

# **Interactive Algorithms for Unsupervised Machine Learning**

Akshay Krishnamurthy

CMU-CS-15-116

June 2015

School of Computer Science  
Carnegie Mellon University  
Pittsburgh, PA 15213

**Thesis Committee:**

Aarti Singh, Chair

Maria-Florina Balcan

Barnabás Póczós

Larry Wasserman

Sanjoy Dasgupta (UCSD)

John Langford (Microsoft Research)

*Submitted in partial fulfillment of the requirements  
for the degree of Doctor of Philosophy.*

Copyright © 2015 Akshay Krishnamurthy

This research was sponsored by the Air Force Office of Scientific Research under grant number FA95501010382 and the National Science Foundation under grant numbers IIS-1116458, IIS-1247658, IIS-1252412, DGE-0750271, and DGE-0750271. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

**Keywords:** Statistical Machine Learning, Interactive Learning, Unsupervised Learning, Matrix Completion, Subspace Learning, Hierarchical Clustering, Network Tomography.

*To my parents.*



## Abstract

This thesis explores the power of interactivity in unsupervised machine learning problems. Interactive algorithms employ feedback-driven measurements to reduce data acquisition costs and consequently enable statistical analysis in otherwise intractable settings. Unsupervised learning methods are fundamental tools across a variety of domains, and interactive procedures promise to broaden the scope of statistical analysis. We develop interactive learning algorithms for three unsupervised problems: subspace learning, clustering, and tree metric learning. Our theoretical and empirical analysis shows that interactivity can bring both statistical and computational improvements over non-interactive approaches. An over-arching thread of this thesis is that interactive learning is particularly powerful for *non-uniform* datasets, where non-uniformity is quantified differently in each setting.

We first study the subspace learning problem, where the goal is to recover or approximate the principal subspace of a collection of partially observed data points. We propose statistically and computationally appealing interactive algorithms for both the matrix completion problem, where the data points lie on a low dimensional subspace, and the matrix approximation problem, where one must approximate the principal components of a collection of points. We measure uniformity with the notion of incoherence, and we show that our feedback-driven algorithms perform well under much milder incoherence assumptions.

We next consider clustering a dataset represented by a partially observed similarity matrix. We propose an interactive procedure for recovering a clustering from a small number of carefully selected similarity measurements. The algorithm exploits non-uniformity of cluster size, using few measurements to recover larger clusters and focusing measurements on the smaller structures. In addition to coming with strong statistical and computational guarantees, this algorithm performs well in practice.

We also consider a specific metric learning problem, where we compute a latent tree metric to approximate distances over a point set. This problem is motivated by applications in network tomography, where the goal is to approximate the network structure using only measurements between pairs of end hosts. Our algorithms use an interactively chosen subset of the pairwise distances to learn the latent tree metric while being robust to either additive noise or a small number of arbitrarily corrupted distances. As before, we leverage non-uniformity inherent in the tree metric structure to achieve low sample complexity.

Finally, we study a classical hypothesis testing problem where we focus on show fundamental limits for non-interactive approaches. Our main result is a precise characterization of the performance of non-interactive approaches, which shows that, on particular problems, all non-interactive approaches are statistically weaker than a simple interactive one. These results bolster the theme that interactivity can bring about statistical improvements in unsupervised problems.



## Acknowledgments

First and foremost, I would like to thank Aarti Singh, my advisor, who has played a central role in shaping my research interest, style, and ability. Aarti has a keen awareness for broad context and perspective of our research that I strive to develop. She has challenged me to think deeply about research problems, encouraged me to pursue my own research interests, and provided me the support and freedom to grow individually. Her support, guidance, wisdom, and encouragement all helped shape this thesis and much of my work, and they were all instrumental to my success in graduate school.

I am thankful to Larry Wasserman, whose instruction in courses and guidance in research are a primary reason for my interest in statistical machine learning. Larry's encyclopedic knowledge and grasp of statistics were extremely valuable resources for my research, but I also appreciate his wise advice on personal and career matters. My collaboration with Barnabás Poczo's and Larry has been thought-provoking and fun, and I am thankful that they encouraged me to tackle new problems.

This thesis is a product of reading many papers on interactive learning by my committee members, Nina Balcan, Sanjoy Dasgupta, and John Langford. Nina's unbounded energy and her passion for machine learning are qualities that I strive for, while Sanjoy's comments during my proposal and defense have led me to many new ideas. I am inspired by John's deep understanding of both theory and practice and his ability to push both frontiers with unique and innovative ideas. I am truly excited to work closely with and continue to learn from John over the next year.

I am thankful for many amazing collaborators that I have had the opportunity to work with: Sivaraman Balakrishnan for listening to my ideas and spending the time to think deeply about them, Min Xu for teaching me the importance of rigor, James Sharpnack for teaching me that simple problems can have deep and beautiful answers, and Martin Azizyan and Kirthevasan Kandasamy for always being eager to discuss research and brainstorm with me. You all have become wonderful friends, and I look forward to future meetings and collaborations.

I would like to thank many faculty and staff members at Carnegie Mellon University for conversations and interactions that I cherish. It has been fun to discuss statistics problems in the gym with Ryan Tibshirani, who has become a good friend. I had a wonderful TA experience with Venkat Guruswami, and he, along with Anupam Gupta, have encouraged me to learn more about Theoretical Computer Science. I am thankful for many conversations with Mor Harchol-Balter that helped convince me to pursue a career in academia. I am also grateful to all of the phenomenal staff members, but particularly Deb Cavlovich, Catherine Copetas, and Diane Stidle who greatly enriched my life at CMU.

I am thankful to Zeeshan Syed and Eu-Jin Goh who supported me during my internship at Google and helped develop my engineering ability. I am also thankful to Alekh Agarwal, Miro Dudík, Kai-Wei Chang and many others at Microsoft Research

NYC for a fun and productive internship. I am looking forward to spending another year at MSR and continuing to collaborate with and learn from everyone at the lab.

Many friends in Pittsburgh and elsewhere helped temper the challenges of graduate school and I am truly fortunate to have such a supportive network: My officemates, Sarah Loos, Jeremiah Blocki, and Colin White; fellow computer science students Dana and Yair Movshovitz-Attias, Gabe and Cat Weisz, John Wright, David Witmer, Kevin Waugh, Erik Zawadzki, JP Dickerson, Jamie Morgenstern; machine learning friends Matus Telgarsky, Aaditya Ramdas, Gautam Dasarathy, Mladen Kolar, and Willie Neiswanger; ultimate frisbee friends Jesse Kummer, Aaron Kane, Ben Clark, Nipunn Koorapati, Jeremy Kanter, Nick Audette, Andy Fassler, Lily Nguyen, and Carolyn Norwood; California friends Robbie Paolini, Ravi Raghavan, and Hassan Khan who moved to Pittsburgh with me; and all of my close friends from college, high school, and beyond. Thank you all for the amazing memories!

Lastly, I would like to thank my family: my parents, my grandparents, and my brother, Jayant Krishnamurthy. It was a unique and wonderful experience to attend graduate school with Jayant and his support was invaluable. I will cherish our many attempts at collaboration and our conversations about research and life. I am eternally thankful to my parents who have both been wonderful role models. Thank you so much for your enduring love and support!



# Contents

- 1 Introduction** **1**
- 1.1 Overarching Themes . . . . . 2
- 1.2 Overview of Results . . . . . 3
  - 1.2.1 Interactive Subspace Learning . . . . . 3
  - 1.2.2 Interactive Hierarchical Clustering . . . . . 4
  - 1.2.3 Interactive Latent Tree Metric Learning . . . . . 5
  - 1.2.4 Passive and Interactive Sampling in Normal Means Inference . . . . . 5
- 1.3 Related Work . . . . . 6
  
- 2 Interactive Matrix Completion** **9**
- 2.0.1 Preliminaries . . . . . 10
- 2.1 Related Work . . . . . 12
  - 2.1.1 Related work on Matrix and Tensor Completion . . . . . 12
  - 2.1.2 Related work on Matrix Approximation . . . . . 13
- 2.2 Matrix and Tensor Completion . . . . . 15
  - 2.2.1 Necessary conditions for non-interactive sampling . . . . . 18
- 2.3 Matrix Approximation . . . . . 20
  - 2.3.1 Comparison with related results . . . . . 22
- 2.4 Proofs . . . . . 23
  - 2.4.1 Proof of Theorem [2.1](#) and Corollary [2.2](#) . . . . . 23

2.4.2	Proof of Theorem 2.3 . . . . .	30
2.4.3	Proof of Theorem 2.4 . . . . .	31
2.4.4	Proof of Theorem 2.5 and related propositions . . . . .	33
2.5	Empirical Results . . . . .	36
2.6	Conclusions . . . . .	40
<b>3</b>	<b>Interactive Hierarchical Clustering</b>	<b>41</b>
3.1	Related Work . . . . .	43
3.2	Main Results . . . . .	44
3.2.1	An Interactive Clustering Framework . . . . .	45
3.2.2	Interactive Spectral Clustering . . . . .	48
3.2.3	Active $k$ -means clustering . . . . .	49
3.2.4	Fundamental Limits . . . . .	50
3.3	Experimental Results . . . . .	52
3.3.1	Practical Considerations . . . . .	52
3.3.2	Simulations . . . . .	53
3.3.3	Real World Experiments . . . . .	54
3.4	Proofs . . . . .	56
3.4.1	Proof of Theorem 3.1 . . . . .	56
3.4.2	Proof of Theorem 3.2 . . . . .	61
3.4.3	Proof of Theorem 3.3 . . . . .	64
3.4.4	Proof of Proposition 3.4 . . . . .	65
3.4.5	Proof of Theorem 3.5 . . . . .	66
3.5	Discussion . . . . .	67
<b>4</b>	<b>Interactive Latent Tree Metric Learning</b>	<b>69</b>
4.1	Related Work . . . . .	71

4.2	Background . . . . .	72
4.3	Algorithms . . . . .	75
4.3.1	Additive Noise . . . . .	75
4.3.2	Persistent Noise . . . . .	77
4.4	Experiments . . . . .	80
4.4.1	Simulations . . . . .	81
4.4.2	Real World Experiments . . . . .	82
4.5	Proofs . . . . .	83
4.5.1	Proof of Theorem 4.1 . . . . .	83
4.5.2	Proof of Theorem 4.2 . . . . .	85
4.5.3	Proof of Theorem 4.3 . . . . .	85
4.5.4	Proof of Theorem 4.4 . . . . .	91
4.6	Conclusion . . . . .	92
<b>5</b>	<b>Minimaxity in the Structured Normal Means Problem</b>	<b>93</b>
5.1	Related Work . . . . .	95
5.2	Main Results . . . . .	96
5.2.1	Bounds on the Minimax Risk . . . . .	96
5.2.2	Minimax-Optimal Recovery . . . . .	98
5.2.3	The Experimental Design Setting . . . . .	102
5.3	Examples . . . . .	103
5.3.1	$k$ -sets . . . . .	104
5.3.2	Biclusters . . . . .	104
5.3.3	Stars . . . . .	105
5.3.4	Random Codes . . . . .	106
5.4	Discussion . . . . .	106
5.5	Proofs . . . . .	107

5.5.1	Proof of Theorem 5.1 . . . . .	107
5.5.2	Proof of Theorem 5.5 . . . . .	109
5.5.3	Proof of Proposition 5.6 . . . . .	110
5.5.4	Proof of Theorem 5.4 . . . . .	110
5.5.5	Calculations for the examples . . . . .	112
<b>6</b>	<b>Conclusions</b>	<b>119</b>
<b>A</b>	<b>Concentration Inequalities</b>	<b>121</b>
	<b>Bibliography</b>	<b>123</b>

# List of Figures

- 2.1 (a): Probability of success of Algorithm 1 versus fraction of samples per column ( $p = m/d$ ) with  $r = 10, \mu_0 = 1$ . (b): Data from (a) plotted against samples per column,  $m$ . (c): Probability of success of Algorithm 1 versus fraction of samples per column ( $p = m/d$ ) with  $n = 500, \mu_0 = 1$ . (d): Data from (c) plotted against rescaled sample probability  $p/(r \log r)$ . . . . . 37
- 2.2 (a): Probability of success of Algorithm 1 versus fraction of samples per column ( $p = m/d$ ) with  $n = 500, r = 10$ . (b): Data from (a) plotted against rescaled sampling probability  $p/\mu_0$ . (c): Probability of success of SVT versus rescaled sampling probability  $np/\log(n)$  with  $r = 5, \mu_0 = 1$ . (d): Probability of success of Algorithm 1 and SVT versus sampling probability for matrices with highly coherent row space with  $r = 5, n = 100$ . . . . . 37
- 2.3 (a): An example matrix with with highly non-uniform column norms and (b) the sampling pattern of Algorithm 3. (c): Relative error as a function of sampling probability  $p$  for different target rank  $r$  ( $\mu = 1$ ). (d): The same data where the  $y$ -axis is instead  $\epsilon/\sqrt{r}$ . . . . . 38
- 2.4 (a): Relative error of Algorithm 3 as a function of sampling probability  $p$  for different size matrices with fixed target rank  $r = 10$  and  $\mu = 1$ . (b): The same data where the  $y$ -axis is instead  $\sqrt{p}\epsilon$ . (c): Relative error for interactive and non-interactive sampling on matrices with uniform column lengths (column coherence  $\mu = 1$  and column norms are uniform from  $[0.9, 1.1]$ ). (d): Relative error for interactive and non-interactive sampling on matrices with highly nonuniform column lengths (column coherence  $\mu = 1$  and column norms are from a standard Log-Normal distribution). . . . . 39
- 2.5 Experiments on real datasets. Left: Log excess risk for passive and interactive matrix approximation algorithms on a 400-node subset of the King internet latency dataset with target rank  $r = 26$ . Right: Log excess risk for passive and interactive matrix approximation algorithms on a  $1000 \times 10,000$  submatrix of the PubChem Molecular Similarity dataset with target rank  $r = 25$ . Passive algorithm is based on uniform sampling followed by hard thresholding of the singular values. . . . . 40

3.1	Sampling pattern of Algorithm 4 . . . . .	46
3.2	Simulation experiments. Top row: Noise thresholds for Algorithm 5, k-means clustering, ACTIVESPECTRAL, and ACTIVEKMEANS with $s = \log^2(n)$ for interactive algorithms. Bottom row from left to right: probability of success as a function of $s$ for $n = 256, \sigma = 0.75$ , outlier fractions on noisy CBM, probing complexity, and runtime complexity. . . . .	53
3.3	Experiments: 3.3(a): Comparison of algorithms on various datasets. 3.3(b): Outlier fractions on datasets with ground truth clustering. 3.3(c): Subset of the NIPS hierarchy. . . . .	55
3.4	Heatmaps of permuted matrices for SNP, Phylo, NIPS, and RTW (from left to right). Algorithms are HEURSPEC, ACTIVESPECTRAL, and ACTIVEKMEANS from top to bottom. . . . .	57
4.1	Possible structures for four leaves in a tree. If $d(w, x) + d(y, z) < d(w, y) + d(x, z) = d(w, z) + d(x, y)$ then structure and labeling is that of (a). If $d(w, x) + d(y, z) = d(w, y) + d(x, z) = d(w, z) + d(x, y)$ then structure is a star (b). . . . .	73
4.2	CDFs of $\epsilon$ values in the 4PC- $\epsilon$ condition for two real world datasets (King [98] and IPlane datasets [129]) along with a dataset of points drawn uniformly from the surface of a sphere, where geodesic distance defines the metric. . . . .	74
4.3	Noise Thresholds for PEARLRECONSTRUCT and RISING. . . . .	81
4.4	Measurements used as a function of $p$ for PEARLRECONSTRUCT, RISING, DFS Ordering [86], SLT [135], and Sequoia [142] . . . . .	81
4.5	CDF of relative error on King (a) and iPlane (b) datasets. . . . .	83
4.6	Measurements used on real world data sets . . . . .	83
5.1	Example structured normal means problem on nine points in two dimensions. Left: polyhedral acceptance regions of MLE. Center: Acceptance regions of Bayes estimator from the optimized prior computed by Algorithm 12. Right: Success probability landscape (success probability for each hypothesis) for the two estimators, demonstrate that the optimized estimator has better minimax risk. . . . .	101
5.2	Left: A realization of the stars problem for a graph with 13 vertices and 34 edges with sampling budget $\tau = 34$ . Edge color reflects allocation of sensing energy and vertex color reflects success probability for MLE under that hypothesis (warmer colors are higher for both). Isotropic (left) has minimum success probability of 0.44 and experimental design (center) has minimum success probability 0.56. Right: Maximum risk for isotropic and experimental design sampling as a function of $\mu$ for stars problem on a 50 and 100-vertex graph. . . . .	105

# Chapter 1

## Introduction

Interactive learning is a framework for statistical analysis in which the inference procedure interacts with the data acquisition mechanism to make feedback-driven measurements. This framework, which is also referred to as active learning, adaptive sampling, or adaptive sensing, has become increasingly popular in recent years as it often significantly reduces overhead associated with data collection. On both theoretical and empirical fronts, interactive learning has been successfully applied to a variety of supervised machine learning [19, 21, 22, 23, 29, 30, 64, 65, 67, 102, 103, 104] and signal processing problems [17, 107, 124, 130, 161]. However, interactive approaches have not experienced the same degree of success for unsupervised learning, and our understanding in this area is quite limited. This thesis addresses this deficiency with an exploration of the power of interactive approaches for unsupervised learning.

Unsupervised learning refers to a broad class of learning problems where the dataset is not endowed with label information and the explicit goal is to identify some structural characteristics of the data. This contrasts with supervised problems where data points are associated with labels, and the goal is to learn an accurate mapping from data points to their labels. Examples of unsupervised learning range from clustering and manifold learning, where the goal is to capture locality information, to hypothesis testing, where the goal is to understand the data-generating process more generically. Unsupervised learning plays an important role in exploratory data analysis, as it provides the statistician with some basic understanding of the dataset.

Unfortunately, unsupervised learning tasks, formulations, and algorithms are extremely varied, making a unified treatment challenging. Our study of interactive approaches for unsupervised learning therefore focuses on several important and representative examples rather than a general treatment. Our choices of examples are motivated by two considerations: the learning problem should be widely studied and practically relevant, and there should be concrete applications where an interactive approach is feasible. Our experience is that ideas in the development of these examples will be applicable in other unsupervised learning problems.

Through these examples, we show that interactive learning offers three distinct advantages. First, interactive algorithms have lower sample requirements than non-interactive ones, and are there-

fore *statistically* more efficient. Secondly, interactive approaches are particularly powerful when the data exhibits high degrees of *non-uniformity*, as the sampling mechanism can focus measurements to accurately capture these aspects of the data. Lastly, interactivity offers a *computational* improvement as these algorithms are often both theoretically and empirically faster than non-interactive ones. These claims are supported by the several examples in this thesis. More formally, our thesis statement is:

**Thesis statement:** Interactive data acquisition facilitates statistically and computationally efficient unsupervised learning algorithms that are particularly well-suited to handle non-uniform datasets.

In the remainder of this chapter, we describe these three advantages in some more detail and then turn to an overview of the main results. We conclude this chapter with a broad discussion of related work on interactive learning.

## 1.1 Overarching Themes

In the context of unsupervised learning, we claim that interactive approaches offers three distinct advantages over non-interactive ones. These are:

1. **Statistical efficiency:** The main appeal of interactive learning is statistical efficiency. Intuitively, by incorporating feedback into the measurement process, an interactive algorithm should be able to achieve suitable statistical performance with fewer measurements than a non-interactive one. Indeed, interactive learning is a strictly more powerful model, but there are many documented examples where interactivity is known to not provide significant statistical improvements over non-interactive approaches [11, 63, 114]. In this thesis, we study a number of unsupervised learning problems and show that interactivity in fact does lead to significantly improved statistical performance.

In the machine learning community, statistical efficiency is usually quantified by *sample complexity*, which is the number of samples required to achieve a certain accuracy in a learning task. In the signal processing literature, a *signal-to-noise ratio*, which measures the problem difficulty, is more commonly used. We use both notions in this thesis, depending on the problem of study, but make fair comparisons to other approaches throughout.

2. **Computational efficiency:** Given the increasing size and complexity of data sets, computational efficiency is an important consideration when designing learning algorithms. In addition to statistical efficiency, we also argue that interactive approaches can be computationally more efficient than non-interactive ones, particularly in unsupervised settings. This claim is challenging to argue theoretically, as it requires establishing a computational lower bound on non-interactive algorithms, and proving such lower bounds is notoriously hard. We instead compare our algorithms against non-interactive ones, both theoretically, in their asymptotic running times, and empirically, via extensive simulation.

We find it surprising that interactive approaches actually lead to computational improve-



ments over non-interactive ones, as many algorithms for interactive supervised learning do not demonstrate this phenomenon [20, 107, 174]. One exception is the algorithm due to Beygelzimer et al. [30], which is often faster than passive learning in practice, but reduces active learning to a possibly NP-hard zero-one loss empirical risk minimization problem. One reason for this is that these algorithms perform sophisticated computations to select future measurements, while we find that, in the unsupervised problems considered here, much simpler sampling techniques suffice. These simple sampling approaches, along with the fact that interactive algorithms can ignore large fractions of the dataset, lead to the computational improvements demonstrated in this thesis.

3. **Coping with non-uniformity:** Lastly, we find that interactive learning algorithms are particularly well-suited to data sets with high degrees of non-uniformity. While non-uniformity is quantified differently in each of the examples considered in this thesis, our algorithms can quickly identify these non-uniformities and focus measurements to accurately capture these aspects of the data. On the other hand, non-interactive approaches have high sample complexities for these non-uniform problems, as one needs many measurements in certain regions to achieve suitable accuracy. Formalizing this argument, we show that interactive approaches have significantly better statistical performance than non-interactive ones on these non-uniform problems.

## 1.2 Overview of Results

In this thesis we study four unsupervised learning problems and develop interactive learning algorithms for these problems. The first three problems can all be formalized as *matrix inference problems*; given feedback-driven access to the entries of a  $d \times n$  matrix  $X$  which may be corrupted with noise, we are interested in recovering some structural property of the matrix. We propose interactive algorithms to recover three different structural properties and compare against non-interactive approaches, ones that either observe the entire matrix or a subset of entries acquired prior to any computation. In all three settings, we show that our interactive algorithms can significantly outperform non-interactive ones, in line with the over-arching themes of this thesis.

### 1.2.1 Interactive Subspace Learning

In the subspace learning problem, the data matrix  $X$  corresponds to a collection of  $n$  points in  $d$  dimensions, and the goal is to recover a subspace of  $\mathbb{R}^d$  that effectively captures the dataset. When the data matrix is fully observed, it is well known that principal components analysis (PCA) identifies a subspace that optimally approximates the data matrix [83]. In the missing data setting that we consider here, this is referred to as the matrix completion or the matrix approximation problem [43, 72, 75, 97, 101, 144].

In Chapter 2, we study three versions of the subspace learning problem and propose novel algorithms that employ interactive sampling to obtain strong performance guarantees. We first

consider the setting where the data points lie exactly on a  $r$ -dimensional subspace, which is referred to as the (noiseless) **matrix completion** problem. Our algorithm interactively identifies entries that are highly informative for learning the column space of the matrix and, consequently, it succeeds even when the row space is highly non-uniform (according to a standard definition of non-uniformity), in contrast with non-interactive approaches. We show that one can exactly recover a  $d \times n$  matrix of rank  $r$  from merely  $\Omega((d+n)r \log^2(r))$  matrix entries using an algorithm with running time that is linear in the matrix size,  $\max\{d, n\}$ , with a mild polynomial dependence on the rank  $r$ . In addition to significantly relaxing uniformity assumptions, this algorithm nearly matches the best known sample complexity and is the fastest known algorithm for matrix completion.

We generalize this algorithm to the **tensor completion** problem, where the data is instead a low-rank tensor. We show that a recursive application of our matrix completion algorithm recovers a rank  $r$  order  $T$  tensor  $X \in \mathbb{R}^{\otimes_{i=1}^T d_i}$  using  $\Omega(r^{T-1} T \sum_{i=1}^T d_i \log^2(r))$  tensor entries, which is the best known sample complexity for this problem [111, 132]. As with the algorithm for the matrix case, this algorithm relaxes uniformity assumptions and is extremely fast.

Lastly, we consider the problem of constructing a low rank approximation to a high-rank input matrix from interactively sampled matrix entries. This is referred to as the **matrix approximation** problem. We propose a simple algorithm that truncates the singular value decomposition of a zero-filled version of the input matrix. The algorithm computes an approximation that is nearly as good as the best rank- $r$  approximation to the input matrix using  $O(nr\mu \log^2(n))$  samples, where  $\mu$  is a uniformity parameter on the matrix columns. We eliminate uniformity assumptions on the row space of the matrix while achieving similar statistical and computational performance to non-interactive methods.

We demonstrate the statistical and computational efficiency of all three of these procedures with extensive empirical evaluation. These results appear in the papers [121, 122].

## 1.2.2 Interactive Hierarchical Clustering

We consider a similarity-based clustering formulation where we are given an  $n \times n$  symmetric matrix  $X$  of pairwise similarities between  $n$  objects. In flat clustering problems the goal is to identify a partitioning of the objects so that pairs of objects in the same group have high similarity and pairs of objects in different groups have low similarity. In hierarchical clustering problems, the goal is to identify this partitioning structure at multiple resolutions. We aim to recover hierarchical cluster structures when the similarity matrix  $X$  is only partially observed.

In Chapter 3, we propose interactive learning algorithms for hierarchical clustering from partially observed pairwise similarity information. Our algorithm runs spectral clustering on a subsampled version of the similarity matrix to resolve the larger cluster structure and then focuses measurements to resolve the finer partitions. We show that this algorithm recovers all clusters of size  $\Omega(\log n)$  using  $O(n \log^2 n)$  similarities and runs in  $O(n \log^3 n)$  time for a dataset of  $n$  objects. In comparison, hierarchical spectral clustering on the fully observed similarity matrix achieves

the same resolution but uses all  $O(n^2)$  similarities and runs in  $O(n^2)$  time [16]. This algorithm is most effective when trying to recover both the large clusters at the top of the hierarchy and the small clusters at the bottom of the hierarchy, or, in other words, when the cluster structure is highly non-uniform.

We complement this algorithmic result with an information-theoretic study of the hierarchical clustering problem. The most important result in this study is a necessary condition for *any* non-interactive algorithm to recover a hierarchical clustering. Comparing this necessary condition with the sufficient condition developed by our interactive algorithm, we mathematically certify the statistical advantage offered by interactivity.

We evaluate this algorithm with a detailed empirical study on simulated and real clustering data sets. We compare with several popular clustering algorithms and show that our proposed algorithm does lead to statistical and/or computational improvements in many cases. This algorithm and its analysis appear in the paper [123]. The information-theoretic study is new here.

### 1.2.3 Interactive Latent Tree Metric Learning

In metric learning problems,  $X \in \mathbb{R}^{n \times n}$  is a distance matrix between  $n$  points, so that the  $(i, j)$ th entry is the distance between the  $i$ th and  $j$ th object. Broadly, the goal is to impute distances between points, and this is typically done by embedding the points into some structured metric space. In the instantiation of this problem that we study, we aim to recover a **latent tree metric**, which associates each object to a leaf of some weighted tree and approximates distances between objects via the distance along the tree. This problem is motivated by research in communication networks showing that packet latencies can be well-approximated by latent tree metrics.

In Chapter 4, we present two algorithms that use interactively sampled pairwise distance measurements to construct a latent tree whose path distances approximate those between the objects. Our first algorithm accommodates measurements perturbed by additive noise, while our second considers a novel noise model that captures missing measurements and the datasets deviations from a tree topology. Both algorithms provably use  $O(n \text{ polylog } n)$  pairwise measurements to construct a tree approximation on  $n$  end hosts and run in nearly linear time. We present simulated and real-world experiments to evaluate both algorithms. These results appear in the paper [120].

### 1.2.4 Passive and Interactive Sampling in Normal Means Inference

The last problem we consider does not fall into the matrix inference framework. We study a structured hypothesis testing problem where the goal is to use data generated from a gaussian distribution to identify which vector, out of a finite collection, is the mean vector. We consider algorithms that are given a sensing budget and asked to allocate measurements across the coordinates, where interactive algorithms can make this allocation in a feedback-driven manner.

Our focus is on understanding how structural assumptions about the collection of mean vectors affects statistical performance, and most of the results pertain to non-interactive approaches. Specifically, for any non-interactive allocation strategy, we give necessary and sufficient conditions under which the identification of the mean vector is possible. We show through many concrete examples, that this analysis leads to optimal non-interactive allocation strategies and inference procedures. We also give a concrete example where a simple interactive procedure significantly outperforms all non-interactive ones.

In this chapter, we also initiate a deeper investigation into the design of optimal estimators. In this direction, we give a sufficient condition, which depends on the structure of the collection of vectors, for the exact optimality of the maximum likelihood estimator. We also design a heuristic algorithm for improving on the maximum likelihood estimator in the cases when it is suboptimal. We provide synthetic examples demonstrating the importance of this improvement.

### 1.3 Related Work

In this section we provide a broad summary of related work on interactive learning. Research on interactive learning is extremely diverse, in part due to the intuitive appeal of the learning paradigm, and we cannot hope to cover all of the work here. Instead we focus attention on the theoretical results.

We categorize the research based on types of learning problems addressed:

1. **Classification and Regression:** When focusing on classification or regression problems, interactive approaches are typically referred to as active learning [57]. In active learning, the learner interacts with the dataset by querying for the response or label of data points. There are three ways of realizing this interaction: *pool-based* [62], where the learner has access to a large number of unlabeled examples; *stream-based* [57, 91], where unlabeled examples are fed one-by-one to the learner and it decides to query for a label; and *query synthesis* [8, 9], where the learner can construct examples to be labeled. Most of the recent attention has focused on either pool-based or stream-based active learning, as the third model is fairly unnatural.

The literature on active learning alone is quite vast, but can roughly be categorized along several axes. In the context of binary classification, researchers have considered hypothesis classes ranging from linear separators through the origin [20, 66, 91] to classes of bounded Vapnik-Chervonenkis dimension [30, 102]. The choice of noise model also plays a role, with choices ranging from noise-free or realizable [62, 63, 66, 91] to parameterized noise models [44, 141], to the most general agnostic case [22, 30, 114]. Lastly apart from these works, there is a sequence of papers on bayesian active learning where a prior distribution is placed on the true hypothesis, and query decisions are made through computations involving the posterior [96, 133].

We refer the reader to Hanneke’s comprehensive treatment of the theoretical issues in ac-

tive learning [105]. For a more applied perspective, many algorithmic techniques for active learning are outlined in the survey by Settles [153].

2. **Sequential Decision Making:** In this class of problems, a learner makes a series of actions, possibly based on situational context, and receives reward based on the quality of the choices made, possibly depending on context. The goal broadly is to obtain large rewards, which amounts to learning how to choose high-quality actions. These problems fall into the interactive learning framework because the actions that a learner makes influence the reward feedback provided, and also possibly influence the future situations. Therefore a learner must tradeoff between choosing actions that provide information about the environment and those that provide large rewards.

The simplest version of the sequential decision making problem is the *multi-arm bandit* problem. In this problem, there are a fixed set of actions and no situational context, so the goal reduces to identifying the best fixed action. An excellent survey of results in this line of research is provided by Bubeck and Cesa-Bianchi [38].

Incorporating situational information into the multi-arm bandit framework yields the *contextual bandit* problem. Here the goal now amounts to finding a policy that maps contexts into actions while achieving high levels of reward. A number of recent algorithms address both parametrized [55, 89, 150], where the reward for an action can be reliably predicted based on some features, and agnostic [4, 13, 31, 78, 127], where no features are available, versions of this problem.

Lastly, the most challenging version of the sequential decision making problem is *reinforcement learning*, where the actions of the learner affect not only the reward and feedback, but also the future situation or context. In some models for this problem, we know of algorithms that achieve nearly optimal statistical performance [14]. An overview of the main techniques for reinforcement learning problems is provided by Sutton and Barto [26].

3. **Unsupervised Learning** There are also a plethora of results on interactive learning for unsupervised problems. The majority of these results stem from the statistics and signal processing communities and focus on various forms of hypothesis testing problems. Some more recent results from the machine learning community address more classical unsupervised problems such as clustering and subspace learning.

In the statistics literature, interactive learning is typically referred to as *sequential experimental design* and includes the seminal works of Wald [169], Chernoff [53], and Robbins [146]. The techniques are strikingly similar to those for the multi-arm bandit problem, and indeed both lines of research stem from the initial works of Robbins and Lai [125].

In the signal processing community, interactive learning is typically referred to as *adaptive sensing*, and the typical goal is multiple hypothesis testing from repeated direct or compressive measurements. When individual hypothesis can be queried, the distilled sensing algorithm [107] is known to outperform non-adaptive sampling schemes, and this work has been extended to some structured settings [161]. When compressive measurements can be taken, results under specific structural constraints are known [17, 124], and unstructured lower bounds show that significant performance improvements over non-adaptive procedures are not possible [11].

Interactive approaches have also been considered for more classical unsupervised learning problems, with most of the focus on clustering and kernel learning. A number of algorithms have been proposed for interactive clustering both in hierarchical and flat settings [15, 18, 27, 87], although many of these approaches consider interactive supervision in the form of constraints on the clustering rather than interactivity with object features or similarities as we do. The advent of crowdsourcing platforms has also led to research on learning via interaction with crowds of workers [160]. Lastly, interactive learning is the de facto standard for problems in network tomography, including topology identification [84], topology-aware clustering [56], and other tasks [45].

# Chapter 2

## Interactive Matrix Completion

In this chapter, we develop interactive algorithms for low rank matrix and tensor completion and matrix approximation. In the completion problem, we would like to exactly recover a low rank matrix (or tensor) after observing only a small fraction of its entries. In the approximation problem, rather than exact recovery, we aim to find a low rank matrix that approximates, in a precise sense, the input matrix, which need not be low rank. In both problems, we are only allowed to observe a small number of matrix entries, although these entries can be chosen sequentially and in a feedback-driven manner.

The measure of uniformity in this chapter is the notion of *incoherence* which pervades the matrix completion literature. We show that interactive sampling allows us to significantly relax the incoherence assumption. Previous analyses show that if the energy of the matrix is spread out fairly uniformly across its coordinates, then uniform-at-random samples suffice for completion or approximation. In contrast, our work shows that interactive sampling algorithms can focus measurements appropriately to solve these problems even if the energy is non-uniformly distributed. Handling non-uniformity is essential in a variety of problems involving outliers, for example network monitoring problems with anomalous hosts, or recommendation problems with popular items. This is a setting where non-interactive algorithms fail, as we will show.

We make the following contributions:

1. For the matrix completion problem, we give a simple algorithm that exactly recovers an  $n \times n$  rank  $r$  matrix using at most  $O(nr\mu_0 \log^2(r))$  measurements where  $\mu_0$  is the coherence parameter on the column space of the matrix (Corollary 2.2). This algorithm outperforms all existing results on matrix completion both in terms of sample complexity and in the fact that we place no assumptions on the row space of the matrix. The algorithm is extremely simple, runs in  $\tilde{O}(nr^2)$  time, and can be implemented in one pass over the matrix.
2. We derive a lower bound showing that in the absence of row-space incoherence, *any* non-interactive scheme must see  $\Omega(n^2)$  entries (Theorem 2.3). This concretely demonstrates the power of interactivity in the matrix completion problem.

3. For the tensor completion problem, we show that a recursive application of our matrix completion algorithm can recover an order- $T$ ,  $n \times \dots \times n$  tensor using  $O(nT^2r^{T-1}\mu_0^{T-1}\log^2 r)$  interactively-obtained samples (Theorem 2.1). This algorithm significantly outperforms all existing results on tensor completion and as above, is quite simple.
4. We complement this with a necessary condition for tensor completion under random sampling, showing that our interactive strategy is competitive with *any* approach based on uniform sampling (Theorem 2.4). This is the first sample complexity lower bound for tensor completion, although it is weaker than the lower bound for the matrix completion case in Corollary 2.2.
5. For matrix approximation, we analyze an algorithm that, after an interactive sampling phase, approximates the input matrix by the top  $r$  ranks of an appropriately rescaled zero-filled version of the matrix. We show that with just  $O(nr\mu\log^2(n))$  samples, this approximation is competitive with the best rank  $r$  approximation of the matrix (Theorem 2.5). Here  $\mu$  is a coherence parameter on each column of the matrix; as before we make no assumptions about the row space of the input. Again, this result significantly outperforms existing results on matrix approximation from non-interactively collected samples.

This chapter is organized as follows: we conclude this introduction with some basic definitions and then turn to related work in Section 2.1. The main results for the exact completion problems are given in Section 2.2 while our matrix approximation algorithm and analysis are in Section 2.3. Proofs are provided in Section 2.4 and we provide some simulation that validate our theoretical results in Section 2.5. We conclude the chapter in Section 2.6.

## 2.0.1 Preliminaries

In this chapter, we are interested in recovering, or approximating, a  $d \times n$  matrix  $X$  given a budget of  $M$  observations, where we assume  $d \leq n$ . We denote the columns of  $X$  by  $x_1, \dots, x_n \in \mathbb{R}^d$  and use  $t$  to index the columns. We use  $x_t(i)$  to denote the  $i$ th coordinate of the column  $x_t$ .

We use capital letters to denote subspaces and we overload notation by using the same symbol to refer to a subspace and any orthonormal basis for that subspace. Specifically, if  $U \subset \mathbb{R}^d$  is a subspace with dimension  $r$ , we may use  $U$  to refer to a  $d \times r$  matrix whose columns are an orthonormal basis for that subspace. We use  $U^\perp$  to denote the orthogonal complement to the subspace  $U$  and  $\mathcal{P}_U$  to refer to the orthogonal projection operator onto  $U$ .

As we are dealing with missing data and sampling, we also need some notation for subsampling operations. Let  $[d]$  denote the set  $\{1, \dots, d\}$  and let  $\Omega$  be a list of  $m$  values from  $[d]$ , possibly with duplicates (One can think of  $\Omega$  as a vector in  $[d]^m$  and  $\Omega(j)$  is the  $j$ th coordinate of this vector). Given such a list  $\Omega$ , we use two different subsampling operations:  $x_\Omega \in \mathbb{R}^m$  is the vector formed putting  $x(i)$  in the  $j$ th coordinate if  $\Omega(j) = i$  and  $\mathcal{R}_\Omega x$  is a zero-filled rescaled version of  $x$  with  $\mathcal{R}_\Omega x(i) = 0$  if  $i \notin \Omega$  and  $\mathcal{R}_\Omega x(i) = dx(i)/|\Omega|$  if  $i \in \Omega$ .

For a  $r$ -dimensional subspace  $U \subset \mathbb{R}^d$ ,  $U_\Omega \in \mathbb{R}^{m \times r}$  is a *matrix* formed by doing a similar



subsampling operation to the *rows* of any orthonormal basis for the subspace  $U$ , e.g. the  $j$ th row of  $U_\Omega$  is the  $i$ th row of  $U$  if  $\Omega(j) = i$ . Note that  $U_\Omega$ , and even the span of the columns of  $U_\Omega$ , may not be uniquely defined, as they both depend on the choice of basis for  $U$ . Nevertheless, we will use  $\mathcal{P}_{U_\Omega}$  to denote the projection onto the span of any single set of columns constructed by this subsampling operation.

These definitions extend to the tensor setting with slight modifications. Let  $\mathbb{X} \in \mathbb{R}^{n_1 \times \dots \times n_T}$  denote an order  $T$  tensor with canonical decomposition:

$$\mathbb{X} = \sum_{k=1}^r a_k^{(1)} \otimes a_k^{(2)} \otimes \dots \otimes a_k^{(T)} \quad (2.1)$$

where  $\otimes$  is the outer product. Define  $\text{rank}(\mathbb{X})$  to be the smallest value of  $r$  that establishes this equality. Note that the vectors  $\{a_k^{(t)}\}_{k=1}^r$  need not be orthogonal, nor even linearly independent.

We then use the `vec` operation to unfold a tensor into a vector and define the inner product  $\langle x, y \rangle = \text{vec}(x)^T \text{vec}(y)$ . For a subspace  $U \subset \mathbb{R}^{\otimes n_i}$ , we write it as a  $(\prod n_i) \times d$  matrix whose columns are  $\text{vec}(u_i)$ ,  $u_i \in U$ . We then define projections and subsampling as in the vector case.

We will frequently work with the truncated singular value decomposition (SVD) of  $X$  which is given by zero-ing out its smaller singular values. Specifically, write  $X = U_r \Sigma_r V_r^T + U_{-r} \Sigma_{-r} V_{-r}^T$  where  $[U_r, U_{-r}]$  (respectively  $[V_r, V_{-r}]$ ) forms an orthonormal matrix and  $\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r)$ ,  $\Sigma_{-r} = \text{diag}(\sigma_{r+1}, \dots, \sigma_d)$  are diagonal matrices with  $\sigma_1 \geq \dots \geq \sigma_r \geq \sigma_{r+1} \geq \dots \geq \sigma_d$ . The truncated singular value decomposition is  $X_r = U_r \Sigma_r V_r^T$ , which is the best rank- $r$  approximation to  $X$  both in Frobenius and spectral norm [83].

In the matrix completion problem, where we aim for exact recovery, we require that  $X$  has rank at most  $r$ , meaning that  $\sigma_{r+1} = \dots = \sigma_n = 0$ . Thus  $X = X_r$ , and our goal is to recover  $X_r$  exactly from a subset of entries. Specifically, we focus on the 0/1 loss; given an estimator  $\hat{X}$  for  $X$ , we would like to bound the probability of error:

$$R_{01}(\hat{X}) \triangleq \mathbb{P}(\hat{X} \neq X). \quad (2.2)$$

In the approximation problem, we relax the low rank assumption but are only interested in approximating the action of  $X_r$ . The goal is to find a rank  $r$  matrix  $\hat{X}$  that minimizes:

$$R(\hat{X}) = \|X - \hat{X}\|_F.$$

The matrix  $X_r$  is the global minimizer (subject to the rank- $r$  constraint), and our task is to approximate this low rank matrix effectively. Specifically, we will be interested in finding matrices  $\hat{X}$  that satisfy excess risk bounds of the form:

$$R(\hat{X}) \triangleq \|X - \hat{X}\|_F \leq \|X - X_r\|_F + \epsilon \|X\|_F \quad (2.3)$$

Rescaling the excess risk term by  $\|X\|_F$  is a form of normalization that has been used before in the matrix approximation literature [75, 76, 92, 149]. While bounds of the form  $(1+\epsilon)\|X - X_r\|_F$

may seem more appropriate when the bottom ranks are viewed as noise term, achieving such a bound seems to require highly accurate approximations of the SVD of the input matrix [77], which is not possible given the extremely limited number of observations in our setting. Equation 2.3 can be interpreted by dividing by  $\|X\|_F$ , which shows that  $\hat{X}$  captures almost as large a fraction of the energy of  $X$  as  $X_r$  does.

Apart from the observation budget  $M$  and the approximation rank  $r$ , the other main quantity governing the difficulty of these problems is the subspace coherence parameter. For a  $r$  dimensional subspace  $U$  of  $\mathbb{R}^d$ , define

$$\mu(U) = \frac{d}{r} \max_{i \in [d]} \|\mathcal{P}_U e_i\|_2^2,$$

which is a standard measure of subspace coherence [144]). The quantity  $\mu_0 \triangleq \mu(U_r)$ , which is bounded between 1 and  $d/r$ , measures the correlation between the column space  $U_r$  and any standard basis element. When this correlation is small, the energy of the matrix is spread out fairly uniformly across the rows of the matrix, although it can be non-uniformly distributed across the columns. We use the column-space coherence  $\mu_0$  instead of the row-space analog, and we will see that the parameter  $\mu_0$  controls the sample complexity of our procedure.

Such an incoherence assumption does not translate appropriately to the approximate recovery problem, since the matrix is no longer low rank, but some measure of uniformity is still necessary. We parameterize the problem by a quantity related to the usual definition of incoherence:

$$\mu = \max_{t \in [n]} \frac{d \|x_t\|_\infty^2}{\|x_t\|_2^2},$$

which is the maximal column coherence. Here, we make no stochastic assumptions, but notice that this is a restriction on the higher ranks of the matrix. We also make no assumptions about the row space of the matrix<sup>1</sup>.

## 2.1 Related Work

The literature on low rank matrix completion and approximation is extremely vast and we do not attempt to cover all of the existing ideas. Instead, we focus on the most relevant lines of work to our specific problems. We briefly mention some related work on adaptive sensing.

### 2.1.1 Related work on Matrix and Tensor Completion

Due to its widespread applicability, the matrix completion problem has received considerable attention in recent years. A series of papers [42, 43, 50, 97, 144] establish that  $\Omega(nr\mu'_0 \log^2(n))$  randomly drawn samples are sufficient for the nuclear norm minimization program to exactly

<sup>1</sup>As before this could equivalently be the column space with assumption on the maximal row coherence.

identify an  $n \times n$  matrix with rank  $r$ . Here  $\mu'_0 \triangleq \max\{\mu(U_r), \mu(V_r)\}$  is the coherence parameter, which measures the uniformity of *both* the row and column spaces of the matrix. Candès and Tao [43] show that nuclear norm minimization is essentially optimal with a  $\Omega(nr\mu_0 \log(n))$  lower bound for uniform-at-random sampling. In contrast, the guarantee for our interactive procedure scales linearly on  $\mu_0 = \mu(U_r)$ , so our algorithm succeeds even when the row space is highly coherent. This is a regime where non-interactive provably fail, as we will show.

There is also a line of work analyzing alternating minimization-style procedures for the matrix completion problem [106, 112, 116]. While the alternating minimization algorithm is a more elegant computational approach, the best sample complexity bounds to-date are either worse by at least a cubic factor in the rank  $r$  or have undesirable dependence on the matrix condition number [116]. In practice however, alternating minimization performs as well as nuclear norm minimization, so this sub-optimality appears to be an artifact of the analysis.

In a similar spirit to our work, Chen *et al.* [52] developed an interactive algorithm which succeeds in the absence of row-space incoherence using  $\Omega(nr\mu_0 \log^2(n))$  samples. In comparison, we operate under the same assumption but achieve an improved sample complexity of  $\Omega(nr\mu_0 \log^2(r))$ . A recent paper of Jin and Zhu [113] further improves slightly on this bound, achieving  $\Omega(nr \log(r))$  sample complexity, but they assume that both the row and column space are incoherent. Interestingly, their algorithm uses non-interactive but non-uniform sampling.

Tensor completion, a natural generalization of matrix completion, is less studied than the matrix case. One challenge stems from the NP-hardness of computing most tensor decompositions, pushing researchers to study alternative structure-inducing norms in lieu of the nuclear norm [93, 132, 162, 163, 164, 173]. Of these, only Mu *et al.* [132] and Yuan and Zhang [173] provide sample complexity bounds for the noiseless setting. Mu *et al.* [132] show that  $\Omega(rn^{T/2})$  random linear measurements suffice to recover a rank  $r$  order- $T$  tensor. Yuan and Zhang [173] instead show that  $\Omega(r^{1/2}n^{3/2})$  entries suffice to recover a rank  $r$  third-order tensor with incoherent subspaces, provided the rank is small. In contrast, the sample complexity of our algorithm is *linear* in dimension  $n$ , improving significantly on these non-interactive results.

## 2.1.2 Related work on Matrix Approximation

A number of authors have studied matrix completion with noise and under weaker assumptions. The most prominent difference between our work and all of these is a relaxation of the main incoherence assumptions. Both Candès and Plan [41], and Keshavan *et al.* [117] require that both the row and column space of the matrix of interest is highly incoherent. Negahban and Wainwright [134] instead use a notion of *spikiness*, but that too places assumptions on the row space of interest. Koltchinskii *et al.* [119] consider matrices with bounded entries, which is related to the spikiness assumption. In comparison, our results make essentially no assumptions about the row space, leading to substantially more generality. This is the thesis of this work; one can eliminate row space assumptions (uniformity assumptions) in matrix recovery problems through interactive sampling.

Another close line of work is on matrix sparsification [1, 2, 12]. Here, the goal is to zero out many entries of a matrix while preserving global properties such as the principal subspace. The main difference from matrix completion is that the entire matrix is observed, which allows one to relax incoherence assumptions. The only result from this line that does not require knowledge of the matrix is a random sampling scheme of Achlioptas and McSherry [1], but it is only competitive with matrix completion when the input has entries of fairly constant magnitude [119]. Interestingly, this requirement is essentially the same as the spikiness assumption [134] and the bounded magnitude assumption [119] in the matrix completion literature.

Several techniques have been proposed for matrix approximation in the fully observed setting, optimizing computational complexity or other objectives. A particularly relevant series of papers is on the column subset selection (CSS) problem, where the span of several judiciously chosen columns is used to approximate the principal subspace. One of the best approaches involves sampling columns according to the statistical leverage scores, which are the norms of the rows of the  $n \times r$  matrix formed by the top  $r$  right singular vectors [36, 37, 77]. Unfortunately, this strategy does not seem to apply in the missing data setting, as the distribution used to sample columns – which are subsequently used to approximate the matrix – depends on the unobserved input matrix. Approximating this distribution seems to require a very accurate estimate of the matrix itself, and this initial estimate would suffice for the matrix approximation problem. This difficulty also arises with volume sampling [100], another popular approach to CSS; the sampling distribution depends on the input matrix and we are not aware of strategies for approximating this distribution in the missing data setting.

In terms of interactive sampling, a number of methods for recovery of sparse, structured, signals have been shown to outperform non-interactive methods [17, 107, 124, 130, 161]. While having their share of differences, these methods can all be viewed as either binary search or local search methods, that iteratively discard irrelevant coordinates and focus measurements on the remainder. In particular, these methods rely heavily on the sparsity and structure of the input signal, and extensions to other settings have been elusive. While a low rank matrix is sparse in its eigenbasis, the search-style techniques from the signal processing community do not seem to leverage this structure effectively and these approaches do not appear to be applicable to our setting.

Some of these interactive sampling efforts focus specifically on recovering or approximating highly structured matrices, which is closely related to our setting. Tanczos and Castro [161] and Balakrishnan *et al.* [17] consider variants of biclustering, which is equivalent to recovering a rank-one binary matrix from noisy observations. Singh *et al.* [158] recover noisy ultrametric matrices while in Chapter 3, we use a similar idea to find hierarchical clustering from interactively sampled similarities. All of these results can be viewed as matrix completion or approximation, but impose significantly more structure on the target matrix than we do here. For this reason, many of these algorithmic ideas also do not appear to be useful in our setting.

---

**Algorithm 1** Interactive Matrix Completion ( $X \in \mathbb{R}^{d \times n}, m$ )

---

1. Let  $\tilde{U} = \emptyset$ .
  2. Randomly draw entries  $\Omega \subset [d]$  of size  $m$  uniformly with replacement.
  3. For each column  $x_t$  of  $X$  ( $t \in [N]$ ):
    - (a) If  $\|x_{t\Omega} - \mathcal{P}_{\tilde{U}_\Omega} x_{t\Omega}\|_2^2 > 0$ :
      - i. Fully observe  $x_t$  and add to  $\tilde{U}$  (orthogonalize  $\tilde{U}$ ).
      - ii. Randomly draw a new set  $\Omega$  of size  $m$  uniformly with replacement.
    - (b) Otherwise  $\hat{x}_t \leftarrow \tilde{U}(\tilde{U}_\Omega^T \tilde{U}_\Omega)^{-1} \tilde{U}_\Omega x_{t\Omega}$ .
  4. Return  $\hat{X}$  with columns  $\hat{x}_t$ .
- 

## 2.2 Matrix and Tensor Completion

In this section we develop the main theoretical guarantees on the exact low-rank completion problems. We first develop our interactive algorithm for matrices and tensors and state their main performance guarantee. We then turn to several necessary conditions for these problems.

Our procedure for the matrix case, whose pseudocode is displayed in Algorithm 1, streams the columns of the matrix  $X$  into memory and iteratively adds directions to an estimate for the column space of  $X$ . The algorithm maintains a subspace  $U$  and, when processing the  $t$ th column  $x_t$ , estimates the norm of  $\mathcal{P}_{U^\perp} x_t$  using only a few entries of  $x_t$ . We will ensure that, with high probability, this estimate will be non-zero if and only if  $x_t$  contains a new direction. If the estimate is non-zero, the algorithm asks for the remaining entries of  $x_t$  and adds the new direction to the subspace  $U$ . Otherwise,  $x_t$  lies in  $U$  and we will see that the algorithm already has sufficient information to complete the column  $x_t$ .

Therefore, the key ingredient of the algorithm is the estimator for the projection onto the orthogonal complement of the subspace  $U$ . This quantity is estimated as follows. Using a list of  $m$  locations  $\Omega$  sampled uniformly with replacement from  $[d]$ , we down-sample both  $x_t$  and an orthonormal basis  $U$  to  $x_{t\Omega}$  and  $U_\Omega$ . We then use  $\|x_{t\Omega} - \mathcal{P}_{U_\Omega} x_{t\Omega}\|_2^2$  as our estimate. It is easy to see that this estimator leads to a test with one-sided error, since the estimator is exactly zero if  $x_t \in U$ . In our analysis, we establish a relative-error deviation bound, which allows us to apply this test in our algorithm.

A subtle but critical aspect of the algorithm is the choice of  $\Omega$ . The list  $\Omega$  always has  $m$  elements, and each element is sampled uniformly with replacement from  $[d]$ . More importantly, we only resample  $\Omega$  when we add a direction to  $U$ . This ensures that the algorithm does not employ too much randomness, which would lead to an undesirable logarithmic dependence on  $n$ .

For tensors, the algorithm becomes recursive in nature. At the outer level of the recursion, the algorithm maintains a candidate subspace  $\mathcal{U}$  for the mode  $T$  subtensors  $\mathbb{X}_i^{(T)}$ . For each of these subtensors, we test whether  $\mathbb{X}_i^{(T)}$  lives in  $\mathcal{U}$  and recursively complete that subtensor if it does

---

**Algorithm 2** Interactive Tensor Completion ( $\mathbb{X}, \{m_t\}_{t=1}^{T-1}$ )

---

1. If  $\mathbb{X}$  is just a vector, sample  $\mathbb{X}$  entirely and return it.
  2. Let  $\mathcal{U} = \emptyset$ .
  3. Randomly draw entries  $\Omega \subset \prod_{t=1}^{T-1} [n_t]$  uniformly with replacement w. p.  $m_{T-1} / \prod_{t=1}^{T-1} n_t$ .
  4. For each mode- $T$  subtensor  $\mathbb{X}_i^{(T)}$  of  $\mathbb{X}$ ,  $i \in [n_T]$ :
    - (a) If  $\|\mathbb{X}_{i\Omega}^{(T)} - \mathcal{P}_{\mathcal{U}\Omega} \mathbb{X}_{i\Omega}^{(T)}\|_2^2 > 0$ :
      - i.  $\hat{\mathbb{X}}_i^{(T)} \leftarrow$  recurse on  $(\mathbb{X}_i^{(T)}, \{m_t\}_{t=1}^{T-1})$
      - ii.  $\mathbb{U}_i \leftarrow \frac{\mathcal{P}_{\mathcal{U}^\perp} \hat{\mathbb{X}}_i^{(T)}}{\|\mathcal{P}_{\mathcal{U}^\perp} \hat{\mathbb{X}}_i^{(T)}\|} \cdot \mathcal{U} \leftarrow \mathcal{U} \cup \mathbb{U}_i$ .
    - (b) Otherwise  $\hat{\mathbb{X}}_i^{(T)} \leftarrow \mathcal{U}(\mathcal{U}_\Omega^T \mathcal{U}_\Omega)^{-1} \mathcal{U}_\Omega \mathbb{X}_{i\Omega}^{(T)}$
  5. Return  $\hat{\mathbb{X}}$  with mode- $T$  subtensors  $\hat{\mathbb{X}}_i^{(T)}$ .
- 

not. Once we complete the subtensor, we add it to  $\mathcal{U}$  and proceed at the outer level. When the subtensor itself is just a column, we observe the columns in its entirety.

Turning to the performance guarantees for these algorithms, we first bound the probability of error for the tensor completion algorithm (Algorithm 2). The guarantee for Algorithm 1 is just a specialization of this result to the order-two case. The following result is based on an analysis of the test statistic and the reconstruction procedure in Algorithm 2. See Section 2.4 for the proof.

**Theorem 2.1.** *Let  $\mathbb{X} = \sum_{i=1}^r \otimes_{t=1}^T a_j^{(t)}$  be a rank  $r$  order- $T$  tensor with subspaces  $A^{(t)} = \text{span}(\{a_j^{(t)}\}_{j=1}^r)$ . Suppose that all of  $A^{(1)}, \dots, A^{(T-1)}$  have coherence bounded above by  $\mu_0$ . For any  $\delta \in (0, 1)$ , Algorithm 2 has  $R_{01}(\hat{\mathbb{X}}) \leq \delta$  provided that we set:*

$$m_t \geq 32Tr^t \mu_0^t \log^2(10rT/\delta). \quad (2.4)$$

With this choice, the total number of samples used is:

$$32 \left( \sum_{t=1}^T n_t \right) r^{T-1} \mu_0^{T-1} T \log(10rT/\delta). \quad (2.5)$$

The running time of the algorithm is:

$$\tilde{O} \left( r^2 \left( \prod_{t=1}^{T-1} n_t \right) + r^T \sum_{t=1}^T n_t + Tr^{2+T} \right), \quad (2.6)$$

when we treat  $\mu_0$  as a constant and ignore logarithmic factors.

In the special case of a  $n \times \dots \times n$  tensor of order  $T$ , the algorithm succeeds with probability at least  $1 - \delta$  using  $\Omega(nr^{T-1/2} \mu_0^{T-1} T^2 \log(Tr/\delta))$  samples, exhibiting a linear dependence on the tensor dimensions. In comparison, all guarantees for tensor completion we are aware of have super-linear dependence on the tensor dimension  $n$  [132, 173]. To our knowledge, the best

known sample complexity is  $O(r^{1/2}n^{3/2})$  for exact recovery of a  $n \times n \times n$  third-order tensor of rank  $r$  [173]. An alternating minimization procedure is known to achieve  $O(r^5n^{3/2})$  sample complexity for this problem [111].

In the noiseless scenario, one can unfold the tensor into a  $n_1 \times \prod_{t=2}^T n_t$  matrix and apply any matrix completion algorithm. Unfortunately, without exploiting the additional tensor structure, this approach will scale with  $\prod_{t=2}^T n_t$ , which is similarly much worse than our guarantee. Note that the naïve procedure that does not perform the recursive step has sample complexity scaling with the product of the dimensions and is therefore much worse than our algorithm.

The most obvious specialization of Theorem 2.1 is to the matrix completion problem. Pseudocode for this algorithm is provided in Algorithm 1

**Corollary 2.2.** *Let  $X \in \mathbb{R}^{d \times n}$  have rank  $r$  and column space  $U$  with coherence  $\mu(U) \leq \mu_0$ . Then for any  $\delta \in (0, 1)$ , the output of Algorithm 2 has risk  $R_{01}(\hat{X}) \leq \delta$  provided that:*

$$m \geq 32r\mu_0 \log^2(10r^2/\delta). \quad (2.7)$$

*The sample complexity is  $dr + nm$  and the running time is  $O(nmr + r^3m + dr^2)$ .*

To the best of our knowledge, this result provides the strongest guarantee for the matrix completion problem. The vast majority of results require both incoherent row and column spaces and are therefore considerably more restrictive than ours [42, 43, 50, 97, 144]. For example, Recht shows that by solving the nuclear norm minimization program, one can recover  $X$  exactly, provided that the number of measurements exceeds  $32(d+n)r \max\{\mu'_0, \mu_1^2\} \log^2(n)$  where recall that  $\mu'_0$  upper bounds the coherence of both the row and column space, and  $\mu_1$  provides another incoherence-type assumption (which can be removed [50]). Our result improves on his not only in relaxing the row space incoherence assumption, but also in terms of sample complexity, as we remove the logarithmic dependence on problem dimension.

As another example, Gittens [95] showed that Nystrom method can recover a rank  $r$  matrix from randomly sampling  $O(r \log r)$  columns. While his result matches ours in terms of sample complexity, he analyzes positive-semidefinite matrices with incoherent principal subspace, which translates to assuming that both row and column spaces are incoherent. Again, in relaxing this assumption, our result is substantially more general.

We mention that the two-phase algorithm of Chen *et al.* [52] based on local coherence sampling allows for coherent row spaces. Their algorithm requires  $O((n+d)r\mu_0 \log(n))$  samples which is weaker than our guarantee in that it has a slightly super-linear dependence on problem dimension. An interesting consequence of Corollary 2.2 is that the amortized number of samples per column is *completely independent* of the problem dimension.

Regarding computational considerations, the algorithm operates in one pass over the columns, and need only store the matrix in condensed form, which requires  $O((n+d)r)$  space. Specifically, the algorithm maintains a (partial) basis for column space and the coefficients for representing each column by that basis, which leads to an optimally condensed representation. Moreover, the computational complexity of the algorithm is *linear* in the matrix dimensions  $d, n$  with mild

polynomial dependence on the rank  $r$ . For this run-time analysis, we work in a computational model where accessing any entry of the matrix is a constant-time operation, which allows us to circumvent the  $\Omega(dn)$  time it would otherwise take to read the input. In comparison, the two standard algorithms for matrix completion, the iterative Singular Value Thresholding Algorithm [40] and alternating least-squares [106, 112], are significantly slower than Algorithm 2, not only due to their iterative nature, but also in per-iteration running time.

## 2.2.1 Necessary conditions for non-interactive sampling

In this section we prove a number of lower bounds for matrix and tensor completion for non-interactive sampling procedures. Note that a parameter counting argument shows that interactive sampling requires  $\Omega(r \sum_{t=1}^n n_t)$  samples. Each entry of a rank  $r$  tensor can be expressed as a polynomial of the vectors in the canonical decomposition, so the observations lead to a polynomial system in  $r \sum_{t=1}^T n_t$  variables. If  $M < r \sum_{t=1}^T n_t - T \binom{r}{2}$  (there are  $T \binom{r}{2}$  orthonormality constraints), then this system is underdetermined, and since it has one solution, it must have infinitely many, so that recovery is impossible. Our algorithm matches this lower bound in its dependence on the tensor dimensions, but is polynomially worse in terms of the rank  $r$ . However for the matrix case, Corollary 2.2 shows that our matrix completion algorithm is nearly optimal, disagreeing only in its dependence on the column incoherence parameter and logarithmic factors. In this section we will show that non-interactive sampling has much more stringent necessary conditions.

Our first result is a necessary condition against non-interactive sampling for the matrix completion problem when the row space is highly coherent. We show that if the matrix has coherent row space, then any non-interactive scheme followed by any recovery procedure requires  $\Omega(dn)$  samples to recover a  $d \times n$  matrix  $X$ .

To formalize our lower bound we fix a sampling budget  $M$  and consider an estimator to be a sampling distribution  $q$  over  $\{(i, j) | i \in [d], j \in [n]\}^M$  and a (possibly randomized) function  $f : \{(\Omega, X_\Omega)\} \rightarrow \mathbb{R}^{d \times n}$  that maps a set of indices and values to a  $d \times n$  matrix. Let  $\mathcal{Q}(M)$  denote the set of all such sampling distributions and let  $\mathcal{F}$  denote the set of all such estimators. Lastly let  $\mathcal{X}(d, n, r, \mu_0)$  denote the set of all  $d \times n$  rank  $r$  matrices with column incoherence at most  $\mu_0$ . We consider the minimax probability of error:

$$R^*(d, n, r, \mu_0, M) = \inf_{f \in \mathcal{F}} \inf_{q \in \mathcal{Q}(M)} \sup_{X \in \mathcal{X}(d, n, r, \mu_0)} \mathbb{P}_{\Omega \sim q} [f(\Omega, X_\Omega) \neq X]$$

where the probability also accounts for potential randomness in the estimator  $f$ . Note that since we make no assumptions about the distribution  $q$  other than excluding interactive distributions, this setup subsumes essentially all non-interactive sampling strategies including uniform-at-random, deterministic, and distributions sampling entire columns. The one exception is the Bernoulli sampling model, where each entry  $(i, j)$  is observed with probability  $q_{ij}$  independently of all other entries, although we believe a similar lower bound holds there.



The following theorem lower bounds success probability of any non-interactive strategy and consequently gives a necessary condition on the sample complexity.

**Theorem 2.3.** *The minimax risk  $R^*$  satisfies:*

$$R^*(d, n, r, \mu_0, M) \geq \frac{1}{2} - \left\lceil \frac{M}{(1 - \frac{r-1}{r\mu_0})d} \right\rceil \frac{1}{2(n-r)}, \quad (2.8)$$

which approaches  $1/2$  whenever:

$$M = o\left(\left((dn - dr)\left(1 + \frac{1}{r\mu_0} - \frac{1}{\mu_0}\right)\right)\right). \quad (2.9)$$

As a concrete instantiation of the theorem, if  $\mu_0$  is bounded from below by any constant  $c > 1$  (which is possible whenever  $r \leq d/c$ ), then the bound approaches  $1/2$  whenever  $M = o(d(n - r))$ . Thus all non-interactive algorithms must have sample complexity that is quadratic in the problem dimension. In contrast, Corollary 2.2 ensures that Algorithm 2 has nearly linear sample complexity, which is a significant improvement over non-interactive algorithms.

The literature contains several other necessary conditions on the sample complexity for matrix completion. A simple argument shows that without any form of incoherence, one requires  $\Omega(dn)$  samples to recover even a rank one matrix that is non-zero in just one entry. This argument also applies to interactive sampling strategies and shows that some measure of incoherence is necessary. With both row and column incoherence, but under uniform sampling, Candes and Tao [43] prove that  $\Omega(\mu'_0 nr \log(n))$  observations are necessary to recover a  $n \times n$  matrix.

One can relax the incoherence assumption by non-uniform non-interactive sampling, although the sampling distribution is matrix-specific as it depends on the local coherence structure [52]. Unfortunately, one cannot compute the appropriate sampling distribution, before taking any measurements. Our result shows that in the absence of row-space incoherence, there is no universal non-interactive sampling scheme that can achieve a non-trivial sample complexity. Thus interactivity is necessary to relax the incoherence assumption in completion problems.

Turning to necessary conditions for tensor completion, we adapt the proof of Candes and Tao [43] to this setting and establish the following lower bound for uniform sampling:

**Theorem 2.4.** *Fix  $1 \leq m, r \leq \min_t n_t$  and  $\mu_0 > 1$ . Fix  $0 < \delta < 1/2$  and suppose that we do not have the condition:*

$$-\log\left(1 - \frac{m}{\prod_{i=1}^T n_i}\right) \geq \frac{\mu_0^{T-1} r^{T-1}}{\prod_{i=2}^T n_i} \log\left(\frac{n_1}{2\delta}\right) \quad (2.10)$$

*Then there exist infinitely many pairs of distinct  $n_1 \times \dots \times n_T$  order- $T$  tensors  $\mathbb{X} \neq \mathbb{X}'$  of rank  $r$  with coherence parameter  $\leq \mu_0$  such that  $\mathcal{P}_\Omega(\mathbb{X}) = \mathcal{P}_\Omega(\mathbb{X}')$  with probability at least  $\delta$ . Each entry is observed independently with probability  $p = \frac{m}{\prod_{i=1}^T n_i}$ .*

Theorem 2.4 implies that as long as the right hand side of Equation 2.10 is at most  $\epsilon < 1$ , and:

$$m \leq n_1 r^{T-1} \mu_0^{T-1} \log\left(\frac{n_1}{2\delta}\right) (1 - \epsilon/2) \quad (2.11)$$

---

**Algorithm 3** Low Rank Approximation  $(X, m_1, m_2)$ 

---

1. Pass 1: For each column, observe  $\Omega_t$  of size  $m_1$  uniformly at random with replacement and estimate  $\hat{c}_t = \frac{d}{m_1} \|x_{t, \Omega_t}\|_2^2$ . Estimate  $\hat{f} = \sum_t \hat{c}_t$ .
  2. Pass 2: Set  $\tilde{X} = 0 \in \mathbb{R}^{d \times n}$ .
    - (a) For each column  $x_t$ , sample  $m_{2,t} = m_2 n \hat{c}_2 / \hat{f}$  observations  $\Omega_{2,t}$  uniformly at random with replacement.
    - (b) Update  $\tilde{X} = \tilde{X} + (\mathcal{R}_{\Omega_{2,t}} x_t) e_t^T$ .
  3. Compute the SVD of  $\tilde{X}$  and output  $\hat{X}$  which is formed by the top- $r$  ranks of  $\tilde{X}$ .
- 

then with probability at least  $\delta$  there are infinitely many matrices that agree on the observed entries. The expected number of samples observed is  $m$ . This gives a necessary condition on the number of samples required for tensor completion. Comparing with Theorem 2.1 shows that our procedure outperforms any non-interactive procedure in its dependence on the tensor dimensions, as our bound does not include a  $\log(n)$  factor. Note that our guarantee matches the polynomial terms in this lower bound in its dependence on  $n, r, \mu_0$ , although the dependence on the tensor order  $T$  is better here.

## 2.3 Matrix Approximation

For the matrix approximation problem, we propose an interactive sampling algorithm to obtain a low-rank approximation to  $X$ . The algorithm (see Algorithm 3 for pseudocode) makes two passes through the columns of the matrix. In the first pass, it subsamples each column uniformly at random and estimates each column norm and the matrix Frobenius norm. In the second pass, the algorithm samples additional observations from each column, and for each  $t$ , places the rescaled zero-filled vector  $\mathcal{R}_{\Omega_{2,t}} x_t$  into the  $t$ th column of a new matrix  $\tilde{X}$ , which is a preliminary estimate of the input,  $X$ . Once the initial estimate  $\tilde{X}$  is computed, the algorithm zeros out all but the top  $r$  ranks of  $\tilde{X}$  to form  $\hat{X}$ . We will show that  $\hat{X}$  has low excess risk, when compared with the best rank- $r$  approximation,  $X_r$ .

A crucial feature of the second pass is that the number of samples per column is proportional to the squared norm of that column. Of course this sampling strategy is only possible if the column norms are known, motivating the first pass of the algorithm, where we estimate precisely this sampling distribution. This feature allows the algorithm to tolerate highly non-uniform column norms, as it focuses measurements on high-energy columns, and leads to significantly better approximation. This idea has been used before, although only in the exactly low-rank case [52].

For the main performance guarantee, we only assume that the matrix has incoherent columns, that is  $d \|x_t\|_\infty^2 / \|x_t\|_2^2 \leq \mu$  for each column  $x_t$ . In particular we make no additional assumptions about the high-rank structure of the matrix. We have the following theorem:

**Theorem 2.5.** Set  $m_1 \geq 32\mu \log(n/\delta)$  and assume  $n \geq d$  and that  $X$  has  $\mu$ -incoherent columns. With probability  $\geq 1 - 2\delta$ , Algorithm 3 computes an approximation  $\hat{X}$  such that:

$$\|X - \hat{X}\|_F \leq \|X - X_r\|_F + \|X\|_F \left( 6\sqrt{\frac{r\mu}{m_2}} \log\left(\frac{d+n}{\delta}\right) + \left( 6\sqrt{\frac{r\mu}{m_2}} \log\left(\frac{d+n}{\delta}\right) \right)^{1/2} \right)$$

using  $n(m_1 + m_2)$  samples. In other words, the output  $\hat{X}$  satisfies  $\|X - \hat{X}\|_F \leq \|X - X_r\|_F + \epsilon\|X\|_F$  with probability  $\geq 1 - 2\delta$  and with sample complexity:

$$32n\mu \log(n/\delta) + \frac{576}{\epsilon^4} nr\mu \log^2\left(\frac{d+n}{\delta}\right). \quad (2.12)$$

The proof is deferred to Section 2.4. The theorem shows that the matrix  $\hat{X}$  serves as nearly as good an approximation to  $X$  as  $X_r$ . Specifically, with  $O(nr\mu \log^2(d+n))$  observations, one can compute a suitable approximation to  $X$ . The running time of the algorithm is dominated by the cost of computing the truncated SVD, which is at most  $O(d^2n)$ .

While the dependence between the number of samples and the problem parameters  $n, r$ , and  $\mu$  is quite mild and matches existing matrix completion results, the dependence on the error  $\epsilon$  in Equation 2.12 seems undesirable. This dependence arises from our translation of a bound on  $\|\tilde{X} - X\|_2$  into a bound on  $\|\hat{X} - X\|_F$ , which results in the  $m_2^{-1/4}$ -dependence in the error bound. We are not aware of better results in the general setting, but a number of tighter translations are possible under various assumptions. We mention just two such results here.

**Proposition 2.6.** Under the same assumptions as Theorem 2.5, suppose further that  $X$  has rank at most  $r$ . Then with probability  $\geq 1 - 2\delta$ :

$$\|X - \hat{X}\|_F \leq 20\|X\|_F \sqrt{\frac{r\mu}{m_2}} \log\left(\frac{d+n}{\delta}\right)$$

This proposition tempers the dependence on the error  $\epsilon$  from  $1/\epsilon^4$  to  $1/\epsilon^2$  in the event that the input matrix has rank at most  $r$ . This gives a relative error guarantee for Algorithm 3 on the matrix completion problem, which improves on the one implied by Theorem 2.5. Note that this guarantee is weaker than Corollary 2.2, but Algorithm 3 is much more robust to relaxations of the low rank assumption as demonstrated in Theorem 2.5.

A similarly mild dependence on  $\epsilon$  can be derived under the assumption that  $X = A + R$ ,  $A$  has rank  $r$  and  $R$  is some perturbation, which has the flavor of existing noisy matrix completion results. Here, it is natural to recover the parameter  $A$  rather than the top  $r$  ranks of  $X$  and we have the following parameter recovery guarantee for Algorithm 3:

**Proposition 2.7.** Let  $X = A + R$  where  $A$  has rank at most  $r$ . Suppose further that  $X$  has  $\mu$ -incoherent columns and set  $m_1 \geq 32\mu \log(n/\delta)$ . Then with probability  $\geq 1 - 2\delta$ :

$$\|\hat{X} - A\|_F \leq 20\sqrt{\frac{r\mu}{m_2}} \log\left(\frac{d+n}{\delta}\right) (\|A\|_F + \|R_\Omega\|_F) + \sqrt{8r}\|R\|_2 \quad (2.13)$$

where the number of samples is  $n(m_1 + m_2)$  and  $\Omega$  is the set of all entries observed over the course of the algorithm.

To interpret this bound, let  $\|A\|_F = 1$ , and let  $R$  be a random matrix whose entries are independently drawn from a Gaussian distribution with variance  $\sigma^2/(dn)$ . Note that this normalization for the variance is appropriate in the high-dimensional setting where  $n, d \rightarrow \infty$ , since we keep the signal-to-noise ratio  $\|A\|_F^2/\|R\|_F^2 = 1/\sigma^2$  constant. The last term can be ignored, since by the standard bound on the spectral norm of a Gaussian matrix,  $\|R\|_2 = O(\sigma\sqrt{\frac{1}{d}\log((n+d)/\delta)})$  which will be lower order [1]. We can also bound  $\|R_\Omega\|_F \leq O(\sigma\sqrt{\frac{m_1+m_2}{d}\log((n+d)/\delta)})$  using a Gaussian tail bound. With  $m_1 \leq m_2$  we arrive at:

$$\|\hat{X} - A\|_F \leq c_\star \left( \sqrt{\frac{r\mu}{m_2}} + \sigma\sqrt{\frac{r\mu}{d}} \right) \log^2 \left( \frac{d+n}{\delta} \right),$$

where  $c_\star$  is some positive constant. In the high dimensional setting, when  $r\mu = \tilde{o}(d)$ , this shows that Algorithm 3 consistently recovers  $A$  as long as  $m_2 = \tilde{\omega}(r\mu)$ . This second condition implies that the total number of samples uses is  $\tilde{\omega}(nr\mu)$ .

### 2.3.1 Comparison with related results

The closest result to Theorem 2.5 is the result of Koltchinskii et al. [119] who consider a soft-thresholding procedure and bound the approximation error in squared-Frobenius norm. They assume that the matrix has bounded entry-wise  $\ell_\infty$  norm and give an entry-wise squared-error guarantee of the form:

$$\|\hat{X} - X\|_F^2 \leq \|X - X_r\|_F^2 + cdn\|X\|_\infty^2 \frac{nr \log(d+n)}{M} \quad (2.14)$$

where  $M$  is the total number of samples and  $c$  is a constant. Their bound is quite similar to ours in the relationship between the number of samples and the target rank  $r$ . However, since  $dn\|X\|_\infty^2 \geq \|X\|_F^2$ , their bound is significantly worse in the event that the energy of the matrix is concentrated on a few columns.

To make this concrete, fix  $\|X\|_F = 1$  and let us compare the matrix where every entry is  $\frac{1}{\sqrt{dn}}$  with the matrix where one column has all entries equal to  $\frac{1}{\sqrt{d}}$ . In the former, the error term in the squared-Frobenius error bound of Koltchinskii et al. is  $nr \log(d+n)/M$  while our bound on Frobenius error is, modulo logarithmic factors, the square root of this quantity. In this example, the two results are essentially equivalent. For the second matrix, their bound deteriorates significantly to  $n^2r \log(d+n)/M$  while our bound remains the same. Thus our algorithm is particularly suited to handle matrices with non-uniform column norms.

Apart from interactive sampling, the difference between our procedure and the algorithm of Koltchinskii et al. [119] is a matter of soft- versus hard-thresholding of the singular values of the zero-filled matrix. In the setting of Proposition 2.7, soft thresholding seems more appropriate, as the choice of regularization parameter allows one to trade off the amount of signal and noise captured in  $\hat{X}$ . While in practice one could replace the hard thresholding step with soft thresholding

in our algorithm, there are some caveats with the theoretical analysis. First, soft-thresholding does not ensure that  $\hat{X}$  will be at most rank  $r$ , so it is not suitable for the matrix approximation problem. Second, the resulting error guarantee depends on the sampling distribution, which cannot be translated to the Frobenius norm unless the distribution is quite uniform [119, 134]. Thus the soft-thresholding procedure does not give a Frobenius-norm error guarantee in the non-uniform setting that we are most interested in.

The majority of other results on low rank matrix completion focus on parameter recovery rather than approximation [41, 117, 134]. It is therefore best to compare with Proposition 2.7, where we show that Algorithm 3 consistently recovers the parameter,  $A$ . These results exhibit similar dependence between the number of samples and the problem parameters  $n, r, \epsilon$  but hold under different notions of uniformity, such as spikiness, boundedness, or incoherence. Our result agrees with these existing results but holds under a much weaker notion of uniformity.

Lastly, we emphasize the effect of interactive sampling in our bound. We do not need *any* uniformity assumption over the columns of the input matrix  $X$ . All existing works on noisy low rank matrix completion or matrix approximation from missing data have some assumption of this form, be it incoherence [41, 117], spikiness [134], or bounded  $\ell_\infty$  norm [119]. The detailed comparison with the result of Koltchinskii et al. gives a precise characterization of this effect and shows that in the absence of such uniformity, our interactive sampling algorithm enjoys a significantly lower sample complexity.

In the event of uniformity, our algorithm performs similarly to existing ones. Specifically, we obtain the same relationship between the number of samples  $M$ , the dimensions  $n, d$  and the target rank  $r$ . If we knew *a priori* that the matrix had uniform column lengths, we could omit the first pass of the algorithm, sample uniformly in the second pass and avoid interactivity.

## 2.4 Proofs

In this section we provide detailed proofs of the results in this section. Some well known large-deviation inequalities, that are used throughout this thesis, are stated in the appendix.

### 2.4.1 Proof of Theorem 2.1 and Corollary 2.2

Before turning to the proofs of Theorem 2.1 and Corollary 2.2, we prove several results involving incoherence and the concentration of orthogonal projections under random subsampling.

#### Intermediary Results for Theorem 2.1 and Corollary 2.2

This first intermediary result shows that the test statistic used in Algorithm 2 concentrates sharply around its mean. Specifically, this theorem analyzes the test based on the projection  $\|x_\Omega -$

$\mathcal{P}_{U_\Omega} x_\Omega \|_2^2$ . The proof of this theorem uses various versions of Bernstein's inequality, and improves on the result of Balzano *et al.* [25]. It is the key ingredient to the analysis of these algorithms.

**Theorem 2.8.** *Let  $U$  be an  $r$ -dimensional subspace of  $\mathbb{R}^d$  and  $y = x + v$  where  $x \in U$  and  $v \in U^\perp$ . Fix  $\delta > 0$  and  $m \geq \max\{\frac{8}{3}r\mu(U) \log(2r/\delta), 4\mu(v) \log(1/\delta)\}$  and let  $\Omega$  be an index set of  $m$  entries sampled uniformly with replacement from  $[d]$ . With probability  $\geq 1 - 4\delta$ :*

$$\frac{m(1 - \alpha) - r\mu(U) \frac{\beta}{1-\gamma}}{d} \|v\|_2^2 \leq \|y_\Omega - \mathcal{P}_{U_\Omega} y_\Omega\|_2^2 \leq (1 + \alpha) \frac{m}{d} \|v\|_2^2 \quad (2.15)$$

where  $\alpha = \sqrt{2 \frac{\mu(v)}{m} \log(1/\delta) + \frac{2\mu(v)}{3m} \log(1/\delta)}$ ,  $\beta = (1 + 2 \log(1/\delta))^2$ , and  $\gamma = \sqrt{\frac{8r\mu(U)}{3m} \log(2r/\delta)}$ .

This result showcases much stronger concentration of measure than the result of Balzano *et al.* [25]. The main difference is in the definitions of  $\alpha$  and  $\beta$ , which in their work have worse dependence on the coherence parameter  $\mu(v)$ . These improvements play out into our stronger sample complexity guarantee for the matrix and tensor completion algorithms.

The proof of this theorem is based on three deviation bounds controlling the effect of subsampling. We state and prove these lemmas before turning to the proof of Theorem 2.8.

**Lemma 2.9.** *With the same notations as in Theorem 2.8, with probability  $\geq 1 - 2\delta$ :*

$$(1 - \alpha) \frac{m}{d} \|v\|_2^2 \leq \|v_\Omega\|_2^2 \leq (1 + \alpha) \frac{m}{d} \|v\|_2^2 \quad (2.16)$$

*Proof.* The proof is an application of Bernstein's inequality (Theorem A.1). Let  $\Omega(i)$  denote the  $i$ th coordinate in the sample and let  $X_i = v_{\Omega(i)}^2 - \frac{1}{d} \|v\|_2^2$  so that  $\sum_{i=1}^m X_i = \|v_\Omega\|_2^2 - \frac{m}{d} \|v\|_2^2$ . The variance and absolute bounds are:

$$\sigma^2 = \sum_{i=1}^m \mathbb{E} X_i^2 \leq \frac{m}{n} \sum_{i=1}^n v_i^4 \leq \frac{m}{n} \|v\|_\infty^2 \|v\|_2^2, \quad R = \max \|X_i\| \leq \|v\|_\infty^2.$$

Bernstein's Inequality then shows that:

$$\mathbb{P} \left( \left| \sum_{i=1}^m X_i \right| \geq t \right) \leq 2 \exp \left( \frac{-t^2}{2 \|v\|_\infty^2 \left( \frac{m}{d} \|v\|_2^2 + \frac{1}{3} t \right)} \right).$$

Setting  $t = \alpha \frac{m}{d} \|v\|_2^2$  and using the definition  $\mu(v) = d \|v\|_\infty^2 / \|v\|_2^2$  this bound becomes:

$$\mathbb{P} \left( \left| \sum_{i=1}^m X_i \right| \geq \alpha \frac{m}{d} \|v\|_2^2 \right) \leq 2 \exp \left( \frac{-\alpha^2}{2\mu(v)(1 + \alpha/3)} \right)$$

And plugging in the definition of  $\alpha$  ensures that the probability is upper bounded by  $2\delta$ .



**Lemma 2.10.** *With the same notation as Theorem 2.8 and provided that  $m \geq 4\mu(v) \log(1/\delta)$ , with probability at least  $1 - \delta$ :*

$$\|U_{\Omega}^T v_{\Omega}\|_2^2 \leq \beta \frac{m}{d} \frac{r\mu(U)}{d} \|v\|_2^2 \quad (2.17)$$

*Proof.* The proof is an application of the vector version of Bernstein's inequality (Proposition A.2). Let  $u_i \in \mathbb{R}^r$  denote the  $i$ th row of an orthonormal basis for  $U$  and set  $X_i = u_{\Omega(i)} v_{\Omega(i)}$ . Since  $v \in U^{\perp}$ , the  $X_i$ s are centered so we are left to compute the variance:


$$\sum_{i=1}^m \mathbb{E} \|X_i\|^2 = \frac{m}{d} \sum_{j=1}^d \|u_j v_j\|^2 \leq \frac{m}{d} \frac{r\mu(U)}{d} \|v\|_2^2 = V$$

Applying Proposition A.2 and re-arranging, we have that with probability at least  $1 - \delta$ :

$$\|U_{\Omega}^T v_{\Omega}\|_2 \leq \sqrt{V} + \sqrt{4V \log(1/\delta)} = \sqrt{\frac{m}{d} \frac{r\mu}{d}} \|v\|_2 \left(1 + 2\sqrt{\log(1/\delta)}\right)$$

As long as:

$$t = \sqrt{4V \log(1/\delta)} \leq V (\max_i \|X_i\|)^{-1}$$

Since  $\max_i \|X_i\| \leq \|v\|_{\infty} \sqrt{r\mu/d}$  and using the incoherence assumption on  $v$  this condition translates to  $m \geq 4\mu(v) \log(1/\delta)$ . Squaring the above inequality proves the lemma. 

**Lemma 2.11** ([25]). *Let  $\delta > 0$  and  $m \geq \frac{8}{3} r\mu(U) \log(2r/\delta)$ . Then*

$$\|(U_{\Omega}^T U_{\Omega})^{-1}\|_2 \leq \frac{d}{(1 - \gamma)m} \quad (2.18)$$

*with probability at least  $1 - \delta$  provided that  $\gamma < 1$ . In particular  $U_{\Omega}^T U_{\Omega}$  is invertible.*

*Proof of Theorem 2.8.* We begin with the decomposition:


$$\|y_{\Omega} - \mathcal{P}_{U_{\Omega}} y_{\Omega}\|_2^2 = \|v_{\Omega}\|_2^2 - v_{\Omega}^T U_{\Omega} (U_{\Omega}^T U_{\Omega})^{-1} U_{\Omega}^T v_{\Omega}. \quad (2.19)$$

Next, let  $W_{\Omega}^T W_{\Omega} = (U_{\Omega}^T U_{\Omega})^{-1}$ , which is valid provided that  $U_{\Omega}^T U_{\Omega}$  is invertible (which we will subsequently ensure). We have:

$$v_{\Omega}^T U_{\Omega} (U_{\Omega}^T U_{\Omega})^{-1} U_{\Omega}^T v_{\Omega} = \|W_{\Omega} U_{\Omega}^T v_{\Omega}\|_2^2 \leq \|W_{\Omega}\|_2^2 \|U_{\Omega}^T v_{\Omega}\|_2^2 = \|(U_{\Omega}^T U_{\Omega})^{-1}\| \| \|U_{\Omega}^T v_{\Omega}\|_2^2,$$

which means that:

$$\|v_{\Omega}\|_2^2 - \|(U_{\Omega}^T U_{\Omega})^{-1}\| \| \|U_{\Omega}^T v_{\Omega}\|_2^2 \leq \|y_{\Omega} - \mathcal{P}_{U_{\Omega}} y_{\Omega}\|_2^2 \leq \|v_{\Omega}\|_2^2. \quad (2.20)$$

The theorem now follows immediately from Lemmas 2.9, 2.10, and 2.11, which control the quantities in the above inequalities. 

Another significant component of the proof involves controlling the incoherence of various subspaces that appear throughout the execution of the algorithm. The following lemmas control precisely these quantities.

**Lemma 2.12.** *Let  $U_1 \subset \mathbb{R}^{n_1}, U_2 \subset \mathbb{R}^{n_2}, \dots, U_T \subset \mathbb{R}^{n_T}$  be subspaces of dimension at most  $d$ , let  $W_1 \subset U_1$  have dimension  $d'$ . Define  $\mathbb{S} = \text{span}(\{\otimes_{i=1}^T u_i^{(t)}\}_{i=1}^d)$ . Then:*

$$(a) \quad \mu(W_1) \leq \frac{\dim(U_1)}{d'} \mu(U_1).$$

$$(b) \quad \mu(\mathbb{S}) \leq d^{T-1} \prod_{i=1}^T \mu(U_i).$$


*Proof.* For the first property, since  $W_1$  is a subspace of  $U_1$ ,  $\mathcal{P}_{W_1} e_j = \mathcal{P}_{W_1} \mathcal{P}_{U_1} e_j$  so  $\|\mathcal{P}_{W_1} e_j\|_2^2 \leq \|\mathcal{P}_{U_1} e_j\|_2^2$ . The result now follows from the definition of incoherence.

For the second property, we instead compute the incoherence of:

$$\mathbb{S}' = \text{span} \left( \left\{ \otimes_{t=1}^T u^{(t)} \right\}_{u^{(t)} \in U_t \forall t} \right)$$

which clearly contains  $\mathbb{S}$ . Note that if  $\{u_i^{(t)}\}$  is an orthonormal basis for  $U_t$  (for each  $t$ ), then the outer product of all combinations of these vectors is a basis for  $\mathbb{S}'$ . We now compute:

$$\begin{aligned} \mu(\mathbb{S}') &= \frac{\prod_{i=1}^T n_i}{\prod_{t=1}^T \dim(U_t)} \max_{k_1 \in [n_1], \dots, k_T \in [n_T]} \|\mathcal{P}_{\mathbb{S}'}(\otimes_{t=1}^T e_{k_t})\|^2 \\ &= \frac{\prod_{i=1}^T n_i}{\prod_{t=1}^T \dim(U_t)} \max_{k_1, \dots, k_T} \sum_{i_1, \dots, i_T} \langle \otimes_{t=1}^T u_{i_t}^{(t)}, \otimes_{t=1}^T e_{k_t} \rangle^2 \\ &= \frac{\prod_{i=1}^T n_i}{\prod_{t=1}^T \dim(U_t)} \max_{k_1, \dots, k_T} \sum_{i_1, \dots, i_T} \prod_{t=1}^T (u_{i_t}^{(t)T} e_{k_t})^2 \\ &= \frac{\prod_{i=1}^T n_i}{\prod_{t=1}^T \dim(U_t)} \prod_{j=1}^T \max_{k_j} \sum_{i=1}^r (u_i^{(t)T} e_{k_j})^2 \leq \prod_{t=1}^T \mu(U_t) \end{aligned}$$

Now, property (a) establishes that  $\mu(\mathbb{S}) \leq \frac{r^T}{r} \mu(\mathbb{S}')$  which is the desired result. 

Theorem 2.8, Lemma 2.12, and some algebraic manipulations, yields the following corollary, which we use in the analysis of the Algorithm 2:


**Corollary 2.13.** *Suppose that  $\tilde{U}$  is a subspace of  $U$  and  $x_t \in U$  but  $x_t \notin \tilde{U}$ . Observe a set of coordinates  $\Omega \subset [d]$  of  $m$  entries sampled uniformly at random with replacement. If  $m \geq 32r\mu_0 \log^2(2r/\delta)$  then with probability  $\geq 1 - 4\delta$ ,  $\|x_{t\Omega} - \mathcal{P}_{\tilde{U}_\Omega} x_{t\Omega}\|_2 > 0$ . If  $x_t \in \tilde{U}$ , then conditioned on the fact that  $U_\Omega^T U_\Omega$  is invertible,  $\|x_{t\Omega} - \mathcal{P}_{\tilde{U}_\Omega} x_{t\Omega}\|_2 = 0$  with probability 1.*



*Proof.* The second statement follows from the fact that if  $x_t \in \tilde{U}$ , then  $x_{t\Omega} \in \tilde{U}_\Omega$ , so the projection onto the orthogonal complement is identically zero. As for the first statement, we apply Theorem 2.8, noting that the conditions on  $m$  are satisfied.

We now verify that the lower bound is strictly positive. By Lemma 2.12(a), we know that any vector  $v$  in  $U$  has coherence  $\mu(v) \leq r\mu_0$  and similarly any subspace  $\tilde{U} \subset U$  has  $\dim(\tilde{U})\mu(\tilde{U}) \leq r\mu_0$ . Plugging in  $m$  into the definition  $\alpha, \gamma$ , and using the previous facts, we see that  $\alpha < 1/2$  and  $\gamma < 1/3$ . We are left with:

$$\|x_{t\Omega} - \mathcal{P}_{\tilde{U}_\Omega} x_{t\Omega}\|_2^2 \geq \frac{1}{d} \left( \frac{m}{2} - \frac{3r\mu\beta}{2} \right)$$

and the lower bound is strictly positive whenever  $3r\mu\beta \leq m$ . Plugging in the definition of  $\beta$ , we see that this relation is also satisfied, concluding the proof. 

### Proof of Corollary 2.2

Corollary 2.2 is considerably simpler to prove than Theorem 2.1, so we prove the former in its entirety before proceeding to the latter. First notice that our estimates  $\tilde{U}$  for the column space is always a subspace of the true column space, since we only ever add in fully observed vectors that live in the column space. Also notice that we only resample the set  $\Omega$  at most  $r + 1$  times, since the matrix is exactly rank  $r$ , and we only resample when we find a linearly independent column. Thus with probability  $1 - (r + 1)\delta$ , by application of Lemma 2.11 from the appendix, all of the matrices  $\tilde{U}_\Omega^T \tilde{U}_\Omega$  are invertible.

When processing the  $t$ th column, one of two things can happen. Either  $x_t$  lives in our current estimate for the column space, in which case we know from the above corollary that with probability 1,  $\|x_{t\Omega} - \mathcal{P}_{\tilde{U}_\Omega} x_{t\Omega}\|_2^2 = 0$ . This holds since we have already accounted for the probability that  $U_\Omega^T U_\Omega$  is not-invertible. When this happens we do not obtain additional samples and just need to ensure that we reconstruct  $x_t$ , which we will see below. If  $x_t$  does not live in  $U$ , then with probability  $\geq 1 - 4\delta$  the estimated projection is strictly positive (by Corollary 2.13), in which case we fully observe the new direction  $x_t$  and augment our subspace estimate. In fact, this failure probability includes the event that  $U_\Omega^T U_\Omega$  is not invertible.

Since  $X$  has rank at most  $r$ , this latter case can happen no more than  $r$  times, and via a union bound, the failure probability is  $\leq 4r\delta + \delta$ . Here, the last factor of  $\delta$  ensures that the last subsampled projection operator is well behaved. In other words, with probability  $\geq 1 - 4r\delta - \delta$ , our estimate  $U$  at the end of the algorithm is exactly the column space of  $X$ .

The vectors that were not fully observed are recovered exactly as long as  $(U_\Omega^T U_\Omega)^{-1}$  is invertible. This follows from the fact that, if  $x_t \in U$ , we can write  $x_t = U\alpha_t$  and we have:

$$\hat{x}_t = U(U_\Omega^T U_\Omega)^{-1} U_\Omega^T U_\Omega \alpha_t = U\alpha_t = x_t$$

We already accounted for the probability that these matrices are invertible. We showed above that the total failure probability is at most  $5r\delta$  when  $m \geq 32r\mu_0 \log^2(2r/\delta)$ , so by setting  $m \geq 32r\mu_0 \log^2(10r^2/\delta)$ , the total failure probability is at most  $\delta$ .

For the running time, per column, the dominating computational costs involve the projection  $\mathcal{P}_{\tilde{v}_\Omega}$  and the reconstruction procedure. The projection involves several matrix multiplications and the inversion of a  $r \times r$  matrix, which need not be recomputed on every iteration. Ignoring the matrix inversion, this procedure takes at most  $O(mr)$  time per column, since the vector and the projector are subsampled to  $m$ -dimensions, for a total running time of  $O(nmr)$ . At most  $r$  times, we must recompute  $(U_\Omega^T U_\Omega)^{-1}$ , which takes  $O(r^2 m)$ , contributing a factor of  $O(r^3 m)$  to the total running time. Finally, we run the Gram-Schmidt process once over the course of the algorithm, which

takes  $O(dr^2)$  time.



## Proof of Theorem 2.1

We now generalize the above proof to the tensor completion case and prove Theorem 2.1. We first focus on the recovery of the tensor in total, expressing this in terms of failure probabilities in the recursion. Then we inductively bound the failure probability of the entire algorithm. Finally, we compute the total number of observations. For now, define  $\tau_T$  to be the failure probability of recovering a  $T$ -order tensor.

By Lemma 2.12, the subspace spanned by the mode- $T$  tensors has incoherence at most  $r^{T-2}\mu_0^{T-1}$  and rank at most  $r$  and each slice has incoherence at most  $r^{T-1}\mu_0^{T-1}$ . The subspace spanned by the mode- $T$  sub-tensors is based on the outer product of the subspaces  $\{A^{(i)}\}_{i=1}^{T-1}$  so it is based on the outer product of  $T-1$  subspaces, all with coherence bounded by  $\mu_0$  and dimension at most  $r$ . This means that the subspace spanned by the mode- $T$  sub-tensors has incoherence  $r^{T-2}\mu_0^{T-1}$  and each slice is a 1-dimensional subspace of this  $r$ -dimensional subspace, so it has incoherence that is a factor of  $r$  larger.

By the same argument as Corollary 2.13, we see that with  $m_{T-1} \geq 32r^{T-1}\mu_0^{T-1} \log^2(2r/\delta_{T-1})$  the projection test succeeds in identifying informative sub-tensors (those not in our current basis) with probability  $\geq 1 - 4\delta_{T-1}$ . With a union bound over these  $r$  sub-tensors, the failure probability becomes  $\tau_T \leq 4r\delta_{T-1} + \delta_{T-1}$ , not counting the probability that we fail in recovering these sub-tensors, which is  $r\tau_{T-1}$ .

For each order  $T-1$  tensor that we have to recover, the subspace of interest has incoherence at most  $r^{T-3}\mu_0^{T-2}$  and with probability  $\geq 1 - 4r\delta_{T-2}$  we correctly identify each informative sub-tensor as long as  $m_{T-2} \geq 32r^{T-2}\mu_0^{T-2} \log^2(2r/\delta_{T-2})$ . Again the failure probability is at most  $\tau_{T-1} \leq 4r\delta_{T-2} + \delta_{T-2} + r\tau_{T-2}$ .

To compute the total failure probability we proceed inductively.  $\tau_1 = 0$  since we completely observe any one-mode tensor (vector). The recurrence relation is:

$$\tau_t = 4r\delta_{t-1} + \delta_{t-1} + r\tau_{t-1}. \quad (2.21)$$

Which in words means that we complete  $r$  subtensors of order  $T - 1$ ,  $r^2$  tensors of order  $T - 2$  and so on, observing  $r^{T-1}$  order 1 tensors (or vectors) in full. The total failure probability is therefore bounded by:

$$\tau_T = \sum_{t=1}^{T-1} 5r^{T-t}\delta_t. \quad (2.22)$$

The requirement on  $m_t$  is:

$$m_t \geq 32r^t\mu_0^t \log^2(2r^t/\delta_t).$$

To achieve risk at most  $\delta$ , one can set  $m_t \geq 32Tr^t\mu_0^t \log^2(10rT/\delta)$ , which concludes the proof of the statistical guarantee for Algorithm 2.

We also compute the sample complexity inductively. Let  $\eta_T$  denote the number of samples needed to complete an order  $T$  tensor. Then  $\eta_1 = n_1$  and:

$$\eta_t = n_tm_{t-1} + r\eta_{t-1}$$

So that  $\eta_T$  is upper bounded as:

$$\eta_T \leq \sum_{t=1}^T n_tm_{t-1}r^{T-t} \leq 32T \left( \sum_{t=1}^T n_t \right) r^{T-1}\mu_0^{T-1} \log^2(10rT/\delta)$$

when we set  $m_t$  as above.

The running time is computed in a similar way to the matrix case. To complete an order  $T$  tensor, we must complete  $r$  order  $T - 1$  tensors, and additionally process each subtensor. As in the matrix case, processing all of the  $n_T$  subtensors requires  $m_{T-1}r$  time per column to do all vector and matrix multiplications,  $O(r^3m_{T-1})$  time to do the matrix inversions, and  $O(r^2 \prod_{t=1}^{T-1} n_t)$  to perform Gram-Schmidt. If the running time to complete a order  $t$  tensors is denote  $\kappa_t$ , then the running time is inductively defined as:

$$\kappa_t = r\kappa_{t-1} + O \left( n_tm_{t-1}r + r^3m_{t-1} + r^2 \prod_{i=1}^{t-1} n_i \right), \quad (2.23)$$

with  $\kappa_1 = n_1$ . Using the fact that  $m_t = \tilde{O}(r^t)$  and that  $r \leq \min_t \{n_t\}$ , the total running time can be bounded by:

$$\tilde{O} \left( \sum_{t=1}^T n_tr^T + Tr^{2+T} + r^2 \prod_{t=1}^{T-1} n_t \right)$$

where we are treating  $\mu_0$  as a constant and ignoring logarithmic factors.



## 2.4.2 Proof of Theorem 2.3

The proof of the necessary condition in Theorem 2.3 is based on a standard reduction-to-testing style argument. For ease of notation, we suppress the dependence on the parameters to  $R^*$ ,  $\mathcal{Q}$ , and  $\mathcal{X}$ . The high-level architecture is to consider a subset  $\mathcal{X}' \subset \mathcal{X}$  of inputs and lower bound the Bayes risk. Specifically, if we fix a prior  $\pi$  supported on  $\mathcal{X}'$ ,

$$\begin{aligned} R^* &= \inf_{f \in \mathcal{F}} \inf_{q \in \mathcal{Q}} \max_{X \in \mathcal{X}} \mathbb{P}_{\Omega \sim q} [f(\Omega, X_\Omega) \neq X] \\ &\geq \inf_{f \in \mathcal{F}} \inf_{q \in \mathcal{Q}} \mathbb{E}_{\Omega \sim q, X \sim \pi} [\mathbb{P}_f [f(\Omega, X_\Omega) \neq X]] \\ &\geq \inf_{f \in \mathcal{F}} \min_{\Omega: |\Omega|=M} \mathbb{E}_{X \sim \pi} [\mathbb{P}_f [f(\Omega, X_\Omega) \neq X]] \end{aligned}$$

The first step is a standard one in information theoretic lower bounds and follows from the fact that the maximum dominates any expectation over the same set. The second step is referred to as Yao's Minimax Principle in the analysis of randomized algorithms, which says that one need only consider deterministic algorithms if the input is randomized. It is easily verified by the fact that in the second line, the inner expression is linear in  $q$ , so it is minimized on the boundary of the simplex, which is a deterministic choice of  $\Omega$ . We use  $\mathbb{P}_f$  to emphasize that  $f$  can be randomized, although it will suffice to consider deterministic  $f$ .

Let  $\pi$  be the uniform distribution over  $\mathcal{X}' \subset \mathcal{X}$ . The minimax risk is lower bounded by:

$$R^* \geq 1 - \max_{\Omega} \mathbb{E}_{X \sim \pi} |\{X' \in \mathcal{X}' | X'_\Omega = X_\Omega\}|^{-1}$$

since if there is more than one matrix in  $\mathcal{X}'$  that agrees with  $X$  on  $\Omega$ , the best any estimator could do is guess. Notice that since  $X$  is drawn uniformly, this is equivalent to considering an  $f$  that deterministically picks one matrix  $X' \in \mathcal{X}'$  that agrees with the observations.

To upper bound the second term, define  $\mathcal{U}_\Omega = \{X \in \mathcal{X}' : |\{X' \in \mathcal{X}' | X'_\Omega = X_\Omega\}| = 1\}$  which is the set of matrices that are uniquely identified by the entries  $\Omega$ . Also set  $\mathcal{N}_\Omega = \mathcal{X}' \setminus \mathcal{U}_\Omega$ , which is the set of matrices that are not uniquely identified by  $\Omega$ . We may write:

$$\max_{\Omega} \mathbb{E}_{X \sim \pi} |\{X' \in \mathcal{X}' | X'_\Omega = X_\Omega\}|^{-1} \leq \max_{\Omega} \frac{1}{2} + \frac{|\mathcal{U}_\Omega|}{2|\mathcal{X}'|}$$

Since if  $X \in \mathcal{N}_\Omega$ , there are at least two matrices that agree on those observations, so the best estimator is correct with probability no more than  $1/2$ .

We now turn to constructing a set  $\mathcal{X}'$ . Set  $l = \frac{d}{r\mu_0}$ . The left singular vectors  $u_1, \dots, u_{r-1}$  will be constant on  $\{1, \dots, l\}, \{l+1, \dots, 2l\}$  etc. while the first  $r-1$  right singular vectors  $v_1, \dots, v_{r-1}$  will be the first  $r-1$  standard basis elements. We are left with:

$$d - (r-1)l = d - \frac{r-1}{r} \frac{d}{\mu_0} \triangleq dc_1,$$

coordinates where we will attempt to hide the last left singular vector. Here we defined  $c_1 = 1 - \frac{r-1}{r\mu_0}$ , which is not a constant, but will ease the presentation. For  $u_r$ , we pick  $l$  coordinates

out of the  $dc_1$  remaining, pick a sign for each and let  $u_r$  have constant magnitude on those coordinates. There are  $2^l \binom{dc_1}{l}$  possible choices for this vector. The last right singular vector is one of the  $n - r$  remaining standard basis vectors. Notice that our choice of  $l$  ensures that every matrix in this family meets the column space incoherence condition.

To upper bound  $|\mathcal{U}_\Omega|$  notice that since  $u_r$  can have both positive and negative signs, a matrix is uniquely identified only if all of the entries corresponding to the last singular vector are observed. Thus observations in the  $t$ th column only help to identify matrices whose last rank was hidden in that column. If we use  $m_t$  observations on the  $t$ th column, we uniquely identify  $2^l \binom{m_t}{l}$  matrices, where  $\binom{m_t}{l} = 0$  if  $m_t < l$ . In total we have:

$$|\mathcal{X}'| = (n - r)2^l \binom{dc_1}{l} \quad \text{and} \quad |\mathcal{U}_\Omega| = 2^l \sum_{i=r}^n \binom{m_i}{l}$$

We are free to choose  $m_i$  to maximize  $|\mathcal{U}_\Omega|$  subject to the constraints  $m_i \leq dc_1$  and  $\sum_i m_i \leq M$ , the total sensing budget. Optimizing over  $m_i$  is a convex maximization problem with linear constraints, and consequently the solution is on the boundary. By symmetry, this means that that best sampling pattern is to observe columns in their entirety and devote the remaining observations to one more column. With  $M$  observations, we can observe  $\frac{M}{c_1 n}$  columns fully, leading to the bounds:

$$|\mathcal{U}_\Omega| \leq 2^l \lceil \frac{M}{c_1 n} \rceil \binom{nc_1}{l}, \quad \text{and} \quad \frac{|\mathcal{U}_\Omega|}{|\mathcal{X}'|} \leq \lceil \frac{M}{c_1 n} \rceil \frac{1}{n_2 - r},$$

which, after plugging in for  $c_1$ , leads to the lower bound on the risk.



### 2.4.3 Proof of Theorem 2.4

We start by giving a proof in the matrix case, which is a minor correction of the proof by Candes and Tao [43]. Then we turn to the tensor case, where only small adjustments are needed to establish the result. We work in the Bernoulli model, noting that Candes' and Tao's arguments demonstrate how to adapt these results to the uniform-at-random sampling model.

#### Matrix Case

In the matrix case, suppose that  $l_1 = \frac{n_1}{r}$  and  $l_2 = \frac{n_2}{\mu_0 r}$  are both integers. Define the following blocks  $R_1, \dots, R_r \subset [n_1]$  and  $C_1, \dots, C_r \subset [n_2]$  as:

$$\begin{aligned} R_i &= \{l_1(i-1) + 1, l_1(i-1) + 2, \dots, l_1 i\} \\ C_i &= \{l_2(i-1) + 1, l_2(i-1) + 2, \dots, l_2 i\} \end{aligned}$$

Now consider the  $n_1 \times n_2$  family of matrices defined by:

$$\mathcal{M} = \left\{ \sum_{k=1}^r u_k v_k^T \mid u_k = [1, \sqrt{\mu_0}]^n \circ \mathbf{1}_{R_k}, v_k = \mathbf{1}_{C_k} \right\}. \quad (2.24)$$

The  $\circ$  operator is the Hadamard operator, which performs entry-wise multiplication.  $\mathcal{M}$  is a family of block-diagonal matrices where the blocks have size  $l_1 \times l_2$ . Each block has constant rows, but each row may take any value in  $[1, \sqrt{\mu_0}]$ . For any  $M \in \mathcal{M}$ , the incoherence of the column space can be computed as:

$$\mu(U) = \frac{n_1}{r} \max_{j \in [n_1]} \|\mathcal{P}_U e_j\|_2^2 = \frac{n_1}{r} \max_{k \in [r]} \max_{j \in [n_1]} \frac{(u_k^T e_j)^2}{(u_k^T u_k)^2} \leq \frac{n_1}{r} \max_{k \in [r]} \frac{\mu_0}{(n_1/r)} = \mu_0$$

A similar calculation reveals that the row space is also incoherent with parameter  $\mu_0$ .

Unique identification of  $M$  is not possible unless we observe at least one entry from each row of each diagonal block. If we did not, then we could vary that corresponding coordinate in the appropriate  $u_k$  and find infinitely many matrices  $M' \in \mathcal{M}$  that agree with our observations, have rank and incoherence at most  $r$  and  $\mu_0$  respectively. Thus, the probability of successful recovery is no larger than the probability of observing one entry of each row of each diagonal block.

The probability that any row of any block is unsampled is  $\pi_1 = (1-p)^{l_2}$  and the probability that all rows are sampled is  $(1-\pi_1)^{n_1}$ . This must upper bound the success probability  $1-\delta$ . Thus:

$$-n_1 \pi_1 \geq n_1 \log(1-\pi_1) \geq \log(1-\delta) \geq -2\delta$$

or  $\pi_1 \leq 2\delta/n_1$  as long as  $\delta < 1/2$ . Substituting  $\pi_1 = (1-p)^{l_2}$  we obtain:

$$\log(1-p) \leq \frac{1}{l_2} \log\left(\frac{2\delta}{n_1}\right) = \frac{\mu_0 r}{n_2} \log\left(\frac{2\delta}{n_1}\right)$$

as a necessary condition for unique identification of  $M$ .

Exponentiating both sides, writing  $p = \frac{m}{n_1 n_2}$  and the fact that  $1 - e^{-x} > x - x^2/2$  gives us:

$$m \geq n_1 \mu_0 r \log\left(\frac{n_1}{2\delta}\right) (1 - \epsilon/2)$$

when  $\mu_0 r / n_2 \log(\frac{n_1}{2\delta}) \leq \epsilon < 1$ .

## Tensor Case

Fix  $T$ , the order of the tensor and suppose that  $l_1 = \frac{n_1}{r}$  is an integer. Moreover, suppose that  $l_t = \frac{n_t}{\mu_0^r}$  is an integer for  $1 < t \leq T$ .

Define a set of blocks, one for each mode and the family

$$B_i^{(t)} = \{l_t(i-1) + 1, l_t(i-1) + 2, \dots, l_t i\} \quad \forall i \in [r], t \in [T]$$

$$\mathcal{M} = \left\{ \sum_{i=1}^r \otimes_{t=1}^T a_i^{(t)} \left| \begin{array}{l} a_i^{(1)} = [1, \sqrt{\mu_0}]^n \circ \mathbf{1}_{B_i^{(1)}} \\ a_i^{(t)} = \mathbf{1}_{B_i^{(t)}}, 1 < t \leq T \end{array} \right. \right\}$$

This is a family of block-diagonal tensors and just as before, straightforward calculations reveal that each subspace is incoherent with parameter  $\mu_0$ . Again, unique identification is not possible unless we observe at least one entry from each row of each diagonal block. The difference is that in the tensor case, there are  $\prod_{i \neq 1} l_i$  entries per row of each diagonal block so the probability that any single row is unsampled is  $\pi_1 = (1-p)^{\prod_{i \neq 1} l_i}$ . Again there are  $n_1$  rows and any algorithm that succeeds with probability  $1-\delta$  must satisfy:

$$-n_1 \pi_1 \geq n_1 \log(1 - \pi_1) \geq \log(1 - \delta) \geq -2\delta$$

Which implies  $\pi_1 \leq 2\delta/n_1$  (assuming  $\delta < 1/2$ ). Substituting in the definition of  $\pi_1$  we have:

$$\log(1-p) \leq \frac{1}{\prod_{i \neq 1} l_i} \log\left(\frac{2\delta}{n_1}\right) = \frac{\mu_0^{T-1} r^{T-1}}{\prod_{i \neq 1} n_i} \log\left(\frac{2\delta}{n_1}\right)$$

The same approximations as before yield the bound (as long as  $\frac{\mu_0^{T-1} r^{T-1}}{\prod_{i \neq 1} n_i} \log\left(\frac{n_1}{2\delta}\right) \leq \epsilon < 1$ )

$$m \geq n_1 \mu_0^{T-1} r^{T-1} \log\left(\frac{n_1}{2\delta}\right) (1 - \epsilon/2).$$



## 2.4.4 Proof of Theorem 2.5 and related propositions

To prove the main approximation theorem (Theorem 2.5), we must analyze the three phases of the algorithm. The analysis of the first phase is fairly straightforward; we show that under the incoherence assumption, one can compute a reliable estimate of each column norm from a very small number of measurements per column. For the second phase, we show that by sampling according to the re-weighted distribution using the column-norm estimates, the matrix  $\tilde{X}$  is close to  $X$  in spectral norm. We then translate this spectral norm guarantee into a approximation guarantee for  $\hat{X} = \tilde{X}_r$ .

Let us start with this translation. We use a lemma of [1].

**Lemma 2.14** ([1]). *Let  $A$  and  $N$  be any matrices and write  $\hat{A} = A + N$ . Then:*

$$\|A - \hat{A}_k\|_2 \leq \|A - A_k\|_2 + 2\|N_k\|_2$$

$$\|A - \hat{A}_k\|_F \leq \|A - A_k\|_F + \|N_k\|_F + 2\sqrt{\|N_k\|_F \|A_k\|_F}$$

The lemma states that if  $\hat{A} - A$  is small, then the top  $k$  ranks of  $\hat{A}$  is nearly as good an approximation to  $A$  as is the top  $k$  ranks of  $A$  itself. Notice that all of the error terms only depend on rank- $k$  matrices. We will use this lemma with  $\tilde{X}$  and  $X$  and of course with the target rank as  $r$ . We will soon show that  $\|X - \tilde{X}\|_2 \leq \epsilon \|X\|_F$ , which implies:

$$\begin{aligned} \|X - \hat{X}\|_F &\leq \|X - X_r\| + \|(X - \tilde{X})_r\|_F + 2\sqrt{\|(X - \tilde{X})_r\|_F \|X_r\|_F} \\ &\leq \|X - X_r\| + \sqrt{r} \|X - \tilde{X}\|_2 + 2\sqrt{\sqrt{r} \|X - \tilde{X}\|_2 \|X\|_F} \\ &\leq \|X - X_r\| + \|X\|_F (\sqrt{r}\epsilon + 2r^{1/4}\epsilon^{1/2}) \end{aligned} \quad (2.25)$$

So if we can obtain a bound on  $\|X - \tilde{X}\|_2$  of that form, we will have proved the theorem.

As for Propositions 2.6 and 2.7, the translation uses the first inequality of Achlioptas and McSherry [1]. If  $X$  is rank  $r$ , the matrix  $\hat{X} - X$  has rank at most  $2r$ , which means that:

$$\|\hat{X} - X\|_F \leq \sqrt{2r} \|\hat{X} - X\|_2 \leq 2\sqrt{2r} \|\tilde{X} - X\|_2 \leq 2\sqrt{2r}\epsilon \|X\|_F$$

For the second proposition, we first bound  $\|\hat{X} - M\|_2$  and then use the same argument.

$$\begin{aligned} \|\hat{X} - M\|_2 &\leq \|\hat{X} - X\|_2 + \|R\|_2 \leq \|X - X_r\|_2 + 2\epsilon \|X\|_F + \|R\|_2 \\ &\leq 2\|R\|_2 + 2\epsilon(\|M\|_F + \|R_{\Omega^C}\|_F). \end{aligned}$$

To arrive at the second line, we use the fact that  $X_r$  is the best rank  $r$  approximation to  $X$ , so  $\|X - X_r\|_2 \leq \|X - M\|_2 = \|R\|_2$ . We also use the triangle inequality on the term  $\|X\|_F$ , but use the fact that since the algorithm never looked at  $X$  on  $\Omega^C$  it is fair to set  $R_{\Omega^C} = 0$ .

Let us now turn to the first phase. In our analysis of the Algorithm 1, we proved that the norm of an incoherent vector can be approximated by subsampling. Specifically, Lemma 2.9 shows that with high probability, the estimates  $\hat{c}_t$  once appropriately rescaled are trapped between  $(1 - \alpha)c_t$  and  $(1 + \alpha)c_t$  where  $\alpha = \sqrt{2\mu/m_1 \log(n/\delta)} + \frac{2\mu}{3m_1} \log(n/\delta)$ . The same is of course true for  $\hat{f}$ . Setting  $m_1 \geq 32\mu \log(n/\delta)$  we find that  $\alpha \leq 1/2$ , meaning that by using in total  $32n\mu \log(n/\delta)$  samples in the first phase, we approximate the target sampling distribution to within a multiplicative factor of  $1/2$  with probability  $\geq 1 - \delta$ .

For the second pass, we show that  $\tilde{X}$  is close to  $X$  in spectral norm. We use the following lemma:

**Lemma 2.15.** *Provided that  $(1 - \alpha)c_t \leq \frac{d}{m_1}\hat{c}_t \leq (1 + \alpha)c_t$  and  $(1 - \alpha)f \leq \frac{d}{m_1}\hat{f} \leq (1 + \alpha)f$ , with probability  $\geq 1 - \delta$ :*

$$\|\tilde{X} - X\|_2 \leq \|X\|_F \sqrt{\frac{1 + \alpha}{1 - \alpha}} \left( \sqrt{\frac{4}{m_2} \max\left(\frac{d}{n}, \mu\right) \log\left(\frac{d + n}{\delta}\right)} + \frac{4}{3} \sqrt{\frac{d\mu}{m_2 n}} \log\left(\frac{d + n}{\delta}\right) \right)$$

*Proof.* Under the uniform at random sampling model, we will apply the non-commutative Bernstein inequality (Proposition A.4) to bound  $\|\tilde{X} - X\|_2$ . Recall that for each column  $x_t$ , we observe a set of  $m_{2,t} = m_2 n \frac{\hat{c}_t}{f}$  observations and form the zero-filled vector  $y_t$  defined by:

$$y_t = \frac{d}{m_{2,t}} \sum_{s=1}^{m_{2,t}} x_t(i_s) e_{i_s}$$



where  $\{i_s\}_{s=1}^{m_{2,t}}$  are the observations. Since the set of observations is sampled with replacement (although duplicates in each half of the sample are thrown out), each entry of  $y_t$  occurs with probability  $d/m_{2,t}$ , so  $y_t$  is an unbiased estimate of  $x_t$ . So we will apply the rectangular Matrix Bernstein inequality to  $y_t e_t^T - x_t e_t^T$ . Moreover:

$$\|y_t e_t^T - x_t e_t^T\| \leq \|y_t\| \|e_t\| + \|x_t\| \leq \left(1 + \sqrt{\frac{d\mu}{m_{2,t}}}\right) \|x_t\| \leq 2\sqrt{\frac{d\mu}{m_{2,t}}} \|x_t\|$$

which follows by the triangle inequality, Cauchy-Schwarz and the chain of inequalities:

$$\|y_t\|_2 \leq \sqrt{m_{2,t}} \|y_t\|_\infty \leq \frac{d}{\sqrt{m_{2,t}}} \|x_t\|_\infty \leq \sqrt{\frac{d\mu}{m_{2,t}}} \|x_t\|_2$$

When we plug in for  $m_{2,t}$  we get:

$$\|y_t e_t^T - x_t e_t^T\| \leq 2\sqrt{\frac{d\mu}{m_{2,t}} \frac{c_t}{\hat{c}_t} \hat{f}} \leq 2\|X\|_F \sqrt{\frac{d\mu}{m_{2,t}} \frac{1+\alpha}{1-\alpha}}$$

where  $\alpha$  is the error bound from the first phase of the algorithm.

As for the variance terms in Proposition A.4, both turn out to be quite small as we will soon see. For the first term:

$$\begin{aligned} \left\| \sum_{t=1}^n \mathbb{E} e_t y_t^T y_t e_t^T - e_t x_t^T x_t e_t^T \right\| &= \left\| \sum_{t=1}^n e_t e_t^T (\mathbb{E} \|y_t\|^2 - \|x_t\|^2) \right\| = \\ &= \left\| \sum_{t=1}^n e_t e_t^T \left(\frac{d}{m_{2,t}} - 1\right) \|x_t\|^2 \right\| \leq 2d \max_{t \in [n]} \frac{\|x_t\|^2}{m_{2,t}} \end{aligned}$$

The first equality is straightforward while the second follows from linearity of expectation and the fact that each coordinate of  $y_t$  is non-zero with probability  $m_{2,t}/d$ . The third line follows from the fact that applying the sum leads to an  $n \times n$  diagonal matrix with  $\frac{d}{m_{2,t}} \|x_t\|^2$  on the diagonal. When we use our definition of  $m_{2,t}$  this becomes:

$$\left\| \sum_{t=1}^n \mathbb{E} e_t y_t^T y_t e_t^T \right\| \leq \frac{2d}{m_{2,t}} \|X\|_F^2 \frac{1+\alpha}{1-\alpha}$$

For the second term, we have:

$$\begin{aligned} \left\| \sum_{t=1}^n \mathbb{E} y_t e_t^T e_t y_t^T - \mathbb{E} x_t e_t^T e_t x_t^T \right\| &= \left\| \sum_{t=1}^n \mathbb{E} y_t y_t^T - x_t x_t^T \right\| = \left\| \sum_{t=1}^n \left(\frac{d}{m_{2,t}} - 1\right) \text{diag}(x_t(1)^2, \dots, x_t(d)^2) \right\| \\ &\leq \max_{i \in [d]} \sum_{t=1}^n \frac{2d}{m_{2,t}} x_t(i)^2 \leq \sum_{i=1}^n \frac{2\mu}{m_{2,t}} \|x_t\|_2^2 \leq \|X\|_F^2 \frac{2\mu}{m_{2,t}} \frac{1+\alpha}{1-\alpha} \end{aligned}$$

Here the first equality is trivial while the second one uses the fact that off diagonals of  $y_t y_t^T$  are unbiased for  $x_t x_t^T$  and hence we are left with a diagonal matrix. To arrive at the second line we note that the spectral norm a diagonal matrix is simply the largest diagonal entry. Then we apply the incoherence assumption and final our sampling distribution.

At this point we may apply the inequality which states that with probability  $\geq 1 - \delta$ :

$$\left\| \sum_{t=1}^n y_t e_t^T - x_t x_t^T \right\| \leq \|X\|_F \sqrt{\frac{1+\alpha}{1-\alpha}} \left( \sqrt{\frac{4}{m_2} \max\left(\frac{d}{n}, \mu\right) \log\left(\frac{d+n}{\delta}\right)} + \frac{4}{3} \sqrt{\frac{d\mu}{m_2 n}} \log\left(\frac{d+n}{\delta}\right) \right)$$



The interactive sampling procedure has a dramatic effect on the bound in Lemma 2.15. If one sampled uniformly across the columns, then both terms grows with the squared norm of the largest column rather than with the average squared norms, which is much weaker when the matrix energy is concentrated on a few columns. This is precisely when the row space is coherent.

To wrap up, recall that  $1 \leq \mu \leq d$  and  $n \geq d$ . Setting  $m_1 \geq 32\mu \log(n/\delta)$  so that  $\alpha \leq 1/2$ , the bound in Lemma 2.15 is dominated by:

$$\|\tilde{X} - X\|_2 \leq \|X\|_F \frac{10}{\sqrt{3}} \sqrt{\frac{\mu}{m_2}} \log\left(\frac{d+n}{\delta}\right).$$



Returning to Equation 2.25 we can now substitute in for  $\epsilon$  and conclude the proof.

## 2.5 Empirical Results

We perform a number of simulations to analyze the empirical performance of both Algorithms 1 and 3. The first set of simulations, in Figures 2.1 and 2.2, examine the behavior of Algorithm 1. We work with square matrices where the column space is spanned by binary vectors, constructed so that the matrix has the appropriate rank and coherence. The row space is spanned by either random gaussian vectors in the case of incoherent row space or a random collection of standard basis elements if we want high coherence.

In the first two figures (2.1(a) and 2.1(b)) we study the algorithms dependence on the matrix dimension. For various matrix sizes, we record the probability of exact recovery as we vary the number of samples allotted to the algorithm. We plot the probability of recovery as a function of the fraction of samples per column, denote by  $p$ , (Figure 2.1(a)) and as a function of the total samples per column  $m$  (Figure 2.1(b)). It is clear from the simulations that  $p$  can decrease with matrix dimension while still ensuring exact recovery. On the other hand, the curves in the second figure line up, demonstrating that the number of samples per column remains fixed for fixed

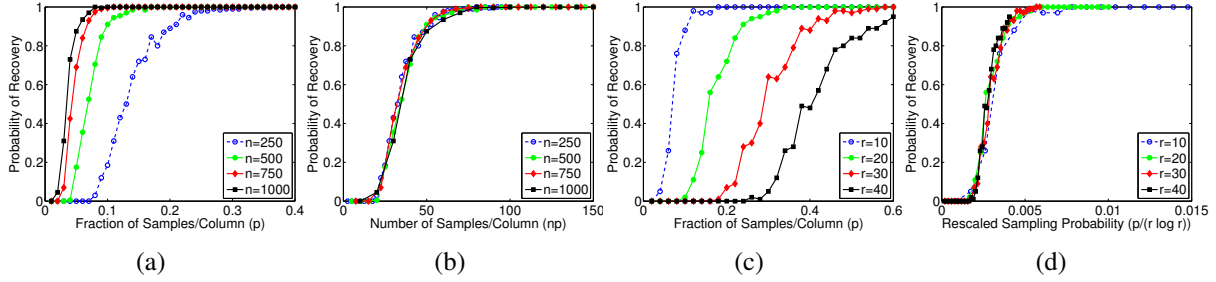


Figure 2.1: (a): Probability of success of Algorithm 1 versus fraction of samples per column ( $p = m/d$ ) with  $r = 10, \mu_0 = 1$ . (b): Data from (a) plotted against samples per column,  $m$ . (c): Probability of success of Algorithm 1 versus fraction of samples per column ( $p = m/d$ ) with  $n = 500, \mu_0 = 1$ . (d): Data from (c) plotted against rescaled sample probability  $p/(r \log r)$ .

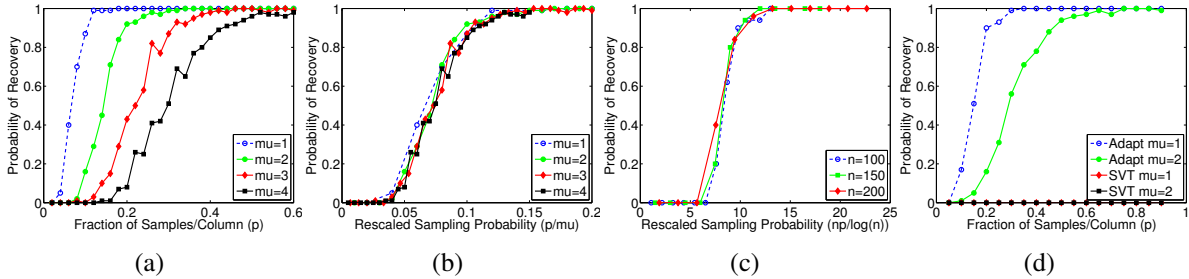


Figure 2.2: (a): Probability of success of Algorithm 1 versus fraction of samples per column ( $p = m/d$ ) with  $n = 500, r = 10$ . (b): Data from (a) plotted against rescaled sampling probability  $p/\mu_0$ . (c): Probability of success of SVT versus rescaled sampling probability  $np/\log(n)$  with  $r = 5, \mu_0 = 1$ . (d): Probability of success of Algorithm 1 and SVT versus sampling probability for matrices with highly coherent row space with  $r = 5, n = 100$ .

probability of recovery. This behavior is predicted by Corollary 2.2, which shows that the total number of samples scales linearly with dimension, so that the number of samples per column remains constant.

In Figures 2.1(c) and 2.1(d) we show the results of a similar simulation, instead varying the matrix rank  $r$ , with dimension fixed at 500. The first figure shows that the fraction of samples per column must increase with rank to ensure successful recovery while second shows that the ratio  $p/(r \log r)$  governs the probability of success. Figures 2.2(a) and 2.2(b) similarly confirm a linear dependence between the incoherence parameter  $\mu_0$  and the sample complexity. Notice that the empirical dependence on rank is actually a better than what is predicted by Corollary 2.2, which suggests that  $r \log^2 r$  is the appropriate scaling. Our theorem does seem to capture the correct dependence on the coherence parameter.

In the last two plots we compare Algorithm 1 against the Singular Value Thresholding algorithm (SVT) of Cai et al. [40]. The SVT algorithm is a non-interactive iterative algorithm for nuclear norm minimization from a set of uniform-at-random observations. In Figure 2.2(c), we show that

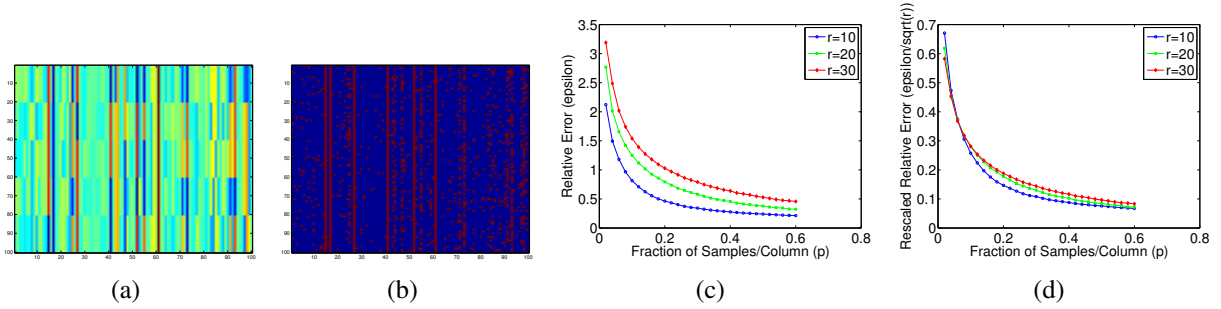


Figure 2.3: (a): An example matrix with highly non-uniform column norms and (b) the sampling pattern of Algorithm 3. (c): Relative error as a function of sampling probability  $p$  for different target rank  $r$  ( $\mu = 1$ ). (d): The same data where the  $y$ -axis is instead  $\epsilon/\sqrt{r}$ .

the success probability is governed by  $np/\log(n)$ , which is predicted by the existing analysis of the nuclear norm minimization program. This dependence is worse than for Algorithm 1, whose success probability is governed by  $np$  as demonstrated in Figure 2.1(b). Finally, in Figure 2.2(d), we record success probability versus sample complexity on matrices with maximally coherent row spaces. The simulation shows that our algorithm can tolerate coherent row spaces while the SVT algorithm cannot.

For Algorithm 3, we display the results of a similar set of simulations in Figures 2.3 and 2.4. Here, we construct low rank matrices whose column spaces are spanned by binary vectors and whose columns are also constant in magnitude on their support. The length of the columns is distributed either log-normally, resulting in non-uniform column lengths, or uniformly between 0.9 and 1.1. We then corrupt this low rank matrix by adding a gaussian matrix whose entries have variance  $\frac{1}{dn}$ . In Figure 2.3(a) we show a matrix constructed via this process and in Figure 2.3(b) we show the set of entries sampled by Algorithm 3 on this input. From the plots, it is clear that the algorithm focuses its measurements on the columns with high energy, while using very few samples to capture the columns with lower energy.

In Figure 2.3(c), we plot the relative error, which is the  $\epsilon$  in Equation 2.3, as a function of the average fraction of samples per column (averaged over columns, as we are using non-uniform sampling) for  $500 \times 500$  matrices of varying rank. In the next plot, Figure 2.3(d), we rescale the relative error by  $\sqrt{r}$ , to capture the dependence on rank predicted by Theorem 2.5.

As we increase the number of observations, the relative error decreases quite rapidly. Moreover, the algorithm needs more observations as the target rank  $r$  increases. Qualitatively both of these effects are predicted by Theorem 2.5. Lastly, the fact that the curves in Figure 2.3(d) nearly line up suggests that the relative error  $\epsilon$  does scale with  $\sqrt{r}$ .

In Figure 2.4(a), we plot the relative error as a function of the average fraction of samples,  $p$ , per column for different matrix sizes. We rescale this data by plotting the  $y$ -axis in terms of  $\sqrt{p}\epsilon$  (Figure 2.4(b)). From the first plot, we see that the error quickly decays, while a smaller fraction of samples are needed for larger problems. In the second plot, we see that rescaling the error by

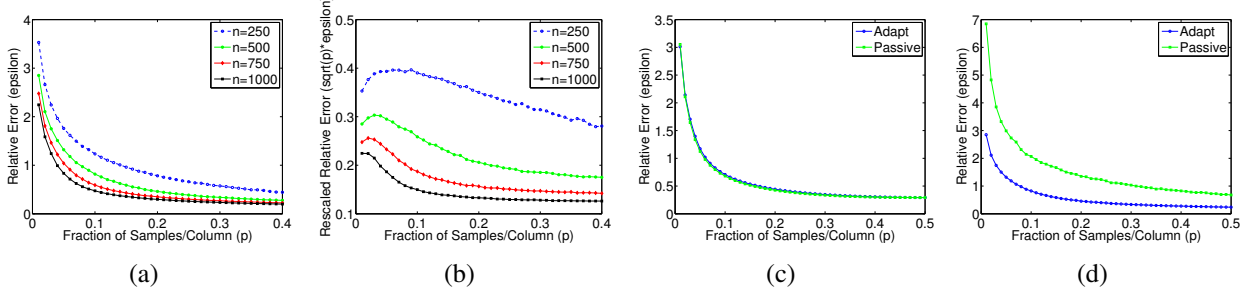


Figure 2.4: (a): Relative error of Algorithm 3 as a function of sampling probability  $p$  for different size matrices with fixed target rank  $r = 10$  and  $\mu = 1$ . (b): The same data where the  $y$ -axis is instead  $\sqrt{p}\epsilon$ . (c): Relative error for interactive and non-interactive sampling on matrices with uniform column lengths (column coherence  $\mu = 1$  and column norms are uniform from  $[0.9, 1.1]$ ). (d): Relative error for interactive and non-interactive sampling on matrices with highly nonuniform column lengths (column coherence  $\mu = 1$  and column norms are from a standard Log-Normal distribution).

$\sqrt{p}$  has the effect of flattening out all of the curves, which suggests that the relationship between  $\epsilon$  and the number of samples is indeed  $\epsilon\sqrt{p} \asymp 1$  or that  $\epsilon \asymp \frac{1}{\sqrt{p}}$ . This phenomenon is predicted by Proposition 2.7.

In the last set of simulations, we compare our algorithm with an algorithm that first performs uniform sampling and then hard thresholds the singular values to build a rank  $r$  approximation. In Figure 2.4(c), we use matrices with uniform column norms, and observe that both algorithms perform comparably. However, in Figure 2.4(d), when the column norms are highly non-uniform, we see that Algorithm 3 dramatically outperforms the passive sampling approach. This confirms our claim that interactive sampling leads to better approximation when the energy of the matrix is not uniformly distributed.

Finally, we compare Algorithm 3 with a non-interactive matrix approximation algorithm on two real datasets. The non-interactive algorithm is the same hard thresholding algorithm used in Figures 2.4(c) and 2.4(d). The first dataset is a 400-node subset of the King internet latency dataset taken from Gummadi et al. [98]. This dataset is much larger but has many missing entries so we used a  $400 \times 400$  submatrix with minimal missing entries for our experiment. The second dataset is a  $1000 \times 10,000$  submatrix of the PubChem molecular similarity dataset [35]. We used target rank 26 for the King dataset and 25 for the PubChem dataset.

In Figure 2.5, we record the log excess risk as a function of the fraction of samples for both interactive and non-interactive matrix approximation algorithms. The interactive algorithm outperforms the non-interactive one on both datasets, but the improvement is more drastic for the King dataset. Moreover, the performance improvements are, in absolute terms, better in the low-sample regime, which is apparent since the separation between the curves stays roughly constant, but this is on a logarithmic scale. This demonstrates that interactive sampling is favorable for these matrix approximation problems, and should be used in applications where it is feasible.

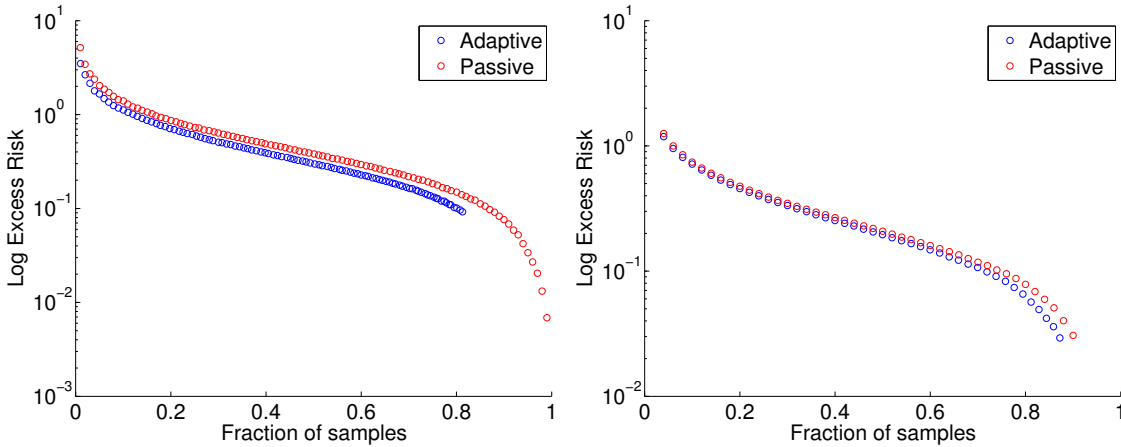


Figure 2.5: Experiments on real datasets. Left: Log excess risk for passive and interactive matrix approximation algorithms on a 400-node subset of the King internet latency dataset with target rank  $r = 26$ . Right: Log excess risk for passive and interactive matrix approximation algorithms on a  $1000 \times 10,000$  submatrix of the PubChem Molecular Similarity dataset with target rank  $r = 25$ . Passive algorithm is based on uniform sampling followed by hard thresholding of the singular values.

## 2.6 Conclusions

This paper considers the two related problems of low rank matrix completion and matrix approximation. In both problems, we show how to use interactive sampling to overcome uniformity assumptions that have pervaded the literature. Our algorithms focus measurements on interesting columns (in the former, the columns that contain new directions and in the latter, the high energy columns) and have performance guarantees that are significantly better than any known passive algorithms in the absence of uniformity. Moreover, they are competitive with state-of-the-art passive algorithms in the presence of uniformity. Our algorithms are conceptually simple, easy to implement, and fairly scalable.

Turning to the themes of this thesis, we showed how interactive sampling enables a relaxation of uniformity requirements for these completion and approximation problems. Specifically, our algorithms do not require incoherence assumptions on the row space to succeed, and we showed that all non-interactive procedures do. Our algorithms are also statistically and computationally efficient, which can be seen both theoretically and empirically in simulations. Thus we believe that these completion problems make a compelling case for interactive learning.

# Chapter 3

## Interactive Hierarchical Clustering

Clustering problems involve assigning objects to one or more groups, so that objects in the same group are very similar while objects in different groups are dissimilar. In hierarchical clusterings, the groups have multiple resolutions, so that a large cluster may be recursively divided into smaller sub-clusters. These types of problems are ubiquitous; they are fundamental tools in exploratory data analysis, data mining, and many scientific domains. There exist many effective algorithms for clustering, but as data sets increase in size, the fact that these algorithms require *every* pairwise similarity between objects poses a serious measurement and/or computational burden and limits the scope for application. It is therefore practically appealing to develop clustering algorithms that are effective on large scale problems but also have low measurement and computational overhead.

To achieve low overhead, we focus on reducing the number of similarity measurements required for clustering. This approach results in an immediate reduction in measurements in applications where similarities are observed directly, but it can also provide dramatic computational gains in applications where similarities between objects are computed via some kernel evaluated on observed object features. The case of internet topology inference is an example of the former, where covariance in the packet delays observed at nodes reflects the similarity between them. Obtaining these similarities requires injecting probe packets into the network and places a significant burden on network infrastructure. Phylogenetic inference and other biological sequence analyses are examples of the latter, where computationally intensive edit distances are often used. Note that both situations result in a low memory footprint as fewer pairwise similarities need to be stored. In both cases our algorithms have dramatically lower overhead than many popular algorithms.

In this chapter, we propose a novel approach to hierarchical clustering through interactivity, an algorithmic paradigm where only a small number of informative similarities are measured. We develop a *meta-algorithm* that iteratively applies a base clustering algorithm to small groups of objects and that can be instantiated with any similarity-based clustering algorithm. This meta-algorithm allows the user to specify a level of interactivity, and we provide theoretical analysis that quantifies the resulting trade-off between measurement overhead and computation time on

one hand, and statistical accuracy on the other.

As an example, we apply our framework to spectral clustering. Spectral clustering is a popular clustering technique that relies on the structure of the eigenvectors of the Laplacian of the similarity matrix. These algorithms have received considerable attention in recent years because of their empirical success, but they suffer from the fact that they require all  $n(n - 1)/2$  similarities and must compute a spectral decomposition of the  $n \times n$  similarity matrix, which on large datasets can be computationally prohibitive, in terms of both running time and space. Our interactive algorithm avoids both of these limitations by subsampling few objects in each round and only computing eigenvectors of very small sub-matrices. By appealing to previous statistical guarantees [16], we can show that this algorithm has desirable theoretical properties, both in terms of statistical and computational performance.

We also establish several necessary conditions in the noisy constant block model, under which we analyze spectral clustering. We give lower bounds on the sample complexity for interactive procedures in the noiseless case and also a lower bound on non-interactive procedures for noisy hierarchical clustering. Comparing this latter lower bound with the analysis for our interactive spectral clustering algorithm concretely demonstrates that interactivity is a powerful learning paradigm for hierarchical clustering from pairwise similarity information.

Our detailed contributions are:

1. We develop a principled method for converting a non-interactive non-hierarchical clustering algorithm into an interactive hierarchical one, and we show how performance guarantees on the subroutine translate into performance guarantees for hierarchical clustering (Theorem 3.1). This technique can be thought of as a simple reduction: we reduce the interactive hierarchical clustering problem to non-interactive flat clustering problem.
2. As an example, we give a detailed statistical analysis of the interactive spectral clustering algorithm derived by our reduction. In a model for similarity based clustering, we show that this interactive spectral algorithm use  $O(n \text{polylog}(n))$  pairwise similarities and runs in  $O(n \text{polylog}(n))$  time, to obtain a hierarchical clustering on  $n$  objects (Theorem 3.2).
3. We prove that any similarity based clustering algorithm must obtain  $\Omega(n \log n / \log \log n)$  similarities, even in the absence of noise (Theorem 3.3). This lower bound certifies the near-optimality of our approach.
4. We also show lower bounds against non-interactive approaches. In the same model used for Theorem 3.2, we show that *any* non-interactive sampling strategy followed by *any* recovery algorithm must use  $\Omega(n^2)$  measurements to achieve the same statistical performance as our non-interactive approach using  $O(n \log^2 n)$  (Theorem 3.5). This certifies the power of interactivity for this problem.
5. We complement these theoretical results with detailed empirical evaluation.

This chapter provides support for our thesis on all fronts. Our interactive clustering algorithm is statistically more powerful than non-interactive approaches in the sense that we can recover cluster structure with far fewer measurements. It also has theoretically and empirically faster



running time as certified by Theorem 3.2 and our experimental evaluation. Lastly, we measure uniformity by the size of the smallest cluster we hope to recover, where smaller clusters make the problem less uniform, and our results are magnified in the presence of non-uniformity.

### 3.1 Related Work

There is a large body of work on hierarchical and partitional clustering algorithms, many coming with various theoretical guarantees, but only few algorithms attempt to minimize the number of pairwise similarities used [24, 87, 154]. Along this line, the work of Eriksson *et al.* [87] and Shamir and Tishby [154] is closest in flavor to ours.

Eriksson *et al.* [87] develop an interactive algorithm for hierarchical clustering and analyze the correctness and measurement complexity under a noise model where a small fraction of the similarities are inconsistent with the hierarchy. This bears resemblance to the persistent noise model that we study in Chapter 4, although the learning task considered there is substantially different. They show that for a constant fraction of inconsistent similarities, their algorithm can recover hierarchical clusters up to size  $\Omega(\log n)$  using  $O(n \log^2 n)$  similarities. Our analysis for ACTIVE SPECTRAL yields similar results in terms of noise tolerance, measurement complexity, and resolution, but in the context of i.i.d. subgaussian noise rather than inconsistencies. Our algorithm is also computationally more efficient.

Another approach to minimizing the number of similarities used is via perturbation theory, which suggests that randomly sampling the entries of a similarity matrix preserves properties such as its spectral norm [1]. With this result, the Davis-Kahan theorem suggests that spectral clustering algorithms, which look at the eigenvectors of the Laplacian associated with the similarity matrix, can succeed in recovering the clusters. This intuition is formalized by Shamir and Tishby [154] who analyze a binary spectral algorithm that randomly samples  $b$  entries from the similarity matrix. They show an  $\ell_2$  bound on difference between the eigenvectors from before and after subsampling, but such a bound does not immediately translate into a strong exact recovery guarantee. Indeed, to use this bound in the constant block model that we study here, one would need  $b = \Omega(n^2)$  measurements to obtain an exact recovery guarantee, which provides essentially no improvement. Our work, translated to the flat clustering setting is much stronger; Theorem 3.2 implies that  $O(n \log n)$  similarities are needed to recover the clustering. Furthermore, we can give guarantees on the size of smallest cluster  $\Omega(\log n)$  that can be recovered in a hierarchy by *selectively* sampling similarities at each level.

Recently Voevodski *et al.* [168] proposed an interactive algorithm for flat  $k$ -way clustering that selects  $O(k)$  landmarks and partitions the objects using distances to these landmarks. Theoretically, the authors guarantee approximate-recovery of clusters of size  $\Omega(n)$  using  $O(nk)$  pairwise distances. This idea of selecting landmarks bears strong resemblance to the first phase of our interactive clustering algorithm and also has connections to the Landmark MDS algorithm of de Silva and Tenenbaum [71]. These approaches are tied to specific algorithms, while our framework is much more general. Moreover, we guarantee exact cluster recovery (under mild

assumptions) rather than approximate recovery, which translates into guarantees on hierarchical clustering. This distinction is important because of the recursive nature of hierarchical clustering.

A related direction is the body of work on efficient streaming and online algorithms for approximating the  $k$ -means and  $k$ -medians objectives (See, e.g., [47, 157]). As with Voevodski et al. [168], the guarantees for these algorithms do not immediately translate into an exact recovery guarantee, making it challenging to transform these approaches into hierarchical clustering algorithms. Moreover, the success of spectral clustering in practice suggests that an efficient spectral algorithm would also be very appealing. While there have been advances in this direction, the majority of these require the entire similarity matrix be known *a priori* [92]. Apart from [154], we know of no other spectral algorithm that optimizes the number of similarities.

Another related line of research focuses on building data structure for fast nearest neighbor computations of a point set. Many of these structures build hierarchical clusterings of the data points so that traversing the tree to find the nearest neighbor of a data point can be done in logarithmic time [28, 172]. Both the vantage point tree and the cover tree have the additional property that only  $O(n \log(n))$  distances are used to create the hierarchical clustering, which translates to both measurement and computational efficiency in our setting. The main differences are: (a) these algorithms assume a metric space, (b) the algorithms do not partition the points at each level, but rather create overlapping coverings, and (c) the algorithms insert points into the structure iteratively in contrast with the recursive partitioning of our algorithm. Further, we are not aware of any statistical analysis of these data structures for the hierarchical clustering problem.

There are also a few papers that consider alternative models of interaction for clustering problems. Two types of interaction in the literature are supervision via must-link and cannot-link constraints [27, 170], and via split or merge requests of an existing clustering [15, 18]. In these models, interactivity supplements the pairwise similarities that are available up front and enables guarantees under weaker separation assumptions. In contrast, in our setting, the similarities are not available up front and we employ interactivity to selectively obtain them. Consequently, our setting is more challenging than even the fully observed case.

Lastly, our approach loosely falls into the framework of reductions for machine learning [32]. The broad theme of this work is to leverage existing algorithms to solve more complex learning tasks, and existing results show how many prediction problems, including structured prediction [69], contextual bandits [4, 78], multi-class classification [126], can all be reduced to binary classification. Our work shows how interactive hierarchical clustering can be reduced to non-interactive non-hierarchical clustering, so that existing algorithms for the latter can immediately be applied to the former.

## 3.2 Main Results

We first clarify some notation and introduce a hierarchical clustering model that we will analyze. We refer to  $\mathcal{A}$  as any flat clustering algorithm, which takes as parameters a dataset and a natural

---

**Algorithm 4** ACTIVECLUSTER( $\mathcal{A}, s, \{x_i\}_{i=1}^n, k$ )

---

**if**  $n \leq s$  **then return**  $\{x_i\}_{i=1}^n$   
Draw  $S \subseteq \{x_i\}_{i=1}^n$  of size  $s$  uniformly at random.  
 $C'_1, \dots, C'_k \leftarrow \mathcal{A}(S, k)$ .  
Set  $C_1 \leftarrow C'_1, \dots, C_k \leftarrow C'_k$ .  
**for**  $x_i \in \{x_i\}_{i=1}^n \setminus S$  **do**  
     $\forall j \in [k], \alpha_j \leftarrow \frac{1}{|C'_j|} \sum_{x_l \in C'_j} K(x_i, x_l)$ .  
     $C_{\arg\max_{j \in [k]} \alpha_j} \leftarrow C_{\arg\max_{j \in [k]} \alpha_j} \cup \{x_i\}$ .  
**end for**  
**output**  $\{C_j, \text{ACTIVECLUSTER}(\mathcal{A}, s, C_j, k)\}_{j=1}^k$

---

number  $k$ , indicating the number of clusters to produce. Throughout the chapter,  $k$  will denote the number of clusters at any split, and we will assume that  $k$  is known and fixed across the hierarchy. We let  $n$  be the number of objects in the dataset and define  $s$  to be a parameter to our algorithm, influencing the number of measurements used by our algorithm. The parameter  $s$  reflects a tradeoff between the measurement overhead and the accuracy; increasing  $s$  increases the robustness of our method at the cost of requiring more measurements. Finally, our algorithms employ an abstract, possibly noisy similarity function  $K$ , which can model both cases where similarities are measured directly and where they are computed via some kernel function based on observed object features.

**Definition 3.1.** A  $k$ -way **hierarchical clustering**  $\mathcal{C}$  on objects  $\{x_i\}_{i=1}^n$  is a collection of clusters such that  $C_0 \triangleq \{x_i\}_{i=1}^n \in \mathcal{C}$  and for each  $C_i, C_j \in \mathcal{C}$  either  $C_i \subset C_j, C_j \subset C_i$  or  $C_i \cap C_j = \emptyset$ . For any cluster  $C$ , if  $\exists C'$  with  $C' \subset C$ , then there exists a set  $\{C_i\}_{i=1}^k$  of disjoint clusters such that  $\bigcup_{i=1}^k C_i = C$ .

Every hierarchical clustering  $\mathcal{C}$  has a parameter  $\eta$  that quantifies how balanced the clusters are at any split. Formally,  $\eta \geq \max_{\text{splits}\{C_1, \dots, C_k\}} \frac{\max_i |C_i|}{\min_i |C_i|}$ , where each split is a non-terminal cluster, partitioned into  $\{C_i\}_{i=1}^k$ .  $\eta$  upper bounds the ratio between the largest and smallest clusters sizes across all splits in  $\mathcal{C}$ . This type of balancedness parameter has been used in previous analyses of clustering algorithms [16, 87], and it is common to assume that the clustering is not too unbalanced. For clarity of presentation, we will state our results assuming  $\eta = O(1)$ , although our proofs contain a precise dependence between the sampling parameter  $s$  and  $\eta$ .

### 3.2.1 An Interactive Clustering Framework

Our primary contribution is the introduction of a novel framework for hierarchical clustering that is efficient both in terms of the number of similarities used and the algorithmic running time. To recover any single split of the hierarchy, we run a flat clustering algorithm  $\mathcal{A}$  on a small subset of the data to compute a seed clustering of the dataset. Using this initial clustering, we place each remaining object into the seed cluster for which it is most similar on average. This results in a flat clustering of the entire dataset, using only similarities to the objects in the small subset.

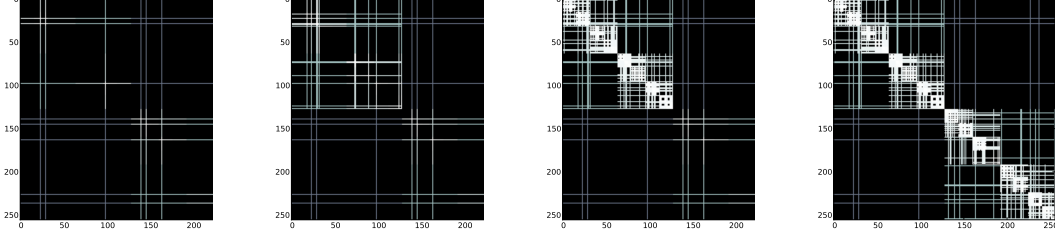


Figure 3.1: Sampling pattern of Algorithm 4

By recursively applying this procedure to each cluster, we obtain a hierarchical clustering, using a small fraction of the similarities. In this recursive phase, we do not observe any measurements between clusters at the previous split, i.e. to partition  $C_j$ , we only observe similarities between objects in  $C_j$ . This results in an interactive algorithm that focuses its measurements to resolve the higher-resolution cluster structure.

Pseudocode for the meta-algorithm is shown in Algorithm 4. As a demonstration, in Figure 3.1, we show the sampling pattern of Algorithm 4 on the first and second splits of a hierarchy, in addition to the patterns at the end of the computation. Only the similarities shown in white are needed. As is readily noticeable, the algorithm uses very few similarities but is stable able to recover this hierarchical clustering.

Our main theoretical contribution is a characterization of Algorithm 4 in terms of probability of success in recovering the true hierarchy (denoted  $C^*$ ), measurement, and runtime complexity. To make these guarantees, we will need some mild restrictions on the similarity function  $K$ , which ensure that the similarities agree with the hierarchy (up to some random noise):

**K1** For each  $x_i \in C_j \in C^*$  and  $j' \neq j$ :

$$\min_{x_k \in C_j} \mathbb{E}[K(x_i, x_k)] - \max_{x_k \in C_{j'}} \mathbb{E}[K(x_i, x_k)] \geq \gamma > 0$$

where expectations are taken with respect to the possible randomness in  $K$ .

**K2** For each object  $x_i \in C_j$ , and a set of  $M_j$  objects of size  $m_j$  drawn uniformly from  $C_j \setminus \{x_i\}$ , we have:

$$\mathbb{P} \left( \min_{x_k \in C_j} \mathbb{E}[K(x_i, x_k)] - \sum_{x_k \in M_j} \frac{K(x_i, x_k)}{m_j} > \epsilon \right) \leq 2 \exp \left\{ \frac{-2m_j \epsilon^2}{\sigma^2} \right\},$$

where  $\sigma^2 \geq 0$  parameterizes the randomness in the similarity function  $K$ . Similarly, a set  $M_{j'}$  of size  $m_{j'}$  drawn uniformly from cluster  $C_{j'}$  with  $j' \neq j$  satisfies:

$$\mathbb{P} \left( \sum_{x_k \in M_{j'}} \frac{K(x_i, x_k)}{m_{j'}} - \max_{x_k \in C_{j'}} \mathbb{E}[K(x_i, x_k)] > \epsilon \right) \leq 2 \exp \left\{ \frac{-2m_{j'} \epsilon^2}{\sigma^2} \right\}$$

K1 states that the similarity from an object  $x_i$  to its cluster should, in expectation, be larger than the similarity from that object to any other cluster. This is related to the Tight-Clustering condition used by Eriksson *et al.* [87] and less stringent than earlier results which assume that within- and between-cluster similarities are constant and bounded in expectation [147]. Moreover, an assumption of this form seems necessary to ensure that one could identify the clustering with access to a non-random similarity function,  $K$ . K2 enforces that within- and between-cluster similarities concentrate appropriately. This condition is satisfied, for example, if similarities are constant in expectation, perturbed with independent subgaussian noise. We emphasize that K2 subsumes many of the assumptions of previous clustering analyses (for example [16, 147]). Moreover, if the similarity function is deterministic, then K2 is altogether unnecessary, and some improvements to our algorithm are possible (see Proposition 3.4).

Our main results characterizes Algorithm 4 under assumptions K1 and K2:

**Theorem 3.1.** *Let  $\{x_i\}_{i=1}^n$  be a dataset with true hierarchical clustering  $C^*$ , let  $K$  be a similarity function satisfying assumptions K1 and K2 and consider any flat clustering algorithm  $\mathcal{A}$  with the following property:*

**A1** *For any dataset  $\{y_i\}_{i=1}^m$  with clustering  $C'$  where  $K$  satisfies K1 and K2,  $\mathcal{A}(\{y_i\}_{i=1}^m, k)$  recovers the first split of  $C'$  with probability at least  $1 - c_1 m k e^{-m}$  for some constant  $c_1 > 0$ .*

*Then Algorithm 4, on input  $(\mathcal{A}, s, \{x_i\}_{i=1}^n, k)$ :*

**R1** *recovers all clusters of size at least  $s$  with probability:*

$$1 - c_0 n \exp\left(\frac{-s}{2(1+\eta)^2}\right) - c_1 n^2 \exp(-s) - C_\eta n k \log n \exp\left(\frac{-\gamma^2 s}{4\sigma^2(1+\eta)^2}\right) \quad (3.1)$$

*for universal positive constants  $c_0, c_1$  and for another constant  $C_\eta$  that depends only on  $\eta$ . This probability of success is  $1 - o(1)$  as long as  $s = \omega\left(\max\{1, \frac{\sigma^2}{\gamma^2}\} \log(nk)\right)$ .*

**R2** *uses  $O(ns \log n)$  similarity measurements.*

**R3** *runs in time  $O(nA_s + ns \log n)$  where  $\mathcal{A}$  on a datasets of size  $s$  runs in time  $O(A_s)$ .*

At a high level, the theorem says that the clustering guarantee for a flat, non-interactive algorithm,  $\mathcal{A}$ , can be translated into a hierarchical clustering guarantee for an interactive version of  $\mathcal{A}$ , and that this new algorithm enjoys significantly reduced measurement and runtime complexity. The only property needed by  $\mathcal{A}$  is that it recovers a flat clustering with very high probability. While the probability of success seems strangely high, we will show that for a fairly intuitive model, a simple spectral clustering algorithm meets assumption A1. Verifying that the model satisfies the conditions K1 and K2, immediately results in a guarantee for the interactive version of this spectral algorithm.

We defer the proof of this theorem, and all theoretical results in this chapter to Section 3.4. However, before proceeding, some remarks are in order. First, by plugging in the lower bound for  $s$  into the upper bound on the measurement complexity, we see that Algorithm 4 needs

---

**Algorithm 5** SPECTRALCLUSTER( $W$ )

---

Compute Laplacian  $L = D - W$ ,  $D_{ii} = \sum_{j=1}^n W_{ij}$

$v_2 \leftarrow$  smallest non-constant eigenvector of  $L$ .

$C_1 \leftarrow \{i : v_2(i) \geq 0\}$ ,  $C_2 \leftarrow \{j : v_2(j) < 0\}$

**output**  $\{C_1, C_2\}$ .

---

$O(n \log(nk) \log n)$  similarities, which is considerably less than the  $O(n^2)$  similarities required by a non-interactive algorithm. Second, at the lower bound for  $s$ , we see that unless  $\mathcal{A}$  runs in exponential time, Algorithm 4 runs in  $\tilde{O}(n)$ , which is polynomially faster than *any* clustering algorithm that observes all of the similarities, as such an algorithm must take  $\Omega(n^2)$  time.

### 3.2.2 Interactive Spectral Clustering

To make the guarantees in Theorem 3.1 more concrete, we show how to translate the result into a real guarantee for a specific subroutine algorithm  $\mathcal{A}$ . We study a simple spectral algorithm (See pseudocode in Algorithm 5) into an interactive clustering algorithm, using the analysis from Balakrishnan et al. [16]. The algorithm operates on hierarchically structured similarity matrices referred to as the **noisy Constant Block Matrices** (again from Balakrishnan et al. [16]).

We study the special case of binary hierarchical clustering, where each non-terminal cluster is partitioned into exactly two groups. As a naming convention, we identify a cluster by a string  $\xi$  of  $L$  and  $R$  symbols. The two sub-clusters of a non-terminal cluster  $C_\xi$  are  $C_{\xi \circ L}$  and  $C_{\xi \circ R}$ .

The noisy Constant Block Model is defined using this terminology as follows:

**Definition 3.2.** A similarity matrix  $W$  is a **noisy constant block matrix** (noisy CBM) if  $W \triangleq A + R$  where  $A$  is ideal and  $R$  is a perturbation matrix:

- An **ideal similarity matrix** is characterized by off-block diagonal similarity values  $\beta_\xi \in [0, 1]$  for each cluster  $C_\xi$  such that if  $x \in C_{\xi \circ L}$  and  $y \in C_{\xi \circ R}$ , where  $C_{\xi \circ L}$  and  $C_{\xi \circ R}$  are two sub-clusters of  $C_\xi$  at the next level in a binary hierarchy, then  $A_{x,y} = \beta_\xi$ . Additionally,  $\min\{\beta_{\xi \circ R}, \beta_{\xi \circ L}\} \geq \beta_\xi$ . Define  $\gamma = \min\{\min_\xi\{\min\{\beta_{\xi \circ R}, \beta_{\xi \circ L}\} - \beta_\xi\}, \beta_0\}$ , where  $\beta_0$  is the minimum overall similarity.
- A symmetric  $(n \times n)$  matrix  $R$  is a **perturbation matrix** with parameter  $\sigma$  if (a)  $\mathbb{E}(R_{ij}) = 0$ , (b) the entries of  $R$  are subgaussian, that is  $\mathbb{E}(\exp(tR_{ij})) \leq \exp\left(\frac{\sigma^2 t^2}{2}\right)$  and (c) for each row  $i$ ,  $R_{i1}, \dots, R_{in}$  are independent.

To apply Theorem 3.1, we need to verify that the assumption K1 and K2 are met and Algorithm 5 succeeds with exponentially high probability. Checking that these conditions hold provided the signal-to-noise ratio is large enough results in the following guarantees for ACTIVE SPECTRAL, the interactive version of Algorithm 5. Proof of this theorem is deferred to Section 3.4.

**Theorem 3.2.** Let  $W$  be a noisy CBM with and  $\eta = O(1)$ , and with  $n \geq n_0$ , the latter of which is a universal constant. Then for any  $m \geq s$ , ACTIVE SPECTRAL succeeds in recovering

all clusters of size  $m$  with probability  $1 - o(1)$  as long as  $s = \omega\left(\max\{1, \frac{\sigma^2}{\gamma^4}, \frac{\sigma^4}{\gamma^4}\} \log(n)\right)$ . ACTIVE SPECTRAL uses  $O(ns \log n)$  measurements and runs in  $O(ns^2 \log s + ns \log n)$  time.

This theorem quantifies the tradeoff between statistical robustness and measurement complexity for the hierarchical spectral algorithm. On one end, if  $\gamma^2/\sigma = \Omega(1)$ , then ACTIVE SPECTRAL can successfully recover clusters of size  $\log n$  while using  $O(n \log^2 n)$  measurements. At the other end of this spectrum, if  $s = \Theta(n)$ , then we can tolerate  $\frac{\gamma^2}{\sigma^2} \asymp \sqrt{\frac{n}{\log n}}$ , but can only recover clusters of size  $\Theta(n)$ . This is essentially the same as the result of Balakrishnan et. al. [16], who show that by using  $O(n^2)$  measurements, one can tolerate noise that grows fairly rapidly with  $n$ . Varying  $s$  allows for interpolation between these two extremes.

Several remarks are in order:

1. First, note that the condition  $s$  must grow faster than  $\log(n)$  implies that the smallest clusters of the hierarchy cannot be recovered. These clusters are irrecoverably buried in noise, so one should not expect that recovery is possible.
2. The condition that  $\gamma^2/\sigma = \Omega(1)$  is undesirable for several reasons. Since the similarities are bounded between zero and one,  $\gamma^2 \leq \gamma$ , so this condition is more stringent than requiring that  $\gamma/\sigma = \Omega(1)$ , which is a more natural measure for the signal-to-noise ratio. Secondly, if the minimum cluster size remains fixed as  $n$  grows,  $\gamma$  must decrease, which implies that we require  $\sigma \rightarrow 0$  for consistent recovery.
3. On the other hand if the depth of the hierarchy remains fixed as  $n$  increases, then  $\gamma$  can remain constant, so that it suffices to have  $\sigma = O(1)$  for exact recovery. Unfortunately, for this to happen, the minimum cluster size must scale linearly with  $n$ , although in this case one can still aggressively subsample the matrix to recover all of these clusters.

The proof of this Theorem is not quite a direct application of Theorem 3.1. Instead, we show that Algorithm 5 meets assumption A1 modulo a term that depends on  $\gamma$  and  $\sigma$ , and then we plug this into Equation 3.1 (replacing the second exponential term). Solving for  $s$  in the updated version of Equation 3.1 proves the theorem. If we instead directly applied Theorem 3.1, we would require the SNR to be  $\Omega(1)$  and arrive at one end of the tradeoff between robustness and measurement complexity. Our approach allows one to see how varying  $s$  affects the tolerable SNR.

### 3.2.3 Active $k$ -means clustering

It is also possible to insert Lloyd’s algorithm for  $k$ -means clustering into our framework, but we cannot prove statistical performance guarantees since it is unknown whether Lloyd’s algorithm satisfies assumption A1 for any meaningful model.  $k$ -means helps illuminate the differences between observing similarities directly and computing similarities from observed object features. Conventionally,  $k$ -means fits into the latter framework. Here, the interactive version does not enjoy a reduced measurement complexity, because all objects must be observed, but it can lead to running time improvements as fewer distance/kernel evaluations are required.

A less traditional way to use  $k$ -means is to represent each object as a  $n$ -dimensional vector of its similarity to each other object. Here, we can apply  $k$ -means to a  $n \times n$  similarity matrix, much like we can apply Spectral Clustering, and this algorithm can be made interactive using our framework. While we cannot develop theoretical guarantees for this algorithm, which we call ACTIVEKMEANS, our experiments demonstrate that it performs very well in practice.

### 3.2.4 Fundamental Limits

**Universal Limits:** We now turn to lower bounding the number of similarities needed to recover a hierarchical clustering. We first give a necessary condition on the number of similarities needed by *any* algorithm in the absence of noise. Note that this result applies to both interactive and non-interactive algorithms. We prove the following theorem:

**Theorem 3.3.** *Consider a noiseless hierarchical clustering problem on  $n$  objects with minimum cluster size  $m$ . For any algorithm  $\mathcal{A}$ , if  $\mathcal{A}$  guarantees exact recover of the true hierarchy it must be the case that  $\mathcal{A}$  has measurement complexity*

$$\frac{n \log_2 \left( \frac{n}{m\epsilon} \right)}{\log_2 [\log_2(n/m) + 1]}.$$

This theorem asserts that  $\Omega(n \log n / \log \log n)$  measurements are necessary to recover all constant sized clusters, even in the absence of noise. The proof uses two main ideas. First, we use a combinatorial argument to count the total number of hierarchical clusterings on  $n$  objects with minimum cluster size  $m$ . Then, we use an adversarial construction, whereby a learner attempts to identify the clustering while an adversary attempts to hide it. In similar spirit to version-space algorithms, we show that for any query made by the algorithm, the adversary can provide a response that does not significantly reduce the number of consistent clusterings. Combining this with the counting argument gives a necessary condition on the number of queries any algorithm must make.

Notice that Theorem 3.1 shows that Algorithm 4 uses  $\Omega(n \log^2 n)$  similarities to recover clusters up to size  $\log n$ . This difference in measurement complexity in comparison with the necessary condition in Theorem 3.3 is due to the fact that Algorithm 4 was designed to be robust to noise, and to get closer to the fundamental limit, one must build a more brittle algorithm. Specifically, one can nearly achieve the limit in Theorem 3.3 by a simple algorithm that samples one point, thresholds the similarities between all of the objects and that point to form two clusters, and then recursively partitions. We show rigorously that this algorithm will recover all of the clusters and that it uses  $O(n \log n)$  similarities, which we summarize in the following:

**Proposition 3.4.** *Let  $\mathcal{C}^*$  be a 2-way hierarchy with balance factor  $\eta = 1$  where  $K$  satisfies K1 and K2 with  $\sigma = 0$ . Then there exists an algorithm that uses  $n \log(n/m)$  similarities and deterministically recovers all clusters of size at least  $m$  in  $\mathcal{C}^*$ .*

This shows that roughly  $n \log n$  similarities are necessary and sufficient to recover hierarchical clusterings on  $n$  objects. Interestingly, if  $m$  is large, then far fewer similarities are necessary, and



as we will see non-interactive algorithms come close to achieving these fundamental limits. If  $m = n/2$ , or just one partition is required, the algorithm used to prove Proposition 3.4 is actually non-interactive, and it meets the necessary condition established in Theorem 3.3, which applies to both interactive and non-interactive algorithms. On the other hand, for  $m \ll n$ , interactivity appears to be necessary to achieving low measurement complexities.

**Limits on non-interactive algorithms:** In order to demonstrate performance gains from the interactive sampling model, it is important to establish necessary conditions on non-interactive procedures. This is precisely the content of our next result.

To state the theorem precisely, we need some new definitions. We define a class of models  $\mathcal{H}(n, m, \gamma)$  as the set of all hierarchical clusterings on  $n$  objects where the minimum cluster size is  $m$  and the difference between within and between cluster similarities is  $\gamma$  at every level of the hierarchy. Thus, every model  $\mathcal{C} \in \mathcal{H}(n, m, \gamma)$  corresponds to a  $n \times n$  similarity matrix  $M[\mathcal{C}]$ . In the non-interactive setting, we are given a sensing budget  $\tau$  and are allowed to distribute this sensing budget across the coordinates of the similarity matrix. A sensing strategy is a matrix  $B \in \mathbb{R}_+^{n \times n}$  such that  $\sum_{i,j} b_{ij} \leq \tau$ . Given this, our observation is the matrix:

$$A_{ij} = M_{ij}[\mathcal{C}] + B_{ij}^{-1/2} \mathcal{N}(0, 1) = \mathcal{N}(M_{ij}[\mathcal{C}], B_{ij}^{-1})$$

Note that this setup is a generalization of non-interactive sampling for hierarchical clustering (with  $\sigma^2 = 1$ ) considered earlier, as one can obtain a non-integral number of samples per similarity, rather than just a single sample for a subset of the similarities. A typical sampling approach for hierarchical clustering has  $B_{ij} \in \{0, 1\}$  for all  $i, j$  with  $\sum_{i,j} B_{ij}$  as the measurement budget. Our set up strictly generalizes this class of sampling strategies.

Given such an observation, the goal of a recovery algorithm  $T$  will be to identify the model  $\mathcal{C}$  with low probability of error. Specifically we are interested in lower bounding the minimax risk:

$$R(\mathcal{H}(n, m, \gamma), \tau) = \inf_{B: \|B\|_1 \leq \tau, T} \sup_{\mathcal{C} \in \mathcal{H}(n, m, \gamma)} \mathbb{P}[T(A) \neq \mathcal{C}]$$

Notice that this probability of error is exactly the same as the probability that the algorithm  $T$  fails to recover all clusters of size  $m$ . This is a special case of the structured normal means problem studied in Chapter 5. A consequence of Theorem 5.5 and Proposition 5.6 is the following:

**Theorem 3.5.** *If  $n, m$  are both powers of two, then the minimax risk, when observing the entire matrix, is bounded from below by  $1/2$  when  $\gamma \leq \sqrt{\frac{\log(nm/6)}{(8m-4)}}$ . Under budget constraint  $\tau$ , the minimax risk is bounded from below by  $1/2$  when:*

$$\gamma \leq \sqrt{\frac{\binom{n}{2}}{\tau(8m-4)} \log(nm/6)}$$

The first part of this theorem, where the entire matrix is observed, was established by Balakrishnan et al. [16], although our proof technique is more general. The second part of the theorem

is entirely new, and it lower bounds the performance of any passive sampling strategy followed by any recovery algorithm for the hierarchical clustering problem. To compare with our interactive approach and the guarantee in Theorem 3.2, we set  $\gamma = \Theta(1)$  and solve for  $\tau$ , arriving at  $\tau \asymp \frac{n^2 \log(nm)}{m}$ . We compare this bound to our interactive approach.

To do this comparison, note that since we are in a constant SNR regime, we can simply apply Theorem 3.2. This result says that the interactive procedure uses  $O(ns \log(n))$  measurements and recovers all clusters of size  $m \geq s$ , provided that  $s = \omega(\log n)$ . If we set  $s = \Theta(\log^2 n)$ , we see that we can recover clusters of size  $m$  with sensing budget *independent* of  $m$ , i.e. with sensing budget  $\Theta(n \text{polylog}(n))$ , provided that  $m = \Omega(\log^2 n)$ . Non-interactive procedures can achieve nearly linear sensing budget only if the smallest cluster sizes are also nearly linear.

This shows significant performance gain due to interactive sampling, and this gain is most striking when recovering big clusters, at the top of the hierarchy and small clusters towards the bottom. This is line with the main theme of our thesis, that interactive procedures are particularly powerful in the presence of non-uniformity, which in this case relates to the cluster sizes.

### 3.3 Experimental Results

In this section we describe our empirical evaluation of the interactive clustering approaches described in Section 3.2. We start with several practical considerations.

#### 3.3.1 Practical Considerations

ACTIVESPECTRAL as stated has some shortcomings that enable theoretical analysis but that are undesirable for practical applications. Specifically, the fact that  $k$  is known and constant across splits in the hierarchy and the balancedness condition are both assumptions that are likely to be violated in any real-world setting. We therefore develop a variant of ACTIVESPECTRAL, called HEURSPEC, with several heuristics.

First, we employ the popular eigengap heuristic, in which the number of clusters  $k$  is chosen so that the gap in eigenvalues  $\lambda_{k+1} - \lambda_k$  of the Laplacian is large. Secondly, we propose discarding all subsampled objects with low degree (when restricted to the sample) in the hopes of removing underrepresented clusters from the sample. In the averaging phase, if an object is not highly similar to any cluster represented in the sample, we create a new cluster for this object. We expect that in tandem, these two heuristics will help us recover small clusters. By comparing the performance of HEURSPEC to that of ACTIVESPECTRAL, we indirectly evaluate these heuristics.

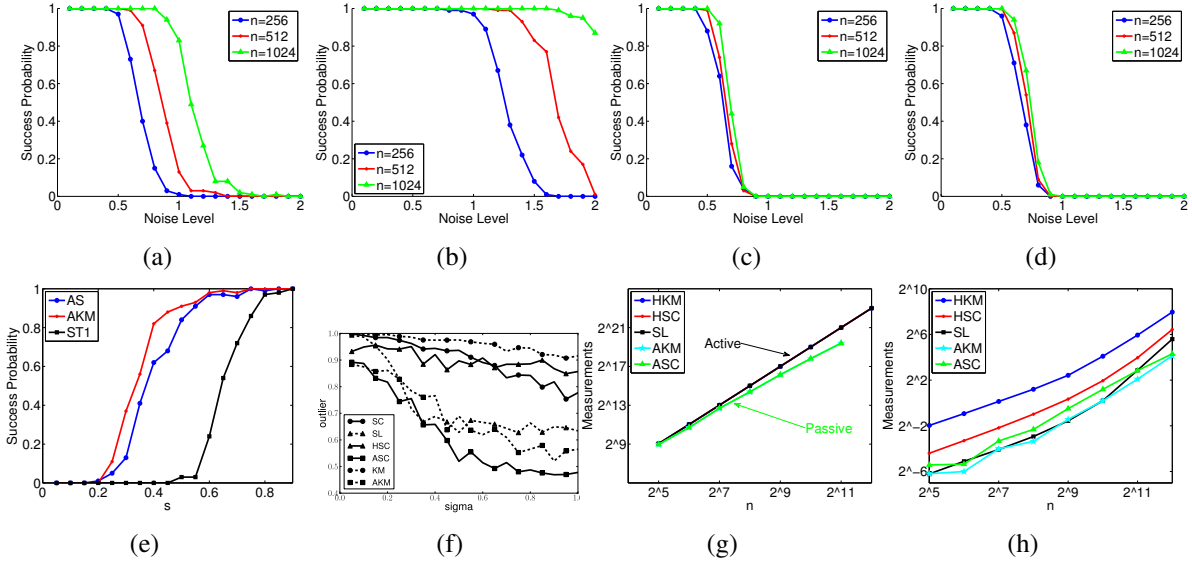


Figure 3.2: Simulation experiments. Top row: Noise thresholds for Algorithm 5, k-means clustering, ACTIVE SPECTRAL, and ACTIVE K MEANS with  $s = \log^2(n)$  for interactive algorithms. Bottom row from left to right: probability of success as a function of  $s$  for  $n = 256, \sigma = 0.75$ , outlier fractions on noisy CBM, probing complexity, and runtime complexity.

### 3.3.2 Simulations

In this section we present some empirical results on synthetic data. By Theorem 3.2, we expect ACTIVE SPECTRAL to be robust to a constant amount of noise  $\sigma$ , meaning that it will recover all sufficiently large splits with high probability. In comparison, Balakrishnan *et al.* [16], show that spectral clustering can tolerate noise growing with  $n$ . We contrast these guarantees by plotting the probability of successful recovery of the first split in a noisy CBM as a function of  $\sigma$  for different  $n$  in Figure 3.2. 3.2(a) demonstrates that indeed the noise tolerance of spectral clustering grows with  $n$  while 3.2(c) demonstrates that ACTIVE SPECTRAL enjoys constant noise tolerance. Figures 3.2(b) and 3.2(d) suggest that similar guarantees may hold for  $k$ -means and ACTIVE K MEANS.

Our theory also predicts that increasing the sampling parameter improves the performance of ACTIVE SPECTRAL. To demonstrate this, we plot the probability of successful recovery of the first split of a noisy CBM of size  $n = 256$  as a function of  $s$  for fixed noise variance. We compare three algorithms, ACTIVE SPECTRAL, ACTIVE K MEANS, and Algorithm 1 of Shamir and Tishby [154], which subsamples entries of the similarity matrix. In theory, ACTIVE SPECTRAL requires  $\Omega(n \log n)$  total measurements to recover a single split, whereas Shamir and Tishby [154] show that their algorithm requires  $\Omega(n \log^{3/2} n)$  (recall that this does not immediately translate into a clustering guarantee). Figure 3.2(e) demonstrates that this improvement is also noticeable in practice.

The simulations in Figures 3.2(a)-(e) only examine the ability of our algorithms to recover the

first split of a hierarchy, while our theory predicts that all sufficiently large clusters can be reliably recovered. One way to measure this is the **outlier fraction** metric between the clustering returned by an algorithm and the true hierarchy [87]. For any triplet of objects  $x_i, x_j, x_k$  we say that the two clusterings **agree** on this triplet if they both group the same pair of objects deeper in the hierarchy relative to the third object and disagree otherwise. The outlier fraction is simply the fraction of triplets for which the two clusterings agree.

In Figure 3.2(f), we plot the outlier fraction for six algorithms as a function of  $\sigma$  on the noisy HBM. The algorithms are: Hierarchical Spectral (SC), Single Linkage (SL), HEURSPEC (HSC), ACTIVESPECTRAL (ASC), Hierarchical  $k$ -Means (KM), and ACTIVEKMEANS (AKM). These experiments demonstrate that the non-interactive algorithms (except single linkage) are much more robust to noise than the corresponding interactive ones, as predicted by our theory, but also that the heuristics described in Section 3.3.1 have dramatic impact on performance.

Lastly, we verify the measurement and run time complexity guarantees for our interactive algorithms in comparison to the non-interactive versions. In Figure 3.2(g) and 3.2(h), we plot the number of measurements and running time as a function of  $n$  on a log-log plot for each algorithm. The three non-interactive algorithms have steeper slopes than the interactive ones, suggesting that they are polynomially more expensive in both cases.

### 3.3.3 Real World Experiments

To demonstrate the practical performance of our framework, we apply our algorithms to three real-world datasets and one additional synthetic dataset. The datasets are: The set of articles from NIPS volumes 0 through 12 [148], a subset of NPIC500 co-occurrence data from the Read-the-Web project [131] which we call RTW, a SNP dataset from the HGDP [138], and a synthetic phylogeny dataset produced using `phyclus` [49].

The NIPS dataset consists of 1740 machine learning research articles from Neural Information Processing Systems Volumes 0-12. Each article was converted into a TF-IDF vector and pairs of vectors were compared using cosine similarity.

The RTW data is a subsampled version of the NPIC500 co-occurrence dataset. It originally consisted of 88k noun-phrases and 99k contexts with NP-context co-occurrence information. We further down-sampled to 2000 NPs and used TF-IDF and cosine similarity to construct a noun-phrase by noun-phrase co-occurrence matrix.

The SNP dataset consists of base pair information at 2810 loci for 957 individuals. The dataset is annotated into three levels, where each individual is assigned a population, country of origin, and continent. Each individual has two haplotype sequences, and we arbitrarily chose the maternal haplotype. We measure similarity using edit distance. In this case, computing all pairwise similarities is computationally intensive; it took over 1 hour for this computation.

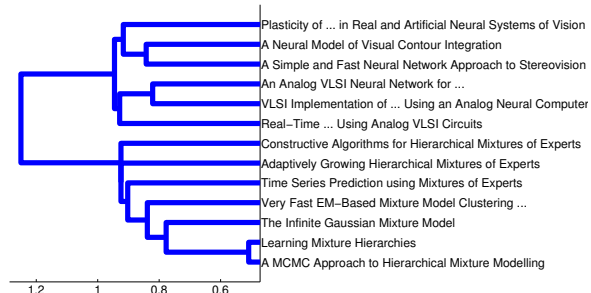
The phylogeny dataset is a synthetic phylogeny generated by the `phyclus` R package. It consists of 2048 genetic sequences, each consisting of 2000 base pairs. `phyclus` also generates a

Algorithm	HKM	HRC	Probes	Time (s)
SNP				
HEURSPEC	0.022	475	0.38	1350
ACTIVESPECTRAL	0.019	19.1	0.13	450
ACTIVEKMEANS	0.018	12.5	0.12	420
$k$ -means	0.0028	18.7	1	160
Spectral	0.0075	130	1	5660
Phylo				
HEURSPEC	0.020	371	0.29	2500
ACTIVESPECTRAL	0.012	22.9	0.071	600
ACTIVEKMEANS	0.012	25	0.071	555
$k$ -means	0.0017	22.9	1	967
Spectral	0.0022	23.5	1	997
NIPS				
HEURSPEC	0.0088	65.7	0.19	140
ACTIVESPECTRAL	0.010	1.5	0.094	79.4
ACTIVEKMEANS	0.011	1.37	0.12	29
$k$ -means	0.0017	1.66	1	723
Spectral	0.0033	6.30	1	26200
RTW				
HEURSPEC	0.0079	18.1	0.41	419
ACTIVESPECTRAL	0.0084	0.64	0.13	151
ACTIVEKMEANS	0.0073	0.485	0.22	70.9

(a)

Algorithm	SNP	Phylo
HEURSPEC	0.596	0.878
ACTIVESPECTRAL	0.374	0.971
ACTIVEKMEANS	0.383	0.94

(b)



(c)

Figure 3.3: Experiments: 3.3(a): Comparison of algorithms on various datasets. 3.3(b): Outlier fractions on datasets with ground truth clustering. 3.3(c): Subset of the NIPS hierarchy.

reference phylogeny that serves as ground truth. As with the SNP data, we measured similarity using edit distance. Computing all pairs of similarities took over 4 hours.

In the phylogeny and SNP datasets, we have access to a reference tree that can be used in our evaluation. In these cases we can report the outlier fraction, as we did in simulation. However, the other datasets lack such ground truth and, without it, evaluating the performance of each algorithm is non-trivial. Indeed, there is no well-established metric for this sort of evaluation.

For this reason, we employ two distinct metrics to evaluate the quality of hierarchical clusterings. They are a hierarchical  $K$ -means objective (HKM) [115] and an analogous hierarchical ratio-cut (HRC) objective, both of which are natural generalizations of the  $k$ -means and ratio cut objectives respectively, averaging across clusters, and removing small clusters as they bias the objectives. Formally, let  $\mathcal{C}$  be the hierarchical clustering and let  $\bar{\mathcal{C}}$  be all of the clusters in  $\mathcal{C}$  that are larger than  $\log n$ . For each  $C \in \bar{\mathcal{C}}$  let  $x_C$  be the cluster center. Then:

$$\begin{aligned} \text{HKM}(\mathcal{C}) &= \frac{1}{|\bar{\mathcal{C}}|} \sum_{C \in \bar{\mathcal{C}}} \frac{1}{|C|} \sum_{x_j \in C} \frac{x_j^T x_C}{\|x_j\| \|x_C\|} \\ \text{HRC}(\mathcal{C}) &= \frac{1}{|\bar{\mathcal{C}}|} \sum_{C \in \bar{\mathcal{C}}} \sum_{C_k \subseteq C} \frac{K(C_k, C \setminus C_k)}{2|C_k|} \end{aligned}$$

In Table 3.3(a) and 3.3(b), we record experimental results across the datasets for our algorithms.

On the read-the-web dataset, we were unable to run the non-interactive algorithms. On the SNP and phylogeny datasets, we include computing similarities via edit distance in the running time of each algorithm, noting that computing all pairs takes 6500 and 15000 seconds respectively. The immediate observation is that these algorithms are extremely fast; on the SNP and phylogeny datasets, where computing similarities is the bottleneck, our approach leads to significant performance improvements. Moreover, the algorithms perform well by our metrics; they find clusterings that score well according to HKM and HRC, or that have reasonable agreement with the reference clustering<sup>1</sup>.

We are also interested in more qualitatively understanding the performance of these algorithms. For the NIPS data, we manually collected a small subset of articles and visualized the hierarchy produced by ACTIVEKMEANS restricted to these objects. The hierarchy in Figure 3.3(c) is what one would expect on the subset, attesting to the performance ACTIVEKMEANS. On the other hand, this same evaluation on the RTW data demonstrates that interactive algorithms do not perform well on this dataset, while the non-interactive algorithms do. We suspect this is because the RTW dataset consists of many small clusters that do not get sampled by our approach.

For the SNP and phylogeny datasets, the permuted heatmaps are clear enough to be used in qualitative evaluations. These heatmaps are shown in Figure 3.4, and they suggest that all three interactive algorithms perform very well on these datasets. Heatmaps for the remaining datasets are less clear, but are included for completeness.

## 3.4 Proofs

In this section we provide proofs for all theorems in this chapter.

### 3.4.1 Proof of Theorem 3.1

Before beginning the proof of the three claims in Theorem 3.1, we first state and prove two simple lemmas bounding the number of splits and levels in a balanced hierarchy.

**Lemma 3.6.** *A  $k$ -way hierarchical clustering on  $n$  objects has at most  $\frac{n}{k-1}$  splits.*

*Proof.* A hierarchical clustering can be represented as a rooted tree  $\mathcal{T}$ , where each leaf is a singleton cluster and each internal node corresponds to a cluster containing all objects below this node. Every  $k$ -way hierarchy can be represented by a  $k$ -ary tree and the number of internal nodes in the  $k$ -ary tree exactly corresponds to the number of splits in the  $k$ -way hierarchy. Let  $f(x)$  be the number of internal nodes in a  $k$ -ary tree with  $x$  leaves. It is easy to see that the recurrence

<sup>1</sup> The SNP dataset is a  $k$ -way hierarchy and our algorithms (apart from HEURSPEC) recover binary hierarchies that cannot have high agreement with the reference.

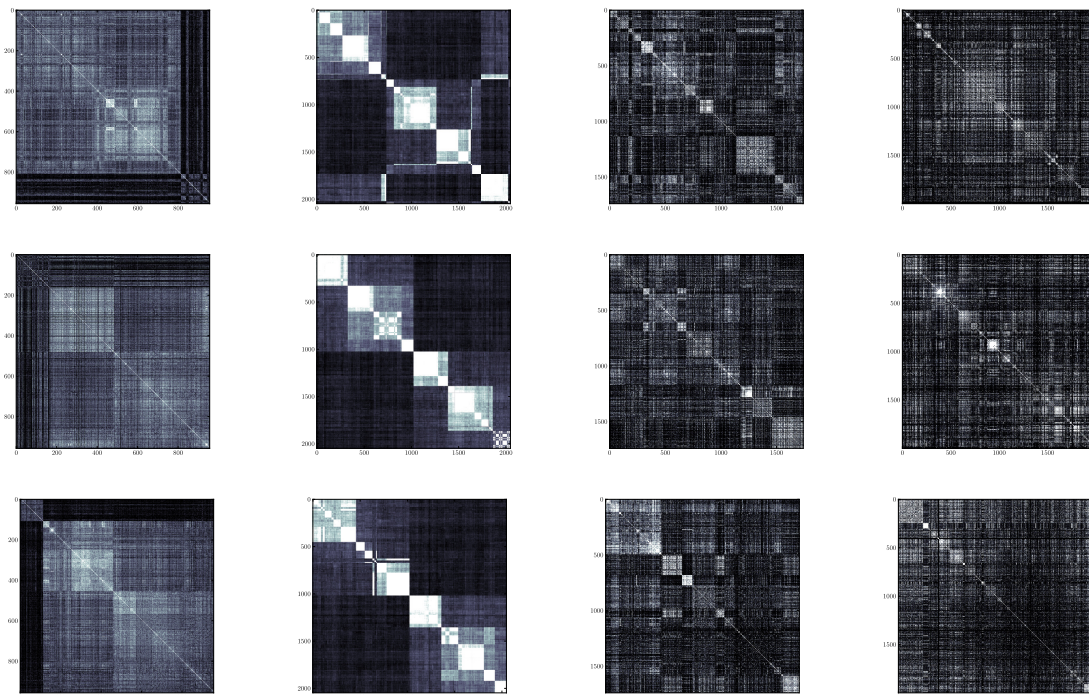



Figure 3.4: Heatmaps of permuted matrices for SNP, Phylo, NIPS, and RTW (from left to right). Algorithms are HEURSPEC, ACTIVESPECTRAL, and ACTIVEKMEANS from top to bottom.


$f(x) = f(x - k + 1) + 1$  holds for all  $x \geq k$  and  $f(x) = 1$  for all  $0 < x < k$ . Solving this recurrence, we see that  $f(n) \leq \frac{n}{k-1}$  proving the Lemma. 

**Lemma 3.7.** Let  $\eta$  be the balance factor of the hierarchy and let  $l$  be the total number of levels in the hierarchy. Then:

$$l \leq \frac{1}{\log\left(\frac{1+\eta}{\eta}\right)} \log n \leq C_\eta \log n$$

*Proof.* Note that for any split, the larger of the two clusters has  $\frac{\eta}{1+\eta}$  fraction of the nodes. After  $l$  levels, we want the largest cluster to have size at most 1, or

$$\left(\frac{\eta}{1+\eta}\right)^l n \leq 1$$

Solving for  $l$  in this equation yields the result. 

We now turn to proving the theorem. In the proof, we will define several failure events and first show that the algorithm succeeds if none of the failure events occur. We will then proceed to bound the probability of each of the failure events.

We establish some notation before proceeding. In the true hierarchy, we will denote each partition problem (or split) by  $\mathcal{S}_1, \dots, \mathcal{S}_{\frac{n}{k-1}}$  (recall that by Lemma 3.6, there are at most  $\frac{n}{k-1}$  of these). Moreover, each split except for the split at the root of the hierarchy has a parent split, which is the clustering problem directly above it in the hierarchy. For a split  $\mathcal{S}_i$ , denote its parent split by  $\mathcal{S}_{\pi(i)}$  so that  $\pi(i)$  is the index of  $i$ 's parent in the hierarchy.

For each split  $\mathcal{S}_i$ , we have three types of error events: a subsampling error event, a error event on the correctness of the algorithms  $\mathcal{A}$  and an error event on the averaging phase. In the subsampling phase, we will report an error, if the subsampled balance factor for the clustering problem at split  $i$ ,  $\hat{\eta}$  is larger than  $2\eta + 1$  (we will precisely define  $\hat{\eta}$  subsequently). If  $\hat{\eta} \leq 2\eta + 1$ , then the assumption that  $\eta = O(1)$  implies that  $\hat{\eta} = O(1)$  so that  $\mathcal{A}$  can successfully cluster the subsample. Formally, these error events are defined as follows:

$$\begin{aligned} S_i &= \{\text{at split } \mathcal{S}_i, \hat{\eta} \geq 2\eta + 1\} \\ A_i &= \{\text{Algorithm } \mathcal{A} \text{ fails at split } \mathcal{S}_i\} \\ V_i &= \{\text{Averaging fails at level } \mathcal{S}_i\} \end{aligned}$$

It is easy to see that:

$$\mathbb{P}[\text{failure}] \leq \mathbb{P}\left[\bigcup_{i=1}^n S_i \cup A_i \cup V_i\right]. \quad (3.2)$$

In words, the algorithm only fails if one of these error events occurs. At this point, one could use a union bound to decompose this further into a sum of failure probabilities, but it is challenging to bound each failure probability independently of the other events. Instead, we will appeal to the following lemma to upper bound the right hand side via a more suitable decomposition.

**Lemma 3.8.** *Let  $B_0, B_1, \dots, B_t$  be events in some measurable space. Then:*

$$\mathbb{P}\left[\bigcup_{i=0}^t B_i\right] \leq \mathbb{P}[B_0] + \sum_{i=1}^t \mathbb{P}[B_i | \neg B_0, \dots, \neg B_{i-1}]$$

*Proof.* First, the following identity is straightforward:

$$\bigcup_{i=0}^t B_i = \bigcup_{i=0}^t \left( B_i \cap \bigcap_{j=0}^i \neg B_j \right)$$

Now, using a union bound and the chain rule:

$$\begin{aligned} \mathbb{P}\left[\bigcup_{i=0}^t B_i\right] &\leq \sum_{i=0}^t \mathbb{P}\left[B_i \bigcap_{j=0}^i \neg B_j\right] = \sum_{i=0}^t \mathbb{P}\left[\bigcap_{j=0}^i \neg B_j\right] \mathbb{P}\left[B_i \bigcap_{j=0}^i \neg B_j\right] \\ &\leq \sum_{i=0}^t \mathbb{P}[B_i | \neg B_0, \dots, \neg B_{i-1}], \end{aligned}$$





Where in the last step we used that probabilities must be upper bounded by 1.

Using Lemma 3.8, we can decompose the right hand side of Equation 3.2 as:

$$\begin{aligned} & \mathbb{P}[S_1] + \mathbb{P}[A_1 | \neg S_1] + \mathbb{P}[V_1 | \neg S_1, \neg A_1] + \\ & + \sum_{i=2}^{\frac{n}{k-1}} \mathbb{P}[S_i | \neg S_0, \neg A_0, \neg V_0, \dots, \neg S_{i-1}, \neg A_{i-1}, \neg V_{i-1}] + \\ & \mathbb{P}[A_i | \neg S_0, \neg A_0, \neg V_0, \dots, \neg S_{i-1}, \neg A_{i-1}, \neg V_{i-1}, \neg S_i] + \\ & \mathbb{P}[V_i | \neg S_0, \neg A_0, \neg V_0, \dots, \neg S_{i-1}, \neg A_{i-1}, \neg V_{i-1}, \neg S_i, \neg A_i] \end{aligned}$$

Next we exploit independence of events to simplify each of the expressions. In particular, we have the following independence assertions: each subsampling phase is independent of all previous error events, conditioned on the successful recovery of the corresponding parent clustering, each execution of the algorithm succeeds (or fails) independent of every previous failure event, conditioned on the success of subsampling at that split, and each averaging phase succeeds (or fails) independent of every previous failure event, conditioned on the success of sampling and the black-box algorithm at that split. With this assertions we can reduce the above expression to:

$$\mathbb{P}[S_1] + \sum_{i=2}^{\frac{n}{k-1}} \mathbb{P}[S_i | \neg A_{\pi(i)}, \neg V_{\pi(i)}] + \sum_{i=1}^{\frac{n}{k-1}} \mathbb{P}[A_i | \neg S_i] + \sum_{i=1}^{\frac{n}{k-1}} \mathbb{P}[V_i | \neg S_i, \neg A_i]$$

In the subsequent sections, we will bound each of these conditional probabilities. By showing that the sum of these conditionals is small, we will arrive an an upper bound on the failure probability of our algorithm.

## The Subsampling Phase

Here we bound the probability of the event  $S_i$ , conditioned on the successful recovery of  $S_i$ 's parent cluster. We need to demonstrate that the balance factor  $\hat{\eta}$ , restricted to the subsample, is upper bounded by  $2\eta + 1$  after subsampling  $s$  objects, and moreover we have to do this across all splits of the hierarchy. Consider one split at first; we have  $n$  objects and  $k$  clusters  $C_1, \dots, C_k$ , and define the random variables  $X_1, \dots, X_s \in [k]$  which indicates cluster membership of the  $i$ th draw. Define the estimators  $\hat{c}_j = \sum_{i=1}^s \mathbf{1}[X_i = j]$ , so that  $\mathbb{E}[\frac{\hat{c}_j}{s}] = |C_j|/n$ . In both the cases, of sampling with and without replacement, we can apply Hoeffding's inequality and union bound over the cluster  $C_i$ . Technically, in the case of sampling without replacement we must apply a bound due to Serfling [152], but it is no worse than Hoeffding's inequality for independent random variables. We obtain:

$$\forall j. \mathbb{P} \left( \left| \frac{1}{s} \hat{c}_j - \frac{|C_j|}{n} \right| > \epsilon \right) \leq 2k \exp\{-2s\epsilon^2\}$$

Using Lemma 3.6, and a union bound over the events  $S_i$ , and inverting the concentration inequality, we have that with probability  $1 - \delta_1$ , for all splits  $\mathcal{S}_i$  and cluster  $C_j$ :

$$\left| \frac{1}{s} \hat{c}_j - \frac{|C_j|}{|\mathcal{S}_i|} \right| \leq \sqrt{\frac{\log(2nk/(k-1)) + \log(1/\delta_1)}{2s}} \leq \sqrt{\frac{\log 4n + \log(1/\delta_1)}{2s}},$$

where  $|\mathcal{S}_i|$  is the total number of objects to be clustered at split  $\mathcal{S}_i$ . Using the fact that the hierarchy has balance factor  $\eta$ , which holds here since we are conditioning on successful recovery of the parent cluster at each step, we obtain:

$$\begin{aligned} \frac{1}{s} \max_j \hat{c}_j &\leq \frac{\eta}{1+\eta} + \sqrt{\frac{\log 4n + \log(1/\delta_1)}{2s}} \\ \frac{1}{s} \min_j \hat{c}_j &\geq \frac{1}{1+\eta} - \sqrt{\frac{\log 4n + \log(1/\delta_1)}{2s}}, \end{aligned}$$

and the modified balance factor  $\hat{\eta}$  is the ratio of these two quantities. Setting  $\delta_1 = 4n \exp\{\frac{-s}{2(1+\eta)^2}\}$  gives that  $\hat{\eta} \leq 2\eta + 1 = O(1)$  so that the event  $S_i$  does not hold, across all splits  $\mathcal{S}_i$ . This is the first term in Equation 3.1, and if  $s$  meets the lower bound in the theorem, we have that  $\delta_1 = o(1)$  as needed.

## The Clustering Phase

In the clustering phase, we simply need to add up the probabilities of failure for all executions of the algorithm  $\mathcal{A}$ , conditioned on the fact that the subsampling phase for this split yielded a constant balance factor. By assumption  $\mathcal{A}$  fails on an input of size  $s$  with probability  $O(sk c_1 \exp(-s))$ . With a union bound across all splits, the probability of any execution of  $\mathcal{A}$  failing is  $O(\frac{nsk c_1}{k-1} \exp(-s)) = O(n^2 c_1 \exp(-s))$  (where we used Lemma 3.6). This is the second term in the bound in Equation 3.1 and as long as  $s = \omega(\log n)$ , this failure probability is  $o(1)$ .

## The Averaging Phase

Here our goal is to show that as long as subsampling and the subroutine clustering algorithm succeeded, then the averaging phase will also succeed with high probability. The guarantees for the averaging phase follow from assumption K2. In order to ensure that we place every object in its correct cluster, we require that:

$$\frac{1}{\hat{c}_j} \sum_{x_k \in \hat{C}_j} K(x_i, x_j) > \frac{1}{\hat{c}_{j'}} \sum_{x_j \in \hat{C}_{j'}} K(x_i, x_j),$$

for all  $x_i \in C_j$ , for all  $j' \neq j$  and across all splits. Here, we say that  $\hat{C}_j = \{x_j \in C_j\} \cap \{x_j \in S\}$  and  $\hat{c}_j = |\hat{C}_j|$  for all  $j$ . By assumption K2 and a union bound, we have that:

$$\begin{aligned} \frac{1}{\hat{c}_j} \sum_{x_k \in \hat{C}_j} K(x_i, x_k) &\geq \min_{x_k \in C_j} \mathbb{E}[K(x_i, x_k)] - \sigma \sqrt{\frac{\log(C_\eta n) + \log \log n + \log(4/\delta_3)}{2\hat{c}_j}} \\ \frac{1}{\hat{c}_{j'}} \sum_{x_k \in \hat{C}_{j'}} K(x_i, x_k) &\leq \max_{x_k \in C_{j'}} \mathbb{E}[K(x_i, x_k)] + \sigma \sqrt{\frac{\log(C_\eta kn) + \log \log n + \log(4/\delta_3)}{2\hat{c}_{j'}}} \end{aligned}$$

For the within cluster similarities we union bounded over each of the  $C_\eta \log n$  levels, because each object belongs to only one cluster per level. For between cluster similarities, we union bounded over the  $C_\eta \log n$  levels and the  $k - 1 \leq k$  clusters that we will compare to for each object  $x_i$ . Both equations hold with probability  $1 - \delta_3$ , because we used  $\delta_3/2$  as the individual probability of failure. Note also that we replace  $M_j$  in assumption K2 with the sets  $\hat{C}_j$  and  $\hat{C}_{j'}$ ; because those sets are chosen uniformly at random, we can make this replacement.

Replacing  $\hat{c}_j$  and  $\hat{c}_{j'}$  both with the lower bound on the subsampled cluster sizes, arising from the bound on  $\hat{\eta}$ , and observing that if the lower bound for the first expression is larger than the upper bound for the second expression, we will make no mistakes at all splits of the hierarchy, we obtain the following lower bound on  $\gamma$ , defined in assumption K1:

$$\gamma > 2\sigma \sqrt{\frac{(1 + \eta)}{s} (\log(C_\eta kn) + \log \log n + \log(4/\delta_3))} \quad (3.3)$$

Solving this equation for  $\delta_3$  gives:

$$\delta \leq 4C_\eta nk \log n \exp \left\{ \frac{-\gamma^2 s}{4\sigma^2(1 + \eta)} \right\}. \quad (3.4)$$

This is the final term in Equation 3.1.

## Proof of R1 and R2

The measurement complexity and running time are straightforward calculations. At each level of the hierarchy we obtain at most  $ns$  similarities and there are at most  $O(\log n)$  levels of the hierarchy, so that the total measurement complexity is  $O(ns \log n)$ . As for the running time, we only ever call  $\mathcal{A}$  on problems of size  $s$ , and there are at most  $n$  such clustering problems which gives the first term in the running time. The second term is the total running time for all averaging

phases across the hierarchy, which at each level takes  $ns$  time.



### 3.4.2 Proof of Theorem 3.2

Theorem 3.2 is almost a direct application of Theorem 3.1. We must first verify that the noisy CBM family satisfies the assumption K1, K2, and that Algorithm 5 satisfies something close to

assumption A1. In the noisy CBM, since, in expectation, the within cluster similarities are at least  $\gamma$  larger than the between cluster similarities, assumption K1 is satisfied. Assumption K2 is also satisfied with  $\sigma^2$  exactly corresponding to the noise variance of the subgaussian perturbation, and this follows from the fact that subgaussian random variables enjoy exponential concentration.

To check that an assumption of A1-type is satisfied, we will have to reproduce some of the proof of Balakrishnan *et al.* [16]. All of the facts stated here without proof are from [16]. First, Lemma 7 in [16] characterizes the spectral properties of the Laplacian of the constant block matrix  $A$ , without perturbation. If the eigenvalues of  $L_A$  are  $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$  and the eigenvectors are  $v^{(1)}, \dots, v^{(n)}$ , then they show:

1.  $v^{(1)} = \frac{1}{\sqrt{n}} \mathbf{1}$  with  $\lambda_1 = 0$ .
2.  $\sqrt{\frac{1}{\eta n}} \leq |v_i^{(2)}| \leq \sqrt{\frac{\eta}{n}}$  for all  $i \in [n]$ , with  $\lambda_2 = n\beta_0$ .  $\beta_0$  is the similarity between objects that are separated at the first split of the hierarchy. Moreover, the sign pattern of  $v^{(2)}$  reveals the coarsest partition of the clustering that generates  $A$ .
3.  $\frac{n}{1+\eta}(\eta\beta_0 + \min\{\beta_L, \beta_R\}) \leq \lambda_3 \leq \frac{n}{1+\eta}(\beta_0 + \eta \max\{\beta_L, \beta_R\})$ .

Analysis of the perturbation matrix reveals the its Laplacian has spectral norm bounded by:

$$\|L_R\|_2 \leq 2\sigma\sqrt{n}\sqrt{2\log n + 2\log \frac{4}{\delta}},$$

with probability at least  $1 - \delta$ . Equipped with the spectral properties and the bound on the perturbation, we can apply the Davis-Kahan theorem [70]. Let  $L_W$  be the Laplacian matrix of  $W$  and denote the eigenvalues  $\mu_1 \leq \mu_2 \leq \dots \leq \mu_n$  and the associated eigenvectors  $u^{(1)}, \dots, u^{(n)}$ . The Davis-Kahan theorem states that:

$$\|u^{(i)} - v^{(i)}\|_2 \leq \frac{\sqrt{2}\|L_R\|_2}{\xi_i},$$

where  $\xi_i = \min_{i \neq j} |\lambda_i - \lambda_j|$ . This bound shows that the eigenvectors of  $L_W$  are close to the eigenvectors of  $L_A$ , which we know reveal the cluster structure.

However, a more refined bound is possible. Let  $k = u^{(2)} - v^{(2)}$ . We will proceed to bound  $\|k\|_\infty$ , which will give us more precise control on the eigenvector deviation. Some algebra shows that:

$$|k(i)| \leq \frac{\overbrace{|v^{(2)}(i)(\lambda_2 - \mu_2)|}^{T_1} + \overbrace{|A_i k|}^{T_2} + \overbrace{|L_{Ri} v^{(2)}|}^{T_3} + \overbrace{|R_i k|}^{T_4}}{\underbrace{|\mu_2 - D_{Ai} - D_{Ri}|}_{T_5}}$$

The term  $T_1$  is bounded through application of Weyl's inequality and the properties of  $L_A$ :

$$T_1 = |v^{(2)}(i)(\lambda_2 - \mu_2)| \leq |v^{(2)}(i)| |\lambda_2 - \mu_2| \leq \sqrt{\frac{\eta}{n}} \|L_R\|_2 \leq 2\sigma\sqrt{\eta}\sqrt{2\log n + 2\log \frac{4}{\delta}}$$

The term  $T_2$  is bounded by:

$$T_2 = |A_i k| \leq \|A_i\|_2 \|k\|_2 \leq \frac{\sqrt{n}\beta_{\max}\sqrt{2}\|L_R\|_2}{\xi_2} \leq \frac{2\sqrt{2}\sigma\sqrt{2\log n + 2\log\frac{4}{\delta}}}{\gamma\frac{\eta}{1+\eta}}$$

These two bounds hold jointly with probability at least  $1 - \delta$ .

For  $T_3$ , we decompose into  $D_{R_i}v^{(2)}(i)$  and  $R_iv^{(2)}$ . The former is a subgaussian with scale factor at most  $\sigma\sqrt{\eta}$  since  $v^{(2)}(i) \leq \sqrt{\eta/n}$  and  $D_{R_i}$  is a subgaussian with scale factor  $\sigma\sqrt{n}$ . For the latter, since  $v^{(2)}$  is a unit vector,  $R_iv^{(2)}$  is a subgaussian with scale factor  $\sigma$ . Taking a union bound across all  $i$  and using a standard sub-gaussian tail bound, we have that:

$$T_3 \leq 4\sigma\sqrt{\eta\log(2n/\delta)}$$

with probability at least  $1 - \delta$ .

For  $T_4$ , we have  $T_4 \leq |R_i k| \leq \|R_i\|_2 \|k\|_2$ , and  $\|R_i\|_2 \leq \|R\|_2$ , so for  $n$  large enough under the  $1 - \delta$  event used to bound  $T_1$ , we have:

$$\|R_i\|_2 \leq C\sigma\sqrt{n},$$

for some absolute constant  $C$ . This gives:

$$T_4 \leq C\sigma\sqrt{n}\frac{\sqrt{2}\|L_R\|_2}{\xi_2} \leq \frac{2\sqrt{2}C\sigma^2\sqrt{2\log n + 2\log\frac{4}{\delta}}}{\gamma\frac{\eta}{1+\eta}}$$

Lastly, we must lower bound  $T_5$ . We write:

$$T_5 = |\mu_2 - D_{A_i} - D_{R_i}| = |D_{A_i} + D_{R_i} - \mu_2|.$$

Note that:

$$D_{A_i} \geq \frac{n}{1+\eta}(\eta\beta_0 + \min\{\beta_L, \beta_R\}) \geq n\beta_0 + \frac{n\gamma}{1+\eta},$$

while  $\mu_2 \leq n\beta_0 + \|L_R\|_2 \leq n\beta_0 + \|D_R\|_2 + \|R\|_2$  by Weyl's inequality and the triangle inequality. This gives:

$$T_5 \geq \left| \frac{n\gamma}{1+\eta} - 2\|D_R\|_2 - \|R\|_2 \right|,$$

provided that the expression inside the absolute value is positive. We will now show this is indeed the case. Under the same  $1 - \delta$  probability event used to bound  $T_1$ , we have that:

$$2\|D_R\|_2 - \|R\|_2 \leq 3\sigma\sqrt{n}\sqrt{2\log n + 2\log\frac{4}{\delta}},$$

so provided that  $\sigma \leq \frac{\gamma\sqrt{n}}{3(1+\eta)\sqrt{2\log n + 2\log 4/\delta}}$ , we will have:

$$2\|D_R\|_2 - \|R\|_2 \leq \frac{n\gamma}{2(1+\eta)}$$

So that  $|T_5| \geq \frac{n\gamma}{2(1+\eta)}$ .

Combining all of the bounds, we have that with probability at least  $1 - 2\delta$ :

$$\|k\|_\infty \leq \frac{2(1+\eta)}{n\gamma} \left( 6\sigma\sqrt{\eta} + \frac{2\sqrt{2}\sigma}{\gamma\frac{\eta}{1+\eta}} + \frac{2\sqrt{2}C\sigma^2}{\gamma\frac{\eta}{1+\eta}} \right) \sqrt{2\log n + 2\log(4/\delta)}.$$

The algorithm succeeds if  $\|k\|_\infty \leq \sqrt{\frac{1}{m}}$ . Rearranging, if


$$\sigma \leq \frac{c\gamma}{1+\eta} \min \left\{ \frac{1}{\eta} \frac{\sqrt{n}}{\sqrt{\log n + \log(4/\delta)}}, \frac{\gamma\sqrt{\eta}}{1+\eta} \frac{\sqrt{n}}{\sqrt{\log n + \log(4/\delta)}}, \sqrt[4]{\frac{n\eta}{\log n + \log(4/\delta)}} \right\},$$

then the algorithm succeeds with probability at least  $1 - \delta$ , where  $c$  is some universal constant.

Suppressing dependence on  $\eta$  and rearranging to solve for  $\delta$ , we have:

$$\delta \leq n \exp \left( -n \min \left\{ \frac{\gamma^2}{\sigma^2}, \frac{\gamma^4}{\sigma^2}, \frac{\gamma^4}{\sigma^4} \right\} \right),$$

which meets assumption A1 provided that  $\gamma^2/\sigma = \Omega(1)$ . This is true since  $\gamma^2 \leq \gamma$ , which means that  $\gamma^p/\sigma^p$  is also  $\Omega(1)$  for any  $p \geq 1$ . We actually use this bound directly to obtain the second

term in Equation 3.1, and this allows us to obtain a guarantee for all values of  $\gamma$  and,  $\sigma$ . 

### 3.4.3 Proof of Theorem 3.3

The lower bound will be based on an adversarial construction. Let  $n$  and  $m$  be powers of two with  $n \geq m$ . We will consider a perfectly balanced binary hierarchy and say that the clusters at level  $l$  all have cluster size  $n/2^l$ . This means that  $l = 0$  corresponds to the cluster containing all of the objects and at the bottom of the hierarchy and the largest value of  $l = \log_2(n/m)$ . The similarity between two objects grouped at a level  $l$ , but not grouped at level  $l+1$  will be  $\gamma_l$ , which is fixed and known to the algorithm.

Each time the learner makes a measurement, the adversary will respond with a value that is consistent with all existing measurements, but that keeps the number of consistent hierarchical clusterings as large as possible. The goal for the learner is to whittle down the size of the consistent set until there is just a single consistent hypothesis, and as soon as this happens the learner has successfully recovered the clustering. The goal for the adversary is to provide as

little information as possible to the learner at each round, so that the learner must make many measurements before she is confident about the clustering.

For each query, the adversary responds with one of  $\{\gamma_i\}_{i=1}^l$ , so there are  $l$  possible choices. If the current size of the consistent set is  $S$ , then by the pigeon-hole principle, there must exist a choice of response by the adversary for which the size of the subsequent consistent set is at least  $S/l$ . Therefore, after  $T$  rounds, the adversary can ensure that the size of the consistent set has reduced multiplicatively by no more than  $l^T$ . If the size of the consistent set is initially  $S_0$  (we will compute this size shortly), a necessary condition for the learner's success is:

$$\frac{S_0}{l^T} \leq 1$$

We now compute the number of hierarchical clusterings that extend  $l$  levels. We start by considering all  $n!$  permutations on  $n$  objects, and aim to count the number of permutations that induce the same hierarchical clustering. Let  $T(n)$  be the number of permutations that induce the same hierarchical clustering, where the smallest clusters have size  $m$ . This means that  $T(m) = m!$ . We compute  $T(n)$  recursively: at the top level, we can permute the two clusters and then use any of the permutations on the two sub-clusters, leading to the recurrence  $T(n) = 2T(n/2)^2$ , with the base case  $T(m) = m!$ . This recurrence solves to  $T(n) = 2^{n/m-1}(m!)^{n/m}$ , so that the number of hierarchical clusterings with smallest cluster size  $m$  is:

$$S_0 = \frac{n!}{2^{n/m-1}(m!)^{n/m}}$$

The necessary condition becomes:

$$\frac{n!}{2^{n/m-1}(m!)^{n/m} \log_2(n/m)^T} \leq 1$$

Using the bound  $(n/e)^n \sqrt{2\pi n} \leq n! \leq n^n$  which follows from Stirling's approximation, the necessary condition is:

$$T \geq \frac{n \log_2(\frac{n}{me})}{\log_2[\log_2(n/m) + 1]}$$




### 3.4.4 Proof of Proposition 3.4

The algorithm we will use is the following:

1. Pick an object  $x_i$
2. Take the  $n/2$  objects  $x_j$  with the largest  $K(x_i, x_j)$  values and place them in a cluster  $C_1$ . Place the remaining objects in a cluster  $C_2$ .
3. Recursively partition  $C_1$  and  $C_2$ .

Under the assumption that  $\mathcal{C}^*$  is a balanced binary hierarchy, and under assumptions K1 and K2, this algorithm correctly recovers the clustering. This is true because between cluster similarities are strictly smaller than within cluster similarities, so every partitioning step is exact.

To recover a cluster of size  $s$ , we use exactly  $s$  similarities. Therefore, to recover all clusters up to size  $m$ , we use  $n \log(n/m)$  similarities, proving the theorem. 

### 3.4.5 Proof of Theorem 3.5

For the first claim, by Theorem 5.1, we must lower bound the quantity  $W(\mathcal{H}, \alpha)$ . We interpret  $\mathcal{H}$  as a collection of vectors defined by  $v_{\mathcal{C}} = \text{vec}(M[\mathcal{C}])$  for each  $\mathcal{C} \in \mathcal{H}$ . We must lower bound:

$$W(\mathcal{H}, \alpha) = \max_{\mathcal{C} \in \mathcal{H}} \sum_{\mathcal{C}' \neq \mathcal{C}} \exp(\|v_{\mathcal{C}} - v_{\mathcal{C}'}\|_2^2 / \alpha)$$

When  $n$  and  $m$  are both powers of two, a subset of  $\mathcal{H}$  is the set of all perfectly balanced binary hierarchical clusterings with minimum cluster size  $m$ . Let  $\mathcal{C}_0$  be one of these models. Consider perturbing  $\mathcal{C}_0$  by taking an object and swapping that object with another one in the adjacent cluster at the deepest level of the hierarchy. There are  $nm/2$  such perturbations and any perturbation  $\mathcal{C}$  has  $\|v_{\mathcal{C}_0} - v_{\mathcal{C}}\|_2^2 = \gamma^2(8m - 4)$ . This gives the lower bound of:

$$W(\mathcal{H}, \alpha) \geq \frac{nm}{2} \exp\left(\frac{\gamma^2}{\alpha}(8m - 4)\right)$$

By Theorem 5.1, if  $W(\mathcal{H}, 1) \geq 3$ , then the minimax risk is bounded from above by  $1/2$ . Applying our lower bound and solving for  $\gamma$  proves the first part of the result.

For the second claim, if we certify that the uniform sampling strategy minimizes  $W(\mathcal{H}, \alpha, B)$  under the budget constraint, then we can immediately apply Theorem 5.5. We will use Proposition 5.6 to achieve this.

Focusing only on the set of perfectly balanced binary hierarchical clusterings with minimum cluster size  $m$ , which we call  $\mathcal{H}'$ , it is easy to see that when  $\hat{B}$  is uniform, every one of these hypotheses achieves the maximum in the definition of  $W(\mathcal{H}', \alpha, B)$ . Moreover, notice that for every pair of pairs objects  $\{a, b\}, \{a', b'\}$ , there is a bijection  $p$  over  $\mathcal{H}$  based on swapping  $a$  with  $a'$  and  $b$  with  $b'$  in the hierarchy such that for any hypothesis  $v_{\mathcal{C}}$ , we have  $v_{\mathcal{C}}(a, b) = v_{p(\mathcal{C})}(a', b')$ . This claim is fairly easy to see. If in  $\mathcal{C}$ ,  $a$  and  $b$  are clustered at some level  $l$ , then by swapping  $a$  with  $a'$  and  $b$  with  $b'$  to form  $p(\mathcal{C})$ ,  $a'$  and  $b'$  are clustered at level  $l$  in  $p(\mathcal{C})$  so both terms will be identical because we are in a constant block model.



Since  $p$  is a bijection, when we take  $\pi$  to be uniform over the hypotheses, we have:

$$\begin{aligned}
& \mathbb{E}_{\mathcal{C} \sim \pi} \sum_{\mathcal{C}' \neq \mathcal{C}} (v_{\mathcal{C}}(a, b) - v_{\mathcal{C}'}(a, b))^2 \exp(-\|v_{\mathcal{C}'} - v_{\mathcal{C}}\|_2^2) \\
&= \mathbb{E}_{\mathcal{C} \sim \pi} \sum_{\mathcal{C}' \neq \mathcal{C}} (v_{p(\mathcal{C})}(a', b') - v_{p(\mathcal{C}')}(a', b'))^2 \exp(-\|v_{p(\mathcal{C}')} - v_{p(\mathcal{C})}\|_2^2) \\
&= \mathbb{E}_{\mathcal{C} \sim \pi} \sum_{\mathcal{C}' \neq \mathcal{C}} (v_{\mathcal{C}}(a', b') - v_{\mathcal{C}'}(a', b'))^2 \exp(-\|v_{\mathcal{C}'} - v_{\mathcal{C}}\|_2^2).
\end{aligned}$$

This means we may apply Proposition 5.6, which certifies that the uniform sampling minimizes the function  $W(\mathcal{H}', \alpha, B)$  under budget constraint.

Equipped with this fact, we can reproduce the calculation above but with  $B_i = \tau / \binom{n}{2}$ , giving:

$$W(\mathcal{H}, \alpha, \tau) \geq \frac{nm}{2} \exp\left(\frac{\gamma^2}{\alpha} \frac{\tau}{\binom{n}{2}} (8m - 4)\right)$$



### 3.5 Discussion

In this chapter, we developed several interactive algorithms for hierarchical clustering. These algorithms have strong computational and statistical guarantees, and in the case of spectral clustering, provably outperform all non-interactive approaches.

This chapter supports the main claim of this thesis: that interactivity leads to improvements in computational and statistical performance, particularly when datasets exhibit non-uniformity. In this chapter non-uniformity is measured in terms of the sizes of the clusters to be recovered. Our interactive clustering approach is particularly powerful when one must recover large clusters at the top levels of the hierarchy, and small clusters deeper in the hierarchy. In this case, the interactive algorithms developed here significantly improve on non-interactive approaches in both computational and statistical efficiency.



# Chapter 4

## Interactive Latent Tree Metric Learning

Knowledge of a network's topology and internal characteristics such as delay times and losses is crucial to maintaining seamless operation of network services. Yet typical networks of interest are incredibly large and decentralized so that these global properties are not directly available, but rather must be inferred from a small number of indirect measurements. Network tomography [45, 167] is a promising approach that aims to gather such knowledge using only end-to-end measurements between nodes at the periphery of a network with limited cooperation from core routers. The design of algorithms that reliably and accurately recover network characteristics from these measurements is an important research direction.

Most current methods focus on single source network tomography; they use similarity of delay or similarity of loss measurements from a single source to multiple nodes, caused by shared path segments, to infer a tree topology between the source and end nodes. The assumption of a tree topology is justified under the premise of shortest path routing from the source to each end node. These procedures either rely on infrequently deployed multicast probes ([33, 79, 80, 81]) or use a series of back-to-back, carefully coordinated, unicast probes ([56, 82, 86, 135, 166]), which makes the method sensitive to packet re-orderings and asynchrony between end nodes. These issues limit the applicability of single source tomography methods.

Multiple source network tomography is an alternative approach that uses measurements between pairs of end nodes that form an additive metric on a graph. Several network measures such as end-to-end delay, loss, or hop counts between pairs of end nodes form an (approximate) additive metric, as a path measurement is the sum of the measure along links constituting the path. It is possible to learn such metrics using light-weight measurement such as hop counts extracted from packet headers [84] or pings. If the given measurements form an additive metric on an acyclic or tree graph, a variety of methods can be used to reconstruct the underlying structure [108, 135, 137]. However, typically, the underlying graph is not an exact tree as peering links between different network providers introduce cycles and violate the tree assumption, again limiting the effect of existing methods.

Given the size and complexity of the Internet, the practicality of any network tomography al-

gorithm should be evaluated not only by its noise tolerance and robustness to violations of any modeling assumptions, but also by its measurement or probing complexity (the number of measurements/probes needed as a function of the number of end hosts in the network). State-of-the-art methods for both single- and multi-source network tomography typically suffer in at least one of these directions. Many methods do not optimize and/or provide rigorous guarantees on the number of measurements needed to recover the underlying graph structure, while others are not guaranteed to be robust to noise in these measurements. Moreover, to the best of our knowledge, no method, with the exception of [7, 142], consider violations of the assumption that the underlying topology is a tree. In this chapter, we address all of these deficiencies.

Unfortunately, additive metrics can be unidentifiable given just pairwise distance measurements, and therefore one must impose some structural restrictions. Motivated by recent work [142] showing that internet latency and bandwidth can be well *approximated* by path lengths on trees, our work, much like existing network tomography results, is grounded in a tree metric assumption. However, we introduce two models to capture violations of this assumption: (a) an *additive noise* model, where all measurements are corrupted by additive subgaussian noise, resulting in small deviations from the tree metric, and (b) a *persistent noise* model in which a fraction of the measurements are arbitrarily corrupted. The persistent noise model also captures the effects of missing measurements due to dropped packets or unresponsive nodes. Even under these noise models, our algorithms have strong guarantees about correctness and measurement complexity.

Specifically, we present two algorithms that use interactively selected light-weight probes to construct a weighted tree whose path lengths provide a faithful representation of the pairwise measurements between end hosts in the network. While the additional nodes in the resulting tree need not correspond to hidden network elements, such a representation enables distance approximations between unmeasured hosts, closest neighbor/server selection, and topology-aware clustering all of which can improve performance of network services.

Our contributions can be summarized as follows:

1. We present algorithms for the multi-source network tomography problem that improve on existing work in at least one of two regards: our algorithms have strong correctness guarantees in the presence of noisy measurements, which can capture violations of the tree-metric assumption, and, by intelligent use of light-weight probes, they come with bounds on probing/measurement complexity.
2. Our first algorithm addresses the additive noise model. It uses  $O(pl \log^2 p)$  pairwise measurements in the presence of noise and  $O(pl \log p)$  measurements in the absence of noise, where  $p$  is the number of end hosts in the network and  $l$  is the maximum degree of any node, to construct a tree that accurately reflects the measurements. As our guarantees hold even for highly unbalanced tree structures, this improves on existing work [86, 135] that requires balanced-ness restrictions.
3. Under the persistent noise model, our second algorithm uses  $O(pl \log^2 p)$  pairwise measurements to construct a tree approximation, even when a fixed fraction of the measurements are arbitrarily corrupted. Robustness to persistent noise, however, comes at the cost

of requiring some balanced-ness of the underlying tree.

This chapter also lends evidence to the three overarching themes of this thesis. While we will not fully characterize non-interactive approaches for these problems, we will see that our interactive procedures are statistically and computationally efficient in comparison with naïve non-interactive procedures. We will also see that uniformity, measured by the degree of the underlying tree affects the statistical efficiency, so that our interactive algorithms are particularly suited for non-uniform (low-degree) problems.

This chapter is organized as follows. Section 4.1 discusses related work and comparisons to our algorithms. We provide background definitions and formally specify the multi-source tomography problem in Section 4.2. Our first algorithm that uses selective pairwise measurements to recover an unrooted, unbalanced tree topology is presented in Section 4.3.1, along with an analysis of its measurement complexity and tolerance to additive noise corrupting the measurements. In Section 4.3.2, we present our main algorithm, RISING (Robust Identification using Selective Information of Network Graphs) and analyze its robustness to persistent noise as well as its measurement complexity. We validate the proposed algorithms using simulations as well as real Internet measurements from the King [98] and IPPlane datasets [129] in Section 4.4 and provide proofs in Section 4.5. We conclude in Section 4.6.

## 4.1 Related Work

Initial work towards mapping the Internet was based on injecting TTL (Time-to-Live)-limited probe packets called `traceroutes` that record the exact path traversed by the packet [73, 159]. Since `traceroute` is based on augmenting Time-To-Live information in packets, `traceroute`-based tomography approaches are inconsistent when there are several paths between two network elements. Moreover, anonymous routers [171] and router aliases [99] do not augment packet headers, and firewalls as well as network address translation (NAT) boxes simply block `traceroute` packets, posing significant challenges to `traceroute`-based tomography.

Among the various algorithms for single-source tomography, two recent methods are particularly relevant to our work: the DFS-ordering algorithm of Eriksson et al. [86] and the work of Ni et al. [135]. The first provably uses  $O(pl \log p)$  probes to recover a balanced  $l$ -ary tree topology; however, the authors make no claims about the correctness of the algorithm in the presence of noisy measurements. Ni et al. present the Sequential Logical Topology (SLT) algorithm, that uses  $O(pl \log p)$  ( $O(pl \log^2 p)$  under additive noise) probes to recover balanced  $l$ -ary trees while also guaranteeing correct recovery of the topology when measurements are corrupted by additive noise. Our first algorithm improves on the work of Ni et al. by relaxing the balanced-ness assumption while maintaining the same measurement complexity.

In multi-source tomography, a number of algorithms [61, 85, 90] find Euclidean or non-Euclidean embeddings that accurately reflect the measurements. While some of these algorithms have strong measurement complexity guarantees [85], they do not capture the inherent hierarchi-

cal structure of the network and thus may be less useful than algorithms that recover tree or more intuitive models. In addition to the embedding-based algorithms, the work of Rabbat and Nowak [140] casts the multi-source tomography problem as a set of statistical hypothesis test that differentiates topological structure between two senders and two receivers. While their approach is algorithmically more straightforward, they only identify the presence of a shared link between the senders and the receivers and cannot distinguish all possible topological configurations between four end hosts as we can.

If the measurements form an additive tree metric, then a host of algorithms could be used to build a tree representation [108, 137, 145], some coming with measurement complexity bounds. However, the tree metric assumption does not hold in practice, and as shown in [142], network measurements such as latency and bandwidth only *approximate* additive tree metrics. It is consequently important to design algorithms that are robust to violations of the tree metric properties.

Sequoia [142] is one algorithm designed for this purpose. Unfortunately, it comes with no guarantees on correctness in the presence of these violations, and while it seems to use only a limited number of probes in practice, it lacks measurement complexity bounds. In this paper, we build on this line of work by designing an algorithm with theoretical guarantees on correctness and measurement complexity. Another method that addresses more general graph structures, beyond trees, was proposed recently in [7]. However, this method also does not optimize the measurement complexity.

Our work, and network tomography in general, have strong connections to the task of learning the structure of latent variable graphical models and to problems in phylogenetic inference. For example, in [137] and [54], algorithms are proposed to learn tree-structured graphical models using pairwise empirical correlations obtained from measurements of variables associated with leaf nodes. Under this setup, the correlations form an exact, rather than approximate, tree metric. Moreover, due to the different measurement model, this work does not explicitly optimize the number of pairwise measurements used. Our first algorithm is indeed based on the work of Pearl and Tarsi [137] and hence we call it PEARLRECONSTRUCT.

In phylogenetics, the task of learning an evolutionary tree using genetic sequence data from several extant species is closely related to the single-source tomography problem. Several algorithms, such as the neighbor-joining algorithm [94, 135, 151] have been applied to both problems. Also see [33], [68], and [88] for more details. To the best of our knowledge, the algorithms we propose are novel and do not exist in the phylogenetics literature.

## 4.2 Background

Let  $\mathcal{X} \triangleq \{x_i\}_{i=1}^p$  denote the end hosts in a network and let  $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$  be a function representing the true distances between the nodes, so that  $d(x_i, x_j)$  is the distance, as measured in the network, between the hosts  $x_i$  and  $x_j$ .

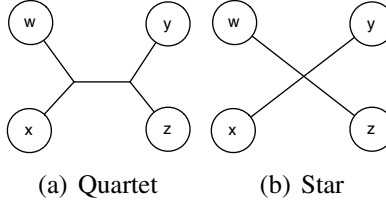


Figure 4.1: Possible structures for four leaves in a tree. If  $d(w, x) + d(y, z) < d(w, y) + d(x, z) = d(w, z) + d(x, y)$  then structure and labeling is that of (a). If  $d(w, x) + d(y, z) = d(w, y) + d(x, z) = d(w, z) + d(x, y)$  then structure is a star (b).

Our work focuses on distance functions  $d$  that form approximate additive tree metrics. Specifically, let  $\mathcal{T} = (\mathcal{V}, \mathcal{E}, c)$  be a weighted tree with vertices  $\mathcal{V}$ , edges  $\mathcal{E}$  and weights  $c$ , for which  $\mathcal{X}$  is the set of leaves. To avoid identifiability issues, our focus will be on **minimal** trees, for which each internal node has degree  $\geq 3$  and each edge has strictly positive weight. An additive tree metric on  $\mathcal{X}$  is a function  $d_{\mathcal{T}}$  such that  $d_{\mathcal{T}}(x_i, x_j) \triangleq \sum_{(x_k, x_l) \in \text{Path}(x_i, x_j)} c(x_k, x_l)$ , that is the distance between two points is the sum of the edge weights along the *unique* path between them. A useful property of additive tree metrics is the **four-point condition**:

**Definition 4.1.** A metric  $(\mathcal{X}, d)$  satisfies the **four-point condition** (4PC) if for any set of points  $w, x, y, z \in \mathcal{X}$  ordered such that  $d(w, x) + d(y, z) \leq d(w, y) + d(x, z) \leq d(w, z) + d(x, y)$ ,  $d(w, y) + d(x, z) = d(w, z) + d(x, y)$ .

The 4PC is related to the *quartet test*, a common technique for resolving tree structures (Indeed, there are a host of quartet-based algorithms for phylogenetic inference, for example [143]). The quartet test is used to identify the structure between any four leaves in a tree using only the pairwise distances between those leaves. It is easy to see that any four leaves either form a structure like that in Figure 4.1(a) or a star (Figure 4.1(b)), and using the 4PC we can identify not only which structure but also the correct labeling of the leaves (See Figure 4.1 for more details).

Any metric that satisfies the four-point condition is a tree metric for some tree. Unfortunately, latency and hop counts in real networks do not exactly fit into this framework, but only approximate tree metrics [142]. One characterization of this approximation is the 4PC- $\epsilon$  condition which requires  $d(w, z) + d(x, y) \leq d(w, y) + d(x, z) + 2\epsilon \min\{d(w, x), d(y, z)\}$  for some parameter  $\epsilon$  instead of the equality in Definition 4.1. Metrics for which  $\epsilon$  values are low can be well approximated by tree metrics, and empirical studies showing that real network measurements satisfy 4PC- $\epsilon$  for small values of  $\epsilon$  motivates the use of this model.

In this work, we take a more statistical approach and instead assume that  $d(x_i, x_j) = d_{\mathcal{T}}(x_i, x_j) + g(x_i, x_j)$  where the function  $g$  models the networks deviations from a tree metric. This approach allows us to not only formally state the multi-source network tomography problem but also to make rigorous guarantees about the performance of our algorithms. We focus on two models for these deviations:

1. **Additive Noise Model** – In this model,  $g(x_i, x_j)$  is drawn from a subgaussian with  $\sigma^2$  as

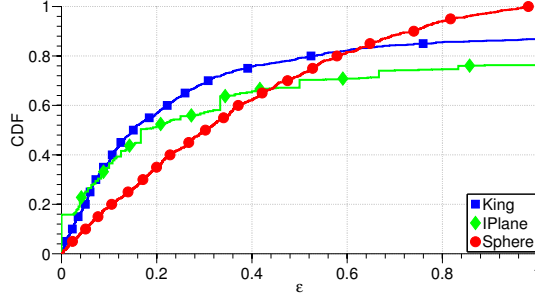


Figure 4.2: CDFs of  $\epsilon$  values in the 4PC- $\epsilon$  condition for two real world datasets (King [98] and IPlane datasets [129]) along with a dataset of points drawn uniformly from the surface of a sphere, where geodesic distance defines the metric.

a scale factor<sup>1</sup>. The small perturbation model studied in single source network tomography (See for example [135]) is similar to this as subgaussian noise is bounded, with high probability, by a small constant (depending on  $\sigma^2$ ). This model captures the inherent randomness in certain types of measurements, such as latencies. Under this formulation we allow each measurement to be observed several ( $n$ ) times.

2. **Persistent Noise Model** – Here  $g(x_i, x_j) = 0$  with probability  $q$ , independent of all other  $x_i$  and  $x_j$ , and with probability  $1 - q$ ,  $g(x_i, x_j)$  is arbitrary (or adversarially) chosen. We believe this is a reasonable model of how the measurements do not exactly form a tree metric, due to violations caused by peering links, unresponsive nodes or missing measurements. To more accurately model violations of tree metric assumptions, multiple request for a measurement all reveal the same (possibly incorrect) value, so we only obtain one sample of each measurement. To the best of our knowledge, there are no other efforts to study this noise model.

While [142] capitalized on the fact that  $\sim 80\%$  of the quartets satisfy 4PC with a small perturbation  $\epsilon$ , we also note that  $\sim 20\%$  of the quartets do not satisfy the 4PC even with  $\epsilon = 1$ , which corresponds to triangle inequality violations (See Figure 4.2 where we plot the CDF of  $\epsilon$  values for two real-world datasets). We attempt to address both of these phenomena with our two noise models: additive noise to capture the small deviations from 4PC and persistent noise to capture the larger perturbations. In this chapter, we addresses these two types of noise separately, but note that our second algorithm can be modified to handle both types of noise simultaneously.

We are now prepared to formally specify our problem:

**Problem 4.1.** *Given a noisy metric space  $(\mathcal{X}, d)$  equipped with a noisy metric  $d = d_{\mathcal{T}} + g$  for some tree  $\mathcal{T}$ , recover  $\mathcal{T}$  and  $d_{\mathcal{T}}$  while minimizing the number of measurements of  $d$ .*

In this chapter, we develop algorithms for this problem under the assumption that  $g$  corresponds to one of the models above. We first define several quantities that appear in the sequel. For any tree  $\mathcal{T}$ , let  $\text{lvs}(\mathcal{T})$  denote the set of leaf nodes of  $\mathcal{T}$  and let  $\text{deg}(\mathcal{T})$  denote the maximum degree

<sup>1</sup>A random variable  $X$  is **subgaussian** with scale factor  $\sigma^2$  if  $\mathbb{P}(\exp(tX)) \leq \exp(\sigma^2 t^2/2)$ . This family encompasses both gaussian and bounded random variables.



---

**Algorithm 6** PEARLRECONSTRUCT( $\mathcal{X}, d, \gamma$ )

---

Initialize  $T_3$  as a star tree on  $x_1, x_2, x_3$

**for**  $i = 4 \dots p$  **do**

$T_i = \text{PearlAdd}(x_i, T_{i-1}, d, \gamma)$

**end for**

Return  $T_p$

---

of the tree. For convenience we will define  $l \triangleq \text{deg}(\mathcal{T})$ .

For any three nodes  $x, y$ , and  $z$  in a tree, let  $\text{ancestor}(x, y, z)$  be the unique node that is the shared common ancestor of  $x, y$  and  $z$ . This node is the unique point along which the three paths between all pairs of  $x, y$ , and  $z$  intersect and distances to this point can be computed by (where  $a = \text{ancestor}(x, y, z)$ ):

$$d_{\mathcal{T}}(x, a) \triangleq \frac{1}{2}(d_{\mathcal{T}}(x, y) + d_{\mathcal{T}}(x, z) - d_{\mathcal{T}}(y, z)) \quad (4.1)$$

To avoid propagation of additive noise in ancestor computations, we only use distances between true leaf nodes (nodes in  $\mathcal{X}$ ). To compute the ancestor and associated distances between three nodes  $x, y, z$ , some of which may not be leaves, we use a *surrogate* leaf node for each non-leaf in the computation. A surrogate leaf node for  $x$  is one for which  $x$  is on the path between that leaf and both  $y$  and  $z$ . The restriction to minimal trees guarantees existence of surrogate leaf nodes.

## 4.3 Algorithms

We now describe two algorithms for multi-source network tomography and present guarantees on correctness and measurement complexity. Our first algorithm, PEARLRECONSTRUCT addresses the additive noise model while our second, RISING addresses the persistent model.

### 4.3.1 Additive Noise

The idea behind our first algorithm is to construct the tree  $\mathcal{T}$  by iteratively attaching the leaves. To add leaf  $x_i$ , we perform an *intelligent* search to find a pair of nodes  $x_j, x_k$  such that the distance between  $x_i$  and  $\text{ancestor}(x_i, x_j, x_k)$  is minimized. This information, along with the fact that  $x_i$  is not in the same subtree as either  $x_j$  or  $x_k$  (which we also determine), tell us how to add  $x_i$  to the tree.

Our search is intelligent in that we choose  $x_j$  and  $x_k$  to rule out large portions of the tree at every step. Specifically, by choosing a point with fairly balanced subtrees (known as the *pearl point*), we can determine which of these subtrees  $x_i$  belongs to and focus our search to a subtree that is a fraction of the original size, using a constant number of measurements. Formally, for any

---

**Algorithm 7** PEARLADD( $x_i, T_{i-1}, d, \gamma$ )

---

 $T_c = T_{i-1}$ **while**  $|\text{lvs}(T_c)| > 2$  **do**Choose a subtree  $T_{out}$  such that:

$$\frac{|\text{lvs}(T_c)|}{\text{deg}(T_c)+1} < |\text{lvs}(T_c) \setminus \text{lvs}(T_{out})| < \frac{|\text{lvs}(T_c)|\text{deg}(T_c)}{\text{deg}(T_c)+1}.$$

 $r \leftarrow$  parent of  $T_{out}$  in  $T_c$ Let  $T_{sub} \neq T_{out}$  be any other subtree of  $T_c$  rooted at  $r$  and choose  $x_k \in \text{lvs}(T_{sub}), x_j \in \text{lvs}(T_{out})$ . $y \leftarrow$  ancestor( $x_i, x_j, x_k$ ), compute  $d(x_i, y), d(x_j, y)$ , and  $d(x_k, y)$ , using surrogates.If  $d(x_j, y) + \gamma/2 < d(x_j, r)$ , then  $T_c \leftarrow T_{out} \cup \{r\}$ If  $d(x_k, y) + \gamma/2 < d(x_k, r)$ , then  $T_c \leftarrow T_{sub} \cup \{r\}$ Otherwise  $T_c \leftarrow T_c \setminus \{T_{sub} \cup T_{out}\}$ **end while****if**  $|T_c| = 1$  **then**Attach  $x_i$  to  $T_c$  with edge length  $d(x_i, y)$ .**else** $T_c$  has two nodes  $r$  and  $r'$ . Choose leaves  $x_k$  and  $x_j$  such that  $r$  is on the path between  $x_k$  and  $r'$ , and  $r'$  is on the path between  $x_j$  and  $r$ . $y \leftarrow$  ancestor( $x_i, x_k, x_j$ ).If  $|d(x_k, y) - d(x_k, r)| < \gamma/2$ , then attach  $x_i$  to  $r$ .If  $|d(x_j, y) - d(x_j, r')| < \gamma/2$ , then attach  $x_i$  to  $r'$ .Otherwise, insert  $y$  between  $r$  and  $r'$  (with edge weights  $d(x_k, y) - d(x_k, r)$  and  $d(x_j, y) - d(x_j, r')$ ) and attach  $x_i$  to  $y$  with edge weight  $d(x_i, y)$ .**end if**Return  $T_{i-1}$  updated to include  $x_i$ .

---

directed instance of a tree  $\mathcal{T}$ , the pearl point is the internal node in a tree for which the number of leaves below that node is between  $|\text{lvs}(\mathcal{T})|/(\text{deg}(\mathcal{T}) + 1)$  and  $|\text{lvs}(\mathcal{T})|\text{deg}(\mathcal{T})/(\text{deg}(\mathcal{T}) + 1)$ . As we show, using the pearl point results in a strong upper bound on the number of measurements used while ensuring correctness of the algorithm. As the algorithm carefully chooses which pairwise distances to query, our algorithm is *interactive*.

PEARLRECONSTRUCT is related to the algorithm in [137], the Sequential Logical Topology (SLT) algorithm [135], and the Sequoia algorithm [142]. Our search parallels that of [137], but by using triplet tests rather than quartet tests and by incorporating slack into our search, PEARLRECONSTRUCT is robust to additive noise while their algorithm is not. On the other hand, the SLT algorithm is robust to noise, but they do not begin their search at the pearl point of the tree, and thus their measurement complexity guarantees only hold for balanced trees, while our guarantees are more general. The Sequoia algorithm also adopts some of the same ideas, but since their search is heuristic, they do not provide bounds on the number of measurements used.

The algorithm involves a parameter  $\gamma$  that is a lower bound on the edge weights in the true tree  $\mathcal{T}$ . This parameter is critical for identifying two nodes separated by a short edge in the presence of noise and is a robust version of the minimality condition. Similar parameters have been used

in related results [135].

Pseudocode for PEARLRECONSTRUCT is shown in Algorithms 6 and 7. Our main correctness guarantee is the following; proof of the result is deferred to Section 4.5.

**Theorem 4.1.** *Let  $(\mathcal{X}, d)$  be a noisy metric space with  $|\mathcal{X}| = p$  where  $d = d_{\mathcal{T}} + g$  for a tree  $\mathcal{T}$  with minimum edge length  $\geq \gamma$  and consider the additive noise model with scale factor at most  $\sigma^2$ . Fix any  $\delta \in (0, 1)$ . If, for each pairwise distance queried, PEARLRECONSTRUCT uses the average over  $n$  samples and*

$$n > 18 \frac{\sigma^2}{\gamma^2} \log(2p^2/\delta), \quad (4.2)$$

*then with probability at least  $1 - \delta$ , PEARLRECONSTRUCT successfully recovers  $\mathcal{T}$  and the edge weights in the tree with at most  $\gamma/2$  additive error.*

This theorem is a correctness guarantee for PEARLRECONSTRUCT. In the absence of noise, the algorithm always succeeds in recovering the tree topology  $\mathcal{T}$  along with all pairwise distances in the metric  $d_{\mathcal{T}}$ . In the additive noise model, the algorithm fails with some probability  $\delta$ , but with the remaining probability it recovers the tree topology and the edge weights in the tree with error at most  $\gamma/2$ . This implies accurate recovery of all of the pairwise distances in the tree, where the level of accuracy for any distance is linear in the number of edges between the two nodes.

It remains to bound the total number of measurements used by the algorithm. The following theorem upper bounds this quantity.

**Theorem 4.2.** *PEARLRECONSTRUCT uses  $O(pl \frac{\sigma^2}{\gamma^2} \log^2 p)$  pairwise measurements.*

For constant-degree tree metrics, we see that the algorithm uses a slightly super-linear number of measurements. This is a polynomial improvement over a naïve algorithm that would repeatedly query for all pairwise distances and average away noise. This naïve algorithm would use  $\tilde{O}(p^2 \frac{\sigma^2}{\gamma^2})$  measurements, which is quadratic in the network size. By making measurements in an interactive fashion, we obtain a significantly reduced sampling requirement.

Note that this bound also leads to a bound on the running time of the algorithm. For each node we insert, we compute the pearl point and perform quartet tests at most  $O(l \log(p))$  times. Since the pearl point can be computed in linear time, the algorithm runs in  $O(p^2 l \text{polylog}(p))$  time.

### 4.3.2 Persistent Noise

For the persistent noise model, we propose a divisive algorithm; it recursively partitions the leaves into groups corresponding to subtrees of  $\mathcal{T}$ . Each partitioning step identifies one internal node in the tree, and by repeated applications of our algorithm, we identify all internal nodes that satisfy certain properties (detailed in Theorem 4.3).

A top-down partitioning algorithm allows us to use voting schemes that are robust to persistent noise. Specifically, we identify groups of nodes by repeatedly performing quartet or triplet tests

---

**Algorithm 8** RISING( $\mathcal{X}, d, m$ )

---

Randomly choose  $M \subset \mathcal{X}$  with  $|M| = m$   
For  $x_i, x_j \in M$ , compute  $s(x_i, x_j) = \max_{x_k \in M} |\{x_{k'} \in M : d(x_i, x_k) - d(x_j, x_k) = d(x_i, x_{k'}) - d(x_j, x_{k'})\}|$   
Run Single Linkage Clustering using similarity function  $s$  to cluster  $M$  into  $\mathcal{C}$  with  $|\mathcal{C}| = 3$ .  
**for**  $x_i \in \mathcal{X} \setminus M$  **do**  
    VOTE( $x_i, \mathcal{C}, d$ )  
**end for**  
Initialize  $T$  with 1 node  $r$   
**for**  $C \in \mathcal{C}$  **do**  
     $T_{sub} \leftarrow \text{SPLIT}(C, \mathcal{X} \setminus C, d, m)$ .  
    Choose clusters  $C_1, C_2 \in \mathcal{C} \setminus C$   
    weight( $r, \text{root}(T_{sub})$ )  $\leftarrow \text{EDGELENGTH}(C_1, C_2, T_{sub}, d)$   
**end for**  
Return  $T$

---

and deciding on the structure agreed on by the majority. However, to ensure that these groups are sufficiently large, we require a balancedness condition:

**Definition 4.2** (Balance Factor). *We say that  $\mathcal{T}$  has **balance factor**  $\eta$  if  $\eta$  is the smallest number for which there exists a node  $r$  such that for all internal nodes  $h$  (including  $r$ ), with subtrees  $T_1(h), \dots, T_k(h)$  directed away from  $r$ :*

$$\eta \geq \frac{\max_i |lvs(T_i(h))|}{\min_i |lvs(T_i(h))|}.$$

To identify a single internal node  $r$  our algorithm randomly samples a subset of the leaves, forms a clustering of this subset, and then places each remaining leaf into one cluster. After recursively partitioning each cluster, we compute edge lengths using a voting scheme. In the clustering phase, we compute a similarity function  $s$  on the sampled leaves where  $s(x_i, x_j)$  is large if the two leaves belong in the same subtree of  $\mathcal{T}$ , viewed with  $r$  as the root. We partition the sampled nodes into two clusters in most cases (to find the first split we partition into three). Each of these clusters is comprised of leaves from one or more subtrees rooted at  $r$ , but the leaves from any of the subtree are contained wholly in one cluster.

Once we have clustered the sampled nodes, we use voting to determine the group assignments for the remaining nodes. To place a node  $x_i$ , we compute quartet structures (See Figure 4.1) between  $x_i$  and  $x_j, x_k, x_l$  (each from different clusters) and record which node  $x_i$  paired with in the quartet test. We place  $x_i$  into the cluster that most commonly paired with  $x_i$ .

The computations required to find the initial partition of leaves are slightly different from those required for subsequent splits. To highlight these differences, we present pseudocode for recovering the first partition in Algorithm 8 and for subsequent partitions in Algorithm 9. These algorithms rely on two subroutines which we show in Algorithms 10 and 11.

Before presenting our theoretical guarantees, we remark that while our results analyze RISING in

---

**Algorithm 9** SPLIT( $\mathcal{S}, \mathcal{Y}, d, m$ )

---

Randomly choose  $M \subset \mathcal{S}$  with  $|M| = m$   
For each  $x_k \in M$ , draw  $Z(k)$  randomly from  $\mathcal{Y}$ .  
For  $x_i, x_j \in M$ , compute  $s(x_i, x_j) = |\{x_k \in M : d(x_i, x_k) - d(x_j, x_k) = d(x_i, x_{Z(k)}) - d(x_j, x_{Z(k)})\}|$ .  
Run Single Linkage Clustering using similarity function  $s$  to cluster  $M$  into  $\mathcal{C}$  with  $|\mathcal{C}| = 3$ .  
**for**  $x_i \in \mathcal{S} \setminus M$  **do**  
    VOTE( $x_i, \mathcal{C} \cup \{\mathcal{Y}\}, d$ )  
**end for**  
Initialize  $T$  with 1 node  $r$   
**for**  $C \in \mathcal{C}$  **do**  
     $T_{sub} \leftarrow$  SPLIT( $C, \mathcal{Y} \cup (\mathcal{S} \setminus C), d, m$ ).  
    Choose  $C' \in \mathcal{C} \setminus C$   
    weight( $r, \text{root}(T_{sub})$ )  $\leftarrow$  EDGELength( $C', \mathcal{Y}, T_{sub}, d$ )  
**end for**  
Return  $T$

---

---

**Algorithm 10** VOTE( $x, \mathcal{C}, d$ )

---

Let  $C_1, C_2, C_3 \in \mathcal{C}$   
 $VC_1, VC_2, VC_3 \leftarrow 0$   
**for**  $n \in \{1, \dots, \min_{C \in \mathcal{C}} |C|\}$  **do**  
    Choose  $x_1 \in C_1, x_2 \in C_2, x_3 \in C_3$ .  
     $VC_i \leftarrow VC_i + 1$  if  $x$  pairs with  $x_i$  w.r.t. the other two.  
    If  $x_i, x_1, x_2, x_3$  form a star, ignore this vote.  
**end for**  
Place  $x$  in  $C_i$  where  $VC_i = \text{argmax}\{VC_1, VC_2, VC_3\}$

---

the presence of only persistent noise, with slight modifications the algorithm can be made robust to both persistent and additive noise. The main change would involve incorporating slack into the quartet tests, much like we have done in PEARLRECONSTRUCT. The analysis for this modified algorithm would incorporate the techniques used in Theorem 4.1 (specifically concentration of subgaussian random variables) into our current proofs. However, for clarity of presentation, our analysis guarantees the correctness of RISING under only persistent noise.

**Theorem 4.3.** *Let  $(\mathcal{X}, d)$  be a metric where  $d \triangleq d_{\mathcal{T}} + g$  for a tree  $\mathcal{T}$  with bounded balance factor  $\eta$  and where  $g$  is from the persistent noise model with probability of an uncorrupted entry  $\geq q$  with  $q^6 > C_{\eta,l}$ . Then with probability  $\geq 1 - 1/p$ , every execution of RISING and SPLIT, with parameter  $m$ , will correctly identify an internal node provided that:*

$$m > c_{\eta,l} \frac{\log(pm)}{(q^6 - C_{\eta,l})^2} \quad (4.3)$$

where  $1/2 \leq C_{\eta,l} < 1$ ,  $c_{\eta,l}$  are constants depending on  $\eta$  and  $l$ .

This theorem is a correctness guarantee for the RISING algorithm, although the flavor of guaran-

---

**Algorithm 11** EDGELength( $C_1, C_2, T_{sub}, d$ )

---

$C_L \leftarrow$  leaves in one subtree of  $T_{sub}$   
 $C_R \leftarrow$  leaves in another subtree of  $T_{sub}$   
**for**  $n \in \{1, \dots, \min\{m, |C_1|, |C_2|, |C_L|, |C_R|\}\}$  **do**  
    Draw  $w \in C_1, x \in C_2, y \in C_L, z \in C_R$   
    Record  $\frac{1}{2}d(w, y) + d(x, z) - d(w, x) - d(y, z)$   
**end for**  
Return the most frequently occurring recorded value

---

tee is quite different from that in Theorem 4.1. This theorem ensures that any internal node for which every subtree has size at least  $m$  will be recovered by repeated calls to Algorithm 9. In the absence of noise, we can choose  $m$  to be a function of  $|\mathcal{S}|$ , the subset of leaves passed into the SPLIT routine. However, with noise,  $m$  must be  $\Omega(\log p)$  and if  $\mathcal{S}$  is too small for this, then  $\mathcal{S}$  cannot be further resolved, and thus  $\log p$  limits the recovery resolution.

In Section 4.5, we give a precise characterization of  $C_{\eta,l}$ , which plays a critical role in RISING's robustness to noise. While  $C_{\eta,l} < 1$  for all values of  $\eta$  and  $l$ , it grows with these quantities. Specifically, the minimum value for  $C_{\eta,l}$  is  $1/2$ , which happens when  $\eta = 1$  and  $l = 2$ . This corresponds to a perfectly balanced binary tree, which is the easiest case for the persistent noise setting.

We now upper bound the number of measurements used by the algorithm:

**Theorem 4.4.** *On trees with bounded balance factor, RISING uses  $O(pml \log p)$  measurements where  $l$  is the maximum degree of the tree  $\mathcal{T}$ .*

Setting  $m$  as in Theorem 4.3, we see that RISING recovers all identifiable internal nodes while using  $O(pq^{-6} \log^2(p))$  measurements. Comparing with a naïve, non-interactive algorithm that obtains all measurements, this is a polynomial improvement in sample complexity, demonstrating the power of interactivity for this problem. We are not aware of any more sophisticated non-interactive approaches for this setting.

## 4.4 Experiments

We perform several experiments on simulated and real-world topologies to assess the validity of our theoretical results and to demonstrate the performance of our algorithms. We study how increasing noise affects our algorithms ability to correctly recover the topology and also how the number of measurements used compares to related algorithms.

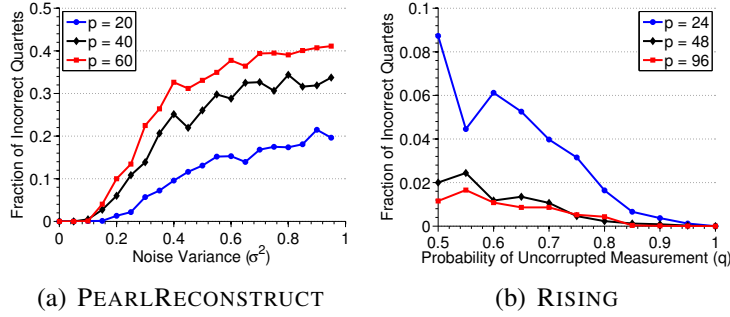


Figure 4.3: Noise Thresholds for PEARLRECONSTRUCT and RISING.

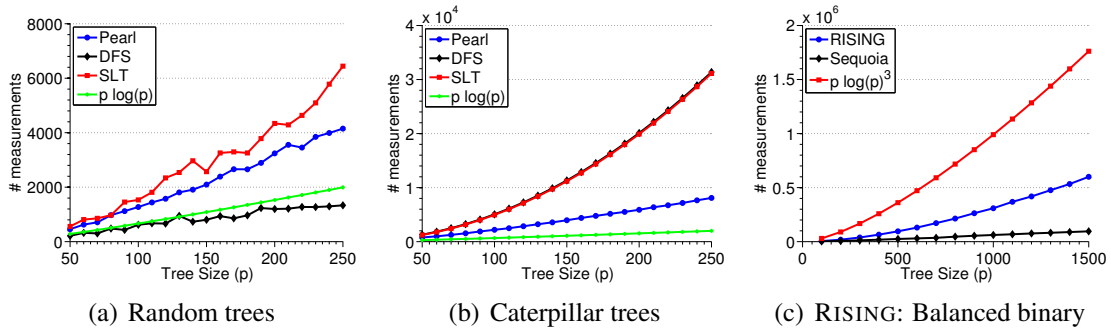


Figure 4.4: Measurements used as a function of  $p$  for PEARLRECONSTRUCT, RISING, DFS Ordering [86], SLT [135], and Sequoia [142]

## 4.4.1 Simulations

In simulations, we demonstrate how our algorithms tolerate noise, how this tolerance scales with  $p$ , and how the number of measurements used scales with  $p$ . For these experiments, we generate tree topologies and obtain pairwise distances by computing unweighted path lengths along the tree to represent hop counts in a network. We then perturb this pairwise distance matrix with additive or persistent noise and run our algorithms on this perturbed matrix. We assess the correctness of our algorithms by computing the fraction of quartets for which the structure in the reference tree matches that in the algorithm’s output.

For RISING, in simulations we always choose  $m = \log^2 |\mathcal{S}|$  (even with noise), which as mentioned, satisfies the conditions of Theorem 4.3 in the absence of noise. For our real world experiments, we use  $m = \log p$ .

Our first experiment studies how PEARLRECONSTRUCT and RISING perform in the presence of noise. In Figures 4.3(a) and 4.3(b) we plot the fraction of incorrect quartets averaged over 20 trials for PEARLRECONSTRUCT and RISING respectively, as a function of the noise for different values of  $p$ . In Figure 4.3(a) we verify three properties of PEARLRECONSTRUCT: (i) in the absence of noise, it deterministically recovers the true topology as predicted by Lemma 4.5,

(ii) as the noise variance increases, PEARLRECONSTRUCT becomes less accurate, (iii) on larger topologies, PEARLRECONSTRUCT requires lower noise variance. This last property follows from Equation 4.2 since if  $n$  is constant (we took  $n = 1$  for these experiments), we require  $\sigma^2 = O(\frac{1}{\log p})$  in order to guarantee successful recovery, and this upper bound decreases with  $p$ .

For RISING, in Figure 4.3(b), we observe the opposite phenomenon; larger topologies can tolerate more persistent noise. This matches our bounds in Theorem 4.3, which allows  $q$  to approach a constant as  $m, p \rightarrow \infty$ . As before, we also observe that in the absence of noise, we deterministically recover the underlying topology, although we note that we used balanced binary trees for these experiments. For highly unbalanced trees, we cannot make this deterministic guarantee.

To assess the measurement complexity of our algorithms, we record how many measurements each algorithm uses as a function of  $p$ , in the absence of noise. These plots are shown in Figure 4.4. As is noticeable in Figure 4.4(a), the measurement complexity for PEARLRECONSTRUCT appears to be  $O(p \log p)$ . We also show the measurement complexity for the DFS Ordering algorithm of Eriksson et al [86] and the Sequential Logical Topology (SLT) algorithm [135], both of which are single-source tomography methods with provable  $O(p \log p)$  complexity on balanced trees. The trees used here are randomly generated, and we see that the SLT algorithm performs worse than PEARLRECONSTRUCT, while DFS Ordering seems to use a constant multiplicative factor fewer measurements.

However, in the worst case, PEARLRECONSTRUCT enjoys considerable advantage over both SLT and DFS Ordering as can be seen in Figure 4.4(b). In this experiment, we used highly unbalanced trees and we see that the measurement complexity of both SLT and DFS Ordering scale at  $O(p^2)$ , while PEARLRECONSTRUCT continues to scale at  $O(p \log p)$ .

In Figure 4.4(c), we compare RISING to the Sequoia algorithm of [142]. While Sequoia comes with no guarantees about correctness or measurement complexity, it appears to use very few measurements in practice. RISING on the other hand appears to use a multiplicative factor of  $\log p$  more measurements than Sequoia, which we confirmed empirically. However, as we show in our real world experiments, Sequoia is less robust to noise, even when customized to use a similar number of measurements as RISING. We also emphasize that RISING comes with guarantees on correctness in the presence of noise while Sequoia does not.

## 4.4.2 Real World Experiments

In addition to verifying our theoretical results, we are interested in assessing the practical performance of our algorithms on real network tomography datasets. We use two datasets: the King dataset [98] of pairwise latencies and a dataset of hop counts between PlanetLab [139] hosts measured using iPlane [129]. We selected a 500-node subset of the 1740-node King dataset. The iPlane dataset consists of 193 end hosts.

We ran three algorithms, PEARLRECONSTRUCT, RISING, and Sequoia, on both datasets and plot the distribution of *relative error* values for each algorithm. Given the constructed tree met-



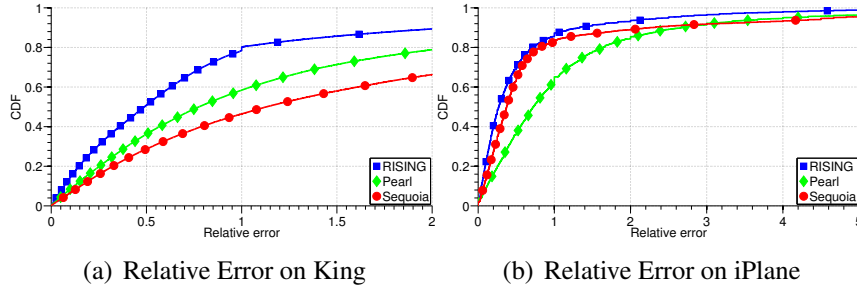


Figure 4.5: CDF of relative error on King (a) and iPlane (b) datasets.

Dataset	Hosts	Total	Pearl	RISING	Sequoia
King	500	125250	8321	43608	42599
iPlane	194	18721	2480	12309	11574

Figure 4.6: Measurements used on real world data sets

ric  $(X, \hat{d})$  and the true metric  $(X, d)$ , we measure relative error for each pairwise distance as  $\frac{|\hat{d}(x_i, x_j) - d(x_i, x_j)|}{d(x_i, x_j)}$ . This quantity reflects how well the tree metric approximates the true distances in the network. These plots are shown in Figures 4.5(a) and 4.5(b). We see that on both datasets, RISING outperforms both Sequoia and PEARLRECONSTRUCT, with substantial improvements on the King dataset. PEARLRECONSTRUCT performs moderately well on both datasets.

Lastly, we recorded the number of measurements used by the algorithms on the two datasets in Figure 4.6. Note that Sequoia can be used to build many trees where the recovered pairwise distances is the median distance across all trees. To ensure a fair comparison, we build several trees so that Sequoia and RISING use a similar number of measurements. However, even with several trees, RISING performs better than Sequoia.

## 4.5 Proofs

### 4.5.1 Proof of Theorem 4.1

First, we consider the noiseless scenario.

**Lemma 4.5.** *Let  $(X, d)$  be a minimal tree metric on  $\mathcal{T}$  with  $|X| = p$ . Then PEARLRECONSTRUCT on input  $(X, d)$  recovers  $\mathcal{T}$  and  $d$  exactly.*

*Proof.* We start with  $T_3$ , the tree on leaves  $x_1, x_2$  and  $x_3$ . Every minimal tree on 3 leaves has the same structure as  $T_3$ , so we know this is correct. Moreover, since  $d(x_i, y)$  for  $i \in \{1, 2, 3\}$  and  $y = \text{ancestor}(x_1, x_2, x_3)$  is given by Equation 4.1, the edge weights in  $T_3$  are also correct.

We now analyze the `add` procedure, showing that it correctly places  $x_i$  into the tree so that  $T_i$  is the correct minimal tree on  $x_1, \dots, x_i$  with the correct edge weights. We proceed by case analysis: for any root  $r$  with subtrees  $T_{out}$  and  $T_{sub}$ , it must be that either  $x_i$  belongs in  $T_{out}$ ,  $T_{sub}$  or in  $T_c \setminus \{T_{sub} \cup T_{out}\}$ . For any  $x_k \in T_{sub}, x_j \in T_{out}$ , if  $x_i$  belongs in  $T_{out}$ , then it must be the case that  $d(x_j, y) < d(x_j, r)$  or else the shared common ancestor between  $x_i, x_j$ , and  $x_k$  could not possibly lie in  $T_{out}$ . Similarly, if  $x_i$  belongs in  $T_{sub}$  then it must be that  $d(x_k, y) < d(x_j, r)$ . Finally, if  $x_i$  lies in neither subtree, then  $\text{ancestor}(x_i, x_j, x_k) = r$ .

In each case, we update  $T_c$  so that it still contains the location where  $x_i$  should be added. Since we choose  $T_{sub}$  and  $T_{out}$  to be non-empty subtrees, the size of  $T_c$  decreases on every iteration, so the algorithm must eventually exit the loop.

When this happens,  $|T_c| \leq 2$  and  $T_c$  contains the location of  $x_i$ . If  $|T_c| = 1$ , then the only place to add  $x_i$  is as a child of the node in  $T_c$ . This only happens if  $\text{ancestor}(x_i, x_j, x_k) = r$  in the last iteration of the while loop, so the distance  $d(x_i, y)$  is the correct edge weight for the new edge.

If  $|T_c| = 2$ , then we use two additional leaves to determine how to place  $x_i$ . Case analysis reveals that our procedure correctly places  $x_i$  into  $T_c$ . Thus, we conclude that the `add` procedure correctly update  $T_{i-1}$  to contain  $x_i$ . By iteratively applying this argument, we arrive at the claim.



Turning to the noisy setting, we can no longer deterministically guarantee correct recovery of  $\mathcal{T}$ , but instead require a probabilistic analysis. In the algorithm, we choose three nodes  $x_i, x_j$  and  $x_k$  and compute distances between these nodes and  $y \triangleq \text{ancestor}(x_i, x_j, x_k)$ . We need to be able to correctly determine if  $y$  lies between the root  $r$  and  $x_j$ , between  $r$  and  $x_k$ , or elsewhere in the tree. We therefore seek to bound  $|\hat{d}(x_k, y) - d(x_k, y)|$  and  $|\hat{d}(x_j, y) - d(x_j, y)|$  where  $\hat{d}$  corresponds to our empirical estimate of the distance based on  $n$  samples.


To arrive at these bounds, we first derive concentration inequalities for the directly observed measurements. Specifically, by application of the Subgaussian tail bound and the union bound we have that with probability  $\geq 1 - \delta$ :

$$|\hat{d}(x_i, x_j) - d(x_i, x_j)| \leq \sqrt{\frac{2\sigma^2 \log(2p^2/\delta)}{n}},$$

for all leaves  $x_i, x_j, i, j \in [p]$ . Using this bound along with Equation 4.1, immediately reveals that the distance in the estimated tree between any two nodes deviates from the correct distance by at most  $\frac{3}{2} \sqrt{\frac{2\sigma^2 \log(2p^2/\delta)}{n}}$ .

In order for the algorithm to work, we need to ensure that we can identify when the ancestor node  $y$  equals the root node  $r$ , in spite of the deviations. If:

$$\gamma > 3 \sqrt{\frac{2\sigma^2 \log(2p^2/\delta)}{n}}, \tag{4.4}$$

then with high probability we will not confuse the nodes  $y$  and  $r$ , since distances to each node only deviate by half that. Inverting Equation 4.4 yields the bound on  $n$  in the theorem. 

### 4.5.2 Proof of Theorem 4.2

We study the add procedure. By Lemma 1 in [137], we know that for any  $T_c$  there exists a subtree  $T_{out}$  for which:


$$\frac{|\text{ivs}(T_c)|}{\deg(T_c) + 1} < |\text{ivs}(T_c) \setminus \text{ivs}(T_{out})| < \frac{|\text{ivs}(T_c)| \deg(T_c)}{\deg(T_c) + 1}$$

Let  $l_c = \deg(T_c)$ . The fact that  $|\text{ivs}(T_c) \setminus \text{ivs}(T_{out})| < \frac{|\text{ivs}(T_c)| l_c}{l_c + 1}$  means that  $|\text{ivs}(T_{out})| \geq \frac{|\text{ivs}(T_c)|}{l_c + 1}$ . Writing  $T_c^i$  to denote  $T_c$  after  $i$  iterations of the loop, we see that no matter how the search proceeds,  $|\text{ivs}(T_c^i)| \leq \frac{l_c}{l_c + 1} |\text{ivs}(T_c^{i-1})|$ .

Thus the number of iterations required to place  $x_i$  in  $T_{i-1}$  is at most  $\log_{\frac{l_c+1}{l_c}}(i-1) \leq 2l_c \log(i-1)$ . This follows since:

$$\log_{\frac{l_c+1}{l_c}}(i-1) = \frac{\log(i-1)}{\log\left(1 + \frac{1}{l_c}\right)} \leq \frac{2l_c^2}{2l_c - 1} \log(i-1) \leq 2l_c \log(i-1)$$

The first inequality is based on the Taylor expansion  $\log(1 + 1/x) \geq \frac{1}{x} - \frac{1}{2x^2}$  and the second one holds provided that  $l_c \geq 1$ , which is always true here. Since each loop iteration uses a constant number of pairwise distance measurements,  $l_c$  is upper bounded by  $l$  the maximum degree of  $\mathcal{T}$ , and we call add at most  $p$  times, we see that the measurement complexity is  $O(pl \log p)$  in the absence of noise.

Finally, recall from Theorem 4.1 that if  $n$  is  $O(\log p)$  we can guarantee exact recovery of the tree. We must therefore observe each measurement  $O(\log p)$  times and including this multiplicative factor results in the stated bound. 

### 4.5.3 Proof of Theorem 4.3

We first state and prove several lemmas, and then turn to the task of recovering the splits.

**Lemma 4.6 (Sampling).** *Let  $\mathcal{T}$  have balance factor  $\eta$  and maximum degree  $k$ . Then in all iterations of RISING and SPLIT, with probability  $\geq 1 - \frac{2}{pk}$ , the sampled subtree of  $\mathcal{T}$  with leaf set  $M$  has balance factor:*

$$\hat{\eta} \leq 2\eta + 1,$$

as long as  $m \geq 4(1 + (k-1)\eta)^2 \log(pk)$ .

*Proof.* In this proof we will simultaneously work with all of the recursive calls of RISING. Since each call recovers one internal node, and there can be no more than  $p$  internal nodes in  $\mathcal{T}$ , we can enumerate the calls from 1 to  $p$ . Each call operates on a subset of leaf nodes and we will refer to the tree induced by those leaves as  $T^s$  for the  $s$ th call.

For fixed  $s$ , define the random variables  $Z_{ij}, i \in [m], j \in [k]^2$ , which takes value 1 if the  $i$ th leaf sampled belongs in  $T_j^s$ , the  $j$ th subtree of  $r$  (the root of  $T^s$ ). Further define  $\hat{T}_j^s$  to be the sampled version of  $T_j^s$ , that is the tree  $T_j^s$  restricted to only the leaves in  $M$ . Notice that  $\mathbb{E}[Z_{ij}] = \frac{|\text{lvs}(T_j^s)|}{|\text{lvs}(T^s)|}$  and that  $|\text{lvs}(\hat{T}_j^s)| = \sum_{i=1}^m Z_{ij}$ . By Hoeffding's inequality we have that:

$$\mathbb{P}\left(\left|\frac{1}{m}|\text{lvs}(\hat{T}_j^s)| - \frac{|\text{lvs}(T_j^s)|}{|\text{lvs}(T^s)|}\right| > \epsilon\right) \leq 2 \exp\{-2m\epsilon^2\},$$

for any single  $j \in [k], s \in [p]$ . We would like to do this across all calls to SPLIT, and for each subtree in any of the calls. We take a union bound across all internal nodes and all subtrees, and then rewrite to introduce dependence on the balance factor  $\eta$ , noting that  $|\text{lvs}(T_{(k)}^s)| \leq \eta |\text{lvs}(T_{(1)}^s)|$  for any internal node<sup>3</sup>. This gives us that:

$$\begin{aligned} \frac{1}{m}|\text{lvs}(\hat{T}_{(1)}^s)| &\geq \frac{|\text{lvs}(T_{(1)}^s)|}{|\text{lvs}(T^s)|} - \sqrt{\frac{\log(2pk/\delta_1)}{2m}} \\ \frac{1}{m}|\text{lvs}(\hat{T}_{(k)}^s)| &\leq \frac{\eta |\text{lvs}(T_{(1)}^s)|}{|\text{lvs}(T^s)|} + \sqrt{\frac{\log(2pk/\delta_1)}{2m}} \end{aligned}$$

Note that since we have established concentration inequalities for all subtrees, the new balance factor  $\hat{\eta}$  depends only on the lower bound for the smallest subtree size and the upper bound for the largest subtree size. Now let  $m = c \log(pk)$  and set  $\delta_1 = \frac{2}{pk}$ . With these settings we have:

$$\frac{1}{m}|\text{lvs}(\hat{T}_{(1)}^s)| \geq \frac{|\text{lvs}(T_{(1)}^s)|}{|\text{lvs}(T^s)|} - \sqrt{\frac{1}{c}} \quad \text{and} \quad \frac{1}{m}|\text{lvs}(\hat{T}_{(k)}^s)| \leq \frac{\eta |\text{lvs}(T_{(1)}^s)|}{|\text{lvs}(T^s)|} + \sqrt{\frac{1}{c}}$$

The new balance factor is the ratio of these two quantities. To find the worst case  $\hat{\eta}$ , we need to maximize with respect to  $|\text{lvs}(T_{(1)}^s)|$ . It is easy to verify that the maximum is achieved at the smallest possible size for  $T_{(1)}^s$ , and given a balance factor of  $\eta$ , we have that  $|\text{lvs}(T_{(1)}^s)| \geq \frac{|\text{lvs}(T^s)|}{1+(k-1)\eta}$ , achieved when the remaining subtrees are all of the same size. Plugging in this value for  $|\text{lvs}(T_{(1)}^s)|$  we have:

$$\hat{\eta} \leq \frac{\frac{\eta}{1+(k-1)\eta} + \sqrt{\frac{1}{c}}}{\frac{1}{1+(k-1)\eta} - \sqrt{\frac{1}{c}}}$$

<sup>2</sup>we use  $[m]$  to denote  $\{1, \dots, m\}$

<sup>3</sup>we use  $T_{(1)}^s, \dots, T_{(k)}^s$  to denote the subtrees of  $T^s$  in increasing sorted order by number of leaves

Now as long as  $c \geq (1 + (k - 1)\eta)^2$ , this quantity is guaranteed to be positive and if  $c = 4(1 + (k - 1)\eta)^2$ , then some algebra shows that:

$$\hat{\eta} \leq 2\eta + 1$$



**Lemma 4.7** (Clustering). *Suppose that the probability of an uncorrupted entry  $q^4 > C_{\hat{\eta},k}$  and:*

$$m > c_{\hat{\eta},k} \frac{\log(m^2/\delta_2)}{(q^4 - C_{\hat{\eta},k})} \quad (4.5)$$

for constants  $C_{\hat{\eta},k} < 1, c_{\hat{\eta},k}$  that depend on  $\hat{\eta}$  and the max degree  $k$ . Then with probability  $\geq 1 - \delta_2$ , Single Linkage clustering on  $M$  using  $s(x_i, x_j)$  as the similarity between  $x_i$  and  $x_j$  partitions  $M$  such that either each subtree is entirely contained in one cluster  $C \in \mathcal{C}$ , or if a subtree is split across clusters, those clusters contain no nodes from other subtrees.

**Remark 4.1.** *While we have suppressed dependence on  $\hat{\eta}$  in Lemma 4.7, we note that a critical condition for correctness is that  $\hat{\eta} = O(1)$ . This condition ensures that single linkage clustering completely groups any individual subtrees of  $\mathcal{T}$  before merging it with any other subtree and is required for our algorithms to be robust to noise.*

*Proof.* The proofs for RISING and SPLIT are almost identical. We tailor our proof to the former, noting where modifications need to be made for the latter.

Our strategy is to lower bound the quantity  $s(x_i, x_j)$  for any pair of leaves  $x_i, x_j$  that belong to the same subtree and to upper bound  $s(x_i, x_k)$  if  $x_i$  and  $x_k$  do not belong to the same subtree. Under the conditions on  $q$ , we show that this lower bound exceeds the upper bound and this guarantees that one subtree will be fully contained in any cluster before any two subtrees are merged. This means that either a subtree is fully contained in a cluster or if it is split across clusters, no nodes from other subtrees are in these clusters.

To assist in our analysis we use the following notation. As above, we write  $\hat{T}_i$  to be the  $i$ th subtree of  $r$  the root node in the definition of balance factor, restricted to the leaves in  $M$ . Let  $s^*(x_i, x_j)$  be the value of  $s(x_i, x_j)$  with this subsampling but in the absence of any noise in the distances. Let  $G_{ij}$  be the group of nodes  $x_k$  that all have the same  $d(x_i, x_k) - d(x_j, x_k)$  value and that achieve the maximum in the computation of  $s(x_i, x_j)$ . In particular, this means  $s^*(x_i, x_j) = |G_{ij}|$ . Define  $\hat{T}_{(1)}, \dots, \hat{T}_{(k)}$  to be the subtrees of the subsampling ordered by increasing number of leaves. Finally define  $\kappa_{min}^T \triangleq \sum_{i=1}^{k-1} |\text{lvs}(\hat{T}_{(i)})|$  and  $\kappa_{min}^A \triangleq \sum_{i=k-1}^k |\text{lvs}(\hat{T}_{(i)})|$ .  $\kappa_{min}^T$  is a lower bound on  $s^*(x_i, x_j)$  for  $x_i, x_j$  in the same subtree and  $\kappa_{min}^A$  is an upper bound on  $m - s^*(x_i, x_k)$  for  $x_i, x_k$  in different subtrees.

We now lower bound  $s(x_i, x_j)$  for  $x_i, x_j$  in the same subtree. In the presence of noise, any node  $x_t \in G_{ij}$  remains in  $G_{ij}$  as long as  $d(x_i, x_t)$  and  $d(x_j, x_t)$  are not corrupted, which occurs with probability at least  $q^2$ . Thus:

$$\mathbb{E}[s(x_i, x_j)] \geq q^2 s^*(x_i, x_j)$$

Since each  $x_t$  contributes to  $s(x_i, x_j)$  independently and since there are  $|G_{ij}|$  nodes  $x_t$ , we can use Hoeffding's Inequality, coupled with a union bound, to show that with probability  $\geq 1 - \delta_{c1}$ :

$$s(x_i, x_j) \geq q^2 s^*(x_i, x_j) - m \sqrt{\frac{\log(m^2/\delta_{c1})}{2\kappa_{min}^T}}, \quad (4.6)$$

for all pairs  $i, j$  that belong in the same subtree. This is our lower bound.

For SPLIT, we analogously define  $G_{ij} \triangleq \{k : d(x_i, x_k) - d(x_j, x_k) = d(x_i, x_{Z(k)}) - d(x_j, x_{Z(k)})\}$  and we require that four measurements are uncorrupted. The above argument, tailored to this scenario gives (with probability  $\geq 1 - \delta_{c1}$ ):

$$\mathbb{E}[s(x_i, x_j)] \geq q^4 s^*(x_i, x_j) \quad \text{and} \quad s(x_i, x_j) \geq q^4 s^*(x_i, x_j) - m \sqrt{\frac{\log(m^2/\delta_{c1})}{2\kappa_{min}^T}}$$

For the upper bound, we can see that a node can contribute to  $s(x_i, x_k)$  if it contributes to  $s^*(x_i, x_k)$  and it uses no corrupted measurements or if it does not contribute to  $s^*(x_i, x_k)$  and it contains a corrupted measurement. For the first case, we will assume pessimistically that all of the nodes  $x_t \in G_{ik}$  contribute to  $s(x_i, x_k)$ . For the latter, we again perform a worst case analysis where we assume any  $x_t \notin G_{ij}$  for which either  $d(x_i, x_t)$  or  $d(x_k, x_t)$  are corrupted contributes to  $s(x_i, x_k)$ . Thus any  $x_t$  contributes with probability  $1 - q^2$ . If we write  $s_2(x_i, x_k)$  to denote the number of nodes  $x_t \notin G_{ij}$  that could contribute to  $s(x_i, x_k)$ , then by the same techniques as above, we arrive at the following upper bound:

$$\begin{aligned} \mathbb{E}[s_2(x_i, x_k)] &\leq (1 - q^2)(m - s^*(x_i, x_k)) \\ s_2(x_i, x_k) &\leq (1 - q^2)(m - s^*(x_i, x_k)) + m \sqrt{\frac{\log(m^2/\delta_{c2})}{2\kappa_{min}^A}} \end{aligned}$$

Where the second statement holds with probability  $\geq 1 - \delta_{c2}$ .

In order to ensure success of our clustering algorithm, we need the lower bound for  $s(x_i, x_k)$  to be larger than the upper bound for  $s(x_i, x_k)$ .

Setting  $\delta_2 \triangleq \delta_{c1} = \delta_{c2}$ , we can now bound  $q$  as:

$$\begin{aligned} q^2 &\geq \frac{m}{m + s^*(x_i, x_j) - s^*(x_i, x_k)} + \sqrt{\frac{1}{2} \log(m^2/\delta_2)} \\ &\times \left[ \frac{s^*(x_i, x_j) \sqrt{1/\kappa_{min}^T} + (m - s^*(x_i, x_k)) \sqrt{1/\kappa_{min}^A}}{m + s^*(x_i, x_j) - s^*(x_i, x_k)} \right] \end{aligned}$$

For this inequality to hold, we require that  $s^*(x_i, x_j) \geq s^*(x_i, x_k)$ , but this is always the case since  $s^*(x_i, x_j) - s^*(x_i, x_k) \geq |\text{lvs}(\hat{T}_{(1)})|$ , i.e. the size of the smallest subtree.

To better illustrate the dependence between  $q$  and the various parameters of the problem, we simplify the expression using the following bounds (which are straightforward to verify):

$$\kappa_{min}^T \geq \frac{m\hat{\eta}}{1 + (k-1)\hat{\eta}}, \quad \kappa_{min}^A \geq \frac{2m\hat{\eta}}{1 + (k-1)\hat{\eta}}, \quad |\text{Ivs}(\hat{T}_{(1)})| \geq \frac{m}{1 + (k-1)\hat{\eta}}$$

Using this bounds we arrive at the following lower bound on  $q^2$ :


$$q^2 \geq \frac{1 + \frac{1+\sqrt{2}}{2} \sqrt{\frac{\log(m^2/\delta_2)(1+(k-1)\hat{\eta})}{m\hat{\eta}}}}{1 + \frac{1}{1+(k-1)\hat{\eta}}}$$

Specifically, this means that the constant  $C_{\hat{\eta},k}$  and  $c_{\hat{\eta},k}$  in the lemma are:

$$C_{\hat{\eta},k} = \frac{1 + (k-1)\hat{\eta}}{2 + (k-1)\hat{\eta}} \quad \text{and} \quad c_{\hat{\eta},k} = \frac{(1 + \sqrt{2})(1 + (k-1)\hat{\eta})^{3/2}}{2\sqrt{\hat{\eta}}(2 + (k-1)\hat{\eta})}$$

Plugging in these constants and reorganizing the expression results in Equation 4.5. Both constants depend on both  $\hat{\eta}$  and  $k$ , however notice that  $C_{\hat{\eta},k} < 1$  and both  $C_{\hat{\eta},k}$  and  $c_{\hat{\eta},k}$  are smaller for  $\hat{\eta}$  close to 1. Thus we see that it is easier to cluster more balanced trees.

The analysis for SPLIT is the same, except that we require  $q^4$  to be greater than the right hand side of above lower bound on  $q^2$ . Since this dependence is worse than the one for RISING, we

use this expression in our result. 

**Lemma 4.8 (Voting).** *Suppose that  $q^6 > C_{\hat{\eta},k}$ . Then with probability  $\geq 1 - \delta_3$  the voting phase of RISING and SPLIT correctly partition the leaves into their subtrees as long as:*

$$m > c_{\hat{\eta},k} \frac{\log(p/\delta_3)}{(q^6 - C_{\hat{\eta},k})^2}, \quad (4.7)$$

for some constants  $c_{\hat{\eta},k}, C_{\hat{\eta},k}$  that depends on  $\hat{\eta}$  and  $k$ .

*Proof.* The voting procedure works by taking one node from each cluster in  $\mathcal{C}$  and computing the quartet between those three nodes and the node we are trying to place,  $x_i$ . Suppose that  $x_i$  belongs in cluster  $C^*$ ; then it must be the case that  $C^* \in \mathcal{C}$  or there exists some  $C' \in \mathcal{C}$  such that  $C^* \subset C'$ . This latter case can happen if we merge two subtrees in the clustering phase.

Since  $\mathcal{C}$  always has cardinality 3 in Algorithm 10, when we draw one node from each of the three clusters one of two things can happen. If we draw a node from  $C^*$  then in the absence of noise, this quartet would correctly vote that  $x_i$  belongs in the cluster  $C'$ . If on the other hand, we draw a node from  $C' \setminus C^*$ , then in the absence of noise this quartet would vote that  $x_i$  forms a star. Our analysis must consider both of these scenarios.

Specifically, let  $Z_i$  be the indicator that the  $i$ th quartet test correctly voted that  $x_i$  belongs in  $C'$ . We perform  $z \triangleq |\hat{T}_{(1)}|$  rounds of voting and by application of a Hoeffding's Inequality and a union bound:

$$\mathbb{P} \left( \frac{|C^*|}{|C'|} q^6 - \frac{1}{z} \sum_{i=1}^z Z_i > \epsilon \right) \leq \exp\left\{-\frac{1}{c} m \epsilon^2\right\}$$

$$Z \triangleq \frac{1}{z} \sum_{i=1}^z Z_i > \frac{|C^*|}{|C'|} q^6 - \sqrt{\frac{c \log(p/\delta_3)}{m}},$$

for each  $x_i \in \mathcal{X} \setminus M$  and for some constant  $c$  that depends only on  $\hat{\eta}$  and  $k$  ( $c = 1 + (k-1)\hat{\eta} \geq 1/|\text{Lvs}(\hat{T}_{(1)})|$ ). We see that with probability  $\delta_3$ , the fraction of correct votes is bounded from below as long as  $m = \omega(\sqrt{\log p})$  so that the second expression  $\rightarrow 0$  as  $p \rightarrow \infty$ .

We will need a similar concentration bound on the number of votes that form a star. Define  $W_i$  to be the indicator that the  $i$ th quartet test correctly forms a star. By a similar argument we see that with probability  $\geq 1 - \delta_3$ :

$$W \triangleq \frac{1}{z} \sum_{i=1}^z W_i \geq \frac{|C'| - |C^*|}{|C'|} q^6 - \sqrt{\frac{c \log(p/\delta_3)}{m}}$$

for all  $x_i \in \mathcal{X} \setminus M$ .

To guarantee that we place  $x_i$  correctly, we will pessimistically assume that every vote not for  $C'$  and not for a star will vote for the same  $C \in \mathcal{C}$ ,  $C \neq C'$ . Thus the fraction of votes for  $C$  is  $1 - Z - W$  and we require that  $Z > 1 - Z - W$ . Some algebra shows that this is true if:

$$q^6 > \frac{|C'|}{|C'| + |C^*|} + 3\sqrt{\frac{c \log(p/\delta_3)}{m}}$$

Inverting this equation gives us the lower bound on  $m$  in the Lemma. The constant  $C_{\hat{\eta},k}$  is exactly

$$\frac{|C'|}{|C'| + |C^*|} \leq \frac{1+(k-1)\hat{\eta}}{2+(k-1)\hat{\eta}} \text{ which is the same as the constant in Lemma 4.7.}$$



## Recovering One Split

Each time we call RISING or SPLIT we attempt to recover one internal node of the tree. In terms of dependence on  $m$ , we showed above that as long as  $m$  is sufficiently large, the sampling phase will result in a new balance factor  $\hat{\eta}$  that is not too different from the original balance factor  $\eta$  and that Single Linkage will produce clusters that reflect the subtrees. Combining the bounds on  $m$  from all three phases, we have the following lower bound on  $m$ :

$$m > c_{\hat{\eta},k} \frac{\log(pm^2/\delta)}{(q^6 - C_{\hat{\eta},k})^2}$$



And the restrictions on the probability of an uncorrupted entry arise from the clustering and voting phases, but the voting phase's condition is more stringent. We therefore need  $q^6 > C_{\hat{\eta},k}$

Finally, we require that the balance factor of the tree  $\eta = O(1)$  so that  $\hat{\eta}$  will also be a constant for large enough  $m$  with high probability.

Putting these conditions together, we can characterize the dependence on  $m$  and  $p$  under which successful recovery of a single split is possible. Specifically, we have that if  $m = \Omega(\log(p/\delta))$ , then with probability  $\geq 1 - \delta$  (where  $\delta \triangleq \delta_1 + \delta_2 + \delta_3$ ), we correctly recover one internal node.

## Recovering All Splits

There are at most  $p$  internal nodes in the tree. To recover all of these with probability  $1 - o(1)$ , we set each  $\delta_i = O(1/p)$ , and again characterize the dependence between  $m$  and  $p$ . In the sampling phase, we require that  $m = \omega(\log p)$  to ensure that  $\hat{\eta}$  does not grow with  $p$ . In clustering, we similarly require  $m = \omega(\log(m^2 p))$ . Finally, in the voting phase, we see that  $m = \omega(\log(p))$ .

These bounds determine conditions for successful recovery of the entire tree.



## 4.5.4 Proof of Theorem 4.4

We will analyze each level of the tree. Since  $\eta$  is bounded, there are  $O(\log p)$  levels of the tree.

At each level, let  $\mathcal{C}$  be the set of all groups we are trying to split at this level, that is each  $C \in \mathcal{C}$  is the set of nodes passed in as the first parameter to SPLIT, or in the case of the first call,  $\mathcal{C}$  just contains one set with all of the nodes. For each group  $C \in \mathcal{C}$  let  $p_C$  denote the number of nodes in  $C$  and let  $m_C$  denote the value of the parameter  $m$  which can be a function of  $|C|$ <sup>4</sup>.

For each cluster  $C$ , we require  $m_C(m_C + 1)/2$  measurements between sampled nodes and, in SPLIT, an additional  $m_C$  measurements from the set  $\mathcal{Y}$ . In the voting phase, we vote on  $p_C - m_C$  nodes and for each node we require  $m_C + 1$  measurements to the sampled nodes and to one node in  $\mathcal{Y}$ . Putting this together, we have that at any level, we use:

$$\sum_{C \in \mathcal{C}} \frac{m_C(m_C + 1)}{2} + m_C + (p_C - m_C)(m_C + 1) \leq \sum_{C \in \mathcal{C}} p_C(m_C + 1) \leq p(m + 1),$$

as long as  $m_C > 1$  for all  $C$ , and where  $m \triangleq m_p$  is the value of  $m$  passed into the call to RISING, i.e. it is the largest value of  $m$  across all calls to RISING and SPLIT. Here we used that  $\sum_{C \in \mathcal{C}} p_C = p$ . Therefore, regardless of the balancedness of the tree, at each level we use  $O(pm)$  measurements, and as described above, there are  $O(\log p)$  levels resulting in a measurement complexity of  $O(pm \log p)$ . The factor of  $l$  arises because each call to SPLIT splits the subtrees of a node into two groups; it may take up to  $l$  calls to recover each internal node.

<sup>4</sup>Specifically  $m = m(|C|)$  can be any increasing function of  $|C|$

Lastly, we can compute edge lengths using  $O(m)$  measurements. Since this is dominated by the above bounds, we ignore this dependence.



## 4.6 Conclusion

In this chapter we studied the multi-source network tomography problem. We developed two algorithms, with theoretical guarantees, to construct tree metrics that approximate distances between end hosts in a network. We also demonstrated the effectiveness of these algorithms on real world datasets.

Turning to the themes of this thesis, this chapter lends evidence to our three claims about interactive learning. First, while we did not explicitly compare with non-interactive algorithms, we did show that our interactive approaches have strong guarantees on both statistical performance and measurement complexity. In contrast, naïve non-interactive approaches would have significantly higher measurement complexity to achieve the same level of statistical performance. Thus, we see evidence for the fact that interactivity lends statistical power in unsupervised problems.

Regarding computation, our two algorithms are also computationally efficient. As we saw, both of our algorithms have  $O(p^2 \text{polylog}(p))$  computational complexity. Naïve algorithms have significantly worse running time; the most obvious algorithm would compute all quartets and stitch these structures together, and therefore run in  $O(p^4)$  time. While it would be desirable to have linear time algorithms, we already see supporting evidence for the fact that interactivity brings computational efficiency.

Lastly, in the additive noise model, we measured uniformity via the degree of the tree. We saw that the measurement complexity of both algorithms degrades as the problems become more uniform (higher degree), and if the tree has  $\Omega(p)$  degree, then our algorithm matches a naïve non-interactive approach. This lends evidence to our claim that interactivity is powerful in the presence of non-uniformity.

## Chapter 5

# Minimaxity in the Structured Normal Means Problem

The prevalence of high-dimensional signals in modern scientific investigation has inspired an influx of research on recovering *structural information* from noisy data. These problems arise across a variety of scientific and engineering disciplines; for example identifying cluster structure in communication or social networks, multiple hypothesis testing in genomics, or anomaly detection in vision and sensor networking. Broadly speaking, this line of work shows that high-dimensional statistical inference can be performed at low signal-to-noise ratios provided that the data exhibits low-dimensional structure. Specific structural assumptions include sparsity [107], low-rankedness [74], cluster structure [118], and many others [46].

The literature in this direction focuses on three inference goals: detection, localization or recovery, and estimation or denoising. Detection tasks involve deciding whether an observation contains some meaningful information or is simply ambient noise, while recovery and estimation tasks involve more precisely characterizing the information contained in a signal. Specifically, in recovery problems, the goal is to identify, from a finite collection of signals, which signal produced the observed data. The estimation or denoising problem involves leveraging structural information to produce high-quality estimates of the signal generating the data. These problems are closely related, but also exhibit important differences, and this chapter focuses on the recovery problem.

One frustration among researchers is that algorithmic and analytic techniques for these problem differ significantly for different structural assumptions. This issue was recently resolved in the context of the estimation problem, where the *atomic norm* [46] has provided a unifying algorithmic and analytical framework, but such a theory for detection and recovery problems remains elusive. In this chapter, we provide a unification for the recovery problem, giving us better understanding of how signal structure affects statistical performance.

Modern measurement technology also often provides flexibility in designing strategies for data acquisition, and this adds an element of complexity to inference tasks. As a concrete example,

crowdsourcing platforms allow for *interactive* data acquisition, which can be used to recover cluster structure with lower measurement overhead [123, 160]. Non-interactive experimental design-based (i.e. non-uniform) data acquisition is also enabled by modern sensing technology, leading to two important questions: (1) How do we design sensing strategies for structure recovery problems? (2) When should interactive acquisition be preferred to non-interactive acquisition? We provide an answer to the first of these questions, and progress toward an answer to the latter.

To concretely describe our main contributions, we now develop the decision-theoretic framework of this chapter. We study the **structured normal means problem** defined by a finite collection of vectors  $\mathcal{V} = \{v_j\}_{j=1}^M \subset \mathbb{R}^d$  that index a family of probability distributions  $\mathbb{P}_j = \mathcal{N}(v_j, I_d)$ . An estimator  $T$  for the family  $\mathcal{V}$  is a measurable function from  $\mathbb{R}^d$  to  $[M]$ , and its maximum risk is:

$$\mathcal{R}(T, \mathcal{V}) = \sup_{j \in [M]} \mathcal{R}_j(T, \mathcal{V}), \quad \mathcal{R}_j(T, \mathcal{V}) = \mathbb{P}_j[T(y) \neq j],$$

where we always use  $y \sim \mathbb{P}_j$  to be the observation. We are interested in the **minimax risk**:

$$\mathcal{R}(\mathcal{V}) = \inf_T \mathcal{R}(T, \mathcal{V}) = \inf_T \sup_{j \in [M]} \mathbb{P}_j[T(y) \neq j]. \quad (5.1)$$

We call this the **isotropic setting** because each gaussian has spherical covariance. We are specifically interested in understanding how the complexity of the family  $\mathcal{V}$  influences the minimax risk. This setting encompasses recent work on sparsity recovery [107], biclustering [39, 118], and many graph-based problems [161]. An example to keep in mind is the  $k$ -sets problem, where the collection  $\mathcal{V}$  is formed by vectors  $\mu \mathbf{1}_S$  for subsets  $S \subset [d]$  of size  $k$  and some signal strength parameter  $\mu$ .

We also study the **experimental design setting**, where the learning algorithm can specify a sensing strategy, defined by a vector  $B \in \mathbb{R}_+^d$ . Using this strategy, under  $\mathbb{P}_j$ , the observation is:

$$y(i) \sim v_j(i) + B(i)^{-1/2} \mathcal{N}(0, 1) = \mathcal{N}(v_j(i), B(i)^{-1}), \forall i \in [d]. \quad (5.2)$$

If  $B(i) = 0$ , then we say that  $y(i) = 0$  almost surely. We call this distribution  $\mathbb{P}_{j,B}$ , to denote the dependence both on the target signal  $v_j$  and the sensing strategy  $B$ . The total measurement effort, or *budget*, used by the strategy is  $\|B\|_1$ , and we are typically interested in signal recovery under some budget constraint. Specifically, the minimax risk in this setting is:

$$\mathcal{R}(\mathcal{V}, \tau) = \inf_{T, B: \|B\|_1 \leq \tau} \sup_{j \in [M]} \mathbb{P}_{j,B}[T(y) \neq j]. \quad (5.3)$$

With this formalization, we can now state our main contributions:

1. We give nearly matching upper and lower bounds on the minimax risk for both isotropic and experimental design settings (Theorems 5.1 and 5.5). This result matches many special cases that we are aware of [161], which we show through examples. Moreover, in examples with an asymptotic flavor (defined below), this shows that the maximum likelihood estimator (MLE) achieves the minimax rate.

2. In the isotropic case, we derive a condition on the family  $\mathcal{V}$  under which the MLE exactly achieves the minimax risk, which certifies optimality of this estimator. In this case, we also give a heuristic algorithm that exploits connections to Bayesian inference and attempts to improve on the MLE. This algorithm gives some insights into how to appropriately regularize an inference problem.
3. We give sufficient conditions that certify an optimality property of an experimental design strategy and also give an algorithm for computing such a strategy prior to data acquisition. We give an example where a non-uniform strategy outperforms the isotropic one and two examples (one well-known and one new) where interactive strategies provably outperform *all* non-interactive ones. This latter result shows that interactive sampling can be significantly more powerful than non-interactive experimental design.

## 5.1 Related Work

The structured normal means problem has a rich history in statistics, although the majority of work focus on detection or estimation in nonparametric settings, for example when the signals belong to Besov or Sobolev spaces [109, 110]. More recently attention has turned to combinatorial structures and the finite dimensional case. This line is motivated by statistical applications involving complex data sources, such as tasks in graph-structured signal processing [156], and the broad goal is to understand how combinatorial structures affect both statistics and computation in these inference problems.

Focusing on detection problems, a number of papers study various combinatorial structures, including  $k$ -sets [3], cliques [161], paths [10], and clusters [156] in graphs, and for many of these problems, near-optimal detection is possible. For example, Addario-Berry et al. [3] show that to test between the null hypothesis that every component of the vector is  $\mathcal{N}(0, 1)$  and the alternative that  $k$  components have mean  $\mu$ , the detection threshold is  $\mu \asymp \sqrt{\log(1 + \frac{d}{k^2})}$ . This means that if  $\mu$  grows faster than this threshold, one can achieve error probability tending to zero, and if  $\mu$  grows slower than this threshold, all procedures have error probability tending to 1. This style of result is now available for several examples, although a unifying theory for detection problems is still undeveloped.

Turning to recovery or localization, again several specific examples have been analyzed. The most popular example is the biclustering problem, where  $\mathcal{V}$  corresponds to  $d_1 \times d_2$  matrices of the form  $\mu \mathbf{1}_{C_l} \mathbf{1}_{C_r}^T$  with  $C_l \subset [d_1], C_r \subset [d_2]$  [39, 118, 161]. However, apart from this example and a few others [161], minimax bounds for the recovery problem are largely unknown. Moreover, we are unaware of a broadly applicable analysis, like the method we develop here.

A unified treatment is possible for estimation problems, where the atomic norm framework gives sharp phase transitions for the maximum likelihood estimator [6, 46]. The atomic norm is a generic approach for encoding structural assumptions by decomposing the signal into a sparse convex combination of a set of base atoms (e.g., one-sparse vectors). While this line primarily

focuses on linear inverse problems [6, 46], there are results for the estimation problem described above [34, 136]. While we are unaware of minimax bounds for either setting, it is well known that the mean squared error of the MLE is related to the *statistical dimension* of the cone formed by the atoms. Unfortunately, atomic norm techniques rely on convex relaxation which enables estimation but not recovery, as the minimax probability of error for any dense family is one. Moreover, the non-convexity of our risk poses new challenges that do not arise with the strongly-convex mean squared error objective.

While much of the literature has focused on the isotropic case, there has been recent interest in experimental design or interactive methods, aiming to quantify the statistical improvements enabled by interactivity. The first result in this line is a simple interactive procedure for the  $k$ -sets recovery problem due to Haupt, Castro and Nowak [107]. More recently, Tanczos and Castro [161] study more structured instantiations and show more significant statistical improvements via interactive methods. Their work makes important progress, but it does not address the general problem, as they hand-craft sampling algorithms for each example. A unifying, interactive algorithm was proposed in the bandit optimization setting [48], but, in our setting, it is not known to improve on non-interactive approaches. To our knowledge, a unifying interactive algorithm and a satisfactory characterization of the advantages offered by interactive sampling remain elusive open questions. This chapter makes progress on the latter by developing lower bounds against all non-interactive approaches.

Lastly, there is a close connection between our setting and the channel coding problem in an Additive White Gaussian Noise (AWGN) Channel [59, 60]. In channel coding, we are tasked with designing a large code  $\mathcal{V}$  such that if we send the codeword  $v_j$ , an observer, upon observing  $y \sim \mathcal{N}(v_j, I_d)$ , can reliably predict the codeword sent. While the error metric is usually the same as in our setup, typical coding-theoretic results focus on *codebook design*, rather than error analysis for a particular codebook, which is our focus here. To our knowledge, the results here do not appear in the information theory literature.

## 5.2 Main Results

In this section we develop the main results of the chapter. We start by bounding the minimax risk in the isotropic setting, then develop a certificate of optimality for the maximum likelihood estimator, and turn to the algorithmic question of computing minimax optimal estimators. Lastly, we turn to the experimental design setting. We provide proofs in Section 5.5.

### 5.2.1 Bounds on the Minimax Risk

In the isotropic case, recall that we are given a finite collection  $\mathcal{V}$  of vectors  $\{v_j\}_{j=1}^M$  and an observation  $y \sim \mathcal{N}(v_j, I_d)$  for some  $j \in [M]$ . Given such an observation, a natural estimator is the maximum likelihood estimator (MLE), which outputs the index  $j$  for which the observation

was most likely to have come from. This estimator is defined as:

$$T_{\text{MLE}}(y) = \operatorname{argmax}_{j \in [M]} \mathbb{P}_j(y) = \operatorname{argmin}_{j \in [M]} \|v_j - y\|_2^2. \quad (5.4)$$

We will analyze this estimator, which partitions  $\mathbb{R}^d$  based on a Voronoi Tessellation of the set  $\mathcal{V}$ .

As stated, the running time of the estimator is  $O(Md)$ , but it is worth pausing to remark briefly about computational considerations. In many examples of interest, the class  $\mathcal{V}$  is combinatorial in nature, so  $M$  may be exponentially large, and efficient implementations of the MLE may not exist. However, as our setup does not preclude unstructured problems, the input to the estimator is the complete collection  $\mathcal{V}$ , so the running time of the MLE is linear in the input size. If the particular problem is such that  $\mathcal{V}$  can be compactly represented (e.g. it has combinatorial structure), then the estimator may not be polynomial-time computable. This presents a real issue, as researchers have shown that a minimax-optimal polynomial time estimator is unlikely to exist for the biclustering problem [51, 128], which we study in Section 5.3. However, since the primary interest of this work is statistical in nature, we will ignore computational considerations for most of our discussion.

We now turn to a characterization of the minimax risk, which involves analysis of the MLE. The following function, which we call the **Exponentiated Distance Function**, plays a fundamental role.

**Definition 5.1.** For a family  $\mathcal{V}$  and  $\alpha > 0$ , the *Exponentiated Distance Function (EDF)* is:

$$W(\mathcal{V}, \alpha) = \max_{j \in [M]} W_j(\mathcal{V}, \alpha) \quad \text{with} \quad W_j(\mathcal{V}, \alpha) = \sum_{k \neq j} \exp\left(\frac{-\|v_j - v_k\|_2^2}{\alpha}\right) \quad (5.5)$$

In the following theorem, we show that the EDF governs the performance of  $T_{\text{MLE}}$ . More importantly, this function also leads to a lower bound on the minimax risk, and the combination of these two statements shows that the MLE is nearly optimal for *any* structured normal means problem.

**Theorem 5.1.** Fix  $\delta \in (0, 1)$ . If  $W(\mathcal{V}, \delta) \leq \delta$ , then  $\mathcal{R}(\mathcal{V}) \leq \mathcal{R}(\mathcal{V}, T_{\text{MLE}}) \leq \delta$ . On the other hand, if  $W(\mathcal{V}, 2(1 - \delta)) \geq 2^{\frac{1}{1-\delta}} - 1$ , then  $\mathcal{R}(\mathcal{V}) \geq \delta$ .

In particular, by setting  $\delta = 1/2$  above, the second statement in the theorem may be replaced by: If  $W(\mathcal{V}, 1) \geq 3$ , then  $\mathcal{R}(\mathcal{V}) \geq 1/2$ . This setting often aids interpretability of the lower bound.

Notice that the value of  $\alpha$  disagrees between the lower and upper bounds, and this leads to a gap between the necessary and sufficient conditions. This is not purely an artifact of our analysis, as there are many examples where the MLE does not exactly achieve the minimax risk. However, most structured normal means problems also have an asymptotic flavor, specified by a sequence of problems  $\mathcal{V}_1, \mathcal{V}_2, \dots$ , and a signal-strength parameter  $\mu$ , with observation  $y \sim \mathcal{N}(\mu v_j, I_d)$  for some signal  $v_j$  in the current family. In this asymptotic framework, we are interested in how  $\mu$  scales with the sequence to drive the minimax risk to one or zero. Almost all existing examples in the literature are of this form [161], and in all such problems, Theorem 5.1 shows that the

MLE achieves the minimax rate. To our knowledge, such a comprehensive characterization of recovery problems is entirely new.

Application of Theorem 5.1 to instantiations of the structured normal means problem requires bounding the EDF, which is significantly simpler than the typical derivation of this style of result. In particular, proving a lower bound no longer requires construction of a specialized subfamily of  $\mathcal{V}$  as was the de facto standard in this line of work [118, 161]. In Section 5.3, we show how simple calculations can recover existing results.

Turning to the proof, the EDF arises naturally as an upper bound on the failure probability of the MLE after applying a union bound and a Gaussian tail bound. Indeed the fact that the EDF upper bounds the minimax risk is not particularly surprising. It is however more surprising that it also provides a lower bound on the minimax risk. We obtain this bound via application of Fano's Inequality, but we use a version that allows a non-uniform prior and explicitly construct this prior using the EDF. This leads to our more general lower bound.

## 5.2.2 Minimax-Optimal Recovery

Theorem 5.1 shows that that maximum likelihood estimator achieves near-optimal performance for *all* structured normal means recovery problems. By near-optimal, we mean that in problems with some asymptotic flavor, where the family of vectors grows but also becomes more separated, the maximum likelihood estimator achieves the minimax rate. However, in many cases the MLE is not the optimal estimator, i.e. it does not achieve the exact minimax risk. In this section, we use deeper connections between the minimax risk and the Bayes risk to address this gap. Specifically, we give a sufficient condition for the minimax optimality of the MLE, and we will also design an algorithm that in other cases produces an estimator with better minimax risk.

Our approach is based on a well-known connection between the minimax risk and the Bayes risk. For a structured normal means problem defined by a family  $\mathcal{V}$ , the *Bayes risk* for an estimator  $T$  under prior  $\pi \in \Delta_{M-1}$  is given by:

$$B_\pi(T) = \sum_{j=1}^M \pi_j \mathbb{P}_j[T(y) \neq j].$$

We say that an estimator  $T$  is the Bayes estimator for prior  $\pi$  if it achieves the minimum Bayes risk. A simple calculation reveals the structure of the Bayes estimator for any prior  $\pi$  and this structural characterization is essential to our development.

**Proposition 5.2.** *For any prior  $\pi$ , the Bayes estimator  $T_\pi$  has polyhedral acceptance regions, that is the estimator is of the form:*

$$T(Y) = j \text{ if } y \in A_j,$$

with  $A_j = \{x : \Gamma_j x \geq b_j\}$  and  $\Gamma_j \in \mathbb{R}^{M \times d}$  has  $v_j - v_k$  in the  $k$ th row and  $b_j$  has  $\frac{1}{2}(\|v_j\|_2^2 - \|v_k\|_2^2) + \log \frac{\pi_k}{\pi_j}$  in the  $k$ th entry. These polyhedral sets  $A_j$  partition the space  $\mathbb{R}^d$ .



We also exploit the relationship between the minimax risk and the Bayes risk. This is a well known result, although for completeness we provide a proof in Subsection 5.5.4. The prior  $\pi$  below is known as the *least-favorable prior*.

**Proposition 5.3.** *Suppose that  $T$  is a Bayes estimator for some prior  $\pi$ . If the risk  $\mathcal{R}_j(T) = \mathcal{R}_{j'}(T)$  for all  $j \neq j' \in [M]$ , then  $T$  is a minimax optimal estimator.*

Our main theoretical result leverages this proposition along with the structural characterization of Bayes estimators to certify minimax optimality of the MLE. The sufficient condition for optimality depends on a particular structure of the family  $\mathcal{V}$ :

**Definition 5.2.** *A family  $\mathcal{V}$  is **unitarily invariant** if there exists a set of orthogonal matrices  $\{R_i\}_{i=1}^N$  such that for each vector  $v \in \mathcal{V}$ , the set  $\{R_i v\}_{i=1}^N$  is exactly  $\mathcal{V}$ .*

In other words, the instance  $\mathcal{V}$  can be generated by applying the orthogonal transforms to any fixed vector in the collection. Unitarily invariant problems exhibit high degrees of symmetry and via Proposition 5.3, can be shown to be a sufficient condition for the optimality of the MLE.

**Theorem 5.4.** *If  $\mathcal{V}$  is unitarily invariant, then the MLE is minimax optimal.*

Some remarks about the theorem are in order:

1. This theorem reduces the question of optimality to a purely geometric characterization of the family  $\mathcal{V}$  and, as we will see, many well studied problems are unitarily invariant. One common family of orthogonal matrices is the set of all permutation matrices on  $\mathbb{R}^d$ .
2. This result does not characterize the risk of the MLE; it only shows that no other estimator has better risk. Specifically, it does not provide an analytic bound that is sharper than Theorem 5.1. From a practitioner’s perspective, an optimality certificate for an estimator is more important than a bound on the risk as it help govern practical decisions, although risk bounds enable theoretical comparison.
3. Lastly, the result is not asymptotic in nature but rather shows that the MLE achieves the *exact* minimax risk for a fixed family  $\mathcal{V}$ . We are not aware of any other results in the literature that certify optimality of the MLE under our measure of risk.

The proof of this theorem is based on the observation that the point-wise risk  $\mathcal{R}_j(\mathcal{V})$  is exactly  $1 - \mathbb{P}_j[A_j]$  where  $\mathbb{P}_j$  is the gaussian measure centered at  $v_j$  and  $A_j$  is a particular polytope based on the Voronoi Tessellation of the point set  $\mathcal{V}$ . We use this characterization and the unitary invariance of the family to show that the risk landscape for the MLE is constant across the hypotheses  $v_j$ . Finally, we employ a dual characterization of the minimax risk, to show that if the risk landscape is constant, then the MLE must be optimal.

In problems where Theorem 5.4 can be applied, we now have a complete story for the isotropic case. We know that the MLE exactly achieves the minimax risk and Theorem 5.1 also gives satisfactory upper and lower bounds. However, many problems of interest do not have unitarily invariant structure, and, in many of these problems, the MLE is suboptimal.

To improve on the MLE in these settings we now develop an algorithm for finding a better estimator. Our approach is to optimize over the space of priors  $\pi$  in an iterative fashion, starting

---

**Algorithm 12** PRIOROPT( $\mathcal{V}$ )

---

Initialize  $\pi(j) = 1/M$  for each  $j \in [M]$ .

For each  $j \in [M]$ , compute  $g_j = \mathbb{P}_j[A_j]$  where  $A_j = \{x \in \mathbb{R}^d : \Gamma_j x \geq b_j\}$  where  $\Gamma_j$  has  $v_j - v_k$  in the  $k$ th row and  $b_j$  has  $\|v_j\|^2/2 - \|v_k\|^2/2 + \log(\pi_k/\pi_j)$  in the  $k$ th entry.

**while** Entropy of  $\frac{\vec{g}}{\sum_{j=1}^m g_j}$  is far from  $\log M$ . **do**

Let  $j_{\min} = \operatorname{argmin}_j g_j$  and  $j_{\max} = \operatorname{argmax}_j g_j$ .

Update  $\pi(j_{\min}) = \pi(j_{\min}) + \eta_t$ ,  $\pi(j_{\max}) = \pi(j_{\max}) - \eta_t$  for some step size  $\eta_t$ .

Recompute  $g_j$  using new prior.

**end while**

---

at the uniform prior  $\pi_0$ , whose Bayes estimator is the MLE. At each iteration, we compute the risk functional of the current Bayes estimator, which, by Proposition 5.2, is related to gaussian volumes of a collection of polytopes. These gaussian volumes can be approximated with Monte Carlo sampling. We find the hypothesis  $j_{\min}$  and  $j_{\max}$  with lowest and highest risk respectively and adjust the prior by shifting mass from  $j_{\min}$  to  $j_{\max}$ . The aim is to move the prior so as to flatten out the risk functional. See Algorithm 12 for a more precise sketch.

The algorithm is based on zero-th order ascent of the entropy of a particular distribution. The distribution is the normalized risk functional, and the parameter space is the prior distribution  $\pi$  on the hypothesis. By maximizing entropy, we aim to make this distribution uniform which amounts to making the risk functional constant. By Proposition 5.3, this would lead to a minimax optimal estimator. Specifically, the algorithm aims to solve the following program:

$$\operatorname{maximize}_{\pi \in \Delta_{M-1}} H(g_{1,\pi}, g_{2,\pi}, \dots, g_{M,\pi}) \quad (5.6)$$

where  $g_{j,\pi} = \mathbb{P}_j[A_{j,\pi}]$  and  $A_{j,\pi}$  is the polytope given in Proposition 5.2 for prior  $\pi$  and  $H : \mathbb{R}_+^M \rightarrow \mathbb{R}$  is the entropy functional after normalizing the argument to lie on the simplex.

Unfortunately, it is not clear whether Program 5.6 is convex in the parameter  $\pi$ . The main challenge in analyzing this program is that the point-wise risks  $g_{j,\pi}$  involve the gaussian volume of arbitrary polyhedral sets, and these are not, in general, analytically tractable quantities. Therefore it is not clear how the parameter  $\pi$  affects the objective function here, which precludes analysis of this algorithm.

The gaussian volumes also pose a computational barrier, as even computing these volumes tend to be difficult. While there has been research on approximating the gaussian volume of a convex set [58], these algorithms require that one initially knows a small ball contained entirely in the set. In principle, one could do this here by solving a linear program to find a point in the interior of the polytope, but we find that Monte Carlo sampling is significantly more straightforward. Monte Carlo sampling seemed to work well for problems in moderate dimension.

So while this algorithm is not known to have convergence guarantees, iterates do tend to have higher entropy and therefore have more uniform risk landscapes. BayesThis means that even though the algorithm does not necessarily find the least favorable prior, it does lead to a prior whose Bayes estimator is an improvement over the maximum likelihood estimator, except for in

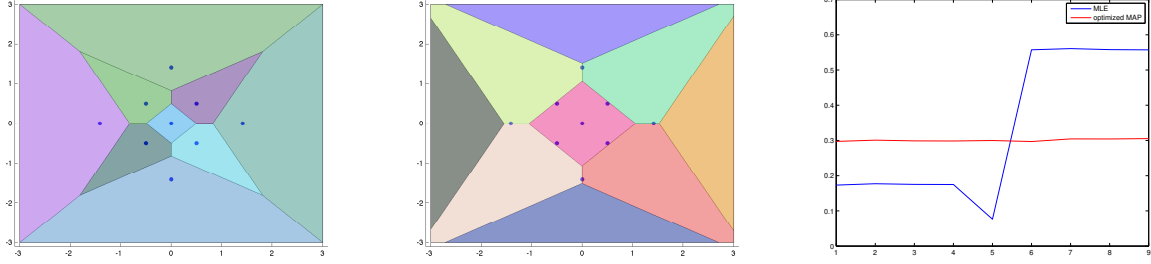


Figure 5.1: Example structured normal means problem on nine points in two dimensions. Left: polyhedral acceptance regions of MLE. Center: Acceptance regions of Bayes estimator from the optimized prior computed by Algorithm 12. Right: Success probability landscape (success probability for each hypothesis) for the two estimators, demonstrate that the optimized estimator has better minimax risk.

cases where the MLE is optimal. For many problems, it is therefore worth running even a few iterations of this algorithm to obtain a slightly better estimator.

One interpretation of this algorithm is in terms of *regularization*. The prior computed can be viewed as a regularizer and the ensuing Bayes estimator can be viewed as a regularized MLE. With this lens, Algorithm 12 can be thought of as computing a good regularizer for the structured normal means problem defined by  $\mathcal{V}$ . Unfortunately, we have no rigorous guarantees on Algorithm 12, although we hope this interpretation can influence future work on choosing regularizers.

In Figure 5.1, we demonstrate this algorithm and compare against the MLE. The example has nine points in two dimensions (which enables visualization) and the left panel shows the polyhedral acceptance regions of the MLE. The central panel shows the acceptance region of the Bayes estimator computed by Algorithm 12 and the right panel shows the risk landscape of these two estimators. Specifically, in the third panel, the  $x$ -axis corresponds to the nine hypotheses, and the lines denote  $\mathbb{P}_j[A_j]$ , which, as we saw, is just one minus the risk for hypothesis  $j$ . The minimax risk is therefore one minus the minimum value on these curves.

Notice that the risk landscape of the optimized estimator is essentially constant which roughly certifies that it is minimax optimal (By Proposition 5.3). More qualitatively, the minimax risk of this optimized estimator is significantly better than that of the MLE. The reason for this is that, under the MLE, the acceptance region for the central hypothesis is very small, so the MLE has low acceptance probability for that hypothesis. The optimized estimator uses an expanded acceptance region for this hypothesis which increases the acceptance probability and decreases the minimax risk. Of course, this comes at the cost of decreasing the acceptance probability for other hypotheses, which leads to a flattening of the risk landscape.

### 5.2.3 The Experimental Design Setting

Recall the experimental design setting, where the statistician specifies a strategy  $B \in \mathbb{R}_+^d$  and receives observation  $y \sim \mathbb{P}_{j,B}$  given by Equation 5.2. Our main insight is that the choice of  $B$  only changes the metric structure of  $\mathbb{R}^d$ , and this change can be incorporated into the proof of Theorem 5.1. Specifically, the likelihood for hypothesis  $j$ , under sampling strategy  $B$  is:

$$\mathbb{P}_j(y|B) = \prod_{i=1}^d \sqrt{\frac{B(i)}{2\pi}} \exp(-B(i)(v_j(i) - y(i))^2/2)$$

and the maximum likelihood estimator is:

$$T_{\text{MLE}}(y, B) = \operatorname{argmin}_{j \in [M]} \|v_j - y\|_B^2$$

where  $\|v\|_B^2 = \sum_{i=1}^d v(i)^2 B(i)$  is the Mahalanobis norm induced by the diagonal matrix  $\operatorname{diag}(B)$ .

Theorem 5.1 can be ported directly to this setting, leading to the following:

**Theorem 5.5.** Fix  $\delta \in (0, 1)$  and any sampling strategy  $B$  with  $\|B\|_1 \leq \tau$ . Define the **Sampling Exponentiated Distance Function SEDF**:

$$W(\mathcal{V}, \alpha, B) = \max_{j \in [M]} \sum_{k \neq j} \exp\left(\frac{-\|v_j - v_k\|_B^2}{\alpha}\right) \quad (5.7)$$

If  $W(\mathcal{V}, 8, B) \leq \delta$  then  $\mathcal{R}(\mathcal{V}, \tau) \leq \mathcal{R}(\mathcal{V}, T_{\text{MLE}}(y, B)) \leq \delta$ . Conversely, if  $W(\mathcal{V}, 2(1 - \delta), B) \geq 2^{\frac{1}{1-\delta}} - 1$ , then  $\inf_T \sup_{j \in [M]} \mathbb{P}_{j,B}[T(y) \neq j] \geq \delta$ .

The structure of the theorem is almost identical to that of Theorem 5.1, but it is worth making some important observations. First, the theorem holds for any non-interactive sampling strategy  $B \in \mathbb{R}_+^d$ , so the upper bound is strictly more general than Theorem 5.1. Secondly, any non-interactive strategy can be used to derive an upper bound on the minimax risk, but the same is not true for the lower bound. Instead the lower bound provided by the theorem is dependent on the strategy, so one must still minimize over sampling strategies to lower bound  $\mathcal{R}(\mathcal{V}, \tau)$ . Note that this theorem also applies to the non-isotropic or heteroscedastic case with known, shared covariance.

Fortunately, the SEDF is convex in  $B$  so it can be numerically minimized over the polyhedron  $\{z : 0 \leq z_i \leq 1, \sum_{i=1}^d z_i \leq B\}$ . Specifically, for any  $\alpha$ , we solve the convex program:

$$\operatorname{minimize}_{B \in \mathbb{R}_+^d, \|B\|_1 \leq \tau} \max_{j \in [M]} \sum_{k \neq j} \exp\left(\frac{-\|v_j - v_k\|_B^2}{\alpha}\right), \quad (5.8)$$

To obtain the sampling strategy  $\hat{B}$  that minimizes the SEDF. For example, solving Program 5.8 with  $\alpha = 1$  results in a strategy  $\hat{B}$ , and if  $W(\mathcal{V}, 1, \hat{B}) \geq 3$ , then we know that the minimax risk  $\mathcal{R}(\mathcal{V}, \tau)$  over all strategies is at least  $1/2$ . On the other hand, solving with  $\alpha = 8$  to obtain a (different) sampling strategy  $\hat{B}$  and then using  $\hat{B}$  with the MLE would give the tightest upper bound

on the risk attainable by our proof technique. In Section 5.3, we demonstrate an example where this optimization leads to a non-uniform sampling strategy that outperforms uniform sampling.

In the general setting, it is challenging to analytically certify that an allocation strategy  $\hat{B}$  minimizes the SEDF, but in some cases it is possible. Specializing the first-order optimality conditions for Program 5.8 to our setting gives the following:

**Proposition 5.6.** *Let  $\hat{B}$  be an sampling strategy with  $\|B\|_1 = \tau$ . Let  $S(\hat{B}) \subset \mathcal{V}$  be the set of hypotheses achieving the maximum in  $W(\mathcal{V}, \alpha, \hat{B})$  and let  $\pi$  be a distribution on  $S(\hat{B})$ . If, for all  $i, i' \in [d]$ ,*

$$\mathbb{E}_{j \sim \pi} \sum_{k \neq j} (v_k(i) - v_j(i))^2 \exp(-\|v_k - v_j\|_B^2) = \mathbb{E}_{j \sim \pi} \sum_{k \neq j} (v_k(i') - v_j(i'))^2 \exp(-\|v_k - v_j\|_B^2),$$

*then  $\hat{B}$  is a minimizer of  $W(\mathcal{V}, \alpha, B)$  subject to  $\|B\|_1 \leq \tau$ .*

While application of this result could involve a number of non-trivial calculations, there are many cases where it does lead to analytic lower bounds for particular problems. Specifically, the result is especially useful when  $\hat{B}$  is uniform across the coordinates, and  $S(\hat{B}) = [M]$ , so that all of the hypotheses achieve the maximum. In this case, it often suffices to choose  $\pi$  to be uniform over the hypotheses and exploit the high degree of symmetry to demonstrate the condition holds. As we will see in Section 5.3, many examples studied in the literature exhibit the requisite symmetry for this proposition to be applied in a straightforward manner.

We remark that Tanczos and Castro [161] establish a similar sufficient condition for the uniform sampling strategy to be optimal. Their result however is slightly less general in that it only certifies optimality for the uniform sampling strategy, whereas ours, in principle, can be applied more universally. In addition, their result applies only to problems where the hypotheses are of the form  $\mu 1_S$  for a collection of subsets while ours is more general, and this generality is important for some examples (e.g., the hierarchical clustering example in Section 5.3). The other main difference is that their approach is not based on the SEDF, so their result is not directly applicable here.

## 5.3 Examples

To demonstrate the scope of our results, we present four instantiations of structured normal means problems, and derive results easily attainable from our general approach. These examples have the asymptotic flavor described before, where we are interested in how a signal strength parameter  $\mu$  scales with a sequence of problem instances. To simplify presentation, we state the results in terms of the *minimax rate*  $\psi$  and use the notation  $\mu \asymp \psi$  where  $\psi$  is a function that depends on the parameters of the sequence (e.g., the dimension). This notation means that if  $\mu = \omega(1)\psi$ , then the minimax risk can be driven to zero and conversely, if  $\mu = o(1)\psi$ , then the minimax risk approaches one asymptotically.

The first example, the  $k$ -sets problem, is well studied, and as a warmup, we show how our technique recovers existing results. The second example is the biclustering problem; this problem is interesting because there is polynomial separation between non-interactive and interactive procedures, and our technique can be used to establish lower bounds on all non-interactive approaches. The third example is a graph-structured signal processing problem, and this example is interesting because our technique generalizes existing results, but also because uniform sampling may not be optimal. In the last example, we use Theorem 5.1 to demonstrate the achievability of the channel capacity in additive white gaussian noise (AWGN) channel, showing how an easy calculation can reproduce the proof of Shannon [155]. The requisite calculations for these examples are deferred to Section 5.5.

### 5.3.1 $k$ -sets

In the  $k$ -sets problem, we have  $M = \binom{d}{k}$  and each vector  $v_j = \mathbf{1}_{S_j}$  where  $S_j \subset [d]$  and  $|S_j| = k$ . The observation is  $y \sim \mathcal{N}(\mu v_j, I_d)$  for some hypothesis  $j$ .

**Corollary 5.7.** *The minimax rate for the  $k$ -sets problem is  $\mu \asymp \sqrt{\log(k(d-k))}$  and with budget constraint  $\tau$ , it is  $\mu \asymp \sqrt{\frac{d}{\tau} \log(k(d-k))}$ . In the isotropic case, the MLE is minimax optimal.*

This corollary follows simply by bounding the EDF for the  $k$ -sets problem using binomial approximations. Using Proposition 5.6, it is easy to verify that uniform sampling is optimal here, which immediately gives the second claim. Finally using the set of all permutation matrices and exploiting symmetry, we can easily verify that this class is unitarily invariant. These bound agrees with established results in the literature [161].

### 5.3.2 Biclusters

In the biclustering problem, we instead work over  $\mathbb{R}^{d \times d}$  and let  $M = \binom{d}{k}^2$ . We parametrize the class  $\mathcal{V}$  with two indices so that  $v_{ij} = \mathbf{1}_{S_i} \mathbf{1}_{S_j}^T$  is a  $d \times d$  matrix with  $k^2$  non-zeros with  $|S_i| = |S_j| = k$ . The observation is  $y \sim \mathcal{N}(\mu \text{vec}(v_{ij}), I_{d^2})$  for a hypothesis  $(i, j)$ .

**Corollary 5.8.** *The minimax rate for the biclustering problem is  $\mu \asymp \sqrt{\frac{\log(k(d-k))}{k}}$  and with budget constraint  $\tau$ , it is  $\mu \asymp \sqrt{\frac{d^2}{\tau k} \log(k(d-k))}$ . In the isotropic case, the MLE is minimax optimal.*

Our bounds agree with existing analyses of this class [39, 118, 161]. Obtaining this result involves simply bounding the EDF using binomial approximations as in the  $k$ -sets example, and straightforward applications of Theorem 5.4 and Proposition 5.6 with the uniform distribution.

The biclustering problem is interesting because a simple interactive algorithm has significantly better statistical performance. The algorithm first samples coordinates of the matrix randomly,

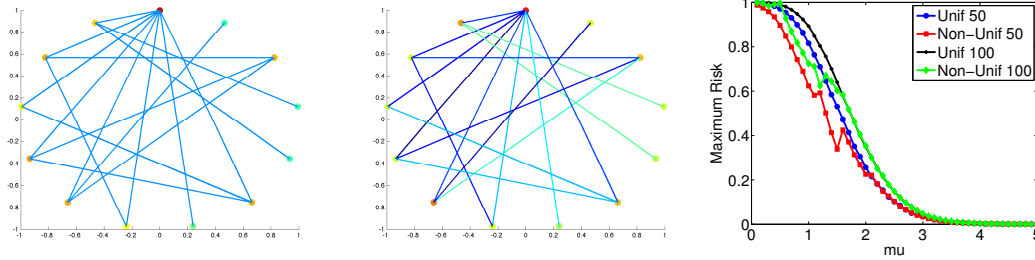


Figure 5.2: Left: A realization of the stars problem for a graph with 13 vertices and 34 edges with sampling budget  $\tau = 34$ . Edge color reflects allocation of sensing energy and vertex color reflects success probability for MLE under that hypothesis (warmer colors are higher for both). Isotropic (left) has minimum success probability of 0.44 and experimental design (center) has minimum success probability 0.56. Right: Maximum risk for isotropic and experimental design sampling as a function of  $\mu$  for stars problem on a 50 and 100-vertex graph.

with enough energy so as to reliably test if a coordinate is active or not, until it finds an active coordinate. It then senses on the row and column of that coordinate and identifies the rows and columns that are active in the bicluster. Tanczos and Castro [161] show that this algorithm succeeds if  $\mu = \omega\left(\sqrt{\left(\frac{d^2}{\tau k^2} + \frac{d}{\tau}\right) \log d}\right)$ , which is a factor of  $\sqrt{k}$  smaller than the lower bound established here, demonstrating concrete statistical gains from interactivity. Note that this separation is known [161]. We provide a crude by sufficient analysis of this interactive algorithm in Section 5.5.

### 5.3.3 Stars

Let  $G = (V, E)$  be a graph and let the edges be numbered  $1, \dots, d$ . The class  $\mathcal{V}$  is the set of all stars in the graph, that is the vector  $v_j \in \{0, 1\}^d$  is the indicator vector of all edges emanating from the  $j$ th node in the graph. Again the observation is  $y \sim \mathcal{N}(\mu v_j, I_d)$  for some  $j \in [|V|]$ .

**Corollary 5.9.** *In the stars problem if the ratio between the maximum and minimum degree is bounded by a constant, i.e.  $\frac{\text{deg}_{\max}}{\text{deg}_{\min}} \leq c$ , then the minimax rate is  $\mu \asymp \sqrt{\frac{\log(|V| - \text{deg}_{\min})}{\text{deg}_{\min}}}$ .*

Again this agrees with a recent result of Tanczos and Castro [161], who consider  $s$ -stars of the complete graph, formed by choosing a vertex, and then activating  $s$  of the edges emanating out of that vertex. The two bounds agree in the special case of the complete graph with  $s = |V| - 1$ , but otherwise are incomparable, as they consider different problem structures. Note that the degree requirement here is not fundamental in Theorem 5.1, but rather a shortcoming of our calculations.

We highlight this example because the uniform allocation strategy does not necessarily minimize  $W(\mathcal{V}, \alpha, B)$ . In Figure 5.2, we construct a graph according to the Barabási-Albert model [5] and consider the class of stars on this graph. The simulation results show that optimizing the SEDF to find a sampling strategy is never worse than uniform sampling, and for low signal strengths it can lead to significantly lower maximum risk. We believe the increases in the right-most plot

are caused by numerical instability arising from the non-smooth optimization problem 5.8. Note that the risk for both uniform and non-uniform sampling approaches zero as  $\mu \rightarrow \infty$ , so for large  $\mu$ , there is little advantage to optimizing the sampling scheme.

### 5.3.4 Random Codes

Consider a collection  $\mathcal{V}$  of  $M$  vectors with coordinates that are i.i.d.  $\mathcal{N}(0, P)$ . In expectation over the draw of the  $M$  vectors, the Bayes risk of the maximum likelihood estimator under the uniform prior is:

$$\mathbb{E}_{\mathcal{V}} \mathbb{E}_{j \sim \text{Unif}[M]} \mathbb{P}_j[\text{error}] \leq (M - 1)(1 + P/2)^{-d/2}$$

In other words if  $M = o((1 + P/2)^{d/2})$ , then the maximum likelihood decoder can drive the probability of error to zero as  $d \rightarrow \infty$ .

In information theoretic terms, this quick calculation roughly says that there exists a rate  $R = \log(M)/d = \frac{1}{2} \log(1 + P/2) - \omega(1/d)$  code with power constraint  $P$ , that can be reliably transmitted over an additive white noise gaussian (AWGN) channel with noise variance 1. This nearly matches Shannon's Channel Coding theorem [59] which says that the rate cannot exceed the channel capacity, which in our case is  $\frac{1}{2} \log(1 + P)$ .

There are two small weaknesses of this calculation in comparison with the classical achievability of the channel capacity. The first is that our bound involves the term  $\log(1 + P/2)$  instead of  $\log(1 + P)$  in the definition of channel capacity. We suspect this is due to weakness in our bounding technique in Theorem 5.1, which in part allows for significantly more generality than this special case. The second is that the codewords we use are drawn from  $\mathcal{N}(0, P)$  so they will exceed the power constraint  $\|v\|_2^2 \leq P$  with constant probability. This shortcoming can be remedied by instead using  $\mathcal{N}(0, P - c/d)$  and applying well known  $\chi^2$  deviation bounds.

## 5.4 Discussion

In this chapter, we studied the structured normal means problem and gave a unified characterization of the minimax risk both for isotropic and experimental design settings. Our work gives insights into how to choose estimators (e.g., the optimality certificate for the MLE) and how to design sampling strategies for structure recovery problems. Our lower bounds are critical in demonstrating separation between non-interactive and interactive sampling, which is an important research direction.

There are a number of exciting directions for future work, including extensions to other structure discovery problems such as detection, and to other observation models, such as compressive observations. We are most interested in developing a unifying theory for interactive sampling, analogous to the theory developed here. The challenges with developing such an understanding are both algorithmic and information theoretic, and we are excited to tackle these challenges.



## 5.5 Proofs

### 5.5.1 Proof of Theorem 5.1

**Analysis of MLE:** We first analyze the maximum likelihood estimator:

$$T_{MLE}(y) = \operatorname{argmin}_{j \in [M]} \|v_j - y\|_2^2$$

This estimator succeeds as long as  $\|v_k - y\|_2^2 > \|v_{j^*} - y\|_2^2$  for each  $k \neq j^*$ , when  $y \sim \mathbb{P}_{j^*}$ . This condition is equivalent to:

$$\|v_k - y\|_2^2 > \|v_{j^*} - y\|_2^2 \Leftrightarrow \langle \epsilon, v_k - v_{j^*} \rangle < \frac{1}{2} \|v_{j^*} - v_k\|_2^2$$

where  $\epsilon \sim \mathcal{N}(0, 1)$ . This follows from writing  $y = v_{j^*} + \epsilon$  and then expanding the squares. So we must simultaneously control all of these events, for fixed  $j^*$ :

$$\begin{aligned} & \mathbb{P}_{\epsilon \sim \mathcal{N}(0, I_d)} [\forall k \neq j^* . \langle \epsilon, v_k - v_{j^*} \rangle < \|v_{j^*} - v_k\|_2^2 / 2] \\ &= 1 - \mathbb{P}_{\epsilon \sim \mathcal{N}(0, I_d)} [\exists k \neq j^* . \langle \epsilon, v_k - v_{j^*} \rangle \geq \|v_{j^*} - v_k\|_2^2 / 2] \\ &\geq 1 - \sum_{k \neq j^*} \mathbb{P}_{\epsilon \sim \mathcal{N}(0, I_d)} [\langle \epsilon, v_k - v_{j^*} \rangle \geq \|v_{j^*} - v_k\|_2^2 / 2] \end{aligned}$$

By a gaussian tail bound, this probability is:

$$\mathbb{P}_{\epsilon \sim \mathcal{N}(0, I_d)} [\langle \epsilon, v_k - v_{j^*} \rangle \geq \|v_{j^*} - v_k\|_2^2 / 2] \leq \exp \left\{ -\frac{1}{8} \|v_{j^*} - v_k\|_2^2 \right\}$$

So that the total failure probability is upper bounded by:

$$\mathbb{P}_{j^*} [\hat{j} = j^*] \leq \sum_{k \neq j^*} \exp \left\{ -\frac{1}{8} \|v_{j^*} - v_k\|_2^2 \right\} = W_{j^*}(\mathcal{V}, 8)$$

So if  $j$  is the truth, then the probability of error is smaller than  $\delta$  when  $W_j(\mathcal{V}, 8) \leq \delta$ . For the maximal (over hypothesis choice  $j$ ) probability of error to be smaller than  $\delta$ , it suffices to have  $W(\mathcal{V}, 8) \leq \delta$ .

**Fundamental Limit:** We now turn to the fundamental limit. We start with a version of Fano's inequality with non-uniform prior.

**Lemma 5.10** (Non-uniform Fano Inequality). *Let  $\Theta = \{\theta\}$  be a parameter space that indexes a family of probability distributions  $P_\theta$  over a space  $\mathcal{X}$ . Fix a prior distribution  $\pi$ , supported on  $\Theta$  and consider  $\theta \sim \pi$  and  $X \sim P_\theta$ . Let  $f : \mathcal{X} \rightarrow \Theta$  be any possibly randomized mapping, and let  $p_e = \mathbb{P}_{\theta \sim \pi, X \sim P_\theta} [f(X) \neq \theta]$  denote the probability of error. Then:*

$$p_e \geq 1 - \frac{\sum_{\theta} \pi(\theta) KL(P_\theta || P_\pi) + \log 2}{H(\pi)}$$

where  $P_\pi(\cdot) = \mathbb{E}_{\theta \sim \pi} P_\theta(\cdot)$  is the mixture distribution. In particular, we have:

$$\inf_f \sup_\theta \mathbb{P}_{X \sim P_\theta}[f(X) \neq \theta] \geq \inf_f \mathbb{E}_{\theta \sim \pi} \mathbb{P}_{X \sim P_\theta}[f(X) \neq \theta] \geq 1 - \frac{\sum_\theta \pi(\theta) KL(P_\theta || P_\pi) + \log 2}{H(\pi)}$$

*Proof.* Consider the Markov Chain  $\theta \rightarrow X \rightarrow \hat{\theta} \triangleq f(X)$  where  $\theta \sim \pi$  and  $X|\theta \sim P_\theta$ . Let  $E = \mathbf{1}[\hat{\theta} \neq \theta]$ .

$$H(E|X) + H(\theta|E, X) = H(E, \theta|X) = H(\theta|X) + H(E|\theta, X) \geq H(\theta|X)$$

Now,  $H(E|\theta, X) \geq 0$  and since conditioning only reduces entropy, we have the inequality


$$\begin{aligned} H(\theta|X) &\leq H(p_e) + H(\theta|E, X) = H(p_e) + H(\theta|E=0, X)P[E=0] + H(\theta|E=1, X)P[E=1] \\ &= H(p_e) + p_e H(\theta) \end{aligned}$$

which proves the usual version of Fano's inequality. We want to write  $H(\theta|X)$  in terms of the KL divergence, using the mixture distribution  $P_\pi$ .

$$\begin{aligned} H(\theta|X) &= H(\theta, X) - H(X) = \int \sum_\theta \pi(\theta) P_\theta(x) \log \left( \frac{\sum_\theta \pi(\theta) P_\theta(x)}{\pi(\theta) P_\theta(x)} \right) dx \\ &= \sum_\theta \pi(\theta) \int P_\theta(x) \log \left( \frac{P_\pi(x)}{P_\theta(x)} \right) dx - \sum_\theta \pi(\theta) \log \pi(\theta) \\ &= - \sum_\theta \pi(\theta) KL(P_\theta || P_\pi) + H(\pi) \end{aligned}$$

Combining these gives the bound:

$$H(p_e) + p_e H(\pi) \geq H(\pi) - \sum_\theta \pi(\theta) KL(P_\theta || P_\pi),$$

By upper bounding  $H(p_e) \leq \log 2$  and rearranging we prove the claim. 

For a distribution  $\pi \in \Delta_{M-1}$  over the hypothesis, let  $P_\pi(\cdot) = \sum_k \pi_k P_k(\cdot)$  be the mixture distribution. Then Fano's inequality (Lemma 5.10) states that the minimax probability of error is lower bounded by:

$$\begin{aligned} \mathcal{R}(\mathcal{V}) &= \inf_T \sup_j \mathbb{P}_j[T(y) \neq j] \geq \inf_T \mathbb{E}_{j \sim \pi} \mathbb{E}_{y \sim j} \mathbf{1}[T(y) \neq j] \\ &\geq 1 - \frac{\mathbb{E}_{k \sim \pi} KL(P_k || P_\pi) + \log 2}{H(\pi)}. \end{aligned}$$

Fix  $\delta \in (0, 1)$  and let  $j^* = \operatorname{argmax}_{j \in [M]} W_j(2(1-\delta))$ . We will use a prior based on this quantity:

$$\pi_k \propto \exp \left( - \frac{\|v_{j^*} - v_k\|_2^2}{2(1-\delta)} \right)$$

With this prior, the entropy becomes:

$$\begin{aligned}
H(\pi) &= \sum_k \pi_k \log \left( \frac{\sum_i \exp \left( -\frac{\|v_{j^*} - v_i\|_2^2}{2(1-\delta)} \right)}{\exp \left( -\frac{\|v_{j^*} - v_k\|_2^2}{2(1-\delta)} \right)} \right) \\
&= \log(W(\mathcal{V}, 2(1-\delta)) + 1) + \sum_k \pi_k \frac{\|v_{j^*} - v_k\|_2^2}{2(1-\delta)} \\
&= \log(W(\mathcal{V}, 2(1-\delta)) + 1) + \frac{1}{1-\delta} \sum_k \pi_k KL(P_k || P_{j^*})
\end{aligned}$$

The 1 inside the first log comes from the fact that in the definition  $W_{j^*}$ , we do not include the term involving  $j^*$  in the sum, while our prior  $\pi$  does place mass proportional to 1 on hypothesis  $j^*$ . The term involving the KL-divergence follows from the fact that the KL between two gaussians is one-half the  $\ell_2^2$ -distance between their means.

Looking at the lower bound from Fano's inequality, we see that if:

$$\mathbb{E}_{k \sim \pi} KL(P_k || P_\pi) + \log 2 \leq (1-\delta)H(\pi) = (1-\delta) \log(W(\mathcal{V}, 2(1-\delta)) + 1) + \sum_k \pi_k KL(P_k || P_{j^*})$$

then the probability of error is lower bounded by  $\delta$ . Of course it is immediate that:

$$\begin{aligned}
\sum_k \pi_k KL(P_k || P_{j^*}) &= \sum_k \pi_k \int P_k(x) \log \left( \frac{P_k(x) P_\pi(x)}{P_\pi(x) P_{j^*}(x)} \right) \\
&= \sum_k \pi_k \int P_k(x) \log \frac{P_k(x)}{P_\pi(x)} + \sum_k \int \pi_k P_k(x) \log \frac{P_\pi(x)}{P_{j^*}(x)} \\
&= \sum_k \pi_k KL(P_k || P_\pi) + KL(P_\pi || P_{j^*}) \geq \mathbb{E}_k KL(P_k || P_\pi)
\end{aligned}$$

So the condition reduces to requiring that:

$$\log 2 \leq (1-\delta) \log(W(\mathcal{V}, 2(\delta-1)) + 1).$$

After some algebra, this is equivalent to:

$$W(\mathcal{V}, 2(\delta-1)) \geq 2^{\frac{1}{1-\delta}} - 1$$



## 5.5.2 Proof of Theorem 5.5

The proof of Theorem 5.5 is essentially the same as the proof of Theorem 5.1, coupled with two observations. First, for a sampling strategy  $B \in \mathbb{R}_+^d$  the maximum likelihood estimator is:

$$T_{\text{MLE}}(y, B) = \operatorname{argmin}_{j \in [M]} \|v_j - y\|_B^2$$

so the analysis of the MLE depends on the Mahalanobis norm  $\|\cdot\|_B$  instead of the  $\ell_2$  norm.

Similarly, the KL divergence between the distribution  $\mathbb{P}_{j,B}$  and  $\mathbb{P}_{k,B}$  depends on the Mahalanobis norm  $\|\cdot\|_B$  instead of the  $\ell_2$  norm. Specifically, we have:

$$KL(\mathbb{P}_{j,B}||\mathbb{P}_{k,B}) = \frac{1}{2}\|v_j - v_k\|_B^2.$$

The lower bound proof instead use this metric structure, but the calculations are equivalent.



### 5.5.3 Proof of Proposition 5.6

To simplify the presentation, let  $f(B) = W(\mathcal{V}, \alpha, B)$ .  $f(B)$  is convex and (strictly) monotonically decreasing, so we know that the minimum will be achieved when the constraint is tight, i.e. when  $\|B\|_1 = \tau$ . The Lagrangian is:

$$\mathcal{L}(B, \lambda) = f(B) + \lambda(\|B\|_1 - \tau)$$

and the minimum is achieved at  $\hat{B}$ , with  $\|\hat{B}\|_1 = \tau$ , if there is a value  $\hat{\lambda}$  such that  $0 \in \partial\mathcal{L}(\hat{B}, \hat{\lambda})$ . Observing that the subgradient is  $\partial f(B) + \lambda\mathbf{1}$ , it suffices to ignore the Lagrangian term and instead ensure that  $\partial f(B) \propto \mathbf{1}$ .  $f(B)$  is a maximum of  $M$  convex functions, where  $f_j(B)$  is the function corresponding to hypothesis  $v_j$ , and, by direct calculation, the subgradient of this function  $f_j(B)$  is:

$$\frac{\partial f_j(B)}{\partial B_i} = \sum_{k \neq j} -(v_k(i) - v_j(i))^2 \exp(-\|v_k - v_j\|_B^2).$$

Moreover, the subgradient of the maximum of a set of functions is the convex hull of the subgradients of all functions achieving the maximum. This means that if there exists a distribution  $\pi$ , supported over the maximizers of  $f(\hat{B})$ , such that the expectation of the subgradients is constant,

we have certified optimality of  $\hat{B}$ . This is precisely the condition in the Proposition.



### 5.5.4 Proof of Theorem 5.4

**Proof of Proposition 5.2:** To prove Proposition 5.2, we make two claims. First we certify that for a prior  $\pi$ , the Maximum a Posteriori (MAP) estimator is a Bayes estimator for prior  $\pi$ . Given  $\pi$ , the map estimator is:

$$T_\pi(y) = \underset{j}{\operatorname{argmax}} \pi(j) \exp\{-\|v_j - y\|_2^2/2\}$$

Define the posterior risk of an estimator  $T$  to be the expectation of the loss, under the posterior distribution on the hypothesis. In our case this is:

$$r(T|y) = \sum_{j=1}^M \mathbf{1}[T(y) \neq j] \pi(j|y) \quad \text{where} \quad \pi(j|y) \propto \pi(j) \exp\{-\|v_j - y\|_2^2/2\}.$$

For a fixed  $y$ , this quantity is minimized by letting  $T(y)$  be the maximizer of the posterior, as this makes the 0 – 1 loss term zero for the largest  $\pi(j|y)$  value. Thus for each  $y$  we minimize the posterior risk by letting  $T(y)$  be the MAP estimate. The result follows by the well known fact that if an estimator minimizes the posterior risk at each point, then it is the Bayes estimator.

This argument shows that the only types of estimators we need to analyze are MAP estimators under various priors. This gives us the requisite structure to prove Proposition 5.2.


Specifically, for a prior  $\pi$ , for the MAP estimate to predict hypothesis  $j$ , it must be the case that:

$$\forall k \neq j. \quad \pi_j \exp\{-\|v_j - y\|_2^2/2\} \geq \pi_k \exp\{-\|v_j - v_k\|_2^2/2\}.$$

This can be simplified to:


$$\langle v_j - v_k, y \rangle \geq \frac{1}{2}(\|v_j\|_2^2 - \|v_k\|_2^2) + \log \frac{\pi_k}{\pi_j}.$$

Thus the acceptance region for the hypothesis  $j$  is the set of all points  $y$  that satisfy all of these

$M - 1$  inequalities. This is exactly the polyhedral set  $A_j$ . 

**Proof of Proposition 5.3:** We provide a proof of this well-known result showing that the Bayes estimator with uniform risk landscape is minimax optimal. Let  $T_\pi$  be the Bayes estimator under prior  $\pi$  and let  $T_0$  be some other estimator. Since  $T_\pi$  has constant risk landscape, we know that  $\max_j \mathcal{R}_j(\mathcal{V}, T_\pi) = B_\pi(T_\pi)$ , or the minimax risk for  $T_\pi$  is equal to its Bayes risk. We know that the Bayes risk of  $T_0$  is at most the minimax risk for  $T_0$ , i.e.  $B_\pi(T_0) \leq \max_j \mathcal{R}_j(\mathcal{V}, T_0)$ . If it were the case that  $T_0$  had strictly lower minimax risk, then we have:

$$B_\pi(T_0) \leq \max_j \mathcal{R}_j(\mathcal{V}, T_0) < \max_j \mathcal{R}_j(\mathcal{V}, T_\pi) \leq B_\pi(T_\pi).$$

However, this is a contradiction since  $T_\pi$  is the Bayes estimator under prior  $\pi$ , meaning that it minimizes the Bayes risk. 

**Proof of Theorem 5.4:** Our goal is to apply Proposition 5.3. By the fact that  $\mathcal{R}_j(\mathcal{V}, T) = 1 - \mathbb{P}_j[A_j]$  where  $A_j$  is  $T$ 's acceptance region for hypothesis  $j$ , we must show that the  $\mathbb{P}_j$  probability content of the acceptance regions are constant. Ignoring the normalization factor of the gaussian density, this is:

$$\int_{A_j} \exp\{-\|v_j - x\|_2^2/2\} dx,$$

where  $A_j = \{z | \Gamma_j z \geq b_j\}$  as defined in Proposition 5.2. We will exploit the unitary invariance of the family.

For any pair of hypothesis  $j, k$ , let  $R_{jk}$  be the orthogonal matrix such that  $v_k = R_{jk}v_j$  and note that  $R_{kj}$ , the orthogonal matrix that maps  $v_k$  to  $v_j$ , is just  $R_{jk}^T$ . This also means that  $R_{jk}R_{jk}^T = R_{jk}R_{kj} = I$ . Via a change of variables  $x = R_{kj}y$ , the integrand becomes:

$$\exp\{-\|v_j - R_{kj}y\|_2^2/2\} = \exp\{-\|R_{jk}v_j - R_{jk}R_{kj}y\|_2^2/2\} = \exp\{-\|v_k - y\|_2^2/2\}.$$

Thus, we have translated to the  $P_k$  measure.


As for the region of integration, first note that since  $v_i = R_{ji}v_j$ , it must be the case that  $\|v_j\|_2^2 = \|v_i\|_2^2$  for all  $j, i \in [M]$ . This means that the vector  $b_j$  defining the acceptance region, which for the MLE has coordinates  $b_j(i) = \frac{1}{2}(\|v_j\|_2^2 - \|v_i\|_2^2)$ , is just the all-zeros vector. The region of integration is therefore:

$$\{z | \Gamma_j z \geq 0\} = \{z | \Gamma_j R_{kj} z \geq 0\}.$$

We must check that this polytope is exactly  $A_k$ , which means that we must check that for each  $i$ ,  $(v_j - v_i)^T R_{kj}$  is a row of the  $\Gamma_k$  matrix. But:

$$(v_j - v_i)^T R_{kj} = v_j^T R_{jk}^T - v_i^T R_{jk}^T = v_k^T - v_i^T R_{jk}^T.$$

Since  $v_i$  can generate the family  $\mathcal{V}$ , it must be the case that  $R_{jk}v_i \in \mathcal{V}$  so that this difference does correspond to some row of  $\Gamma_k$ . Since we apply the same unitary operator to all of the rows, it must be the case that the number of distinct rows is unchanged, or in other words, there is a bijection from the rows in  $\Gamma_j R_{kj}$  to the rows in  $\Gamma_k$ . Therefore, the transformed region of integration, after the change of variable  $x = R_{kj}y$  is exactly the acceptance region  $A_k$ , and the integrand is the  $\mathbb{P}_k$  measure. This means that  $\mathbb{P}_k[A_k] = \mathbb{P}_j[A_j]$  and this is true for all pairs  $(j, k)$ , so that the risk

landscape is constant. By Proposition 5.3, this certifies optimality of the MLE. 

### 5.5.5 Calculations for the examples

**Calculations for  $k$ -Sets:** We must upper and lower bound  $W(\mathcal{V}, \alpha)$ . First note that by symmetry, every hypothesis achieves the maximum, so it suffices to compute just one of them:

$$W(\mathcal{V}, \alpha) = \sum_{k \neq j} \exp(-\|v_k - v_j\|_2^2/\alpha) = \sum_{s=1}^k \binom{k}{s} \binom{d-k}{s} \exp(-2s\mu^2/\alpha).$$

This follows by noting that the  $\ell_2^2$  distance between two hypothesis is the symmetric set difference between the two subsets, and then by a simple counting argument. Using well known bounds on

binomial coefficients, we obtain:

$$\begin{aligned}
W(\mathcal{V}, \alpha) &\leq \sum_{s=1}^k \exp(s \log(ke/s) + s \log((d-k)e/s) - 2s\mu^2/\alpha) \\
&= \sum_{s=1}^k \exp(s \log(e^2k(d-k)/s^2) - 2s\mu^2/\alpha) \\
&\leq k \exp(\log e^2k(d-k) - 2\mu^2/\alpha) \quad \text{if } 2\mu^2/\alpha \geq \log(e^2k(d-k))
\end{aligned}$$

This is smaller than  $\delta$  whenever  $\mu^2 \geq \alpha \log(ek(d-k)/\delta)$ , which subsumes the requirement above. For the lower bound:

$$W(\mathcal{V}, \alpha) \geq \sum_{s=1}^k \exp(s \log(k/s) + s \log((d-k)/s) - 2s\mu^2/\alpha) \geq \exp(-2\mu^2/\alpha + \log(k(d-k)))$$

which goes to infinity if  $\mu^2 = o(\alpha \log(k(d-k)))$ .

To certify that the uniform allocation strategy minimizes  $W(\mathcal{V}, \alpha, B)$ , we apply Proposition 5.6. Fix  $\tau$  and let  $\hat{B}$  be such that  $\hat{B}(i) = \tau/d$ . By symmetry, every hypothesis achieves the maximum under this allocation strategy, and we will take  $\pi$  to be the uniform distribution over all hypothesis.

For a hypothesis  $j$  and a coordinate  $i$ , the subgradient  $\frac{\partial f_j(B)}{\partial B(i)}$  at  $\hat{B}_i$  depends on whether  $v_j(i) = 0$  or not. If  $v_j(i) = 0$ , then:

$$\frac{\partial f_j(B)}{\partial B(i)} = \mu^2 \sum_{s=1}^k \binom{d-k-1}{s-1} \binom{k}{k-s} \exp(-2\tau\mu^2s^2/d),$$

and if  $v_j(i) = \mu^2$  then:

$$\frac{\partial f_j(B)}{\partial B(i)} = \mu^2 \sum_{s=1}^k \binom{d-k}{s} \binom{k-1}{k-s} \exp(-2\tau\mu^2s^2/d).$$

Both of these follow from straightforward counting arguments. Notice that the value of the subgradient depends only on whether  $v_j(i) = 0$  or not, and under the uniform distribution  $\pi$ ,  $\mathbb{E}_{j \sim \pi} v_j(i) = \mathbb{E}_{j \sim \pi} v_j(i')$ . This implies that the constant vector is in the subgradient of  $f(B)$  at  $\hat{B}$ , so that  $\hat{B}$  is the minimizer of  $W(\mathcal{V}, \alpha, B)$  subject to  $\|B\|_1 \leq \tau$ .

We have already done the requisite calculation to bound the minimax risk under sampling. The calculations above show that if  $\mu = \omega(\sqrt{\frac{d}{\tau}} \log(k(d-k)))$  then the maximum likelihood estimator, when using the uniform sampling strategy has risk tending to zero. Conversely if  $\mu = o(\sqrt{\frac{d}{\tau}} \log(k(d-k)))$  then the minimax risk, for *any* allocation strategy tends to one.

**Calculation for Biclusters:** Due to symmetry, all hypotheses achieve the maximum and therefore, we can directly calculate  $W(\mathcal{V}, \alpha)$ . We use the notation  $C_n^i$  to denote the binomial coefficient  $\binom{n}{i}$ .

$$\begin{aligned} W(\mathcal{V}, \alpha) &= \sum_{s_r=1}^k \sum_{s_c=1}^k C_k^{s_r} C_k^{s_c} C_{d-k}^{s_r} C_{d-k}^{s_c} \exp\left(-\frac{2\mu^2}{\alpha}(s_r(k-s_c) + s_c(k-s_r) + s_r s_c)\right) \\ &\quad + \sum_{s_r=1}^k C_k^{s_r} C_{d-k}^{s_r} \exp\left(-\frac{2\mu^2}{\alpha}(s_r k)\right) + \sum_{s_c=1}^k C_k^{s_c} C_{d-k}^{s_c} \exp\left(-\frac{2\mu^2}{\alpha}(s_c k)\right) \end{aligned}$$

This last two term comes from the case where  $s_c = 0$  or  $s_r = 0$ , which is all of the hypotheses that share the same columns but disagree on the rows (or share the same rows but disagree on the columns). Using binomial approximations, the first term can be upper bounded by:

$$\begin{aligned} &\leq \sum_{s_r=1}^k \sum_{s_c=1}^k \exp\left(s_r \log \frac{k(d-k)e^2}{s_r^2} + s_c \log \frac{k(d-k)e^2}{s_c^2} - \frac{2\mu^2}{\alpha}(s_r(k-s_c/2) + s_c(k-s_r/2))\right) \\ &\leq \sum_{s_r=1}^k \exp\left(s_r \left(\log \frac{k(d-k)e^2}{s_r^2} - \frac{k\mu^2}{\alpha}\right)\right) \sum_{s_c=1}^k \exp\left(s_c \left(\log \frac{k(d-k)e^2}{s_c^2} - \frac{k\mu^2}{\alpha}\right)\right). \end{aligned}$$

The two terms here are identical, so we will just bound the first one:

$$\begin{aligned} &\sum_{s_r=1}^k \exp\left(s_r \left(\log \frac{k(d-k)e^2}{s_r^2} - \frac{k\mu^2}{\alpha}\right)\right) \\ &\leq \sum_{s_r=1}^k \exp\left(s_r (\log(k(d-k)e^2) - k\mu^2/\alpha)\right) \\ &\leq k \exp\left(\log(k(d-k)e^2) - k\mu^2/\alpha\right) \quad \text{if } \mu^2 \geq \frac{\alpha}{k} \log(k(d-k)e^2) \end{aligned}$$

Applying this inequality to both terms gives a bound on  $W(\mathcal{V}, \alpha)$ . This bound is smaller than  $\delta$  as long as  $\mu \geq \sqrt{\frac{c}{k\alpha} \log(k(d-k)e/\delta)}$  for some universal constant  $c$ . Again this subsumes the condition required for the inequality to hold.

The other two terms are essentially the same. Using binomial approximations, both expressions can be bounded as:

$$\begin{aligned} \sum_{s_r=1}^k C_k^{s_r} C_{d-k}^{s_r} \exp\left(-\frac{2\mu^2}{\alpha}(s_r k)\right) &= \sum_{s_r=1}^k \exp\left(s_r \log(e^2 k(d-k)/s_r^2) - 2s_r k\mu^2/\alpha\right) \\ &\leq k \exp\left(\log(k(d-k)e^2) - 2k\mu^2/\alpha\right) \quad \text{if } \mu^2 \geq \frac{\alpha}{2k} \log(k(d-k)e^2). \end{aligned}$$

These bounds lead to the same minimax rate as above.



For the lower bound, we again use binomial approximations.

$$\begin{aligned}
W(\mathcal{V}, \alpha) &\geq \sum_{s_r=1}^k \sum_{s_c=1}^k \exp \left( s_r \log \frac{k(d-k)}{s_r^2} + s_c \log \frac{k(d-k)}{s_c^2} - \frac{2\mu^2}{\alpha} (s_r(k-s_c/2) + s_c(k-s_r/2)) \right) \\
&\geq \sum_{s_r=1}^k \exp \left( s_r \left( \log \frac{k(d-k)e^2}{s_r^2} - \frac{2k\mu^2}{\alpha} \right) \right) \sum_{s_c=1}^k \exp \left( s_c \left( \log \frac{k(d-k)e^2}{s_c^2} - \frac{2k\mu^2}{\alpha} \right) \right) \\
&\geq \exp(\log(k(d-k) - 2\mu^2 k/\alpha)^2)
\end{aligned}$$

This lower bound goes to infinity if  $\mu = o(\sqrt{\frac{1}{k} \log(k(d-k))})$  lower bounds the minimax rate.

To certify that the uniform allocation strategy minimizes  $W(\mathcal{V}, \alpha, B)$ , we apply Proposition 5.6. Fix  $\tau$  and let  $\hat{B}$  be such that  $\hat{B}((a, b)) = \tau/d^2$  for all  $(a, b) \in [d] \times [d]$ . By symmetry, every hypothesis achieves the maximum under this allocation strategy, and we will take  $\pi$  to be the uniform distribution over all hypothesis.

For a hypothesis  $j$ , let  $f_j(B)$  denote the term in the SEDF centered around  $j$ . For a hypothesis  $j$  based on clusters  $S_l, S_r$  and a coordinate  $(a, b)$ , the subgradient  $\frac{\partial f_j(B)}{\partial B(a, b)}$  at  $\hat{B}(a, b)$  depends on whether  $a \in S_l$  and  $b \in S_r$ . If  $a \notin C_l$  and  $b \notin C_r$ , then:

$$\left. \frac{\partial f_j(B)}{\partial B(a, b)} \right|_{B=\hat{B}} = \frac{-\mu^2}{\alpha} \sum_{s_r=1}^k \sum_{s_c=1}^k C_{d-k-1}^{s_r-1} C_k^{s_r} C_{d-k-1}^{s_c-1} C_k^{s_c} \exp\left(\frac{-2\tau\mu^2}{\alpha d^2} (s_r(k-s_c/2) + s_c(k-s_r/2))\right).$$

This follows by direct calculation. Similar calculations yield the other cases:

$$\left. \frac{\partial f_j(B)}{\partial B(a, b)} \right|_{B=\hat{B}} = \frac{-\mu^2}{\alpha} \sum_{s_r=0}^{k-1} \sum_{s_c=1}^k C_{d-k}^{s_r} C_{k-1}^{s_r} C_{d-k-1}^{s_c-1} C_k^{s_c} \exp\left(\frac{-2\tau\mu^2}{\alpha d^2} (s_r(k-s_c) + s_c(k-s_r) + s_r s_c)\right).$$

$$\left. \frac{\partial f_j(B)}{\partial B(a, b)} \right|_{B=\hat{B}} = \frac{-\mu^2}{\alpha} \sum_{s_r=1}^k \sum_{s_c=0}^{k-1} C_{d-k-1}^{s_r-1} C_k^{s_r} C_{d-k}^{s_c} C_{k-1}^{s_c} \exp\left(\frac{-2\tau\mu^2}{\alpha d^2} (s_r(k-s_c) + s_c(k-s_r) + s_r s_c)\right).$$

$$\left. \frac{\partial f_j(B)}{\partial B(a, b)} \right|_{B=\hat{B}} = \frac{-\mu^2}{\alpha} \sum_{s_r=0}^{k-1} \sum_{s_c=0}^{k-1} C_{d-k}^{s_r} C_{k-1}^{s_r} C_{d-k}^{s_c} C_{k-1}^{s_c} \exp\left(\frac{-2\tau\mu^2}{\alpha d^2} (s_r(k-s_c) + s_c(k-s_r) + s_r s_c)\right).$$

These correspond to the cases  $a \in S_l, b \notin S_r$ ,  $a \notin S_l, b \in S_r$  and the case where  $a \in S_l, b \in S_r$  respectively. The main point is that the value of the subgradient depends only on presence or absence of the row/column in the cluster, and under the uniform distribution  $\pi$ , each row/column is equally likely to be in the cluster. This means that for every coordinate  $(a, b)$  taking the expected subgradient with respect to the uniform distribution over hypotheses yields the same expression. So the constant vector is in the subgradient of  $f(B)$  at  $\hat{B}$ , so that  $\hat{B}$  is the minimizer of  $W(\mathcal{V}, \alpha, B)$  subject to  $\|B\|_1 \leq \tau$ .

We have already done the requisite calculation to bound the minimax risk under sampling. The calculations above show that if  $\mu = \omega(\sqrt{\frac{d^2}{k\tau} \log(k(d-k))})$  then the maximum likelihood estimator, when using the uniform sampling strategy has risk tending to zero. Conversely if  $\mu = o(\sqrt{\frac{d^2}{k\tau} \log(k(d-k))})$  then the minimax risk, for *any* allocation strategy tends to one.

The biclusters family is clearly unitarily invariant with respect to the set of orthonormal matrices that permute the rows and columns independently. The family is easiest to describe as acting on the matrices  $\mathbf{1}_{S_l} \mathbf{1}_{S_r}^T$ . Let  $P_l, P_r$  be any two  $d \times d$  permutation matrices. Then the matrix  $P_l \mathbf{1}_{S_l} (\mathbf{1}_{S_r} P_r)^T$  is clearly another hypothesis, and as we vary  $P_l$  and  $P_r$  we generate all of the hypothesis. Note that these permutations are unitary operators on the matrix space  $\mathbb{R}^{d \times d}$ , which allows us to apply Theorem 5.4.

For the analysis of the interactive algorithm, let us first bound the probability that the algorithm makes a mistake on any single coordinate. Consider sampling a coordinate  $x$  with mean  $\mu$  and noise variance  $1/b$ . A Gaussian tail bound reveals that:

$$\mathbb{P}[|x - \mu| \geq \epsilon] \leq 2 \exp(-2b\epsilon^2).$$

We will sample no more than  $d^2$  coordinates and we will sample each coordinate with the same amount of energy  $b$ . So by the union bound, the probability that we make a single mistake in classifying a coordinate that we query is bounded by  $\delta/2$  as long as:

$$\mu \geq 2\epsilon = \sqrt{\frac{2}{b} \log(4d^2/\delta)}.$$

We now need to bound  $b$ , which depends on the total number of coordinates queried by the algorithm. In the first phase of the algorithm, we sample coordinates uniformly at random until we hit one that is active. Since each sample hits an active coordinate with probability  $k^2/d^2$ :

$$\mathbb{P}[\text{hit active coordinate in } T \text{ samples}] = 1 - (1 - k^2/d^2)^T \geq 1 - \frac{1}{e^{Tk^2/d^2}},$$

or if  $T = \frac{d^2}{k^2} \log(2/\delta)$ , the probability that we hit an active coordinate in  $T$  samples will be at least  $1 - \delta/2$ . The total number of samples we use then can be upper bounded by  $2d + \frac{d^2}{k^2} \log(2/\delta)$ , which means that we can allocate our budget  $\tau$  evenly over these coordinates. Therefore we can set  $b = \tau(2d + \frac{d^2}{k^2} \log(2/\delta))^{-1}$ , and plugging into the condition on  $\mu$  above proves the result.

**Calculation for Stars:** For the stars problem, define  $\text{Nb}(j) \subset V$  to be the neighbors of the vertex  $j$  in the graph. For a fixed hypothesis  $j$ , we have

$$\begin{aligned} W_j(\mathcal{V}, \alpha) &= \sum_{k \neq j} \exp(-\|v_k - v_j\|_2^2 / \alpha) \\ &= \sum_{k \in \text{Nb}(j)} \exp(-\mu^2(\deg(k) + \deg(j) - 2)/\alpha) + \sum_{k \notin \text{Nb}(j)} \exp(-\mu^2(\deg(k) + \deg(j))/\alpha) \\ &\leq \exp(-\mu^2 \deg_{\min}/\alpha - \mu^2 \deg(j)/\alpha) (\deg(j) \exp(2\mu^2/\alpha) + |V| - \deg(j)) \end{aligned}$$

This last inequality follows by replacing every  $\deg(k)$  with  $\deg_{\min}$ , the lower bound on the degrees. This last expression is maximized with  $\deg(j) = \deg_{\min}$ , which can be observed by noticing that the derivative with respect to  $\deg(j)$  is negative. This gives the bound:

$$W(\mathcal{V}, \alpha) \leq \exp(-2\mu^2 \deg_{\min}/\alpha) (\deg_{\min} \exp(2\mu^2/\alpha) + |V| - \deg_{\min})$$

One can lower bound  $W(\mathcal{V}, \alpha)$  by choosing the hypothesis  $j$  with  $\deg(j) = \deg_{\min}$  and then replacing all other degree terms with  $\deg_{\max}$  in the above calculations. This gives:

$$W(\mathcal{V}, \alpha) \geq \exp\left(-\frac{\mu^2}{\alpha}(\deg_{\min} + \deg_{\max})\right) \left(\deg_{\min} e^{2\mu^2/\alpha} + |V| - \deg_{\min}\right)$$

**Calculation for Random Codes:** In the proof of Theorem 5.1, we saw that for a hypothesis  $j$ , we can bound the probability of error by:

$$\mathbb{P}_j[\text{error}] \leq \sum_{k \neq j} \exp(-\|v_j - v_k\|_2^2/8)$$

This means that:

$$\begin{aligned} \mathbb{E}_{\mathcal{V}} \mathbb{E}_{j \sim \text{Unif}([M])} \mathbb{P}_j[\text{error}] &\leq \mathbb{E}_{\mathcal{V}} \frac{1}{M} \sum_{j=1}^M \sum_{k \neq j} \exp(-\|v_j - v_k\|_2^2/8) \\ &= (M-1) \mathbb{E}_{v, v'} \exp(-\|v - v'\|_2^2/8) = (M-1) \prod_{j=1}^d \mathbb{E}_{x \sim \chi_1^2} \exp(-Px/4) \\ &= (M-1)(1 + P/2)^{-d/2}. \end{aligned}$$

Notice that the only inequality in this sequence is the first one, which is essentially an application of Theorem 5.1. The last equality is based on the moment-generating function of a  $\chi_1^2$  random variable.

To achieve the bound on the rate of the code, set this final expression to be at most some  $f(d)$  which is  $o(1)$ . Then the probability of error is at most  $f(d) \rightarrow 0$  and the rate  $R$  is:

$$R = \frac{\log M}{d} = \frac{1}{2} \log(1 + P/2) + \frac{\log(f(d))}{d} = \frac{1}{2} \log(1 + P/2) - \omega(1/d)$$



# Chapter 6

## Conclusions

In this thesis we studied interactive and non-interactive algorithms for several unsupervised learning problems with a focus on understanding the advantages in statistics and computation enabled by the interactive paradigm. Demonstration of the statistical advantage of interactive learning requires both new algorithmic ideas and new technology to establish fundamental limits on learning paradigms. In this thesis we made progress in both directions; we developed several new interactive learning algorithms and also showed strong limits on non-interactive algorithms. Combined these sets of results make a compelling statistical case for interactive learning.

In the examples considered here, we saw how *uniformity* governed the level of statistical improvement offered by interactive learning. In problem instances with high degrees of non-uniformity, which was measured differently in each problem, we saw that interactive approaches are significantly stronger than non-interactive ones. While there is at present no unifying theory capturing this effect, but we believe that the examples considered here lend evidence to the importance of non-uniformity for interactive learning.

Regarding computation, many of the interactive algorithms developed here are faster than existing non-interactive ones. We made claims about the computational advantage of interactivity in a non-rigorous way, as establishing running-time lower bounds is often quite challenging. Nevertheless, we find this claim to be quite surprising, as it is not, at present, demonstrated in the literature on interactive supervised learning.

While there is still much to be explored regarding interactivity in machine learning, we hope the results in this thesis have made a compelling case for this paradigm. We look forward to future advances in this direction and a deeper understanding of interactive learning.



# Appendix A

## Concentration Inequalities

Here we collect a number of well-known large deviation bounds used throughout the thesis.

**Proposition A.1** (Scalar Bernstein). *Let  $X_1, \dots, X_n$  be independent, centered scalar random variables with  $\sigma^2 = \sum_{i=1}^n \mathbb{E}[X_i^2]$  and  $R = \max_i |X_i|$ . Then:*

$$\mathbb{P} \left( \sum_{i=1}^n X_i \geq t \right) \leq \exp \left\{ \frac{-t^2}{2\sigma^2 + \frac{2}{3}Rt} \right\} \quad (\text{A.1})$$

**Proposition A.2** (Vector Bernstein [97]). *Let  $X_1, \dots, X_n$  be independent centered random vectors with  $\sum_{i=1}^n \mathbb{E} \|X_i\|_2^2 \leq V$ . Then for any  $t \leq V(\max_i \|X_i\|_2)^{-1}$ :*

$$\mathbb{P} \left( \left\| \sum_{i=1}^n X_i \right\|_2 \geq \sqrt{V} + t \right) \leq \exp \left\{ \frac{-t^2}{4V} \right\} \quad (\text{A.2})$$

**Proposition A.3** (Matrix Bernstein [165]). *Let  $X_1, \dots, X_n$  be independent, random, self-adjoint matrices with dimension  $d$  satisfying:*

$$\mathbb{E}X_k = 0 \quad \text{and} \quad \|X_k\|_2 \leq R \text{ almost surely.}$$

Then, for all  $t \geq 0$ ,

$$\mathbb{P} \left( \left\| \sum_{k=1}^n X_k \right\| \geq t \right) \leq d \exp \left( \frac{-t^2/2}{\sigma^2 + Rt/3} \right) \quad \text{where} \quad \sigma^2 = \left\| \sum_{k=1}^n \mathbb{E}X_k^2 \right\|$$

**Proposition A.4** (Rectangular Matrix Bernstein [165]). *Let  $X_1, \dots, X_n$  be independent random matrices with dimension  $d_1 \times d_2$  satisfying:*

$$\mathbb{E}X_k = 0 \quad \text{and} \quad \|X_k\|_2 \leq R \text{ almost surely.}$$

*Define:*

$$\sigma^2 = \max \left\{ \left\| \sum_{k=1}^n \mathbb{E}(X_k X_k^T) \right\|_2, \left\| \sum_{k=1}^n \mathbb{E}(X_k^T X_k) \right\|_2 \right\}.$$

*Then, for all  $t \geq 0$ ,*

$$\mathbb{P} \left( \left\| \sum_{k=1}^n X_k \right\|_2 \geq t \right) \leq (d_1 + d_2) \exp \left( \frac{-t^2/2}{\sigma^2 + Rt/3} \right).$$



# Bibliography

- [1] Dimitris Achlioptas and Frank Mcsherry. Fast computation of low-rank matrix approximations. *Journal of the ACM*, April 2007. [14](#), [22](#), [33](#), [34](#), [43](#)
- [2] Dimitris Achlioptas, Zohar S. Karnin, and Edo Liberty. Near-optimal entrywise sampling for data matrices. In *Advances in Neural Information Processing Systems*, 2013. [14](#)
- [3] Louigi Addario-Berry, Nicolas Broutin, Luc Devroye, and Gábor Lugosi. On combinatorial testing problems. *The Annals of Statistics*, 2010. [95](#)
- [4] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert E Schapire. Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. In *International Conference on Machine Learning*, 2014. [7](#), [44](#)
- [5] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 2002. [105](#)
- [6] Dennis Amelunxen, Martin Lotz, Michael B. McCoy, and Joel A. Tropp. Living on the edge: Phase transitions in convex programs with random data. *Information and Inference*, 2014. [95](#), [96](#)
- [7] Animashree Anandkumar, Avinatan Hassidim, and Jonathan Kelner. Topology discovery of sparse random graphs with few participants. *ACM SIGMETRICS Performance Evaluation Review*, 2011. [70](#), [72](#)
- [8] Dana Angluin. Queries and concept learning. *Machine learning*, 1988. [6](#)
- [9] Dana Angluin. Queries revisited. *Theoretical Computer Science*, 2004. [6](#)
- [10] Ery Arias-Castro and Emmanuel J. Candès. Searching for a trail of evidence in a maze. *The Annals of Statistics*, 2008. [95](#)
- [11] Ery Arias-Castro, Emmanuel J. Candès, and Mark A. Davenport. On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 2013. [2](#), [7](#)
- [12] Sanjeev Arora, Elad Hazan, and Satyen Kale. A fast random sampling algorithm for sparsifying matrices. *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 2006. [14](#)
- [13] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*, 2002. [7](#)
- [14] Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. In *Advances in neural information processing systems*, 2009. [7](#)

- [15] Pranjali Awasthi, Maria-Florina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. In *International Conference on Machine Learning*, 2013. 8, 44
- [16] Sivaraman Balakrishnan, Min Xu, Akshay Krishnamurthy, and Aarti Singh. Noise Thresholds for Spectral Clustering. In *Advances in Neural Information Processing Systems*, 2011. 5, 42, 45, 47, 48, 49, 51, 53, 62
- [17] Sivaraman Balakrishnan, Mladen Kolar, Alessandro Rinaldo, and Aarti Singh. Recovering block-structured activations using compressive measurements. *arXiv:1209.3431*, 2012. 1, 7, 14
- [18] Maria-Florina Balcan and Avrim Blum. Clustering with Interactive Feedback. In *Algorithmic Learning Theory*, 2008. 8, 44
- [19] Maria-Florina Balcan and Steve Hanneke. Robust Interactive Learning. In *Conference on Learning Theory*, 2011. 1
- [20] Maria-Florina Balcan and Phil Long. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, 2013. 3, 6
- [21] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Conference on Learning Theory*, 2007. 1
- [22] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 2009. 1, 6
- [23] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning Journal*, 2010. 1
- [24] Maria-Florina Balcan, Yingyu Liang, and Pramod Gupta. Robust Hierarchical Clustering. *Conference on Learning Theory*, 2010. 43
- [25] Laura Balzano, Benjamin Recht, and Robert D. Nowak. High-dimensional matched subspace detection when data are missing. In *IEEE International Symposium on Information Theory*. IEEE, 2010. 24, 25
- [26] Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998. 7
- [27] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In *SIAM International Conference on Data Mining*, 2004. 8, 44
- [28] Alina Beygelzimer, Sham Kakade, and John Langford. Cover trees for nearest neighbor. In *International Conference on Machine Learning*, 2006. 44
- [29] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *International Conference on Machine Learning*, 2009. 1
- [30] Alina Beygelzimer, John Langford, Zhang Tong, and Daniel J. Hsu. Agnostic Active Learning Without Constraints. In *Advances in Neural Information Processing Systems*, 2010. 1, 3, 6
- [31] Alina Beygelzimer, John Langford, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandit algorithms with supervised learning guarantees. In *Artificial Intelligence and Statistics*, 2011. 7

- [32] Alina Beygelzimer, Hal Daumé III, John Langford, and Paul Mineiro. Learning Reductions that Really Work. *arXiv preprint arXiv:1502.02704*, 2015. 44
- [33] Shankar Bhamidi, Ram Rajagopal, and Sébastien Roch. Network delay inference from additive metrics. *Random Structures & Algorithms*, 2010. 69, 72
- [34] Badri Narayan Bhaskar, Gongguo Tang, and Benjamin Recht. Atomic norm denoising with applications to line spectral estimation. *IEEE Transactions on Signal Processing*, 2013. 96
- [35] Evan E Bolton, Yanli Wang, Paul A Thiessen, and Stephen H Bryant. PubChem: integrated platform of small molecules and biological activities. *Annual reports in computational chemistry*, 2008. 39
- [36] Christos Boutsidis, Michael W. Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. *ACM-SIAM Symposium on Discrete Algorithms*, 2009. 14
- [37] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near optimal column-based matrix reconstruction. In *IEEE Symposium on Foundations of Computer Science*, 2011. 14
- [38] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 2012. 7
- [39] Cristina Butucea and Yuri I. Ingster. Detection of a sparse submatrix of a high-dimensional noisy matrix. *Bernoulli*, 2013. URL <http://projecteuclid.org/euclid.bj/1386078616>. 94, 95, 104
- [40] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010. 18, 37
- [41] Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 2010. 13, 23
- [42] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009. 12, 17
- [43] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 2010. 3, 12, 13, 17, 19, 31
- [44] Rui M. Castro and Robert D. Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 2008. 6
- [45] Rui M Castro, Mark J Coates, Gang Liang, Robert Nowak, and Bin Yu. Network Tomography: Recent Developments. *Statistical Science*, 2004. 8, 69
- [46] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 2012. 93, 95, 96
- [47] Moses Charikar, Liadan O’Callaghan, and Rina Panigrahy. Better streaming algorithms for clustering problems. In *STOC*, 2003. 44
- [48] Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial

- Pure Exploration of Multi-Armed Bandits. In *Advances in Neural Information Processing Systems*, 2014. 96
- [49] Wei-Chen Chen. *Phylogenetic Clustering with R package phyclus*, 2010. URL <http://thirteen-01.stat.iastate.edu/snoweye/phyclus/>. 54
- [50] Yudong Chen. Incoherence-optimal matrix completion. *arXiv:1310.0154*, 2013. 12, 17
- [51] Yudong Chen and Jiaming Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv:1402.1267*, 2014. 97
- [52] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Coherent matrix completion. In *International Conference on Machine Learning*, 2014. 13, 17, 19, 20
- [53] Herman Chernoff. *Sequential analysis and optimal design*. Siam, 1972. 7
- [54] Myung Jin Choi, Vincent Y F Tan, Animashree Anandkumar, and Alan S Willsky. Learning Latent Tree Graphical Models. *Journal of Machine Learning Research*, 2011. 72
- [55] Wei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *Artificial Intelligence and Statistics*, 2011. 7
- [56] Mark J. Coates, Rui M. Castro, and Robert D. Nowak. Maximum likelihood network topology identification from edge-based unicast measurements. *ACM SIGMETRICS*, 2002. 8, 69
- [57] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine learning*, 1994. 6
- [58] Ben Cousins and Santosh Vempala. A cubic algorithm for computing gaussian volume. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1215–1228. SIAM, 2014. 100
- [59] Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 96, 106
- [60] Imre Csiszar and János Körner. *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press, 2011. 96
- [61] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A Decentralized Network Coordinate System. In *ACM SIGCOMM*, 2004. 71
- [62] Sanjoy Dasgupta. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 2005. 6
- [63] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in neural information processing systems*, 2005. 2, 6
- [64] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*, 2006. 1
- [65] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 2011. 1
- [66] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *Learning Theory*. Springer, 2005. 6

- [67] Sanjoy Dasgupta, Claire Monteleoni, and Daniel J. Hsu. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2008. 1
- [68] Constantinos Daskalakis, Elchanan Mossel, and Sebastien Roch. Phylogenies without Branch Bounds: Contracting the Short, Pruning the Deep. *SIAM J. on Discrete Mathematics*, 2011. 72
- [69] Hal Daumé Iii, John Langford, and Daniel Marcu. Search-based structured prediction. *Machine learning*, 75(3):297–325, 2009. 44
- [70] Chandler Davis and W. M. Kahan. The Rotation of Eigenvectors by a Perturbation. III. *SIAM Journal on Numerical Analysis*, 1970. 62
- [71] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, 2002. 43
- [72] Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *ACM-SIAM symposium on Discrete algorithm*, 2006. 3
- [73] B Donnet, P Raoult, T Friedman, and M Crovella. Deployment of an Algorithm for Large-Scale Topology Discovery. *IEEE Journal of Selected Areas in Communications, Special Issue on Sampling the Internet*, 2006. 71
- [74] David Donoho and Matan Gavish. Minimax risk of matrix denoising by singular value thresholding. *The Annals of Statistics*, 2014. 93
- [75] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo Algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 2006. 3, 11
- [76] Petros Drineas, Ravi Kannan, and Michael W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 2006. 11
- [77] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 2008. 12, 14
- [78] Miroslav Dudík, Daniel Hsu, Satyen Kale, Nikos Karampatziakis, John Langford, Lev Reyzin, and Tong Zhang. Efficient optimal learning for contextual bandits. In *Uncertainty and Artificial Intelligence*, 2011. 7, 44
- [79] Nick G. Duffield and Francesco Lo Presti. Network tomography from measured end-to-end delay covariance. *IEEE/ACM Transactions on Networking*, 2004. 69
- [80] Nick G. Duffield, Joseph Horowitz, and Francesco Lo Presti. Adaptive multicast topology inference. *IEEE INFOCOM*, 2001. 69
- [81] Nick G. Duffield, Joseph Horowitz, Francesco Lo Presti, and Don Towsley. Multicast topology inference from measured end-to-end loss. *IEEE Transactions on Information Theory*, 2002. 69
- [82] Nick G. Duffield, Francesco Lo Presti, Vern Paxson, and Don Towsley. Network loss tomography using striped unicast probes. *IEEE/ACM Transactions on Networking*, 2006.

- [83] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1936. 3, 11
- [84] Brian Eriksson, Paul Barford, Robert Nowak, and Mark Crovella. Learning network structure from passive measurements. *ACM SIGCOMM onference on Internet measurement*, 2007. 8, 69
- [85] Brian Eriksson, Paul Barford, and Robert D. Nowak. Estimating Hop Distance Between Arbitrary Host Pairs. In *IEEE INFOCOM*, 2009. 71
- [86] Brian Eriksson, Gautam Dasarathy, Paul Barford, and Robert D. Nowak. Toward the Practical Use of Network Tomography for Internet Topology Discovery. *IEEE INFOCOM*, 2010. xiv, 69, 70, 71, 81, 82
- [87] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert D. Nowak. Active Clustering: Robust and Efficient Hierarchical Clustering using Adaptively Selected Similarities. *AISTATS*, 2011. 8, 43, 45, 47, 54
- [88] Joseph Felsenstein. *Inferring Phylogenies*. Sinauer Associates, 2004. 72
- [89] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems*, 2010. 7
- [90] P Francis, S Jamin, C Jin, Y Jin, D Raz, Y Shavitt, and L Zhang. IDMaps: A Global Internet Host Distance Estimation Service. *IEEE/ACM Transactions on Networking*, 2001. 71
- [91] Yoav Freund, H Sebastian Seung, Eli Shamir, and Naftali Tishby. Selective sampling using the query by committee algorithm. *Machine learning*, 1997. 6
- [92] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 2004. 11, 44
- [93] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 2011. 13
- [94] Olivier Gascuel and Mike Steel. Neighbor-Joining Revealed. *Molecular Biology and Evolution*, 2006. 72
- [95] Alex Gittens. The spectral norm error of the naive Nystrom extension. *arXiv:1110.5305*, October 2011. 17
- [96] Daniel Golovin, Andreas Krause, and Debajyoti Ray. Near-optimal bayesian active learning with noisy observations. In *Advances in Neural Information Processing Systems*, 2010. 6
- [97] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, March 2011. 3, 12, 17, 121
- [98] Krishna P. Gummadi, Stefan Saroiu, and Steven D. Gribble. King: Estimating latency between arbitrary internet end hosts. In *SIGCOMM Workshop on Internet measurment*. ACM, 2002. xiv, 39, 71, 74, 82

- [99] M H Gunes and K Sarac. Resolving IP Aliases in Building Traceroute-based Internet Maps. *IEEE/ACM Transactions on Networking*, 2009. 71
- [100] Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012. 14
- [101] Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 2011. 3
- [102] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *International conference on Machine Learning*, 2007. 1, 6
- [103] Steve Hanneke. Teaching dimension and the complexity of active learning. *Conference on Learning Theory*, 2007. 1
- [104] Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 2011. 1
- [105] Steve Hanneke. Theory of Disagreement-Based Active Learning. *Foundations and Trends in Machine Learning*, 2014. 7
- [106] Moritz Hardt. Understanding Alternating Minimization for Matrix Completion. In *Foundations of Computer Science*, 2014. 13, 18
- [107] Jarvis Haupt, Rui Castro, and Robert Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 2011. 1, 3, 7, 14, 93, 94, 96
- [108] Jotun J Hein. An optimal algorithm to reconstruct trees from additive distance data. *Bulletin of Mathematical Biology*, 1989. 69, 72
- [109] Yuri I. Ingster. Minimax testing of nonparametric hypotheses on a distribution density in the  $L_p$  metrics. *Theory of Probability & Its Applications*, 1987. 95
- [110] Yuri I. Ingster and Irina A. Suslina. *Nonparametric goodness-of-fit testing under Gaussian models*. Springer-Verlag, 2003. 95
- [111] Prateek Jain and Sewoong Oh. Provable Tensor Factorization with Missing Data. In *Advances in Neural Information Processing Systems*, 2014. 4, 17
- [112] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM Symposium on Theory of Computing*, 2013. 13, 18
- [113] Rong Jin and Shenghuo Zhu. CUR Algorithm with Incomplete Matrix Observation. *arxiv:1403.5647*, 2014. 13
- [114] Matti Kääriäinen. Active learning in the non-realizable case. In *Algorithmic Learning Theory*, 2006. 2, 6
- [115] David Kauchak and Sanjoy Dasgupta. An Iterative Improvement Procedure for Hierarchical Clustering. In *Advances in Neural Information Processing Systems*, 2004. 55
- [116] Raghunandan H. Keshavan. *Efficient Algorithms for Collaborative Filtering*. PhD thesis, Stanford University, 2012. 13
- [117] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 2010. 13, 23

- [118] Mladen Kolar, Sivaraman Balakrishnan, Alessandro Rinaldo, and Aarti Singh. Minimax Localization of Structural Information in Large Noisy Matrices. *Advances in Neural Information Processing Systems*, 2011. 93, 94, 95, 98, 104
- [119] Vladimir Koltchinskii, Karim Lounici, and Alexandre B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 2011. 13, 14, 22, 23
- [120] Akshay Krishnamurthy and Aarti Singh. Robust multi-source network tomography using selective probes. In *IEEE INFOCOM*, 2012. 5
- [121] Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sampling. *Advances in Neural Information Processing Systems*, 2013. 4
- [122] Akshay Krishnamurthy and Aarti Singh. On the Power of Adaptivity in Matrix Completion and Approximation. *arxiv:1407.3619*, 2014. 4
- [123] Akshay Krishnamurthy, Sivaraman Balakrishnan, Min Xu, and Aarti Singh. Efficient active algorithms for hierarchical clustering. In *International Conference on Machine Learning*, 2012. 5, 94
- [124] Akshay Krishnamurthy, James Sharpnack, and Aarti Singh. Recovering graph-structured activations using adaptive compressive measurements. *arXiv:1305.0213*, 2013. 1, 7, 14
- [125] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 1985. 7
- [126] John Langford and Alina Beygelzimer. Sensitive error correcting output codes. In *Learning Theory*, pages 158–172. Springer, 2005. 44
- [127] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, 2008. 7
- [128] Zongming Ma and Yihong Wu. Computational barriers in minimax submatrix detection. *arXiv:1309.5914*, 2013. 97
- [129] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An information plane for distributed services. In *Symposium on operating systems design and implementation*, 2006. URL <http://iplane.cs.washington.edu>. xiv, 71, 74, 82
- [130] Matthew Malloy and Robert Nowak. Sequential analysis in high-dimensional multiple testing and sparse recovery. *IEEE International Symposium on Information Theory*, 2011. 1, 14
- [131] Tom Mitchell. Noun Phrases in Context 500 Dataset, 2009. URL [http://www.cs.cmu.edu/~tom/10709\\_fall09/RTWdata.html](http://www.cs.cmu.edu/~tom/10709_fall09/RTWdata.html). 54
- [132] Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. Square Deal: Lower Bounds and Improved Relaxations for Tensor Recovery. *arxiv:1307.5870*, 2013. 4, 13, 16
- [133] Mohammad Naghshvar, Tara Javidi, and Kamalika Chaudhuri. Noisy bayesian active learning. In *Allerton Conference on Communication, Control, and Computing*, 2012. 6
- [134] Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted



- matrix completion: optimal bounds with noise. *The Journal of Machine Learning Research*, 2012. [13](#), [14](#), [23](#)
- [135] Jian Ni, Haiyong Xie, Sekhar Tatikonda, and Yang Richard Yang. Efficient and Dynamic Routing Topology Inference From End-to-End Measurements. *IEEE/ACM Transactions on Networking*, 2010. [xiv](#), [69](#), [70](#), [71](#), [72](#), [74](#), [76](#), [77](#), [81](#), [82](#)
- [136] Samet Oymak and Babak Hassibi. Sharp mse bounds for proximal denoising. *arXiv:1305.2714*, 2013. [96](#)
- [137] Judea Pearl and Michael Tarsi. Structuring causal trees. *Journal of Complexity*, 1986. [69](#), [72](#), [76](#), [85](#)
- [138] Trevor J. Pemberton, Mattias Jakobsson, Donald F. Conrad, Graham Coop, Jeffrey D. Wall, Jonathan K. Pritchard, Pragna I. Patel, and Noah A. Rosenberg. Using population mixtures to optimize the utility of genomic databases: linkage disequilibrium and association study design in India. *Annals of human genetics*, 2008. [54](#)
- [139] Larry Peterson, Andy Bavier, Marc E. Fiuczynski, and Steve Muir. Experiences building planetlab. In *Symposium on operating systems design and implementation*, 2006. [82](#)
- [140] Michael Rabbat and Robert D. Nowak. Multiple source, multiple destination network tomography. In *IEEE INFOCOM*, 2004. [72](#)
- [141] Maxim Raginsky and Alexander Rakhlin. Lower bounds for passive and active learning. In *Advances in Neural Information Processing Systems*, 2011. [6](#)
- [142] Venugopalan Ramasubramanian, Dahlia Malkhi, Fabian Kuhn, Mahesh Balakrishnan, Archit Gupta, and Aditya Akella. On the treeness of internet latency and bandwidth. *ACM SIGMETRICS*, 2009. [xiv](#), [70](#), [72](#), [73](#), [74](#), [76](#), [81](#), [82](#)
- [143] Vincent Ranwez and Olivier Gascuel. Quartet-based phylogenetic inference: improvements and limits. *Molecular Biology and Evolution*, 2001. [73](#)
- [144] Benjamin Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 2011. [3](#), [12](#), [17](#)
- [145] Lev Reyzin and Nikhil Srivastava. On the Longest Path Algorithm for Reconstructing Trees from Distance Matrices. *Information Processing Letters*, 2007. [72](#)
- [146] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 1952. [7](#)
- [147] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 2011. [47](#)
- [148] Sam T. Roweis. NIPS Articles 1987-1999, 2002. URL <http://cs.nyu.edu/~roweis/data.html>. [54](#)
- [149] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 2007. [11](#)
- [150] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 2010. [7](#)
- [151] N Saitou and M Nei. The Neighbor-Joining Method: A New Method for Reconstructing

- Phylogenetic Trees. *Molecular Biology and Evolution*, 1987. 72
- [152] Robert J Serfling. Probability inequalities for the sum in sampling without replacement. *The Annals of Statistics*, pages 39–48, 1974. 59
- [153] Burr Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2009. 7
- [154] Ohad Shamir and Naftali Tishby. Spectral Clustering on a Budget. *AISTATS*, 2011. 43, 44, 53
- [155] Claude Elwood Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 1949. 104
- [156] James Sharpnack, Akshay Krishnamurthy, and Aarti Singh. Near-optimal Anomaly Detection in Graphs using Lovasz Extended Scan Statistic. *Advances in Neural Information Processing Systems*, 2013. 95
- [157] Michael Shindler, Adam Meyerson, and Alex Wong. Fast and Accurate k-Means for Large Datasets. In *NIPS*, 2011. 44
- [158] Aarti Singh, Akshay Krishnamurthy, Sivaraman Balakrishnan, and Min Xu. Completion of high-rank ultrametric matrices using selective entries. In *IEEE International Conference on Signal Processing and Communications*, 2012. 14
- [159] N Spring, R Mahajan, and D Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM*, 2002. 71
- [160] Omer Tamuz, Ce Liu, Serge Belongie, Ohad Shamir, and Adam Tauman Kalai. Adaptively learning the crowd kernel. *arXiv:1105.1033*, 2011. 8, 94
- [161] Ervin Tánzos and Rui Castro. Adaptive sensing for estimation of structured sparse signals. *arXiv:1311.7118*, 2013. 1, 7, 14, 94, 95, 96, 97, 98, 103, 104, 105
- [162] Ryota Tomioka and Taiji Suzuki. Convex Tensor Decomposition via Structured Schatten Norm Regularization. In *Advances in Neural Information Processing Systems*, 2013. 13
- [163] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. Estimation of low-rank tensors via convex optimization. *arxiv:1010.0789*, 2010. 13
- [164] Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. Statistical Performance of Convex Tensor Decomposition. In *Advances in Neural Information Processing Systems*, 2011. 13
- [165] Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, August 2011. 121
- [166] Yolanda Tsang, Mehmet Yildiz, Paul Barford, and Robert Nowak. Network radar: tomography from round trip time measurements. In *ACM SIGCOMM*, 2004. 69
- [167] Yehuda Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. *Journal of the American Statistical Association*, 1996. 69
- [168] Konstantin Voevodski, Maria-Florina Balcan, Heiko Röglin, Shang-Hua Teng, and Yu Xia. Active clustering of biological sequences. *The Journal of Machine Learning Research*, March 2012. 43, 44

- [169] Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 1945. [7](#)
- [170] Xiang Wang and Ian Davidson. Active spectral clustering. In *International Conference on Data Mining, ICDM*, 2010. [44](#)
- [171] B Yao, R Viswanathan, F Chang, and D Waddington. Topology Inference in the Presence of Anonymous Routers. In *IEEE INFOCOM*, 2003. [71](#)
- [172] Peter N Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Symposium on Discrete Algorithms*, 1993. [44](#)
- [173] Ming Yuan and Cun-Hui Zhang. On Tensor Completion via Nuclear Norm Minimization. *arxiv:1405.1773*, 2014. [13](#), [16](#), [17](#)
- [174] Chicheng Zhang and Kamalika Chaudhuri. Beyond disagreement-based agnostic active learning. In *Advances in Neural Information Processing Systems*, 2014. [3](#)