Interactive Algorithms for Unsupervised Machine Learning

Akshay Krishnamurthy

October 7

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Aarti Singh (Chair)
Maria Florina Balcan
Barnabás Poczós
Larry Wasserman
Sanjoy Dasgupta (UCSD)
John Langford (Microsoft Research)

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

Abstract

This thesis explores the power of interactivity in unsupervised machine learning problems. Interactive algorithms employ feedback driven measurements to mitigate the cost of data acquisition 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 mechanisms and inference procedures 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. In addition, 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 in a low dimensional subspace, and the matrix approximation problem, where one must approximate the principal components of an arbitrary collection of points. We measure uniformity with the notion of incoherence, which is known to be necessary for non-interactive algorithms, 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 hierarchical clustering from a small number of carefully selected similarity measurements. The algorithm exploits non-uniformity of cluster size by using few measurements to recover larger clusters and then focusing measurements on identifying the smaller structures. In addition to coming with strong statistical and computational guarantees, this algorithm performs well in practice.

Finally we 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.

Throughout we complement our theoretical results with empirical evaluations.

Contents

Chapter 1

Introduction

Interactive learning is a framework for statistical analysis in which the inference procedure interacts with the data acquisition mechanism and makes feedback-driven measurements. This framework, which is also referred to as active learning, adaptive sampling, or adaptive sensing, has become increasing 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 [7, 8, 9, 10, 13, 14, 27, 28, 29, 45, 46, 47] and signal processing problems [5, 49, 60, 63, 76]. 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.

We show that interactive learning offers two distinct advantages. The first is that interactive approaches are particularly powerful when the data exhibits high degrees of non-uniformity. Interactive mechanisms can identify these non-uniformities and focus measurements to accurately capture these aspects of the data. The second is that interactive algorithms are often both theoretically and empirically faster than non-interactive ones. Both of these claims are supported by several examples in this thesis.

1.1 Overview

In this chapter we summarize completed work, proposed work and a timeline for the thesis. In Chapter 2, we begin our study of subspace learning problems by presenting our results on matrix and tensor completion. We address the noisy subspace learning problem in Chapter 3 where we develop an interactive matrix approximation algorithm and characterize its performance. In Chapter 4, we turn to the clustering problem, presenting a recursive spectral algorithm for hierarchical clustering. We study the tradeoff between signal-to-noise ratio and measurement complexity in this setting and compare our approach with non-interactive clustering algorithms. Finally, in Chapter 5, we address the latent tree metric learning problem. We present algorithms for both additive and sparse adversarial noise models and characterize their measurement complexity, statistical performance, and running time. Throughout we present empirical results to complement our theoretical findings.

1.2 Completed Work

1. **Matrix Completion:** We study low rank matrix and tensor completion and propose novel algorithms that employ interactive sampling to obtain strong performance guarantees. Our algorithms

interactively identify entries that are highly informative for learning the column space of the matrix or tensor and, consequently, they succeed even when the row space is highly coherent (non-uniform), in contrast with non-interactive approaches. We show that one can exactly recover a $n \times n$ matrix of rank r from merely $\Omega(nr\log^2(r))$ matrix entries using an algorithm with running time that is linear in the matrix size, n. In addition to significantly relaxing incoherence assumptions, this algorithm nearly matches the best known sample complexity and is the fastest known algorithm for matrix completion. We also show that one can recover an order T tensor using $\Omega(nr^{T-1}T^2\log^2(r))$ entries, a significant improvement on recent non-interactive approaches. We complement our study with simulations that verify our theory and demonstrate the scalability of our algorithms.

2. **Matrix Approximation:** We consider the problem of constructing a low rank approximation to a high-rank input matrix from interactively sampled matrix entries. 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 μ is a coherence 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.

3. **Clustering:** We develop an adaptive sampling procedure for recovering a binary hierarchical clustering from pairwise similarity information. The algorithm runs spectral clustering on a subsampled version of the similarity matrix to resolve the coarse partitions of the hierarchy 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 [4]. Through extensive experimentation, we also demonstrate that this approach is practically appealing.

4. **Metric Learning:** Motivated by work suggesting that packet latencies in a communication network can be well-approximated by tree metrics, we present two algorithms that use selective pairwise distance measurements between peripheral nodes to construct a latent tree whose end-to-end distances approximate those in the network. Our first algorithm accommodates measurements perturbed by additive noise, while our second considers a novel noise model that captures missing measurements and the network's 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.

1.3 Proposed Work

1. **Fundamental limits for passive algorithms:** A characterization of the fundamental limits for non-interactive algorithms in many of the problems introduced earlier has remained unresolved. Such a characterization is essential to demonstrating the power of interactive learning. In matrix completion, we know that passive algorithms have high sample complexity in the absence of row-space incoherence, a setting where adaptive sampling is known to succeed [58]. Similarly, lower bounds against non-interactive uniform sampling are known for matrix approximation problems [55, 66]. However, for hierarchical clustering, latent tree metric learning, and matrix approximation with other sampling distributions, these questions are still open. We aim to establish these fundamental limits for non-interactive algorithms, bolstering the case for interactive ones.

2. **Deeper analysis of interactive clustering:** There are many opportunities for improvements and

extensions to our work on interactive clustering. Our current analysis is restricted to binary hierarchies and fairly balanced cluster sizes at each level of the hierarchy. One avenue to eliminate these restrictions is to recursively apply an algorithm for k-way clustering that succeeds in the presence of non-uniform cluster sizes at each level of the hierarchy. Recently, a peeling-style algorithm that finds large clusters, removes them, and then focuses on finding smaller ones, has been shown to successfully recover non-uniform k-way clusterings in the graph clustering setting [1]. We conjecture that a similar algorithm will succeed in the noisy similarity setting. We aim to apply this algorithm to handle non-binary hierarchies with non-uniform cluster sizes.

3. **General purpose algorithms for Matrix Approximation and Clustering:** Our current approach for matrix approximation is not fully harnessing the power of adaptive sampling, and we conjecture that there is room for substantial improvements. Specifically, while the scaling between the number of measurements, the target rank, and the problem size match related results, the dependence on the error tolerance is polynomially worse. The algorithm only uses two rounds of measurement, and we suspect that one can achieve better performance with more interaction. We propose an algorithm that first approximates the leading direction of the matrix and then iteratively focuses measurements on the residual to capture the remaining directions.

   Interestingly, the algorithm bears striking similarity to our proposal for the hierarchical and flat clustering problems described before. We therefore aim to analyze this algorithm in both the matrix approximation and the clustering settings.

4. **Adaptive Compressive Matrix Approximation:** It is also natural to consider the compressive version of the matrix approximation problem. Here, the matrix is observed through a sequence of (interactively) chosen linear measurements and as before, the goal is to compete with the best low-rank approximation. We have analyzed a non-interactive algorithm that works best when the columns of the matrix all have similar norm [61]. As in previous examples, we aim to show that this uniformity assumption can be relaxed via adaptive sampling.

1.4 Exploratory Work

1. **Lower bounds for interactive algorithms:** While the construction of an adaptive algorithm, coupled with a lower bound against non-interactive algorithms, suffices to establish the advantages of adaptivity, it is also interesting to understand the fundamental limits of this framework. Results of this flavor would not only certify optimality or sub-optimality of our algorithms, but would provide a comprehensive characterization of interactive algorithms for unsupervised learning. While lower bounds for adaptive algorithms have been established for some signal processing problems [2], bringing these to unsupervised learning appears to be challenging.

2. **Interactive Approaches in other metric learning problems** Our work on metric learning considers one particular setting, namely that of recovering a latent tree metric, which is inspired by network tomography applications. However, there is a wide class of other metric structures, such as euclidean or other finite-dimensional ℓ_p metrics and geodesics on smooth manifolds, for which our techniques are not applicable. It is well known that subsampling approaches can yield good approximations for some of these problems, and therefore it seems interesting to explore interactive approaches [30]. Understanding the noise tolerance of both passive and interactive algorithms for these problems is another direction for future research.

1.5 Thesis Statement

In this thesis, I aim to demonstrate the statistical and computational power of interactivity in unsupervised learning with specific focus on settings with high degrees of non-uniformity. Formally,

“Interactive data acquisition facilitates statistically and computationally efficient unsupervised learning algorithms that are particularly well-suited to handle non-uniform datasets.”

1.6 Timeline

Below is a timeline for the completion of my proposed work. I plan to graduate in the Summer of 2015.

1. **Fall 2014:** Thesis Proposal.
2. **Fall 2014:** Analysis of improved matrix approximation algorithm. Conference submission or improvements to existing JMLR submission.
3. **Fall 2014:** Lower bounds for matrix approximation.
4. **Spring 2015:** Interactive flat clustering.
5. **Spring 2015:** Adaptive compressive matrix approximation. Journal submission.
6. **Summer 2015:** Thesis writing and defense.

1.7 Acknowledgements

The work in this document is in collaboration with my advisor, Aarti Singh, and peers, Martin Azizyan, Sivaraman Balakrishnan, and Min Xu. I am also grateful to the National Science Foundation, who supported this work through an NSF Graduate Research Fellowship.

Chapter 2

Matrix and Tensor Completion

2.1 Introduction and Related Work

In the matrix completion problem, we would like to recover a low rank matrix after observing only a small fraction of its entries. In this chapter, we propose interactive algorithms for low rank matrix completion and the closely-related tensor completion problems.

Our study is motivated not only by prior theoretical results in favor of adaptive sensing but also by several applications where adaptive sensing is feasible. In recommender systems, obtaining a measurement amounts to asking a user about an item, an interaction that has been deployed in production systems. Another application pertains to network tomography, where a network operator is interested in inferring latencies between hosts in a communication network while injecting a few packets into the network. The operator, being in control of the network, can adaptively sample the matrix of pair-wise latencies, potentially reducing the total number of measurements.

A theme of this work is that interactive or adaptive sampling allows one to relax incoherence assumptions pervasive in the matrix completion literature. Previous analyses show that if the energy of the matrix is spread out fairly uniformly across its coordinates, then passive uniform-at-random samples suffice for completion. In contrast, our work shows that adaptive 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 or highly active users. This is a setting where passive algorithms provably fail, as we show.

Due to its widespread applicability, the matrix completion problem has received considerable attention in recent years. A series of papers [20, 21, 24, 42, 72] establish that $\Omega(nr\mu_0 \log^2(n))$ randomly drawn samples are sufficient for the nuclear norm minimization program to exactly identify an $n \times n$ matrix with rank r. Here μ_0 is the coherence parameter, which measures the uniformity of the row and column spaces of the matrix. Candès and Tao [21] show that nuclear norm minimization is essentially optimal with a $\Omega(nr\mu_0 \log(n))$ lower bound for uniform-at-random sampling.

There is also a line of work analyzing alternating minimization-style procedures for the matrix completion problem [48, 50, 53]. While the alternating minimization algorithm is computationally more elegant than nuclear norm minimization, 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 [53]. 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.* [25] 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 [51] 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. 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 [41, 65, 77, 78, 79, 82]. Of these, only Mu *et al.* [65] and Yuan and Zhang [82] provide sample complexity bounds for the noiseless setting. Mu *et al.* [65] show that $\Omega(rn^{T/2})$ random linear measurements suffice to recover a rank r order-T tensor. Yuan and Zhang [82] 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.

In this chapter 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 $\Omega(nr\mu_0 \log^2(r))$ measurements where μ_0 is the coherence parameter on the column space of the matrix. This algorithm outperforms all existing results on matrix completion both in terms of sample complexity (with the exception of [51]) and in the fact that we place no assumptions on the row space of the matrix. The algorithm is extremely simple, runs in $O(nr^2)$ time, and can be implemented in one pass over the columns of the matrix.

2. We complement this sufficient condition with a lower bound showing that in the absence of row-space incoherence, *any* passive scheme must see $\Omega(n^2)$ entries. This concretely demonstrates the power of adaptivity in the matrix completion problem.

3. For the tensor completion problem, we establish that $\Omega(nr^{T-1}T^2 \log r)$ adaptively chosen samples are sufficient for recovering a $n \times \ldots \times n$ order T tensor of rank r.

2.2 Main Results

Before proceeding, we establish some notation. In the matrix completion problem we are interested in recovering, a $n \times n$ matrix X of rank at most r from a set of at most M observations[1]. We focus on the $0/1$ loss; given an estimator $\hat{X}$ for X, we would like to bound the probability of error:

$$R_{01}(\hat{X}) \triangleq \mathbb{P}\left(\hat{X} \neq X\right). \tag{2.1}$$

Apart from the observation budget M and the rank r, the other main quantity governing the difficulty of this problem is the subspace coherence parameter. For an r dimensional subspace U of $\mathbb{R}^n$, define

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

which is the standard measure of subspace coherence [72]. The quantity $\mu(U)$, which is bounded between 1 and n/r, measures how correlated the subspace U is with any single standard basis element. If U is the column space of X and $\mu_0 \triangleq \mu(U)$ is small, then 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 will see that the parameter μ_0 controls the sample complexity of our adaptive procedure.

[1] All results also apply to non-square matrices, with appropriate modifications.

The definitions translate naturally to the tensor setting. Let $\mathbb{M} \in \mathbb{R}^{n \times \dots \times n}$ denote an order T tensor that can be decomposed into a sum of r rank one tensors The mode-t sub-tensors of $\mathbb{M}$, denoted by $\mathbb{M}_i^{(t)}$, are order $T-1$ tensors obtained by fixing the i-th coordinate of the t-th mode. For example, if $\mathbb{M}$ is an order 3 tensor, then $\mathbb{M}_i^{(3)}$ are the frontal slices. The subspaces associated with the tensor are denote $A^{(t)}$ which is the span of the mode-t fibers (i.e. columns, rows, etc.).

Equipped with these definitions, we now describe our interactive matrix completion algorithm. The procedure 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 the orthogonal projection $\mathcal{P}_{U^\perp} x_t$ from a subsampled version of x_t. 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, the algorithm can complete the column by solving an over-determined linear system.

The main ingredient of the algorithm is the estimator for the quantity $\|\mathcal{P}_{U^\perp} x\|_2^2$ when the vector x is partially observed. If x is subsampled to the coordinates $\Omega \subset [n]$ of size m, denoted by x_Ω, then we also subsample the rows of an orthonormal basis for U to form U_Ω, and estimate with $\|(I - U_\Omega (U_\Omega^T U_\Omega)^\dagger U_\Omega^T) x_\Omega\|_2^2$. By analyzing this estimator and the recovery procedure, we are able to prove the following theorem characterizing the statistical and computational performance of the algorithm [2]:

Theorem 1. *Let $X \in \mathbb{R}^{n \times n}$ be a matrix of rank r whose column space U has coherence $\mu(U) \leq \mu_0$. Then the interactive matrix completion algorithm, when sampling m entries per column, has risk:*

$$R_{01}(\hat{X}) \leq 10r^2 \exp\left\{-\sqrt{\frac{m}{32r\mu_0}}\right\} \tag{2.2}$$

provided that $m \geq 4r\mu_0 \log(2r/\delta)$. Equivalently, whenever $m \geq 32r\mu_0 \log^2(10r^2/\delta)$, we have $R_{01}(\hat{X}) \leq \delta$. The sample complexity is $nr + nm$ and the running time is $O(nmr + nr^2 + r^3 m)$.

In other words, our algorithm exactly recovers the input matrix using $\Omega(nr\mu_0 \log^2(r))$ samples and in $O(n\text{poly}(r))$ time. To the best of our knowledge, this result provides not only the best sample complexity (with the exception of [51]), but also the best computational complexity for the matrix completion problem. Moreover, this algorithm does not require row-space incoherence, in contrast with all existing passive approaches. As incoherence is a measure of data uniformity, this supports all facets of our thesis.

In fact, row-space incoherence is necessary for passive sampling to achieve non-trivial sample complexity bounds as shown in the following result:

Theorem 2 (Informal). *For any passive sampling strategy, if $\mu_0 \geq c > 1$, then $M = \Omega((n^2 - nr))$ samples are necessary to recover an $n \times n$ matrix of rank r and column incoherence bounded by μ_0.*

This theorem, coupled with Theorem 1, shows strong separation between interactive and non-interactive approaches for matrix completion. The proof is based on minimax theory, where for any sampling strategy, we consider the probability of error on the worst case input matrix for that strategy. This minimax risk is lower bounded by placing a uniform distribution on a specific family of input matrices and then by counting the number of matrices that are indistinguishable from a given set of samples.

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 sub-tensors $\mathbb{M}_i^{(T)}$. For each of these sub-tensors, we test whether $\mathbb{M}_i^{(T)}$ lives in $\mathcal{U}$ and recursively complete that sub-tensor if it does not. Once we complete the sub-tensor, we add it to $\mathcal{U}$ and proceed at the outer level. When the sub-tensor itself is just a column; we observe the columns in its entirety. We are able to establish the following performance guarantee:

Theorem 3 (Informal). *Suppose that all but the last subspace $A^{(T)}$ have coherence bounded above by μ_0. Then the recursive algorithm recovers a rank r order-T tensor using $O(T^2 n (r\mu_0)^{T-1} \log^2(Tr))$ samples.*

[2] All proofs and algorithm details are presented in the original publications [57, 58].

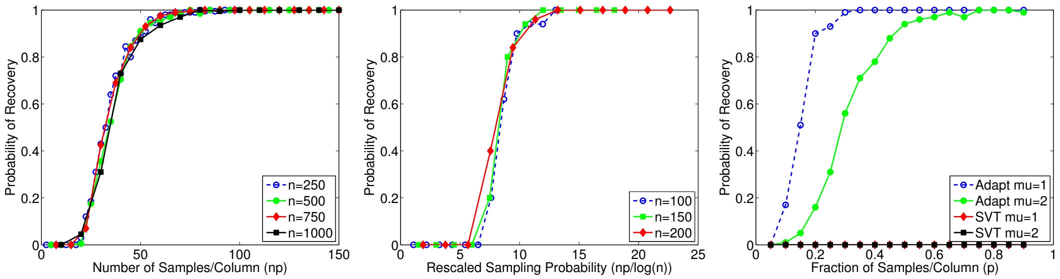


Figure 2.1: Left: Probability of recovery for the interactive MC algorithm as a function of the number of samples per column. Center: Probability of recovery for the SVT algorithm as a function of rescaled sampling probability $np/\log n$. Right: Probability of recovery for both SVT and the interactive algorithm on matrices with coherent row space.

2.2.1 Simulations

We now turn to empirical results. In Figure 2.1 we present some simulations comparing our algorithm with the Singular Value Thresholding algorithm of [18]. In the first plot, we record the probability of exact recovery as a function of the number of samples per column $m = np$ for matrices of varying size n. The fact that these curves line up demonstrates that sample complexity scales linearly with problem size, so that the number of samples per column remains constant. On the other hand, for the SVT algorithm (second plot), one must instead plot the success probability as a function of $np/\log(n)$ for the curves to line up, which suggests that the sample complexity for this algorithm scales with $n\log(n)$ (ignoring other parameters). In the last plot, we record the success probability versus sampling probability on matrices with maximally coherent row spaces. The results of this simulation clearly demonstrate that our algorithm can tolerate coherent row spaces while SVT cannot.

To confirm the computational improvements, we ran our matrix completion algorithm on large-scale matrices and compared with SVT [3]. As a concrete example, recovering a 10000×10000 matrix of rank 100 takes close to 2 hours with SVT, while it takes less than 5 minutes with our algorithm.

2.3 Future Work

There are two important avenues for future work. The first is to develop a more practical interactive sampling algorithm. It is often not tractable to observe a column in its entirety, and this limits the applicability of our current algorithm. While we have established the power of interactivity for matrix completion, it would be interesting to design more practical procedures that retain the favorable statistical and computational properties of our algorithm.

Another worthwhile direction is to address the question of optimality for the tensor completion problem. On one hand, it seems possible to adapt the proof of Theorem 2 to the tensor case to establish a lower bound against passive algorithms. However, since a rank r order T tensor has nrT parameters, it seems likely that our adaptive algorithm (Theorem 3) has suboptimal dependence on both r and T, and it would be interesting to understand whether this is fundamental or if it can be addressed with a better algorithm.

[3] See Table 1 of [57] and Table 5.1 of [18].

Chapter 3

Matrix Approximation

3.1 Introduction and Related Work

In the matrix approximation problem, we aim to to find a low rank matrix that approximates, in a precise sense, the input, which need not be low rank. This generalizes the matrix completion problem discussed in Chapter 2. In particular, this setting encompasses the noisy low rank matrix completion problem which has received considerable attention in recent years [19, 55, 66].

Computing a low rank approximation to a given matrix, more commonly referred to as principal components analysis, is a fundamental preprocessing tool in scientific applications. In many such applications, each entry of the data matrix corresponds to the outcome of an experiment or measurement process. For example, in biological applications an entry may record the effect of a drug on a particular protein. In these settings, one can leverage interactivity to guide the sequence of experiments, and our results show that one can make significantly fewer measurements with little loss in the quality of approximation.

In this chapter, we analyze an algorithm that, after an adaptive 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 $n \times n$ input matrix. Here μ is a coherence parameter on each column of the matrix; as in Chapter 2 we make no assumptions about the row space of the input. By eliminating this assumption, this algorithm significantly outperforms existing results on matrix approximation from passively collected samples.

Existing work on matrix approximation with missing data has focused on passively collected samples. These methods rely on measures of uniformity on both row and column spaces, whether it be incoherence [19, 54], spikiness [66], or a boundedness assumption [55]. In comparison, our adaptive algorithm achieves low error even on matrices with row spaces that violate these assumptions, a setting where existing passive algorithms fail.

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 [16, 17, 33]. 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 high-quality estimate seems difficult to obtain in the missing data setting. This difficulty also arises with volume sampling [44], 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.

3.2 Main Results

In this section, we highlight our main results for the matrix approximation problem. Given a $n \times n$ matrix X, we are interested in approximating the action of X_r, the matrix formed by zero-ing out all but the largest r singular values of X. Specifically, we are interested in optimizing the risk $R(\hat{X}) = \|X - \hat{X}\|_F$ and aim to achieve excess risk bounds of the form:

$$R(\hat{X}) \triangleq \|X - \hat{X}\|_F \leq \|X - X_r\|_F + \epsilon \|X\|_F, \tag{3.1}$$

under the constraint that $\hat{X}$ has rank at most r. Rescaling the excess risk term by $\|X\|_F$ is a form of normalization that has been used before in the matrix approximation literature [31, 32, 40, 74]. This bound 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.

We parameterize the problem by a quantity related to the usual definition of incoherence:

$$\mu = \max_{t \in [n]} \frac{n||x_t||_\infty^2}{||x||_2^2}, \tag{3.2}$$

which is the maximal column coherence. It will be important that μ is sufficiently small. We make no assumptions about the row space of the matrix.

Our algorithm for matrix approximation 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 $\Omega_{2,t} \subset [n]$ from each column, and for each t, places the rescaled zero-filled vector $\mathcal{R}_{\Omega_{2,t}} x_t = \sum_{j=1}^n \frac{n}{|\Omega_{2,t}|} x_t(j) \mathbf{1}[j \in \Omega_{2,t}]$ 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 the final estimate $\hat{X}$.

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 these norms. 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.

For the main performance guarantee, we only assume that the matrix has incoherent columns as defined in Equation 3.2. We have the following theorem [1]:

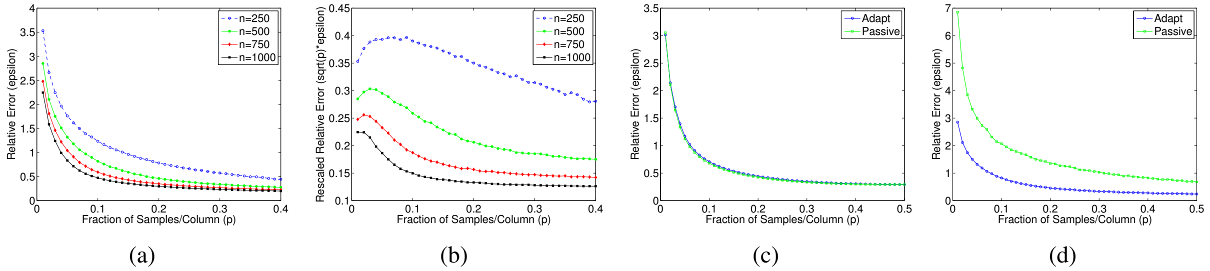
Theorem 4. *Let X be an $n \times n$ matrix and define μ as in Equation 3.2. In the first pass observe $m_1 \geq 32\mu \log(n/\delta)$ entries from each column and in the second pass observe m_2 entries per column on average. With probability $\geq 1 - 2\delta$, the adaptive algorithm computes an approximation $\hat{X}$ such that:*

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

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:

$$32n\mu \log(n/\delta) + \frac{576}{\epsilon^4} nr\mu \log^2\left(\frac{2n}{\delta}\right). \tag{3.3}$$

[1] See [58] for algorithm details and proofs.



(a) (b) (c) (d)

Figure 3.1: (a): Relative error of the adaptive algorithm 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 adaptive and passive sampling on matrices with uniform column lengths (column coherence $\mu = 1$ and column norms are uniform from $[0.9, 1.1]$). (c): Relative error for adaptive and passive sampling on matrices with highly non-uniform column lengths (column coherence $\mu = 1$ and column norms are from a standard Log-Normal distribution).

The theorem shows that the matrix $\hat{X}$ serves as nearly as good an approximation to X as X_r. Specifically, with $O(nr\mu \log^2(n))$ observations, one can compute a suitable approximation to X.

The closest result to Theorem 4 is the result of Koltchinskii *et al.* [55] who consider a soft-thresholding procedure and bound the approximation error in squared-Frobenius norm. They assume that the matrix has bounded entrywise ℓ_∞ norm and give an entrywise squared-error guarantee of the form:

$$\|\hat{X} - X\|_F^2 \le \|X - X_r\|_F^2 + cn^2\|X\|_\infty^2 \frac{nr\log(n)}{M} \tag{3.4}$$

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 $n^2\|X\|_\infty^2 \ge \|X\|_F^2$, their bound is significantly worse if 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 $\frac{1}{n}$ with the matrix where one column has all entries equal to $\frac{1}{\sqrt{n}}$. In the former, Koltchinskii *et al.* bound the squared-Frobenius error by $nr\log(n)/M$ 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 $n^2 r\log(n)/M$ while our bound remains the same. Thus our algorithm is particularly suited to handle matrices with non-uniform column norms.

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

The main computational bottleneck of the algorithm involves obtaining the leading singular vectors of the zero-filled matrix, which also governs the running time of the passive algorithm [55]. Thus our interactive algorithm does not exhibit computational gains over non-interactive approaches. However, the result and the above discussion does support our claim that interactive algorithms are particularly powerful for non-uniform datasets.

3.2.1 Simulations

We now mention some empirical evaluations. In Figure 3.1(a), we plot the relative error, which is the ϵ in Equation 3.1, 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 3.1(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 $\sqrt{p}$ flattens out all of the curves, which suggests that the relationship between ϵ and the number of samples is indeed $\epsilon \asymp \frac{1}{\sqrt{p}}$. This scaling is actually better than the dependence predicted by Theorem 4, but can be explained by specializations of the general result [2].

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 3.1(c), we use matrices with uniform column norms, and observe that both algorithms perform comparably. However, in Figure 3.1(d), when the column norms are highly non-uniform, we see that the adaptive algorithm dramatically outperforms the passive sampling approach. This confirms our claim that adaptive sampling leads to better approximation when the energy of the matrix is not uniformly distributed.

3.3 Future Work

There are three main lines of proposed work. First, the excess risk bounds like Equation 3.1 are undesirable for the matrix approximation problem because they can be quite weak when most of the energy of the matrix is concentrated in the top ranks. Instead one would prefer **relative error** bounds of the form:

$$\|X - \hat{X}\|_F \leq (1+\epsilon)\|X - X_r\|_F \tag{3.5}$$

While there are computationally efficient algorithms to achieve this form of guarantee without computing the SVD, the techniques used do not seem to apply to the missing data setting, even with interactivity. Nevertheless, there may be room for improvement over Theorem 4, as the algorithm is not harnessing the full power of interactivity. We suspect that one can achieve better performance with more rounds of interaction, and would like to analyze an algorithm of this form.

The second direction involves establishing lower bounds against passive algorithms. Again here some results are known [55, 66], but they only consider uniform-at-random sampling models and also stochastic noise. We would like to establish lower bounds against *all* non-interactive sampling distributions for the matrix approximation problem.

Finally, we would like to study extensions to the adaptive compressive matrix approximation problem, where the matrix is observed through a sequence of random projections. Here we believe that one can again eliminate uniformity assumptions used in non-interactive algorithms for this problem [61].

[2] See Proposition 5 in [58].

Chapter 4

Clustering

4.1 Introduction and Related Work

Clustering, an ubiquitous task in exploratory data analysis, data mining, and several application domains, involves 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 a hierarchical clustering, the groups have multiple resolutions, so that a large cluster may be recursively divided into smaller sub-clusters. There exist many effective algorithms for clustering, but as modern data sets get larger, the fact that these algorithms require *every* pairwise similarity between objects poses a serious measurement and/or computational burden and limits the practicality of these algorithms. It is therefore appealing to develop effective clustering algorithms with low measurement and computational overhead.

To achieve both measurement and computational improvements, we focus on using interactivity to reduce the number of similarity measurements required for clustering. This approach results in immediate reduction in measurement overhead 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. In the former, our algorithm injects fewer packets than existing techniques, and in the latter our algorithm is dramatically faster than popular algorithms.

In this chapter, we propose a novel interactive hierarchical clustering algorithm based on spectral clustering. Spectral clustering is a very popular family of algorithms that relies on the structure of the eigenvectors of the Laplacian of the similarity matrix. These algorithms have received considerable attention in recent years due to its empirical success, but they suffers from the fact that they require all $n(n-1)/2$ similarities between the n objects to be clustered and must compute a spectral decomposition, which can be computationally prohibitive on large datasets. Our adaptive algorithm avoids both limitations by subsampling few objects in each round and only computing eigenvectors of very small submatrices. By appealing to previous statistical guarantees [4], we can show that this algorithm has desirable theoretical properties, both in terms of statistical and computational performance.

While there is a large body of work on hierarchical and partitional clustering, only a few algorithms attempt to minimize the number of pairwise similarities used [11, 39, 75]. Along this line, the work of Eriksson et. al. [39] and Shamir and Tishby [75] is closest in flavor to ours.

Algorithm 1 `ActiveSpectral`$(s, \{x_i\}_{i=1}^n)$

```
if $n \leq s$ then return $\{x_i\}_{i=1}^n$
Draw $S \subseteq \{x_i\}_{i=1}^n$ of size $s$ u.a.r.
$C'_l, C'_r \leftarrow$ SpectralCluster(S).
Set $C_l \leftarrow C'_l, C_r \leftarrow C'_r$.
for $x_i \in \{x_i\}_{i=1}^n \setminus S$ do
    $\alpha_l \leftarrow \frac{1}{|C'_l|} \sum_{x_j \in C'_l} K(x_i, x_j)$ and analogously for $\alpha_r$.
    If $\alpha_l > \alpha_r$, add $x_i$ to $C_l$, else add to $C_r$.
end for
output $\{C_l, C_r,$ ActiveSpectral$(s, C_l)\}$
```

Algorithm 2 `SpectralCluster`$(\{x_i\}_{i=1}^n)$

```
if $n \leq 1$ then return $\{x_i\}_{i=1}^n$
Compute pairwise similarity matrix $W \in \mathbb{R}^{n \times n}$ where $W_{ij} = K(x_i, x_j)$.
Compute Laplacian matrix $L = D - W$, $D_{ii} = \sum_{j=1}^n W_{ij}$.
$v_2 \leftarrow$ smallest non-constant eigenvector of $L$.
$C_l \leftarrow \{i : v_2(i) \geq 0\}, C_r \leftarrow \{j : v_2(j) < 0\}$.
output $\{C_l, C_r\}$.
```

Eriksson et. al. [39] develop an adaptive algorithm for hierarchical clustering and analyze the correctness and measurement complexity of this algorithm under a noise model where a small fraction of the similarities are inconsistent with the hierarchy. Our analysis 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. Shamir and Tishby [75] analyze a binary spectral algorithm based on randomly subsampling similarities but they require $\Omega(n^2)$ similarities to perfectly recover a two-way flat clustering. Our work, translated to their setting improves this guarantee; Theorem 6 implies that our algorithm only needs $\Omega(n \log n)$ similarities. 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.

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 [12], and via split or merge requests of an existing clustering [3, 6]. 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.

4.2 Main Results

We focus on binary hierarchical clusterings over n objects defined as:

Definition 5. *A* ***binary 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 each non-terminal cluster* $C \in \mathcal{C}$*, there exists two cluster* $C_l, C_r \subset C$ *such that* $C_l \cup C_r = C$*.*

Our algorithm for active spectral clustering is displayed in Algorithm 1 alongside pseudocode for one variant of spectral clustering that we will use. To recover a single split of the hierarchy, the algorithm subsamples s points and only uses similarities to these landmark points. The first step of the algorithm is to cluster the landmarks using the spectral clustering algorithm in Algorithm 2. 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 landmark objects.

By recursively applying this procedure to each cluster, we obtain a hierarchical clustering. Since in this recursive phase we do not observe measurements between clusters at the previous split, this results in an interactive algorithm that focuses its measurements to resolve the fine-grained cluster structure.

We analyze the algorithm under the **noisy Hierarchical Block Model** (noisy HBM) used in previous work [4, 59]. At a high level, this model adds subgaussian perturbation to an ideal similarity matrix for which within cluster similarities are larger than between cluster similarities. We have the following theorem characterizing the performance of the algorithm:

Theorem 6. *If the noise variance and balance factor are both constant, then the* `ActiveSpectral` *algorithm, when run on a noisy hierarchical block model, succeeds in recovering all clusters of size s with probability $1 - o(1)$ provided that $s = \Omega(\log n)$. The algorithm uses $O(ns \log n)$ measurements and runs in $O(ns^2 \log s + ns \log n)$ time.*

There are several tradeoffs worth mentioning here. To capture the tradeoff between the noise level and measurement overhead, it is best to consider recovering only clusters of size $\Omega(n)$. Our theorem states that `ActiveSpectral` can tolerate a constant amount of noise while using only $O(n \log^2 n)$ measurements. On the other hand, the result of Balakrishnan et. al. [4] shows that using $O(n^2)$ measurements, one can tolerate a noise level of $\sqrt{n/\log n}$. Varying s allows for interpolation between these two extremes.

The other tradeoff is between the noise level and the size of the smallest cluster recoverable, where one should consider constant noise level. In this setting, the non-interactive algorithm recovers clusters of size $\Omega(\log n)$ but uses $O(n^2)$ similarities. In contrast, our algorithm achieves the same statistical performance but uses only $O(n \log^2 n)$ measurements. Note that this setting exhibits highly non-uniform cluster sizes (ranging from $\Omega(n)$ to $\Omega(\log n)$), showing that interactivity is particularly powerful in the presence of such non-uniformity.

Setting $s = \Theta(\log n)$ which suffices to tolerate a constant noise level, our algorithm runs in $O(n\text{polylog}(n))$. In contrast, the algorithm that operates on the fully-observed similarity matrix [4] computes eigenvectors of a $n \times n$ matrix and has at least quadratic computational complexity. This speedup is dramatically magnified in settings where evaluating pairwise similarity is computationally demanding. Consequently, the interactive algorithm is computationally very appealing.

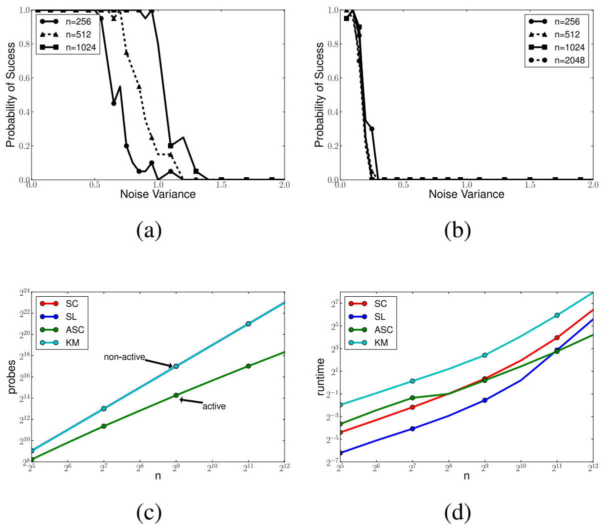
4.2.1 Experiments

We now present some experimental results. In Figure 4.1 we plot the results of several simulations using the interactive and non-interactive spectral clustering algorithms. By Theorem 6 and the above discussion, we expect `ActiveSpectral` to recover all splits of size $\Omega(n)$ in the presence of a constant amount of noise, and we expect the non-interactive spectral algorithm [4] to tolerate noise growing with n at rate $\sigma \asymp \sqrt{n/\log n}$. We contrast these guarantees by plotting the probability of successful recovery of the first split in a noisy HBM as a function of noise variance for different n in Figures 4.1(a) and 4.1(b). The first figure demonstrates that indeed the noise tolerance of the non-interactive algorithm grows with n while the second demonstrates that `ActiveSpectral` enjoys constant noise tolerance.

We next verify the measurement and run time complexity guarantees for `ActiveSpectral` in comparison with three passive clustering algorithms. In Figure 4.1(c) and 4.1(d), we plot the number of measurements and running time as a function of n on a log-log plot for each algorithm. The passive algorithms have steeper slopes, suggesting that they are polynomially more expensive in both cases.

In Table 4.1(e) we record the results from evaluating our adaptive algorithm and two popular passive algorithms on four datasets: the set of articles from NIPS volumes 0 through 12 from [73], a subset of NPIC500 co-occurence data from the Read-the-Web project [64] which we call RTW, a SNP dataset from the HGDP [69], and a synthetic phylogeny dataset produced using `phyclust` [23]. Since the reference clustering is not available in all of these datasets [1], we employ two distinct metrics to evaluate

[1] It is available in Phylo and SNP datasets, see [59] for more experimental results



(a)

(b)

(c)

(d)

Algorithm	HKM	HRC	Probes	Time (s)
	SNP			
ACTIVESPEC	0.019	19.1	0.13	450
k-means	0.0028	18.7	1	160
Spectral	0.0075	130	1	5660
	Phylo			
ACTIVESPEC	0.012	22.9	0.071	600
k-means	0.0017	22.9	1	967
Spectral	0.0022	23.5	1	997
	NIPS			
ACTIVESPEC	0.010	1.5	0.094	79.4
k-means	0.0017	1.66	1	723
Spectral	0.0033	6.30	1	26200
	RTW			
ACTIVESPEC	0.0084	0.64	0.13	151

(e)

Figure 4.1: Figure (a) and (b): Noise thresholds for spectral and active spectral clustering. Figure (c): Measurement complexity for interactive and non-interactive algorithms. Figure (d): Running time for interactive and non-interactive algorithms. Table (e): Real world experiments.

the quality of hierarchical clusterings. They are a hierarchical K-means objective (HKM) [52] 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. Good hierarchical clusterings minimize both of these object values.

In Table 4.1(e), we record experimental results across the datasets for `ActiveSpectral`, the hierarchical spectral clustering algorithm of [4], and a natural hierarchical extension of Lloyd's algorithm for k-means clustering. On the read-the-web dataset, we were unable to run the non-interactive algorithms. The immediate observation is that `ActiveSpectral` is extremely fast; on the SNP and phylogeny datasets where computing similarities is the bottleneck, interactivity leads to significant computational improvements. Moreover, the algorithm performs fairly well according to the HKM and HRC metrics although it is worse than the non-interactive algorithms. We believe that this gives empirical evidence in favor of our algorithm; it allows one to find a suitable tradeoff between robustness on one hand and measurement and computational efficiency on the other.

4.3 Future Work

There are two main avenues for future work. The first is to establish lower bounds against passive algorithms for hierarchical clustering. We already have lower bounds in the fully observed setting, but it is important to introduce a measurement budget to better compare with interactive approaches.

The second is an extension to flat clusterings with non-uniform sizes. Recently, a peeling technique was shown to succeed in the graph clustering setting with non-uniform cluster sizes [1] and we would like to extend this to the noisy block model setting. It would also be interesting to combine the flat and hierarchical algorithms yielding a general algorithm for interactive clustering.

Chapter 5

Learning Latent Tree Metrics

5.1 Introduction and Related Work

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 for todays incredibly large and decentralized networks, these global properties are not directly available, but must be inferred from indirect measurements. Network tomography [22, 81] is a promising approach that aims to gather such knowledge using only end-to-end measurements between nodes at the periphery of a network without cooperation from core routers. In this chapter, we contribute to this important direction with two algorithms that accurately recover network characteristics from end-to-end measurements.

Given the size and complexity of the Internet, the practicality of any network tomography algorithm should be evaluated by its noise tolerance and robustness to violations of any modeling assumptions, as well as its measurement complexity. State-of-the-art methods typically suffer in at least one of these directions. Some methods do not optimize and/or provide rigorous guarantees on the number of measurements needed, while others do not guarantee robustness to noise. We consider both noise tolerance and measurement overhead and provide algorithms with rigorous guarantees along both axes.

Our work falls into the category of *multi-source* tomography, where measurements can be obtained between any pair of end hosts, rather than *single-source* tomography, where all measurements are initiated from a single host. Multi-source tomography offers the practical advantage of using extremely simple measurements such as hop counts or latencies, while single-source tomography relies on infrequently deployed multicast probes ([15, 34, 35, 36]) or complex packet sequences ([26, 37, 38, 67, 80]) to obtain similarity information. However, since the set of links traversed by packets emanating from a single host form a tree structure, single-source methods focus on inferring tree topologies, while multi-source methods face the challenge of recovering more general graph structures.

Motivated by recent work [71] showing that internet latency and bandwidth can be well approximated by path lengths on trees, our algorithms are designed to construct tree topologies and consequently tree metrics. However, we introduce two models to capture violations of the tree-metric assumption: (a) an *additive noise* model, where all measurements are corrupted by additive subgaussian noise, resulting in small deviations from the tree metric properties, and (b) a *persistent noise* model in which a fraction of the measurements are arbitrarily corrupted. Even under these noise models, our algorithms have provable guarantees about correctness and measurements complexity. Empirically, they perform well on multi-source network tomography datasets which do not satisfy the tree-metric assumption.

Our algorithms, and indeed several existing methods for network tomography [38, 67], are *interactive* in that they employ feedback-driven measurement mechanisms. Interactivity allows one to use significantly fewer measurements and consequently reduces traffic injected into the network. In particular, we will use a nearly-linear number of measurements (in the number of end hosts) to recover the network structure.

5.2 Main Results

Let $\mathcal{X} = \{x_i\}_{i=1}^n$ denote the end hosts in a network and let $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$ be a function representing the true distances between the network hosts. Our work focuses on distance metrics d that approximate additive tree metrics. Specifically, let $\mathcal{T} = (\mathcal{V}, \mathcal{E}, c)$ be a tree with vertices $\mathcal{V}$, edges $\mathcal{E}$ and edge weights given by the cost function c, for which $\mathcal{X}$ is the set of leaves. An additive tree metric on $\mathcal{X}$ is the function $d_\mathcal{T}$ such that $d_\mathcal{T}(x_i, x_j) = \sum_{(y,z)\in \text{Path}(x_i,x_j)} c(y, z)$, that is the distance between two points is the sum of the edge weights along the unique path between them. A common algorithmic tool for resolving tree metrics is the *quartet test*, which resolves the structure between any four leaves in a tree using only pairwise distances between those leaves [68]. We will make extensive use of this algorithmic primitive.

In this work, we model network distances as $d(x_i, x_j) = d_\mathcal{T}(x_i, x_j) + g(x_i, x_j)$, where the tree $\mathcal{T}$ is unknown and the function g captures the network's deviations from a tree metric. This approach gives us a firm mathematical basis under which we can make rigorous guarantees about the performance of our algorithms. We focus on two models for these deviations:

1. **Additive Noise:** Here, $g(x_i, x_j)$ is drawn from a subgaussian distribution with scale factor σ^2. This captures inherent randomness in certain measurement types, such as latencies. We allow repeated measurements, with fresh randomness, and aim to minimize the total number of measurements.
2. **Persistent Noise:** Here $g(x_i, x_j) = 0$ with probability q (q is known), independent of all other x_i and x_j, and with probability $1-q$, $g(x_i, x_j)$ is arbitrarily or adversarially chosen. This models gross deviations from the tree metric assumption caused by peering links, unresponsive nodes, or missing measurements. Repeated measurements cannot be used to average away this form of noise.

In the additive noise setting, we have the following theorem [1]:

Theorem 7 (Informal)**.** *Assume that the tree $\mathcal{T}$ has minimum edge length γ and maximum degree l. Then, in the additive noise model, there is an algorithm that exactly recovers the topology $\mathcal{T}$ and recovers the edge weights to within additive $O(\gamma)$ using $O(nl\frac{\sigma^2}{\gamma^2}\log^2(n))$ measurements.*

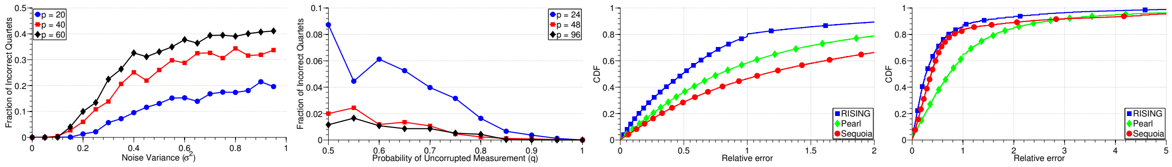
The idea behind the algorithm, which we call PearlReconstruct, is to iteratively attach leaves to a candidate solution tree $\mathcal{T}$. To add leaf x_i, we perform an intelligent series of quartet tests to find a pair of nodes x_j, x_k such that the distance between x_i and the shared ancestor of x_i, x_j, x_k is minimized. This information and a few more quartet tests determines how to add x_i to the tree. We choose to perform a quartet test involving an internal node with fairly balanced subtrees (known as the *pearl point* [68]) at each round, which allows us to reduce the search to a subtree that is multiplicative smaller using only a constant number of measurements. This leads to a logarithmic measurement complexity per node, although there is linear dependence on the degree l.

To be robust to noise, we repeat each measurement $O(\frac{\sigma^2}{\gamma^2}\log(n))$ times, which ensures that our distance estimates deviate from the truth by at most an additive $O(\gamma)$. This ensure not only that all quartet tests agree with the tree $\mathcal{T}$ so that we recover the topology exactly, but also that each edge weight is estimated to within $O(\gamma)$. This leads to the sample complexity bound in the theorem.

In the persistent noise setting, we have the following:

Theorem 8 (Informal)**.** *There is an algorithm that recovers all internal nodes of $\mathcal{T}$ for which every subtree has size at least $\Omega(q^{-6}\log(n))$ that uses $O(nq^{-6}\log^2(n))$ measurements.*

[1] Details, formal theorem statements, and proofs can be found in [56].



(a) (b) (c) (d)

Figure 5.1: Network Tomography simulations and experiments. (a): Additive noise tolerance of algorithm from Theorem 7. (b): Persistent noise tolerance of algorithm from Theorem 8. Relative errors on King dataset (c) and iPlane dataset (d).

The algorithm, named RISING, recursively partitions the leaves into groups corresponding to subtrees of $\mathcal{T}$ so that each partitioning step identifies one internal node of the tree. In more detail, the partitioning step involves first sampling a subset of leaves, clustering these leaves into subtrees, and then placing the remaining leaves into these subtrees. In the clustering phase, we compute a similarity function s on the sampled leaves where $s(x_i, x_j)$ is large if x_i, x_j belong in the same subtree. The similarity function is the number of quartets that pair x_i, x_j together, among all quartets formed by the subset of leaves. We compute the similarity function and then run the single linkage clustering algorithm to partition the subset.

In the second part of the partitioning step, we again use quartet tests to place each remaining leaf. For each leaf x_i, we compute quartet structures between x_i and three nodes, each from a different subtree, and place x_i into the subtree that most commonly paired with it. The dependence on q^6 is natural, as there are six distances used in each quartet test. Similarly, the requirement that only internal nodes with large enough subtrees can be recovered is natural since small subtrees are irrecoverably buried in noise.

Returning to our themes of computational efficiency and non-uniformity, the additive noise algorithm runs in $O(n\text{polylog}(n))$ time, while the persistent algorithm focuses measurements to resolve fine-grained structures of the underlying tree. As we are not aware of non-interactive algorithms that operate on subsampled versions of the input distance matrix, it is difficult to make comparisons to our methods. In the additive noise setting, the non-interactive approaches examine all pairwise distances and consequently have $\Omega(n^2)$ running times. In the persistent noise setting, it seems unlikely that non-interactive subsampling would yield as strong a guarantee as Theorem 8, as one would not have enough measurements to resolve internal nodes with small subtrees. Thus, this setting also lends evidence to our thesis.

5.2.1 Experiments

We now present some experimental results. In Figures 5.1(a) and 5.1(b) we assess the noise tolerance of our two algorithms on synthetic data. We measure error in terms of the fraction of quartets in the recovered tree that disagree with the reference tree. The figures show that both algorithms are exact in the absence of noise and robust to small amounts of the appropriate form of noise (additive or persistent). Moreover for both algorithms, the noise tolerance improves as the network size increases.

Figures 5.1(c) and 5.1(d) display experimental results on real-world network tomography datasets. We use two datasets: the King dataset [43] of pairwise latencies and a dataset of hop counts between PlanetLab [70] hosts measured using iPlane [62]. We selected a 500-node subset of the 1740-node King dataset. The iPlane dataset consists of 193 end hosts. We compared both of our algorithms with the Sequoia algorithm [71] on these datasets. Note that the Sequoia algorithm can be used to build many trees and uses the median distance across all trees as its distance estimates. To make a fair comparison, we ran Sequoia so that it used roughly the same number of measurements as RISING (Theorem 8).

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

Table 5.1: Measurements used on real world data sets

We plot the distribution of relative error values for each algorithm in Figures 5.1(c) and 5.1(d). Given the constructed tree metric $(\mathcal{X}, \hat{d})$ and the true metric $(\mathcal{X}, d)$, we measure relative error for each pairwise distance as $\frac{|\hat{d}(x,y)-d(x,y)|}{d(x,y)}$. We see that on both datasets, RISING outperforms both Sequoia and PearlReconstruct (Theorem 7), with substantial improvements on the King dataset. PearlReconstruct performs well on the King dataset, but not so well on the iPlane dataset.

Lastly, in Table 5.1 we record the number of measurements used by the algorithms on the two datasets. Note that all the algorithms use only a fairly small subset of the measurements, and the fraction of measurements used decreases on larger datasets. PearlReconstruct uses far fewer measurements than the other two algorithms, but is statistically much worse. Nevertheless, these results show that our algorithms can be used to robustly learn latent tree metrics using only a small number of pairwise distance measurements.

5.3 Future Work

We see several potential directions for future research:

1. One direction is to accommodate both persistent and additive noise simultaneously in the multi-source network tomography problem. This would lead to a more practical algorithm.
2. It is also worth understanding if a less brittle clustering algorithm will improve the statistical guarantees of the the RISING algorithm. The single linkage algorithm requires a strong guarantee on the similarity function that in turn introduces a unfavorable dependence on the error probability q and on the set of nodes that we can recover. It seems plausible that a more robust clustering algorithm can temper these dependencies.
3. Another restriction of the RISING algorithm is that it requires the tree to be fairly balanced. This ensures that the similarity function is robust to persistent noise, as there are enough leaves in each of the target subtrees. It would be worthwhile to design an algorithm that is robust to persistent noise but that also accommodates unbalanced tree structures.
4. Lastly, it seems worthwhile to understand the fundamental limits and the power of passive subsampling for this problem. This would allow for better comparison with our interactive approaches.

Bibliography

[1] Nir Ailon, Yudong Chen, and Xu Huan. Breaking the Small Cluster Barrier of Graph Clustering. *arxiv:1302.4549*, 2013. 2, 4.3

[2] Ery Arias-Castro, Emmanuel J. Candès, and Mark A. Davenport. On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 2013. 1

[3] Pranjal Awasthi, Maria-Florina Balcan, and Konstantin Voevodski. Local algorithms for interactive clustering. In *International Conference on Machine Learning*, 2013. 4.1

[4] Sivaraman Balakrishnan, Min Xu, Akshay Krishnamurthy, and Aarti Singh. Noise Thresholds for Spectral Clustering. In *Advances in Neural Information Processing Systems*, 2011. 3, 4.1, 4.2, 4.2, 4.2.1, 4.2.1

[5] Sivaraman Balakrishnan, Mladen Kolar, Alessandro Rinaldo, and Aarti Singh. Recovering block-structured activations using compressive measurements. *arXiv:1209.3431*, 2012. 1

[6] Maria-Florina Balcan and Avrim Blum. Clustering with Interactive Feedback. In *Algorithmic Learning Theory*, 2008. 4.1

[7] Maria-Florina Balcan and Steve Hanneke. Robust Interactive Learning. In *Conference on Learning Theory*, 2011. 1

[8] Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *Conference on Learning Theory*, 2007. 1

[9] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 2009. 1

[10] Maria-Florina Balcan, Steve Hanneke, and Jennifer Wortman Vaughan. The true sample complexity of active learning. *Machine Learning Journal*, 2010. 1

[11] Maria-Florina Balcan, Yingyu Liang, and Pramod Gupta. Robust Hierarchical Clustering. *Conference on Learning Theory*, 2010. 4.1

[12] Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Active Semi-Supervision for Pairwise Constrained Clustering. In *SIAM International Conference on Data Mining*, 2004. 4.1

[13] Alina Beygelzimer, Sanjoy Dasgupta, and John Langford. Importance weighted active learning. In *International Conference on Machine Learning*, 2009. 1

[14] Alina Beygelzimer, John Langford, Zhang Tong, and Daniel J. Hsu. Agnostic Active Learning Without Constraints. In *Advances in Neural Information Processing Systems*, 2010. 1

[15] Shankar Bhamidi, Ram Rajagopal, and Sébastien Roch. Network delay inference from additive metrics. *Random Structures & Algorithms*, 2010. 5.1

[16] 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. 3.1

[17] Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near optimal column-based matrix reconstruction. In *IEEE Symposium on Foundations of Computer Science*, 2011. 3.1

[18] Jian-Feng Cai, Emmanuel J. Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 2010. 2.2.1, 3

[19] Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 2010. 3.1

[20] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 2009. 2.1

[21] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 2010. 2.1

[22] Rui M Castro, Mark J Coates, Gang Liang, Robert Nowak, and Bin Yu. Network Tomography: Recent Developments. *Statistical Science*, 2004. 5.1

[23] Wei-Chen Chen. *Phylogenetic Clustering with R package phyclust*, 2010. URL http://thirteen-01.stat.iastate.edu/snoweye/phyclust/. 4.2.1

[24] Yudong Chen. Incoherence-optimal matrix completion. *arXiv:1310.0154*, 2013. 2.1

[25] Yudong Chen, Srinadh Bhojanapalli, Sujay Sanghavi, and Rachel Ward. Coherent matrix completion. In *International Conference on Machine Learning*, 2014. 2.1

[26] Mark J Coates, Rui M Castro, and Robert D Nowak. Maximum likelihood network topology identification from edge-based unicast measurements. *ACM SIGMETRICS*, 2002. 5.1

[27] Sanjoy Dasgupta. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems*, 2006. 1

[28] Sanjoy Dasgupta. Two faces of active learning. *Theoretical Computer Science*, 2011. 1

[29] Sanjoy Dasgupta, Claire Monteleoni, and Daniel J. Hsu. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems*, 2008. 1

[30] Vin de Silva and Joshua B. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. *Advances in Neural Information Processing Systems*, 2002. 2

[31] 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. 3.2

[32] 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. 3.2

[33] Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM Journal on Matrix Analysis and Applications*, 2008. 3.1

[34] Nick G. Duffield and Francesco Lo Presti. Network tomography from measured end-to-end delay covariance. *IEEE/ACM Transactions on Networking*, 2004. 5.1

[35] Nick G. Duffield, Joseph Horowitz, and Francesco Lo Presti. Adaptive multicast topology inference. *IEEE INFOCOM*, 2001. 5.1

[36] 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. 5.1

[37] Nick G. Duffield, Francesco Lo Presti, Vern Paxson, and Don Towsley. Network loss tomography using striped unicast probes. *IEEE/ACM Transactions on Networking*, 2006. 5.1

[38] Brian Eriksson, Gautam Dasarathy, Paul Barford, and Robert Nowak. Toward the Practical Use of Network Tomography for Internet Topology Discovery. *IEEE INFOCOM*, 2010. 5.1

[39] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert Nowak. Active Clustering: Robust and Efficient Hierarchical Clustering using Adaptively Selected Similarities. *AISTATS*, 2011. 4.1

[40] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 2004. 3.2

[41] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 2011. 2.1

[42] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, March 2011. 2.1

[43] 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. 5.2.1

[44] Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *ACM-SIAM symposium on Discrete Algorithms*. SIAM, 2012. 3.1

[45] Steve Hanneke. A bound on the label complexity of agnostic active learning. In *International conference on Machine Learning*, 2007. 1

[46] Steve Hanneke. Teaching dimension and the complexity of active learning. *Conference on Learning Theory*, 2007. 1

[47] Steve Hanneke. Rates of convergence in active learning. *The Annals of Statistics*, 2011. 1

[48] Moritz Hardt. Understanding Alternating Minimization for Matrix Completion. In *Foundations of Computer Science*, 2014. 2.1

[49] Jarvis Haupt, Rui Castro, and Robert Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 2011. 1

[50] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *ACM Symposium on Theory of Computing*, 2013. 2.1

[51] Rong Jin and Shenghuo Zhu. CUR Algorithm with Incomplete Matrix Observation. *arxiv:1403.5647*, 2014. 2.1, 1, 2.2

[52] David Kauchak and Sanjoy Dasgupta. An Iterative Improvement Procedure for Hierarchical Clustering. In *Advances in Neural Information Processing Systems*, 2004. 4.2.1

[53] Raghunandan H. Keshavan. *Efficient Algorithms for Collaborative Filtering*. PhD thesis, Stanford University, 2012. 2.1

[54] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 2010. 3.1

[55] 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. 1, 3.1, 3.2, 3.2, 3.3

[56] Akshay Krishnamurthy and Aarti Singh. Robust multi-source network tomography using selective probes. In *IEEE INFOCOM*, 2012. 1

[57] Akshay Krishnamurthy and Aarti Singh. Low-rank matrix and tensor completion via adaptive sam-

pling. *Advances in Neural Information Processing Systems*, 2013. 2, 3

[58] Akshay Krishnamurthy and Aarti Singh. On the Power of Adaptivity in Matrix Completion and Approximation. *arxiv:1407.3619*, 2014. 1, 2, 1, 2

[59] Akshay Krishnamurthy, Sivaraman Balakrishnan, Min Xu, and Aarti Singh. Efficient active algorithms for hierarchical clustering. In *International Conference on Machine Learning*, 2012. 4.2, 1

[60] Akshay Krishnamurthy, James Sharpnack, and Aarti Singh. Recovering graph-structured activations using adaptive compressive measurements. *arXiv:1305.0213*, 2013. 1

[61] Akshay Krishnamurthy, Martin Azizyan, and Aarti Singh. Subspace learning from extremely compressed measurements. *arXiv:1404.0751*, 2014. 4, 3.3

[62] 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`. 5.2.1

[63] Matthew Malloy and Robert Nowak. Sequential analysis in high-dimensional multiple testing and sparse recovery. *IEEE International Symposium on Information Theory*, 2011. 1

[64] Tom Mitchell. Noun Phrases in Context 500 Dataset, 2009. URL `http://www.cs.cmu.edu/~tom/10709_fall09/RTWdata.html`. 4.2.1

[65] Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. Square Deal: Lower Bounds and Improved Relaxations for Tensor Recovery. *arxiv:1307.5870*, 2013. 2.1

[66] Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted matrix completion: optimal bounds with noise. *The Journal of Machine Learning Research*, 2012. 1, 3.1, 3.3

[67] 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. 5.1

[68] Judea Pearl and Michael Tarsi. Structuring causal trees. *Journal of Complexity*, 1986. 5.2, 5.2

[69] 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. 4.2.1

[70] Larry Peterson, Andy Bavier, Marc E. Fiuczynski, and Steve Muir. Experiences building planetlab. In *Symposium on operating systems design and implementation*, 2006. 5.2.1

[71] Venugopalan Ramasubramanian, Dahlia Malkhi, Fabian Kuhn, Mahesh Balakrishnan, Archit Gupta, and Aditya Akella. On the treeness of internet latency and bandwidth. *ACM SIGMETRICS*, 2009. 5.1, 5.2.1

[72] Benjamin Recht. A simpler approach to matrix completion. *The Journal of Machine Learning Research*, 2011. 2.1, 2.2

[73] Sam T. Roweis. NIPS Articles 1987-1999, 2002. URL `http://cs.nyu.edu/~roweis/data.html`. 4.2.1

[74] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geo-

metric functional analysis. *Journal of the ACM (JACM)*, 2007. 3.2

[75] Ohad Shamir and Naftali Tishby. Spectral Clustering on a Budget. *AISTATS*, 2011. 4.1

[76] Ervin Tánczos and Rui Castro. Adaptive sensing for estimation of structured sparse signals. *arXiv:1311.7118*, 2013. 1

[77] Ryota Tomioka and Taiji Suzuki. Convex Tensor Decomposition via Structured Schatten Norm Regularization. In *Advances in Neural Information Processing Systems*, 2013. 2.1

[78] Ryota Tomioka, Kohei Hayashi, and Hisashi Kashima. Estimation of low-rank tensors via convex optimization. *arxiv:1010.0789*, 2010. 2.1

[79] Ryota Tomioka, Taiji Suzuki, Kohei Hayashi, and Hisashi Kashima. Statistical Performance of Convex Tensor Decomposition. In *Advances in Neural Information Processing Systems*, 2011. 2.1

[80] Yolanda Tsang, Mehmet Yildiz, Paul Barford, and Robert Nowak. Network radar: tomography from round trip time measurements. In *ACM SIGCOMM*, 2004. 5.1

[81] Yehuda Vardi. Network Tomography: Estimating Source-Destination Traffic Intensities from Link Data. *Journal of the American Statistical Association*, 1996. 5.1

[82] Ming Yuan and Cun-Hui Zhang. On Tensor Completion via Nuclear Norm Minimization. *arxiv:1405.1773*, 2014. 2.1