide the proofs of the main results of our paper. We place the proof of Theorem 3 in the end of Appendix because this proof will use the results in Algorithm 1 and Corollary 9.

Proof of Theorem 5:

We first show that τ_i-rooted sub-class is not empty. For any vertex $\tau_i \in \tau$, we just need to construct a DAG $\mathcal{D}$ in which no v-structures occurs and all edges adjacent to v are oriented out of v. The maximum cardinality search algorithm introduced by Tarjan and Yannakakis (1984) can be used to construct $\mathcal{D}$. Let p be the number of vertices in $\mathcal{U}$, the algorithm labels the vertices from p to 1 in decreasing order. We first label τ_i with p. As the next vertex to label, select an unlabeled vertex adjacent to the largest number of previously labeled vertices. We can obtain a directed acyclic graph $\mathcal{D}$ by orienting the undirected edges of $\mathcal{U}$ from higher number to lower number. Tarjan and Yannakakis (1984) show that no v-structures occur in $\mathcal{D}$ if $\mathcal{U}$ is chordal. Hence in $\mathcal{D}$, there is no v-structure and all edges adjacent to v are oriented out of v. We have that $\mathcal{D}$ is a τ_i-rooted equivalent DAG of $\mathcal{U}$, thus τ_i-rooted sub-class is not empty.

To prove that the p sub-classes, τ_i-rooted sub-classes for $i = 1, \cdots, p$, form a disjoint partition of Markov equivalence class represented by $\mathcal{U}$, we just need to show that every equivalent DAG of $\mathcal{U}$ is in only one of p sub-classes.

For any equivalent DAG of $\mathcal{U}$, denoted by $\mathcal{D}$, since $\mathcal{D}$ is a directed acyclic graph, there exists an order of its vertices such that all edges are oriented from the preceding vertices to their succeeding ones. Denoted by τ_i the first vertex of this order, we have that all edges adjacent to τ_i are oriented out of τ_i. Clearly, $\mathcal{D}$ is in the τ_i-rooted sub-class.

Below, we show that $\mathcal{D}$ is not in any other rooted sub-class. Suppose that $\mathcal{D}$ is also in another τ_j-rooted sub-class ($i \neq j$). Clearly, τ_i and τ_j are not adjacent. Since $\mathcal{U}$ is connected, we can find a shortest path $\mathcal{L} = \{\tau_i - \tau_k - \cdots - \tau_l - \tau_j\}$ from τ_i to τ_j with more than one edge. The DAG $\mathcal{D}$ is in both τ_i-rooted and τ_j-rooted sub-classes, so we have that $v_i \to v_k$ and $v_j \to v_l$ are in $\mathcal{D}$. Considering all vertices in $\mathcal{L}$, there must be a head to head like $\cdot \to \cdot \leftarrow \cdot$ in $\mathcal{D}$, and the two heads are not adjacent in $\mathcal{D}$ since $\mathcal{L}$ is shortest path. Consequently, a v-structure appears in $\mathcal{D}$. This is a contradiction because $\mathcal{U}$ is an undirected chordal graph and $\mathcal{D}$ must be a DAG without v-structures. ■

Proof of Theorem 7:

Consider the proof of Theorem 6 in He and Geng (2008), we set the intervention variable to be v. If v is a root, Theorem 7 becomes a special case of Theorem 6 in He and Geng (2008). ■

Proof of Corollary 9:

Theorem 5 shows that for any $i \in \{1, 2, \cdots, p\}$, the τ_i-rooted sub-class of $\mathcal{U}$ is not empty and these p sub-classes form a disjoint partition of Markov equivalence class represented by $\mathcal{U}$. This implies Corollary 9 directly. ■

Proof of Corollary 10:

Since $\{\mathcal{U}_{\tau_j}^{(v_i)}\}_{j=1,\cdots,l}$ are l isolated undirected chordal sub-graphs of $\mathcal{U}^{(v_i)}$, the orientations of the undirected edges in a component is irrelevant to the other undirected components. This results in Equation (3) follow directly. ■

Proof of *Theorem* 8:

We first introduce the following Algorithm 3 that can construct a rooted essential graph.

Algorithm 3: Find the v-rooted essential graph of $\mathcal{U}$

Input: $\mathcal{U}$: an undirected and connected chordal graph; v: a vertex of $\mathcal{U}$.
Output: the v-rooted essential graph of $\mathcal{U}$

```
Set H = U;
for each edge · − v in U do
    Orient · − v to · ← v in H.
repeat
    for each edge y − z in H do
        Rule 1: if there exists · → y − z in H, and · and z are not adjacent then
            Orient y − z to y → z in H
        Rule 2: if there exists y → · → z in H then
            Orient y − z to y → z in H
until no more undirected edges in H can be oriented;
return H
```

A similar version of Algorithm 3 is used in He and Geng (2008) to construct an essential graph given some directed edges. From the proof of Theorem 6 in He and Geng (2008), we have that the output of Algorithm 3, H, is the v-rooted essential graph of $\mathcal{U}$, that is, $H = \mathcal{U}^{(v)}$. Moreover, according to Theorem 7, we have that a v-rooted essential graph is a

chain graph and its isolated undirected subgraphs are chain components. From Algorithm 1, we know that $\mathcal{O}$ contains all isolated undirected subgraphs of $\mathcal{G}$.

To prove that the output $\mathcal{G}$ of Algorithm 1 is the v-rooted essential graph $\mathcal{U}^{(v)}$ of $\mathcal{U}$ and $\mathcal{O}$ is the set of all chain components of $\mathcal{U}^{(v)}$, we just need to show that $\mathcal{G} = H$ given the same $\mathcal{U}$ and v.

By comparing Algorithm 1 to Algorithm 3, we find that in Algorithm 1, Rule 1 that is shown in Algorithm 3 is used repeatedly, and in the output $\mathcal{G}$ of Algorithm 1, undirected edges can no longer be oriented by the Rule 1. If we further apply Rule 2 in Algorithm 3 to orient undirected edges in $\mathcal{G}$ until no undirected edges satisfy the condition in Rule 2. Denote the output as $\mathcal{G}'$. Clearly, the output $\mathcal{G}'$ is the same as H obtained from Algorithm 3, that is, $\mathcal{G}' = H$. Therefore, to show $\mathcal{G} = H$, we just need to show $\mathcal{G} = H'$, that is, the condition in Rule 2 does not hold for any undirected edge in $\mathcal{G}$.

In Algorithm 1, we generate a set T in each loop of "while" and the sequence is denoted by $\{T_1, \cdots, T_n\}$. Setting $T_0 = \{v\}$, we have five facts as following

Fact 1 All undirected edges in $\mathcal{G}$ occur in the subgraphs over T_i for $i = 1, \cdots, n$.

Fact 2 All edges in $\mathcal{G}$ between T_i and T_{i+1} are oriented from T_i to T_{i+1} for $i = 0, \cdots, n-1$.

Fact 3 There is no edge between T_i and T_j if $|i - j| > 1$.

Fact 4 There are no v-structures in $\mathcal{G}$.

Fact 5 there is no directed cycle (all edges are directed) in $\mathcal{G}$.

Suppose there exist three vertices x, y and z such that both $y \to x \to z$ and $y - z$ occur in $\mathcal{G}$. Then a contradiction is implied.

Since $y - z$ occurs in $\mathcal{G}$, from **Fact 1**, there exists a set, denoted as T_i containing both y and z. Moreover, $y \to x \to z$ occurs in $\mathcal{G}$, from **Fact 2** and **Fact 3**, we have that $x \in T_i$.

Next we show that x, y and z have the same parents in T_{i-1}. First, y and z have the same parents in T_{i-1}; otherwise $y - z$ will be oriented to a directed edge. Denote by P_1 the same parents of y and z in T_{i-1} and by P_2 the parents of x in T_{i-1}. Second, for any $u \in P_1$, if u is not a parent of x, then $z - x$ in $\mathcal{U}$ will be oriented to $z \to x$ in $\mathcal{G}$ according to Algorithm 1. We have that u is also a parent of x and consequently, $P_1 \subseteq P_2$. Third, For any $u \in P_2$, u must be a parent of y according to **Fact 4**.

We have that $P_2 \subseteq P_1$, and finally $P_2 = P_1$. We get that neither $y \to x$ nor $x \to z$ is oriented by any directed edge $u \to y$ or $u \to x$ with $u \in T_{i-1}$ since $P_2 = P_1$.

Let $u_1 \in T_i$ and $u_1 \to y$ be the directed edge that orients $y - x$ in $\mathcal{U}$ to $y \to x$ in $\mathcal{G}$. Clearly, $u_1 \to y$ occurs in $\mathcal{G}$, and u_1 and x are not adjacent. Since $y - z$ is not directed in $\mathcal{G}$, $u_1 - z$ must occur in $\mathcal{U}$. Moreover, $x \to z$ occurs in $\mathcal{G}$ and u_1 and x are not adjacent, we have that $u_1 - z$ will be oriented to $u_1 \leftarrow z$ in $\mathcal{G}$. Clearly, there occurs a directed cycle $u_1 \to y \to x \to z \to u_1$ in $\mathcal{G}$. This is a contradiction according to **Fact 5**. We have that the condition of Rule 2 does not hold for any undirected edge in $\mathcal{G}$, and consequently, $\mathcal{G} = H$ holds.

■

Proof of Corollary 11:

According to Corollary 9, Corollary 10, and Theorem 8, the output is the size of Markov equivalence class represented by $\mathcal{U}$. ■

Proof of Theorem 3:

Proof of (1):

For a UCCG $\mathcal{U}_{p,n}$, if $n = p-1$, then the graph $\mathcal{U}_{p,n}$ is a tree. For any vertex in $\mathcal{U}_{p,n}$, we have that $\mathcal{U}_{p,n}^{(v)}$ is a DAG according to Algorithm 1. Thus SizeMEC($\mathcal{U}_{p,n}^{(v)}$) = 1. Then, according to Corollary 9, SizeMEC($\mathcal{U}_{p,n}$) = p.

Proof of (2):

For a UCCG $\mathcal{U}_{p,n}$, if $n = p$, then the graph $\mathcal{U}_{p,n}$ has one more edge than tree. Because $\mathcal{U}_{p,n}$ is chordal, a triangle occurs in $\mathcal{U}_{p,n}$. For any vertex v in $\mathcal{U}_{p,n}$, we have that SizeMEC($\mathcal{U}_{p,n}^{(v)}$) = 2. Consequently, we have that SizeMEC($\mathcal{U}$) = $2p$ according to Corollary 9.

Proof of (3):

Let $v_1, \cdots, v_p$ be the p vertices of $\mathcal{U}_{p,n}$. There are only two pairs of vertices that are nonadjacent since $p(p-1)/2-2$ edges appear in $\mathcal{U}_{p,n}$. We first prove that these two pairs have a common vertex. Suppose $v_i - v_j$ and $v_k - v_l$ do not occur in $\mathcal{U}_{p,n}$ and v_i, v_j, v_k, v_l are distinct vertices. Consider the subgraph induced by v_i, v_j, v_k, v_l of $\mathcal{U}_{p,n}$. Clearly, the cycle $v_i - v_k - v_j - v_l - v_i$ occurs in the induced graph and $\mathcal{U}_{p,n}$ is not a chordal graph. We have that the missing two edges in $\mathcal{U}_{p,n}$ are like $v_1 - v_2 - v_3$.

According to Corollary 9, we have that

$$\textit{SizeMEC}(\mathcal{U}_{p,n}) = \sum_{i=1}^{p} \textit{SizeMEC}(\mathcal{U}_{p,n}^{(v_i)}).$$

We first consider $\mathcal{U}_{p,n}^{(v_1)}$. All edges adjacent to v_2 in $\mathcal{U}_{p,n}^{(v_1)}$ are oriented to directed edges whose arrow is v_2 according to Algorithm 1 since v_2 is a neighbor of all neighbors of v_1, and v_1, v_2 are not adjacent in $\mathcal{U}_{p,n}^{(v_1)}$. Removing v_2 from $\mathcal{U}_{p,n}^{(v_1)}$, we have that the induced graph over $v_1, v_3, \cdots, v_p$ is a completed graph. This implies that the induced graph over $v_3, \cdots, v_p$ is an undirected completed graph with $p-2$ vertices. Therefore, we have $\textit{SizeMEC}(\mathcal{U}_{p,n}^{(v_1)}) = (p-2)!$.

Similarly, we can get that $\textit{SizeMEC}(\mathcal{U}_{p,n}^{(v_3)}) = (p-2)!$ since v_1 and v_3 are symmetric in $\mathcal{U}_{p,n}$.

Consider $\mathcal{U}_{p,n}^{(v_2)}$, according to Algorithm 1, for any vertex v_j other than v_1 and v_3, we have that $v_2 \to v_j$, $v_j \to v_1$ and $v_j \to v_3$ occur in $\mathcal{U}_{p,n}^{(v_2)}$, and all other edges in $\mathcal{U}_{p,n}^{(v_2)}$ are undirected. Therefore, there are two isolated chain components in $\mathcal{U}_{p,n}^{(v_2)}$, one contains the edge $x_1 - x_3$ and the other is the subgraph induced by $v_4, \cdots, v_p$. We have the size of first chain component is 2 and the second is $(p-3)!$ since it is a completed graph with $p-3$ vertices. According to Corollary 10, $\textit{SizeMEC}(\mathcal{U}_{p,n}^{(v_2)}) = 2(p-3)!$.

We now consider $\mathcal{U}_{p,n}^{(v_4)}$. According to Algorithm 1, to construct $\mathcal{U}_{p,n}^{(v_4)}$, we first orient the undirected edges adjacent to v_4 in $\mathcal{U}_{p,n}$ to directed edges out of v_4. Since v_4 is adjacent to all other vertices in $\mathcal{U}_{p,n}$, there are no subgraphs like $v_4 \to v_i - v_j$ with v_4 and v_j nonadjacent. This results in the chain component of $\mathcal{U}_{p,n}^{(v_4)}$ being a graph with $p-1$ vertices and $(p-1)(p-2)/2-2$ edges (only $v_1 - v_2 - v_3$ missing). We have that $\textit{SizeMEC}(\mathcal{U}_{p,n}^{(v_4)}) = \textit{SizeMEC}(\mathcal{U}_{p-1,(p-1)(p-2)/2-2})$.

Similarly, we can get that $SizeMEC(\mathcal{U}^{(v_i)}) = SizeMEC(\mathcal{U}_{p-1,(p-1)(p-2)/2-2})$ for any $i \geq 4$ since exchanging the labels of these vertices will not change $\mathcal{U}$.

Therefore, we have proved the following formula

$$SizeMEC(\mathcal{U}_{p,n}) = (p-3)SizeMEC(\mathcal{U}_{p-1,(p-1)(p-2)/2-2}) + 2(p-2)! + 2(p-3)!,$$

Finally, we show that

$$SizeMEC(\mathcal{U}_{p,n}) = (p^2 - p - 4)(p-3)!$$

satisfies the formula and initial condition. First, we have $SizeMEC(\mathcal{U}_{4,4}) = (16-8) * 1 = 8$. Suppose $SizeMEC(\mathcal{U}_{p,n}) = (p^2 - p - 4)(p-3)!$ holds for $p = j-1$,

$$\begin{aligned}
&SizeMEC(\mathcal{U}_{j,j(j-1)/2-2}) \\
&= (j-3)SizeMEC(\mathcal{U}_{j-1,(j-1)(j-2)/2-2}) + 2(j-2)! + 2(j-3)! \\
&= (j-3)\{[(j-1)^2 - (j-1) - 4][(j-1)-3]!\} + 2(j-2)! + 2(j-3)! \\
&= [(j-1)^2 - (j-1) - 4 + 2(j-2) + 2](j-3)! \\
&= (j^2 - j - 4)!(j-3)!
\end{aligned}$$

As a result, $SizeMEC(\mathcal{U}_{p,p(p-1)/2-2}) = (p^2 - p - 4)(p-3)!$ holds for $p = j$.

Proof of (4):

From the condition, only one pair of vertices, denoted by v and u, is not adjacent in $\mathcal{U}_{p,n}$. Consider a v-rooted equivalence sub-class, all undirected edges adjacent to u are oriented to directed edges with arrows pointing to u, and all other edges can be orientated as a completed undirected graph. We have that $\text{SizeMEC}(\mathcal{U}_{p,n}^{(v)}) = (p-2)!$. Similarly, we have that $\text{SizeMEC}(\mathcal{U}_{p,n}^{(u)}) = (p-2)!$. For any vertex w other than v and u, consider any DAG in the w-rooted equivalent sub-class, all edges adjacent to w are oriented away from w, and all other edges form a new chain component with $p-1$ vertices and $(p-1)(p-2)/2-1$ edges. Consider $\text{SizeMEC}(\mathcal{U}_{p,p(p-1)/2-1})$ as a function of p, denoted by $f(p)$. When $p = 3$, we have $f(3) = 3$. Hence we have following formula:

$$f(p) = (p-2)f(p-1) + 2((p-2)!).$$

Now, we show that

$$f(p) = 2(p-1)! - (p-2)!$$

satisfies the formula and initial condition. First, we have $f(3) = 2 * 2 - 1 = 3$. Suppose $f(p) = 2(p-1)! - (p-2)!$ holds for $p = j-1$,

$$\begin{aligned}
f(j) &= (j-2)f(j-1) + 2(j-2)! \\
&= (j-2)(2(j-2)! - (j-3)!) + 2(j-2)! \\
&= 2(j-2)(j-2)! - (j-2)! + 2(j-2)! \\
&= (j-2)!(2j-3) \\
&= 2(j-1)! - (j-2)!
\end{aligned}$$

As a result, $f(p) = 2(p-1)! - (p-2)!$ holds for $p = j$.

Proof of (5):

If $\mathcal{U}$ is an undirected and connected graph with p vertices, and $p(p-1)/2$ edges, then the graph is a complete graph. There are $p!$ DAGs in the Markov equivalence class. ■

References

S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997.

R. Castelo and M. D. Perlman. Learning essential graph Markov models from data. *Studies in Fuzziness and Soft Computing*, 146:255–270, 2004.

D. M. Chickering. Learning equivalence classes of Bayesian-network structures. *The Journal of Machine Learning Research*, 2:445–498, 2002.

M. Finegold and M. Drton. Robust graphical modeling of gene networks using classical and alternative t-distributions. *The Annals of Applied Statistics*, 5(2A):1057–1080, 2011.

N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science Signaling*, 303(5659):799, 2004.

S.B. Gillispie and M.D. Perlman. The size distribution for Markov equivalence classes of acyclic digraph models. *Artificial Intelligence*, 141(1-2):137–155, 2002.

Yangbo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9:2523–2547, 2008.

Yangbo He, Jinzhu Jia, and Bin Yu. Reversible mcmc on markov equivalence classes of sparse directed acyclic graphs. *The Annals of Statistics*, 41(4):1742–1779, 2013a.

Yangbo He, Jinzhu Jia, and Bin Yu. Supplement to "reversible mcmc on markov equivalence classes of sparse directed acyclic graphs". *arXiv preprint arXiv:1303.0632*, 2013b.

D. Heckerman, C. Meek, and G. Cooper. A Bayesian approach to causal discovery. *Computation, Causation, and Discovery*, pages 143–67, 1999.

R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt, and M. Gerstein. A Bayesian networks approach for predicting protein-protein interactions from genomic data. *Science*, 302(5644):449, 2003.

M. H. Maathuis, M. Kalisch, and P. Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 2009. ISSN 0090-5364.

C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410, 1995.

S. Meganck, P. Leray, and B. Manderick. Learning causal bayesian networks from observations and experiments: A decision theoretic approach. In *Modeling Decisions for Artificial Intelligence*, pages 58–69. Springer, 2006.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge Univ Pr, 2000.

M.D. Perlman. Graphical model search via essential graphs. *Contemporary Mathematics*, 287:255–266, 2001.

P. Spirtes, C.N. Glymour, and R. Scheines. *Causation, Prediction, and Search*. The MIT Press, 2001.

R. E. Tarjan and M. Yannakakis. Simple linear-time algorithms to test chordality of graphs, test acyclicity of hypergraphs, and selectively reduce acyclic hypergraphs. *SIAM Journal on Computing*, 13:566, 1984.

T. Verma. A linear-time algorithm for finding a consistent expansion of a partially oriented graph,". Technical report, Technical Report R-180, UCLA Cognitive Systems Laboratory, 1992.

T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, page 270. Elsevier Science Inc., 1990.