Learning from Subsampled Data: Active and Randomized Strategies

Fabian Wauthier

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2013-94
http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-94.html

May 17, 2013

Learning from Subsampled Data:
Active and Randomized Strategies

by

Fabian Lutz-Frank Wauthier

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

and the Designated Emphasis

in

Communication, Computation and Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Chair
Professor Yun S. Song
Professor Dan Klein
Professor Ian Holmes

Spring 2013

Learning from Subsampled Data:
Active and Randomized Strategies

Abstract

Learning from Subsampled Data:
Active and Randomized Strategies

by

Fabian Lutz-Frank Wauthier

Doctor of Philosophy in Electrical Engineering and Computer Sciences

with the Designated Emphasis in

Communication, Computation and Statistics

University of California, Berkeley

Professor Michael I. Jordan, Chair

In modern statistical applications, we are often faced with situations where there is either too little or too much data. Both extremes can be troublesome: Interesting models can only be learnt when sufficient amounts of data are available, yet these models tend to become intractable when data is abundant. An important thread of research addresses these difficulties by subsampling the data prior to learning a model. Subsampling can be active (i.e. active learning) or randomized. While both of these techniques have a long history, a direct application to novel situations is in many cases problematic. This dissertation addresses some of these issues.

We begin with an active learning strategy for spectral clustering when the cost of assessing individual similarities is substantial or prohibitive. We give an active spectral clustering algorithm which iteratively adds similarities based on information gleaned from a partial clustering and which improves over common alternatives.

Next, we consider active learning in Bayesian models. Complex Bayesian models often require an MCMC-based method for inference, which makes a naïve application of common active learning strategies intractable. We propose an approximate active learning method which reuses samples from an existing MCMC chain in order to speed up the computations.

Our third contribution looks at the effects of randomized subsampling on Gaussian process models that make predictions about outliers and rare events. Randomized subsampling risks making outliers even rarer, which, in the context of Gaussian process models, can lead to overfitting. We show that Heavy-tailed stochastic processes can be used to improve robustness of regression and classification estimators to such outliers by selectively shrinking them more strongly in sparse regions than in dense regions.

Finally, we turn to a theoretical evaluation of randomized subsampling for the purpose of inferring rankings of objects. We present two simple algorithms that predict a total order

over n objects from a randomized subsample of binary comparisons. In expectation, the algorithms match an $\Omega(n)$ lower bound on the sample complexity for predicting a permutation with fixed expected Kendall tau distance. Furthermore, we show that given $O(n \log(n))$ samples, one algorithm recovers the true ranking with uniform quality, while the other predicts the ranking more accurately near the top than the bottom. Due to their simple form, the algorithms can be easily extended to online and distributed settings.

Contents

List of Figures

List of Tables

Acknowledgments

This dissertation would not have been possible without the tremendous support of my advisor, mentors, collaborators, as well as my friends and family.

First and foremost I am immensely grateful to my advisor Michael Jordan, whose enduring encouragement and support guided me throughout my years at the University of California, Berkeley. Mike was instrumental in growing and shaping my love for statistical machine learning, and for teaching me to think independently. Complimenting his influence was an outstanding group of Ph.D. students and postdocs at the Statistical Artificial Intelligence Laboratory as well as long-term cube-member Po-Ling Loh, all of whom further influenced my research and education. I was fortunate to be able to do a number of internships with Nebojsa Jojic at MSR, Redmond, which expanded my horizons, taught me to think creatively and led to the research presented in Chapters 2 and 3. My sincere thanks also go to Purnamrita Sarkar, Stefanie Jegelka and Roland Dunbrack for helpful discussions, to Matthias Seeger, Chris Williams, Rich Zemel and Steve Mann for channelling my early research interests and to the many housemates I had for "whole wheat" conversations not really about whole wheat: Kurt Miller, Alexandre Bouchard-Côté, Blaine Nelson, Alex Shyr, Dave Golland, Andre Wibisono, Lester Mackey, Percy Liang, Garvesh Raskutti, Teodor Moldovan, Linda Tran, Janet Hui and Jennifer Hui. Mark Tarses was responsible for a steady stream of free chocolate, cakes and puddings, fuelling many late working nights. I want to especially acknowledge Sam Roweis, whose course on graphical models at the University of Toronto sparked my love for statistical machine learning while I was visiting there. His incredible energy was truly inspirational and his encouragement ultimately led me down this path. He is sorely missed.

Finally, none of this would have been possible without my parents Fritz and Marieta, as well as my siblings Patrick, Bettina and Sylvia. But most importantly, I want to thank my amazing girlfriend, Sophia, for supporting and sustaining me.

Chapter 1

Introduction

In modern machine learning applications, we frequently encounter situations with either too little or too much training data: In many research areas (e.g., human immunology, crowdsourcing, etc.), there is a perennial paucity of data, while in many other disciplines (e.g., genomics, astronomy, meteorology, etc.) there has been an explosion of available data. Both extremes can be troublesome since interesting statistical models can only be learnt from sufficiently large datasets yet these models tend to become intractable when scaled. An important thread of research aims to address these difficulties by *subsampling* the data prior to learning. Subsampling can be divided into two main paradigms, which are suitable in different scenarios.

Active Learning. A statistician can often actively guide the data-collection process. Biological, physical and social experiments, for example, are usually designed with a statistical question in mind. Active learning is a class of methods that try to automate such experimental design in algorithmic form. The goal is to sharply reduce the data acquisition cost (either time or money) by carrying out only a small subset of possible experiments that will be most "useful" for the purpose of learning a model. In this sense, active learning is prototypically used to create or grow a dataset. However, it also applies to compressing a large dataset by selecting a subset of examples. While the core idea of active subsampling is intuitive, its practical implementation leaves considerable room for interpretation. For one, active learning strategies commonly depend crucially on the internals of the statistical model. As the models vary, so do the active learning algorithms. Some key distinctions can be made between active learning for Bayesian and frequentist models, however, even within these classes there is an abundance of competing frameworks. A recent literature review of some of these can be found in Settles [68].

Randomized Subsampling. Active data acquisition is not always feasible. For example, when data is collected from online click-through streams or purchasing preferences, data collection is necessarily passive. In other cases, the computations required for active learning might be too time-consuming or expensive to allow efficient compression of a large dataset

that has already been collected. In cases as these, randomized subsampling can be viewed as a degenerate form of active learning, where the subset of experiments is chosen randomly (and perhaps not even by the statistician). This interpretation is useful both when slowly building up a dataset from incoming data, or when compressing an existing, large dataset. A multitude of different applications have led to a variety of randomized subsampling strategies. Some well-known examples include stratified subsampling, cluster subsampling, and of course simple randomized subsampling. A comprehensive survey can be found in [53]. A main attraction of the randomized framework is that it is much easier to analyze than its active counterpart. In particular, statistical learning theory (e.g. [9]) can in some cases be used to quantify how well a model estimated on a subsample of data will generalize to the complete dataset. For this approach to be successful, the randomized subsample must adequately reflect properties of the test dataset. The simplest tool to ensure this is to sample the data uniformly at random (with replacement). Alternatively, concentration inequalities (e.g. [8]) can often be used to quantify how much an estimator based on a random subsample will deviate from its mean. Again, independent, random sampling considerably simplifies these analyses, though some statements can also be made when the samples are weakly dependent.

1.1 Outline

A direct application of the above subsampling strategies is in many cases problematic. On the one hand, active learning algorithms must usually be tailored to specific models. As new models are being developed there is an ongoing need to develop effective active learning strategies for them. This process necessarily pushes active learning into novel terrain where new computational and statistical problems begin to dominate. On the other hand, randomized subsampling risks throwing away too much data. In cases where we are interested in learning from outliers, extremes or other rare events, a drastic reduction in training set size can make these examples even rarer. Depending on the statistical model, over- and underfitting can then become an issue. Lastly, the size of the random subsample has a significant impact on the final model and guidance on its choice is of considerable interest in practical situations. In this dissertation we demonstrate algorithmic and theoretical progress on these topics. The contributions are as follows:

Active Spectral Clustering. To begin, we focus our attention on a new method for active spectral clustering in Chapter 2. Spectral clustering is a widely used method for organizing data that only relies on pairwise similarity measurements. This makes its application to non-vectorial data straightforward in principle, provided all pairwise similarities are available. However, in recent years, numerous examples have emerged in which the cost of assessing similarities is substantial or prohibitive. We propose an active learning algorithm for spectral clustering that incrementally measures only those similarities that are most likely to remove uncertainty in an intermediate clustering solution. In many applications, similarities are not only costly to compute, but also noisy. We extend our algorithm to maintain

running estimates of the true similarities, as well as estimates of their accuracy. Using this information, the algorithm updates only those estimates which are relatively inaccurate and whose update would most likely remove clustering uncertainty. We compare our methods on several datasets, including a realistic example where similarities are expensive and noisy. The results show a significant improvement in performance compared to the alternatives. This work was published in [85].

Bayesian Bias Mitigation for Crowdsourcing. In many applications, Bayesian models are a valuable tool for learning from small amounts of possibly noisy data. Unfortunately, complex Bayesian models often require MCMC-based inference, which makes active learning impractical. We demonstrate this phenomenon in Chapter 3 where we develop Bayesian Bias Mitigation for Crowdsourcing (BBMC), a statistical model for learning from possibly biased crowdsourced label data. Specifically, we model each labeler in a crowd as being influenced by a set of shared random effects, allowing labelers to give systematically different responses. In doing so, we go beyond current methodologies which model all labels as coming from a single latent truth, corrupted by bias and noise. Our approach can account for more complex bias patterns that arise in ambiguous or hard labeling tasks. Due to the complexity of the model, inference is done using a Gibbs sampler. Unfortunately, active learning is commonly considered infeasible with Gibbs sampling inference. We propose a general approximation strategy for Markov chains to efficiently quantify the effect of a perturbation on the stationary distribution and specialize this approach to active learning. Experiments show BBMC to outperform many common heuristics. This work was published in [88].

Heavy-Tailed Process Priors for Selective Shrinkage. In Chapter 4 we investigate the effects of randomized subsampling on Gaussian process models. We demonstrate a biological application where learning from outliers in the dataset is of significant interest. Unfortunately, when randomly subsampling a dataset, these outliers become even rarer, which in the context of Gaussian processes can lead to overfitting. We show that heavy-tailed stochastic processes (which we construct from Gaussian processes via a copula), can be used to improve robustness of regression and classification estimators to such outliers by selectively shrinking them more strongly in sparse regions than in dense regions. We carry out a theoretical analysis to show that selective shrinkage occurs when the marginals of the heavy-tailed process have sufficiently heavy tails. The analysis is complemented by experiments on biological data which indicate significant improvements of estimates in sparse regions while producing competitive results in dense regions. This work was published in [87].

Efficient Ranking from Pairwise Comparisons. Chapter 5 is concerned with an analysis of a randomized subsampling algorithm. Ranking n objects from pairwise comparisons is a core machine learning problem, arising in recommender systems, ad placement, player ranking, biological applications and many others. In many practical situations the true pairwise comparisons cannot be actively measured, but a subset of all $n(n-1)/2$ comparisons

is passively and noisily observed. We present two simple algorithms that predict a total order over n objects from such data. In expectation, the algorithms are shown to match an $\Omega(n)$ lower bound on the sample complexity for predicting a permutation with fixed expected Kendall tau distance. Furthermore, if instead an average of $O(n \log(n))$ binary comparisons are measured, then one algorithm recovers the true ranking in a uniform sense, while the other predicts the ranking more accurately near the top than the bottom. We extend the algorithms to online and distributed learning settings, with benefits over traditional alternatives. This work was published as [86].

Chapter 2

Active Spectral Clustering

Together with the recent explosion of available data, we are now seeing a much greater variety of data types. Moving away from pure vectorial data, it is now common to treat sequences, trees or even graphs as individual datapoints. Computational biology, natural language processing and computer vision are example sources of such data. To accomodate this every-growing variety of data types, a range of new algorithms are being developed. As these algorithms emerge, we seek new active learning procedures to accomodate them. In this chapter we will focus on *spectral clustering* as a promising algorithm for clustering a variety of data types and will demonstrate an improved active learning algorithm that outperforms previous methodologies.

2.1 Spectral Clustering

Clustering is a fundamental problem involving summarizing, indexing and classifying various types of data. As data sets become larger and more complex, algorithms that depend on pairwise similarities—rather than fixed length feature vector representations—are growing increasingly popular. An important example is spectral clustering, which partitions data through a spectral analysis of the Laplacian matrix induced by the similarity graph.

Although methods based on pairwise similarities are growing in popularity, a practical difficulty is that pairwise similarities can be expensive to acquire, be it due to computational requirements, need for human input, or lack of observability. In protein clustering, for example, a computationally expensive alignment process may be necessary before two proteins can be compared. Other examples in computational biology require a combinatorial search for each pairwise similarity, even when the datapoint can be represented in a compact form (a protein sequence is usually less than 1000 letters long). A counterpart to these computational issues is that often the only practical way to obtain similarities is to query human annotators. Here, measurements are not only expensive, but frequently also noisy. In this paper we consider the task of organizing a stream of snapshots taken by a wearable camera at a rate of about one photo per 20 seconds [49]. Clustering these snapshots is beyond the

capabilities of existing computer vision algorithms, so human guidance is necessary to either cluster the images, or learn improved models to do the clustering for us. As the human subject (e.g., an Alzheimer's patient), wears the camera during an open-ended observation period, it is not clear a priori what the clusters should be. A way to tackle this problem is to ask human annotators to rate how similar any two photos are and then to cluster using this data. (For example, photos may be deemed similar if they were taken in similar locations.) This is one of many examples where crowdsourcing, albeit expensive, can be used to collect pairwise similarity measurements. As a final example, in some situations the objects that are being compared can disappear over time, making retrospective comparisons difficult. Consider viral strains, which, if not preserved in a laboratory, may disappear, leaving behind only indirect assessments with other viruses. While similarities among preserved viruses can be acquired at a relatively high cost in the lab (e.g., crossreactivity of the immune responses), similarities involving the extinct strain are not available at any cost. In addition to this time barrier, geographic, legal and policy barriers can also make certain pairwise similarities inaccessible. All these examples illustrate that similarities may be noisy and arbitrarily expensive to compute, to the extreme where certain similarities are completely unavailable.

The potentially significant cost of obtaining pairwise similarities motivates the search for tradeoffs between desired clustering quality and the required amount of data. In this paper, we study this question in the setting of the spectral clustering of n objects when we do not have access to all pairwise measurements initially, but we can iteratively query the similarities from an external black-box procedure (which may impose restrictions on which of the subset of all $n(n-1)/2$ similarities can be queried). In this active learning formulation the goal is to find a good approximation to the true clustering based on as few similarity evaluations as possible, thus reducing the overall cost (i.e. compute cycles, human tasks performed, laboratory material and other data collection expenses).

A recent contribution in this direction has been made by Shamir and Tishby [69]. Their approach was designed for querying arbitrary similarity matrices, including those where no clustering structure is apparent. This paper significantly improves on their method by exploiting the fact that most realistic pairwise similarity matrices, even if incomplete, do exhibit clustering structure. Consider Figure 2.1, where on the left we show an incomplete matrix that we might realistically encounter and on the right a matrix constructed from random similarities. Shamir and Tishby assume no structure in the matrix (such as in the matrix on the right) and query measurements that maximally change the overall clustering solution. In contrast, we tailor our active learning algorithm to work well with *realistic* matrices (such as the one on the left) where information about the true clustering emerges quickly and can be leveraged to guide an improved query selection strategy.

An aspect of spectral clustering that is commonly ignored is that similarity measurements are generally noisy. Active spectral clustering methods have so far not taken these uncertainties into account. We extend our algorithm in this direction to allow repeat measurements of noisy similarities, and use those to compute running estimates of the true similarities, as well as estimates of their accuracy. We extend our algorithm to measure that similarity which is inaccurate and whose update would most likely remove clustering uncertainty.

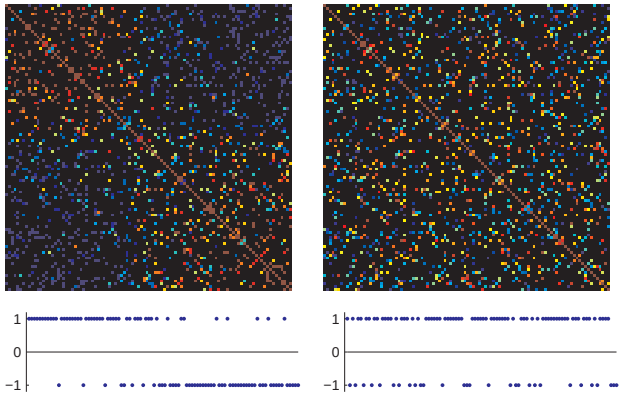


Figure 2.1: Two incomplete similarity matrices. The left was constructed from median image similarities, measured using Amazon Mechanical Turk (see Section 2.6). We permuted the rows and columns so that any clusters would become visible as diagonal blocks. Then we set 82% of the similarities to 0 (black regions). The right matrix was constructed similarly, but starting from similarities sampled uniformly in $[0, 1]$. The left matrix still shows clustering structure, which is also visible in the sign vector of the second Laplacian eigenvector, plotted below (see Sections 2.3 and 2.4 for details). The right matrix had almost no structure to start with, so no structure is visible in the corresponding eigenvector.

The paper is organized as follows. In Section 2.2 we review related research. Section 2.3 presents some background on spectral clustering. Next, we describe our active learning algorithm in Section 2.4. In Section 2.5 we extend the algorithm to take measurement noise into account. We present experiments on synthetic and real datasets in Section 2.6 and conclude with final remarks in Section 2.7.

2.2 Related Research

Spectral Clustering Spectral clustering was popularized in the machine learning community by Shi and Malik [72] and shortly afterwards revisited by Ng, Jordan and Weiss [58]. Since then, it has permeated the literature and become a firm part of the practitioner's toolbox. Over the years, numerous connections to other research fields have been made, a comprehensive review of which can be found in [82]. However, very little work in the spectral clustering literature has explicitly investigated situations when only a subset of all similarities is known, possibly contaminated by noise.

Matrix Completion One simple approach to adapting spectral clustering to the case of missing similarities is to exploit the burgeoning literature on matrix completion methods. A particularly straightforward approach is to impute a constant value (e.g., zero) for the

missing similarities. Such approaches are often used in practice, and some of its properties have been analyzed theoretically [69].

More sophisticated methods can be deployed under the assumption that entire rows or columns of the similarity matrix can be measured. The driving assumption behind these methods is that the true matrix is of low rank, so that a small subset of elements approximately captures the global matrix structure. Among these approaches, the Nyström method [27, 91] is perhaps the most well known. Other algorithms that use row/column sampling include for instance [25, 26, 32]. Applications of some of these methods to spectral clustering are given in [29, 66]. In less controlled situations, we do not have access to entire rows/columns of measurements, but only an arbitrary subset; in this setting it is still possible to exploit a low-rank assumption for matrix completion [1, 80]. There is debate about the value of these methods in the spectral clustering setting; in particular, Shamir and Tishby [69] argued that the low rank assumptions can be unrealistic in many spectral clustering applications and demonstrated that the Nyström method often performs poorly. Additionally, most of the row/column sampling methods require that the sampling distribution depends on the entire similarity matrix [25, 26, 27, 32]. Because this matrix is unknown, these methods are of limited applicability in our setting. Finally, we highlight that with the exception of Huang et al. [43], relatively little work has been done on analyzing the influence of incomplete or perturbed similarity matrices on the spectral clustering solution.

Active Learning Until recently, the study of active learning for spectral clustering was restricted to settings where the entire similarity matrix is known (perhaps approximately) and an external oracle can be repeatedly queried for additional *linkage constraints* between objects (of the form *must-link* or *cannot-link*). When the similarity matrix only approximately captures the desired clustering, adding such constraints iteratively can help resolve ambiguous boundary cases. Relevant examples include [7, 54, 84, 92]. We note in particular the work of Xu et al. [92], in which the constraints are absorbed by modifying the similarity matrix in a way that is akin to measuring similarities of higher quality. The focus in active learning for spectral clustering has only recently shifted to scenarios in which explicit costs are imposed on the measurement of similarities. This focus is exemplified by Shamir and Tishby [69], who propose and analyze an active learning method based on matrix perturbation theory [78].

2.3 Spectral Clustering

We begin by presenting our notation and summarizing the key ideas of spectral clustering. For further details we refer the reader to von Luxburg [82]. Given n objects, denote by W the $n \times n$ symmetric matrix of pairwise similarities among these objects. Typically, $0 \leq w_{ij} \leq 1, i, j = 1, \ldots, n$ and $w_{ii} = 1, i = 1, \ldots, n$. Let $D = \mathrm{diag}(W\mathbf{1})$ be the diagonal matrix of row sums of W. With this notation, the *unnormalized Laplacian matrix* induced by W is then given by $L = D - W$. Spectral clustering partitions the n objects into two

groups by thresholding the second eigenvector v_2 of L. Specifically, if we let the partition be encoded by variables $c_i \in \{-1, +1\}, i = 1, \ldots, n$, then

$$c_i = 2\,[v_2(i) > 0] - 1. \tag{2.1}$$

Here, we use the notation $v_2(i)$ to indicate the i^{th} component of the vector v_2. Because the partitioning is trivial to compute from the second eigenvector, we will occasionally refer to the eigenvector itself as the spectral clustering solution, rather than the partitioning.

Spectral clustering only sees the data as filtered through the matrix W. Thus it is possible to adapt the spectral approach to the clustering of non-vectorial data such as graphs, sequences and sets; it suffices that similarity scores can be computed for these objects. This is generally accomplished via a kernel function, and computationally efficient kernels are available for certain kinds of structured objects [34, 70]. Unfortunately, however, kernel formulations are often too rigid to be adapted to specific needs, and often lack interpretability. As we move to more complex datasets, the notion of similarity a practitioner is interested in may not be captured by a kernel. Indeed, as highlighted in the Introduction, in many practical examples the similarities cannot be evaluated by a computer at all, but must be provided by an experiment or human annotator.

2.4 Active Learning

In this section we propose an active learning strategy that attempts to alleviate the above issues. Our work is based on a matrix perturbation argument for an intermediate estimate of the Laplacian matrix. Given incomplete measurements, we estimate the true Laplacian matrix as

$$\hat{L} = \hat{D} - \hat{W}, \tag{2.2}$$

where $\hat{W}$ is the matrix of all pairwise measurements with zero imputed for unknown entries, and $\hat{D} = \text{diag}(\hat{W}\mathbf{1})$. The motivation for imputation with zero can be seen by rewriting the spectral clustering problem. The second eigenvector of the Laplacian $\hat{L}$ can be found as

$$\hat{v}_2 = \text{argmin}_v v^\top \hat{L} v = \text{argmin}_v \sum_{ij} \hat{w}_{ij}(v(i) - v(j))^2 \tag{2.3}$$

$$\text{s.t.} \quad v^\top v = 1 \tag{2.4}$$

$$v^\top \mathbf{1} = 0. \tag{2.5}$$

Thus, similarities act as weights on soft constraints between eigenvector components. By imputing zero for missing similarities we merely ignore those constraints which are not supported by a measurement.

For any set of similarities $\hat{W}$, the second eigenvector $\hat{v}_2$ gives the best guess for an embedding of the objects on the line. The embedding is such that two groups of similar

objects are embedded away from zero, on the negative or positive orthant, respectively. Any objects that are approximately equally similar to all remaining objects are embedded near zero. This is intuitive, for these are the objects that cannot clearly be assigned to either of the two groups. Indeed, since a mean can be found by minimizing a mean squared error, we see from Eq. (2.3) that an object i with approximately constant similarities $\hat{w}_{ij}$ to remaining objects j should be embedded near the average of their embedding locations $\hat{v}_2(j)$. On the other hand, if the data actually clusters well and is reasonably balanced, then we expect the second eigenvector v_2 of the true Laplacian L to have elements with magnitude on the order of $1/\sqrt{n}$, since $v_2^\top v_2 = 1$. In this way, the elements of most realistic embeddings v_2 should be expected to be bounded away from zero. Spectral clustering based on incomplete data $\hat{W}$ partitions the objects by looking at the signs of the embedding $\hat{v}_2$ (Eq. (2.1), with threshold at zero). Consequently, objects which are embedded near zero are the objects about whose cluster label we should be most "uncertain" about.

In many practical cases, a relatively small amount of data suffices so that $\hat{v}_2$ already indicates a useful clustering. The left sign vector shown in Figure 2.1 demonstrates this on a real dataset. We use such partial information as a guide towards measurements that more quickly reveal the true nature of the clustering. In this approach, our earlier intuition about the magnitude of $\hat{v}_2$ components plays a crucial role. More specifically, our active learning strategy uses matrix perturbation theory to reveal that entry of $\hat{W}$ for which a constant perturbation would change the *minimum magnitude element* of $\hat{v}_2$ the most. The rationale is that by focussing on small magnitude components, we more quickly move them away from the cluster boundary (i.e. 0), and thus reduce uncertainty in the partial clustering. In effect, we try to choose measurements that help us push the embedding clusters further apart. If the data actually clusters well, this should quickly guide us to the clean clustering we expect to find.

Suppose we have the Laplacian eigenvector decomposition $\hat{L} = \sum_{p=1}^{n} \hat{\lambda}_p \hat{v}_p \hat{v}_p^\top$ and that $\hat{\lambda}_1 \leq \hat{\lambda}_2 \ldots \leq \hat{\lambda}_n$. For spectral clustering, $\hat{\lambda}_1 = 0$ and $v_1 = \mathbf{1}/\sqrt{n}$. Matrix perturbation theory (e.g. Stewart and Sun [78], Chapter V, Section 2.3) gives the first order change of the second eigenvector as

$$\frac{d\hat{v}_2}{d\hat{w}_{ij}} = \sum_{p>2}^{n} \frac{\hat{v}_2^\top \left[\partial \hat{L}/\partial \hat{w}_{ij}\right] \hat{v}_p}{\hat{\lambda}_2 - \hat{\lambda}_p} \hat{v}_p, \tag{2.6}$$

provided $\hat{\lambda}_2$ has multiplicity 1. Note that $\partial \hat{L}/\partial \hat{w}_{ij} = (e_i - e_j)(e_i - e_j)^\top$, where e_i is the indicator vector of i. If $k_{min} = \text{argmin}_k |\hat{v}_2(k)|$, the change to the smallest magnitude element of $\hat{v}_2$ is $d\hat{v}_2(k_{min})/d\hat{w}_{ij}$. Our proposed active learning algorithm, IU-RED, is given in Algorithm 1.

A recent algorithm due to Shamir and Tishby [69] has a similar structure. The main steps are shown in Algorithm 2, which we refer to as S&T throughout.[1] The algorithm chooses measurements that maximize the *global* change to $\hat{v}_2$ by maximizing the norm on line 2.

[1]The full algorithm requires a "budget" parameter b which specifies the maximum number of measure-

Algorithm 1: IU-RED

$S = \{(i,j) : i,j \in \{1,\ldots,n\}, i < j\}$
$\hat{W} = I$
for $t = 1,\ldots,n(n-1)/2$
  $\hat{L} = \text{diag}(\hat{W}\mathbf{1}) - \hat{W} = \sum_{p=1}^{n} \hat{\lambda}_p \hat{v}_p \hat{v}_p^\top$
  $k_{min} = \text{argmin}_k |\hat{v}_2(k)|$ (1)
  $(i^*, j^*) = \text{argmax}_{(i,j)\in S} \left| \frac{d\hat{v}_2(k_{min})}{d\hat{w}_{ij}} \right|$ (2)
  $= \text{argmax}_{(i,j)\in S} \left| \sum_{p>2}^{n} \frac{\hat{v}_2^\top [\partial \hat{L}/\partial \hat{w}_{ij}] \hat{v}_p}{\hat{\lambda}_2 - \hat{\lambda}_p} \hat{v}_p(k_{min}) \right|$ (3)
  $S = S \setminus \{(i^*, j^*)\}$
  $\hat{w}_{i^*j^*} = w_{i^*j^*}, \hat{w}_{j^*i^*} = w_{j^*i^*}$
return Second eigenvector of $\hat{L} = \text{diag}(\hat{W}\mathbf{1}) - \hat{W}$

The reasoning is the following: As more measurements are acquired, the estimate $\hat{v}_2$ will necessarily converge to the true eigenvector v_2, regardless of the query ordering, since only a finite number of measurements can be made. Since constant perturbations to elements of $\hat{W}$ can have varying effects on $\hat{v}_2$, we should choose to update that element where the effect is largest.

If the similarity matrices were random, we would not expect to see partial clusterings emerge in $\hat{v}_2$, even with fairly large amounts of data. Figure 2.1 has highlighted this. In this unstructured setting, Shamir and Tishby's method may well be the best we can do, for it targets the global change in the clustering solution. Our algorithm exploits that practical similarity matrices are highly structured even when severely subsampled and uses this partial information as a guide for query selection. Our experiments emphasize that the empirical improvements of our method are significant, even though the algorithmic differences may at first appear minor.

2.5 Measurement Noise

In many settings, only noisy similarities can be measured. In the Section 2.6, for example, we consider a crowdsourcing application where similarities are manually assigned by human labelers. A significant factor there is that even cooperative workers may disagree on similarity scores. Noisy similarities can be a fundamental problem, yet their impact on spectral clustering has not been thoroughly understood. Huang et al. [43] are among the few to

ments that can be requested from an oracle. For this paper, we set $b = n(n-1)/2$. Another version of their algorithm interleaves active selection with random selection, which we consider in our experiments.

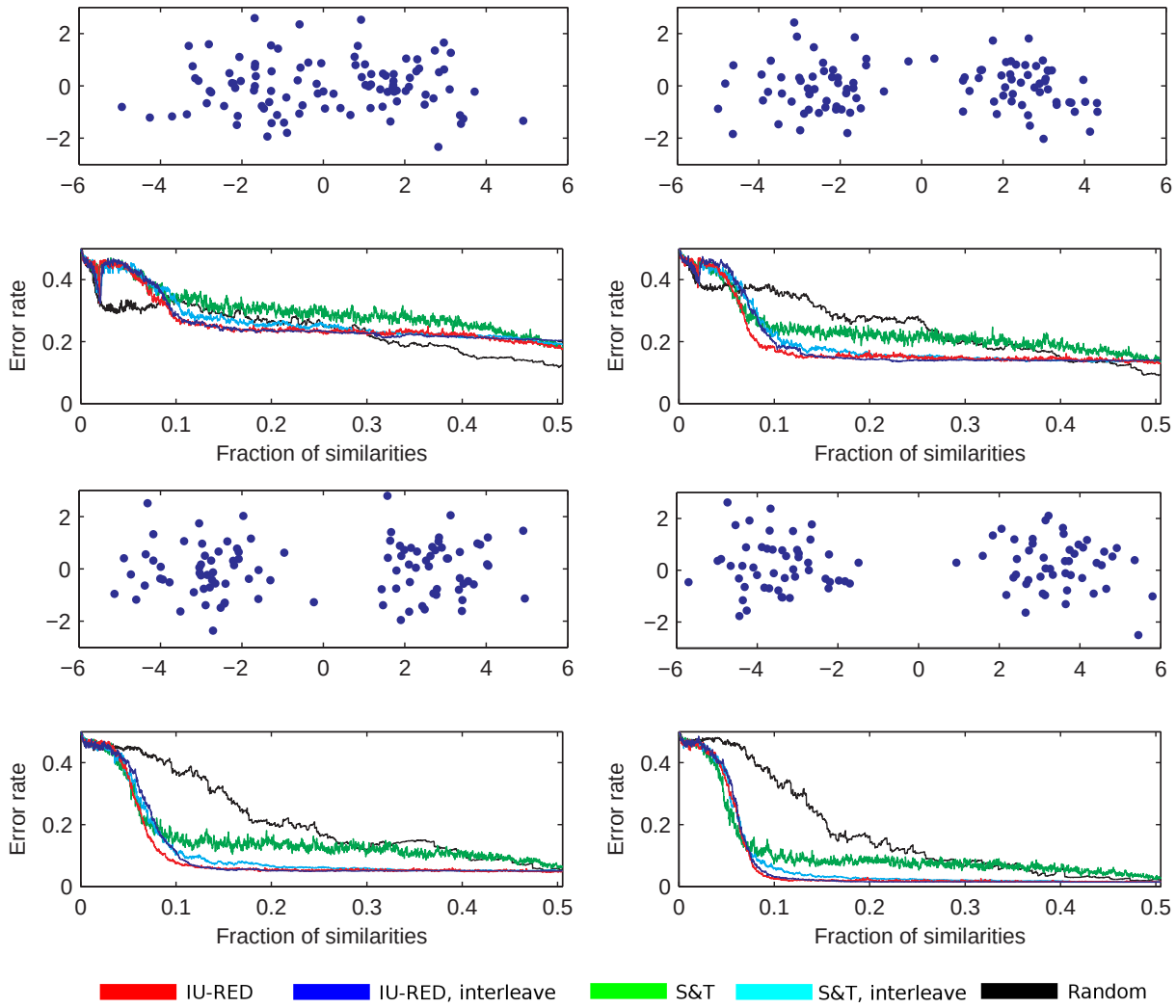


Figure 2.2: Results on synthetic datasets. For each dataset we sampled a total of 200 points from two Gaussians of increasing separation. Methods are evaluated on the average misclustering error rate, relative to a spectral clustering solution with complete data. Our proposed methods are "IU-RED" and "IU-RED, interleave".

investigate the effects of perturbations when all similarities are known. To our knowledge, active spectral clustering with costly and noisy measurements has not been considered.

The simplest way to deal with noise is to take the mean or median of multiple repeated measurements. However, given m repeated measurements of normally distributed similarities, both the mean and median have a standard deviation that is only a factor of $O(1/\sqrt{m})$ smaller than that of a single measurement. Thus, to halve the standard deviation we need about four times as many measurements. Thus, measuring every similarity multiple times is a fairly expensive way to reduce the effects of noise. It is especially wasteful since it is likely

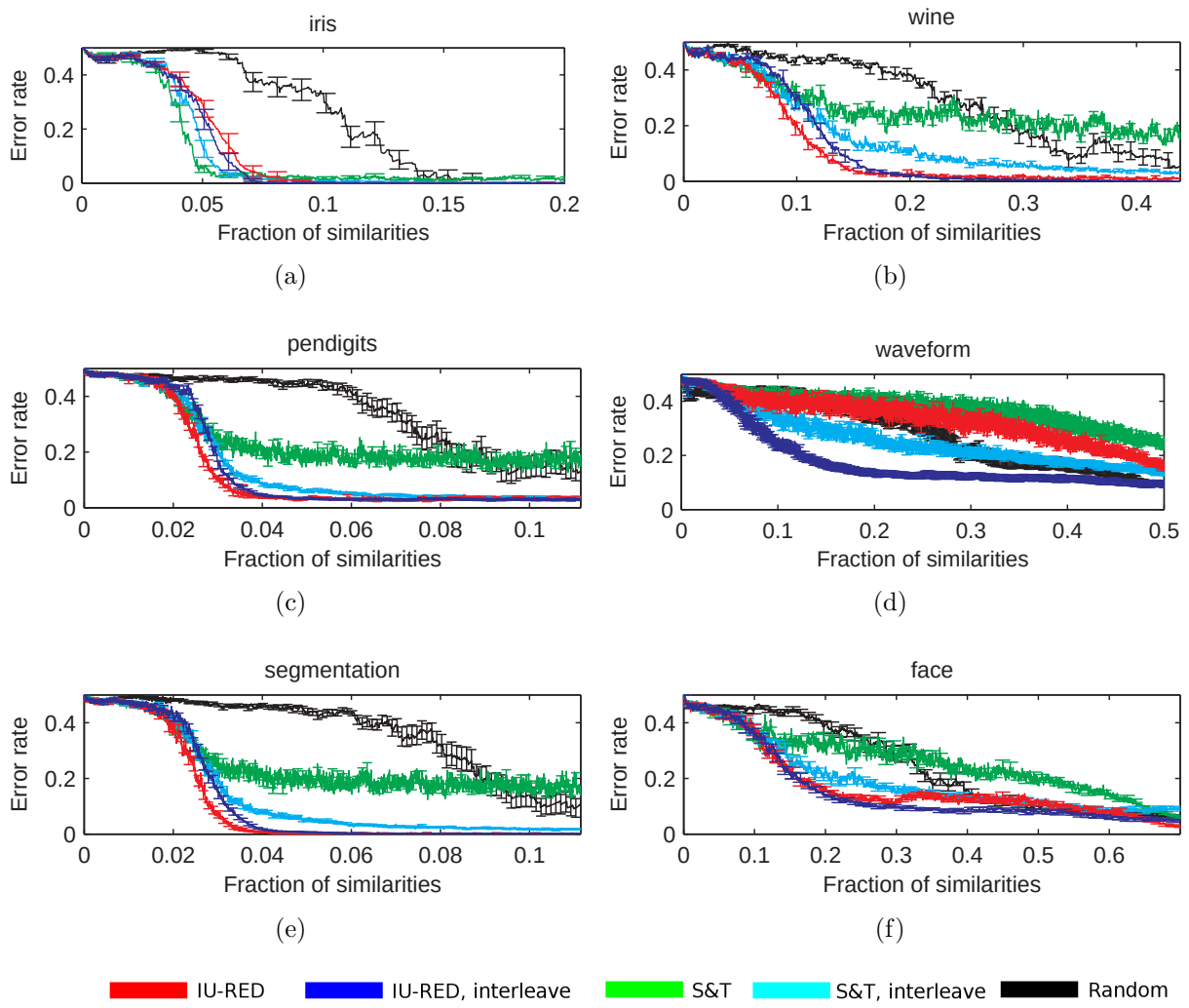


Figure 2.3: Results on several real datasets. Methods are evaluated on the average misclustering error rate, relative to a spectral clustering solution with complete data. Our proposed methods are "IU-RED" and "IU-RED, interleave". Note that the scaling of the x-axis changes between plots.

that only similarities of objects close to the cluster boundaries need to be known accurately to resolve the decision boundary. This section gives an active learning algorithm that asks for repeat measurements only when the measurement can significantly change the current solution and when our uncertainty in the true similarity is large.

We have extended IU-RED to maintain a "running median" estimate for each similarity. At any time, the true similarity is estimated as the median of any repeat measurements. Additionally, we maintain for each similarity estimate $\hat{w}_{ij}$ an estimate of its standard deviation, $\hat{\sigma}_{ij}$. When no measurements were made, we let the standard deviation be that of a uniform

Algorithm 2: S&T [69]

$S = \{(i,j) : i,j \in \{1,\ldots,n\}, i < j\}$
$\hat{W} = I$
for $t = 1,\ldots,n(n-1)/2$
$\hat{L} = \text{diag}(\hat{W}\mathbf{1}) - \hat{W} = \sum_{p=1}^{n} \hat{\lambda}_p \hat{v}_p \hat{v}_p^\top$ (1)
$(i^*, j^*) = \text{argmax}_{(i,j)\in S} \left\| \frac{d\hat{v}_2}{d\hat{w}_{ij}} \right\|_2^2$ (2)
$= \text{argmax}_{(i,j)\in S} \left\| \sum_{p>2}^{n} \frac{\hat{v}_2^\top \left[\partial \hat{L}/\partial \hat{w}_{ij}\right] \hat{v}_p}{\hat{\lambda}_2 - \hat{\lambda}_p} \hat{v}_p \right\|_2^2$ (3)
$S = S \setminus \{(i^*, j^*)\}$
$\hat{w}_{i^*j^*} = w_{i^*j^*}, \hat{w}_{j^*i^*} = w_{j^*i^*}$
return Second eigenvector of $\hat{L} = \text{diag}(\hat{W}\mathbf{1}) - \hat{W}$

on [0, 1]; i.e., $\hat{\sigma}_{ij} = \sqrt{1/12} \approx 0.2887$. After a single measurement we set $\hat{\sigma}_{ij} = s$, an estimate of the population standard deviation, and we let $\hat{\sigma}_{ij} = s/\sqrt{m}$ for m repeat measurements.[2] IU-RED was then modified to choose that measurement (possibly a repeat) where the product $\hat{\sigma}_{ij} |d\hat{v}_2(k_{min})/d\hat{w}_{ij}|$ is maximal. This amounts to choosing that measurement where our current estimate is uncertain *and* where the uncertainty matters. The S&T algorithm can be similarly modified and is considered in that form during our experiments.

2.6 Experiments

We begin this section by evaluating the basic IU-RED algorithm of Section 2.4 on synthetic and real datasets. Subsection 2.6 then considers the extension to noisy measurements.

Our methods include IU-RED and a version of IU-RED with interleaved random selection. We compared against three alternatives: S&T, S&T with interleaved random selection, and random selection only. The last three methods have been previously evaluated in Shamir and Tishby [69] using a similar evaluation methodology as we consider here. We consider binary classification, which can be extended to more than two clusters by recursive splitting or by expanding the reasoning outlined here to more eigenvectors. Shamir and Tishby also evaluated a Nyström algorithm [29], but often found performance to be poor if the low rank assumption was not met. We evaluate all methods on the misclustering error, relative to a spectral clustering solution with complete dat and give average results over 20 runs.

[2]Several variations of this theme could be considered. Given enough samples, frequentist confidence intervals of the median could be estimated using the bootstrap. Alternatively, given prior information one could use Bayesian techniques to estimate posterior means and variances.

Figure 2.4: Spectral clustering of photos with complete data. The top row shows example views of the kitchen, and the bottom row example views of the living room. Humans can easily determine that photos in each row were probably taken in the same room, but a computer algorithm would have difficulty solving this task.

Synthetic Datasets

We first present results on a simple clustering task in which the data is drawn from mixtures of Gaussians with two components. The data and results are shown in Figure 2.2. For each dataset we sampled a total of 200 points from two Gaussians of increasing separation. We normalized the data to lie in the unit hypercube and used a standard radial basis function kernel to compute similarities. As expected, random selection usually performs worst, except when the cluster separation is minimal. Compared to our two algorithms, S&T does poorly even on easy problems. As reported in [69], interleaving with random selection significantly boosts performance. IU-RED outperforms both versions of S&T early on, but ties with them towards the end. Interestingly, while interleaving IU-RED appears to eventually stabilize the error rates, it slightly hurts performance early on.

Real Datasets

We have also evaluated our algorithm on a variety of real datasets. Five of the sets in this subsection are from the UCI repository [30] (iris, wine, pendigits, waveform, segmentation). An additional dataset concerning the similarity of faces is available from the University of Washington [59]. We followed [43, 69] and normalized the data to lie in the unit hypercube and used the Gaussian kernel to compute similarities.[3] Figures 2.3(a)–2.3(e) show the results on the UCI datasets. Figure 2.3(f) shows results on the face dataset. Note that the scaling of the x-axis changes between plots. The difference between methods is amplified on these datasets. Except for the iris dataset in Figure 2.3(a), S&T performs relatively poorly, and

[3]Although the UCI datasets were previously analyzed in [69], the results are not directly comparable, since their evaluations used different kernel parameters. Also, since each of these datasets contains more than two classes, it is possible that we chose two different classes for evaluating the spectral clustering.

is eventually outperformed by random selection. On three of five UCI datasets IU-RED outperforms all other methods early on. Interleaving usually increases the error initially, but eventually leads to a more stable algorithm with marginally lower error. The exception is the waveform dataset in Figure 2.3(d) where interleaving helps significantly. Lastly, on the face dataset IU-RED also outperforms S&T early on, with slight gains for interleaving.

We conducted a further experiment to assess whether the superior performance of IU-RED over S&T can be seen after a single active learning step, or only emerges after many such steps. In this experiment, we sampled a subset of similarities of approximately constant size uniformly at random and measured the decrease in error rate over one active selection step. These results were averaged over 30000 restarts. The first five rows of Table 2.1 show results for UCI data, the sixth row for the face data, and the last row for similarity matrices with off-diagonal entries uniform in [0, 1]. On four out of five UCI datasets and on the face dataset, IU-RED decreases the error more than S&T. The one dataset where we perform worse is the waveform dataset, which Figure 2.3(d) shows to be challenging for all methods. Overall, our method performs much better than S&T on structured similarity matrices. On random matrices both perform poorly, but now S&T performs better than IU-RED.

Wearable Camera Dataset

Our next experiment focuses on the realistic example outlined in the Introduction where similarities are hard to compute and noisy. The data of interest is a photo stream, acquired by a wearable camera at a rate of about one photo per 20 seconds. Clustering the data by the location at which an image was taken may be useful in a variety of applications, ranging from health (e.g., in diagnosis and life quality improvement for Alzheimer's patients) to summarization of personal memories. Because the data is collected in an entirely unconstrained way, analyzing it is beyond the capabilities of current unsupervised algorithms. The top row of Figure 2.4, for example, shows five images taken in the same kitchen. Clustering clearly requires some human input; at the very least a preliminary annotation that could be used to train supervised computer vision algorithms. It is impractical to ask annotators to simply label images by their location, since salient locations and their number only become evident as the stream progresses and may change from week to week, and from subject to subject. Also, locations may be interconnected, and multiple locations might be visible from the same viewpoint. In our data, for example, an open kitchen connects to the living room so that large parts of the space can be perceived to belong to both rooms. It is much more natural to collect similarities between images and to infer a clustering from that data.

We took this approach in order to cluster 100 images taken from the photo stream described above [49]. Of these, about 50 were taken in an open kitchen, and 50 were taken in the adjacent living room. Some example images are shown in Figure 2.4. We asked workers on Amazon Mechanical Turk to rate, on a scale from 1 to 10, how likely it was that a given pair of images was taken in the same room, with 10 indicating certainty. The user ratings were divided by 10 and then used as similarities. Humans are adept at matching rooms by a loose jumble of visible objects, making this task fairly realistic. Indeed, the two

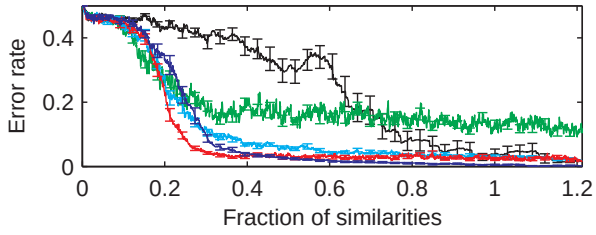
rows of Figure 2.4 show representative examples from a partition that was found by spectral clustering using the complete median similarity matrix, with the median running over three repeat measurements. A subsampled version of the median similarity matrix was shown in Figure 2.1 on the left. To collect one similarity for each pair of photos costs a total of US$74, so the three repeats required for the median cost a total of US$222.

We first evaluated our algorithms on the median similarity matrix using the same methods as before. The results are shown in Figure 2.5(a). The legend is the same as in Figure 2.2. Note that the x-axis is scaled to extend beyond 1.0, to account for the three repeat measurements necessary to compute one median similarity. A fraction of f indicates that a total of $fn(n-1)/2$ pairwise measurements were made. As before, our method outperforms a number of competitors early on. The results can be also interpreted in terms of the amount of money that must be expended to achieve a clustering result of fixed quality. Each image comparison cost us US$0.045 on Amazon Mechanical Turk. Table 2.2 shows the resulting approximate cost in US$ for each algorithm in order to achieve an error rate of 0.05. IU-RED is at least 4 times cheaper than S&T, which costs more than 30% of the complete-labelling cost. This difference can easily render larger image clustering tasks than ours impractical.

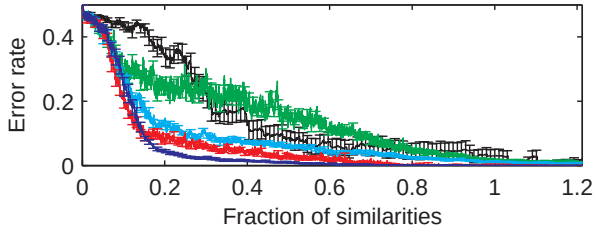
We also evaluated how IU-RED and S&T compare over only one active choice. The result is shown in the row of Table 2.1 labeled "photos." As before, our active learning framework outperforms that of Shamir and Tishby.

Next, we consider the extension of IU-RED to deal with noisy similarities, as outlined in Section 2.5. Figure 2.6 illustrates the type of noise encountered in this labelling task. We allow up to three repeat measurements of similarities that are known with high uncertainty and which can potentially change the current solution. The results are shown in Figure 2.5(b). For comparison, Figure 2.5(c) shows results when no repeat measurements are allowed and the standard deviation is not estimated. The latter is the extreme counterpart to measuring every similarity three times. All error rates are relative to a spectral clustering computed from the complete median similarity matrix, averaged over 20 runs. As before, we scaled the x-axes to show the *effective* fraction of pairwise measurements that was made. For the no-repeats framework in Figure 2.5(c) this fraction cannot be larger than 1.0. Another important consequence of this measuring framework is that all algorithms should yield approximately the same average error rate at a fraction of 1.0, since at that point every algorithm has observed one complete set of (noisy) similarities. The differences between algorithms are thus only appreciated in the first half of the Figure 2.5(c).

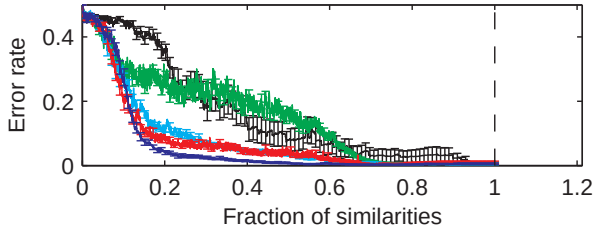
Both versions of IU-RED continue to beat the remaining algorithms in either of the two new settings. However, random interleaving now helps slightly, where it was detrimental before. With the exception of S&T, most methods improve early on compared to Figure 2.5(a). On the one hand this is intuitive, since as long as the similarities are not extremely noisy, three measurements do not convey three times as much information as one. One the other hand, it suggests that moderately noisy measurements can still be quite informative. Even so, we emphasize again that all algorithms should perform identically at a fraction of 1.0 in Figure 2.5(c). Maintaining running estimates and their accuracies is therefore preferable as it does not force performance to equalize once a fixed measurement quantum has been



(a)



(b)



(c)

Figure 2.5: Results on the "photos" dataset. Figure 2.5(a) shows results when true similarities are estimated as the median of three measurements. Figure 2.5(b) shows results when the algorithm is allowed choose which repeat measurements to make. Up to three repeats are allowed. Figure 2.5(c) shows results when similarities are estimated by a single noisy measurement. The legend is the same as in Figure 2.3.

Figure 2.6: Worker disagreement for two image comparisons on Amazon Mechanical Turk. For the left two photos, workers agreed they were certainly taken in the same room; for the right two, one worker asserted they were definitely not.

reached. Indeed, both IU-RED and IU-RED with random interleaving perform marginally better in Figure 2.5(b) than Figure 2.5(c) once a fraction of 1.0 measurements is reached.

Dataset	IU-RED	S&T
iris	0.0078 ± 0.0002	0.0016 ± 0.0001
wine	0.0103 ± 0.0003	-0.0010 ± 0.0003
pendigits	0.0067 ± 0.0002	-0.0007 ± 0.0002
waveform	-0.0018 ± 0.0002	0.0008 ± 0.0001
segmentation	0.0104 ± 0.0002	0.0002 ± 0.0002
face	0.0009 ± 0.0001	0.0001 ± 0.0001
photos	0.0126 ± 0.0003	-0.0021 ± 0.0002
uniform	-0.0057 ± 0.0009	-0.0037 ± 0.0007

Table 2.1: Average decrease in error rate across one selection step. IU-RED generally decreases the error more than S&T.

Random	S&T	S&T, inter.	IU-RED	IU-RED, inter.
\$53	> \$70	\$32	\$17	\$21

Table 2.2: Approximate labelling costs in US$ to achieve a 0.05 error rate on the photos dataset.

2.7 Conclusions

In this paper we have presented and evaluated an active learning algorithm for spectral clustering. Our main insight is that similarity matrices are not random, but usually exhibit clear clustering structure. Even when observing only a small fraction of the data, this structure becomes evident. Furthermore, assuming that the data clusters well, the final v_2 will usually have elements well away from zero. Motivated by these observations, our algorithm uses the current estimate $\hat{v}_2$ to choose measurements that will be most useful in removing elements close to zero, i.e., to push the two clusters in $\hat{v}_2$ apart. We have applied this algorithm to a range of datasets and showed that it generally outperforms a related algorithm by Shamir and Tishby.

The effects of costly and noisy similarities have so far been ignored in the active learning setting. We propose an extension of our algorithm that maintains running estimates of the true similarities as well as their accuracies. By taking these accuracies into account during query selection, we can potentially avoid unnecessary repeat measurements and speed up the learning process in noisy settings.

Rahimi and Recht [62] previously showed that a version of spectral clustering related to normalized cuts [72] clusters data by finding a hyperplane that cuts the data in a lifted space. The signed distances of points to the hyperplane are given by rescaled elements of an Laplacian eigenvector, and the partitioning can be done by taking the sign of the distances. If we have that $W\mathbf{1} = c\mathbf{1}$, for some $c > 0$, then their result implies that our version of spectral clustering also finds such a hyperplane, and that the signed distances are proportional to the second eigenvector v_2. Our active learning approach can then be interpreted as choosing measurements that can maximally perturb the margin between a hyperplane and the lifted

datapoints. Rahimi and Recht's observation has recently been used to derive an active learning rule for the spectral graph transducer [7]. Here W is completely known, but for each object an additional binary class label can be queried. The rule chooses to label that point next which is currently closest to a hyperplane. Similar heuristics have also been employed in a number of other classifiers and clustering frameworks [13, 54, 81, 92]. In all these methods, however, the pairwise similarities are assumed to be known initially (either implicitly or explicitly) but additional labels or constraints can be queried. In contrast, our setting allows for incomplete similarities which we can (perhaps only noisily) measure at high cost.

Chapter 3

Bayesian Bias Mitigation for Crowdsourcing

As models become more complex, traditional methodologies for active learning tend to become harder to implement. For example, consider complex Bayesian models, which often rely on an MCMC-based method for inference. Here, active learning is commonly thought to be computationally infeasible, since a naïve scoring implementation would require running many additional MCMC chains. We consider this problem in the context of a Bayesian model for crowdsourcing. Because the crowdsourced data is complex, the Bayesian model is also complex, and we chose to use an MCMC-based inference procedure. After introducing the model, we propose a new MCMC-based framework for tractable, approximate active learning, which reuses samples from an existing MCMC chain for approximate scoring. This avoids running extra MCMC chains and outperforms the naïve approach.

3.1 Crowdsourcing

Crowdsourcing is becoming an increasingly important methodology for collecting labeled data, as demonstrated among others by Amazon Mechanical Turk, reCAPTCHA, Netflix, and the ESP game. Motivated by the promise of a wealth of data that was previously impractical to gather, researchers have focused in particular on Amazon Mechanical Turk as a platform for collecting label data [75, 77]. Unfortunately, the data collected from crowdsourcing services is often very dirty: Unhelpful labelers may provide incorrect or biased responses that can have major, uncontrolled effects on learning algorithms. Bias may be caused by personal preference, systematic misunderstanding of the labeling task, lack of interest or varying levels of competence. Further, as soon as malicious labelers try to exploit incentive schemes in the data collection cycle yet more forms of bias enter.

The typical crowdsourcing pipeline can be divided into three main steps: 1) *Data collection.* The researcher farms the labeling tasks to a crowdsourcing service for annotation and possibly adds a small set of gold standard labels. 2) *Data curation.* Since labels from

the crowd are contaminated by errors and bias, some filtering is applied to curate the data, possibly using the gold standard provided by the researcher. 3) *Learning*. The final model is learned from the curated data.

At present these steps are often treated as separate. The data collection process is often viewed as a black box which can only be minimally controlled. Although the potential for active learning to make crowdsourcing much more cost effective and goal driven has been appreciated, research on the topic is still in its infancy [24, 71, 93]. Similarly, data curation is in practice often still performed as a preprocessing step, before feeding the data to a learning algorithm [44, 64, 73, 75, 77, 83]. We believe that the lack of systematic solutions to these problems can make crowdsourcing brittle in situations where labelers are arbitrarily biased or even malicious, such as when tasks are particularly ambiguous/hard or when opinions or ratings are solicited.

Our goal in the current paper is to show how crowdsourcing can be leveraged more effectively by treating the overall pipeline in a Bayesian framework. We present Bayesian Bias Mitigation for Crowdsourcing (BBMC) as a way to achieve this. BBMC makes two main contributions.

The first is a flexible latent feature model that describes each labeler's idiosyncrasies through multiple shared factors and allows us to combine data curation and learning (steps 2 and 3 above) into one inferential computation. Most of the literature accounts for the *effects* of labeler bias by assuming a single, true latent labeling from which labelers report noisy observations of some kind [20, 21, 24, 44, 64, 71, 73, 75, 89, 90, 93, 94]. This assumption is inappropriate when labels are solicited on subjective or ambiguous tasks (ratings, opinions, and preferences) or when learning must proceed in the face of arbitrarily biased labelers. We believe that an unavoidable and necessary extension of crowdsourcing allows multiple distinct (yet related) "true" labelings to co-exist, but that at any one time we may be interested in learning about only one of these "truths." Our BBMC framework achieves this by modeling the *sources* of labeler bias through shared random effects.

Next, we want to perform active learning in this model to actively query labelers, thus integrating step 1 with steps 2 and 3. Since our model requires Gibbs sampling for inference, a straightforward application of active learning is infeasible: Each active learning step relies on many inferential computations and would trigger a multitude of subordinate Gibbs samplers to be run within one large Gibbs sampler. Our second contribution is a new methodology for solving this problem. The basic idea is to approximate the stationary distribution of a perturbed Markov chain using that of an unperturbed chain. We specialize this idea to active learning in our model and show that the computations are efficient and that the resulting active learning strategy substantially outperforms other active learning schemes.

The paper is organized as follows: We discuss related work in Section 3.2. In Section 3.3 we propose the latent feature model for labelers and in Section 3.4 we discuss the inference procedure that combines data curation and learning. Then we present a general method to approximate the stationary distribution of perturbed Markov chains and apply it to derive an efficient active learning criterion in Section 3.5. In Section 3.6 we present comparative results and we draw conclusions in Section 3.7.

3.2 Related Work

Relevant work on active learning in multi-teacher settings has been reported in [24, 71, 93]. Sheng et al. [71] use the multiset of current labels with a random forest label model to score which task to next solicit a repeat label for. The quality of the labeler providing the new label does not enter the selection process. In contrast, Donmez et al. [24] actively choose the labeler to query next using a formulation based on interval estimation, utilizing repeated labelings of tasks. The task to label next is chosen separately from the labeler. In contrast, our BBMC framework can perform meaningful inferences even without repeated labelings of tasks and treats the choices of which labeler to query on which task as a joint choice in a Bayesian framework. Yan et al. [93] account for the effects of labeler bias through a coin flip observation model that filters a latent label assignment, which in turn is modeled through a logistic regression. As in [24], the labeler is chosen separately from the task by solving two optimization problems. In other work on data collection strategies, Wais et al. [83] require each labeler to first pass a screening test before they are allowed to label any more data. In a similar manner, reputation systems of various forms are used to weed out historically unreliable labelers before collecting data.

Consensus voting among multiple labels is a commonly used data curation method [77, 83]. It works well when low levels of bias or noise are expected but becomes unreliable when labelers vary greatly in quality [71]. Earlier work on learning from variable-quality teachers was revisited by Smyth et al. [73] who looked at estimating the unknown true label for a task from a set of labelers of varying quality without external gold standard signal. They used an EM strategy to iteratively estimate the true label and the quality of the labelers. The work was extended to a Bayesian formulation by Raykar et al. [64] who assign latent variables to labelers capturing their mislabeling probabilities. Ipeirotis et al. [44] pointed out that a biased labeler who systematically mislabels tasks is still more useful than a labeler who reports labels at random. A method is proposed that separates low quality labelers from high quality, but biased labelers. Dekel and Shamir [21] propose a two-step process. First, they filter labelers by how far they disagree from an estimated true label and then retrain the model on the cleaned data. They give a generalization analysis for anticipated performance. In a similar vein, Dekel and Shamir [20] show that, under some assumptions, restricting each labeler's influence on a learned model can control the effect of low quality or malicious labelers. Together with [64, 90, 94], [20] and [21] are among the recent lines of research to combine data curation and learning. Work has also focused on using gold standard labels to determine labeler quality. Going beyond simply counting tasks on which labelers disagree with the gold standard, Snow et al. [75] estimate labeler quality in a Bayesian setting by comparing to the gold standard.

Collaborative filtering has looked extensively at completing sparse matrices of ratings [79]. Given some gold standard labels, collaborative filtering methods could also be used to curate data in a sparse label matrix. However, collaborative filtering generally does not combine this inference with the learning of a labeler-specific model for prediction (step 3). With the exception of [96], active learning has not been studied in the collaborative filtering setting.

3.3 Modeling Labeler Bias

In this section we specify a Bayesian latent feature model that accounts for labeler bias and allows us to combine data curation and learning into a single inferential calculation. For ease of exposition we will focus on binary classification, but our method can be generalized. Suppose we solicited labels for n tasks from m labelers. In practical settings it is unlikely that a task is labeled by more than 3–10 labelers [83]. Let task descriptions $x_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, be collected in the matrix X. The label responses are recorded in the matrix Y so that $y_{i,l} \in \{-1, 0, +1\}$ denotes the label given to task i by labeler l. The special label 0 denotes that a task was not labeled. A researcher is interested in learning a model that can be used to predict labels for new tasks. When consensus is lacking among labelers, our desideratum is to predict the labels that the researcher (or some other expert) would have assigned, as opposed to labels from an arbitrary labeler in the crowd. In this situation it makes sense to stratify the labelers in some way. To facilitate this, the researcher r provides gold standard labels in column r of Y to a small subset of the tasks. Loosely speaking, the gold standard allows our model to curate the data by softly combining labels from those labelers whose responses will useful in predicting r's remaining labels. It is important to note that our model is entirely symmetric in the role of the researcher and labelers. If instead we were interested in predicting labels for labeler l, we would treat column l as containing the gold standard labels. The researcher r is just another labeler, the only distinction being that we wish to learn a model that predicts r's labels. To simplify our presentation, we will accordingly refer to labelers in the crowd and the researcher occasionally just as "labelers," indexed by l, and only use the distinguishing index r when necessary. We account for each labeler l's idiosyncrasies by assigning a parameter $\beta_l \in \mathbb{R}^d$ to l and modeling labels $y_{i,l}$, $i = 1, \ldots, n$, through a probit model $p(y_{i,l}|x_i, \beta_l) = \Phi(y_{i,l} x_i^\top \beta_l)$, where $\Phi(\cdot)$ is the standard normal CDF. This section describes a joint Bayesian prior on parameters β_l that allows for parameter sharing; two labelers that share parameters have similar responses. In the context of this model, the two-step process of data curation and learning a model that predicts r's labels is reduced to posterior inference on β_r given X and Y. Inference softly integrates labels from relevant labelers, while at the same time allowing us to predict r's remaining labels.

Latent feature model

Labelers are not independent, so it makes sense to impose structure on the set of β_l's. Specifically, each vector β_l is modeled as the sum of a set of latent factors that are shared across the population. Let z_l be a latent binary vector for labeler l whose component $z_{l,b}$ indicates whether the latent factor $\gamma_b \in \mathbb{R}^d$ contributes to β_l. In principle, our model allows for an infinite number of distinct factors (i.e., z_l is infinitely long), as long as only a finite number of those factors is active (i.e., $\sum_{b=1}^{\infty} z_{l,b} < \infty$). Let $\gamma = (\gamma_b)_{b=1}^{\infty}$ be the concatenation of the factors γ_b. Given a labeler's vector z_l and factors γ we define the parameter $\beta_l = \sum_{b=1}^{\infty} z_{l,b} \gamma_b$.

For multiple labelers we let the infinitely long matrix $Z = (z_1, \ldots, z_m)^\top$ collect the vectors z_l and define the index set of all observed labels as $L = \{(i,l) : y_{i,l} \neq 0\}$. The likelihood is then written as

$$p(Y|X,\gamma,Z) = \prod_{(i,l)\in L} p(y_{i,l}|x_i,\gamma,z_l) = \prod_{(i,l)\in L} \Phi(y_{i,l} x_i^\top \beta_l). \tag{3.1}$$

To complete the model we need to specify priors for γ and Z. We define the prior distribution of each γ_b to be a zero-mean Gaussian $\gamma_b \sim \mathcal{N}(0, \sigma^2 I)$, and let Z be governed by an Indian Buffet Process (IBP) $Z \sim \text{IBP}(\alpha)$, parameterized by α [40]. The IBP is a stochastic process on infinite binary matrices consisting of vectors z_l. A central property of the IBP is that with probability one, a sampled matrix Z contains only a finite number of nonzero entries, thus satisfying our requirement that $\sum_{b=1}^{\infty} z_{l,b} < \infty$. In the context of our model this means that when working with finite data, with probability one only a finite set of features is active across all labelers. To simplify notation in subsequent sections, we use this observation and collapse an infinite matrix Z and vector γ to finite dimensional equivalents. From now on, we think of Z as the finite matrix having all zero-columns removed. Similarly, we think of γ as having all blocks γ_b corresponding to zero-columns in the original matrix Z removed. With probability one, the number of columns $K(Z)$ of Z is finite so we may write $\beta_l = \sum_{b=1}^{K(Z)} z_{l,b}\gamma_b \triangleq Z_l^\top \gamma$, with $Z_l = z_l \otimes I$ the Kronecker product of z_l and I.

3.4 Inference: Data Curation and Learning

We noted before that our model combines data curation and learning in a single inferential computation. In this section we lay out the details of a Gibbs sampler for achieving this. Given a task j which was not labeled by r (and possibly no other labeler), we need the predictive probability

$$p(y_{j,r} = +1|X,Y) = \int p(y_{j,r} = +1|x_j,\beta_r) p(\beta_r|X,Y) d\beta_r. \tag{3.2}$$

To approximate this probability we need to gather samples from the posterior $p(\beta_r|Y,X)$. Equivalently, since $\beta_r = Z_r^\top \gamma$, we need samples from the posterior $p(\gamma, z_r|Y,X)$. Because latent factors can be shared across multiple labelers, the posterior will softly absorb label information from labelers whose latent factors tend to be similar to those of the researcher r. Bayesian inference $p(\beta_r|Y,X)$ combines data curation and learning by weighting label information through an inferred sharing structure. Importantly, the posterior is informative even when no labeler in the crowd labeled any of the tasks the researcher labeled.

Gibbs sampling

For Gibbs sampling in the probit model one commonly augments the likelihood in Eq. (3.1) with intermediate random variables $T = \{t_{i,l} : y_{i,l} \neq 0\}$. The generative model for the label

$y_{i,l}$ given x_i, γ and z_l first samples $t_{i,l}$ from a Gaussian $\mathcal{N}(\beta_l^\top x_i, 1)$. Conditioned on $t_{i,l}$, the label is then defined as $y_{i,l} = 2\,\mathbf{1}[t_{i,l} > 0] - 1$. Figure 3.1(a) summarizes the augmented graphical model by letting β denote the collection of β_l variables. We are interested in sampling from $p(\gamma, z_r|Y, X)$. The Gibbs sampler for this lives in the joint space of T, γ, Z and samples iteratively from the three conditional distributions $p(T|X, \gamma, Z), p(\gamma|X, Z, T)$ and $p(Z|\gamma, X, Y)$. The different steps are:

Sampling T given X, γ, Z: We independently sample elements of T given X, γ, Z from a truncated normal as

$$(t_{i,l}|X, \gamma, Z) \sim \mathcal{N}^{y_{i,l}}(t_{i,l}|\gamma^\top Z_l x_i, 1), \tag{3.3}$$

where we use $\mathcal{N}^{-1}(t|\mu, 1)$ and $\mathcal{N}^{+1}(t|\mu, 1)$ to indicate the density of the negative- and positive-orthant-truncated normal with mean μ and variance 1, respectively, evaluated at t.

Sampling γ given X, Z, T: Straightforward calculations show that conditional sampling of γ given X, Z, T follows a multivariate Gaussian

$$(\gamma|X, Z, T) \sim \mathcal{N}(\gamma|\mu, \Sigma), \tag{3.4}$$

where

$$\Sigma^{-1} = \frac{I}{\sigma^2} + \sum_{(i,l)\in L} Z_l x_i x_i^\top Z_l^\top \qquad\qquad \mu = \Sigma \sum_{(i,l)\in L} Z_l x_i t_{i,l}. \tag{3.5}$$

Sampling Z given γ, X, Y: Finally, for inference on Z given γ, X, Y we may use techniques outlined in [40]. We are interested in performing active learning in our model, so it is imperative to keep the conditional sampling calculations as compact as possible. One simple way to achieve this is to work with a finite-dimensional approximation to the IBP: We constrain Z to be an $m \times K$ matrix, assigning each labeler at most K active latent features. This is not a substantial limitation; in practice the truncated IBP often performs comparably, and for $K \to \infty$ converges in distribution to the full IBP [40]. Let $m_{-l,b} = \sum_{l' \neq l} z_{l',b}$ be the number of labelers, excluding l, with feature b active. Define $\beta_l(z_{l,b}) = z_{l,b}\gamma_b + \sum_{b' \neq b} z_{l,b'}\gamma_{b'}$ as the parameter β_l either specifically including or excluding γ_b. Now if we let $z_{-l,b}$ be the column b of Z, excluding element $z_{l,b}$ then updated elements of Z can be sampled one by one as

$$p(z_{l,b} = 1|z_{-l,b}) = \frac{m_{-l,b} + \frac{\alpha}{K}}{n + \frac{\alpha}{K}} \tag{3.6}$$

$$p(z_{l,b}|z_{-l,b}, \gamma, X, Y) \;\propto\; p(z_{l,b}|z_{-l,b}) \prod_{i:y_{i,l}\neq 0} \Phi(y_{i,l} x_i^\top \beta_l(z_{l,b})). \tag{3.7}$$

After reaching approximate stationarity, we collect samples $(\gamma^s, Z^s), s = 1, \ldots, S$, from the Gibbs sampler as they are generated. We then compute samples from $p(\beta_r|Y, X)$ by writing $\beta_r^s = Z_r^{s\top}\gamma^s$.

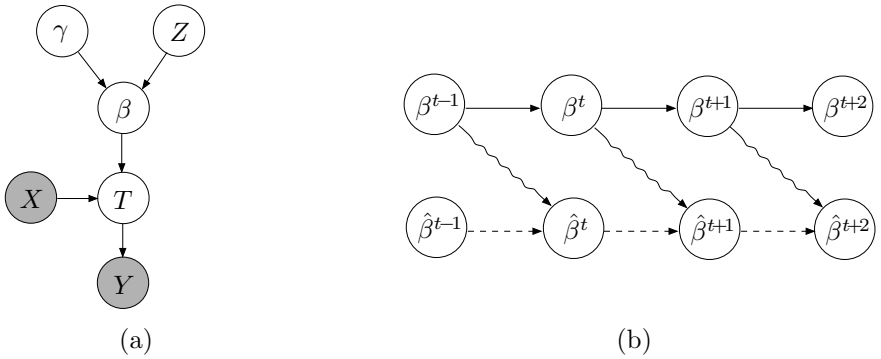


Figure 3.1: 3.1(a) A graphical model of the augmented latent feature model. Each node corresponds to a collection of random variables in the model. 3.1(b) A schematic of our approximation scheme. The top chain indicates an unperturbed Markov chain, the lower a perturbed Markov chain. Rather than sampling from the lower chain directly (dashed arrows), we transform samples from the top chain to approximate samples from the lower (wavy arrows).

3.5 Active Learning

The previous section outlined how, given a small set of gold standard labels from r, the remaining labels can be predicted via posterior inference $p(\beta_r|Y, X)$. In this section we take an active learning approach [15, 51] to incrementally add labels to Y so as to quickly learn about β_r while reducing data acquisition costs. Active learning allows us to guide the data collection process through model inferences, thus integrating the *data collection*, *data curation* and *learning* steps of the crowdsourcing pipeline. We envision a unified system that automatically asks for more labels from those labelers on those tasks that are most useful in inferring β_r. This is in contrast to [71], where labelers cannot be targeted with tasks. It is also unlike [24] since we can let labelers be arbitrarily unhelpful, and differs from [93] which assumes a single latent truth.

A well-known active learning criterion popularized by Lindley [51] is to label that task next which maximizes the prior-posterior reduction in entropy of an inferential quantity of interest. The original formulation has been generalized beyond entropy to arbitrary utility functionals $U(\cdot)$ of the updated posterior probability [15]. The functional $U(\cdot)$ is a model parameter that can depend on the type of inferences we are interested in. In our particular setup, we wish to infer the parameter β_r to predict labels for the researcher r. Suppose we chose to solicit a label for task i' from labeler l', which produced label $y_{i',l'}$. The utility of this observation is $U(p(\beta_r|y_{i',l'}))$. The average utility of receiving a label on task i' from labeler l' is $\mathcal{I}((i', l'), p(\beta_r)) = E(U(p(\beta_r|y_{i',l'})))$, where the expectation is taken with respect to the predictive label probabilities $p(y_{i',l'}|x_{i'}) = \int p(y_{i',l'}|x_{i'}, \beta_{l'})p(\beta_{l'})d\beta_{l'}$. Active learning chooses that pair (i', l') which maximizes $\mathcal{I}((i', l'), p(\beta_r))$. If we want to choose the next task for the

researcher to label, we constrain $l' = r$. To query the crowd we let $l' \neq r$. Similarly, we can constrain i' to any particular value or subset of interest. For the following discussion we let $U(p(\beta_r|y_{i',l'})) = \|E_{p(\beta_r)}(\beta_r) - E_{p(\beta_r|y_{i',l'})}(\beta_r)\|_2$ be the ℓ_2 norm of the difference in means of β_r. Picking the task that shifts the posterior mean the most is similar in spirit to the common criterion of maximizing the Kullback-Leibler divergence between the prior and posterior.

Active learning for MCMC inference

A straightforward application of active learning is impractical using Gibbs sampling, because to score a single task-labeler pair (i', l') we would have to run two Gibbs samplers (one for each of the two possible labels) in order to approximate the updated posterior distributions. Suppose we started with k task-labeler pairs that active learning could choose from. Depending on the number of selections we wish to perform, we would have to run $k \lesssim g \lesssim k^2$ Gibbs samplers *within* the topmost Gibbs sampler of Section 3.4. Clearly, such a scoring approach is not practical. To solve this problem, we propose a general purpose strategy to approximate the stationary distribution of a perturbed Markov chain using that of an unperturbed Markov chain. The approximation allows efficient active learning in our model that outperforms naïve scoring both in speed and quality.

The main idea can be summarized as follows. Suppose we have two Markov chains, $p(\beta_r^t|\beta_r^{t-1})$ and $\hat{p}(\hat{\beta}_r^t|\hat{\beta}_r^{t-1})$, the latter of which is a slight perturbation of the former. Denote the stationary distributions by $p_\infty(\beta_r)$ and $\hat{p}_\infty(\hat{\beta}_r)$, respectively. If we are given the stationary distribution $p_\infty(\beta_r)$ of the unperturbed chain, then we propose to approximate the perturbed stationary distribution by

$$\hat{p}_\infty(\hat{\beta}_r) \approx \int \hat{p}(\hat{\beta}_r|\beta_r) p_\infty(\beta_r) d\beta_r. \tag{3.8}$$

If $\hat{p}(\hat{\beta}^t|\hat{\beta}^{t-1}) = p(\hat{\beta}^t|\hat{\beta}^{t-1})$ the approximation is exact. Our hope is that if the perturbation is small enough the above approximation is good. To use this with MCMC, we first run the unperturbed MCMC chain to approximate stationarity, and then use samples of $p_\infty(\beta_r)$ to compute approximate samples from $\hat{p}_\infty(\hat{\beta}_r)$. Figure 3.1(b) shows this scheme visually.

To map this idea to our active learning setup we conceptually let the unperturbed chain $p(\beta_r^t|\beta_r^{t-1})$ be the chain on β_r induced by the Gibbs sampler in Section 3.4. The perturbed chain $\hat{p}(\hat{\beta}_r^t|\hat{\beta}_r^{t-1})$ represents the chain where we have added a new observation $y_{i',l'}$ to the measured data. If we have S samples β_r^s from $p_\infty(\beta_r)$, then we approximate the perturbed distribution as

$$\hat{p}_\infty(\hat{\beta}_r) \approx \frac{1}{S} \sum_{s=1}^{S} \hat{p}(\hat{\beta}_r|\beta_r^s), \tag{3.9}$$

and the active learning score as $U(p(\beta_r|y_{i',l'})) \approx U\Big(\hat{p}_\infty(\hat{\beta}_r)\Big)$. To further specialize this strategy to our model we first rewrite the Gibbs sampler outlined in Section 3.4. We suppress

mentions of X and Y in the subsequent presentation. Instead of first sampling $(T|\gamma^{t-1}, Z)$ from Eq. (3.3), and then sampling $(\gamma^t|T, Z)$ from Eq. (3.4), we combine them into one larger sampling step $(\gamma^t|\gamma^{t-1}, Z)$. Starting from a fixed γ^{t-1} and Z we sample from γ^t as

$$\left(\gamma^t|\gamma^{t-1}, Z\right) \stackrel{d}{=} \eta_\Sigma + \mu = \Sigma \left[\eta_{\sigma^{-2}I} + \sum_{(i,l)\in L} Z_l x_i \left[\eta_1 + \left(t_{i,l}|\gamma^{t-1}, Z\right)\right]\right], \tag{3.10}$$

where η_Σ is a zero-mean Gaussian with covariance Σ, and η_1 a standard normal random variable. If it were feasible, we could also absorb the intermediate sampling of Z into the notation and write down a single induced Markov chain $(\beta_r^t|\beta_r^{t-1})$, as referred to in Eqs. (3.8) and (3.9). As this is not possible, we will account for Z separately. We see that the effect of adding a new observation $y_{i',l'}$ is to perturb the Markov chain in Eq. (3.10) by adding an element to L. Supposing we added this new observation at time $t-1$, let $\Sigma_{(i',l')}$ be defined as Σ but with (i', l') added to L. Straightforward calculations detailed in the Appendix give that, conditioned on γ^{t-1}, Z, we can write the first step of the perturbed Gibbs sampler as a function of the unperturbed Gibbs sampler. If we let $A_{i',l'} = \Sigma Z_{l'} x_{i'} x_{i'}^\top Z_{l'}^\top / (1 + x_{i'}^\top Z_{l'}^\top \Sigma Z_{l'} x_{i'})$ for compactness, then we yield

$$\left(\gamma^t_{(i',l')}|\gamma^{t-1}, Z\right) \stackrel{d}{=} (I - A_{i',l'}) \left(\gamma^t|\gamma^{t-1}, Z\right) + \Sigma_{(i',l')} Z_{l'} x_{i'} \left[\eta_1 + \left(t_{i',l'}|\gamma^{t-1}, Z\right)\right]. \tag{3.11}$$

To approximate the utility $U(\cdot)$ we now appeal to Eq. (3.9) and estimate the difference in means using recent samples $\gamma^s, Z^s, s = 1, \ldots, S$ from the unperturbed sampler. In terms of Eqs. (3.10) and (3.11),

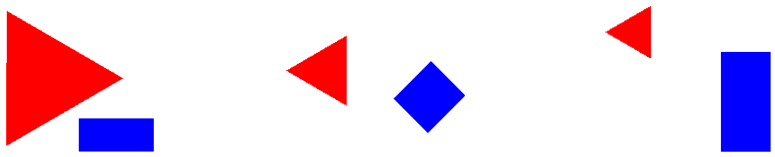
$$U(p(\beta_r|y_{i',l'})) = \left\| E_{p(\beta_r)}(\beta_r) - E_{p(\beta_r|y_{i',l'})}(\beta_r) \right\|_2 \tag{3.12}$$

$$\approx \left\| E\left(\frac{1}{S-1} \sum_{s=2}^{S} {Z_r^{s-1}}^\top \left[\left(\gamma|\gamma^{s-1}, Z^{s-1}\right) - \left(\gamma_{(i',l')}|\gamma^{s-1}, Z^{s-1}\right)\right] \right) \right\|_2. \tag{3.13}$$

Note that the sample γ^s is a realization of $(\gamma|\gamma^{s-1}, Z^{s-1})$. We have used this to approximate $E((\gamma|\gamma^{s-1}, Z^{s-1})) \approx \gamma^s$. Thus, the sum only runs over $S-1$ terms. Finally, we use samples from the Gibbs sampler to approximate $p(y_{i',l'}|x_{i'})$ and estimate $\mathcal{I}((i', l'), p(\beta_r))$ for querying labeler l' on task i'. The full derivation of the criterion is given in the Appendix.

3.6 Experiments

We evaluated our active learning method on an ambiguous localization task which asked labelers on Amazon Mechanical Turk to determine if a triangle was to the left or above a rectangle. Examples are shown in Figure 3.2. Tasks such as these are important for learning computer vision models of perception. Rotation, translation and scale, as well as aspect ratios, were pseudo-randomly sampled in a way that produced ambiguous tasks. We expected



(a) (b) (c)

Figure 3.2: Examples of easy and ambiguous labeling tasks. We asked labelers to determine if the triangle is to the left or above the square.

labelers to use centroids, extreme points and object sizes in different ways to solve the tasks, thus leading to structurally biased responses. Additionally, our model will also have to deal with other forms of noise and bias. The gold standard was to compare only the centroids of the two objects. For training we generated 1000 labeling tasks and solicited 3 labels for each task. Tasks were solved by 75 labelers with moderate disagreement. To emphasize our results, we retained only the subset of 523 tasks with disagreement. We provided about 60 gold standard labels to BBMC and then performed inference and active learning on β_r so as to learn a predictive model emulating gold standard labels. We evaluated methods based on the log likelihood and error rate on a held-out test set of 1101 datapoints.[1] All results shown in Table 3.1 were averaged across 10 random restarts. We considered two scenarios. The first compares our model to other methods when no active learning is performed. This will demonstrate the advantages of the latent feature model presented in Sections 3.3 and 3.4. The second scenario compares performance of our active learning scheme to various other methods. This will highlight the viability of our overall scheme presented in Section 3.5 that ties data collection together with data curation and learning.

First we show performance without active learning. Here only about 60 gold standard labels and all the labeler data is available for training. The results are shown in the top three rows of Table 3.1. Our method, "BBMC," outperforms the other two methods by a large margin. The BBMC scores were computed by running the Gibbs sampler of Section 3.4 with 2000 iterations burnin and then computing a predictive model by averaging over the next 20000 iterations. The alternatives include "GOLD," which is a logistic regression trained only on gold standard labels, and "CONS," which evaluates logistic regression trained on the overall majority consensus. Training on the gold standard only often overfits, and training on the consensus systematically misleads.

Next, we evaluate our active learning method. As before, we seed the model with about 60 gold standard labels. We repeatedly select a new task for which to receive a gold standard label from the researcher. That is, for this experiment we constrained active learning to use $l' = r$. Of course, in our framework we could have just as easily queried labelers in the crowd. Following 2000 steps burnin we performed active learning every 200 iterations

[1]The test set was similarly constructed by selecting from 2000 tasks those on which three labelers disagreed.

	Algorithm	Final Loglik	Final Error
No Active Learning	GOLD	-3716 ± 1695	0.0547 ± 0.0102
	CONS	-421.1 ± 2.6	0.0935 ± 0.0031
	BBMC	$\mathbf{-219.1 \pm 3.1}$	$\mathbf{0.0309 \pm 0.0033}$
Active Learning	GOLD-ACT	-1957 ± 696	0.0290 ± 0.0037
	CONS-ACT	-396.1 ± 3.6	0.0906 ± 0.0024
	RAND-ACT	-186.0 ± 2.2	0.0292 ± 0.0029
	DIS-ACT	-198.3 ± 5.8	0.0392 ± 0.0052
	MCMC-ACT	-196.1 ± 6.7	0.0492 ± 0.0050
	BBMC-ACT	$\mathbf{-160.8 \pm 3.9}$	$\mathbf{0.0188 \pm 0.0018}$

Table 3.1: Prediction results: Various models were evaluated on a test set of 1101 held-out tasks. The top three rows give results without and the bottom six rows results with active learning. For logistic regression models the final log likelihood (loglik) is the log likelihood of the learnt regression on the gold standard labels. For models using our bias model, it is the log likelihood in Eq. (3.1) evaluated on the gold standard labels and averaged over posterior samples of γ and Z. The error rate is computed by taking the maximum likelihood predicted task labels and comparing to the gold standard. Our proposed models, BBMC and BBMC-ACT outperform other algorithms in their category.

for a total of 100 selections. The reported scores were computed by estimating a predictive model from the last 200 iterations. The results are shown in the lower six rows of Table 3.1. Our model with active learning, "BBMC-ACT," outperforms all alternatives. The first alternative we compared against, "MCMC-ACT," does active learning with the MCMC-based scoring method outlined in Section 3.5. In line with our utility $U(\cdot)$ this method scores a task by running two Gibbs samplers within the overall Gibbs sampler and then approximates the expected mean difference of β_r. Due to time constraints, we could only afford to run each subordinate chain for 10 steps. Even then, this method requires on the order of 10×83500 Gibbs sampling iterations for 100 active learning steps. It takes about 11 hours to run the entire chain, while BBMC only requires 2.5 hours. The MCMC method performs very poorly. This demonstrates our point: Since the MCMC method computes a similar quantity as our approximation, it should perform similarly given enough iterations in each subchain. However, 10 iterations is not nearly enough time for the scoring chains to mix and also quite a small number to compute empirical averages, leading to decreased performance. A more realistic alternative to our model is "DIS-ACT," which picks one of the tasks with most labeler disagreement to label next. Baseline alternatives include "GOLD-ACT" and "CONS-ACT" which pick a random task to label and then learn logistic regressions on the gold standard or consensus labels respectively. Those results can be compared against "RAND-ACT," which uses our model and inference procedure but similarly selects tasks at random. As before, we outperform these two baseline methods when no active learning is done.

3.7 Conclusions

Bayesian Bias Mitigation for Crowdsourcing (BBMC) is a framework that unifies the three main steps in the crowdsourcing pipeline: data collection, data curation and learning. Our model aims to capture labeler bias through a flexible latent feature model and conceives of the entire pipeline in terms of probabilistic inference. An important contribution is a general purpose approximation strategy for Markov chains that allows us to efficiently perform active learning, despite relying on Gibbs sampling for inference. Our experiments show that BBMC is fast and greatly outperforms a number of commonly used alternatives.

3.8 Appendix

Perturbations of a one-step Markov chain

Suppose we have an unperturbed Markov chain $(\gamma^t|\gamma^{t-1}, Z)$ as given in Eq. (3.10) and we add a new observation $y_{i',l'}$ to it. We show that conditioned on γ^{t-1}, Z the first step of the perturbed Markov chain can be written as a function of the first step of the unperturbed Markov chain. Let

$$\Sigma^{-1}_{(i',l')} = \frac{I}{\sigma^2} + \sum_{(i,l)\in L\cup(i',l')} Z_l x_i x_i^\top Z_l^\top \tag{3.14}$$

We write $\Sigma_{(i',l')}$ in terms of Σ using the Sherman-Morrison-Woodbury equation

$$\Sigma_{(i',l')} = \Sigma - \frac{\Sigma Z_{l'} x_{i'} x_{i'}^\top Z_{l'}^\top \Sigma}{1 + x_{i'}^\top Z_{l'}^\top \Sigma Z_{l'} x_{i'}}, \tag{3.15}$$

and define $A_{i',l'} = \Sigma Z_{l'} x_{i'} x_{i'}^\top Z_{l'}^\top / (1 + x_{i'}^\top Z_{l'}^\top \Sigma Z_{l'} x_{i'})$. Substituting into Eq. 3.10 and simplifying we yield

$$(\gamma^t_{(i',l')}|\gamma^{t-1}, Z) \stackrel{d}{=} \Sigma_{(i',l')} \left[\eta_{\sigma^{-2}I} + \sum_{(i,l)\in L\cup(i',l')} Z_l x_i \left[\eta_1 + \left(t_{i,l}|\gamma^{t-1}, Z\right)\right]\right] \tag{3.16}$$

$$= (\gamma^t|\gamma^{t-1}, Z) + \Sigma Z_{l'} x_{i'} \left[\eta_1 + \left(t_{i',l'}|\gamma^{t-1}, Z\right)\right] - \tag{3.17}$$

$$\frac{\Sigma Z_{l'} x_{i'} x_{i'}^\top Z_{l'}^\top \Sigma}{1 + x_{i'}^\top Z_{l'}^\top \Sigma Z_{l'} x_{i'}} \left[\eta_{\sigma^{-2}I} + \sum_{(i,l)\in L\cup(i',l')} Z_l x_i \left[\eta_1 + \left(t_{i,l}|\gamma^{t-1}, Z\right)\right]\right] \tag{3.18}$$

$$= (\gamma^t|\gamma^{t-1}, Z) + \Sigma Z_{l'} x_{i'} \left[\eta_1 + \left(t_{i',l'}|\gamma^{t-1}, Z\right)\right] - \tag{3.19}$$

$$\frac{\Sigma Z_{l'} x_{i'} x_{i'}^\top Z_{l'}^\top \Sigma}{1 + x_{i'}^\top Z_{l'}^\top \Sigma Z_{l'} x_{i'}} \left[\Sigma^{-1}(\gamma^t|\gamma^{t-1}, Z) + Z_{l'} x_{i'} \left[\eta_1 + \left(t_{i',l'}|\gamma^{t-1}, Z\right)\right]\right] \tag{3.20}$$

$$= (I - A_{i',l'})(\gamma^t|\gamma^{t-1}, Z) + \Sigma_{(i',l')} Z_{l'} x_{i'} \left[\eta_1 + \left(t_{i',l'}|\gamma^{t-1}, Z\right)\right] \tag{3.21}$$

Eq. (3.21) tells us the first step of the perturbed Gibbs sampler as a function of the first step of the unperturbed Gibbs sampler, all conditioned on γ^{t-1}, Z.

Approximating the mean difference

Utilizing the approximation in Eq 3.9 we wish to approximate the mean difference norm as

$$U(p(\beta_r|y_{i',l'})) = \left\| E_{p(\beta_r)}(\beta_r) - E_{p(\beta_r|y_{i',l'})}(\beta_r) \right\|_2 \tag{3.22}$$

$$\approx \left\| E\left(\frac{1}{S-1} \sum_{s=2}^{S} {Z_r^{s-1}}^\top \left[\left(\gamma|\gamma^{s-1}, Z^{s-1}\right) - \left(\gamma_{(i',l')}|\gamma^{s-1}, Z^{s-1}\right) \right] \right) \right\|_2 \tag{3.23}$$

$$= \left\| E\left(\frac{1}{S-1} \sum_{s=2}^{S} {Z_r^{s-1}}^\top \left[A_{i',l'}\left(\gamma|\gamma^{s-1}, Z^{s-1}\right) - \Sigma_{(i',l')} Z_{l'}^{s-1} x_{i'} \left[\eta_1 + \left(t_{i',l'}|\gamma^{s-1}, Z^{s-1}\right) \right] \right] \right) \right\|_2 . \tag{3.24}$$

Pushing the expectation into the sum we make the following approximation: Rather than computing $E((\gamma|\gamma^{s-1}, Z^{s-1}))$ explicitly, we take the sample $\gamma^s \sim (\gamma|\gamma^{s-1}, Z^{s-1})$ that we already collected as an unbiased estimator of this mean. Computing the conditional expectation of the truncated normal, given a particular observation $y_{i',l'}$, we yield

$$U(p(\beta_r|y_{i',l'})) \approx \left\| \frac{1}{S-1} \sum_{s=2}^{S} {Z_r^{s-1}}^\top \left[A_{i',l'}\gamma^s - \Sigma_{(i',l')} Z_{l'}^{s-1} x_{i'} \left[\mu_{i',l'}^{s-1} + E\left(y_{i',l'} \frac{\phi(\mu_{i',l'}^{s-1})}{y_{i',l'}\Phi(\mu_{i',l'}^{s-1})} \right) \right] \right] \right\|_2, \tag{3.25}$$

where we let $\mu_{i',l'}^{s-1} = {\gamma^{s-1}}^\top Z_{l'}^{s-1} x_{i'}$. The final step is to compute the expectation over possible observations $y_{i',l'}$. For this we estimate the predictive probability $p(y_{i',l'}|X, Y)$ by integrating over the samples γ^s, Z^s collected from the Gibbs sampler.

Chapter 4

Heavy-Tailed Process Priors for Selective Shrinkage

The previous chapters considered active learning strategies to grow a small dataset that was tailored to inferring particular statistical quantities. In this chapter we will begin to consider randomized subsampling and its effects on statistical models. Randomized subsampling can be used both as an algorithm and as a theoretical framework: When vast amounts of data are available a random subsample can be used to make training more tractable. This subsampling is not necessary when only small amounts of data are available to start with. However, in that case we can still treat the data as if it were a random subsample from a larger dataset and perform theoretical analyses on it. In this chapter we are interested in the former usecase of randomized subsampling. We will exhibit situations where subsampling makes it difficult to learn from "outliers". The specific example we will focus on comes from computational biology, where distributions over rare angles in a proteins (so-called rotamers) are a key ingredient in, for example, the Rosetta energy function which is used during the design of proteins. Subsampling a large dataset of rotamers uniformly at random makes these rare examples even rarer, which leads many estimators to overfit them. One well-known model that suffers from this deficiency is Gaussian process classification. In this chapter we will present a technique for controlling the tendency of Gaussian process classification to overfit.

4.1 Overfitting

Gaussian process classifiers (GPCs) [63] provide a Bayesian approach to nonparametric classification with the key advantage of producing predictive class probabilities. Unfortunately, when training data are unevenly sampled in input space, GPCs tend to overfit in the sparsely populated regions. Our work is motivated by an application to protein folding where this presents a major difficulty. In particular, while Nature provides samples of protein configurations near the global minima of free energy functions, protein-folding algorithms, which imitate Nature by minimizing an estimated energy function, necessarily explore regions far

from the minimum. If the estimate of free energy is poor in those sparsely-sampled regions then the algorithm has a poor guide towards the minimum. More generally this problem can be viewed as one of "covariate shift," where the sampling pattern differs in the training and testing phase.

In this paper we investigate a GPC-based approach that addresses overfitting by shrinking predictive class probabilities towards conservative values. For an unevenly sampled input space it is natural to consider a *selective shrinkage* strategy: We wish to shrink probability estimates more strongly in sparse regions than in dense regions. To this end several approaches could be considered. If sparse regions can be readily identified, selective shrinkage could be induced by tailoring the Gaussian process (GP) kernel to reflect that information. In the absence of such knowledge, Goldberg and Williams [37] showed that Gaussian process regression (GPR) can be augmented with a GP on the log noise level. More recent work has focused on partitioning input space into discrete regions and defining different kernel functions on each. Treed Gaussian process regression [39] and Treed Gaussian process classification [12] represent advanced variations of this theme that define a prior distribution over partitions and their respective kernel hyperparameters. Another line of research which could be adapted to this problem posits that the covariate space is a nonlinear deformation of another space on which a Gaussian process prior is placed [18, 67]. Instead of directly modifying the kernel matrix, the observed non-uniformity of measurements is interpreted as being caused by the spatial deformation. A difficulty with all these approaches is that posterior inference is based on MCMC, which can be overly slow for the large-scale problems that we aim to address.

This paper shows that selective shrinkage can be more elegantly introduced by replacing the Gaussian process underlying GPC with a stochastic process that has heavy-tailed marginals (e.g., Laplace, hyperbolic secant, or Student-t). While heavy-tailed marginals are generally viewed as providing robustness to outliers in the *output space* (i.e., the response space), selective shrinkage can be viewed as a form of robustness to outliers in the *input space* (i.e., the covariate space). Indeed, selective shrinkage means the data points that are far from other data points in the input space are regularized more strongly. We provide a theoretical analysis and empirical results to show that inference based on stochastic processes with heavy-tailed marginals yields precisely this kind of shrinkage.

The paper is structured as follows: Section 4.2 provides background on GPCs and highlights how selective shrinkage can arise. We present a construction of heavy-tailed processes in Section 4.3 and show that inference reduces to standard computations in a Gaussian process. An analysis of our approach is presented in Section 4.4 and details on inference algorithms are presented in Section 4.5. Experiments on biological data in Section 4.6 demonstrate that heavy-tailed process classification substantially outperforms GPC in sparse regions while performing competitively in dense regions. The paper concludes with an overview of related research and final remarks in Sections 4.7 and 4.8.

4.2 Gaussian Process Classification and Shrinkage

A Gaussian process (GP) [63] is a prior on functions $z : \mathcal{X} \to \mathbb{R}$ defined through a mean function (usually identically zero) and a symmetric positive semidefinite kernel $k(\cdot, \cdot)$. For a finite set of locations $X = (x_1, \ldots, x_n)$ we write $z(X) \sim p(z(X)) = \mathcal{N}(0, K(X, X))$ as a random variable distributed according to the GP with finite-dimensional kernel matrix $[K(X, X)]_{i,j} = k(x_i, x_j)$. Let y denote an n-vector of binary class labels associated with measurement locations X[1]. We will refer to the label for covariate x_i as $y(x_i)$. For Gaussian process classification (GPC) [63] the probability that a test point x_* is labeled as class $y(x_*) = 1$, given training data (X, y), is computed as

$$p(y(x_*) = 1|X, y, x_*) = \mathbb{E}_{p(z(x_*)|X,y,x_*)} \left(\frac{1}{1 + \exp\{-z(x_*)\}} \right) \tag{4.1}$$
$$p(z(x_*)|X, y, x_*) = \int p(z(x_*)|X, z(X), x_*) p(z(X)|X, y) dz(X).$$

The predictive distribution $p(z(x_*)|X, y, x_*)$ represents a regression on $z(x_*)$ with a complicated observation model $y|z$. The central observation from Eq. (4.1) is that we could selectively shrink the prediction $p(y(x_*) = 1|X, y, x_*)$ towards a conservative value $1/2$ by selectively shrinking $p(z(x_*)|X, y, x_*)$ closer to a point mass at zero.

4.3 Heavy-Tailed Processes via the Gaussian Copula

In this section we construct the heavy-tailed stochastic process by transforming a GP. As with the GP, we will treat the new process as a prior on functions. Suppose that $\text{diag}\,(K(X, X)) = \sigma^2 \mathbf{1}$. We define the heavy-tailed process $f(X)$ with marginal c.d.f. G_b as

$$z(X) \sim \mathcal{N}(0, K(X, X)) \tag{4.2}$$
$$u(X) = \Phi_{0,\sigma^2}(z(X)) \tag{4.3}$$
$$f(X) = G_b^{-1}(u(X)) = G_b^{-1}(\Phi_{0,\sigma^2}(z(X))).$$

Here the function $\Phi_{0,\sigma^2}(\cdot)$ is the c.d.f. of a centered Gaussian with variance σ^2. Presently, we only consider the case when G_b is the (continuous) c.d.f. of a heavy-tailed density g_b with scale parameter b that is symmetric about the origin. Examples include the Laplace, hyperbolic secant and Student-t distribution. We note that other authors have considered asymmetric or even discrete distributions [16, 60, 76] while Snelson et al. [74] use arbitrary monotonic transformations in place of $G_b^{-1}(\Phi_{0,\sigma^2}(\cdot))$. The process $u(X)$ has the density of a Gaussian copula [57, 76] and is critical in transferring the correlation structure encoded by $K(X, X)$ from $z(X)$ to $f(X)$. If we define $z(f(X)) = \Phi_{0,\sigma^2}^{-1}(G_b(f(X)))$, it is well known [45,

[1]To improve the clarity of exposition, we only deal with binary classification for now. A full multiclass classification model is used in our experiments.

52, 60, 74, 76] that the density of $f(X)$ satisfies

$$p(f(X)) = \frac{\prod_{i=1} g_b(f(x_i))}{|K(X,X)/\sigma^2|^{1/2}} \exp\left\{-\frac{1}{2} z(f(X))^\top \left[K(X,X)^{-1} - \frac{I}{\sigma^2}\right] z(f(X))\right\}. \tag{4.4}$$

Observe that if $K(X,X) = \sigma^2 I$ then $p(f(X)) = \prod_{i=1} g_b(f(x_i))$. Also note that if G_b were chosen to be Gaussian, we would recover the Gaussian process. The predictive distribution $p(f(x_*)|X, f(X), x_*)$ can be interpreted as a Heavy-tailed process regression (HPR). It is easy to see that its computation can be reduced to standard computations in a Gaussian model by nonlinearly transforming observations $f(X)$ into z-space. The predictive distribution in z-space satisfies

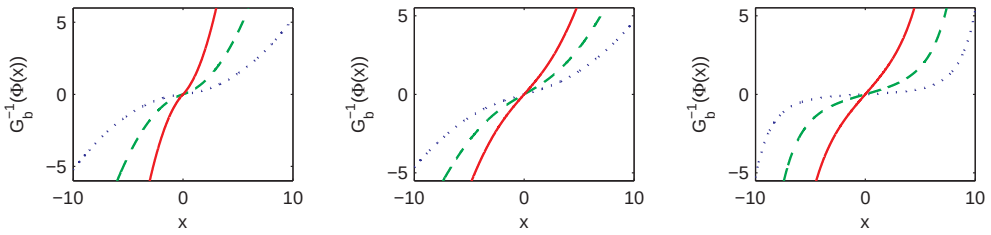
$$p(z(x_*)|X, f(X), x_*) = \mathcal{N}(\mu_*, \Sigma_*) \tag{4.5}$$
$$\mu_* = K(x_*, X)K(X,X)^{-1} z(f(X)) \tag{4.6}$$
$$\Sigma_* = K(x_*, x_*) - K(x_*, X)K(X,X)^{-1}K(X, x_*). \tag{4.7}$$

The corresponding distribution in f-space follows by another change of variables. Having defined the heavy-tailed stochastic process in general we now turn to an analysis of its shrinkage properties.

4.4 Selective Shrinkage

By "selective shrinkage" we mean that the degree of shrinkage applied to a collection of estimators varies across estimators. As motivated in Section 4.2, we are specifically interested in selectively shrinking posterior distributions near isolated observations more strongly than in dense regions. This section shows that we can achieve this by changing the *form* of prior marginals (heavy-tailed instead of Gaussian) and that this induces stronger selective shrinkage than any GPR could induce. Since HPR uses a GP in its construction, which can induce some selective shrinkage on its own, care must be taken to investigate only the additional benefits the transformation $G_b^{-1}(\Phi_{0,\sigma^2}(\cdot))$ has on shrinkage. For this reason we assume a particular GP prior which leads to a special type of shrinkage in GPR and then check how an HPR model built on top of that GP changes the observed behavior.

In this section we provide an idealized analysis that allows us to compare the selective shrinkage obtained by GPR and HPR. Note that we focus on regression in this section so that we can obtain analytical results. We work with n measurement locations, $X = (x_1, \ldots, x_n)$, whose index set $\{1, \ldots, n\}$ can be partitioned into a "dense" set D with $|D| = n - 1$ and a single "sparse" index $s \notin D$. Assume that $x_d = x_{d'}, \forall d, d' \in D$, so that we may let (without loss of generality) $\tilde{K}(x_d, x_{d'}) = 1, \forall d, d' \in D$. We also assert that $x_d \neq x_s \ \forall d \in D$ and let $\tilde{K}(x_d, x_s) = \tilde{K}(x_s, x_d) = 0 \ \forall d \in D$. Assuming that $n > 2$ we fix the remaining entry $\tilde{K}(x_s, x_s) = \epsilon/(\epsilon + n - 2)$, for some $\epsilon > 0$. We interpret ϵ as a noise variance and let $K = \tilde{K} + \epsilon I$.



(a) $g_b(x) = \frac{1}{2b}\exp\left\{-\frac{|x|}{b}\right\}$ (b) $g_b(x) = \frac{1}{2b}\mathrm{sech}\left(\frac{\pi x}{2b}\right)$ (c) $g_b(x) = \frac{1}{b\left(2+(x/b)^2\right)^{3/2}}$

Figure 4.1: Illustration of $G_b^{-1}(\Phi_{0,\sigma^2}(x))$, for $\sigma^2 = 1.0$ with G_b the c.d.f. of 4.1(a) the Laplace distribution 4.1(b) the Hyperbolic secant distribution 4.1(c) a Student-t inspired distribution, all with scale parameter b. Each plot shows samples—dotted, dashed, solid—for growing b. As b grows distributions become heavy-tailed and the gradient of $G_b^{-1}(\Phi_{0,\sigma^2}(x))$ increases.

Denote any distributions computed under the GPR model by $p_{\mathrm{gp}}(\cdot)$ and those computed in HPR by $p_{\mathrm{hp}}(\cdot)$. Using $K(X,X) = K$, define $z(X)$ as in Eq. (4.2). Let y denote a vector of real-valued measurements for a regression task. The posterior distribution of $z(x_i)$ given y, with $x_i \in X$, is derived by standard Gaussian computations as

$$\begin{aligned} p_{\mathrm{gp}}(z(x_i)|X,y) &= \mathcal{N}\left(\mu_i, \sigma_i^2\right) \\ \mu_i &= \tilde{K}(x_i,X)K(X,X)^{-1}y \\ \sigma_i^2 &= K(x_i,x_i) - \tilde{K}(x_i,X)K(X,X)^{-1}\tilde{K}(X,x_i). \end{aligned}$$

For our choice of $K(X,X)$ one can show that $\sigma_d^2 = \sigma_s^2$ for $d \in D$. To ensure that the posterior distributions agree at the two locations we require $\mu_d = \mu_s$, which holds if y satisfies

$$y \in \mathcal{Y}_{\mathrm{gp}} \triangleq \left\{y \middle| \left(\tilde{K}(x_d,X) - \tilde{K}(x_s,X)\right) K(X,X)^{-1}y = 0\right\} = \left\{y \middle| \sum_{d\in D} y(x_d) = y(x_s)\right\}.$$

A similar analysis can be carried out for the induced HPR model. By Eqs. (4.5)–(4.7) HPR inference leads to identical distributions $p_{\mathrm{hp}}(z(x_d)|X,y') = p_{\mathrm{hp}}(z(x_s)|X,y')$ with $d \in D$ if measurements y' in f-space satisfy

$$\begin{aligned} y' \in \mathcal{Y}_{\mathrm{hp}} &\triangleq \left\{y' \middle| \left(\tilde{K}(x_d,X) - \tilde{K}(x_s,X)\right) K(X,X)^{-1}\Phi_{0,\sigma^2}^{-1}(G_b(y')) = 0\right\} \\ &= \left\{y' = G_b^{-1}(\Phi_{0,\sigma^2}(y)) | y \in \mathcal{Y}_{\mathrm{gp}}\right\}. \end{aligned}$$

To compare the shrinkage properties of GPR and HPR we analyze select pairs of measurements in $\mathcal{Y}_{\mathrm{gp}}$ and $\mathcal{Y}_{\mathrm{hp}}$. The derivation requires that $G_b^{-1}(\Phi_{0,\sigma^2}(\cdot))$ is strongly concave on $(-\infty, 0]$, strongly convex on $[0, +\infty)$ and has gradient > 1 on $\mathbb{R}$. To see intuitively why this should hold, note that for G_b with fatter tails than a Gaussian, $|G_b^{-1}(\Phi_{0,\sigma^2}(x))|$ should eventually dominate $|\Phi_{0,b^2}^{-1}(\Phi_{0,\sigma^2}(x))| = (b/\sigma)|x|$. Figure 4.1 demonstrates graphically that

the assumption holds for several choices of G_b, provided b is large enough, i.e., that g_b has sufficiently heavy tails. Indeed, it can be shown that for scale parameters $b > 0$, the first and second derivatives of $G_b^{-1}(\Phi_{0,\sigma^2}(\cdot))$ scale linearly with b. Consider a measurement $0 \neq y \in \mathcal{Y}_{\text{gp}}$ with $\text{sign}\left(y(x_d)\right) = \text{sign}\left(y(x_{d'})\right), \forall d, d' \in D$. Analyzing such y is relevant, as we are most interested in comparing how multiple *reinforcing* observations at clustered locations and a single isolated observation are absorbed during inference. By definition of $\mathcal{Y}_{\text{gp}}$, for $d^* = \text{argmax}_{d \in D}|y(x_d)|$ we have $|y(x_{d^*})| < |y(x_s)|$ as long as $n > 2$. The corresponding element $y' = G_b^{-1}(\Phi_{0,\sigma^2}(y)) \in \mathcal{Y}_{\text{hp}}$ then satisfies

$$|y'(x_s)| = \left|G_b^{-1}(\Phi_{0,\sigma^2}(y(x_s)))\right| > \left|\frac{G_b^{-1}(\Phi_{0,\sigma^2}(y(x_{d^*})))}{y(x_{d^*})} y(x_s)\right| = \left|\frac{y'(x_{d^*})}{y(x_{d^*})} y(x_s)\right|. \tag{4.8}$$

Thus HPR inference leads to identical predictive distributions in f-space at the two locations even though the isolated observation $y'(x_s)$ has disproportionately larger magnitude than $y'(x_{d^*})$, relative to the GPR measurements $y(x_s)$ and $y(x_{d^*})$. As this statement holds for any $y \in \mathcal{Y}_{\text{gp}}$ satisfying our earlier sign requirement, it indicates that HPR systematically shrinks isolated observations more strongly than GPR. Since the second derivative of $G_b^{-1}(\Phi_{0,\sigma^2}(\cdot))$ scales linearly with scale $b > 0$, an intuitive connection suggests itself when looking at inequality (4.8): The heavier the marginal tails, the stronger the inequality and thus the stronger the selective shrinkage effect.

The previous derivation exemplifies in an idealized setting that HPR leads to improved shrinkage of predictive distributions near isolated observations. More generally, because GPR transforms measurements only linearly, while HPR additionally pre-transforms measurements nonlinearly, our analysis suggests that for any GPR we can find an HPR model which leads to stronger selective shrinkage. The result has intuitive parallels to the parametric case: Just as ℓ_1-regularization improves shrinkage of parametric estimators, heavy-tailed processes improve shrinkage of nonparametric estimators. We note that although our analysis kept $K(X, X)$ fixed for GPR and HPR, in practice we are free to tune the kernel to yield a desired scale of predictive distributions. The above analysis has been carried out for regression, but motivates us to now explore heavy-tailed processes in the classification case.

4.5 Heavy-Tailed Process Classification

The derivation of *heavy-tailed process classification* (HPC) is similar to that of standard multiclass GPC with Laplace approximation in Rasmussen and Williams [63]. However, due to the nonlinear transformations involved, some nice properties of their derivation are lost. We revert notation and let y denote a vector of class labels. For a C-class classification problem with n training points we introduce a vector of nC latent function measurements $(f^1(x_1), \ldots, f^1(x_n), f^2(x_1), \ldots, f^2(x_n), \ldots, f^C(x_1), \ldots, f^C(x_n))^\top$. For each block $c \in \{1, \ldots, C\}$ of n variables we define an independent heavy-tailed process prior using Eq. (4.4) with kernel matrix K_c. Equivalently, we can define the prior jointly on f by letting K be a block-diagonal kernel matrix with blocks $K_1, \ldots, K_C$. Each kernel matrix K_c

is defined by a (possibly different) symmetric positive semidefinite kernel with its own set of parameters. The following construction relaxes the earlier condition that $\operatorname{diag}(K) = \sigma^2\mathbf{1}$ and instead views $\Phi_{0,\sigma^2}(\cdot)$ as some nonlinear transformation with parameter σ^2. By this relaxation we effectively adopt Liu et al.'s [52] interpretation that Eq. (4.4) defines the copula. The scale parameters b could in principle vary across the nC variables, but we keep them constant at least within each block of n. Labels y are represented in a 1-of-n form and generated by the following observation model

$$p(y(x_i)^c = 1 | f(x_i)) = \pi_i^c = \frac{\exp\{f^c(x_i)\}}{\sum_{c'} \exp\{f^{c'}(x_i)\}}. \tag{4.9}$$

For inference we are ultimately interested in computing

$$p(y(x_*)^c = 1 | X, y, x_*) = \mathbb{E}_{p(f(x_*)|X,y,x_*)} \left(\frac{\exp\{f^c(x_*)\}}{\sum_{c'} \exp\{f^{c'}(x_*)\}} \right), \tag{4.10}$$

where $f(x_*) = (f^1(x_*), \ldots, f^C(x_*))^\top$. The previous section motivates that improved selective shrinkage will occur in $p(f(x_*)|X, y, x_*)$ if the prior marginals have sufficiently heavy tails.

Inference

As in GPC, most of the intractability lies in computing the distribution $p(f(x_*)|X, y, x_*)$. We use the Laplace approximation to address this issue. A Gaussian approximation to $p(z|X, y)$ is found and then combined with the Gaussian $p(z(x_*)|X, z, x_*)$ to give us an approximation to $p(z(x_*)|X, y, x_*)$. This is then transformed to a (typically non-Gaussian) distribution in f-space using a change of variables. Hence we first seek to find a mode and corresponding Hessian matrix of the log posterior $\log p(z|X, y)$. Recalling the relation $f = G_b^{-1}(\Phi_{0,\sigma^2}(z))$, the log posterior can be written as

$$J(z) \triangleq \log p(y|z) + \log p(z) = y^\top f - \sum_i \log \sum_c \exp\{f^c(x_i))\} - \frac{1}{2} z^\top K^{-1} z - \frac{1}{2} \log |K| + \text{const.}$$

Let Π be an $nC \times n$ matrix of stacked diagonal matrices $\operatorname{diag}(\pi^c)$ for n-subvectors π^c of π. With $W = \operatorname{diag}(\pi) - \Pi\Pi^\top$, the gradients are

$$\begin{aligned} \nabla J(z) &= \operatorname{diag}\left(\frac{df}{dz}\right)(y - \pi) - K^{-1} z \\ \nabla^2 J(z) &= \operatorname{diag}\left(\frac{d^2 f}{dz^2}\right) \operatorname{diag}(y - \pi) - \operatorname{diag}\left(\frac{df}{dz}\right) W \operatorname{diag}\left(\frac{df}{dz}\right) - K^{-1}. \end{aligned}$$

Unlike in Rasmussen and Williams [63], $-\nabla^2 J(z)$ is not generally positive definite owing to its first term. For that reason we cannot use a Newton step to find the mode and instead

resort to a simpler gradient method. Once the mode $\hat{z}$ has been found we approximate the posterior as

$$p(z|X,y) \approx q(z|X,y) = \mathcal{N}\left(\hat{z}, -\nabla^2 J(\hat{z})^{-1}\right),$$

and use this to approximate the predictive distribution by

$$q(z(x_*)|X,y,x_*) = \int p(z(x_*)|X,z,x_*)q(z|X,y)df.$$

Since we arranged for both distributions in the integral to be Gaussian, the resulting Gaussian can be straightforwardly evaluated. Finally, to approximate the one-dimensional integral with respect to $p(f(x_*)|X,y,x_*)$ in Eq. (4.10) we could either use a quadrature method, or generate samples from $q(z(x_*)|X,y,x_*)$, convert them to f-space using $G_b^{-1}(\Phi_{0,\sigma^2}(\cdot))$ and then approximate the expectation by an average. We have compared predictions of the latter method with those of a Gibbs sampler; the Laplace approximation matched Gibbs results well, while being much faster to compute.

Parameter estimation

Using a derivation similar to that in [63], we have for $\hat{f} = G_b^{-1}(\Phi_{0,\sigma^2}(\hat{z}))$ that the Laplace approximation of the marginal log likelihood is

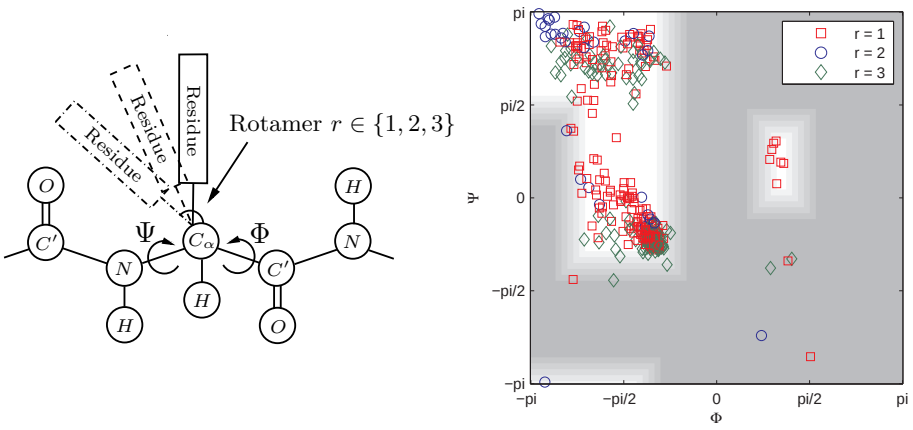
$$\begin{aligned}\log p(y|x) \approx \log q(y|x) &= J(\hat{z}) - \frac{1}{2}\log|-2\pi\nabla^2 J(\hat{z})| \\ &= y^\top \hat{f} - \sum_i \log \sum_c \exp\left\{\hat{f}^c(x_i)\right\} - \frac{1}{2}\hat{z}^\top K^{-1}\hat{z} \\ &\quad - \frac{1}{2}\log|K| - \frac{1}{2}\log|-\nabla^2 J(\hat{z})| + \text{const.}\end{aligned} \tag{4.11}$$

We optimize kernel parameters θ by taking gradient steps on $\log q(y|x)$. The derivative needs to take into account that perturbing the parameters can also perturb the mode $\hat{z}$ found for the Laplace approximation. At an optimum $\nabla J(\hat{z})$ must be zero, so that

$$\hat{z} = K\text{diag}\left(\frac{d\hat{f}}{d\hat{z}}\right)(y - \hat{\pi}), \tag{4.12}$$

where $\hat{\pi}$ is defined as in Eq. (4.9) but using $\hat{f}$ rather than f. Taking derivatives of this equation allows us to compute the gradient $d\hat{z}/d\theta$. Differentiating the marginal likelihood we have

$$\begin{aligned}\frac{d\log q(y|x)}{d\theta} = (y-\hat{\pi})^\top \text{diag}\left(\frac{d\hat{f}}{d\hat{z}}\right)\frac{d\hat{z}}{d\theta} - \frac{d\hat{z}}{d\theta}K^{-1}\hat{z} + \frac{1}{2}\hat{z}^\top K^{-1}\frac{dK}{d\theta}K^{-1}\hat{z} - \\ \frac{1}{2}\text{tr}\left(K^{-1}\frac{dK}{d\theta}\right) - \frac{1}{2}\text{tr}\left(\nabla^2 J(\hat{z})^{-1}\frac{d\nabla^2 J(\hat{z})}{d\theta}\right).\end{aligned}$$



(a) (b)

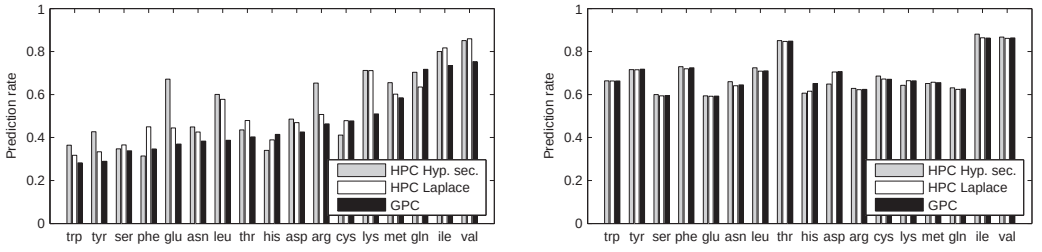
Figure 4.2: 4.2(a) Schematic of a protein segment. The backbone is the sequence of C', N, C_α, C', N atoms. An amino-acid-specific sidechain extends from the C_α atom at one of three discrete angles known as "rotamers." 4.2(b) Ramachandran plot of 400 (Φ, Ψ) measurements and corresponding rotamers (by shapes/colors) for amino-acid arginine (arg). The dark shading indicates the sparse region we considered in producing results in Figure 4.3. Progressively lighter shadings indicate how the sparse region was grown to produce Figure 4.4.

The remaining gradient computations are straightforward, albeit tedious. In addition to optimizing the kernel parameters, it may also be of interest to optimize the scale parameter b of marginals G_b. Again, differentiating Eq. (4.12) with respect to b allows us to compute $d\hat{z}/db$. We note that when perturbing b we change $\hat{f}$ by changing the underlying mode $\hat{z}$ as well as by changing the parameter b which is used to compute $\hat{f}$ from $\hat{z}$. Suppressing the detailed computations, the derivative of the marginal log likelihood with respect to b is

$$\frac{d \log q(y|x)}{db} = (y - \hat{\pi})^\top \frac{d\hat{f}}{db} - \frac{d\hat{z}}{db}^\top K^{-1}\hat{z} - \frac{1}{2}\mathrm{tr}\left(\nabla^2 J(\hat{z})^{-1} \frac{d\nabla^2 J(\hat{z})}{db}\right).$$

4.6 Experiments

To a first approximation, the three-dimensional structure of a folded protein is defined by pairs of continuous backbone angles (Φ, Ψ), one pair for each amino-acid, as well as discrete angles, so-called rotamers, that define the conformations of the amino-acid sidechains that extend from the backbone. The geometry is outlined in Figure 4.2(a). There is a strong dependence between backbone angles (Φ, Ψ) and rotamer values; this is illustrated in the "Ramachandran plot" shown in Figure 4.2(b), which plots the backbone angles for each rotamer (indicated by the shapes/colors). The dependence is exploited in computational approaches to protein structure prediction, where estimates of rotamer probabilities given



(a) (b)

Figure 4.3: Rotamer prediction rates in percent in 4.3(a) sparse and 4.3(b) dense regions. Both flavors of HPC (hyperbolic secant and Laplace marginals) significantly outperform GPC in sparse regions while performing competitively in dense regions.

backbone angles are used as one term in an energy function that models native protein states as minima of the energy. Poor estimates of rotamer probabilities in sparse regions can derail the prediction procedure. Indeed, sparsity has been a serious problem in state-of-the-art rotamer models based on kernel density estimates (Roland Dunbrack, personal communication). Unfortunately, we have found that GPC is not immune to the sparsity problem. To evaluate our algorithm we consider rotamer-prediction tasks on the 17 amino-acids (out of 20) that have three rotamers at the first dihedral angle along the sidechain[2]. Our previous work thus applies with the number of classes $C = 3$ and the covariates being (Φ, Ψ) angle pairs. Since the input space is a torus we defined GPC and HPC using the following von Mises-inspired kernel for d-dimensional angular data:

$$k(x_i, x_j) = \sigma^2 \exp\left\{\lambda\left(\left(\sum_{k=1}^{d} \cos(x_{i,k} - x_{j,k})\right) - d\right)\right\},$$

where $x_{i,k}, x_{j,k} \in [0, 2\pi]$ and $\sigma^2, \lambda \geq 0$[3]. To find good GPC kernel parameters we optimize an ℓ_2-regularized version of the Laplace approximation to the log marginal likelihood reported in Eq. 3.44 of [63]. For HPC we let G_b be either the centered Laplace distribution or the hyperbolic secant distribution with scale parameter b. We estimate HPC kernel parameters as well as b by similarly maximizing an ℓ_2-regularized form of Eq. (4.11). In both cases we restricted the algorithms to training sets of only 100 datapoints. Since good regularization parameters for the objectives are not known a priori we train with and test them on a grid for each of the 17 rotameric residues in ten-fold cross-validation. To find good regularization parameters for a particular residue we look up that combination which, averaged over the ten folds of the remaining 16 residues, produced the best test results. Having chosen the regularization constants we report average test results computed in ten-fold cross validation.

[2]Residues alanine and glycine are non-discrete while proline has two rotamers at the first dihedral angle.

[3]The function $\cos(x_{i,k} - x_{j,k}) = [\cos(x_{i.k}), \sin(x_{i,k})][\cos(x_{j.k}), \sin(x_{j,k})]^\top$ is a symmetric positive semi-definite kernel. By Propositions 3.22 (i) and (ii) and Proposition 3.25 in Shawe-Taylor and Cristianini [70], so is $k(x_i, x_j)$ above.

We evaluate the algorithms on predefined sparse and dense regions in the Ramachandran plot, as indicated by the background shading in Figure 4.2(b). Across 17 residues the sparse regions usually contained more than 70 measurements (and often more than 150), each of which appears in one of the 10 cross validations. Figure 4.3 compares the label prediction rates on the dense and sparse regions. Averaged over all 17 residues HPC outperforms GPC by 5.79% with Laplace and 7.89% with hyperbolic secant marginals. With Laplace marginals HPC underperforms GPC on only two residues in sparse regions—by 8.22% on glutamine (gln), and by 2.53% on histidine (his). On dense regions HPC lies within 0.5% on 16 residues and only degrades once by 3.64% on his. Using hyperbolic secant marginals HPC often improves GPC by more than 10% on sparse regions and degrades by more than 5% only on cysteine (cys) and his. On dense regions HPC usually performs within 1.5% of GPC. In Figure 4.4 we show how the average rotamer prediction rate across 17 residues changes for HPC, GPC, as well as CTGP [12] as we grow the sparse region to include more measurements from dense regions. The growth of the sparse region is indicated by progressively lighter shadings in Figure 4.2(b). As more points are included the significant advantage of HPC lessens. Eventually GPC does marginally better than HPC and much better than CTGP. The values reported in Figure 4.3 correspond to the dark shaded region, which contains an average of 155 measurements per residue.

4.7 Related Research

Copulas [57] allow convenient modelling of multivariate correlation structures as separate from marginal distributions. Early work by Song [76] used the Gaussian copula to generate complex multivariate distributions by complementing a simple copula form with marginal distributions of choice. Popularity of the Gaussian copula in the financial literature is generally credited to Li [50] who used it to model correlation structure for pairs of random variables with known marginals. More recently, the Gaussian process has been modified in a similar way to ours by Snelson et al. [74]. They demonstrate that posterior distributions can better approximate the true noise distribution if the transformation defining the warped process is learned. Jaimungal and Ng [45] have extended this work to model multiple parallel time series with marginally non-Gaussian stochastic processes. Their work uses a "binding copula" to combine several subordinate copulas into a joint model. Bayesian approaches focusing on estimation of the Gaussian copula covariance matrix for a given dataset are given in [23, 60]. Research also focused on estimation in high-dimensional settings [52].

4.8 Conclusions

This paper analyzed learning scenarios where outliers are observed in the input space, rather than the output space as commonly discussed in the literature. We illustrated heavy-tailed processes as a straightforward extension of GPs and an economical way to improve the

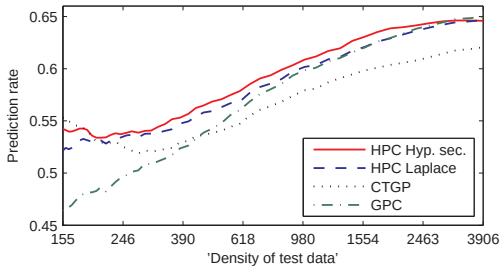


Figure 4.4: Average rotamer prediction rate in the sparse region for both flavors of HPC, standard GPC well as CTGP [12] as a function of the average number of points in the region.

robustness of estimators in sparse regions beyond those of GP-based methods. Importantly, because these processes are based on a GP, they inherit many of its favorable computational properties; predictive inference in regression, for instance, is straightforward. Moreover, because heavy-tailed processes have a parsimonious representation, they can be used as building blocks in more complicated models where currently GPs are used. In this way the benefits of heavy-tailed processes extend to any GP-based model that struggles with covariate shift.

Chapter 5

Efficient Ranking from Pairwise Comparisons

Ranking n objects from pairwise comparisons is a core machine learning problem, arising in recommender systems, ad placement, player ranking, biological applications and others. In many practical situations the true pairwise comparisons cannot be actively measured, but a subset of all $n(n-1)/2$ comparisons is passively and noisily observed. Optimization algorithms (e.g. the SVM) could be used to predict a ranking with fixed expected Kendall tau distance, while achieving an $\Omega(n)$ lower bound on the corresponding sample complexity. However, due to their centralized structure they are difficult to extend to online or distributed settings. In this paper we show that much simpler algorithms can match the same $\Omega(n)$ lower bound in expectation. Furthermore, if instead an average of $O(n \log(n))$ binary comparisons are measured, then one algorithm recovers the true ranking in a uniform sense, while the other predicts the ranking more accurately near the top than the bottom. We discuss extensions to online and distributed ranking, with benefits over traditional alternatives.

5.1 Introduction

Ranking from binary comparisons is a ubiquitous problem in modern machine learning applications. Given a set of n objects and set of (possibly inconsistent) binary comparisons between pairs of objects (such as "player i won against player j," or "the customer bought book i instead of j"), the task is to infer a total order over objects that aggregates the given measurements. Common settings for this problem allow binary comparisons to be measured either actively [4, 5, 46, 11, 35], repeatedly [56, 6, 28], or assume that all $n(n-1)/2$ comparisons are known up to some noise [10, 11]. We believe that in many challenging applications, these assumptions are unrealistic: (1) Active measurements are often infeasible, either because measurements must be made passively (e.g., from click-through data, purchasing preferences), or because pairwise comparisons are too time consuming to measure in series (e.g., measuring protein-protein interactions). (2) Repeated measurements are not

practical if comparisons are derived from the outcomes of sports games or the purchasing behavior of a customer (a customer typically wants to purchase a product only once). (3) The $O(n^2)$ growth of comparisons between n objects usually prohibits exhaustive measuring when n is large.

Since a total order can be uniquely determined by sorting distinct object "scores," it is common to formalize the problem as follows: Given a subset of (possibly noisy) binary comparisons $\bar{c}_{i,j}$ between n objects, we desire a scoring function $\hat{\Pi} : \{1, \ldots, n\} \to \mathbb{R}$ so that $\bar{c}_{i,j} = 1 \iff \hat{\Pi}(i) < \hat{\Pi}(j)$ for as many examples in the training data as possible. Traditional optimization losses targeting this objective are intuitive (e.g., count the number of inversions between the training data and the scoring function,) but discontinuous and non-convex. The substantial literature on learning to rank can be specialized to this setting by learning scoring functions that only depend on the object identity. This approach suggests ways to approximately solve the optimization problem by relaxing the intractable loss to convex surrogates [19, 31, 42, 48]. Although some of these methods (e.g., the SVM) can achieve an $\Omega(n)$ lower bound on a certain sample complexity, we feel that optimization-based approaches may be unnecessarily complex in this situation. The question arises whether simpler algorithms could be equally effective. In this paper we demonstrate that two very simple algorithms achieve the same $\Omega(n)$ lower bound without solving an explicit optimization problem. Furthermore, given slightly more measurements, we can show interesting differences between the two algorithms: The first predicts rankings with approximately uniform quality across the ranking, while the second predicts the true ranking with higher quality near the top of the ranking than the bottom. Additionally, we view the simple form of the algorithms as a significant asset which makes them much easier to extend. As a demonstration, we discuss extensions to online and distributed learning, and highlight important benefits over traditional alternatives.

The paper is organized as follows: We first introduce some notation and quality measures in Section 5.2. In Section 5.3 we discuss related research and background. Section 5.4 presents two simple ranking algorithms and analyzes their performance in terms of the expected Kendall tau distance as well as high probability bounds on rank displacements. In Section 5.5 we evaluate and validate our theoretical findings. We touch on extensions to online and distributed ranking in Section 5.6, before concluding with final thoughts in Section 5.7. The complete proofs for all propositions, lemmas and theorems are collected in the supplementary material.

5.2 Preliminaries

Throughout the paper we denote the true permutation we wish to recover by $\pi^* \in S_n$. We use the notation $\pi(i)$ to indicate the position of object i in permutation π. Without loss of generality, let $\pi^* = (1, 2, \ldots, n)$, so that $\pi^*(j) = j$. We will reveal to an algorithm a subset of binary comparisons, chosen among the $n(n-1)/2$ available pairs. Specifically, each comparison is measured independently with probability $m(n)/n$, so that on average

$O(nm(n))$ measurements are made. Each comparison can be measured only once (i.e. we measure *without* replacement)[1]. The function $m(n)$ is a key quantity; we will characterize various sample complexities in terms of bounds on its growth. For some results we will find that $m(n) \in \Theta(1)$ suffices, while in others we need $m(n) \in \Theta(\log(n))$. We will always assume that $m(n) \in o(n)$. Noiseless binary comparisons are denoted by $c_{i,j} = \mathbf{1}(\pi^*(i) < \pi^*(j))$. A common observation model is to assume that each binary comparison is independently flipped with probability $1-p$, where $p > 1/2$ [10, 28]. To capture the overall measurement process, we introduce binary variables $s_{i,j}$ which indicate whether $c_{i,j}$ was measured, and let $\bar{c}_{i,j}$ be the (possibly noisy) measurement that was made. We will assume throughout that $s_{j,i} = s_{i,j}$ and if $s_{j,i} = s_{i,j} = 1$, then $\bar{c}_{j,i} = 1 - \bar{c}_{i,j}$.

In this paper we analyze the quality of the proposed algorithms in two ways. The first counts the number of inverted binary comparisons of the predicted permutation $\hat{\pi}$ relative to π^*. That is, we use the loss

$$\mathrm{inv}(\hat{\pi}) = \sum_{\pi^*(i)<\pi^*(j)} \mathbf{1}(\hat{\pi}(j) < \hat{\pi}(i)). \tag{5.1}$$

This quantity is also known as the *Kendall tau distance*. Using results in Fulman [33], one can show that if $\hat{\pi}$ is chosen uniformly at random in S_n, then $\mathrm{inv}(\hat{\pi})$ concentrates around $(1/2)(n(n-1)/2)$. To be interesting we will thus require our algorithms to have expected risk $\mathbb{E}(\mathrm{inv}(\hat{\pi})) \leq (\eta/2)(n(n-1)/2)$ for some $0 < \eta < 1$. We note that another common comparison metric is Spearman's footrule

$$\mathrm{dis}(\hat{\pi}) = \sum_{j=1}^{n} |\hat{\pi}(j) - \pi^*(j)|. \tag{5.2}$$

As shown in Diaconis and Graham [22], $\mathrm{inv}(\hat{\pi})$ is related to $\mathrm{dis}(\hat{\pi})$ as $\mathrm{inv}(\hat{\pi}) \leq \mathrm{dis}(\hat{\pi}) \leq 2\mathrm{inv}(\hat{\pi})$. Our results on the expected Kendall tau distance thus directly transfer to Spearman's footrule. We also analyze the prediction $\hat{\pi}$ by how far individual objects are displaced relative to π^*. When appropriate, we will bound the largest displacement

$$\max_j |\hat{\pi}(j) - \pi^*(j)|. \tag{5.3}$$

However, in some cases the recovery is not uniform, warranting a detailed inspection of the set of displacements $\{|\hat{\pi}(j) - \pi^*(j)| : j = 1, \ldots, n\}$.

5.3 Related Research

Several threads of research aim to give various sample complexities in the active ranking setting. Ailon et al. [4], for example, give an active algorithm which produces a permutation

[1] In some other analyses measurements are made independently with replacement [61, 55].

with small loss relative to the optimal loss (which may be zero). This result was refined by Ailon et al. [5] to show that if the true scoring function is linear, one can find a scoring function with small loss (relative to the optimal loss) using $O(n \log^4(n))$ active queries. Braverman and Mossel [11] give an active algorithm with query complexity $O(n \log(n))$ for noisy binary comparisons that produces a ranking in time that is with high probability polynomial. Agarwal [2] has developed a comprehensive theory for bipartite ranking. Here, instead of receiving binary comparisons, we receive binary labels (e.g., relevant/irrelevant) for each object, and the goal is a function which orders negative before positive examples.

A number of recent papers have analyzed lower bounds for the demanding task of *exact* score recovery. Jamieson and Nowak [46], for example, consider the case when the true scoring function reflects the Euclidean distance of object covariates from a global reference point. If objects are embedded in $\mathbb{R}^d$, then any algorithm that exactly identifies the true ranking must sample at least $O(d \log(n))$ comparisons. While this bound can be achieved by an active algorithm, any algorithm that uses only random measurements must see almost all pairwise comparisons in order to exactly predict the true ranking. Gleich and Lim [36] suppose that the true score differences (i.e., $\Pi^*(j) - \Pi^*(i)$, or functions thereof) can be measured. Given an incomplete matrix of such measurements they use low rank matrix completion to estimate the true object scores. If the measurements are in fact score differences, their algorithm recovers the true scores with high probability exactly using between $O(n \log^2(n))$ and $O(n^2 \log^2(n))$ random measurements (depending on the shape of the true scores). Although their work considers random measurements, their theory does not apply when binary comparisons are measured in lieu of score differences. Mitliagkas et al. [55] focus on exactly recovering the preferences expressed by a population of r users. Each user's preferences are recorded by a permutation over objects, which can be queried (either actively or by random sampling) through pairwise comparisons between objects. The randomized sampling result is not helpful in our setting (where $r = 1$) since it then requires $O(n^2 \log(n))$ measurements (with replacement) for exact recovery.

SVM Ranking. It is well-known that the SVM could be used to learn a linear scoring function in the setting of Section 5.2: For each observed comparison $\bar{c}_{i,j}$, create a feature vector $x_{i,j} = e_j - e_i$ (where e_i is a binary indicator vector with a 1 at the i-th coordinate) and associate with it the label $\bar{y}_{i,j} = 2\bar{c}_{i,j} - 1$. Learning a scoring function now reduces to inferring a separating hyperplane w so that the function $\text{sign}(w^\top x_{i,j})$ best predicts the labels $\bar{y}_{i,j}$ on training data. The predicted permutation $\hat{\pi}$ follows from sorting the elements in w. Statistical learning theory shows that in the noiseless case ($p = 1$), the sample complexity for inferring a w which with high probability induces a Kendall tau distance of at most $(\eta/2)(n(n-1)/2)$ is small. Indeed, using results of Radinsky et al. [61] one can show the following proposition, which we prove in the supplementary material

Proposition 5.3.1. *There is a constant d, so that for any $0 < \eta < 1$, if we noiselessly measure dn/η^2 binary comparisons, chosen uniformly at random with replacement, and $n >$*

n_0 is large enough, the SVM will produce a prediction $\hat{\pi}$, which satisfies

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2}. \tag{5.4}$$

The proposition highlights that the SVM needs to sample $\Theta(n/\eta^2)$ examples *with* replacement for an expected risk of at most $(\eta/2)(n(n-1)/2)$. Some algebra then reveals that this amounts to an average of $O(n)$ distinct samples. As the following proposition, a summary of results of Giesen et al. [35], demonstrates, the sample complexity of Proposition 5.3.1 is tight up to constants.

Proposition 5.3.2. *For $\eta < 1$, any randomized, comparison-based algorithm that produces for all π^* a prediction $\hat{\pi}$ with an expected risk of*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2} \tag{5.5}$$

must on expectation use at least $\Omega(n)$ comparisons in the worst case.

The proposition is proved in the supplementary material for completeness. Although the SVM is effectively optimal in this setting, we feel that its direct application is overly heavy handed. The goal of this paper is to exhibit two much simpler algorithms which also achieve the above sample complexity, while being easier to extend to novel applications.

5.4 Two Simple Algorithms

In this section we present two simple rank estimators using the randomized data collection framework outlined in Section 5.2.

Balanced Rank Estimation

We begin this paper by analyzing BRE, which estimates an object's score as the relative difference of the number of items preceding and succeeding it.

Balanced Rank Estimation (BRE):
Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{\sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)}{2m(n)} \propto \sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1).$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

Our first result concerns the expected number of inversions of $\hat{\pi}$ relative to π^*.

Theorem 5.4.1. *For any $0 < \eta < 1$ there is a constant $c(p, \eta) \in \Theta(1/((2p-1)^2\eta^2))$ so that if $m(n)/n \geq c(p, \eta)/n$, and $n > n_0$ is large enough, BRE satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2}. \tag{5.6}$$

To give some intuition for this theorem, we briefly sketch the proof. Since we assumed $\pi^* = (1, \ldots, n)$, the expected Kendall tau distance is

$$\mathbb{E}(\text{inv}(\hat{\pi})) = \sum_{i<j} \mathbb{P}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right). \tag{5.7}$$

The score difference $\hat{\Pi}(i) - \hat{\Pi}(j)$ can be written as a sum of $2n-3$ independent random variables. By controlling their mean, variance and magnitude, if $n > n_0$ is large the following bound can be derived for $i < j$:

$$\mathbb{P}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right) \tag{5.8}$$

$$\leq \exp\left\{-\left[\frac{j-i}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\}. \tag{5.9}$$

Applying this to Eq. (5.7), we bound $\mathbb{E}(\text{inv}(\hat{\pi}))$ by

$$\sum_{k=1}^{n-1}(n-k)\exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} \tag{5.10}$$

$$\leq \int_0^n (n-k)\exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} dk \tag{5.11}$$

$$\leq \frac{n}{n-1}\sqrt{\frac{128}{3}}\frac{1}{(2p-1)\sqrt{m(n)}}\binom{n}{2}. \tag{5.12}$$

Matching this upper bound with the target quantity $(\eta/2)(n(n-1)/2)$, we find $m(n) \in \Theta(1/((2p-1)^2\eta^2))$.

In the noiseless case ($p = 1$), Theorem 5.4.1 guarantees that for any $0 < \eta < 1$, BRE in expectation has the same sample complexity as the SVM in Proposition 5.3.1. In particular, BRE also achieves the $\Omega(n)$ lower bound of Proposition 5.3.2. This may seem at first surprising. However, a similar algorithm was recently shown to have favorable properties in a different context [17].

More informative statements can be made if a slightly larger number measurements is available. As the following theorem shows, given an average of $\Theta(n\log(n))$ measurements, BRE predicts permutations with uniform quality across the entire permutation.

Theorem 5.4.2. *For any $c > 0$ and $0 < \nu < 1$, if each comparison is measured with probability $m(n)/n = c\log(n)/n$, then BRE predicts with probability at least $1 - 2n^{1-a_n \frac{3}{8}(2p-1)^2\nu^2 c}$ a permutation $\hat{\pi}$ with*

$$\max_j |\hat{\pi}(j) - \pi^*(j)| \leq \nu n, \tag{5.13}$$

where a_n is a sequence with $a_n \to 1$.

The crux of the argument is that the estimated scores $\hat{\Pi}(j)$ concentrate around their expectation $\tilde{\Pi}^*(j) \triangleq \mathbb{E}(\hat{\Pi}(j)) = aj/n + b$, where $a = (2p-1)$ and $b \in \mathbb{R}$ (as before we assume $\pi^* = (1, \ldots, n)$). If all scores concentrate uniformly well, they will reveal the true permutation up to local displacements. Using a similar analysis as in Theorem 5.4.1, our proof first establishes the following Bernstein concentration:

$$\mathbb{P}\left(\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > t\right) \tag{5.14}$$

$$\leq 2\exp\left\{-\frac{n}{n+m(n)}\frac{t^2 4m(n)}{2\left(1+\frac{2t}{3}\right)}\right\}, \tag{5.15}$$

to which we then apply a union bound (introducing the $\log(n)$ factor)

$$\mathbb{P}\left(\exists j : \left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > t\right) \tag{5.16}$$

$$\leq 2\exp\left\{-\frac{n}{n+m(n)}\frac{t^2 4m(n)}{2\left(1+\frac{2t}{3}\right)} + \log(n)\right\}. \tag{5.17}$$

Thus, the relative ordering of two objects that are far apart in the π^* (large t) should be harder to confuse than that of nearby objects (small t). Indeed, using the following intuitive lemma, the uniform concentration of scores translates into a uniform bound on displacements $|\hat{\pi}(j) - \pi^*(j)|$.

Lemma 5.4.3. *For any $a > 0$, and $b \in \mathbb{R}$, if $\forall j$, we have $|\hat{\Pi}(j) - (aj/n + b)| \leq t$, then we have that $\forall j$, $|\hat{\pi}(j) - \pi^*(j)| \leq 2tn/a$.*

The lemma applies to the union bound with $a = (2p-1)$. The proof is then completed by setting $t = (2p-1)\nu/2$ and simplifying Eq. (5.17). The following corollary immediately follows from Theorem 5.4.2 and highlights for what constants c the probability given in Theorem 5.4.2 converges.

Corollary 5.4.4. *For $0 < \nu < 1$, there is a constant $c = c(p, \nu)$ with $2/((2p-1)^2\nu^2) < c(p,\nu) < 3/((2p-1)^2\nu^2)$, so that for BRE $\mathbb{P}(\max_j |\hat{\pi}(j) - \pi^*(j)| \leq \nu n) \to 1$.*

Unbalanced Rank Estimation

In many situations, we are not interested in learning the entire permutation accurately but only care about the highest (or lowest) ranked objects. The well-known discounted cumulative gain [47], for example, captures this notion and has been important in the information retrieval literature. More recently, Rudin [65] proposed p-norms for ranking losses that penalize errors near the top more severely than in the tail of the list. The approach has been taken to the ∞-norm limit by Agarwal [3]. When n grows, the number of top elements we are interested in will typically also grow; in many natural phenomena, for example, we expect more extreme examples to appear as we make more observations. Suppose then, that for some $0 < \nu < 1$ we wish to recover the placement of the first νn elements in the permutation with fairly good accuracy, but care less about the remaining $(1-\nu)n$ elements. Surprisingly, a very slight modification of the Balanced Rank Estimation Algorithm yields a method that is useful in this situation. Furthermore, it still only requires a random subset of pairwise comparisons. The new algorithm, URE, estimates an object's score by the fraction of measured items preceding it.

Unbalanced Rank Estimation (URE):
Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{1}{m(n)} \sum_{i \neq j} s_{i,j} \bar{c}^n_{i,j} \propto \sum_{i \neq j} s_{i,j} \bar{c}^n_{i,j}.$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

To begin, we first establish that this algorithm in expectation still achieves the $\Omega(n)$ lower bound given in Proposition 5.3.2.

Theorem 5.4.5. *For any $0 < \eta < 1$, there is a constant $c(p, \eta) \in \Theta(1/((2p-1)^2\eta^2))$ so that if $m(n)/n \geq c(p,\eta)/n$, URE satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2} \binom{n}{2}. \tag{5.18}$$

Similar to Theorem 5.4.1, the proof relies on a tail inequality for the difference $\hat{\Pi}(i) - \hat{\Pi}(j)$. Supposing that $\pi^* = (1, \ldots, n)$, we show in the proof that for $i < j$

$$\mathbb{P}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right) \tag{5.19}$$

$$\leq \exp\left\{ - \left[\frac{j-i}{n}\right]^2 \frac{3}{100}(2p-1)^2 m(n) \right\}. \tag{5.20}$$

As in Theorem 5.4.1 we can use this to bound the Kendall tau distance as

$$\mathbb{E}(\mathrm{inv}(\hat{\pi})) \leq \frac{n}{n-1}\sqrt{\frac{400}{3}} \frac{1}{(2p-1)\sqrt{m(n)}} \binom{n}{2}. \tag{5.21}$$

Finally, equating this upper bound with $(\eta/2)(n(n-1)/2)$ allows us to solve for $m(n) \in \Theta(1/((2p-1)^2\eta^2))$.

Theorem 5.4.5 guarantees in the noiseless case ($p = 1$) that for any $0 < \eta < 1$, URE in expectation achieves the same $\Theta(n/\eta^2)$ sample complexity as the SVM in Proposition 5.3.1. Our main interest in URE, however, is encapsulated in the following theorem which shows that predicted permutations are much more accurate near the top than the bottom if an average of $\Theta(n\log(n))$ measurements are made instead[2].

Theorem 5.4.6. *For any $c > 0$, and $0 < \nu < 1$, if each comparison is measured with probability $m(n)/n = c\log(n)/n$, URE predicts with probability at least*

$$1 - 2n^{1-\frac{3}{2}\left[(2p-1)^2\nu^2/(3(1-p)+(5p-1)\nu)\right]c} \tag{5.22}$$

a permutation $\hat{\pi}$ with

$$|\pi^*(j) - \hat{\pi}(j)| \leq \begin{cases} 4\nu n & \text{if } \pi^*(j) < \nu n \\ 4\sqrt{\nu\pi^*(j)n} & \text{if } \pi^*(j) \geq \nu n \end{cases}. \tag{5.23}$$

The proof parallels that of Theorem 5.4.2 and shows that $\hat{\Pi}(j)$ concentrates around its expectation $\tilde{\Pi}^*(j) \triangleq \mathbb{E}(\hat{\Pi}(j)) = aj/n + b$, with $a = (2p-1)$ and $b \in \mathbb{R}$ (again, we assume $\pi^* = (1, \ldots, n)$). However, while in Theorem 5.4.2 the tail bound was identical for each j, here the scores $\hat{\Pi}(j)$ have variances that depend on j. To build intuition, in the noiseless case ($p = 1$), since the first element $j = 1$ in π^* has no items preceding it (i.e., $\forall i \neq j$ $\bar{c}_{i,j} = c_{i,j} = \mathbf{1}(i < j) = 0$), the estimated score $\hat{\Pi}(j)$ will always be zero and have zero variance, regardless of how many elements we measure. For remaining elements, the mean of the estimated scores will progressively increase down the permutation, as will their variance. The increase in variance brings a decrease in their predictive accuracy, which is reflected in the theory. Specifically, one can show that

$$\mathbb{P}\left(\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > t\right) \tag{5.24}$$

$$\leq 2\exp\left\{-\frac{t^2 m(n)}{2\left(\frac{j}{n}p + (1-p) + \frac{t}{3}\right)}\right\}. \tag{5.25}$$

Before applying a union bound to the above bounds, it is convenient to first eliminate the j-dependence of the upper bounds. To do this, we define the following set of increasing deviation events

$$A_j = \begin{cases} \left\{\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > \sqrt{\nu}t\right\} & \text{if } j < \nu n \\ \left\{\left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| > \sqrt{\frac{j}{n}}t\right\} & \text{if } j \geq \nu n. \end{cases} \tag{5.26}$$

[2]Of course, a minor modification of the algorithm leads to better estimation near the bottom.

Some algebra then gives, for all j,

$$\mathbb{P}(A_j) \leq 2\exp\left\{-\frac{\sqrt{\nu}t^2 m(n)}{2\left(\sqrt{\nu}p + \frac{1}{\sqrt{\nu}}(1-p) + \frac{t}{3}\right)}\right\}, \tag{5.27}$$

which yields the following union bound:

$$\mathbb{P}\big(\cup_{j=1}^{n} A_j\big) \leq 2n\exp\left\{-\frac{\sqrt{\nu}t^2 m(n)}{2\left(\sqrt{\nu}p + \frac{1-p}{\sqrt{\nu}} + \frac{t}{3}\right)}\right\}. \tag{5.28}$$

As in Theorem 5.4.2, we turn this concentration result into a bound on the rank displacement using a lemma.

Lemma 5.4.7. *For $a > 0$, $0 < \gamma < a^2$ and $b \in \mathbb{R}$, if*

$$\left|\hat{\Pi}(j) - \left(\frac{aj}{n} + b\right)\right| \leq \begin{cases} \gamma/a & \text{if } j < \gamma n/a^2 \\ \sqrt{\gamma j/n} & \text{if } j \geq \gamma n/a^2 \end{cases}, \tag{5.29}$$

then

$$|\hat{\pi}(j) - \pi^*(j)| \leq \begin{cases} 4\gamma n/a^2 & \text{if } j < \gamma n/a^2 \\ 4\sqrt{\gamma j n}/a & \text{if } j \geq \gamma n/a^2 \end{cases}. \tag{5.30}$$

The proof of the lemma shows that even if a sorting algorithm breaks ties in the least favorable way, the final rank positions cannot differ too much from the true positions in π^*. The main difficulty for this argument lies in a suitable definition of the sets A_j, which translates into the preconditions used for this lemma. As before, the lemma applies with $a = (2p-1)$. The result follows if for any $0 < \nu < 1$ we set $t = a\sqrt{\nu}$ in the definition of sets A_j, $\gamma = \nu a^2$ in Lemma 5.4.7, simplify Eq. (5.28) and then substitute $\pi^*(j)$ for j.

The following corollary, highlighting suitable constants c, follows immediately from Theorem 5.4.6 and shows for what constants c the probability in Theorem 5.4.6 converges.

Corollary 5.4.8. *For any $0 < \nu < 1$, there is a constant $c = c(p,\nu)$ with $2p/((2p-1)^2\nu) + 2(1-p)/((2p-1)^2\nu^2) \leq c(p,\nu) \leq 3p/((2p-1)^2\nu) + 2(1-p)/((2p-1)^2\nu^2)$, so that as $n \to \infty$ the displacement bounds of Theorem 5.4.6 hold with probability 1.*

Discussion. In both Theorems 5.4.2 and 5.4.6 the size of the bins into which we correctly place objects can be decreased by increasing the number of measurements. If we consider the noiseless case ($p = 1$), Corollary 5.4.8 predicts that to place elements j with $\pi^*(j) < \nu n/2$ into bins half the current size, URE needs on average twice as many comparisons. To correctly place objects j with $\pi^*(j) \geq \nu n$ into bins of half the size URE needs on average four times as many measurements. From Corollary 5.4.4, we see that the behavior of the BRE is rather different. There, a four-fold increase is required to halve the bin sizes uniformly across the

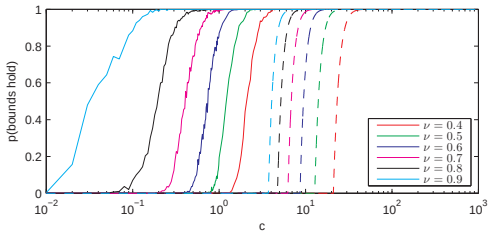
permutation. The cost of URE's improved performance near the top, however, is that for the same amount of data, the bin sizes in the tail are typically larger than those of BRE. Thus, if only the top elements are of interest, URE should be preferred. If a more uniform recovery is desired, BRE should be chosen. We will highlight this tradeoff in Section 5.5 with an example. An advantage in this regard is that the algorithm can be chosen *after* the data has been collected since BRE and URE work with the same type of input data. This fact could be exploited by combining the score estimators in various ways to further improve over the individual prediction results.

5.5 Experiments

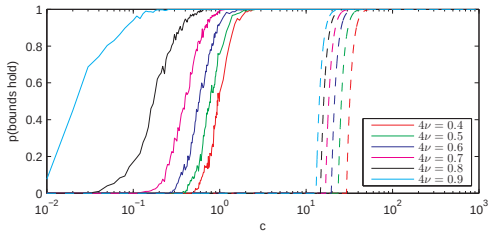
To begin, we empirically validate Theorems 5.4.2 and 5.4.6 in the noiseless case ($p = 1$). The theorems show that if each comparison is measured with probability $c\log(n)/n$, for some constant c, then the deviations $|\pi^*(j) - \hat{\pi}(j)|$ can be controlled with some probability that depends on c. In Figures 5.1(a) and 5.1(b) we show for particular choices of ν in solid the empirical probabilities that the displacement bounds of the theorems hold, as a function of the constant c. Additionally, we show the theoretical lower bounds on these probabilities, as given in Theorems 5.4.2 and 5.4.6. Notice that ν is four times smaller in Figure 5.1(b) than in Figure 5.1(a) so that Theorems 5.4.2 and 5.4.6 predict the same upper bounds on $|\pi^*(j) - \hat{\pi}(j)|$ for j s.t. $\pi^*(j) \leq \nu n$. Empirically, we see that in this case BRE requires more measurements than URE. To highlight the difference in prediction quality, we evaluated both algorithms on an 8000-object permutation. For each of 500 simulation runs, both algorithms saw *exactly* the same set of comparisons. In Figure 5.1(c) we show the median displacement $|\pi^*(j) - \hat{\pi}(j)|$ across the 500 runs, as a function of $\pi^*(j)$. Additionally, the error bars show 1/2 times the standard deviation of the displacements. For $\pi^*(j) < 2000$ URE predicts the correct position with higher accuracy and smaller variance than BRE. However, for large $\pi^*(j) \geq 2000$ BRE outperforms.

5.6 Extensions

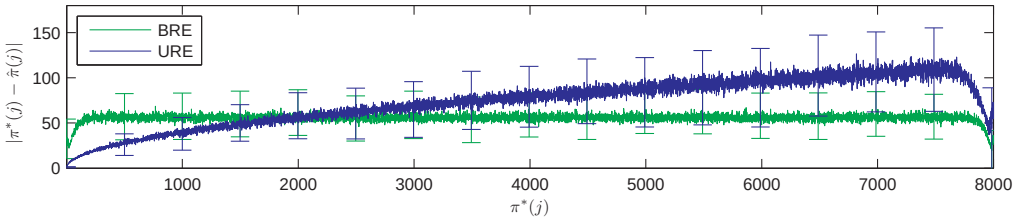
An important benefit of BRE/URE over active methods is that data collection can be trivially parallelized: Comparisons can be collected from independent processes, each measuring within a pre-assigned block of object pairs. Furthermore, the structure of the score estimators makes it easy to extend BRE/URE to several interesting settings. For one, we see applications in online ranking where we wish to grow rankings over n to $n+1$ objects as data streams in. Online versions of BRE/URE are easy to derive, yet lead to similar guarantees as those in Section 5.4. In contrast, the solutions of optimization-based methods can be non-trivial to update when the problem is slightly perturbed. Cauwenberghs and Poggio [14], for example, show that the exact update to an SVM solution requires careful bookkeeping of dual coefficients. The simple structure of BRE/URE also makes them useful in distributed



(a) BRE with lower bounds of Theorem 5.4.2.



(b) URE with lower bounds of Theorem 5.4.6.



(c) BRE and URE on the same set of comparisons. Different prediction characteristics are visible.

Figure 5.1: Empirical validation of Theorems 5.4.2 and 5.4.6. Figures 5.1(a) and 5.1(b) show for various ν in solid the empirical probabilities that the displacement bounds of Theorems 5.4.2 and 5.4.6 hold, if each comparison is measured independently with probability $c\log(n)/n$, as a function of c. To estimate these, we ran 300 noiseless simulations on permutations over 1000 objects and computed the fraction of times the bounds held. The empirical probabilities can be compared to the corresponding lower bounds produced by Theorems 5.4.2 and 5.4.6, which we plot as dashed curves. Figure 5.1(c) shows a direct comparison of our proposed algorithms. For each of 500 runs on an 8000-object permutation task, both algorithms saw *exactly* the same comparisons. Each plot shows the median displacement $|\pi^*(j) - \hat{\pi}(j)|$, as a function of $\pi^*(j)$.

settings where costly coordination and communication among multiple processors can be avoided. We will now explore this extension.

Distributed Ranking

In many situations, the number n of objects being compared is large. For instance, online retailers can easily offer millions of products for sale among which comparisons could be made. In such situations the objects (data points) are often stored on a fixed number K of machines, so that each machine stores about $f = n/K$ data points. A consequence of this distributed storage is that the $O(nm(n))$ comparisons are likely to be collected on distinct machines. A naïve centralized ranking algorithm would collect the individual comparisons at a server for learning, incurring a communication cost of $O(nm(n))$. This cost is prohibitive

if, relative to n, $m(n)$ is large[3]. Distributed, iterative SVM-type algorithms have been developed for such situations [41, 38] however, their application is typically complicated by the need for running multiple iterations which must be coordinated by locking protocols. As a result, the efficiency of these methods can rapidly deteriorate if a single machine fails. A favorable property of BRE/URE is that their simple form lends them much more naturally to distributed extensions, which can avoid locking protocols altogether. The main idea is that the BRE/URE object scores can also be computed from partial scores rather than from individual comparisons. If the number of binary comparisons $O(nm(n))$ is large, then communicating partial scores can be much more efficient. We analyze this setting.

To compute comparisons, any algorithm must start by exchanging object encodings between the K machines. Let the data points allocated to machine k be D_k. There are $K(K-1)/2$ machine pairs $(k < l)$ that need to exchange $f = n/K$ data points from one computer to the other. Overall, this leads to $n(K-1)/2 \in O(nK) = O(n^2/f)$ data points being exchanged. Once the $O(nm(n))$ comparisons have been computed (in distributed fashion), we aggregate them into partial scores. Specifically, denote the set of binary comparisons created by a machine pair $k \leq l$ by

$$\bar{C}_{k,l} = \{\bar{c}_{i,j} : i \in D_k, j \in D_l, s_{i,j} = 1\}. \tag{5.31}$$

Because $\bar{c}_{j,i} = 1 - \bar{c}_{i,j}$ if $s_{j,i} = s_{i,j} = 1$, the set $\bar{C}_{l,k}$ can easily be computed from $\bar{C}_{k,l}$. In the following we will assume that $\bar{C}_{l,k}$ has been implicitly computed in this way whenever necessary. For BRE, use $\bar{C}_{k,l}$ to compute for each pair k, l the following partial scores

$$\hat{\Pi}_{k,l}(j) = \sum_{i \in D_k} s_{i,j}(2\bar{c}_{i,j} - 1) \quad \forall j \in D_l. \tag{5.32}$$

This amounts to a total of $K^2 f = n^2/f$ partial scores. The partial scores for URE follow a similar strategy. To complete the algorithm, the partial scores must be communicated to a central machine at cost $O(n^2/f)$. If $l(j)$ is the machine index $l \in \{1, \ldots, K\}$ so that $j \in D_l$, we combine the partial scores as

$$\hat{\Pi}(j) = \sum_{k=1}^{K} \hat{\Pi}_{k,l(j)}(j). \tag{5.33}$$

The overall communication time is $O(n^2/f)$. In comparison, a naïve centralized algorithm requires communication time $O(nm(n))$. If $m(n) \in O(1/((2p-1)^2\eta^2)) \gg K$ then our proposed algorithm significantly reduces the communication time. For practical applications, the number of machines K is typically less than 100. In this case the our algorithm should be a viable alternative to centralized optimization schemes with η as large as $\eta = 0.1$.

[3]This could be because for a particular problem size n the constant $c(p, \eta) \in \Theta(1/((2p-1)^2\eta^2))$ in Theorems 5.4.1 and 5.4.5 happens to be large, or because the probability p of correctly measuring a comparison decreases quickly as a function of n.

5.7 Conclusions

This paper analyzed two simple algorithms for ranking n objects from a random sample of binary comparisons. We showed that the algorithms in expectation achieve a lower bound on the sample complexity for predicting a ranking with fixed expected Kendall tau distance. As such, they are competitive alternatives to the SVM, which also achieves the lower bound. By giving the algorithm slightly more measurements, we showed that interesting displacement bounds between $\hat{\pi}$ and π^* can be derived.

Because the algorithms rely only on a random subset of pairwise comparisons, data collection can be trivially parallelized. The simple structure of the scoring functions makes them easy to adapt to new situations, such as online or distributed ranking. We showed that in the latter case the communication cost of a traditional centralized optimization approach can be substantially reduced if $(2p-1)^2\eta^2$ is sufficiently small.

This paper has exclusively considered scoring functions $\Pi(j)$ that only depend on the object identity. However, BRE and URE can act as a useful performance baseline even for learning parametric scoring functions, as frequently considered: If the in-sample empirical performance of such parametric ranking functions is worse than that predicted by Theorems 5.4.1 and 5.4.5, the function class may need to be redesigned or more data collected. Moreover, the two algorithms can be used as quick, general-purpose preprocessing algorithms for conventional ranking methods: A small subset of pairwise comparisons can be approximately completed using BRE or URE, irrespective of the true (possibly parametric) ranking function that generated them. This larger set of comparisons could then be useful in learning an improved parametric ranking function.

5.8 Appendix

SVM has Optimal Sample Complexity

In this section we will show that the SVM, applied to ranking has an $O(n)$ sample complexity. A related claim (without complete proof) has been made in [61]. We then show that this sample complexity is tight.

Proposition 5.3.1. *There is a constant d, so that for any $0 < \eta < 1$, if we noiselessly measure dn/η^2 binary comparisons, chosen uniformly at random with replacement, and $n > n_0$ is large enough, the SVM will produce a prediction $\hat{\pi}$, which satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2}. \tag{5.34}$$

Proof. First, note that in the noiseless case the datapoints $x_{i,j}$ with associated labels $\bar{y}_{i,j}$ are linearly separable. The SVM finds such a separator. Since the measured comparisons are chosen uniformly at random with replacement, we can prove the outcome by appealing to learning-theoretic generalization bounds. Using results in (for example) Bousquet et al. [9] and a VC dimension bound of [61], we have the following Lemma:

Lemma. *For $\delta > 0$, and $0 < \eta < 1$, if we noiselessly measure dn/η^2 binary comparisons, chosen uniformly at random with replacement, then for some constant c, with probability at least $1 - \delta$ the SVM produces a permutation $\hat{\pi}$ with*

$$inv(\hat{\pi}) \leq \eta\left[\frac{c}{\sqrt{d}} + \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{dn}}\right]\binom{n}{2}. \tag{5.35}$$

To use this Lemma, first define

$$t = \eta\left[\frac{c}{\sqrt{d}} + \sqrt{\frac{2\log\left(\frac{2}{\delta}\right)}{dn}}\right]\binom{n}{2}. \tag{5.36}$$

Then by the Lemma $\mathbb{P}(\text{inv}(\hat{\pi}) > t) \leq \delta$. Notice that if we plug in $\delta = 1$ into t, we get a value t_1 for which $\mathbb{P}(\text{inv}(\hat{\pi}) > t_1) \leq \delta = 1$

$$t_1 = \eta\left[\frac{c}{\sqrt{d}} + \sqrt{\frac{2\log(2)}{dn}}\right]\binom{n}{2}. \tag{5.37}$$

We will thus assume that for all $t \leq t_1$, we have $\mathbb{P}(\text{inv}(\hat{\pi}) > t) \leq 1$. We can use this result to upper bound $\mathbb{E}(\text{inv}(\hat{\pi}))$ as follows. Since $\text{inv}(\hat{\pi}) \geq 0$,

$$\mathbb{E}(\text{inv}(\hat{\pi})) = \int_0^\infty \mathbb{P}(\text{inv}(\hat{\pi}) > t)dt \leq t_1 + \int_{t_1}^\infty \mathbb{P}(\text{inv}(\hat{\pi}) > t)dt. \tag{5.38}$$

All that remains is to express δ in terms of t. Rewriting, we find that

$$\delta = 2\exp\left\{-\frac{1}{2\sigma_n^2}\left(t-\mu_n\right)^2\right\} \tag{5.39}$$

$$\sigma_n^2 = \frac{\eta^2(n(n-1))^2}{4dn} \qquad \mu_n = \eta\frac{cn(n-1)}{2\sqrt{d}}. \tag{5.40}$$

Returning to our original expectation,

$$\mathbb{E}(\text{inv}(\hat{\pi})) \leq t_1 + \int_{t_1}^{\infty} \delta dt \tag{5.41}$$

$$= t_1 + 2\sqrt{2\pi\sigma_n^2}\int_{t_1}^{\infty}\frac{1}{\sqrt{2\pi\sigma_n^2}}\exp\left\{-\frac{1}{2\sigma_n^2}\left(t-\mu_n\right)^2\right\}dt \tag{5.42}$$

$$= t_1 + 2\sqrt{2\pi\sigma_n^2}\mathbb{P}(\sigma_n z + \mu_n > t_1) \tag{5.43}$$

$$\leq t_1 + 2\sqrt{2\pi\sigma_n^2} \tag{5.44}$$

$$= \eta\left[\frac{c}{\sqrt{d}} + \sqrt{\frac{2\log(2)}{dn}}\right]\binom{n}{2} + 2\eta\sqrt{2\pi}\frac{n(n-1)}{2\sqrt{dn}} \tag{5.45}$$

$$= \eta\left[\frac{c}{\sqrt{d}} + \frac{\sqrt{2\log(2)}+\sqrt{8\pi}}{\sqrt{dn}}\right]\binom{n}{2} \tag{5.46}$$

Suppose we set $d = 16c^2$. Then for any $\eta/4 > \epsilon_0 > 0$, there is an n_0 so that if $n > n_0$,

$$\mathbb{E}(\text{inv}(\hat{\pi})) \leq \left[\frac{\eta}{4} + \epsilon_0\right]\binom{n}{2} \leq \frac{\eta}{2}\binom{n}{2}. \tag{5.47}$$

□

Next we prove the following proposition via results of Giesen et al. [35] to show that the sample complexity in Proposition 5.3.1 is effectively tight.

Proposition 5.3.2. *For $\eta < 1$, any randomized, comparison-based algorithm that produces for all π^* a prediction $\hat{\pi}$ with an expected risk of*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2} \tag{5.48}$$

must on expectation use at least $\Omega(n)$ comparisons in the worst case.

Proof. Recall that for two permutations $\hat{\pi}, \pi^*$, Spearman's footrule distance is defined to be

$$D(\hat{\pi}) = \sum_{j=1}^{n}\left|\hat{\pi}(j) - \pi^*(j)\right|. \tag{5.49}$$

Giesen et al. [35] give the following theorem (we restate slightly):

Theorem (Giesen et al. [35])**.** *Any randomized, comparison-based algorithm that produces for each input permutation π^* a prediction $\hat{\pi}$ with expected Spearman's footrule distance $D(\hat{\pi})$ of at most $n^2/\nu(n)$, must on expectation use at least $n\left(\min(\log(\nu(n)), \log(n)) - 6\right)$ comparisons in the worst case.*

Because Spearman's footrule can be upper bounded by twice Kendall's tau distance [22] (i.e. $D(\hat{\pi}) \leq 2\text{inv}(\hat{\pi})$) we have that if

$$\mathbb{E}\left(\text{inv}(\hat{\pi})\right) \leq \frac{\eta}{2}\binom{n}{2} \tag{5.50}$$

then

$$\mathbb{E}\left(D(\hat{\pi})\right) \leq \mathbb{E}\left(2\text{inv}(\hat{\pi})\right) \leq \eta\binom{n}{2} < \eta\frac{n^2}{2} \tag{5.51}$$

Combining with the above theorem of Giesen et al. [35], we have the desired result. □

Proofs for BRE and URE

Proof of Theorem 5.4.1

Balanced Rank Estimation Algorithm (BRE): Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{\sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)}{2m(n)} \propto \sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)\,. \tag{5.52}$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

Theorem 5.4.1. *For any fixed $0 < \eta < 1$ there is a constant $c(p, \eta) \in \Theta(1/(2p-1)^2\eta^2)$ so that if $m(n)/n \geq c(p,\eta)/n$ and $n > n_0$, the Balanced Rank Estimation Algorithm satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2}\binom{n}{2}. \tag{5.53}$$

Proof. Without loss of generality, we suppose that $\pi^* = (1, 2, \ldots, n)$, so that $\pi^*(j) = j$. The true (rescaled) rank score of object j is denoted by

$$\Pi^*(j) = \frac{1}{n}\sum_{i \neq j} c_{i,j} = \frac{j-1}{n}. \tag{5.54}$$

Knowing the true scores Π^* clearly suffices to predict the permutation π^* with 0 Kendall tau distance. Equivalently, it suffices to know the true scores up to global scaling and additive constants. We will show that the Balanced Rank Estimation Algorithm produces an unbiased estimate of the scores

$$\tilde{\Pi}^*(j) = \Pi^*(j)(2p-1) + \left(1 - p - \frac{1}{2}\right)\frac{n-1}{n}, \tag{5.55}$$

where p parameterizes the observation model. We will then show that for large enough $m(n)$, the probability that we predict an incorrect binary comparison (i.e. we find that $\hat{\Pi}(j) < \hat{\Pi}(i)$ even though $i < j$) will decay fast enough with growing $|j-i|$ to guarantee that the expected Kendall tau distance achieves the target. Our interest lies in computing

$$\mathbb{E}\left(\mathrm{inv}(\hat{\pi})\right) = \mathbb{E}\left(\sum_{i<j} \mathbf{1}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right)\right) \tag{5.56}$$

$$= \sum_{i<j} \mathbb{P}\left(\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right] \geq \left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]\right). \tag{5.57}$$

Bernstein concentration: Introduce binary random variables $b_{i,j}$ that capture if a comparison was flipped or not. Let $b_{i,j} = 1$ indicate that the measurement was not flipped. By assumption, the $b_{i,j}$ are i.i.d. and $\mathbb{P}(b_{i,j} = 1) = p$. Hence, we can write $\bar{c}_{i,j} = b_{i,j}c_{i,j} + (1 - b_{i,j})(1 - c_{i,j})$. We study the difference between the estimated scores and the target scores:

$$\hat{\Pi}(j) - \tilde{\Pi}^*(j) = \hat{\Pi}(j) - \left(\Pi^*(j)(2p-1) + \left(1 - p - \frac{1}{2}\right)\frac{n-1}{n}\right) \tag{5.58}$$
$$= \frac{1}{n}\sum_{i\neq j}\left[\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - 1\right) - \left(c_{i,j}(2p-1) + \left(1 - p - \frac{1}{2}\right)\right)\right]$$

Note that given π^*, the variables in the sum are independent. Some algebra reveals:

$$\mathbb{E}\left(\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - 1\right) - \left(c_{i,j}(2p-1) + \left(1 - p - \frac{1}{2}\right)\right)\right) = 0$$
$$\text{var}\left(\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - 1\right) - \left(c_{i,j}(2p-1) + \left(1 - p - \frac{1}{2}\right)\right)\right) \leq \frac{n}{4m(n)}$$
$$\left|\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - 1\right) - \left(c_{i,j}(2p-1) + \left(1 - p - \frac{1}{2}\right)\right)\right| \leq \frac{n+m(n)}{2m(n)}.$$

The difference $\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right]$ can be written as a sum of $2n-3$ independent terms with magnitude no more than twice and variance no more than four times that stated above. Thus, we can derive a Bernstein concentration inequality [8]. Conditioned on π^*, one can show that

$$\mathbb{P}\left(\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right] \geq \left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]\right) \tag{5.59}$$
$$= \mathbb{P}\left(\frac{2m(n)}{2(n+m(n))}\frac{n}{2n-3}\left(\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right]\right) \geq \frac{2m(n)}{2(n+m(n))}\frac{n}{2n-3}\left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]\right)$$
$$\leq \exp\left\{-\frac{(2n-3)\left(\frac{n}{2n-3}\right)^2\left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]^2}{2\frac{(n+m(n))^2}{m(n)^2}\left(\frac{4n}{4m(n)}\frac{m(n)^2}{(n+m(n))^2} + \frac{\left(\frac{n}{2n-3}\right)\left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]}{3}\frac{m(n)}{n+m(n)}\right)}\right\} \tag{5.60}$$
$$\leq \exp\left\{-\left[\frac{j-i}{n}\right]^2\frac{3}{32}(2p-1)^2 m(n)\right\}, \tag{5.61}$$

where we used the fact that we must eventually have $n > 3$ as well as $m(n) < n-3$ if $n > n_0$.

Returning to the expected Kendall tau distance of Eq. (5.57), we may bound

$$\mathbb{E}\left(\mathrm{inv}(\hat{\pi})\right) \leq \sum_{i<j} \exp\left\{-\left[\frac{j-i}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} \tag{5.62}$$

$$= \sum_{k=1}^{n-1} (n-k) \exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} \tag{5.63}$$

$$\leq \int_0^n (n-k) \exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{32}(2p-1)^2 m(n)\right\} dk \tag{5.64}$$

$$= n^2 \frac{\sqrt{\pi \frac{3}{32}(2p-1)^2 m(n)}\ \mathrm{erf}\left(\sqrt{\frac{3}{32}(2p-1)^2 m(n)}\right)}{2\frac{3}{32}(2p-1)^2 m(n)} + \tag{5.65}$$

$$n^2 \frac{\exp\left\{-\frac{3}{32}(2p-1)^2 m(n)\right\} - 1}{2\frac{3}{32}(2p-1)^2 m(n)} \tag{5.66}$$

$$\leq \frac{n}{n-1}\sqrt{\frac{128}{3}} \frac{2}{2(2p-1)\sqrt{m(n)}} \binom{n}{2}. \tag{5.67}$$

For this bound to be no larger than $\frac{\eta}{2}\binom{n}{2}$, we need

$$\eta > \frac{n}{n-1}\sqrt{\frac{128}{3}} \frac{2}{(2p-1)\sqrt{m(n)}}, \tag{5.68}$$

so that a suitable $m(n)$ exists which satisfies $m(n) \in \Theta(1/((2p-1)^2\eta^2))$. □

Proof of Theorem 5.4.2

Balanced Rank Estimation Algorithm (BRE): Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{\sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)}{2m(n)} \propto \sum_{i \neq j} s_{i,j}(2\bar{c}_{i,j} - 1)\,. \tag{5.69}$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

Theorem 5.4.2. *For $c > 0$, if each comparison is measured with probability $m(n)/n = c\log(n)/n$, Balanced Rank Estimation produces with probability at least $1 - 2n^{1-a_n \frac{3}{8}(2p-1)^2\nu^2 c}$ a permutation $\hat{\pi}$ with*

$$\max_j |\hat{\pi}(j) - \pi^*(j)| \leq \nu n, \tag{5.70}$$

where a_n is a sequence with $a_n \to 1$.

Proof. Without loss of generality, we suppose that $\pi^* = (1, 2, \ldots, n)$, so that $\pi^*(j) = j$. The true (rescaled) rank score of object j is denoted by

$$\Pi^*(j) = \frac{1}{n}\sum_{i \neq j} c_{i,j} = \frac{j-1}{n}. \tag{5.71}$$

Knowing the true scores Π^* clearly suffices to predict the permutation π^* with 0 Kendall tau distance. Equivalently, it suffices to know the true scores up to global scaling and additive constants. We will show that the Balanced Rank Estimation Algorithm produces an unbiased estimate of the scores

$$\tilde{\Pi}^*(j) = \Pi^*(j)(2p-1) + \left(1 - p - \frac{1}{2}\right)\frac{n-1}{n}. \tag{5.72}$$

In particular, we will show that for a large enough $m(n)$, the difference $|\hat{\Pi}(j) - \tilde{\Pi}^*(j)|$ is with high probability small for all j and that we will thus not confuse the relative ordering of two objects that are further than twice this difference apart in π^*.

Bernstein concentration: Introduce binary random variables $b_{i,j}$ that capture if a comparison was flipped or not. Let $b_{i,j} = 1$ indicate that the measurement was not flipped. We assume that $\mathbb{P}(b_{i,j} = 1) = p$ and that the $b_{i,j}$ are i.i.d. Hence, we can write $\bar{c}_{i,j} = b_{i,j}c_{i,j} + (1 - b_{i,j})(1 - c_{i,j})$. As in Theorem 5.4.2, we study the difference between the estimated scores and the rescaled and translated target scores:

$$\hat{\Pi}(j) - \tilde{\Pi}^*(j) = \hat{\Pi}(j) - \left(\Pi^*(j)(2p-1) + \left(1 - p - \frac{1}{2}\right)\frac{n-1}{n}\right) \tag{5.73}$$

$$= \frac{1}{n}\sum_{i \neq j}\left[\frac{n}{2m(n)} s_{i,j}\left(2\left[b_{i,j}c_{i,j} + (1 - b_{i,j})(1 - c_{i,j})\right] - 1\right) - \left(c_{i,j}(2p-1) + \left(1 - p - \frac{1}{2}\right)\right)\right]$$

Note that given π^*, the variables in the sum are independent. The following result, which were previously used in Theorem 5.4.2, can be shown

$$\mathbb{E}\left(\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j}+(1-b_{i,j})(1-c_{i,j})\right]-1\right)-\left(c_{i,j}(2p-1)+\left(1-p-\frac{1}{2}\right)\right)\right)=0$$

$$\text{var}\left(\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j}+(1-b_{i,j})(1-c_{i,j})\right]-1\right)-\left(c_{i,j}(2p-1)+\left(1-p-\frac{1}{2}\right)\right)\right)\leq\frac{n}{4m(n)}$$

$$\left|\frac{n}{2m(n)}s_{i,j}\left(2\left[b_{i,j}c_{i,j}+(1-b_{i,j})(1-c_{i,j})\right]-1\right)-\left(c_{i,j}(2p-1)+\left(1-p-\frac{1}{2}\right)\right)\right|\leq\frac{n+m(n)}{2m(n)}.$$

Thus, we can derive a Bernstein concentration inequality [8]. Conditioned on π^*,

$$\mathbb{P}\left(\left|\hat{\Pi}(j)-\tilde{\Pi}^*(j)\right|>t\right) \tag{5.74}$$

$$=\mathbb{P}\left(\left|\frac{n}{n-1}\frac{2m(n)}{n+m(n)}\hat{\Pi}(j)-\frac{n}{n-1}\frac{2m(n)}{n+m(n)}\tilde{\Pi}^*(j)\right|>\frac{n}{n-1}\frac{2m(n)}{n+m(n)}t\right) \tag{5.75}$$

$$\leq 2\exp\left\{-\frac{(n-1)t^2\left(\frac{n}{n-1}\right)^2}{2\frac{(n+m(n))^2}{4m(n)^2}\left(\frac{n}{4m(n)}\frac{4m(n)^2}{(n+m(n))^2}+\frac{t}{3}\frac{n}{n-1}\frac{2m(n)}{n+m(n)}\right)}\right\} \tag{5.76}$$

$$\leq 2\exp\left\{-\frac{n}{n+m(n)}\frac{t^2 4m(n)}{2\left(1+\frac{2t}{3}\right)}\right\}. \tag{5.77}$$

Since there are n items to be sorted, we apply a union bound

$$\mathbb{P}\left(\exists j:\left|\hat{\Pi}(j)-\tilde{\Pi}^*(j)\right|>t\right)\leq 2\exp\left\{-\frac{n}{n+m(n)}\frac{t^2 4m(n)}{2\left(1+\frac{2t}{3}\right)}+\log(n)\right\}. \tag{5.78}$$

The concentration result tells us that the relative ordering of two objects that are far apart in the true ordering (large t) is harder to confuse than that of nearby objects (small t). Thus, as n gets large, the relative ordering of any sufficiently well-separated pair in π^* should with high probability be predicted correctly in $\hat{\pi}$. Specifically, we have the following Lemma.

Lemma 5.4.3. *For some $a>0$ and $b\in\mathbb{R}$, if $\forall j$, $\left|\hat{\Pi}(j)-(\Pi^*(j)a+b)\right|\leq t$, then $\forall j$, $|\hat{\pi}(j)-\pi^*(j)|\leq 2tn/a$.*

Proof. We have $\hat{\pi}(j)\neq\pi^*(j)$ when one or more elements in $\hat{\Pi}$ are mapped to the wrong side of $\hat{\Pi}(j)$. Equivalently, to bound $|\hat{\pi}(j)-\pi^*(j)|$ we can count how many elements of $\hat{\Pi}$ can at most map to the same value $\hat{\Pi}(j)$ and to assume that the sorting algorithm breaks ties in the least favorable way. Note that

$$\frac{n}{a}\left|\hat{\Pi}(j)-(\Pi^*(j)a+b)\right|=\left|\frac{n}{a}\hat{\Pi}(j)-\left(j-1+\frac{nb}{a}\right)\right|<\frac{tn}{a}. \tag{5.79}$$

Hence, $|\hat{\pi}(j)-\pi^*(j)|\leq 2tn/a$. □

Putting it together: By the definition of the mean score $\tilde{\Pi}^*$ in Eq. (5.72), we see that we need $a = (2p-1)$ for Lemma 5.4.3. Then, in order to show that $\forall j$, $|\hat{\pi}(j) - \pi^*(j)| \leq \nu n$ with high probability, we need that with high probability $\forall j, |\hat{\Pi}(j) - \tilde{\Pi}^*(j)| \leq (2p-1)\nu/2$. Looking at Eq. (5.78), we can achieve this if we let $m(n) \geq c(p,\nu)\log(n)$, for a sufficiently large constant $c(p,\nu)$. Then, as $n \to \infty$, with high probability $\forall j$, $|\hat{\pi}(j) - \pi^*(j)| \leq \nu n$.

Probability of success. The probability that the preconditions to Lemma 5.4.3 hold depend on the constant $c(p,\nu)$. Specifically,

$$\mathbb{P}\left(\forall j : \left|\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right| < \frac{(2p-1)\nu}{2}\right) \tag{5.80}$$

$$\geq 1 - 2\exp\left\{\left[1 - \frac{n}{n + c(p,\nu)\log(n)} \frac{((2p-1)\nu)^2 c(p,\nu)}{2\left(1 + \frac{(2p-1)\nu}{3}\right)}\right]\log(n)\right\} \tag{5.81}$$

$$\geq 1 - 2\exp\left\{\left[1 - \frac{n}{n + c(p,\nu)\log(n)} \frac{3}{8}((2p-1)\nu)^2 c(p,\nu)\right]\log(n)\right\} \tag{5.82}$$

$$= 1 - 2n^{1 - a_n \frac{3}{8}(2p-1)^2\nu^2 c(p,\nu)}, \tag{5.83}$$

where $a_n = n/(n + c(p,\nu)\log(n)) \to 1$. □

Proof of Theorem 5.4.5

Unbalanced Rank Estimation Algorithm (URE): Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{1}{m(n)} \sum_{i \neq j} s_{i,j} \bar{c}_{i,j} \propto \sum_{i \neq j} s_{i,j} \bar{c}_{i,j}. \tag{5.84}$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

Theorem 5.4.5. *For any fixed $0 < \eta < 1$, there is a constant $c(p, \eta) \in \Theta(1/((2p-1)^2\eta^2))$ so that if each comparison is measured with probability at least $m(n)/n \geq c(p, \eta)/n$, the Unbalanced Rank Estimation Algorithm satisfies*

$$\mathbb{E}(inv(\hat{\pi})) \leq \frac{\eta}{2} \binom{n}{2}. \tag{5.85}$$

Proof. Without loss of generality, we suppose that $\pi^* = (1, 2, \ldots, n)$, so that $\pi^*(j) = j$. The true (rescaled) rank of object j is denoted by

$$\Pi^*(j) = \frac{1}{n} \sum_{i \neq j} c_{i,j} = \frac{j-1}{n}. \tag{5.86}$$

It suffices to know the true ranking scores up to global scaling and additive constants. One can show that the Unbalanced Rank Estimation Algorithm produces an unbiased estimate of the scores

$$\tilde{\Pi}^*(j) = \Pi^*(j)(2p-1) + (1-p)\frac{n-1}{n}. \tag{5.87}$$

We will show that for a large enough $m(n)$, the probability that we predict an incorrect binary comparison (i.e. we find that $\hat{\Pi}(j) < \hat{\Pi}(i)$ even though $i < j$) will decay fast enough with growing $|j-i|$ to guarantee that the expected Kendall tau distance achieves the target. Our interest lies in upper bounding

$$\mathbb{E}\left(\text{inv}(\hat{\pi})\right) = \mathbb{E}\left(\sum_{i<j} \mathbf{1}\left(\hat{\Pi}(j) \leq \hat{\Pi}(i)\right)\right) \tag{5.88}$$

$$= \sum_{i<j} \mathbb{P}\left(\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right] \geq \left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]\right). \tag{5.89}$$

Bernstein concentration: We introduce binary random variables $b_{i,j}$ which encode whether or not a comparison was flipped. Let $b_{i,j} = 1$ indicate that the measurement was not

flipped. By assumption, the $b_{i,j}$ are i.i.d. and $\mathbb{P}(b_{i,j} = 1) = p$. With this, we can write $\bar{c}_{i,j} = b_{i,j}c_{i,j} + (1 - b_{i,j})(1 - c_{i,j})$ and so

$$\hat{\Pi}(j) - \tilde{\Pi}^*(j) = \hat{\Pi}(j) - \left(\Pi^*(j)(2p-1) + (1-p)\frac{n-1}{n}\right) \tag{5.90}$$

$$= \frac{1}{n}\sum_{i\neq j}\left[\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - (c_{i,j}(2p-1) + (1-p))\right]. \tag{5.91}$$

Given π^*, the random variables inside the sum are independent and one can show that

$$\mathbb{E}\left(\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - (c_{i,j}(2p-1) + (1-p))\right) = 0 \tag{5.92}$$

$$\frac{1}{n-1}\sum_{i\neq j}\mathrm{var}\left(\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - (c_{i,j}(2p-1) + (1-p))\right) \tag{5.93}$$

$$\leq \frac{n}{m(n)}\frac{1}{n-1}\left[(j-1)p + (n-j)(1-p)\right] \tag{5.94}$$

$$\left|\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - (c_{i,j}(2p-1) + (1-p))\right| \leq \frac{n}{m(n)}. \tag{5.95}$$

The difference $\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right]$ can be written as a sum of $2n-3$ independent, zero-mean random variables, with magnitude at most twice and variance at most four times the above. Using the variance and magnitude bound, we can derive the following Bernstein concentration result [8]. Conditioned on π^*,

$$\mathbb{P}\left(\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right] \geq \left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]\right) \tag{5.96}$$

$$= \mathbb{P}\left(\frac{m(n)}{2n}\frac{n}{2n-3}\left(\left[\hat{\Pi}(i) - \tilde{\Pi}^*(i)\right] - \left[\hat{\Pi}(j) - \tilde{\Pi}^*(j)\right]\right) \geq \frac{m(n)}{2n}\frac{n}{2n-3}\left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]\right)$$

$$\leq \exp\left\{-\frac{(2n-3)\left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]^2}{2\frac{(2(2n-3))^2}{m(n)^2}\left(4\frac{n}{m(n)}\frac{1}{n-1}\left[(j-1)p + (n-j)(1-p)\right]\frac{m(n)^2}{4n^2} + \frac{\left[\tilde{\Pi}^*(j) - \tilde{\Pi}^*(i)\right]}{3}\frac{m(n)}{2(2n-3)}\right)}\right\}$$

$$\leq \exp\left\{-\left[\frac{j-i}{n}\right]^2\frac{3}{100}(2p-1)^2 m(n)\right\}. \tag{5.97}$$

Then plugging this in the expected Kendall tau distance,

$$\mathbb{E}\left(\text{inv}(\hat{\pi})\right) \leq \sum_{i<j} \exp\left\{-\left[\frac{j-i}{n}\right]^2 \frac{3}{100}(2p-1)^2 m(n)\right\} \tag{5.98}$$

$$= \sum_{k=1}^{n-1} (n-k) \exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{100}(2p-1)^2 m(n)\right\} \tag{5.99}$$

$$\leq \int_0^n (n-k) \exp\left\{-\left[\frac{k}{n}\right]^2 \frac{3}{100}(2p-1)^2 m(n)\right\} dk \tag{5.100}$$

$$= n^2 \frac{\sqrt{\pi \frac{3}{100}(2p-1)^2 m(n)}\ \text{erf}\left(\sqrt{\frac{3}{100}(2p-1)^2 m(n)}\right)}{2\frac{3}{100}(2p-1)^2 m(n)} + \tag{5.101}$$

$$n^2 \frac{\exp\{-\frac{3}{100}(2p-1)^2 m(n)\} - 1}{2\frac{3}{100}(2p-1)^2 m(n)} \tag{5.102}$$

$$\leq \frac{n}{n-1}\sqrt{\frac{400}{3}} \frac{2}{2(2p-1)\sqrt{m(n)}} \binom{n}{2}. \tag{5.103}$$

For this bound to be no larger than $\frac{\eta}{2}\binom{n}{2}$, we need

$$\eta > \frac{n}{n-1}\sqrt{\frac{400}{3}} \frac{2}{(2p-1)\sqrt{m(n)}}, \tag{5.104}$$

so that a suitable $m(n)$ exists which satisfies $m(n) \in \Theta\left(1/((2p-1)^2\eta^2)\right)$. □

Proof of Theorem 5.4.6

Unbalanced Rank Estimation Algorithm (URE): Measure each binary comparison independently with probability $m(n)/n$. Define the scores

$$\hat{\Pi}(j) = \frac{1}{m(n)} \sum_{i \neq j} s_{i,j} \bar{c}_{i,j} \propto \sum_{i \neq j} s_{i,j} \bar{c}_{i,j}. \tag{5.105}$$

Predict π^* by the ordering $\hat{\pi}$ of the estimated scores, breaking ties randomly.

Theorem 5.4.6. *For $c > 0$, if each comparison is measured with probability $m(n)/n = c\log(n)/n$, Unbalanced Rank Estimation produces with probability at least*

$$1 - 2n^{1 - \frac{3}{2}\left[(2p-1)^2\nu^2/(3(1-p)+(5p-1)\nu)\right]c} \tag{5.106}$$

a permutation $\hat{\pi}$ with

$$|\pi^*(j) - \hat{\pi}(j)| \leq \begin{cases} 4\nu n & \text{if } \pi^*(j) < \nu n \\ 4\sqrt{\nu}\sqrt{\pi^*(j) n} & \text{if } \pi^*(j) \geq \nu n \end{cases}. \tag{5.107}$$

Proof. Without loss of generality, we suppose that $\pi^* = (1, 2, \ldots, n)$, so that $\pi^*(j) = j$. To prove this theorem, we need to refine the Bernstein concentration from Theorem 5.4.5. The true (rescaled) rank score of object j is denoted by

$$\Pi^*(j) = \frac{1}{n} \sum_{i \neq j} c_{i,j} = \frac{j-1}{n}. \tag{5.108}$$

It suffices to know the true ranking scores up to global scaling and additive constants. We will show that the Unbalanced Rank Estimation Algorithm produces an unbiased estimate of the scores

$$\tilde{\Pi}^*(j) = \Pi^*(j)(2p-1) + (1-p)\frac{n-1}{n}. \tag{5.109}$$

Bernstein concentration: We introduce binary random variables $b_{i,j}$ which encode whether or not a comparison was flipped. Let $b_{i,j} = 1$ indicate that the measurement was not flipped. By assumption, the $b_{i,j}$ are i.i.d. and $\mathbb{P}(b_{i,j} = 1) = p$. With this, we can write $\bar{c}_{i,j} = b_{i,j}c_{i,j} + (1 - b_{i,j})(1 - c_{i,j})$ and so

$$\hat{\Pi}(j) - \tilde{\Pi}^*(j) = \hat{\Pi}(j) - \left(\Pi^*(j)(2p-1) + (1-p)\frac{n-1}{n}\right) \tag{5.110}$$

$$= \frac{1}{n} \sum_{i \neq j} \left[\frac{n}{m(n)} s_{i,j} \left[b_{i,j}c_{i,j} + (1-b_{i,j})(1-c_{i,j})\right] - (c_{i,j}(2p-1) + (1-p))\right]. \tag{5.111}$$

Given π^*, the random variables inside the sum are independent. Furthermore, one can show the following results, previously used in Theorem 5.4.5

$$\mathbb{E}\left(\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j}+(1-b_{i,j})(1-c_{i,j})\right]-(c_{i,j}(2p-1)+(1-p))\right)=0 \tag{5.112}$$

$$\frac{1}{n-1}\sum_{i\neq j}\operatorname{var}\left(\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j}+(1-b_{i,j})(1-c_{i,j})\right]-(c_{i,j}(2p-1)+(1-p))\right) \tag{5.113}$$

$$\leq \frac{n}{m(n)}\frac{1}{n-1}\left[(j-1)p+(n-j)(1-p)\right] \tag{5.114}$$

$$\left|\frac{n}{m(n)}s_{i,j}\left[b_{i,j}c_{i,j}+(1-b_{i,j})(1-c_{i,j})\right]-(c_{i,j}(2p-1)+(1-p))\right|\leq\frac{n}{m(n)}. \tag{5.115}$$

With this we can derive a refined Bernstein concentration result [8]. Conditioned on π^*,

$$\mathbb{P}\left(\left|\hat{\Pi}(j)-\tilde{\Pi}^*(j)\right|>t\right) \tag{5.116}$$

$$=\mathbb{P}\left(\frac{m(n)}{n}\frac{n}{n-1}\left|\hat{\Pi}(j)-\tilde{\Pi}^*(j)\right|>\frac{m(n)}{n}\frac{n}{n-1}t\right) \tag{5.117}$$

$$\leq 2\exp\left\{-\frac{(n-1)t^2\left(\frac{n}{n-1}\right)^2}{2\frac{n^2}{m(n)^2}\left(\frac{n}{m(n)}\frac{1}{n-1}\left[(j-1)p+(n-j)(1-p)\right]\frac{m(n)^2}{n^2}+\frac{t}{3}\frac{m(n)}{n}\frac{n}{n-1}\right)}\right\} \tag{5.118}$$

$$\leq 2\exp\left\{-\frac{t^2m(n)}{2\left(\frac{j}{n}p+(1-p)+\frac{t}{3}\right)}\right\}. \tag{5.119}$$

Let us now substitute different values of t. To begin, if $j<\nu n$, then by rescaling t,

$$\mathbb{P}\left(\left|\hat{\Pi}(j)-\tilde{\Pi}^*(j)\right|>\sqrt{\nu}t\right)\leq 2\exp\left\{-\frac{\nu t^2m(n)}{2\left(\frac{j}{n}p+(1-p)+\frac{t}{3}\sqrt{\nu}\right)}\right\} \tag{5.120}$$

$$\leq 2\exp\left\{-\frac{\sqrt{\nu}t^2m(n)}{2\left(\sqrt{\nu}p+\frac{1}{\sqrt{\nu}}(1-p)+\frac{t}{3}\right)}\right\} \tag{5.121}$$

And if $j\geq\nu n$, by rescaling t,

$$\mathbb{P}\left(\left|\hat{\Pi}(j)-\tilde{\Pi}^*(j)\right|>\sqrt{\frac{j}{n}}t\right)\leq 2\exp\left\{-\frac{t^2m(n)}{2\frac{n}{j}\left(\frac{j}{n}p+(1-p)+\frac{t}{3}\sqrt{\frac{j}{n}}\right)}\right\} \tag{5.122}$$

$$\leq 2\exp\left\{-\frac{\sqrt{\nu}t^2m(n)}{2\left(\sqrt{\nu}p+\frac{1}{\sqrt{\nu}}(1-p)+\frac{t}{3}\right)}\right\}. \tag{5.123}$$

Notice that the upper bounds do not depend on j and are identical in the two cases. Hence, we see that the concentration result becomes strong for small $j < \nu n$ as n gets large, but remains relatively weak for large $j \approx n$. Define the events

$$A_j = \begin{cases} \left\{ \left| \hat{\Pi}(j) - \tilde{\Pi}^*(j) \right| > \sqrt{\nu} t \right\} & \text{if} \quad j < \nu n \\ \left\{ \left| \hat{\Pi}(j) - \tilde{\Pi}^*(j) \right| > \sqrt{\frac{j}{n}} t \right\} & \text{if} \quad j \geq \nu n. \end{cases} \tag{5.124}$$

Applying a union bound, we find

$$\mathbb{P}\left(\bigcup_{j=1}^{n} A_j \right) \leq 2 \exp\left\{ - \frac{\sqrt{\nu} t^2 m(n)}{2\left(\sqrt{\nu} p + \frac{1}{\sqrt{\nu}}(1-p) + \frac{t}{3} \right)} + \log(n) \right\}. \tag{5.125}$$

To use the bound in Eq. (5.125), we first prove the following Lemma.

Lemma 5.4.7. *For some $a > 0$, $0 < \gamma < a^2$ and arbitrary $b \in \mathbb{R}$, if*

$$|\hat{\Pi}(j) - (\Pi^*(j)a + b)| \leq \begin{cases} \gamma/a & \text{if } j < \gamma n/a^2 \\ \sqrt{\gamma j/n} & \text{if } j \geq \gamma n/a^2 \end{cases}, \tag{5.126}$$

then

$$|\hat{\pi}(j) - \pi^*(j)| \leq \begin{cases} 4\gamma n/a^2 & \text{if } j < \gamma n/a^2 \\ 4\sqrt{\gamma j n}/a & \text{if } j \geq \gamma n/a^2 \end{cases}. \tag{5.127}$$

Proof. Let $\tilde{\Pi}^* = \Pi^*(j)a + b$. We need to bound the number of elements of $\hat{\Pi}$ that can appear on the wrong side of $\hat{\Pi}(j)$. The prediction $\hat{\pi}(j)$ can deviate from $\pi^*(j)$ by at most the number of such misplaced elements. It suffices to bound the number of elements of $\hat{\Pi}$ that can take on the same value $\hat{\Pi}(j)$ and then to assume that the Unbalanced Rank Estimation Algorithm breaks ties in the least favorable way. To maximize the number of such confusions, the estimated score $\hat{\Pi}(j)$ must deviate *up* from its mean value $\tilde{\Pi}^*(j)$ as much as possible, since then the most elements k with $\tilde{\Pi}^*(k) > \hat{\Pi}(j)$ can map *down* to $\hat{\Pi}(j)$, and the most elements k with $\hat{\Pi}(j) > \tilde{\Pi}^*(k)$ can map *up* to $\hat{\Pi}(j)$ (The conditions of the lemma ensure that larger deviations are possible for large k than small k.) Specifically, if $j \geq \gamma n/a^2$ then we should have $\hat{\Pi}(j) = \tilde{\Pi}^*(j) + \sqrt{\gamma j/n}$. If $j < \gamma n/a^2$, then we should have, $\hat{\Pi}(j) = \tilde{\Pi}^*(j) + \gamma/a$. For the following, denote $t_j = \sqrt{\gamma j/n}/a$, which is how much the rescaled score $\hat{\Pi}(j)/a$ can differ from $\tilde{\Pi}^*(j)/a = \Pi^*(j) + b/a$ if $j \geq \gamma n/a^2$.

Suppose that $j \geq \gamma n/a^2$. The largest element $\bar{k}$ that can overlap with j satisfies

$$\frac{\tilde{\Pi}^*(\bar{k})n}{a} - t_{\bar{k}} n = \frac{\tilde{\Pi}^*(j)n}{a} + t_j n \tag{5.128}$$

$$\bar{k} - 1 + \frac{bn}{a} - t_{\bar{k}} n = j - 1 + \frac{bn}{a} + t_j n \tag{5.129}$$

$$\bar{k} - t_{\bar{k}} n = j + t_j n \tag{5.130}$$

$$\bar{k} - \frac{\sqrt{n\bar{k}}\sqrt{\gamma}}{a} = j + \frac{\sqrt{nj}\sqrt{\gamma}}{a}. \tag{5.131}$$

Since $\sqrt{\bar{k}} \geq 0$, we use the positive solution given by the quadratic formula:

$$\sqrt{\bar{k}} = \frac{\sqrt{n\gamma}/a + \sqrt{n\gamma/a^2 + 4\sqrt{nj}\sqrt{\gamma}/a + 4j}}{2} \tag{5.132}$$

$$= \frac{\sqrt{n\gamma}/a + \sqrt{\left(\sqrt{n\gamma}/a + 2\sqrt{j}\right)^2}}{2} \tag{5.133}$$

$$= \frac{\sqrt{n\gamma}}{a} + \sqrt{j}. \tag{5.134}$$

With this, we have

$$t_{\bar{k}} = \sqrt{\frac{\bar{k}}{n}}\frac{\sqrt{\gamma}}{a} = \frac{\sqrt{n\gamma}/a + \sqrt{j}}{\sqrt{n}}\frac{\sqrt{\gamma}}{a} = \frac{\gamma}{a^2} + \frac{\sqrt{\gamma}}{a}\sqrt{\frac{j}{n}}. \tag{5.135}$$

By this derivation, at most $t_{\bar{k}}n$ elements k with $\tilde{\Pi}^*(k) > \hat{\Pi}(j)$ can map *down* to $\hat{\Pi}(j)$. Furthermore, at most $t_j n$ elements k with $\hat{\Pi}(j) > \tilde{\Pi}^*(k)$ can map *up* to $\hat{\Pi}(j)$. Taken together, because $t_j \leq t_{\bar{k}}$, at most $(t_j + t_{\bar{k}})n \leq 2t_{\bar{k}}n = 2\gamma n/a^2 + 2\sqrt{\gamma}\sqrt{jn}/a$ elements k can map *onto* $\hat{\Pi}(j)$. This means, even if the sorting algorithm breaks ties in the least favorable way, we have $\forall j \geq \gamma n/a^2$

$$|\hat{\pi}(j) - \pi^*(j)| \leq 2t_{\bar{k}}n = \frac{2\gamma n}{a^2} + \frac{2\sqrt{\gamma jn}}{a}. \tag{5.136}$$

Note that if $j \geq \gamma n/a^2$, the second term in Eq. (5.136) is at least as large as the first term. So $\forall j \geq \gamma n/a^2$

$$|\hat{\pi}(j) - \pi^*(j)| \leq \frac{4\sqrt{\gamma jn}}{a}. \tag{5.137}$$

On the other hand, if $j \leq \gamma n/a^2$, then at most $\bar{k}$ elements can map to $\hat{\Pi}(j)$. The upper limit $\bar{k}$ is largest when $j = \gamma n/a^2$. Thus,

$$|\hat{\pi}(j) - \pi^*(j)| \leq \bar{k} = \sqrt{\bar{k}}^2 = \frac{\gamma n}{a^2} + \frac{2\sqrt{\gamma nj}}{a} + j = \frac{4\gamma n}{a^2}. \tag{5.138}$$

□

We can now prove the theorem. In the context of $\tilde{\Pi}^*$ in Eq. (5.109), $a = (2p - 1)$. For any $0 < \nu < 1$, set $\gamma = \nu a^2$ in Lemma 5.4.7. If we set $t = a\sqrt{\nu}$, then the union bound in Eq. (5.125) controls the probability that the bounds on the scores required by the lemma will be satisfied so that we can use the lemma to draw the desired conclusion. Specifically, with these settings, if $m(n) \geq c(p, \nu)\log(n)$ with the constant $c(p, \nu)$ large enough, then by the lemma we predict with high probability the first νn elements j of π^* with accuracy $|\hat{\pi}(j) - \pi^*(j)| \leq 4\nu n$ and the remaining elements with accuracy $|\hat{\pi}(j) - \pi^*(j)| \leq 4\sqrt{\nu}\sqrt{jn}$.

Probability of success. The probability that the preconditions to Lemma 5.4.7 hold depend on the constant $c(p, \nu)$. Specifically,

$$1 - \mathbb{P}\left(\bigcup_{j=1}^{n} A_j\right) \geq 1 - 2\exp\left\{-\frac{\sqrt{\nu}\left((2p-1)\sqrt{\nu}\right)^2 c(p,\nu)\log(n)}{2\left(\sqrt{\nu}p + \frac{1}{\sqrt{\nu}}(1-p) + \frac{(2p-1)\sqrt{\nu}}{3}\right)} + \log(n)\right\} \tag{5.139}$$

$$= 1 - 2n^{1-\frac{3}{2}\frac{(2p-1)^2\nu^2}{(3(1-p)+(5p-1)\nu)}c(p,\nu)} \tag{5.140}$$

□

Chapter 6

Conclusions

Although the fundamental ideas of active and randomized subsampling reach back several decades (e.g. [51, 95]), their use in the context of novel statistical and computational models continues to reveal problems that must be addressed. In this dissertation we presented algorithmic and theoretic contributions that aim to alleviate some of these issues: Chapter 2 showed an improved active learning algorithm for spectral clustering models. In Chapter 3 we addressed tractability concerns of active learning in complex Bayesian models. We proposed a method to avoid overfitting in Gaussian Process Classification models learnt from a randomized subsample in Chapter 4. Finally, in Chapter 5 we presented theoretical analyses to highlight two simple ranking algorithms as alternatives to learning a ranking using an SVM. The issues we attack are often fundamental and a comprehensive treatment would benefit from an expansion of the work presented in this dissertation.

In the context of Chapter 2, our approach for modelling label uncertainty is in its present form fairly rudimentary, but it highlights the benefit of including probabilistic aspects into an otherwise non-probabilistic clustering model. Future work on active spectral clustering could benefit from a more coherent probabilistic treatment of labelling information, perhaps using a Bayesian observation model. Furthermore, our work does not give a theoretical account why the strategy works so much better than a randomized sampling approach.

The approximate active learning strategy of Chapter 3 highlights a method for approximating the stationary distribution of one Markov chain using the stationary distribution of a perturbed chain. We believe this work to be a first step in making active learning more palatable in complex Bayesian models. However, alternatives can and should be considered. Particle filtering and importance sampling, for instance, are both concerned with approximating samples from one distribution using samples from another, and some of the theory surrounding these techniques could be used to analyze hybrid alternatives for approximate active learning.

At present, the Heavy-tailed Process Classification model of Chapter 4 selectively shrinks posterior class probabilities to $1/2$ in sparse regions. In several circumstances it would be attractive to allow the model to shrink the class probabilities to arbitrary, pre-determined values. A simple way to achieve this would be to model the latent Heavy-tailed processes

to have non-zero mean. This would introduce extra parameters, that could be either fixed a priori or perhaps even learnt.

Chapter 5 analyzed two ranking algorithms that are useful when a random subset of binary comparisons is observed and data-collection cannot be actively guided at all. In many situations the cost of active learning might be less severe so that a small number of active measurements can be made in addition to the randomly sampled comparisons. In this setting, one may be able to develop interesting hybrid algorithms that interpolate between purely randomized ranking (as analyzed in Chapter 5) and traditional sorting algorithms. For instance, once the algorithms of Chapter 5 have been run, additional comparisons could be actively collected to refine the prediction (say) at the top of the permutation.

Bibliography

[1] D. Achlioptas and F. McSherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, 54, 2007.

[2] S. Agarwal. *A Study of the Bipartite Ranking Problem in Machine Learning*. PhD thesis, University of Illinois at Urbana-Champaign, 2005.

[3] S. Agarwal. The infinite push: A new support vector ranking algorithm that directly optimizes accuracy at the absolute top of the list. In *Proceedings of the SIAM International Conference on Data Mining*, 2011.

[4] N. Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13:137–164, 2012.

[5] N. Ailon, R. Begleiter, and E. Ezra. A new active learning scheme with applications to learning to rank from pairwise preferences. *arXiv CoRR*, abs/1110.2136, 2011.

[6] A. Ammar and D. Shah. Ranking: Compare, don't score. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control and Computing (Allerton)*, pages 776–783. 2011.

[7] Z. Bodó, Z. Minier, and L. Csató. Active learning with clustering. *Journal of Machine Learning Research*, 16:127–139, 2011.

[8] S. Boucheron, G. Lugosi, and O. Bousquet. Concentration inequalities. In *Advanced Lectures in Machine Learning*, pages 208–240. Springer, 2004.

[9] O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning*, volume 3176 of *Lecture Notes in Computer Science*, pages 169–207. Springer, 2003.

[10] M. Braverman and E. Mossel. Noisy sorting without resampling. In *Symposium on Discrete Algorithms*, pages 268–276, 2008.

[11] M. Braverman and E. Mossel. Sorting from noisy information. *arXiv CoRR*, abs/0910.1191, 2009.

[12] T. Broderick and R.B. Gramacy. Classification and categorical inputs with treed gaussian process models. *Journal of Classification*, 28(2):244–270, July 2011.

[13] C. Campbell, N. Cristianini, and A. Smola. Query learning with large margin classifiers. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 111–118. Morgan Kaufmann, 2000.

[14] G. Cauwenberghs and T. Poggio. Incremental and decremental support vector machine learning. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 409–415. MIT Press, 2000.

[15] K. Chaloner and I. Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

[16] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2004.

[17] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Transactions on Algorithms*, 6(3):55:1–55:13, 2010.

[18] D. Damian, P.D. Sampson, and P. Guttorp. Bayesian estimation of semi-parametric non-stationary spatial covariance structures. *Environmetrics*, 12:161–178, 2001.

[19] O. Dekel, C. Manning, and Y. Singer. Log-linear models for label ranking. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16 (NIPS)*. MIT Press, 2004.

[20] O. Dekel and O. Shamir. Good learners for evil teachers. In L. Bottou and M. Littman, editors, *Proceedings of the 26th International Conference on Machine Learning (ICML)*. Omnipress, 2009.

[21] O. Dekel and O. Shamir. Vox populi: Collecting high-quality labels from a crowd. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, Montreal, Quebec, Canada, 2009.

[22] P. Diaconis and R.L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.

[23] A. Dobra and A. Lenkoski. Copula Gaussian graphical models. Technical report, 2009.

[24] P. Donmez, J.G. Carbonell, and J. Schneider. Efficiently learning the accuracy of labeling sources for selective sampling. In *Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Paris, France, 2009. ACM.

[25] P. Drineas and R. Kannan. Pass-efficient algorithms for approximating large matrices. In *Proceedings of the Annual Symposium on Discrete Algorithms*, pages 223–232, 2003.

[26] P. Drineas, R. Kannan, and M.W. Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36:158–183, 2006.

[27] P. Drineas and M.W. Mahoney. On the Nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, 2005.

[28] U. Feige, P. Raghavan, D. Peleg, and E. Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

[29] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26, 2004.

[30] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[31] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, 2003.

[32] A. Frieze, R. Kannan, and S. Vempala. Fast Monte Carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51:1025–1041, 2004.

[33] J. Fulman. Stein's method, Jack measure, and the Metropolis algorithm. *Journal of Combinatorial Theory. Series A*, 108(2):275–296, 2004.

[34] T. Gärtner. A survey of kernels for structured data. *SIGKDD Explorations*, 2003.

[35] J. Giesen, E. Schuberth, and M. Stojaković. Approximate sorting. *Fundamenta Informaticae*, 90(1-2):67–72, 2009.

[36] D.F. Gleich and L. Lim. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 60–68. ACM, 2011.

[37] P.W. Goldberg, C.K.I. Williams, and C.M. Bishop. Regression with input-dependent noise: A Gaussian process treatment. In M.I. Jordan, M.J. Kearns, and S.A. Solla, editors, *Advances in Neural Information Processing Systems 10 (NIPS)*, pages 493–499. MIT Press, 1998.

[38] H.P. Graf, E. Cosatto, L. Bottou, I. Durdanovic, and V. Vapnik. Parallel support vector machines: The cascade SVM. In L.K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17 (NIPS)*. MIT Press, 2004.

[39] R.B. Gramacy and H.K.H. Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 2007.

[40] T.L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. Technical report, Gatsby Computational Neuroscience Unit, 2005.

[41] T. Hazan, A. Man, and A. Shashua. A parallel decomposition solver for SVM: Distributed dual ascend using Fenchel duality. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[42] R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

[43] L. Huang, D. Yan, M.I. Jordan, and N. Taft. Spectral clustering with perturbed data. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 705–712. MIT Press, 2009.

[44] P.G. Ipeirotis, F. Provost, and J. Wang. Quality management on Amazon Mechanical Turk. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP, pages 64–67, Washington DC, 2010.

[45] S. Jaimungal and E.K. Ng. Kernel-based copula processes. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*, pages 628–643. Springer, 2009.

[46] K.G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. 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 (NIPS)*, pages 2240–2248. MIT Press, 2011.

[47] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, 20(4):422–446, 2002.

[48] T. Joachims. Training linear SVMs in linear time. In *Proceedings of the 12th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 217–226. ACM, 2006.

[49] N. Jojic, A. Perina, and V. Murino. Structural epitome: A way to summarize one's visual experience. In J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 1027–1035. MIT Press, 2011.

[50] D.X. Li. On default correlation: A copula function approach. Technical Report 99-07, Riskmetrics Group, New York, April 2000.

[51] D.V. Lindley. On a measure of the information provided by an experiment. 27(4):986–1005, December 1956.

[52] H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10:1–37, 2009.

[53] S.L. Lohr. *Sampling: Design and Analysis*. Duxbury Press, 1st edition, December 1999.

[54] P.K. Mallapragada, R. Jin, and A.K. Jain. Active query selection for semi-supervised clustering. In *ICPR*, pages 1–4. IEEE, 2008.

[55] I. Mitliagkas, A. Gopalan, C. Caramanis, and S. Vishwanath. User rankings from comparisons: Learning permutations in high dimensions. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control and Computing (Allerton)*, 2011.

[56] S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In P. Bartlett, F. Pereira, C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 2483–2491. MIT Press, 2012.

[57] R.B. Nelsen. *An Introduction to Copulas*. Springer, 1999.

[58] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14 (NIPS)*, pages 849–856. MIT Press, 2002.

[59] University of Washington Information Design Lab. [idl.ee.washington.edu/SimilarityLearning/Applications/Datasets/]. Face similarity data, 2012.

[60] M. Pitt, D. Chan, and R.J. Kohn. Efficient Bayesian inference for Gaussian copula regression models. *Biometrika*, 93(3):537–554, 2006.

[61] K. Radinsky and N. Ailon. Ranking from pairs and triplets: Information quality, evaluation methods and query complexity. In I. King, W. Nejdl, and H. Li, editors, *Fourth ACM International Conference on Web Search and Data Mining (WSDM)*, pages 105–114. ACM, 2011.

[62] A. Rahimi and B. Recht. Clustering with normalized cuts is clustering with a hyperplane. In *Statistical Learning in Computer Vision*, 2004.

[63] C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[64] 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, April 2010.

[65] C. Rudin. The p-norm push: A simple convex ranking algorithm that concentrates at the top of the list. *Journal of Machine Learning Research*, 10:2233–2271, 2009.

[66] T. Sakai and A. Imiya. Fast spectral clustering with random projection and sampling. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 372–384. Springer, 2009.

[67] A.M. Schmidt and A. O'Hagan. Bayesian inference for nonstationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society*, 65(3):743–758, 2003. Ser. B.

[68] B. Settles. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[69] O. Shamir and N. Tishby. Spectral clustering on a budget. *Proceedings of the Infernational Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[70] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[71] V.S. Sheng, F. Provost, and P.G. Ipeirotis. Get another label? Improving data quality and data mining using multiple, noisy labelers. In *Proceeding of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, Las Vegas, Nevada, 2008. ACM.

[72] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[73] P. Smyth, U.M. Fayyad, M.C. Burl, P. Perona, and P. Baldi. Inferring ground truth from subjective labelling of Venus images. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7 (NIPS)*, pages 1085–1092. MIT Press, 1994.

[74] E. Snelson, C.E. Rasmussen, and Z. Ghahramani. Warped Gaussian processes. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 16 (NIPS)*, pages 337–344. MIT Press, 2004.

[75] R. Snow, B. O'Connor, D. Jurafsky, and A.Y. Ng. Cheap and fast—but is it good? Evaluating non-expert annotations for natural language tasks. In *Proceedings of EMNLP*. Association for Computational Linguistics, 2008.

[76] P.X. Song. Multivariate dispersion models generated from Gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320, 2000.

[77] A. Sorokin and D. Forsyth. Utility data annotation with Amazon Mechanical Turk. In *CVPR Workshop on Internet Vision*, Anchorage, Alaska, 2008.

[78] G.W. Stewart and J. Sun. *Matrix Perturbation Theory*. Computer Science and Scientific Computing. Academic Press, 1990.

[79] X. Su and T. M. Khoshgoftaar. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:4:2–4:2, January 2009.

[80] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Investigation of various matrix factorization methods for large recommender systems. In *Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 6:1–6:8. ACM, 2008.

[81] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, 2:45–66, 2002.

[82] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007.

[83] P. Wais, S. Lingamnei, D. Cook, J. Fennell, B. Goldenberg, D. Lubarov, D. Marin, and H. Simons. Towards building a high-quality workforce with Mechanical Turk. In *NIPS Workshop on Computational Social Science and the Wisdom of Crowds*, Whistler, BC, Canada, 2010.

[84] X. Wang and I. Davidson. Active spectral clustering. In *IEEE International Conference on Data Mining*, pages 561–568, 2010.

[85] F.L. Wauthier, N. Jojic, and M.I. Jordan. Active spectral clustering via iterative uncertainty reduction. In *Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1339–1347, Beijing, China, 2012. ACM.

[86] F.L. Wauthier, N. Jojic, and M.I. Jordan. Efficient ranking from pairwise comparisons. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, 2013.

[87] F.L. Wauthier and M.I. Jordan. Heavy-tailed process priors for selective shrinkage. In J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2406–2414. MIT Press, 2010.

[88] F.L. Wauthier and M.I. Jordan. Bayesian bias mitigation for crowdsourcing. 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 (NIPS)*, pages 1800–1808. MIT Press, 2011.

[89] P. Welinder, S. Branson, S. Belongie, and P. Perona. The multidimensional wisdom of crowds. In J. Lafferty, C. K. I. Williams, R. Zemel, J. Shawe-Taylor, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23 (NIPS)*, pages 2424–2432. MIT Press, 2010.

[90] J. Whitehill, P. Ruvolo, T. Wu, J. Bergsma, and J. Movellan. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22 (NIPS)*, pages 2035–2043. MIT Press, 2009.

[91] C.K.I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. In T.K. Leen, T.G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13 (NIPS)*, pages 682–688. MIT Press, 2001.

[92] Q. Xu, M. desJardins, and K.L. Wagstaff. Active constrained clustering by examining spectral eigenvectors. In *Proceedings of the International Conference on Discovery Science*, pages 294–307, 2005.

[93] Y. Yan, R. Rosales, G. Fung, and J.G. Dy. Active learning from crowds. In L. Getoor and T. Scheffer, editors, *Proceedings of the 28th International Conference on Machine Learning (ICML)*, Bellevue, Washington, 2011.

[94] Y. Yan, R. Rosales, G. Fung, M. Schmidt, G. Hermosillo, L. Bogoni, L. Moy, and J.G. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. In *Proceedings of AISTATS*, volume 9, Chia Laguna, Sardinia, Italy, 2010.

[95] F. Yates. *Sampling Methods for Consensuses and Surveys*. Hafner Publishing Company, 1960.

[96] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H. Kriegel. Probabilistic memory-based collaborative filtering. *IEEE Transactions On Knowledge and Data Engineering*, 16(1):56–69, January 2004.