Data Efficient and Robust Algorithms for Reconstructing Large Graphs

By

Gautam Dasarathy

A dissertation submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN – MADISON

2014

Date of final oral examination: June 2, 2014

The dissertation is approved by the following members of the Final Oral Committee:
Robert Nowak, Professor, Electrical and Computer Engineering
Stark Draper, Associate Professor, Electrical and Computer Engineering
John Gubner, Professor, Electrical and Computer Engineering
Sebastien Roch, Associate Professor, Mathematics
Xiaojin (Jerry) Zhu, Associate Professor, Computer Sciences

To Amma, Appa, Ash, and Jyo

Abstract

Many modern signal and information processing tasks center around the discovery of (significant) interactions in a large, complex system from measurements made about it. This can be thought of as *reconstructing a graph*, that is, the discovery of the (significant) edges of an appropriate graph from data. In this dissertation, we devise provably robust and efficient algorithms for various graph reconstruction problems.

The first problem we consider is the reconstruction of the (router-level) topology of computer networks. We demonstrate a family of algorithms that leverages the structure present in the underlying network to recover the routing topology more accurately and with far fewer probes than previous techniques.

We next show that similar techniques can be brought to bear on the more general problem of hierarchical clustering based on pairwise similarities. In particular, we exhibit algorithms for reliable recovery of the hierarchical clustering that are (a) near optimal in terms of the number of pairwise similarities needed, and (b) robust in terms of the regularity assumptions on the pairwise similarity values.

The third problem concerns the recovery of more general graph structures and is motivated by an application we call "covariance sketching": the recovery of the covariance graph (i.e., the graph indicating the dependence among variables) from compressed samples of the underlying random process. We show that under certain assumptions, this can be done efficiently and robustly using a simple convex program. The theoretical framework we develop has direct implications for compressed acquisition of multi-dimensional signals as well.

Finally, we consider an important problem from quantitative biology – phylogenetic inference from molecular data corresponding to several genes. It is known that the evolutionary history of individual genes might be topologically distinct from each other and from the underlying species tree, possibly confounding phylogenetic analysis. A further complication in practice is that one has to estimate gene trees from molecular sequences of finite length. To address this, we devise a novel algorithm that explicitly takes into account gene tree estimation errors and provably improves over all previous methods in a regime of interest.

Acknowledgements

First and foremost, I wish to express my sincerest thanks to my advisors, Robert Nowak and Stark Draper, both of whom are remarkable teachers, mentors, and colleagues. I spent a significant portion of my time in graduate school working with Rob. His infectious enthusiasm for research has taught me how to truly *sift and winnow*, the Wisconsin way. I wish I am able to emulate both Rob's mentorship and his exceptional ability to find interesting problems in the future. Stark has been ever ready to roll up his sleeves and help me tackle problems, both technical and non-technical, throughout my time at UW. I will always remember and cherish the fantastic times I have had discussing research with him. I look forward to many years of fruitful collaboration with both Rob and Stark.

Many thanks to the members of my thesis committee, John Gubner, Sebastien Roch, and Jerry Zhu, for their valuable feedback and insightful comments. I count myself extremely fortunate to have had the opportunity to interact with all of them closely during my years at UW.

While this dissertation only bears my name in the front page, the work presented here wouldn't have been possible without my collaborators. I learnt a lot from and thoroughly enjoyed working with Badri Bhaskar, Brian Eriksson, Sebastien Roch, Parikshit Shah, and Aarti Singh.

Looking back, it seems to me as though my time in graduate school was quite a bit more enjoyable than what the stories make it out to be. I owe this almost entirely to a wonderful group of colleagues and friends – Laura Balzano, Aniruddha Bhargava, Zac Harmany, Kevin Jamieson, Shirzad Malekpour, Nikhil Rao, Yana Shkel, and Vincent Tan.

I am also very grateful to all my friends outside the lab who helped make my Madison experience something I will treasure all my life.

Next, I wish to thank my family. I will forever be indebted to my parents Sowmya and Dasarathy for their unconditional love, support, and belief all these years. Without their continual encouragement, I will not be where I am today. Ash is simply the best younger brother a person could hope for; even though he is seven years my junior, he has taught me way more than he suspects. I am also grateful to my grandparents who have always been an inexhaustible source of wisdom and inspiration. Lastly, I wish to thank my wife, Jyo. I do not quite have the words to express the amount of gratitude I have felt throughout my time in grad school for her constant motivation, unwavering faith, selfless sacrifices, and boundless love.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Whether it is biological networks of proteins and genes or man-made ones like sensor networks and the Internet, we are surrounded by complex systems composed of entities interacting with and affecting each other. One extremely elegant mathematical formalism for representing such interactions is a *graph*. A graph depicts a set of objects, certain pairs of which are connected by links. These objects are represented by mathematical abstractions called *vertices*, and the links that connect them are simply represented as a collection of 2 element subsets of the set of vertices and are called *edges*. See Section 1.2 for more details.

In studying a complex system, it is not only important to model and understand the entities that make up the system but it is also important to understand how these entities interact; this often enhances our ability to define, conceptualize, visualize, quantify, and simulate the phenomenon being studied. Therefore, unsurprisingly, many modern signal and information processing tasks center around the discovery of (significant) interactions in a complex system from measurements made about it. This can be thought of as *reconstructing a graph*, that is, discovering the (significant) edges of an appropriate graph from data.

As an example of a graph reconstruction problem, consider the situation of a Content Delivery Network (CDN) operator, like Akamai, which maintains a large distributed

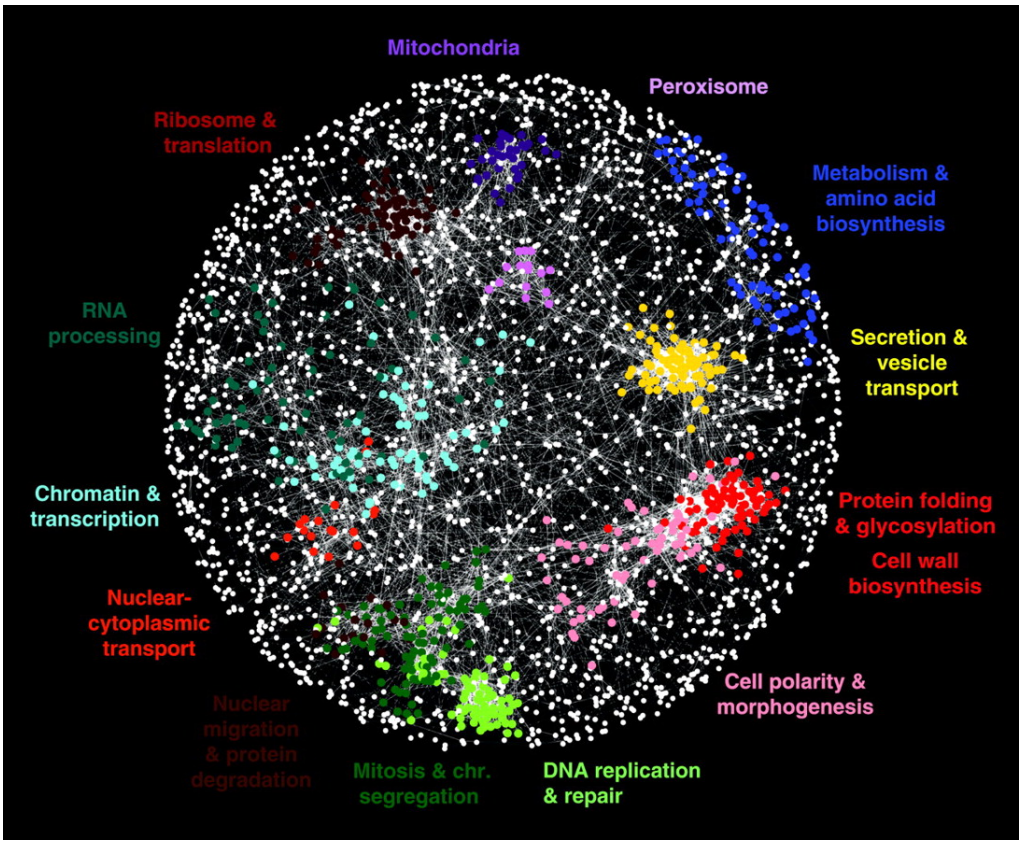


Figure 1.1: The "genetic landscape" of the budding yeast cell (*Saccharomyces cerevisiae*). This interaction map was built by measuring the pairwise correlations between ∼6000 genes. Roughly speaking, the colored regions indicate genes that are known to share similar functions. (Reprinted with permission from [25] – "The Genetic Landscape of a Cell" by M. Costanzo et al.)

system of servers deployed in multiple data centers across the Internet. The CDN operator gets paid by content providers (e.g., Youtube) to serve high-quality content (e.g., cat videos) to end-users quickly. Therefore, the CDN operator would benefit greatly from maintaining an accurate topology of the Internet so that a server close to the end-user may be used to dispatch the content requested. This is a graph reconstruction problem where the graph is (essentially induced by) the network of physical wires connecting the computers. On the other hand, consider the problem of studying the chemical-genetic

interactions in a cell which is crucial to our understanding of the functions of various genes and could have far reaching consequences for the identification of drug targets and therefore, drug discovery. In [25], the authors considered this problem in the context of the budding yeast cell (*Saccharomyces cerevisiae*) and built a genetic interaction map by examining the results of 5.4 million gene-gene pairwise knockout experiments (corresponding to around 6000 genes). This gene network (shown in Figure 1.1) reveals a functional map of the cell in which genes with similar biological functions cluster together in coherent subsets. This is a graph reconstruction problem as well, where the graph being reconstructed is a conceptual graph depicting functional relationships between genes.

In such problems of reconstructing graphs, an important consideration that comes up is the kind and the amount of data available for making these inferences. This is especially relevant in recent years, with the drastic change in the climate of modern data analysis, where it has become commonplace to study extremely large systems. For instance, even a very conservative estimate [55] indicates that there are about 8 billion devices connected to the Internet today! In other words, the CDN operator cannot realistically hope to make measurements about the Internet structure that are either accurate or exhaustive. In a similar vein, even a moderately complex organism like the fruit-fly (*Drosophila melanogaster*) has around 20,000 genes. Therefore, replicating the results of [25], which took about 1 year in the case of the yeast cell, would be nearly impossible in the case of the fruit-fly. Such considerations have revealed an urgent need for the development of techniques of data analytics that are both efficient and robust. Indeed, this need has motivated a significant thrust of the research program which is the protagonist of this dissertation. In what follows, we will revisit the above problems,

among others, and we will use these examples to show how techniques at the intersection of signal processing, statistics, and machine learning can be used to devise provably robust and efficient algorithms for solving graph reconstruction problems. In particular, we will see that by taking careful advantage of the structure present in the underlying problem, one can design reliable algorithms that require much lesser data and tolerate much more "noise". It will also become apparent that taking this systematic approach to designing large scale graph reconstruction algorithms exposes the fundamental complexity of the problem being studied and hence furthers our understanding of the underlying phenomenon.

1.1 Contributions and Organization of the Dissertation

Chapter 2 – Network Radar: Efficient Network Tomography

In Chapter 2, we will encounter a problem that is motivated by a well known issue in the network mapping and measurement community – how does one efficiently infer the structure of the internet? In addition to their appeal to network researchers, accurate and timely maps of the Internet have a wide range of applications and are of particular importance in content delivery, network management, operations and security.

Typically, this topology inference problem is handled using `traceroute`-like probes. These are special messages that report back to the originating station about every hop they make on the network en route their destination. However, there are many practical issues associated with using these probes for network topology inference. For

instance, they levy an undesirable amount of load on the network and are typically blocked by network administrators for this reason. An attractive complement to these `traceroute`-based techniques are the so-called *tomography* techniques, which only use the end-to-end delay information from "ordinary" messages propagating through the network. Unfortunately, traditional techniques for network tomography make timely resolution of large network topologies impossible since they require an infeasible number of probes to perform topology discovery.

In this work, we devised new techniques that bring us closer to the ideal of practical tomographic inference of large scale network topologies. Our methods are based on a novel Depth First Search (DFS) Ordering that clusters "end hosts" based on the amount of infrastructure they share. This can then be used to recover the logical tree topology of the underlying network accurately and efficiently. We will also see the performance of our algorithms, first in simulation, where, for instance, our methods can be seen to reconstruct topologies using less than 2% of the probes required by prior "exhaustive" methods and less than 15% of the probes needed by the current state-of-the-art tomographic approach. We will also see results from a case study in the live Internet where our DFS-based methods can recover the logical router-level topology more accurately and with far fewer probes than prior techniques.

Relevant Publications

Toward the Practical Use of Network Tomography for Internet Topology Discovery. IEEE INFOCOM Conference 2010, San Diego, CA, March 2010. With B. Eriksson, P. Barford, and R. Nowak.

Efficient Network Tomography for Internet Topology Discovery. IEEE/ACM

Transactions on Networking, Vol 20, Issue 3, June 2012. With B. Eriksson, P. Barford, and R. Nowak.

Chapter 3 – Active Clustering: Robust and Efficient Hierarchical Clustering

The network topology discovery problem described above is closely related to the problem of *hierarchical clustering* of a set of objects. Recall that hierarchical clustering is a type of cluster analysis, where the objects in question are grouped into a hierarchy of clusters. This is now a pervasive tool used in a broad range of scientific and engineering applications. Typically, hierarchical clustering of n objects is performed by first computing an $n \times n$ similarity matrix and then processing this similarity matrix to form a hierarchy of clusters. Indeed this is how the prototypical hierarchical clustering algorithms, UPGMA [126], CLINK [34], and SLINK [125] work. However, in many problems it may be expensive to obtain or compute pairwise similarities between the items to be clustered.

In Chapter 3, we will see that it is possible to perform hierarchical clustering of n items using only a small subset of all the pairwise similarities. First, we show that in an ideal setting, where the intra-cluster similarities exceed inter-cluster similarities, it is possible to correctly determine the hierarchical clustering from as few as $3n \log n$ similarities, as opposed to $\binom{n}{2}$. This order of magnitude savings in the number of pairwise similarities is achieved by a sequential and adaptive selection of the similarities, rather than say, picking them at random. We then describe our active clustering method that is robust to a limited fraction of anomalous similarities, and show how even in the presence of these "noisy" similarity values one can resolve the hierarchical clustering efficiently

using only $\mathcal{O}(n \log^2 n)$ pairwise similarities. These theoretical results are followed up with some experiments that demonstrate the efficacy of our algorithms.

Relevant Publications

Active Clustering: Robust and Efficient Hierarchical Clustering using Adaptively Selected Similarities. Artificial Intelligence and Statistics (AISTATS) 2011, Ft. Lauderdale, Florida (Journal length Computer Science Paper). With B.Eriksson, A. Singh, and R. Nowak

An extended version is available at : http://arxiv.org/abs/1102.3887.

Chapter 4 – Sketching Sparse Covariances, Matrices, and Graphs

While the focus of Chapters 2 and 3 is on the reconstruction of trees, in Chapter 4, we will talk about robust and data efficient reconstruction of more general graphs. This chapter is dedicated to the problem of recovering an unknown sparse $p \times p$ matrix X from an $m \times m$ matrix $Y = AXB^T$, where A and B are known $m \times p$ matrices with $m \ll p$.

The main result tells us that there exist constructions of the "sketching" matrices A and B so that even if X has $\mathcal{O}(p)$ non-zeros, it can be recovered exactly and efficiently using a convex program as long as these non-zeros are not concentrated in any single row/column of X. Furthermore, it suffices for the size of Y (the sketch dimension) to scale as $m = \mathcal{O}\left(\sqrt{\#\text{ nonzeros in } X} \times \log p\right)$, which is nearly optimal. We also show that the recovery is robust and stable in the sense that if X is equal to a sparse matrix plus a perturbation, then the convex program we propose produces an approximation with accuracy proportional to the size of the perturbation. Unlike traditional results on sparse recovery, where the sensing matrix produces independent measurements, our sensing

operator is highly constrained (it assumes a tensor product structure). Therefore, proving recovery guarantees require non-standard techniques. Indeed our approach relies on a novel result concerning tensor products of bipartite graphs, which may be of independent interest.

This problem is motivated by the following application, among others. Suppose that we want to estimate the covariance structure $\Sigma \in \mathbb{R}^{p\times p}$ of p variables represented by the (0 mean) random vector $\xi \in \mathbb{R}^p$. This can be considered as a graph reconstruction problem, where the vertices correspond to the p random variables and the edges correspond to the pairwise covariance values. In many applications of interest, like in the genetic landscape example described above or in building protein-protein interaction networks, one determines Σ by conducting correlation tests for each pair of covariates ξ_i, ξ_j and computing an estimate of $\mathbf{E}[\xi_i\xi_j]$ for $i, j = 1, \ldots, p$. This requires one to perform correlation tests for $\mathcal{O}(p^2)$ pairs of covariates, a daunting task when p is large. In other settings of interest, one may obtain k independent *sample paths* of the underlying statistical process and compute an appropriate statistic. When ξ is high-dimensional, it may be infeasible or undesirable to sample and store entire sample paths $\xi^{(1)}, \ldots, \xi^{(k)} \in \mathbb{R}^p$, and it may be desirable to reduce the dimensionality of the acquired samples. Thus in the high-dimensional setting we propose an alternative acquisition mechanism: pool covariates together to form a collection of new variables $Z_1, \ldots, Z_m$, where $m \ll p$. For example one may construct: $Z_1 = \xi_1 + \xi_2 + \xi_6$, $Z_2 = \xi_1 + \xi_4 + \xi_8 + \xi_{12}, \ldots$ and so on; more generally we may have measurements of the form $Z = A\xi$ where $A \in \{0, 1\}^{m\times p}$ and typically $m \ll p$. We call the thus newly constructed covariates $Z = (Z_1, \ldots, Z_m)$ a *sketch* of the random vector ξ. Our theoretical results show that as long as the system satisfies the reasonable assumption that Σ (or equivalently the corresponding graph) is

sparse, i.e., each variable is only (significantly) correlated with a small number of others, these correlations can be provably extracted from the pooled or sketched samples using an algorithm that is both efficient and robust.

Relevant Publications

Covariance Sketching. Allerton Conference on Communication and Control, UIUC. Oct 2012 (R. N. Invited). With P. Shah, B. Bhaskar, and R. Nowak.

Sketching Sparse Covariance Matrices and Graphs. NIPS workshop on Randomized Methods in Machine Learning, Lake Tahoe, Nevada, Dec 2013. With P. Shah, B. Bhaskar, and R. Nowak.

Sketching Sparse Matrices. Under Review in the IEEE Transactions on Information Theory.

Preprint : http://arxiv.org/abs/1303.6544. With P. Shah, B. Bhaskar, and R. Nowak.

Chapter 5 – Phylogenetic Inference from Multiple Loci

We next turn to some recent work on phylogenetic inference in Chapter 5. Here we consider the problem of inferring the evolutionary history of a set of species (phylogeny or species tree) from molecular data corresponding to several genes. It is known that the evolutionary history of individual genes (gene trees) might be topologically distinct from each other and from the underlying species tree, possibly confounding phylogenetic analysis. A further complication in practice is that one has to estimate gene trees from molecular sequences of finite length.

In Chapter 5, we will first introduce a statistical model that explains such gene tree

discordance. We then explain our first full data-requirement analysis of a species tree reconstruction method that takes into account estimation errors at the gene level. We will also describe a novel reconstruction algorithm that is robust and provably improves over all previous methods in a regime of interest. With respect to the data acquisition phase, this chapter departs from the theme of the previous chapters in that there is neither a compressive nor adaptive component involved. Nevertheless, our algorithm is vastly more data efficient than all previous methods in a regime of interest. It is also more computationally efficient than a family of existing methods that use a bayesian approach to reconstruct phylogenies. Furthermore, incorporating an adaptive querying phase into this method will be an interesting avenue for future work and we discuss this at the end of this chapter.

Relevant Publications

New Sample Complexity Bounds for Phylogenetic Inference from Multiple Loci. IEEE International Symposium on Information Theory, Honolulu, Hawaii, July 2014. With R. Nowak, and S. Roch.

Data Requirement for Phylogenetic Inference from Multiple Loci: A New Distance Method. Submitted to IEEE Transactions on Computational Biology and Bioinformatics. Preprint: http://arxiv.org/abs/1404.7055. With R. Nowak, and S. Roch.

1.2 Background and Related Work

In this section, we will get acquainted with some notation and concepts that will be used throughout this dissertation. Since Chapters 2 - 5 are written with the goal of being as self contained as possible, the terminologies and concepts that are specific to those chapters will be introduced as and when we encounter them. We will also see a brief account of some work related to the themes of this dissertation. A more detailed context will be provided in the individual chapters.

1.2.1 Graphs, Trees, and Tree Metrics

Graphs of course play a central role in this dissertation. A ***graph*** $G = (V, E)$ is an ordered pair consisting of two sets V and E, where $E \subset \binom{V}{2}$[1]. The elements of the set V are called ***vertices*** (or ***nodes***) and the elements of the set E are called ***edges***. We will write $V(G)$ and $E(G)$ if the graph G is not clear from the context. Pictorially, the vertices are represented as dots and the edges are represented by lines connecting the dots. For $v \in V$ and $e \in E$, we say that e is ***incident*** on v if $v \in e$, and if $u, v \in V$ are such that $\{u, v\} \in E$, then we say that u and v are ***adjacent*** or that u (resp. v) is a ***neighbor*** of v (resp. u). We sometimes write the edge $\{u, v\}$ simply as uv. The set of all neighbors of $v \in V$ is called its ***neighborhood*** and is denoted by $N_G(v)$, or just $N(v)$. The ***degree*** $\deg_G(v) = \deg(v)$ of a vertex $v \in V$ is defined as the size of its neighborhood, i.e., $\deg(v) = |N(v)|$. We say that $\tilde{G} = (\tilde{V}, \tilde{E})$ is ***isomorphic*** to G (and write $\tilde{G} \cong G$) if there exists a bijection $\phi : V \to \tilde{V}$ such that $uv \in E \Leftrightarrow \phi(u)\phi(v) \in \tilde{E}$. In this dissertation, for the sake of simplicity, we will consider isomorphic graphs to be

[1]Given a set A, we write $\binom{A}{2}$ to denote the set of all 2 element subsets from A.

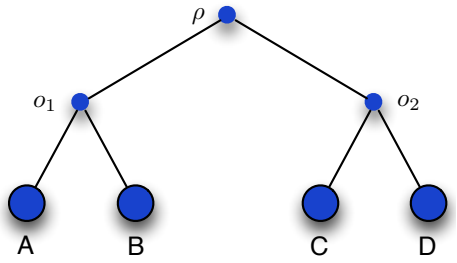


Figure 1.2: The Newick format for representing the topology induced by this tree on the leaf set $\{A, B, C, D\}$ is $((A, B), (C, D))$. Similarly, the Newick format for representing the topology induced by this tree on $\{A, C, D\}$ is $(A, (C, D))$.

the same. We say that $\tilde{G} = (\tilde{V}, \tilde{E})$ is a ***subgraph*** of G (and write $\tilde{G} \subset G$) if $\tilde{V} \subset V$ and $\tilde{E} \subset E$. If $\tilde{G} \subset G$ and $\tilde{G}$ contains all the edges $uv \in E$ with $u, v \in \tilde{V}$, then $\tilde{G}$ is said to be an ***induced subgraph*** of G. A ***path*** $P = (V', E')$ is a nonempty graph which has the form $V' = \{v_1, v_2, \ldots, v_r\}$ and $E' = \{v_1v_2, v_2v_3, v_3v_4, \ldots, v_{r-1}v_r\}$. Suppose $u, v \in V$ and suppose that $P \subset G$, we (will slightly abuse notation and) say that P is a $u - v$ path if $v_1 = u$ and $v_r = v$. We say that the graph G is ***connected*** if for every pair of vertices $u, v \in V$, there exists a $u - v$ path in the graph.

If $r \geq 3$ and if we add the edge $\{v_1, v_r\}$ to the path graph P above, then this graph is called a ***cycle***. An ***acyclic*** graph is one that contains no cycles and such graphs are also called ***forests***. A connected forest is called a ***tree***. We will encounter trees frequently in this dissertation and we will usually denote them by T or $\mathcal{T}$. The vertices of degree 1 in a tree are called its ***leaves*** and are denoted by the set L or $L(T)$. Sometimes, we will talk of rooted trees where a special vertex is chosen to be called the root. We will write ρ or $\rho(T)$ to denote the root of the tree T. Given two vertices u, v in a tree $T = (V, E)$, we observe that there is a *unique* path that connects these vertices in the tree and we denote this path by π_{uv} or π_{uv}^T. We will sometimes consider *weighted* trees. That is, to

each edge $e \in E$, we will associate a weight or a length denoted $d : E \to \mathbb{R}^{>0}$, where $\mathbb{R}^{>0} = \{x \in \mathbb{R}; x > 0\}$. The edge weights naturally induce a metric on the vertices of the tree and in particular on the leaves of the tree as follows. Given any two vertices u, v in the tree, let $d(u, v) = \sum_{e \in \pi_{uv}^T} d(e)$. When restricted to the leaves $L \subset V$ of T, such a metric is called an ***additive metric*** or ***tree metric***.

When dealing with rooted trees, we will sometimes find it useful to denote the subtree induced on some vertices using the Newick format. See Figure 1.2 for an example. We say that the induced metric d forms an ***ultrametric on*** $L(T)$ ***with respect to*** T if for all triplets $u, v, w \in V$ such that $((u, v), w)$ holds, d satisfies

$$d(u, v) < d(u, w) = d(v, w). \tag{1.1}$$

The interested reader is referred to the excellent monograph [121] (and the references therein) for more on tree metrics, ultrametrics and reconstruction of trees given access to such metrics.

1.2.2 Some Probability Theory

In this section, we will record some results from probability theory that will be used in this dissertation.

We will frequently use the fact that well-behaved random variables concentrate quickly around their expected values. The Hoeffding's (tail) inequality [78] is one such result which we will find very useful in this dissertation.

Theorem 1.1 (Hoeffding's inequality)**.** *Let $X_1, \ldots, X_n$ be independent random variables such that $X_i \in [a_i, b_i]$ with probability 1. Then, if we define $S_n = \sum_{i=1}^n X_i$, for any $\epsilon > 0$*

we have

$$\mathbb{P}\left[S_n - \mathbb{E}S_n > \epsilon\right] \leq e^{-2\epsilon^2/\sum_{i=1}^n (b_i-a_i)^2}$$
$$\mathbb{P}\left[S_n - \mathbb{E}S_n < -\epsilon\right] \leq e^{-2\epsilon^2/\sum_{i=1}^n (b_i-a_i)^2}.$$

A version of this inequality was shown for the case when the X_i are Bernoulli random variables by Chernoff [22] and when we encounter Bernoulli random variables, we will use the names "Chernoff bound" and "Hoeffding's inequality" interchangeably.

One disadvantage of this large deviation inequality is that it is does not use much information about the distribution of the random variables. *Bernstein's inequality* [9] is a very well known large deviation inequality that takes into account the variance information of the random variables X_i; we state it here.

Theorem 1.2 (Bernstein's inequality)**.** *Let $X_1, \ldots, X_n$ be zero mean independent random variables such that $X_i \leq 1$ almost surely. Let*

$$\sigma^2 \triangleq \frac{1}{n}\sum_{i=1}^n \mathrm{Var}\left(X_i\right).$$

Then, for any $\epsilon > 0$, with S_n as defined above, we have

$$\mathbb{P}\left[\frac{S_n}{n} > \epsilon\right] \leq e^{-n\epsilon^2/2(\sigma^2+\epsilon/3)} \tag{1.2}$$

In order to compute variances of random variable (for use in, say, the above inequality), the following formula is often useful:

$$\mathrm{Var}\left[X\right] = \mathbb{E}\left[\mathrm{Var}\left(X|Y\right)\right] + \mathrm{Var}\left(\mathbb{E}\left[X|Y\right]\right).$$

This is called the *conditional variance* formula or the *law of total variance* [49].

Now, suppose that the random variables $X_1, \ldots, X_n$ are not independent, then one does not always get such powerful concentration results. However, suppose that there is only *limited dependence* among these random variables, that is, suppose that each X_i is dependent on at most $\delta \leq n$ other random variables. Then, one might hope to extract independent sub-sums from S_n and still apply Hoeffding's inequality to each of those. A principled way of doing this is suggested by the celebrated Hajnal-Szemerédi theorem [71, 90]. Consider a graph on the vertex set $[n] \triangleq \{1, 2, \ldots, n\}$ where there is an edge between vertices i and j if X_i and X_j are dependent. Since this graph has degree δ, the Hajnal-Szemeridi theorem tells us that this graph can be *equitably colored* with $\Theta(\delta)$ colors, i.e., there exists an assignment of colors to the vertices of the graphs such that no two adjacent vertices are monochromatic, and the number of vertices in any two color classes differ by at most 1. In other words, the set $\{X_i\}_{i\in[n]}$ can be partitioned into $k \in \Theta(n/\delta)$ sub-sets $S^{(1)}, \ldots, S^{(k)}$ such that each sub-set has $m \in \Theta(\delta)$ elements and the corresponding random variables in each of them are independent. We can then use the union bound and, say, Hoeffding's inequality (letting $a_i = 0, b_i = 1$) to get the following tail inequality, for instance,

$$\begin{aligned}
\mathbb{P}\left[S_n - \mathbb{E}S_n > \epsilon\right] &= \mathbb{P}\left[\sum_{j\in[k]}\left(\sum_{i\in S^{(j)}} X_i - \mathbb{E}\sum_{i\in S^{(i)}} X_i\right) > \epsilon\right] \\
&\overset{(a)}{\leq} \sum_{j=1}^{k} \mathbb{P}\left[\sum_{i\in S^{(j)}} X_i - \mathbb{E}\sum_{i\in S^{(i)}} X_i > \frac{\epsilon}{k}\right] \\
&\overset{(b)}{\leq} k e^{-2\epsilon^2/mk^2} \\
&\leq C\frac{n}{\delta} e^{-2c\delta\epsilon^2/n^2}
\end{aligned}$$

where (a) follows from the union bound and (b) follows from Theorem 1.1. In the last inequality c and C are positive constants hidden in the $\Theta(\cdot)$ notation above. This can be quite useful when δ is small relative to n and this will prove to be very useful in establishing the results in Chapter 4.

In Chapter 5, the exponential distribution plays a very important role and we will record some of its properties here. We write $X \sim \text{Exp}(\lambda)$ to mean that the random variable X is distributed according to the exponential distribution with *rate* λ. Recall that the probability density function (pdf) of the exponential distribution with rate $\lambda > 0$ is given by

$$f(x) = \lambda e^{-\lambda x} \mathbb{1}\{x \geq 0\},$$

where, $\mathbb{1}\{\cdot\}$ stands for the "indicator function". Therefore, its cumulative distribution function (cdf) is given by $F(x) = (1 - e^{-\lambda x})\mathbb{1}\{x \geq 0\}$. By a straightforward calculation, it can be seen that its mean and variance are respectively given by λ^{-1} and λ^{-2}. We will find the following two properties of the exponential distribution particularly useful.

1. **Distribution of the minimum.** Suppose that the independent random variables $X_i, i = 1, \ldots, n$ are each exponentially distributed with the rates $\lambda_i, i = 1, \ldots, n$. Then, the random variable $Z \triangleq \min\{X_1, \ldots, X_n\}$ is also exponentially distributed

with rate $\lambda = \sum_{i=1}^{n} \lambda_i$. To see this, observe that the cdf of Z is given by

$$\begin{aligned}
\mathbb{P}(Z \leq t) &= 1 - \mathbb{P}(Z > t) \\
&= 1 - \prod_{i=1}^{n} \mathbb{P}(X_i > t) \\
&= 1 - \prod_{i=1}^{n} e^{-\lambda_i t} \\
&= 1 - e^{-\lambda t}.
\end{aligned}$$

2. **Memoryless property.** Suppose $X \sim \text{Exp}(\lambda)$, then the following holds and is called the *memorylessness* of the exponential distribution.

$$\mathbb{P}(X > t + s \mid X > t) = \mathbb{P}(X > s), \qquad t, s \geq 0.$$

To see this, observe that

$$\begin{aligned}
\mathbb{P}(X > t + s \mid X > t) &= \frac{\mathbb{P}(X > t + s)}{\mathbb{P}(X > t)} \\
&= \frac{e^{-\lambda(t+s)}}{e^{-\lambda(t)}} \\
&= \mathbb{P}(X > s).
\end{aligned}$$

1.2.3 Hierarchical Clustering

Hierarchical clustering is a widely used *unsupervised learning* technique. The goal behind unsupervised learning is to find patterns in data without being given training examples or feedback about performance. Using some measure of similarity between groups of

objects, a collection of nested clusters are produced, where the clusters at each level are created by merging the ones at the level below. At the lowest level, each cluster typically contains the individual objects being studied. This unique feature that lets one visualize data at different granularities, makes hierarchical clustering an extremely valuable tool for exploratory data analysis.

Suppose that $X = \{x_1, x_2, \ldots, x_n\}$ is the collection of n items that are being studied.

Definition 1.3. *A* ***cluster*** C *is defined as any non-empty subset of* X*. A collection of clusters* T *is called a* ***hierarchical clustering*** *if the following hold.*

1. $\bigcup_{C_i \in T} C_i = X$*, and*
2. *for any* $C_i, C_j \in T$*, only one of the following is true* ***(i)*** $C_i \subset C_j$*,* ***(ii)*** $C_j \subset C_i$*, or* ***(iii)*** $C_i \cap C_j = \emptyset$*.*

The hierarchical clustering T can be represented as a tree with n leaves where the internal vertices correspond to clusters in T with more than one item and the leaves correspond to the items $x_1, \ldots, x_n$. Figure 1.3 shows a hierarchical clustering of 4 items x_1, x_2, x_3, and x_4 and the corresponding tree representation.

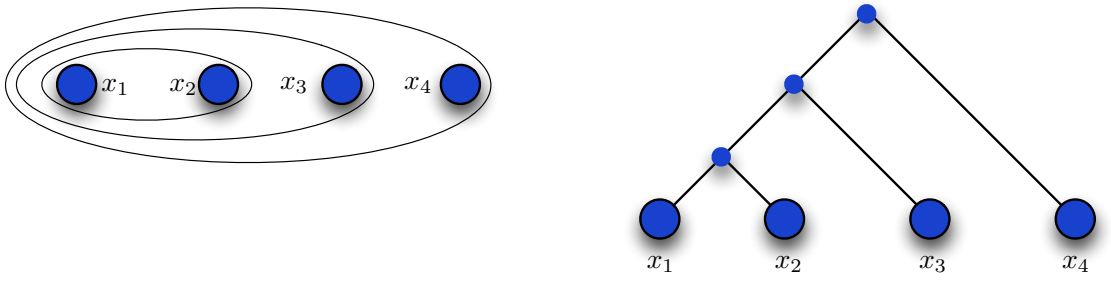


Figure 1.3: An example hierarchical clustering and its corresponding tree representation.

To produce a hierarchical clustering, there are two main families of strategies: bottom-up (or agglomerative), and top-down (or divisive). As the name suggests, bottom-up agglomerative clustering algorithms start at the bottom level and merge clusters recursively to reduce the total number of clusters by one till only one cluster is left. At each stage, two of the most similar clusters (according to some measure of similarity) are chosen and merged into a bigger cluster. On, the other hand, top-down algorithms start with one cluster containing all the items and at each step choose one of the clusters existing to split into two. This division is chosen to maximize some notion of inter-cluster dissimilarity. In general, bottom-up agglomerative clustering algorithms are *computationally* far more efficient than the top-down recursive division algorithms. In Chapter 3, we will revisit hierarchical clustering and focus on designing algorithms that are both robust and efficient in terms of the amount of data required.

See, e.g., [80] and the references therein for more on hierarchical clustering algorithms. There is also an interesting line of work on the theoretical properties of hierarchical clustering that we will not be able to get into in this dissertation. The reader is encouraged to see, e.g., [73, 32, 135] and references therein for more on this.

1.2.4 Covariance Estimation and Graph Reconstruction

Estimation of covariance matrices is a fundamental problem in multivariate statistics. In recent years, there has been exciting progress in addressing this problem in the so-called high dimensional regime, where the number of measurements available to the statistician is far fewer than the total number of variables present. In this regime, the traditional estimate – the empirical sample covariance matrix is known to behave poorly [83, 82].

Accordingly, to cope with the insufficiency of data, various "regularization" techniques have been considered in the literature. For instance, Dempster [36] suggests setting the elements of the inverse covariance matrix to zero as an appropriate form of regularization. When the data follows a multivariate normal distribution, zeros in the inverse covariance matrix implies that the corresponding pair of variables are conditionally independent. In this case, the set of non-zero entries correspond to the set of edges in an associated Gauss-Markov Random Field or Gaussian Graphical Model (see, e.g., [91] for more on this and other graphical models). Therefore, covariance estimation under these assumptions can be considered as a statistical graph estimation problem and there has been a long line of work in this area. For instance, [99] suggests solving a series of lasso regressions and shows that, under certain assumptions, this method consistently recovers the structure of the underlying graph from a limited number of samples. See, e.g., [62, 7, 140] for more on this. In this dissertation, we will not consider this notion of graph reconstruction further and the reader is encouraged to consult the articles cited above and references therein for more on this interesting area of active research.

Enforcing sparsity constraints on the covariance matrix is another notion of regularization for covariance estimation. Such "covariance graph" estimation has also received considerable attention in the literature (see, e.g., [43, 21, 6, 26, 12] and the references therein) and is much closer to the material we will encounter in Chapter 4. The theoretical framework developed in Chapter 4 is motivated partly by the problem of estimating a sparse covariance matrix or covariance graph in a data efficient way. In particular, we will see that, under certain conditions, it is possible to estimate this covariance graph even when given access to highly compressed samples.

1.2.5 Phylogenetic (Tree) Inference

Chapter 5 is based on the problem of reconstructing phylogenies from genetic data. Phylogenies, or phylogenetic relationships, are a depiction of the shared history of a set of species and the goal of phylogenetic inference is to posit a well justified hypothesis of this shared history. In many cases, learning the evolutionary history of a set of species might be the end goal of phylogenetic analysis, and in yet other cases, this shared history might provide important information for comparative studies.

Standard approaches for phylogeny reconstruction can be broadly divided intro three classes: parsimony based methods, likelihood based methods, and distance based methods. Parsimony methods, unlike the other two, are not based on any statistical assumption about evolution. Loosely speaking, these methods try to find the most parsimonious (tree based) explanation for the observed genetic data. Unfortunately, it is known that such techniques are in general computationally intractable [33, 61] and statistically inconsistent (i.e., if we suppose that the data did in fact come from a reasonable statistical model, the parsimony methods sometimes fail to converge on the right solution even when provided an unlimited amount of data) [57]. Likelihood based methods, roughly speaking, assume a statistical model and try to find a maximum likelihood estimate of the tree structure that explains the genetic data [58, 104]. While maximum likelihood is statistically consistent [20], it is still computationally intractable [114]. The final main class of reconstruction methods are the so called distance methods which work by estimating a tree metric on the leaves of the tree and using this to reconstruct the tree. These methods have the advantage of being provably consistent, and of being statistically and computationally efficient [50]. The algorithm we develop in Chapter 5 falls under

this category. See [121] and references therein for more on distance methods.

Chapter 2

Efficient Network Tomography for Network Topology Discovery

In this chapter we consider the problem of efficiently recovering the (logical) routing structure of a computer network. In particular, as in [112, 46], we assume that the logical routing structure of the network is in fact a tree and we endeavor to recover this tree from measurements made about this network. In keeping with the literature on this subject, throughout this chapter, we will use the words graph and network interchangeably and the we will use the word *topology* to mean the connectivity pattern of the computer network, that is, the edges of the graph. We will also use *end-hosts* (resp. router) interchangeably with the leaves (resp. internal vertices) of the routing tree that we seek to learn.

A large section of prior work has focused on Internet (router-level) topology discovery using the network diagnostic tool `traceroute`, e.g., [14, 39, 127]. The `traceroute` tool displays the route and the transit delays of packets across an Internet Protocol (IP) network. The route history is recorded as the round-trip times of the packets received from each successive node in the route. Information about the mean times in each hop are also recorded and relayed to the originating station. This tool is available on a number of modern operating systems including Mac OS, FreeBSD, Linux, and Windows.

The main limitation of such Time To Live (TTL) - limited (i.e., `traceroute` like) probes is the difficulty of reconstructing topologies in the presence of anonymous routers [138] and router aliases [69]. More recent research in [123, 122] has shown how a combination of `traceroute` and Record Route probes can improve the accuracy of topology estimation. However, these Record Route probes are limited in that only a small percentage of the routers connected to the Internet respond to the Record Route option. These TTL-limited techniques are also unable to discover the so-called "Level-2 routing" paths. Finally, all of these techniques, even when deployed to solve moderately sized topology discovery problems, would levy a load on the network that is typically unacceptable for normal operation; they are thus usually detected and blocked by network administrators.

As a result of these limitations of `traceroute` and Record Route probing techniques, significant research has been done recently on inferring Internet topology from end-to-end "tomographic" measurements of delay or packet loss. Initially, the focus of the community was on Multicast Tomographic inference, where a probe was sent from one node in the network to all the nodes whose topology one wished to reconstruct (hence, "multicast"). Then, roughly speaking, the delay or packet-loss information can be used to infer shared infrastructure between nodes; see, e.g., [46, 48] for more details. Multicast inference algorithms were attractive since they sidestepped the issues from the previous paragraph and since one typically only required $\mathcal{O}(n)$ multicast probes to do reliable topology inference. However, since only a small fraction of the IP space supports multicast, these techniques are not practical for wide-scale deployment. Instead, more recent work has focused on network tomography using unicast probes (i.e., probes are only addressed to a single node) [47, 96] which is not limited to small segments of IP space. The main

limitation of unicast inference has been the impractical number of probes ($\mathcal{O}(n^2)$) needed to do topology discovery.

In this chapter, our goal will be to reduce the number of probes required to recover the network topology. Towards this end, our strategy will be to arrange the end hosts in a *Depth-First Search (DFS) Order*. For a collection of end hosts in a tree topology, any of the ordinal lists found from a depth-first search on the end hosts of a tree structure is called a DFS ordering. This can also be considered a topological sort [86] on only the end hosts of a logical topology. This type of soorting has been explored previously in sensor network literature in [109], where a topological sort of the nodes in a sensor network provides efficient routes through the network with lower power consumption.

The main idea is as follows. A DFS ordering clusters end host based on the amount of shared infrastructure. Therefore, under the permutation corresponding to such an ordering, the resulting covariance matrix has a special structure which can be used to create a network topology discovery algorithm that is efficient in the number of pairwise delay measurements it makes. The number of such probes this algorithm needs to resolve the logical topology of a balanced $\ell-$ary tree can be shown to improve upon the number required by the current state-of-the-art [105]. Our assumptions on the underlying probabilistic model is also far less restrictive than the current state-of-the art. It is our belief that this reduction in the number of probes needed is an important step towards unicast tomography being considered a feasible and practical topology discovery mechanism.

2.1 Related Work

The work most directly related to the research in this chapter is the hierarchical clustering based methods explored in [46, 48, 124, 18]. This typically requires obtaining the entire covariance matrix ($O(n^2)$ measurements when there are n end-hosts in the topology). The hierarchical clustering method will be considered a baseline and the hope will be to greatly improve on its performance.

A more efficient probing algorithm is the Sequential Topology Inference algorithm from [105]. This algorithm sequentially builds the logical tree structure and leverages the current estimated logical tree structure to determine where the next probe pair measurements should be performed. This work couples topology inference and measurements by explicitly exploiting the tree structure. For a balanced $\ell-$ary tree (a balanced tree where each non-leaf node has exactly ℓ children) with n leaves, this algorithm reduces the number of probes needed from $\mathcal{O}(n^2)$ for agglomerative clustering, to at most $\ell n \log_\ell(n)$.

In what follows, we will see how improvements to this performance can be obtained by exploiting the structure of not just the tree topology, but the structure of the topology measurements. This helps further reduce the number of probes compared to the current state- of-the-art. Additionally, the Sequential Topology methodology requires strict conditions on the observed pairwise similarities. In Section 2.5, we present an efficient tomography method that requires much less restrictive conditions on the observed pairwise similarity values.

2.2 Depth-First Search (DFS) Order

We consider the standard tomography problem of resolving the logical tree topology rooted at an end host that is transmitting probes through the network. Common to most unicast tomography techniques is the ability to estimate a measure of similarity between pairs of end hosts. This similarity measure can vary depending on the particular probing technology used, such as, observed covariance for the Network Radar technique [133], or observed delay deviation for the sandwich probes method from [18]. The efficient topology reconstruction techniques presented in this chapter are agnostic to the choice of the tomography probing method. Therefore, for two end host x_i, x_j, we denote the observed pairwise similarity as s_{ij}, ignoring the mechanism used to generate the similarity. We will let the matrix of similarities be denoted by S. We also define the shared path matrix P which is such that p_{ij} denotes the number of logical routers shared between the paths that connect the root note to the two end hosts x_i and x_j. We say that the similarity matrix S and the shared path matrix P satisfy the *Monotonicity Condition* (MC) if for all triplets of end hosts x_i, x_j, x_k, $s_{ij} > s_{jk}$ holds if and only if $p_{ij} > p_{jk}$. Notice that this is related to the definition of an ultrametric in Section 1.2.

The foundation for the work in this chapter is the idea of a Depth-First Search (DFS) Ordering of the end hosts. A depth-first search (DFS) is a tree search that starts at the tree root and progresses down the tree, labeling each node and backtracking only when a node has been explored fully (i.e., every child of that node has been labeled). We formally define a DFS Ordering as any ordinal list of the end hosts (which will be considered the leaf nodes of the logical routing tree) that would satisfy the ordering found by a depth-first search of that logical tree structure ignoring the labeling of the

internal nodes of the tree. This notion has been considered previously in literature in a different context as a "topological sort" [86] of the leaf nodes of a tree.

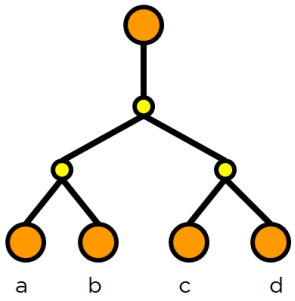


Figure 2.1: Example simple logical topology in a proper DFS Order.

For the tree structure in Figure 2.1, we can find the following valid DFS orderings all of which would satisfy a depth-first search on the tree topology:

$$\{a,b,c,d\} \quad \{a,b,d,c\} \quad \{b,a,c,d\} \quad \{b,a,d,c\}$$
$$\{c,d,a,b\} \quad \{d,c,a,b\} \quad \{c,d,b,a\} \quad \{d,c,b,a\}$$

There are also many possible end host orderings that would violate the DFS ordering property of the tree. For example, the ordering $\{a,c,d,b\}$ would result in a violation.

Now, under such an ordering, let us examine the shared path matrix P. When the matrix P is permuted according to a proper DFS ordering of the topology in Figure 2.1 ($\{a,b,c,d\}$), the resulting matrix P_{proper} takes the following form:

$$P_{\text{proper}} = \left[\begin{array}{c|cccc} - & a & b & c & d \\ \hline a & 2 & 2 & 1 & 1 \\ b & 2 & 2 & 1 & 1 \\ c & 1 & 1 & 2 & 2 \\ d & 1 & 1 & 2 & 2 \end{array}\right]$$

When permuted according to an improper DFS ordering ($\{a, c, d, b\}$) instead, the out-of-order shared path matrix (P_{improper}) takes the form:

$$\mathbf{S}_{improper} = \left[\begin{array}{c|cccc} - & a & c & d & b \\ \hline a & 2 & 1 & 1 & 2 \\ c & 1 & 2 & 2 & 1 \\ d & 1 & 2 & 2 & 1 \\ b & 2 & 1 & 1 & 2 \end{array}\right]$$

The Depth-first Search ordering therefore can be thought of as a clustering of the end hosts based on the amount of shared infrastructure (paths). We state this result formally below.

Proposition 2.1. *Suppose the set of end hosts $\{x_1, x_2, ..., x_n\}$ are in a proper DFS Ordering according to their underlying routing tree, then their shared path matrix P has*

the following structure:

$$p_{i,i+j} \geq p_{i,i+k} \quad \textit{for } 1 \leq j \leq k$$

Proof. Suppose instead that $p_{i,i+j} < p_{i,i+k}$ (for some pair $1 \leq j \leq k$). This means that the end hosts x_i, x_{i+k} have more shared infrastructure than x_i, x_{i+j}. That is, the depth at which x_i and x_{i+k} meet in the tree is deeper than the depth at which x_i and x_{i+j} meet. But by the depth-first search ordering, this would imply that $j > k$ as a depth-first search would encounter x_{i+k} before x_{i+j}, a contradiction. □

2.3 Logical Topology Discovery given a DFS Ordering

In this section, we will show that given a proper DFS ordering, it is straightforward to recover the structure of the tree using very few similarity measurements. Let $T = (V, E)$ be the unknown tree structure and let $\rho \in V$ be the root of this tree. This is the vertex from which we are dispatching packets to the n leaves $L = \{x_1, \ldots, x_n\}$ of the tree.

From Proposition 2.1 and the Monotonicity Condition (MC), it is clear that the similarity matrix S has a structure similar to the shared path matrix P.

Proposition 2.2. *Suppose the set of end hosts $\{x_1, x_2, ..., x_n\}$ are in a proper DFS Ordering, the similarity matrix S will have the following property:*

$$s_{i,i+j} \geq s_{i,i+k} \quad : \textit{for } 1 \leq j \leq k$$

Using the results of this proposition, we can now devise an efficient logical tree reconstruction procedure as shown in Algorithm 1 for a set of end hosts in DFS order with pairwise similarities satisfying the Monotonic Condition.

Algorithm 1 - DFS Ordered Logical Topology Discovery Algorithm

Given:

1. Set of observed pairwise similarities for end hosts in DFS order, $\{s_{1,2}, s_{2,3}, \ldots, s_{n-1,n}\}$.
2. Vertex set $\hat{V} = \{x_1, \ldots, x_n\}$.
3. Edge set $\hat{E} = \emptyset$; tree $\hat{T} = (\hat{V}, \hat{E})$.
4. merge nodes $V' = \{x_1, \ldots, x_n\}$.

Main Body: For $k = \{1, 2, ..., n-1\}$.

1. Find $\hat{j} = \arg\max_{j=1,\ldots,|V'|-1} s_{j,j+1}$.
2. Create new internal node x^*.
3. Add new node. $\hat{V} \leftarrow \hat{V} \cup \{x^*\}$.
4. Add new edges. $\hat{E} \leftarrow \hat{E} \cup \left\{V'(\hat{j}), x^*\right\} \cup \left\{V'(\hat{j}+1), x^*\right\}$
5. Update merge nodes. $V'(\hat{j}) = x^*$. $V'(\hat{j}+1) = \emptyset$.
6. Update similarity. $s_{\hat{j},\hat{j}+1} = 0$.

Proposition 2.3. *Suppose the set of end hosts are in a proper DFS Ordering, $n-1$ probes (corresponding to the the similarity values $s_{i-1,i}$ for $i = \{2, ..., n\}$) are sufficient to reconstruct the unknown logical topology using Algorithm 1.*

Proof. For each end host, bottom-up agglomerative clustering requires only knowledge of which other end host has the most shared topology. Given the Monotonicity Condition,

this is equivalent to finding the end host with the largest similarity magnitude. Typically, finding this entailed obtaining the entire similarity matrix (which corresponds to $\binom{n}{2}$ probes). However, given that the end hosts are in a DFS ordering and that the Monotonicity Condition holds, we know that for end host $x_i, i \in \{1, \ldots, n\}$, the end host x_{i-1} or the end host x_{i+1} is the one that shares the most amount of infrastructure. Therefore, using just the similarity values $s_{i-1,i}$ for $i = \{2, ..., n\}$ and the modified agglomerative clustering algorithm displayed as Algorithm 1, one can recover the entire topology. □

2.4 Margin Based DFS Ordering Estimation

Of course, Algorithm 1 assumes that the DFS ordering is available, which is unrealistic in any non-trivial problem. In this section, we will see that using carefully targeted similarity measurements, this DFS ordering can be efficiently determined. However, to resolve this ordering, a stronger assumption on the similarity matrix S needs to be made. We call this the Margin Condition (MC). Let $\delta > 0$. We say that the observed similarity matrix S satisfies the Margin Condition of order δ with respect to the shared path matrix P if for all triplets x_i, x_j, x_k of end-hosts, the observed similarities satisfies $s_{ij} > s_{jk} + \delta$ if and only if $p_{ij} > p_{jk}$.

Remark. *Delay-based unicast methods exploit the fact that the similarities correspond to the shared queuing delay between pairs of end hosts. Therefore, the value δ can be thought of as the minimum queuing delay covariance induced by a router in the network topology.*

The intuition behind the DFS ordering reconstruction method is as follows. Given an arbitrary ordering of the set of end hosts, consider choosing a single end host x_1 and obtaining the pairwise similarities between x_1 and all other end hosts (i.e.,

$\{s_{1,2}, s_{1,3}, \ldots, s_{1,N}\}$). Some end hosts will have very high pairwise similarity, while others will have significantly less shared infrastructure with the chosen end host and therefore have low pairwise similarity. Sorting these obtained similarity values in a descending order would place the end hosts that have more shared infrastructure at one end of the list, and the end hosts with little shared infrastructure at the other end of the list. Let $\pi : \{2, 3, \ldots, n\} \rightarrow \{2, 3, \ldots, n\}$ be the ordering induced by this sort, i.e., for all $i \geq 2, j \geq 1$, $s_{1,\pi(i)} \geq s_{1,\pi(i+j)}$. Notice that this partial ordering is of course not a proper DFS ordering. For instance, Figure 2.2-(Left) shows that there are regions of "uncertainty" where the full DFS ordering is not revealed.

However, this ordering was obtained from the vantage point of just one end-host. Can we bootstrap this with other vantage points? Consider Figure 2.2-(Right); notice that the Margin Condition guarantees that any similarity will be within a δ deviation of one of three similarity values $\{s_A, s_B, s_C\}$. This suggests a natural iterative procedure. Having correctly ordered the end hosts modulo these clouds of uncertainty, one can now try to order these subclusters. This could be achieved by taking a new vantage point inside this cluster, and then reordering the cluster based on the pairwise similarity values with this new vantage point. With this in mind, we propose a recursive method that, at each iteration, bisects the ordered set of end hosts into two topologically significant clusters. We then recurse on the smaller clusters with a similar procedure.

The simplest approach to this bisection problem is as follows. Given margin value δ, sort the similarity values and find the set I of all the possible bisection candidate end hosts, where $x_i \in I$ if the difference in the value between the $i-$th and the $(i+1)-$th

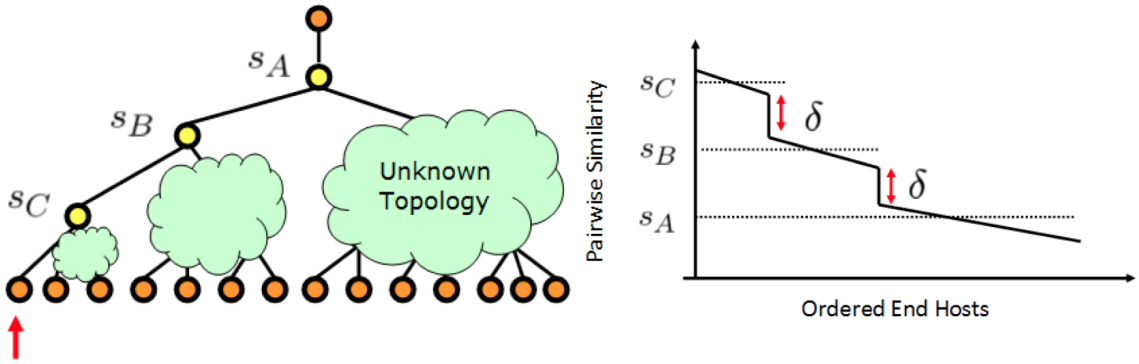


Figure 2.2: Margin-based End Host Ordering - (Left) Example of similarity values from a single end host revealing partial ordering, (Right) - Resulting ordered pairwise similarity values given the margin condition.

similarity values according to π exceeds δ. That is,

$$I = \left\{ i : s_{1,\pi(i)} - s_{i,\pi(i+1)} > \delta \right\}.$$

Notice that this is testing for whether there is a new cluster that begins at the end-host $x_{\pi(i)}$ with respect to x_1. The candidate bisection points will be the end hosts $i^* \in I$ that results in the two sets of end-hosts $X_1 = \{x_1, x_{\pi(2)}, \ldots, x_{\pi(i^*)}\}$ and $X_2 = \left\{x_{\pi(i^*+1)}, \ldots, x_{\pi(n)}\right\}$ being the closest in size to each other. Using this intuition, we present Algorithm 2 in order to find a proper DFS Ordering for a set of end hosts using this recursive bisection methodology. This algorithm comes with the following guarantee, the proof of which we will see in Appendix A.

Proposition 2.4. *Suppose that the unknown routing tree is a balanced $\ell-$ary tree (where each non-leaf node has ℓ children) with n leaves (end hosts) and suppose that the similarity matrix satisfies a Margin Condition of order δ. Algorithm 2 can recover the proper DFS ordering using no more than $p(\ell) n \log_\ell n$, where $p(\ell) \triangleq \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right)$.*

Algorithm 2 - Margin based DFS Ordering Algorithm - marginOrder($\mathbf{X}$, δ)

Given:

1. Unordered set of end hosts with unknown logical topology $\mathbf{X} = \{x_1, \ldots, x_n\}$
2. $\delta > 0$, the similarity margin from the Margin Condition.

Main Body:

1. Find the pairwise similarities values w.r.t $x_1 : \{s_{1,2}, \ldots, s_{1,n}\}$. Sort these in a descending order and let pi denote the permutation that reflects this sort.
2. Find the set I of indices as follows $I = \Big\{i : s_{1,\pi(i)} - s_{1,\pi(i+1)} > \delta\Big\}$.
3. As described in the text, using an index $i^* \in I$, bisect the sorted set $\mathbf{X}$ into two sets most equal in size $\mathbf{X}_1$ and $\mathbf{X}_2$.
4. **If** $|\mathbf{X}_1| > 1$, **then:** $\mathbf{X}_1 \leftarrow$marginOrder$(\mathbf{X}_1, \delta)$.
5. **If** $|\mathbf{X}_2| > 1$, **then:** $\mathbf{X}_2 \leftarrow$marginOrder$(\mathbf{X}_2, \delta)$.
6. Return the final ordered set $\mathbf{X} \leftarrow [\mathbf{X}_1, \mathbf{X}_2]$.

Putting these pieces together gives us our margin-based DFS Ordering topology reconstruction algorithm. The first step in this procedure is to find the DFS ordering of the end hosts using Algorithm 2, and then this DFS ordering is used to resolve the final logical routing tree topology using Algorithm 1. Combining Propositions 2.4 and 2.3, we get the following theoretical guarantee for our topology reconstruction algorithm.

Theorem 2.1. *Using Algorithm 2 and Algorithm 1, the logical topology for a balanced $\ell-$ary tree with n end hosts can be reconstructed using at most $n(p(\ell)\log_\ell n+1)$ pairwise similarities, provided the similarities satisfy the margin condition.*

Note that our margin-based DFS Ordering topology reconstruction methodology has a pairwise probing upper bound that requires fewer pairwise similarities than the current state-of-the-art efficient tomography approach from [105], which also requires a margin condition on the pairwise similarities.

2.5 Monotonicity based DFS Ordering Estimation

In many real world tomography problems, the Margin Condition is too restrictive. Even if there is a margin, it is very unlikely that the network tomographer knows this margin. In this section, we will see a method for efficient reconstruction of the routing tree when only the Monotonicity Condition holds, i.e., for any triple of end hosts x_i, x_j, x_k, the pairwise similarities satisfy $s_{ij} > s_{jk}$ if and only if the corresponding shared path matrix satisfies $p_{ij} > p_{jk}$.

Our ordering estimation algorithm begins much like the margin-based approach above, where the ordering is found by a recursive bisection of the set of end hosts. But, we will use a small number of additional pairwise similarities to reinforce each bisection choice.

Given an arbitrary ordering of the set of end hosts $\{x_1, \ldots, x_n\}$, let us again choose a single end host x_1 and obtain the similarity measurements $\{s_{1,2}, \ldots, s_{1,n}\}$ between x_1 and all other end hosts in the set. We will then use these to define, as before, the permutation $\pi : \{2, \ldots, n\} \rightarrow \{2, \ldots, n\}$ that corresponds to a descending order sort of these similarity values. As in the previous section, the goal will again be to reduce the total number of pairwise measurements needed for recovering the ordering by bisecting the ordered set of end hosts repeatedly and then recursing within each of the partitioned sets. However, since the margin condition no longer holds, one cannot confidently relate the jumps in the similarity values to candidate bisection points. To get around this, we devised a different procedure that allows us to find the bisection point. To understand this procedure, imagine that we are performing standard bottom-up agglomerative clustering on the set of ordered end hosts. The algorithm proceeds by successively merging similar groups of end hosts until all the end hosts find themselves in a single cluster. Prior to the final merge operation, the set of ordered end hosts find themselves in two partitions and our goal will be to recover this partition. However, we cannot use the standard algorithms for doing this since they require the examination of all the $\binom{n}{2}$ pairwise similarities. Instead, in what follows, we see that one can use the information contained in the permutation π to find this split with far fewer than $\mathcal{O}(n^2)$ similarities .

The basic idea behind this procedure is as follows. Consider performing agglomerative clustering on only a subset U of m end hosts evenly spaced in the ordering π as in Figure 2.3-(Left). The final merge of the agglomerative clustering on these m end hosts (Figure 2.3-(Center)) will reveal which size $\frac{n}{m}$ subset contains the bisection point x^*.

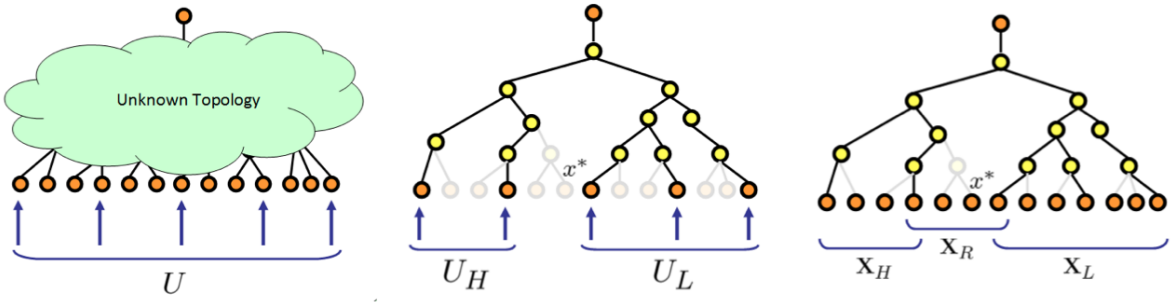


Figure 2.3: (**Left**) Set of end hosts in partial order, with subset of end hosts U given $m = 5$. (**Center**) Tree structure found through agglomerative clustering on the set U, with top-level split partitioning into U_H,U_L. (**Right**) Resulting end host sets X_R, X_L, X_H.

Once this subset is revealed, we choose m new end hosts inside this subset (the set X_R in Figure 2.3-(Right)) and again perform agglomerative clustering to find a subset of interest, this time of size $\frac{n}{m^2}$ again containing the bisection point x^*. This process is repeated until the subset of interest is of size less than or equal to m, after which, we perform one last agglomerative clustering step to resolve the top most split in the tree and thus find the bisection point x^*.

This recursive agglomerative clustering method is stated in Algorithm 3. The advantage of this modified clustering algorithm is of course in the savings it generates. In particular, it is not hard to see that Algorithm 3 uses no more than $m^2 \log_m n$ pairwise similarities.

Algorithm 3 can now be repeated multiple times to find a true DFS ordering on the end hosts. This complete algorithm is displayed as Algorithm 4. Algorithm 4 comes with the following performance guarantee, the proof of which can be found in Appendix A.

Theorem 2.2. *Using Algorithm 4 and Algorithm 1, the logical topology for a balanced $\ell-$ary tree with n end hosts can be reconstructed using no more than $n((\ell+9)\log_2 n + 1)$*

Algorithm 3 - Recursive Agglomerative Clustering Algorithm - recursiveAgg($\mathbf{X}$, m)

Given:

1. Set of end hosts that are partially ordered $\mathbf{X} = \{x_1, \ldots, x_n\}$
2. Parameter m which is the number of hosts that are exhaustively clustered.

Main Body:

1. Pick m uniformly spaces end hosts in $\mathbf{X}$ and call this U.
2. Use bottom up agglomerative clustering on U. Let the last merge of this agglomerative clustering be between U_L and U_H.
3. Find the largest similarity in the 'lower' set $s_L^{\max} \leftarrow \max_{i \in U_L} s_{1,i}$. and call the corresponding end host $x_L^{\max}$. Similarly find the smallest similarity in the 'higher' set $s_H^{\min} \leftarrow \min_{i \in U_H} s_{1,i}$. and call the corresponding end host $x_H^{\min}$.
4. Divide $\mathbf{X}$ into three subsets as follows: the 'higher' set $\mathbf{X}_H \leftarrow \left\{x_i : s_{1,i} \geq s_H^{\min}\right\}$, the 'lower' set $\mathbf{X}_L \leftarrow \{x_i : s_{1,i} \leq s_L^{\max}\}$, and the 'residual' set $\mathbf{X}_R \leftarrow \{\mathbf{X} \setminus \{\mathbf{X}_H \cup \mathbf{X}_L\}\} \bigcup \left\{x_L^{\max}, x_H^{\min}\right\}$.
5. **If** $|\mathbf{X}| \leq m$, **then:** return $x_H^{\min}$.
   Else, Return recursiveAgg($\mathbf{X}_R, m$).

pairwise similarities, provided the similarity matrix satisfies the Monotonicity Condition.

2.6 Experiments

2.6.1 Prior Methods

Hierarchical Clustering

Consider having access to every pairwise similarity value for all n end hosts in the topology. Given complete knowledge of the similarity matrix, we would have knowledge

Algorithm 4 - Monotonicity-Based DFS Ordering Algorithm - monotonicOrder($\mathbf{X}$)

Given:

1. Unordered set of end hosts with unknown logical topology $\mathbf{X} = \{x_1, \ldots, x_n\}$
2. Parameter m which is the number of hosts that are exhaustively clustered.

Main Body:

1. Find the pairwise similarities values w.r.t x_1 : $\{s_{1,2}, \ldots, s_{1,n}\}$. Sort these in a descending order and let pi denote the permutation that reflects this sort.
2. Using Algorithm 3, find the bisection index i^*.
3. Create sorted end host subsets $\mathbf{X}_1 = \{x_1, x_{\pi(1)}, \ldots, x_{\pi(i^*)}\}$, and $\mathbf{X}_2 = \{x_{\pi(i^*)+1}, \ldots, x_{\pi(n)}\}$.
4. **If** $|\mathbf{X}_1| > 1$, **then:** $\mathbf{X}_1 \leftarrow$monotonicOrder$(\mathbf{X}_1, m)$.
5. **If** $|\mathbf{X}_2| > 1$, **then:** $\mathbf{X}_2 \leftarrow$monotonicOrder$(\mathbf{X}_2, m)$.
6. Return the final ordered set $\mathbf{X} \leftarrow [\mathbf{X}_1, \mathbf{X}_2]$.

of which set of end hosts have the largest covariance in the entire topology (within some margin δ), and hence, which set of end hosts have the most shared infrastructure from the root node. For the bottom-up agglomerative clustering algorithm (applied in a networking context, e.g., in [46, 48, 18]), at each step of the algorithm the current set of end hosts with the largest similarity are found, and a logical router is inserted connecting this set of end hosts together. The corresponding rows/columns in the similarity matrix for these two end hosts are then merged together. This process is repeated until there are no more rows/columns in the matrix are left to merge. The main disadvantage to this methodology is that it requires the tomographer to acquire all the $\binom{n}{2}$ similarity values.

Sequential Logical Topology

Informed by the generic tree structure of the topology, the work in [105] shows that the number of probes needed to reconstruct the topology can be considerably reduced when the observed similarities satisfy the margin condition. This algorithm works by sequentially building the tree topology for each end host. For a given end host, the pairwise similarity for this end host and all the nodes that are children of the root node are found. Given the child of the root node with the largest similarity (and thus the most shared topology), c_i^*, the pairwise similarity is found between the end host and the children of the specified child (c_i^*). The similarity value (and margin δ) determines whether the end host is a sibling, child, or descendant of c_i^*. This process is repeated until the leaf node with the largest pairwise similarity is found. On a balanced $\ell-$ary tree, each end host requires at most $\ell \log_\ell n$ pairwise probes, and thus for the entire topology the number of pairwise probes needed is upper bounded by $\ell n \log_\ell n$.

Theoretically, the margin-based DFS ordering algorithm (Algorithm 2 + Algorithm 1)

is found to have the smallest probing complexity upper bound of all algorithms, but requires the restrictive margin condition on the observed similarity values. To compare with the Sequential Topology algorithm ([105]), which also requires the margin condition, we observe that $\frac{p(\ell)}{\ell} \leq 0.5625$ for all choices of l.

The monotonicity-based DFS ordering algorithm (Algorithm 4 + Algorithm 1) has upper bounds requiring more probes than the two margin-based methodologies, but functions under much weaker assumptions. Of course, the monotonicity-based DFS ordering algorithm requires significantly fewer pairwise probes than the agglomerative clustering methodology, which is the only other procedure that is also guaranteed to reconstruct topologies under the monotonic condition.

2.6.2 Synthetic Noise-Free Experiments

Synthetic topologies enable us to analyze the capabilities of the methods we propose with full ground truth and over a range of network sizes. The synthetic topologies are generated using the Heuristically Optimized Topology framework [2]. Heuristically Optimized Topology is one of the latest and most realistic network topology generators that create graphs that have properties that are consistent with many of those observed in the Internet, incorporating societal, engineering, and economic constraints. Using the Orbis topology generator [97], we scale the Heuristically Optimized Topologies to create three different sized topologies, with $n = \{768, 1497, 2261\}$.

To test the performance of our algorithms in a noise-free environment, for each synthetic topology every node is assigned a random similarity value, with synthesized pairwise similarities being the sum of the node similarity values (with the smallest router

similarity assigned, $\delta = 0.1$) along the shared shortest path from the root node to the two end hosts under consideration. Due to this experiment being noise-free with similarities that satisfy the margin condition, all topology reconstruction methodologies will perfectly reconstruct the topologies from the pairwise measurements.

In Table 2.1, we present the number of pairwise probes needed by the prior tomography methodologies, agglomerative clustering and Sequential. By exploiting the tree structure, the state-of-the-art Sequential method requires at most 20% of the pairwise probes the agglomerative cluster methodology requires. In Table 2.2, we present the resulting number of pairwise similarities required to resolve the logical topology for our monotonicity-based and margin-based DFS Ordering algorithms. As seen in the tables, both the DFS Ordering methodologies do significantly better than the exhaustive agglomerative clustering approach. In terms of the 768-end host topology, we obtain a pairwise probe savings with both DFS methodologies requiring at most 2% of the pairwise probes used by the exhaustive agglomerative clustering approach. As expected, the margin-based DFS algorithm consistently requires fewer pairwise similarities than the monotonicity-based DFS methodology. From these experiments it was seen that our new margin-based DFS method requires at most 10% of the number of pairwise similarities that the Sequential method require. Meanwhile, even the Monotonic DFS methodology (which does not require the margin condition) outperforms the Sequential methodology by requiring at most 15% of the number of pairwise probes used by the state-of-the-art approach.

	Agglomerative Clustering	Sequential Algorithm [105]	
Number of end hosts (n)	# Pairwise sim. needed	# Pairwise sim. needed	Percentage of Agglomerative Pairs
768	294,528	52,774	17.9 %
1497	1,119,756	112,375	10.0 %
2261	2,554,930	128,104	5.0 %

Table 2.1: Comparison of the number of probes needed to reconstruct synthetic Orbis topologies between the prior algorithms.

	Margin-Based DFS Ordering Algorithm			Monotonicity-Based DFS Ordering Algorithm		
# End hosts (n)	# Pairwise sim. needed	% Agg. Probes	% Sequential probes	# Pairwise sim. needed	% Agg. Probes	% Sequential probes
768	3,604	1.22 %	6.83 %	5,511	1.87 %	10.44 %
1,497	7,771	0.69 %	6.92 %	10,571	0.94 %	9.41 %
2,261	11,848	0.46 %	9.25 %	17,929	0.70 %	14.0 %

Table 2.2: Comparison of number of probes needed to estimate logical topology of the synthetic orbis topologies. The Margin-based and Monotonicity-based algorithms are compared against the standard agglomerative clustering algorithm and the Sequential [105] algorithm.

2.6.3 Real World Experiments

To observe the performance of our algorithm on real-world topologies, we chose 9 DNS servers located at small-to-medium sized colleges in the New England geographic area. Using the DNS server addresses and `traceroute` probes we discovered the logical tree topology shown in Figure 2.5 starting with University of Wisconsin-Madison as the root node. Using delay-based unicast tomographic probes, the pairwise similarities were found between pairs of the end hosts in the topology.

The delay-based unicast tomographic technique that we used is Network Radar [133]. Network Radar uses round trip time (RTT) measurements as the basis for topology inference and was developed as an attempt to obviate the need for significant coordinated

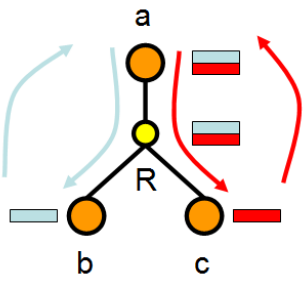


Figure 2.4: Example of Network Radar on simple logical topology.

measurement infrastructure. Consider the simple logical topology in Figure 2.4. Now suppose that two back-to-back packets are dispatched from the end-host a, one addressed to b and the other addressed to c. Both packets originating from end host a will encounter the same path until the internal node R, which depicts a router. It can be assumed that any delays encountered before router R induced by router queuing delays will cause highly correlated delays for both back-to-back packets (due to both packets being in the same router queues). It is also reasonable to assume that any delays encountered between the two packets past router R are uncorrelated. Therefore, we use the level of covariance between the round-trip-time (RTT) delays found from a series of back-to-back packets ($cov\,(d_b, d_c)$) as an estimate for the amount of shared logical topology between paths $\{a, b\}$ and $\{a, c\}$. We refer the reader to [46, 48, 124, 18], for instance, for more on this tomographic technique.

Using the Network Radar tool, we observed 1,500 back-to-back round-trip-time delay samples for every end host pair in our real-world topology (Figure 2.5) . Due to imperfect round-trip-time measurements and other delay noise measured, the sample similarity was found to not be perfectly correlated to the `traceroute` observed shared path length. Therefore, for any estimation procedure based on the sample similarity, there will be

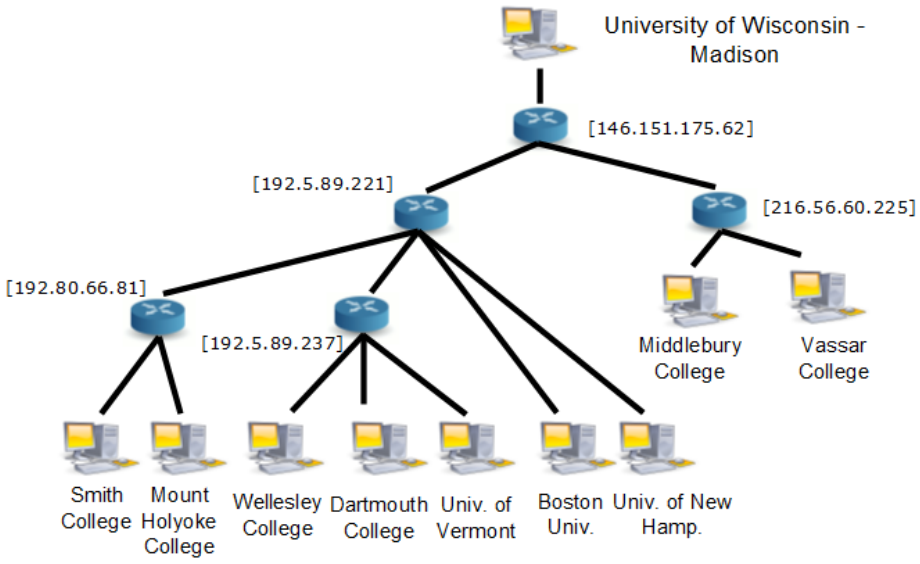


Figure 2.5: Real world topology used to test tomography methods

potential errors in the reconstructed topology [1] . In order to determine the accuracy of our estimated topologies, we must develop a metric that compares our estimated topologies to the ground-truth topology.

Informed by our monotonicity condition, we consider the following accuracy measure for our reconstructed tree topologies. Consider a triple of end hosts $\{x_a, x_b, x_c\}$. From our estimated logical topology, we can make a determination as to whether there is a longer shared path between end hosts x_a and x_b or between the end hosts x_a and x_c. Let us denote the two estimated paths as $\hat{p}_{ab}$ and $\hat{p}_{ac}$ respectively. Let p_{ab} and p_{ac} denote the paths in the true topology. The more accurate our estimated topology, the more often we will be able to make the correct determination that as to whether x_a and x_b have more shared infrastructure or is it x_a and x_c. The percentage of triplets for which we are

[1] This could be improved upon by taking more back-to-back sample probes or using a DAG card to obtain more accurate time information, but here we will assume that neither improvement is available.

correct, p, will be our performance measure.

$$p \triangleq f \sum_{x_a,x_b,x_c \in \binom{\mathbf{x}}{3}} \mathbb{1}\left\{\hat{p}_{ab} > \hat{p}_{ac}\right\} \mathbb{1}\left\{p_{ab} > p_{ac}\right\}, \tag{2.1}$$

where $f = \left(\sum_{x_a,x_b,x_c \in \binom{\mathbf{x}}{3}} \mathbb{1}\left\{\hat{p}_{ab} > \hat{p}_{ac}\right\} \mathbb{1}\left\{p_{ab} > p_{ac}\right\}\right)^{-1}$ is the normalization constant, and $\mathbb{1}\{\}$ is the indicator function.

The baseline for any topology reconstruction algorithm will be to outperform a naive randomly reconstructed topology with end hosts and interior nodes connected at random into a tree topology. Our shared path classification rate (2.1) will be the metric we use to assess how accurate the estimated topologies are.

Results

The Sequential Algorithm and both the DFS Ordering algorithms depend on the initial ordering of the end hosts, to some extent. So, we averaged over 500 random permutations of the end hosts to eliminate any order bias from the results. The two margin-based methodologies (Sequential and margin-based DFS ordering) have a tunable margin parameter,δ, that must be chosen. To give the prior margin-based method (Sequential) every possible advantage, for each experiment, the performance of the Sequential algorithm is shown for the best possible value of δ at each level of probing. Meanwhile, our new margin-based DFS ordering methodology has a constant value of δ across all levels of probing (with $\delta = 0.1$).

For the real-world topology in Figure 2.5, the corresponding shared path classification rate (from Equation (2.1)) for the two margin-based topology reconstruction algorithms (margin-based DFS Ordering and Sequential) and a baseline random random methodology

can be seen in Figure 2.6-(Left) versus a restricted total number of delay probes available. We find that the margin-based DFS ordering methodology performs significantly better than both the Sequential algorithm and the random algorithm. Surprisingly, the Sequential methodology requires a large number of pairwise probes to outperform the random topologies, we believe this is due to a heavy reliance on the margin δ between the similarity measurements. While our margin-based methodology also requires the margin condition, through the exploitation of DFS ordering, our methodology appears to be more robust to violations of the margin condition. Given the theoretical probing complexity, it is very likely that the accuracy improvements for the new DFS Ordering algorithm will further grow as the size of the topology increases.

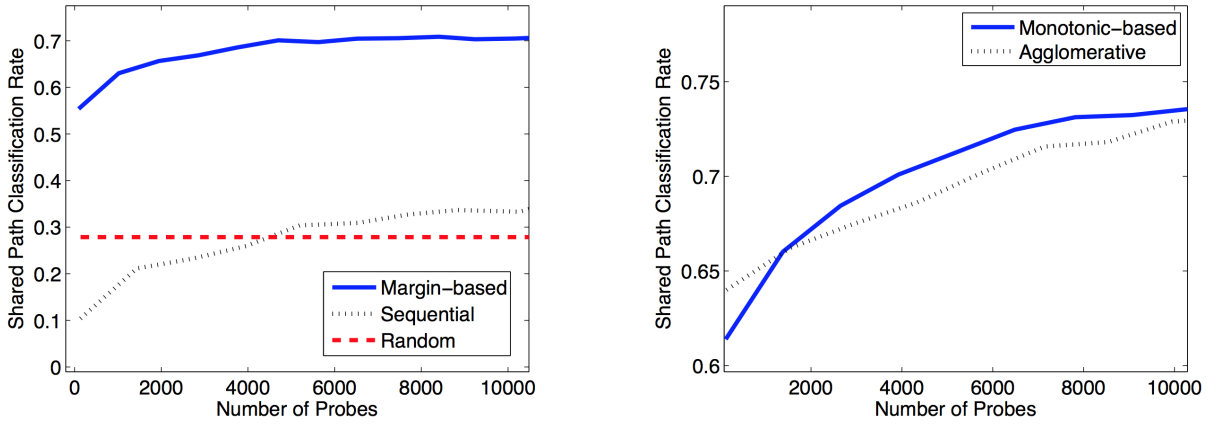


Figure 2.6: (**Left**) Topology reconstruction results for the two margin-based algorithms (margin-based DFS Ordering, Sequential) and a baseline random topology. (**Right**)Topology reconstruction results for the two monotonic-based algorithms (monotonic-based DFS Ordering and agglomerative clustering).

The shared path classification rate (from Equation (2.1)) for the two monotonicity-based topology reconstruction algorithms (monotonicity-based DFS Ordering and agglomerative clustering) can be seen in Figure 2.6-(Right) plotted against a restricted

total number of delay probes available. As seen in the figure, for a wide range of available pairwise probes, our monotonicity-based DFS ordering algorithm results in a more accurate tree topology than the standard bottom-up agglomerative clustering methodology. For example, to obtain the same tree reconstruction accuracy ($p = 0.7$), the monotonic-based DFS methodology requires 1,700 fewer delay-based measurements sent through the network compared with the exhaustive agglomerative clustering algorithm (3,800 delay probes suffices for the DFS ordering algorithm, while agglomerative clustering requires 5,600 delay probes).

2.7 Discussion

Despite a concerted in the network mapping and measurement community, generating accurate maps of the router-level topology of the Internet still remains a compelling open problem in Internet measurement. Standard TTL-limited and Record Route methods for discovering router-level network topology have well known limitations that motivate the development of alternative topology reconstruction methods. One such method is tomographic inference using network delay measurements. While network tomography for topology discovery has been examined in the past, it has yet to be widely used in practice due to it's own set of limitations.

The goal of the work presented in this chapter was to address the shortcomings of RTT measurement-based network tomography for discovering Internet logical topology. Prior procedures for tomographic inference were based on restrictive assumptions or required an impractical number of probes. In this chapter, we saw algorithms that considerably reduce the number of pairwise probes needed to resolve logical topologies. Our ability to

reduce the number of pairwise probes stems from a careful exploitation of a Depth-First Search (DFS) Ordering of the end hosts. We analyzed the capabilities of our algorithms on a set of large-scale synthetically generated topologies. The experiments on these topologies show our new algorithms require only 2% of the probes used by an exhaustive methodology, and roughly 15% of the probes used by the current state-of-the-art. Results from a small-scale real-world Internet experiment further validate the performance of our algorithms. The significant reduction in the number of probes needed opens delay-based tomography techniques for network topology discovery to new avenues of applications.

Bibliographic Note. The material in this chapter is based on joint work with Brian Eriksson, Paul Barford, and Rob Nowak. [51] contains a preliminary version of this work and [52] contains the full version.

Chapter 3

Active Clustering: Robust and Efficient Hierarchical Clustering.

3.1 Introduction

In Chapter 2, we saw efficient algorithms for discovering the tree structured (router level) topology of the internet. This can be thought of as an efficient algorithm for *clustering* computers on a network based on the amount of network topology they share. In the current chapter, we will concretize this intuition and demonstrate how one can develop efficient and robust algorithms for the more abstract problem of hierarchical clustering of arbitrary objects.

Hierarchical clustering based on pairwise similarities arises routinely in a wide variety of social science, scientific, and engineering problems. These problems include inferring gene behavior from microarray data [139], Internet topology discovery [105], sociology [64], and advertising [128]. It is often the case that there is a significant cost associated with obtaining each similarity value. For example, as we saw in Chapter 2, in the case of Internet topology inference, the determination of similarity values requires many probe packets to be sent through the network, which can place a significant burden on the network resources. In other situations, determining how similar a pair of items are might

require expensive experiments or require the input of a human expert, again placing a significant cost on their collection.

The potential cost of obtaining similarities motivates a natural question: Is it possible to reliably cluster items without using all the pairwise similarities? In this chapter, we will see that the answer is yes, particularly under the condition that intracluster similarity values are greater than intercluster similarity values, which we define as the *Tight Clustering Condition*. While the condition may not be satisfied in all situations, it is natural and realistic for problems in which the similarity values are generated by a probabilistic model on a tree in which, say, the dissimilarity between items is a monotonically increasing function of the distance to their nearest common branch point (ancestor). This situation arises naturally in clustering nodes in the Internet [110] or biological species (i.e., phylogenetics) [60]. We then weaken this assumption and extend our results to the case when a significant fraction of the similarities fail to meet this Tight Clustering condition.

As a first question, one might naturally ask whether a randomly chosen subset of similarity values might suffice to reconstruct the hierarchy. We show that this is not the case generally. Instead, we propose an *active* approach that selects similarities in an adaptive fashion. Hence, we call our procedure *active clustering*. In what follows, we will see that under the Tight Clustering condition, our algorithm reliably determines the hierarchical clustering of n items using no more than $3n \log n$ pairwise similarities. A standard clustering procedure would need $n(n-1)/2$ pairwise similarities by comparison. Since it is clear that we must obtain at least one similarity for each of the n items, this is quite close to as good as one could hope to do. As mentioned earlier, the Tight Clustering condition may be too restrictive in practice. To broaden the applicability of the proposed

theory and method, we propose a novel active clustering algorithm for situations where a random subset of the similarities fail to meet this Tight Clustering condition. In this case, we show how using only $\mathcal{O}\left(n \log^2 n\right)$ actively chosen pairwise similarities, we can recovery the underlying hierarchical clustering with high probability.

While there have been some prior attempts at developing robust procedures for hierarchical clustering [5, 88, 68], these works do not try to optimize the number of similarity values needed to robustly identify the true clustering, and mostly require all $\mathcal{O}(n^2)$ similarities. It appears as though the general problem of hierarchical clustering using only a subset of similarities has not been considered before. However, some interesting connections emerge between this problem and prior work on graphical model inference [105, 107], which we further exploit here.

3.2 The Hierarchical Clustering Problem

Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a collection of n items. Our goal will be to recover a *hierarchical clustering* of these items.

Definition 3.1. *A **cluster** C is defined as any non-empty subset of X. A collection of clusters T is called a **hierarchical clustering** if the following hold.*

1. $\cup_{C_i \in T} C_i = X$, *and*
2. *for any $C_i, C_j \in T$, only one of the following is true **(i)** $C_i \subset C_j$, **(ii)** $C_j \subset C_i$, or **(iii)** $C_i \cap C_j = \emptyset$.*

The hierarchical clustering T can be represented as a tree, where each vertex corresponds to a particular cluster. The tree is said to be *binary* or *bifuricating* if every

internal vertex has degree 3. In terms of the clustering, this implies that for every $C_k \in T$ that is not a leaf of the tree, there exists proper subsets C_i and C_j of C_k, such that $C_i \cap C_j = \emptyset$, and $C_i \cup C_j = C_k$. The binary tree is said to be *complete* if it has n leaf nodes, each corresponding to one of the individual items. Without loss of generality, we will assume that T is a complete (possibly unbalanced) binary tree, since any non-binary tree can be represented as a binary tree (e.g., a merging of three clusters can be expressed as a sequence of two pairwise mergings).

Let $\mathbf{S} = \{s_{i,j}\}_{i,j\in[n]}$, where $[n] = \{1, 2, \ldots, n\}$ denote the collection of all pairwise similarities between the items in $\mathbf{X}$; (i.e., $s_{i,j}$ denotes the similarity between x_i and x_j) and we assume $s_{i,j} = s_{j,i}$. The traditional hierarchical clustering problem uses this complete set of similarities to infer T. In order to guarantee that T can be correctly identified from $\mathbf{S}$, the similarities must conform to the hierarchy of T. We consider the following sufficient condition.

Definition 3.2. *The triple* $(\mathbf{X}, T, \mathbf{S})$ *satisfies the* ***Tight Clustering (TC) Condition*** *if for every set of three objects* $\{x_i, x_j, x_k\}$ *such that* $x_i, x_j \in C$ *and* $x_k \notin C$*, for some* $C \in T$*, the pairwise similarities satisfies,* $s_{i,j} > \max\{s_{i,k}, s_{j,k}\}$.

In words, the TC condition implies that the similarity between all pairs within a cluster is greater than the similarity of any item within a cluster with an item outside the cluster. It is easy to see that if the TC condition is satisfied, then the standard bottom-up agglomerative clustering algorithms such as single linkage, average linkage and complete linkage will produce T , given the complete similarity matrix $\mathbf{S}$ [74]. Agglomerative clustering is a recursive process that begins with singleton clusters (i.e., the n individual items to be cluster), and at each step the pair of most similar clusters are merged. The

process is repeated until all items are merged into a single cluster. Various agglomerative clustering algorithms differ in how the similarity between two clusters is defined, but every agglomerative clustering technique requires all $n(n-1)/2$ pairwise similarity values since all similarities must be compared at the very first step.

To properly cluster the items using fewer similarities requires a more sophisticated adaptive approach where similarities are carefully selected in a sequential manner. Before contemplating such approaches, we first demonstrate that adaptivity is necessary, and that simply picking similarities at random will not suffice.

Proposition 3.1. *Let T be a clustering of n objects such that there exists a set of k clusters of size m in T for some $m < n$ such that $k \times m = \Theta(n)$. If $r < n(n-1)/m$ similarities are chosen uniformly at random from S, then any clustering procedure will fail to recover the entire hierarchical clustering with high probability.*

Proof. Consider one of the k clusters, C. In order for any procedure to identify C, we need to see at least $m-1$ of the $\binom{m}{2}$ similarities between items in C. Therefore, we can assume that $r > (m-1)$ without loss of generality. Let $p = \binom{m}{2}/\binom{n}{2}$, which is the probability that a randomly chosen similarity value will be from C and let $\delta := \frac{(m-1)}{r} - p$. Observe that our assumption on r is equivalent to saying $\delta > 0$. Now, if we uniformly sample r similarities (with replacement), then the expected number of similarities we would see between items in C is rp. If we let rX be the actual number of similarities we

observe between items in C, then we have the following.

$$\begin{aligned} P\left(\text{Any procedure succeeds}\right) &\leq P\left(rX > (m-1)\right) \\ &= P\left(X - p > \delta\right) \\ &\overset{(a)}{\leq} \exp(-2r\delta^2) \\ &\overset{(b)}{\leq} \exp(-2(m-1)\delta^2) \end{aligned}$$

where (a) follows from Hoeffding's inequality and (b) follows from the fact that we also need r to be bigger than $(m-1)$. Therefore, the probability that any hierarchical procedure would fail in recovering all of the k clusters is lower bounded by $1 - \exp(-2k(m-1)\delta^2)$ which approaches 1 rapidly as n grows. □

This result shows, for example, that if $m = \log n$, then the number of randomly selected similarities must exceed $n(n-1)/\log n$. This means that almost all of the $\binom{n}{2}$ similarities are required.

3.3 Active Hierarchical Clustering under the TC Condition

From Proposition 3.1, it is clear that unless we acquire almost all of the pairwise similarities, reconstruction of the clustering hierarchy when sampling-at-random will fail with high probability. In this section, we demonstrate that under the assumption that the TC condition holds, an *active clustering* method based on *adaptively selected* similarities enables one to perform hierarchical clustering efficiently. Towards this end,

we consider an algorithm that appears in Appendix II.1 of [107] where the authors are concerned with a very different problem, namely, the identification of causal relationships among binary random variables. We present a slightly modified version in Algorithm 5 and then show how it relates to our hierarchical clustering setup.

From our discussion in the previous section, it is easy to see that the problem of reconstructing the hierarchical clustering T of a given set of objects $X = \{x_1, x_2, \ldots, x_n\}$ can be reinterpreted as the problem of recovering a binary tree whose leaves are $\{x_1, x_2, \ldots, x_n\}$. In what follows, I will use T to represent both the hierarchical clustering and the corresponding tree, interchangeably. In [107], the authors define a special type of test on triples of leaves called the *leadership test* which identifies the "leader" of the triple in question. A leaf x_k is said to be the *leader* of the triple (x_i, x_j, x_k) if the path from the root of the tree to x_k does not contain the nearest common ancestor of x_i and x_j (see Figure 3.1). They then proceed to show that one can reconstruct the entire tree T using only these leadership tests.

Therefore, if we are able to perform these leadership tests with respect to T in our

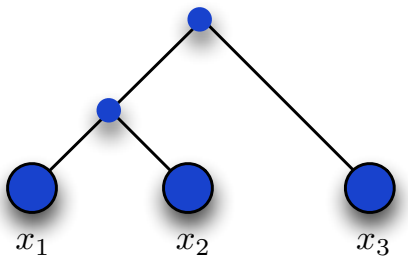


Figure 3.1: Tree structure where x_3 is the leader of the triple $\{x_1, x_2, x_3\}$.

setup, we can correctly reconstruct the hierarchical clustering T. The following lemma demonstrates that in fact, under the TC condition, we do have this ability. In particular, we show that this can be done by performing what we call the "outlier test" which only involves the pairwise similarities between the three objects.

Lemma 3.3. *Let* $\mathbf{X}$ *be a collection of items equipped with pairwise similarities* $\mathbf{S}$ *and a hierarchical clustering* T*. For any three items* $\{x_i, x_j, x_k\}$ *from* $\mathbf{X}$*, define*

$$\texttt{outlier}\,(x_i, x_j, x_k) = \begin{cases} x_i \ : \ \max(s_{i,j}, s_{i,k}) < s_{j,k} \\ x_j \ : \ \max(s_{i,j}, s_{j,k}) < s_{i,k} \\ x_k \ : \ \max(s_{i,k}, s_{j,k}) < s_{i,j} \end{cases} \tag{3.1}$$

If $(\mathbf{X}, T, \mathbf{S})$ *satisfies the TC condition, then* `outlier` (x_i, x_j, x_k) *coincides with the leader of the same triple with respect to the tree structure conveyed by* T*.*

Proof. From the definitions, we first observe that both the leader and the `outlier` of any triple (x_i, x_j, x_k) are unique. Now, let us suppose that x_k is the leader of the triple with respect to T. This can happen if and only if there is a cluster $\mathcal{C} \in T$ such that $x_i, x_j \in \mathcal{C}$ and $x_k \in T \setminus \mathcal{C}$ which automatically implies that $s_{i,j} > \max(s_{i,k}, s_{j,k})$ by the TC condition. Therefore x_k is the `outlier` of the same triple.

Conversely, it is easy to see that if x_k is the outlier of the triple, then no other leaf can be the leader since that will contradict the uniqueness of the `outlier`.

□

Using Algorithm 5 we perform targeted outlier tests for specific item triples in order to reconstruct the hierarchical clustering, T. The following theorem is the main result

Algorithm 5 - Outlier-based Clustering Algorithm

Given : A set of leaves, $\mathbf{X} = \{x_1, x_2, ...x_n\}$, and an initial tree with two (arbitrary) leaves, $T_2 = \{x_1, x_2\}$.
For each $x_i = \{x_3, x_4, ..., x_n\}$

1. Set $T_{sub} = T_{i-1}$ and $p =$ number of leaves in T_{sub}.
2. **While** $p > 2$
   (a) For any node v in T_{sub}, let des(v) denote the number of descendants of v. Pick v such that $\frac{p}{3} < \text{des}(v) \leq \frac{2p}{3}$, which is always possible (see proof of Theorem 3.4).
   (b) Find leaves x_j and x_k such that v is their nearest common ancestor.
   (c) Partition T_{sub} into two : T^1_{sub} is the tree rooted at v and all its descendants, and T^2_{sub} is $(T_{sub} \setminus T^1_{sub}) \cup u)$, where u is any one descendant leaf of v that replaces T^1_{sub} and is considered a leaf in T^2_{sub}.
   (d) Find the outlier of the triple (x_i, x_j, x_k).
   (e) **If** x_i is the outlier, then set $T_{sub} = T^2_{sub}$
   (f) **Else If** x_j (resp. x_k) is the outlier, then set T_{sub} to be the subtree rooted at the son of v which is the nearest ancestor of x_k (resp. x_j).
   (g) Set $p =$ number of leaves in T_{sub}.
3. **If** $p = 2$ **then**
   (a) Let the leaves be x_j and x_k and let their ancestor be v.
   (b) Find the outlier of the triple (x_i, x_j, x_k).
   (c) **If** x_j (resp. x_k) is the outlier, add a new node on the edge of x_k (resp. x_j) and make it the father of x_i.
   (d) **Else If** x_i is the outlier then create a new root and make x_i and v its children.
4. Set $T_i = T_{i-1}$ with x_i attached as above.

of this section and it shows that this adaptive procedure only requires on the order of $n \log n$ pairwise similarity measurements to exactly reconstruct the hierarchical clustering T.

Theorem 3.4. *Assume that the triple* $(\mathbf{X}, T, \mathbf{S})$ *satisfies the Tight Clustering condition where* T *is a complete (possible unbalanced) binary tree that is unknown, and* $\mathbf{X}$ *has* n *items. Then, Algorithm 5 recovers* T *exactly using at most* $3n \log_{3/2} n$ *adaptively selected pairwise similarity values.*

Proof. Consider constructing a sequence of trees $T_1, T_2, \ldots, T_n$ such that $T_n = T$ and an intermediate T_i would be the subtree of T containing the leaves $\{x_1, x_2, \ldots, x_i\}$. Suppose we have formed T_{i-1}. The location where x_i should be added to T_{i-1} (thus forming T_i) is determined as follows. First pick an interior node v from T_{i-1} such that $\frac{i}{3} < \text{des}(v) \leq \frac{2i}{3}$ where $\text{des}(v)$ is the number of leaves that are descendants of v (the existence of such a node follows from Lemma 1 in Appendix II of [107]). Let x_j and x_k be any two leaves of T_{i-1} such that their nearest common ancestor is v. Now, ask for the outlier of the triple (x_i, x_j, x_k). If x_j (or x_k) is the outlier, then repeat the process with the subtree of T_{i-1} rooted at the son of v that is the ancestor of x_k (or x_j). On the other hand, if x_i is the outlier, repeat the process with T_{i-1} where the entire subtree rooted at v is replaced by any descendant of v. An example of this pruning procedure is seen in Figure 3.2.

This process is continued until we have only 2 leaves, say x_j and x_k, left. We call their ancestor v. Now, we perform a final outlier test and if x_j (or x_k) is the outlier, we add a new node on the edge of x_k (or x_j) and make it the father of x_i. If, however, x_i is the outlier, then we create a new root and make x_i and v its children. Notice that by

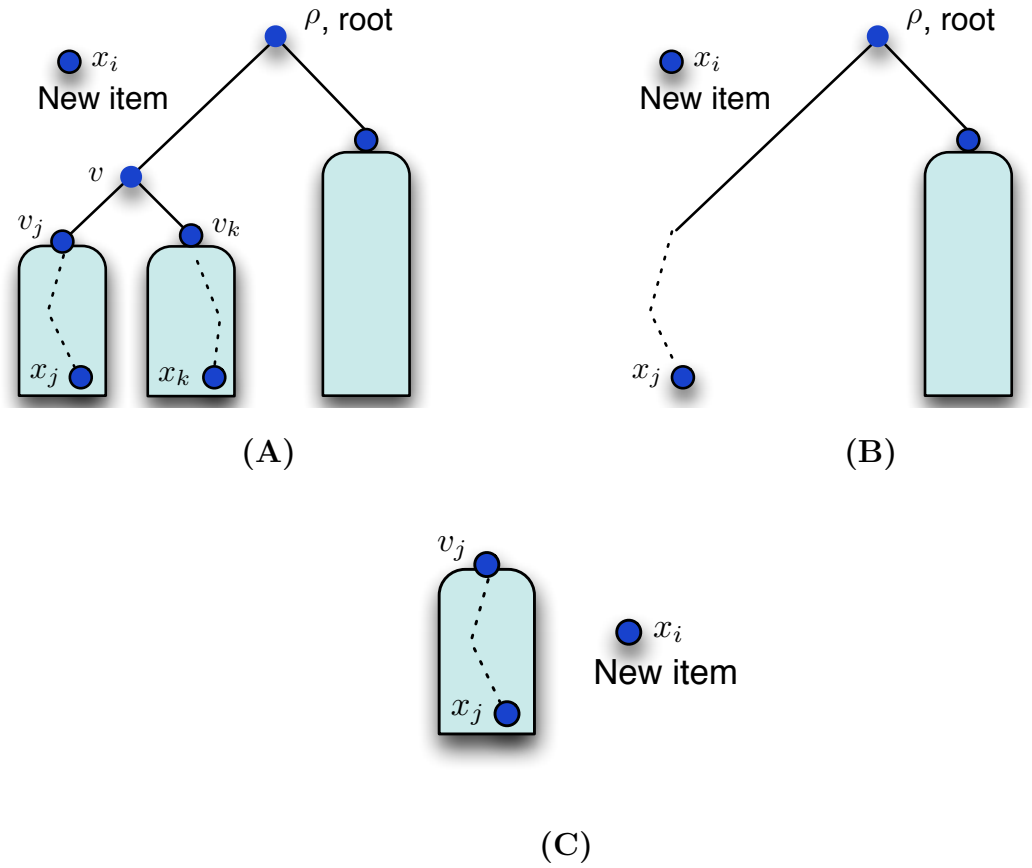


Figure 3.2: Outlier-based Clustering example. (A) T_{sub} at the start of the algorithm, (B) Resulting T_{sub} if x_i is the outlier with internal node v replaced with x_j, (C) Resulting T_{sub} if x_k is the outlier.

Lemma 1, we are guaranteed that the outlier test enables us to correctly identify the leader of a triple which is crucial to the correctness of the algorithm.

As observed in [107], each time we ask for the outlier of a triple, we are left with at most a fraction 2/3 of the leaves of the previous subtree. Therefore, the number of such tests we need to form T_{i+1} is at most $\log_{3/2} i$. Summing this quantity over i, we see that requires at most $n \log_{3/2} n$ outlier tests to reconstruct T. Given that each outlier test only requires 3 pairwise similarity values, we can reconstruct the hierarchical clustering T using at most $3n \log_{3/2} n$ adaptively selected pairwise similarity values.

□

3.4 Robust and Efficient Hierarchical Clustering with a Probably Correct Outlier Test

The TC condition guarantees that the outlier test in Equation 3.1 of any triple correctly coincides with the outlier of the same triple with respect to the underlying hierarchical clustering. However, this condition may be restrictive in real-world applications. For example, noisily or incorrectly measured similarities may cause the similarities to violate the TC condition leading to an incorrect clustering. This motivates developing techniques to discover the hierarchical clustering that are robust to cases where a fraction of the outlier tests may be wrong.

Definition 3.5. *Given* $(\mathbf{X}, T, \mathbf{S})$*, let* $\{x_i, x_j, x_k\}$ *be a random triple of objects drawn according to a uniform measure on the* $\binom{n}{3}$ *triples from* $\mathbf{X}$*. Without loss of generality suppose that* x_k *is the outlier of this triple with respect to* T*, then we define the* ***violation probability upper bound*** *as an upper bound on the probability that the* `outlier` *test fails for* (x_i, x_j, x_k)*, i.e.,*

$$P\left(\texttt{outlier}(x_i, x_j, x_k) \neq x_k\right) \leq q \tag{3.2}$$

We assume that $q < 1/2$*, and that the failures of the* `outlier()` *test occur independently.*

We present a clustering procedure that is robust to violations. To understand the effect of violations, we will compare the "correct" clusters for a set of items whose

similarities satisfy the TC condition with the clusters recovered from the same similarities corrupted by *arbitrary* violations with probability q; by arbitrary we mean that the affected similarities may take arbitrary values. In other words, the goal is to recover the correct clusters from similarity values with a certain fraction of errors/violations.

Now recall the Outlier-Based Clustering procedure from Algorithm 5. Since the procedure is "greedy" in nature, its ability to reconstruct the clustering hierarchy is predicated on the outlier determination technique of Equation 3.1 returning the correct item each time. The latter depends crucially on the TC condition, and the outlier-based clustering scheme can fail to properly cluster the items if violations occur. However, if the frequency of violations is not too large, then it should be possible to correctly cluster the items (at least for clusters in T that are not too small) using an appropriate averaging/voting procedure. The intuition is that if the majority of similarities satisfy the TC condition, then a voting scheme can overcome the effects of violations and indicate the proper clusterings. In this section, we develop a top-down recursive algorithm that is based on this rough idea.

The algorithm begins with the complete set of items $\mathbf{X}$ and then recursively splits this set into smaller clusters. At each step, the algorithm determines the "correct" split of the cluster in question into the two sub-clusters. Let C denote the cluster under consideration and suppose that C splits into two sub-clusters C_L and C_R in the true hierarchy T. That is, $C_L \cap C_R = \emptyset$ and $C_L \cup C_R = C$. The algorithm proceeds by picking a random item $x_j \in C_*$, where C_* can be either C_L or C_R. Now, the algorithm assigns the remaining items to either C_* or C_*^c. Robustness to errors in the outlier test is provided by a voting procedure at each step. See Algorithm 6 for details. In addition to the probability of violation, q, the degree to which T is "balanced" also affects the reliability of recovery.

Definition 3.6. *For each (non-leaf) cluster C, we define the* balance factor $\eta_C := \min\{|C_L|, |C_R|\} \backslash |C|$, *which quantifies how evenly C_L and C_R split C (where $C_L \bigcup C_R = C$ and $C_L \bigcap C_R = \emptyset$). The quantity $\eta \triangleq \min_{C \in T} \eta_C$ reflects the overall degree to which the tree T is balanced.*

Note that $0 < \eta \leq 1/2$. As we will show, η plays a crucial role in the number similarities required to determine T; the closer η is to $1/2$, the fewer similarities are required. This is because η controls the number of random items that need to be selected in each stage before a sufficient number of items are selected from each of the two clusters.

The *Robust Hierarchical Clustering* procedure is presented in Algorithm 6. Roughly speaking, the procedure quantifies how frequently two items tend to agree on outliers drawn from a small random subsample of other items. If they tend to agree, then they are clustered together; otherwise they are not.

Theorem 3.7. *Suppose the Robust Hierarchical Clustering Algorithm (Algorithm 6) is deployed with an agreement threshold $\gamma \in (0, 1/2)$. Then for all clustering problems with $n > 3$ and balance factor and violation probability (η, q) satisfying $(1 - (1 - q)^2) < \gamma < (1 - q)^2 \eta$, with probability at least $1 - \delta$, Algorithm 6 will correctly recover all clusters with size*

$$|C| > c\,(\gamma, \eta, q)\,[\log n + \log(2/\delta)]$$

where $c(\gamma, \eta, q)$ is a constant independent of n.

Remark. *Recall that the "correct" clusters are those corresponding to an ideal set of similarity values satisfying the TC condition. We suppose that the actual similarities provided to the algorithm are a corrupted version of the ideal set. The term "correctly recover" above means that with high probability the algorithm will recover the correct*

clusters (down to a certain minimal size) despite the violations (provided there are not too many). Comparing this result to that in Theorem 3.4, we note four distinctions: 1) we require $\mathcal{O}(n \log^2 n)$ rather than $O(n \log n)$ pairwise similarity values; 2) the degree to which the clusters are balanced now plays a role (in the constant); 3) we cannot guarantee the recovery of clusters smaller than $\log n$ in size, due to the voting requirement; and 4) we are robust for up to a fraction q of the similarities violating the TC condition.

Remark. *The threshold γ is used to judge how well pairs of item agree on randomly selected outliers. The theorem above shows that the algorithm is guaranteed (with high probability) to correctly recover most clusters for a certain range of q and η. Figure 3.3 depicts this range of q and η (the shaded region). Outside of this range the algorithm may perform quite well in practice, but our theorem provides no guarantee there. The lower-right corner of the plot corresponds to the most favorable conditions: $\eta = 1/2$ (well balanced clusters) and $q = 0$ (no violations). In light of Theorem 3.4, we expect to be able to correctly cluster items in this regime. The upper-left corresponds to less balanced clusters and higher levels of violation of the TC condition; both present greater difficulties for the voting procedure and our theory cannot guarantee results beyond a certain level of unbalancedness and/or violations.*

Proof: We start with a single cluster of n items, $C = \{x_1, x_2, ..., x_n\}$. Let us define the top-most split of the hierarchy as C_L, C_R. Our objective will be to use the potentially erroneous similarities values to assign the set of n items to one of the two top-most clusters. To begin, consider using a randomly selected subset of *voting items* $S_V \subset C$ to resolve which items are on either side of the top-most split. Instead of performing a single, possibly erroneous, outlier test on each item in the subset, we now introduce the

Algorithm 6 - Robust Hierarchical Clustering Algorithm

Given :

1. A set of items, $\mathbf{X} = \{x_1, x_2, ...x_n\}$
2. Pairwise similarity budget, $B = cN \log^2 N$ and threshold $\gamma \in (0, 1/2)$

Initialize : The initial cluster, $C = \mathbf{X}$.

Recurvsive Partitioning :

1. Randomly select two independent subsets $S_V, S_A \subset C$ containing $n_V, n_A = \frac{c}{2} \log N$ items each.
2. Randomly select an item $x_j \in C$.
3. For each $x_i \in C \setminus x_j$
   - Define $c_{i,k}$, for $i = 1, \ldots, N$ and $k \in S_A$, to be the number of outliers in S_V; i.e.,
   $$c_{i,k} := \sum_{m \in S_V} \mathbb{1}\left(\texttt{outlier}\left(x_i, x_k, x_m\right) = x_m\right)$$
   where the indicator $\mathbb{1}(\cdot)$ is 1 if its argument is true and 0 otherwise.
   - Compute the *outlier agreement* between x_i and x_j according to
   $$a_{i,j} := \sum_{k \in S_A} \left[\mathbb{1}\left(c_{i,k} > \gamma' \text{ and } c_{j,k} > \gamma'\right) + \mathbb{1}\left(c_{i,k} < \gamma' \text{ and } c_{j,k} < \gamma'\right)\right]$$
   where $\gamma' = \gamma n_V$.
   - Assign item x_i into the same cluster C_j as x_j or C_j^c,
   $$x_i \in \begin{cases} C_j & : \text{if} \quad a_{i,j} \geq \frac{1}{2} n_A \\ C_j^c & : \text{if} \quad a_{i,j} < \frac{1}{2} n_A \end{cases}$$
4. **If** $|C_j| > 2$, then set $C = C_j$ and go to step 1.

   If $\left|C_j^c\right| > 2$, then set $C = C_j^c$ and go to step 1.

   Else, stop.

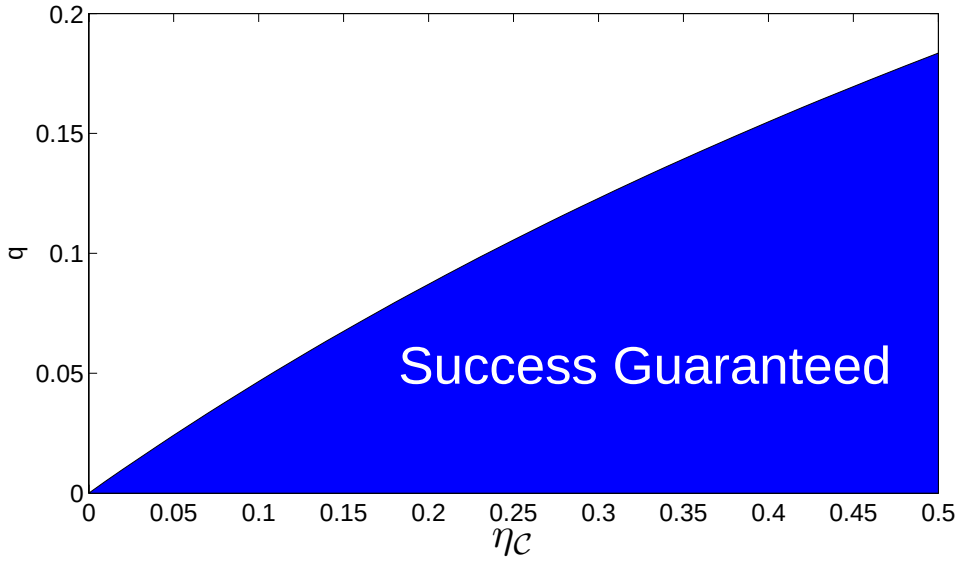


Figure 3.3: The shaded region depicts the range of violation probability q and balance factor η for which Theorem 3.7 can guarantee correct recovery of most clusters.

voting-based *outlier count* variable, $c_{i,j}$ where with respect to a pair of items, x_i, x_j, we count the number of times a voting item ($x_k \in S_V$) is the outlier,

$$c_{i,j} = \sum_{k \in S_V} \mathbb{1}\left(\texttt{outlier}\left(x_i, x_j, x_k\right) = x_k\right) \tag{3.3}$$

Where the indicator function, $\mathbb{1}(A) = 1$ if A is true, and $= 0$ if A is false.

If the size of the voting subset, $n_V \triangleq |S_V|$, is large enough, the following two properties hold with high probability: *(A)* - The set of items S_V contains at least one item in each of the top-most split clusters, C_L, C_R. And *(B)* - The outlier count values $c_{i,j}$ reveal whether two items x_i, x_j are in the same cluster (i.e., $i, j \in C_L$ or $i, j \in C_R$), or the items x_i, x_j are in differing clusters (i.e., $i \in C_L$ and $j \in C_R$, or the reverse), given that the similarity $s_{i,j}$ does not have a violation.

We now state a result that says that given $n_V = \frac{c}{2} \log n$, the above properties hold

with high probability. We prove this in Appendix B

Proposition 3.2. *If $n_V = \frac{c}{2}\log n$, then properties A and B stated above hold for the outlier count values in Equation 3.3 with probability $> 1 - \frac{2\delta_C}{3}$, if the violation probability q and balance factor η satisfy, $(1-(1-q)^2) < \gamma' < (1-q)^2\eta$ and $\frac{c}{2}\log n \geq f(\delta_C, \eta, q, \gamma')$ is large enough.*

Consider the item x_j identified in step 2 of the algorithm and let $x_j \in C_j$, where C_j is either C_L or C_R. For every other item $x_i \in C/x_j$, we want to cluster x_i either in C_j or C_j^c. However, note that we cannot use the cluster counts $c_{i,j}$ directly to decide the placement of x_i as it depends on similarity $s_{i,j}$ not having a violation. We need to account for the case where the value $c_{i,j}$ is erroneous due to a violation in the similarity value $s_{i,j}$. In order to be robust to such errors, we need to implement a second round of voting where all items in an additional independent set of items $S_A \subset C$ inform the placement of the item x_i.

We define a new *agreement count* variable, $a_{i,j}$ to equal the number of times the item x_j agrees with the clustering decision of x_i with respect to the set of agreement items, S_A.

$$a_{i,j} = \sum_{k \in S_A} \left(\mathbb{1}\left(c_{i,k} > \gamma \text{ and } c_{j,k} > \gamma\right) + \mathbb{1}\left(c_{i,k} < \gamma \text{ and } c_{j,k} < \gamma\right) \right) \tag{3.4}$$

where $\gamma = \gamma' n_V$.

The following proposition states that comparing the agreement count variable $a_{i,j}$ to a threshold, with high probability, reliably clusters the items $x_i \in C/x_j$ with $x_j \in C_j$ or places them in C_j^c. Thus, the top-most split of a set of n items can be resolved with high probability using only $\mathcal{O}(n \log n)$ total pairwise similarities. We prove this proposition

in Appendix B.

Proposition 3.3. *We can cluster an item $x_i \in C$ with respect to item $x_j \in C_j$ using the following procedure based on $|S_A| = n_A$ reinforcement items:*

$$x_i \in \begin{cases} C_j & : \text{if} \quad a_{i,j} \geq \frac{1}{2} n_A \\ C_j^c & : \text{if} \quad a_{i,j} < \frac{1}{2} n_A \end{cases} \tag{3.5}$$

If $n_R = \frac{c}{2} \log n$, this will resolve the correct clustering of the items, with probability $> 1 - \frac{\delta_C}{3}$ provided $\frac{c}{2} \log n \geq g(\delta_C, \eta, q, \gamma')$ is large enough.

To resolve the entire hierarchical clustering tree, we need to recursively perform this procedure at each split of the tree. Given a tree of n items with the balance factor η, the following proposition, proved in Appendix B, indicates that there are at most $\mathcal{O}(\log n)$ levels of the tree.

Proposition 3.4. *The tree structure for a set of n items has at most $L \leq \log n / \log(\frac{1}{1-\eta})$ levels, given the balance factor η.*

The entire hierarchical clustering can be resolved if all the clusters at every level are resolved correctly. With L levels, the total number of clusters M is bounded by $\sum_{\ell=1}^{L} 2^\ell \leq 2^{(L+1)} = \Theta(n)$, using the result of Proposition 3.4. Therefore, all clusters (which contain enough voting and reinforcement items) can be resolved with probability $> (1 - n\delta_C/3)^M \geq 1 - 1/n$, if we set $\delta_C = 3/n^3$. And the total number of pairwise similarities needed to reconstruct the clustering tree structure is bounded by $O(cn \log^2 n)$.

Finally, as alluded to earlier, this procedure is guaranteed to succeed only on clusters that have enough items to vote and reinforce with. Therefore, we have the following

bound the size of the smallest resolvable cluster using the Robust Hierarchical Clustering algorithm.

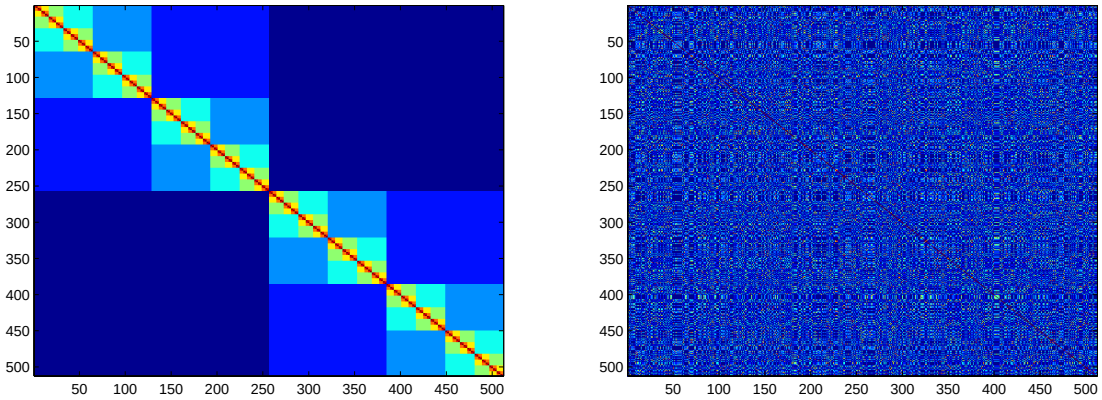
Proposition 3.5. *Using the Robust Hierarchical Clustering algorithm, we can resolve every cluster of size* $|C| > \max\{n_A, n_V\} = \frac{c}{2} \log N$.

From the above discussion and the proofs of Propositions 3.2, 3.3 and 3.4 in the appendix, it is clear that if we want the Robust Hierarchical Clustering algorithm to succeed with probability $> 1 - \delta$, then our theory guarantees the reliable recovery of only those clusters which have size $> c(\gamma, \eta, q) \left[\log n + \log (2/\delta)\right]$ where $c(\gamma, \eta, q)$ is a constant independent of n.

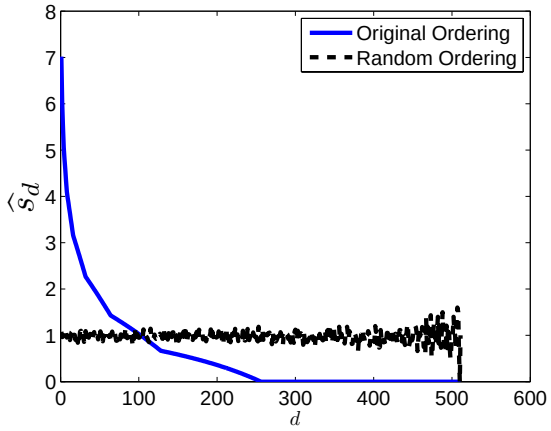
3.5 Experiments

To test our robust clustering methodology we focus on experimental results from a balanced binary tree using synthesized measurements and real-world datasets using genetic microarray data (provided by [37]) and a comparative housing dataset (provided by [72]). The synthetic binary tree experiments allows us to observe the characteristics of our algorithm while controlling the amount of violation of the Tight Clustering (TC) condition, while the real world data gives us perspective on problems where the tree structure and TC condition is assumed, but not known. To compare against our Robust Hierarchical Clustering methodology in Algorithm 6, we will use the exhaustive-based hierarchical clustering reconstruction of bottom-up agglomerative clustering.

In order to quantify the performance of the tree reconstruction algorithms, consider the object partital ordering, $\pi : \{1, 2, ..., n\} \rightarrow \{1, 2, ..., n\}$, resulting from ordering of the items in the reconstructed tree. We will consider the rate of decay of ordered similarity



(A) **(B)**



(C)

Figure 3.4: Synthetic balanced binary tree results for $n = 256$ (A) Original ordering of items from true tree structure, (B) Random ordering of items, (C) Similarity decay, $\widehat{s}_d$, for both orderings.

values off the diagonal of the reordered similarity values, $\widehat{s}_d = \frac{1}{n-d} \sum_{i=1}^{N-d} S_{\pi(i),\pi(i+d)}$. For a set of observed similarities, given the original ordering of the items from the true tree structure, we would expect to find the largest similarity values clustered around the diagonal of the similarity matrix (a balanced binary tree with $n = 256$ is seen in Figure 3.4-(A)). Meanwhile, a random ordering of the items would have the large

similarity values potentially scattered away from the diagonal (Figure 3.4-(B)). The resulting similarity decay for a realization of both orderings for a synthetic binary tree can be found in Figure 3.4-(C). The figure shows the uniform distribution of the randomly ordered similarity values and the steep decay of similarity values in the original ordering. Considering these orderings as a distribution of the pairwise similarities, we can measure the estimated entropy of the reordered similarity distribution,

$$\widehat{E}(\pi) = -\sum_{i=1}^{n-1} \widehat{p}_{\pi_i} \log \widehat{p}_{\pi_i} \tag{3.6}$$

Where $\widehat{p}_{\pi_i} = \left(\sum_{d=1}^{n-1} \widehat{s}_d\right)^{-1} \widehat{s}_i$

For the noise-free binary tree in Figure 3.4-(A), we find that for the original ordering, $\widehat{E}(\pi_{original}) = 5.02$, and for the random ordering, $\widehat{E}(\pi_{random}) = 6.04$. This motivates examining the estimated entropy of our ordering to evaluate the quality of our clustering methodologies, where the smaller the estimated entropy, the higher the quality of our estimated clustering.

For the synthetic binary tree experiments, we created a balanced binary tree with 512 items, where the similarity between each pair of items is the number of shared edges to the root of the tree. As this satisfies the TC condition, violations were synthesized by randomly introducing error into pairwise similarity comparisons with probability q. In Table 3.1, we see the performance of both standard bottom-up agglomerative clustering and our Robust Clustering algorithm for violation probability levels $q = 0.1, 0.2, 0.3$ with results averaged over 10 random realization of the synthetic data. For our Robust Clustering algorithm, we adjust the probing budget parameter between $c = 2$ (requiring

roughly 18% of agglomerative measurements) and $c = 20$ (requiring roughly 69% of agglomerative measurements). In terms of the estimated entropy of the ordering, we find that the implementations of our Robust procedure outperforms agglomerative clustering. In terms of the high noise experiment ($q = 0.3$), we find that with the probing budget parameter $c = 20$, we reconstruct the tree structure with considerably lower estimated entropy while using significantly fewer pairwise similarities when compared with the agglomerative clustering approach.

Table 3.1: Clustering results for synthetic binary tree with $n = 512$ for Agglomerative Clustering and Robust Clustering algorithms.

	Agglo.	Robust (c=2)		Robust (c=20)	
	Entropy	Entropy	% of Agg. Sim.	Entropy	% of Agg. Sim.
$q = 0.1$	5.238	5.028	17.13	5.024	68.48
$q = 0.2$	5.250	5.115	17.31	5.024	69.31
$q = 0.3$	5.238	5.440	18.47	5.024	69.35

In terms of real world datasets, we test our methodologies against both a dataset of genetic data and a collection of housing data. Our gene microarray dataset [37] consists of 1024 yeast genes with 7 expressions each, from which we exhaustively generate the standard Pearson correlation using the expression vectors for every pair of genes. Meanwhile, our housing dataset [72] is taken from a set of 14 features of 500 houses in suburban Boston, again with similarities from Pearson correlation. Our robust clustering methodology is performed on the dataset using two probing budgets (with $c = 2$ and $c = 20$) with results averaged over 10 random permutations of the datasets. The results in Table 3.2 show that again our robust clustering methodology outperforms agglomerative clustering in terms of estimated entropy of the reordered elements, while also not requiring

Table 3.2: Clustering results for real world dataset for Agglomerative Clustering and Robust Clustering algorithms.

	Agglo.	Robust (c=2)		Robust (c=20)	
Dataset	Entropy	Entropy	% of Agg. Sim.	Entropy	% of Agg. Sim.
Gene (n=512)	5.900	5.854	18.21	5.871	70.50
Gene (n=1024)	6.598	6.556	11.74	6.552	55.55
Housing (n=250)	5.177	5.191	28.57	5.169	88.29
Housing (n=500)	5.898	5.831	17.92	5.814	71.79

to observe every pairwise similarity value.

3.6 Discussion

Despite the wide ranging applications of hierarchical clustering, relatively little work has been done on examining the number of pairwise similarity values really needed to discover this pattern in data in the presence of noise. The goal of the work presented in this chapter was to show that under certain conditions, it is possible to use drastically fewer pairwise similarities to infer the hierarchical structure while remaining robust to potential outliers in the data. When the *Tight Clustering* condition holds, the first algorithm we saw, the *Active Clustering* method, requires no more than $3n \log_{\frac{3}{2}} n$ pairwise similarities to discover the underlying hierarchical clustering. We then saw how our robust algorithm can tolerate violations to this strong condition while needing only on the order of $\mathcal{O}\left(n \log^2 n\right)$ similarities to still recover the clustering with high probability.

Bibiliographic Note. The material in this chapter is based on joint work with Brian Eriksson, Aarti Singh, and Rob Nowak. A version of this appears as a journal-length

Computer Science paper in [53]. A slightly expanded version is available at [54].

Chapter 4

Sketching Sparse Graphs, Covariances, and Matrices

4.1 Introduction

Chapters 2 and 3 were concerned with the efficient data reconstruction (and robust) of trees. In this chapter, we will consider the efficient and robust reconstruction of more general graphs. This problem can be phrased abstractly as one of recovering a sparse matrix given access to a compressed linear transformation of the same and then connect it to concrete graph reconstruction problems in Section 4.1.3.

An important feature of many modern data analysis problems is the presence of a large number of variables relative to the amount of available resources. Such high dimensionality occurs in a range of applications in bioinformatics, climate studies, and economics. Accordingly, a fruitful and active research agenda over the last few years has been the development of methods for sampling, estimation, and learning that take into account *structure* in the underlying model and thereby making these problems tractable. A notion of structure that has seen many applications is that of *sparsity*, and methods for sampling and estimating sparse signals have been the subject of intense research in the past few years [16, 15, 42]

In this chapter we will study a more nuanced notion of structure which we call *distributed sparsity*. For what follows, it will be convenient to think of the unknown high-dimensional signal of interest as being represented as a matrix X. Roughly, the signal is said to be distributed sparse if every row and every column of X has only a few non-zeros. We will see that it is possible to design efficient and effective acquisition mechanisms for such signals. Let us begin by considering a few example scenarios where one might encounter distributed sparsity.

- **Covariance Matrices:** Covariance matrices associated to some natural phenomena have the property that each covariate is correlated with only a few other covariates. For instance, it is observed that protein signaling networks are such that there are only a few significant correlations [119] and hence the discovery of the such networks from experimental data naturally leads to the estimation of a covariance matrix (where the covariates are proteins) which is (approximately) distributed sparse. Similarly, the covariance structure corresponding to longitudinal data is distributed sparse [38]. See Section 4.1.3.

- **Multi-dimensional signals:** Multi-dimensional signals such as the natural images that arise in medical imaging [16] are known to be sparse in the gradient domain. When the features in the images are not axis-aligned, not only is the matrix representation of the image gradient sparse, it is also distributed sparse. For a little more on this, see Section 4.1.3

- **Random Sparse Signals and Random Graphs:** Signals where the sparsity pattern is *random* (i.e., each entry is nonzero independently and with a probability q) are also distributed sparse with high probability. The "distributedness" of the

sparsity pattern can be measured using the "degree of sparsity" d which is defined to be the maximum number of non-zeros in any row or column. For random sparsity patterns, we have the following:

Proposition 4.1. *Consider a random matrix $X \in \mathbb{R}^{p\times p}$ whose entries are independent copies of the Bernoulli(γ) [1] distribution where $p\gamma = \Delta = \Theta(1)$. Then for any $\epsilon > 0$, X has at most d 1's in each row/column with probability at least $1-\epsilon$, where*

$$d = \Delta\left(1 + \frac{2\log(2p/\epsilon)}{\Delta}\right).$$

(The proof is straightforward, and available in Appendix C.2.)

In a similar vein, combinatorial graphs have small *degree* in a variety of applications, and their corresponding matrix representation will then be distributed sparse. For instance, Erdos-Renyi random graphs $\mathcal{G}(p,q)$ with $pq = \mathcal{O}(\log p)$ have small degree [13].

4.1.1 Problem Setup and Main Results

Our goal is to invert an underdetermined linear system of the form

$$Y = AXB^T, \tag{4.1}$$

where $A = [a_{ij}] \in \mathbb{R}^{m\times p}, B = [b_{ij}] \in \mathbb{R}^{m\times p}$, with $m \ll p$ and $X \in \mathbb{R}^{p\times p}$. Since the matrix $X \in \mathbb{R}^{p\times p}$ is linearly transformed to obtain the smaller dimensional matrix $Y \in \mathbb{R}^{m\times m}$, we will refer to Y as the *sketch* (borrowing terminology from the computer science

[1] *Recall that if $\chi \sim$ Bernoulli(γ), then $P(\chi = 1) = \gamma$ and $P(\chi = 0) = 1-\gamma$.*

literature [102]) of X and we will refer to the quantity m as the *sketching dimension*. Since the value of m signifies the amount of compression achieved, it is desirable to have as small a value of m as possible.

Rewriting the above using tensor product notation, with $y = \text{vec}(Y)$ and $x = \text{vec}(X)$, we equivalently have

$$y = (B \otimes A)x, \tag{4.2}$$

where $\text{vec}(X)$ simply *vectorizes* the matrix X, i.e., produces a long column vector by stacking the columns of the matrix and $B \otimes A$ is the tensor (or Kronecker) product of B and A, given by

$$\begin{bmatrix} b_{11}A & b_{12}A & \cdots & b_{1p}A \\ b_{21}A & b_{22}A & \cdots & b_{2p}A \\ \vdots & \vdots & \ddots & \vdots \\ b_{m1}A & b_{m2}A & \cdots & b_{mp}A \end{bmatrix}. \tag{4.3}$$

While it is not possible to invert such underdetermined systems of equations in general, the rapidly growing literature on what has come to be known as *compressed sensing* [42] suggests that this can be done under certain assumptions. In particular, taking cues from this literature, one might think that this is possible if x (or equivalently X) has only a few non-zeros.

Let us first consider the case when there are only $k = \Theta(1)$ non-zeros in X, i.e., it is very sparse. Then, it is possible to prove that the optimization program (P_1) recovers X from AXB^T using standard "RIP-based" techniques [16]. We refer the interested reader to the papers by Jokar et al [85] and Duarte et al [44] for more details, but in essence the authors show that if $\delta_r(A)$ and $\delta_r(B)$ are the restricted isometry constants (of order

r) [16] for A and B respectively, then $\delta_r(A \otimes B) = \delta_r(B \otimes A)$ lies in the following interval:

$$\left[\max\left\{\delta_r(A), \delta_r(B)\right\}, \left(1 + \delta_r(A)\right)\left(1 + \delta_r(B)\right) - 1\right].$$

In many interesting problems that arise naturally, as we will see in subsequent sections, a more realistic assumption to make is that X has $\mathcal{O}(p)$ non-zeros and it is this setting we consider for this chapter. Unfortunately, the proof techniques outlined above cannot succeed in such a demanding scenario. As hinted earlier, it will turn out however that one cannot handle arbitrary sparsity patterns and that the non-zero pattern of X needs to be *distributed*, i.e., each row/column of X cannot have more than a few, say d, non-zeros. We will call such matrices $d-$distributed sparse (see Definition 4.5). We explore this notion of structure in more detail in Section 4.3.1.

An obvious, albeit highly impractical, approach to recover a (distributed) sparse X from measurements of the form $Y = AXB^T$ is the following: search over all matrices $\tilde{X} \in \mathbb{R}^{p \times p}$ such that $A\tilde{X}B^T$ agrees with $Y = AXB^T$ and find the sparsest one. One might hope that under reasonable assumptions, such a procedure would return X as the solution. However, there is no guarantee that this approach might work and worse still, such a search procedure is known to be computationally infeasible.

We instead consider solving the optimization program (P_1) which is a natural (convex) relaxation of the above approach.

$$\begin{aligned} \underset{\tilde{X}}{\text{minimize}} \quad & \left\|\tilde{X}\right\|_1 \\ \text{subject to} \quad & A\tilde{X}B^T = Y. \end{aligned} \tag{P_1}$$

Here, by $\|\tilde{X}\|_1$ we mean $\sum_{i,j}\left|\tilde{X}_{i,j}\right|$, i.e., the ℓ_1 norm of $\mathrm{vec}(\tilde{X})$.

The main part of this chapter is devoted to showing that with high probability (P_1) has a unique solution and that this solution equals X. In particular, we prove the following result.

Theorem 4.1. *Suppose that X is $d-$distributed sparse. Also, suppose that $A, B \in \{0,1\}^{m\times p}$ are drawn independently and uniformly from the $\delta-$random bipartite ensemble*[2]*. Then as long as*

$$m = \Omega\left(\sqrt{dp}\log p\right) \quad \text{and} \quad \delta = \mathcal{O}(\log p),$$

there exists a $c > 0$ such that the optimal solution X^* of (P_1) equals X with probability exceeding $1 - p^{-c}$. Furthermore, this holds even if B equals A.

We will prove Theorem 4.1 in Section 4.4. Let us pause here and consider some implications of this theorem.

1. (P_1) does not impose any structural restrictions on X^*. In other words, even though X is assumed to be distributed sparse, this (highly non-convex) constraint need not be factored in to the optimization problem. This ensures that (P_1) is a Linear Program (see e.g., [10]) and can thus be solved efficiently.

2. Recall that what we observe can be thought of as the $\mathbb{R}^{m^2}$ vector $(B \otimes A)x$. Since X is $d-$distributed sparse, x has $\mathcal{O}(dp)$ non-zeros. Now, even if an oracle were

[2] *Roughly speaking, the $\delta-$random bipartite ensemble consists of the set of all 0-1 matrices that have almost exactly δ ones per column. We refer the reader to Definition 4.6 for the precise definition and Section 4.3.2 for more details.*

to reveal the exact locations of these non-zeros, we would require at least $\mathcal{O}(dp)$ measurements to be able to perform the necessary inversion to recover x. In other words, it is absolutely necessary for m^2 to be at least $\mathcal{O}(dp)$. Comparing this to Theorem 4.1 shows that the simple algorithm we propose is *near optimal* in the sense that it is only a logarithm away from this trivial lower bound. This logarithmic factor also makes an appearance in the measurement bounds in the compressed sensing literature [15].

3. Finally, as mentioned earlier, inversion of under-determined linear systems where the linear operator assumes a tensor product structure has been studied earlier [84, 45]. However, these methods are relevant only in the regime where the sparsity of the signal to be recovered is much smaller than the dimension p. The proof techniques they employ will unfortunately not allow one to handle the more demanding situation the sparsity scales linearly in p and if one attempted an extension of their techniques naively to this situation, one would see that the sketch size m needs to scale like $\mathcal{O}(dp \log^2 p)$ in order to recover a $d-$ distributed sparse matrix X. This is of course uninteresting since it would imply that the size of the sketch is bigger than the size of X.

It is possible that Y was not exactly observed, but rather is only available to us as a corrupted version $\hat{Y}$. For instance, $\hat{Y}$ could be Y corrupted by independent zero mean, additive Gaussian noise or in case of the covariance sketching problem discussed in Section 4.1.3, $\hat{Y}$ could be an empirical estimate of the covariance matrix $Y = AXA^T$. In both these cases, a natural relaxation to (P_1) would be the following optimization

program (P_2) (with B set to A in the latter case).

$$\underset{\tilde{X}}{\text{minimize}} \qquad \|A\tilde{X}B^T - \hat{Y}\|_2^2 + \lambda \left\|\tilde{X}\right\|_1 \tag{P_2}$$

Notice that if X was a sparse covariance matrix and if $A = B = I_{p\times p}$, then (P_2) reduces to the soft thresholding estimator of sparse covariance matrices studied in [117].

While our experimental results show that this optimization program (P_2) performs well, we leave its exploration and analysis to future work. We will instead state the following "approximation" result that shows that the solution of (P_1) is close to the optimal $d-$ distributed sparse approximation for any matrix X. The proof is similar to the proof of Theorem 3 in [8] and is provided in Appendix C.3. Given $p \in \mathbb{N}$, let $[p]$ denote the set $\{1, 2, \ldots, p\}$ and let $\mathfrak{W}_{d,p}$ denote the set of all $\Omega \subset [p] \times [p]$ such that

$$\max_{i\in[p]} \{|\Omega \cap \{\{i\} \times [p]\}|, |\Omega \cap \{[p] \times \{i\}\}|\} \leq d.$$

Notice that if a matrix $X \in \mathbb{R}^{p\times p}$ is such that there exists an $\Omega \in \mathfrak{W}_{d,p}$ with the property that $X_{ij} \neq 0$ only if $(i, j) \in \Omega$, then the matrix is $d-$distributed sparse.

Given $\Omega \in \mathfrak{W}_{d,p}$ and a matrix $X \in \mathbb{R}^{p\times p}$, we write X_Ω to denote the projection of X onto the set of all matrices supported on Ω. That is,

$$[X_\Omega]_{i,j} = \begin{cases} X_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } (i,j) \in [p] \times [p].$$

Theorem 4.2. *Suppose that X is an arbitrary $p \times p$ matrix and that the hypotheses of Theorem 4.1 hold. Let X^* denote the solution to the optimization program (P_1). Then,*

there exist constants $c > 0$ *and* $\epsilon \in (0, 1/4)$ *such that the following holds with probability exceeding* $1 - p^{-c}$.

$$\|X^* - X\|_1 \leq \frac{2 - 4\epsilon}{1 - 4\epsilon} \left(\min_{\Omega \in \mathfrak{W}_{d,p}} \|X - X_\Omega\|_1 \right). \tag{4.4}$$

The above theorem tells us that even if X is not structured in any way, the solution of the optimization program (P_1) approximates X as well as the best possible $d-$distributed sparse approximation of X (up to a constant factor). This has interesting implications, for instance, to situations where a $d-$distributed sparse X is corrupted by a "noise" matrix N as shown in the following corollary.

Corollary. *Suppose* $X \in \mathbb{R}^{p \times p}$ *is* $d-$*distributed sparse and suppose that* $\hat{X} = X + N$. *Then, the solution* X^* *to the optimization program*

$$\min_{\tilde{X}} \ \left\| \tilde{X} \right\|_1 \ \textit{subject to } A\tilde{X}B^T = A\hat{X}B^T$$

satisfies

$$\|X^* - X\|_1 \leq \frac{5 - 12\epsilon}{1 - 4\epsilon} \|N\|_1 \tag{4.5}$$

Proof. Let Ω be the support of X. To prove the result, we will consider the following

chain of inequalities.

$$\begin{aligned}
\|X^* - X\|_1 &\leq \left\|X^* - \hat{X}\right\|_1 + \left\|\hat{X} - X\right\|_1 \\
&\overset{(a)}{\leq} \frac{2-4\epsilon}{1-4\epsilon}\left\|\hat{X} - \hat{X}_\Omega\right\|_1 + \left\|\hat{X} - X\right\|_1 \\
&\overset{(b)}{\leq} \frac{3-8\epsilon}{1-4\epsilon}\left\|\hat{X} - X\right\|_1 + \frac{2-4\epsilon}{1-4\epsilon}\left\|\hat{X}_\Omega - X\right\|_1 \\
&\overset{(c)}{\leq} \frac{5-12\epsilon}{1-4\epsilon}\left\|\hat{X} - X\right\|_1 \\
&= \frac{5-12\epsilon}{1-4\epsilon}\|N\|_1 .
\end{aligned}$$

Here (a) follows from Theorem 4.2 since $\Omega \in \mathfrak{W}_{d,p}$ and (b) follows by bounding $\left\|\hat{X} - \hat{X}_\Omega\right\|$ by $\left\|\hat{X} - X\right\|_1 + \left\|\hat{X}_\Omega - X\right\|_1$ using the triangle inequality. (c) follows from the fact that $\left\|\hat{X}_\Omega - X\right\| \leq \left\|\hat{X} - X\right\|$ since X_{Ω^c} is $\mathbf{0}_{p\times p}$. □

4.1.2 The Rectangular Case and Higher Dimensional Signals

While Theorem 4.1, as stated, applies only to the case of square matrices X, we can extend our result in a straightforward manner to the rectangular case. Consider a matrix $X \in \mathbb{R}^{p_1 \times p_2}$ where (without loss of generality) $p_1 < p_2$. We assume that the row degree is d_r (i.e. no row of X has more than d_r non-zeros) and that the column degree is d_c. Consider sketching matrices $A \in \mathbb{R}^{m\times p_1}$ and $B \in \mathbb{R}^{m\times p_2}$ and the sketching operation:

$$Y = AXB^T.$$

Then we have the following corollary:

Corollary. *Suppose that X is distributed sparse with row degree d_r and colum degree*

d_c. Also, suppose that $A \in \{0,1\}^{m\times p_1}$, $B \in \{0,1\}^{m\times p_2}$ are drawn independently and uniformly from the $\delta-$random bipartite ensemble. Let us define $p = \max(p_1, p_2)$ and $d = \max(d_r, d_c)$.

Then if

$$m = \Omega\left(\sqrt{dp}\log p\right) \quad \text{and} \quad \delta = \mathcal{O}(\log p),$$

there exists a $c > 0$ such that the optimal solution X^* of (P_1) equals X with probability exceeding $1 - p^{-c}$.

Proof. Let us define the matrix $\tilde{X} \in \mathbb{R}^{p\times p}$ as

$$\tilde{X} = \begin{bmatrix} X \\ \\ 0 \end{bmatrix},$$

i.e. it is made square by padding additional zero rows. Note that $\tilde{X}$ has degree $d = \max(d_r, d_c)$. Moreover note that the matrix $A \in \mathbb{R}^{m\times p_1}$ can be augmented to $\tilde{A} \in \mathbb{R}^{m\times p}$ via:

$$\tilde{A} = \begin{bmatrix} A & \bar{A} \end{bmatrix}$$

where $\bar{A} \in \mathbb{R}^{m\times(p-p_1)}$ is also drawn from the δ-random bipartite ensemble. Then one has the relation:

$$Y = \tilde{A}\tilde{X}B^T.$$

Thus, the rectangular problem can be reduced to the standard square case considered in Theorem 4.1, and the result follows. □

The above result shows that a finer analysis is required for the rectangular case. For instance, if one were to consider a scenario where $p_1 = 1$, then from the compressed

sensing literature, we know that the result of Corollary 4.1.2 is weak. We believe that determining the right scaling of the sketch dimension(s) in the case when X is rectangular is an interesting avenue for future work.

Finally, we must also state that while the results in this chapter only deal with two-dimensional signals, similar techniques can be used to deal with higher dimensional tensors that are distributed sparse. We leave a detailed exploration of this question to future work.

4.1.3 Applications

It is instructive at this stage to consider a few examples of the framework we set up in this chapter. These applications demonstrate that the modeling assumptions we make viz., tensor product sensing and distributed sparsity are important and arise naturally in a wide variety of contexts.

Covariance Sketching: Covariance Estimation from Compressed Realizations

One particular application that will be of interest to us is the estimation of covariance matrices from sketches of the sample vectors. We call this *covariance sketching*.

Consider a scenario in which the covariance matrix $\Sigma \in \mathbb{R}^{p\times p}$ of a high-dimensional zero-mean random vector $\xi = (\xi_1, \ldots, \xi_p)^T$ is to be estimated. In many applications of interest, one determines Σ by conducting correlation tests for each pair of covariates ξ_i, ξ_j and computing an estimate of $\mathbf{E}[\xi_i \xi_j]$ for $i, j = 1, \ldots, p$. This requires one to perform correlation tests for $\mathcal{O}(p^2)$ pairs of covariates, a daunting task in the high-dimensional setting. Perhaps most importantly, in many cases of interest, the underlying covariance matrix may have structure, which such an approach may fail to exploit. For instance if Σ

is very sparse, it would be vastly more efficient to perform correlation tests corresponding to only the non-zero entries. The chief difficulty of course is that the sparsity pattern is rarely known in advance, and finding this is often the objective of the experiment.

In other settings of interest, one may obtain statistical samples by observing n independent *sample paths* of the statistical process. When ξ is high-dimensional, it may be infeasible or undesirable to sample and store the entire sample paths $\xi^{(1)}, \ldots, \xi^{(n)} \in \mathbb{R}^p$, and it may be desirable to reduce the dimensionality of the acquired samples.

Thus in the high-dimensional setting we propose an alternative acquisition mechanism: pool covariates together to form a collection of new variables $Z_1, \ldots, Z_m$, where $m < p$. For example one may construct:

$$Z_1 = \xi_1 + \xi_2 + \xi_6, \quad Z_2 = \xi_1 + \xi_4 + \xi_8 + \xi_{12}, \quad \ldots$$

and so on; more generally we have measurements of the form $Z = A\xi$ where $A \in \mathbb{R}^{m \times p}$ and typically $m \ll p$. We call the thus newly constructed covariates $Z = (Z_1, \ldots, Z_m)$ a sketch of the random vector ξ.

More formally, the covariance sketching problem can be stated as follows. Let $\xi^{(1)}, \xi^{(2)}, \ldots, \xi^{(n)} \in \mathbb{R}^p$ be n independent and identically distributed $p-$variate random vectors and let $\Sigma \in \mathbb{R}^{p \times p}$ be their unknown covariance matrix. Now, suppose that one has access to the $m-$dimensional *sketch vectors* $Z^{(i)}$ such that

$$Z^{(i)} = A\xi^{(i)}, \quad i = 1, 2, \ldots, n,$$

where $A \in \mathbb{R}^{m \times p}, m < p$ is what we call a *sketching matrix*. The goal then is to recover

Σ using only $\{Z^{(i)}\}_{i=1}^n$. The sketching matrices we will focus on later will have randomly-generated binary values, so each element of $Z^{(i)}$ will turn out to be a sum (or "pool") of a random subset of the covariates.

Notice that the sample covariance matrix computed using the vectors $\left\{Z^{(i)}\right\}_{i=1}^n$ satisfies the following.

$$\begin{aligned}\hat{\Sigma}_Z^{(n)} &:= \frac{1}{n}\sum_{i=1}^n Z^{(i)}(Z^{(i)})^T \\ &= A\left(\frac{1}{n}\sum_{i=1}^n \xi^{(i)}(\xi^{(i)})^T\right)A^T \\ &= A\hat{\Sigma}^{(n)}A^T,\end{aligned}$$

where $\hat{\Sigma}^{(n)} := \frac{1}{n}\sum_{i=1}^n \xi^{(i)}(\xi^{(i)})^T$ is the (maximum likelihood) estimate of Σ from the samples $\xi^{(1)}\ldots,\xi^{(n)}$.

To gain a better understanding of the covariance sketching problem, it is natural to first consider the stylized version of the problem suggested by the above calculation. That is, whether it is possible to efficiently recover a matrix $\Sigma \in \mathbb{R}^{p\times p}$ given the ideal covariance matrix of the sketches $\Sigma_Z = A\Sigma A^T \in \mathbb{R}^{m\times m}$. The analysis in the current chapter focuses on exactly this problem and thus helps in exposing the most unique and challenging aspects of covariance sketching.

The theory developed in this chapter tells us that at the very least, one needs to restrict the underlying random vector ξ to have the property that each ξ_i depends on only a few (say, d) of the other ξ_j's. Notice that this would of course imply that the true covariance matrix Σ will be $d-$distributed sparse. Applying Theorem 4.1, especially the version in which the matrices A and B are identical, to this stylized situation reveals the

following result. If A is chosen from a particular random ensemble and if one gets to observe the covariance matrix $A\Sigma A^T$ of the sketch random vector $Z = A\xi$, then using a very efficient convex optimization program, one can recover Σ exactly.

Now, suppose that ξ and A are as above and that we get to observe n samples $Z^{(i)} = A\xi^{(i)}, i = 1, 2, \ldots, n$ of the sketch $Z = A\xi$. Notice that we can can consider $\hat{\Sigma}^{(n)} := \frac{1}{n}\sum_{i=1}^{n} \xi^{(i)}(\xi^{(i)})^T$ to be a "noise corrupted version" of Σ since we can write

$$\hat{\Sigma}^{(n)} = \Sigma + (\hat{\Sigma}^{(n)} - \Sigma),$$

where, under reasonable assumptions on the underlying distribution, $\left\|\hat{\Sigma}^{(n)} - \Sigma\right\|_1 \to 0$ almost surely as $n \to \infty$ by the strong law of large of numbers. Therefore, an application of Theorem 4.2 tells us that solving (P_1) with the observation matrix $\hat{\Sigma}_Z^{(n)}$ gives us an asymptotically consistent procedure to estimate the covariance matrix Σ from sketched realizations.

We anticipate that our results will be interesting in many areas such as quantitative biology where it may be possible to naturally pool together covariates and measure interactions at this pool level. Our work shows that covariance structures that occur naturally are amenable to covariance sketching, so that drastic savings are possible when correlation tests are performed at the pool level, rather than using individual covariates.

The framework we develop in this chapter can also be used to accomplish *cross covariance sketching*. That is, suppose that ξ and ζ are two zero mean $p-$variate random vectors and suppose that $\Sigma_{\xi\zeta} \in \mathbb{R}^{p\times p}$ is an unknown matrix such that $[\Sigma_{\xi\zeta}]_{ij} = \mathbb{E}[\xi_i\zeta_j]$. Let $\{\xi^{(i)}\}_{i=1}^n$ and $\{\zeta^{(i)}\}_{i=1}^n$ be $2n$ independent and identically distributed random realizations of ξ and ζ respectively. The goal then, is to estimate $\Sigma_{\xi\zeta}$ from the m dimensional *sketch*

vectors $Z^{(i)}$ and $W^{(i)}$ such that

$$Z^{(i)} = A\xi^{(i)}, W^{(i)} = B\zeta^{(i)} \quad i = 1, 2, \ldots, n,$$

where $A, B \in \mathbb{R}^{m \times p}, m < p$.

As above, in the idealized case, Theorem 4.1 shows that the cross-covariance matrix $\Sigma_{\xi\zeta}$ of ξ and ζ can be exactly recovered from the cross-covariance matrix $\Sigma_{ZW} = A\Sigma_{\xi\zeta}B^T$ of the sketched random vectors W and Z as long as $\Sigma_{\xi\zeta}$ is distributed sparse. In the case we have n samples each of the sketched random vectors, an application of Theorem 4.2 to this problem tells us that (P_1) is an efficient and asymptotically consistent procedure to estimate a distributed sparse $\Sigma_{\xi\zeta}$ from compressed realization of ξ and ζ.

We note that the idea of pooling information in statistics, especially in the context of meta analysis is a classical one [76]. For instance the classical Cohen's d estimate uses the idea of pooling samples obtained from different distributions to obtain accurate estimates of a common variance. While at a high level the idea of pooling is related, we note that our notion is qualitatively different in that we propose pooling covariates themselves into sketches and obtain samples in this reduced dimensional space.

Graph Sketching

Large graphs play an important role in many prominent problems of current interest; two such examples are graphs associated to communication networks (such as the internet) and social networks. Due to their large sizes it is difficult to store, communicate, and analyze these graphs, and it is desirable to compress these graphs so that these tasks are easier. The problem of compressing or sketching graphs has recently gained attention in

the literature [1, 63].

In this section we propose a new and natural notion of compression of a given graph $G = (V, E)$. The resulting "compressed" graph is a weighted graph $\hat{G} = (\hat{V}, \hat{E})$, where $\hat{V}$ has a much smaller cardinality than V. Typically, $\hat{G}$ will be a complete graph, but the edge weights will encode interesting and valuable information about the original graph.

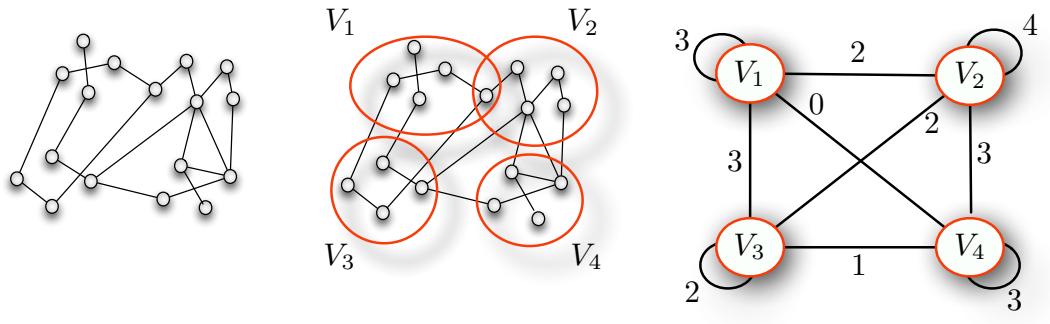
Partition the vertex set $V = V_1 \cup V_2 \cup \ldots \cup V_m$; in the compressed graph $\hat{G}$, each partition V_i is represented by a node. (We note that this need not necessarily be a disjoint partition, and we allow for the possibility for $V_i \cap V_j \neq \emptyset$.) For each pair $V_i, V_j \in \hat{V}$, the associated edge weight is the total number of edges crossing from nodes in V_i to the nodes in V_j in the original graph G. Note that if an index $k \in V_i \cap V_j$, the self edge (k, k) must be included when counting the total number of edges between V_i and V_j. (We point out that the edge $(V_k, V_k) \in \hat{E}$ also carries a non-zero weight; and is precisely equal to the number of edges in G that have both endpoints in V_k. See Fig. 1 for an illustrative example.)

Define A_i to be the (row) indicator vector for the set V_i, i.e.

$$A_{ij} = \begin{cases} 1 & \text{if } j \in V_i \\ 0 & \text{otherwise} \end{cases}$$

If X denotes the adjacency matrix of G, then $Y := AXA^T$ denotes the matrix representation of $\hat{G}$. The sketch Y has two interesting properties:

- The encoding faithfully preserves high-level "cut" information about the original graph. For instance information such as the weight of edges crossing between the partitions V_i and V_j is faithfully encoded. This could be useful for networks where



(a) (b) (c)

Figure 4.1: An example illustrating graph sketching. (a) A graph G with 17 nodes (b) Partitioning the nodes into four partitions V_1, V_2, V_3, V_4 (c) The sketch of the graph G. The nodes represent the partitions and the edges in the sketch represent the total number of edges of G that cross partitions.

the vertex partitions have a natural interpretation such as geographical regions; questions about the total network capacity between two regions is directly available via this method of encoding a graph. Approximate solutions to related questions such as maximum flow between two regions (partitions) can also be provided by solving the problem on the compressed graph.

- When the graph is bounded degree, the results in this chapter show that there exists a suitable random partitioning scheme such that the proposed method of encoding the graph is lossless. Moreover, the original graph G can be unravelled from the smaller sketched graph $\hat{G}$ efficiently using the convex program (P_1).

Multidimensional Signal Processing

Multi-dimensional signals arise in a variety of applications, for instance images are naturally represented as two-dimensional signals $f(\cdot, \cdot)$ over some given domain.

Often it is more convenient to view the signal not in the original domain, but rather in a transformed domain. Given some one dimensional family of "mother" functions $\psi_u(t)$ (usually an orthonormal family of functions indexed with respect to the transform variable u), such a family induces the transform for a one dimensional signal $f(t)$ (with domain $\mathcal{T}$) via

$$\hat{f}(u) = \int_{t \in \mathcal{T}} f(t)\psi_u(t).$$

For instance if $\psi_u(t) := \exp(-i2\pi ut)$, this is the Fourier transform, and if $\psi_u(t)$ is chosen to be a wavelet function (where $u = (a, b)$, the translation and scale parameters respectively) this generates the well-known wavelet transform that is now ubiquitous in signal processing.

Using $\psi_u(t)$ to form an orthonormal basis for one-dimensional signals, it is straightforward to extend to a basis for two-dimensional signal by using the functions $\psi_u(t)\psi_v(r)$. Indeed, this defines a two-dimensional transform via

$$\hat{f}(u, v) = \int_{(t,r) \in \mathcal{X} \times \mathcal{X}} f(t, r)\psi_u(t)\psi_v(r).$$

Similar to the one-dimensional case, appropriate choices of ψ yeild standard transforms such as the two-dimensional Fourier transform and the two-dimensional wavelet transform. The advantage of working with an alternate basis as described above is that signals often have particularly simple representations when the basis is appropriately chosen. It is well-known, for instance, that natural images have a sparse representation in the wavelet basis (see Fig. 4.2). Indeed, many natural images are not only sparse, but they are also distributed sparse, when represented in the wavelet basis. This enables compression by performing "pooling" of wavelet coefficients, as described below.

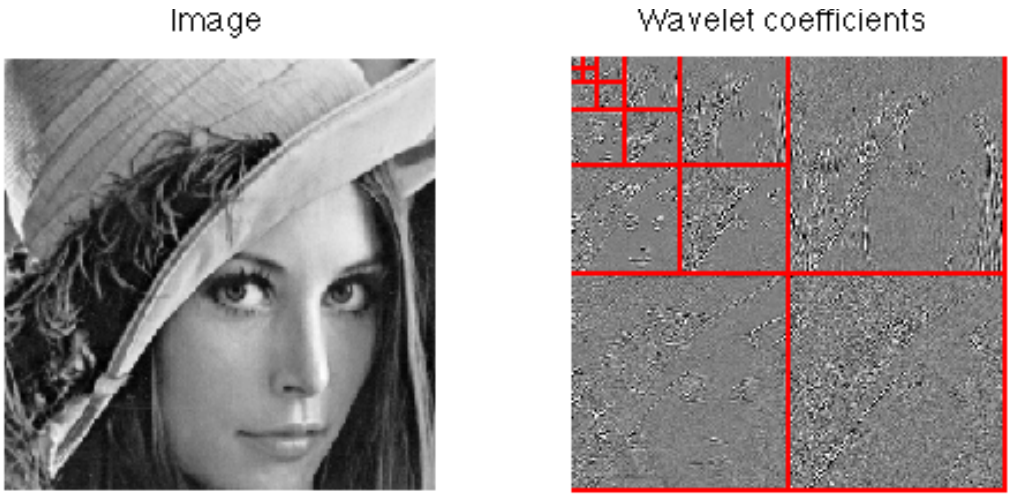


Figure 4.2: The wavelet representation of the image is distributed sparse.

In many applications, it is more convenient to work with discrete signals and their transforms (by discretizing the variables (t, r) and the transform domain variables (u, v)). It is natural to represent the discretization of the two-dimensional signal $f(t, r)$ by a matrix $F \in \mathbb{R}^{p \times p}$. The corresponding discretization of $\psi_u(t)$ can be represented as a matrix $\Psi = [\Psi]_{ut}$, and the discretized version of the $\hat{f}(u, v)$, denoted by $\hat{F}$ is given by:

$$\hat{F} = \Psi F \Psi^T.$$

As noted above, in several applications of interest, when the basis Ψ is chosen appropriately, the signal has a succinct representation and the corresponding matrix $\hat{F}$ is sparse. This is true, for instance, when F represents a natural image and $\hat{F}$ is the wavelet transform of F. Due to the sparse representability of the signal in this basis, it is possible to acquire and store the signal in a *compressive* manner. For instance, instead of sensing the signal F using Ψ (which correpsonds to sensing the signal at every value of the transform variable u), one could instead form "pools" of transform variables

$S_i = \{u_{i1}, u_{i2} \ldots, u_{ik}\}$ and sense the signal via

$$A\Psi = \begin{bmatrix} \sum_{u \in S_1} \Psi_u \\ \vdots \\ \sum_{u \in S_m} \Psi_u \end{bmatrix},$$

where the matrix A corresponds to the pooling operation. This means of compression corresponds to "mixing" measurements at randomly chosen transform domain values u. (When Ψ is the Fourier transform, this corresponds to randomly chosen frequencies, and when Ψ is the wavelet, this corresponds to mixing randomly chosen translation and scale parameters) . When the signal F is acquired in this manner, we obtain measurements of the form:

$$Y = A\hat{F}A^T,$$

where $\hat{F}$ is suitably sparse. Note that one may choose different random mixtures of measurements for the t and r "spatial" variables, in which case one would obtain measurements of the form:

$$Y = A\hat{F}B^T.$$

The theory developed in this chapter shows how one can recover the multi-dimensional signal F from such an undersampled acquisition mechanism. In particular, our results will show that if the pooling of the transform variable is done suitably randomly, then there is an efficient method based on linear programming that can be used to recover the original multi-dimensional signal.

4.1.4 Related Work and the Contributions of this chapter

The problem of recovering sparse signals via ℓ_1 regularization and convex optimization has been studied extensively in the past decade; our work fits broadly into this context. In the signal processing community, the literature on compressed sensing [15, 42] focuses on recovering sparse signals from data. In the statistics community, the LASSO formulation as proposed by Tibshirani, and subsequently analyzed (for variable selection) by Meinshausen and Bühlmann [99], and Wainwright [136] are also closely related. Other examples of structured model selection include estimation of models with a few latent factors (leading to low-rank covariance matrices) [56], models specified by banded or sparse covariance matrices [11, 12], and Markov or graphical models [91, 99, 113]. These ideas have been studied in depth and extended to analyze numerous other model selection problems in statistics and signal processing [19, 17].

Our work is also motivated by the work on sketching in the computer science community; this literature deals with the idea of compressing high-dimensional data vectors via projection to low-dimensions while preserving pertinent geometric properties. The celebrated Johnson-Lindenstrauss Lemma [81] is one such result, and the idea of sketching has been explored in various contexts [3, 87]. The idea of using random bipartite graphs and their related expansion properties, which motivated our approach to the problem, have also been studied in past work [8, 89].

While most of the work on sparse recovery focuses on sensing matrices where each entry is an i.i.d. random variable, there are a few lines of work that explore structured sensing matrices. For instance, there have been studies of matrices with Toeplitz structure [75], or those with random entries with independent rows but with possibly

dependent columns [134, 118]. Also related is the work on deterministic dictionaries for compressed sensing [24], although those approaches yield results that are too weak for our setup.

One interesting aspect of our work is that we show that it is possible to use highly constrained sensing matrices (i.e. those with tensor product structure) to recover the signal of interest. Many standard techniques fail in this setting. Restricted isometry based approaches [16] and coherence based approaches [40, 66, 132] fail due to a lack of independence structure in the sensing matrix. Indeed, the restricted isometry constants as well as the coherence constants are known to be weak for tensor product sensing operators [45, 84]. Gaussian width based analysis approaches [19] fail because the kernel of the sensing matrix is not a uniformly random subspace and hence not amenable to a similar application of Gordon's ("escape through the mesh") theorem. We overcome these technical difficulties by working directly with combinatorial properties of the tensor product of a random bipartite graph, and exploiting those to prove the so-called nullspace property [41, 23].

4.2 Experiments

We demonstrate the validity of our theory with some preliminary experiments in this section.

Figure 4.4 shows a 40×40 distributed sparse matrix on the left side. This matrix was generated by picking a 40×40 tri-diagonal matrix and populating populating the remaining upper triangular part with 38 non-zero entries at randomly chosen coordinates. The non-zero values off the diagonal were chosen to be uniformly random in $[-2, 2]$ and

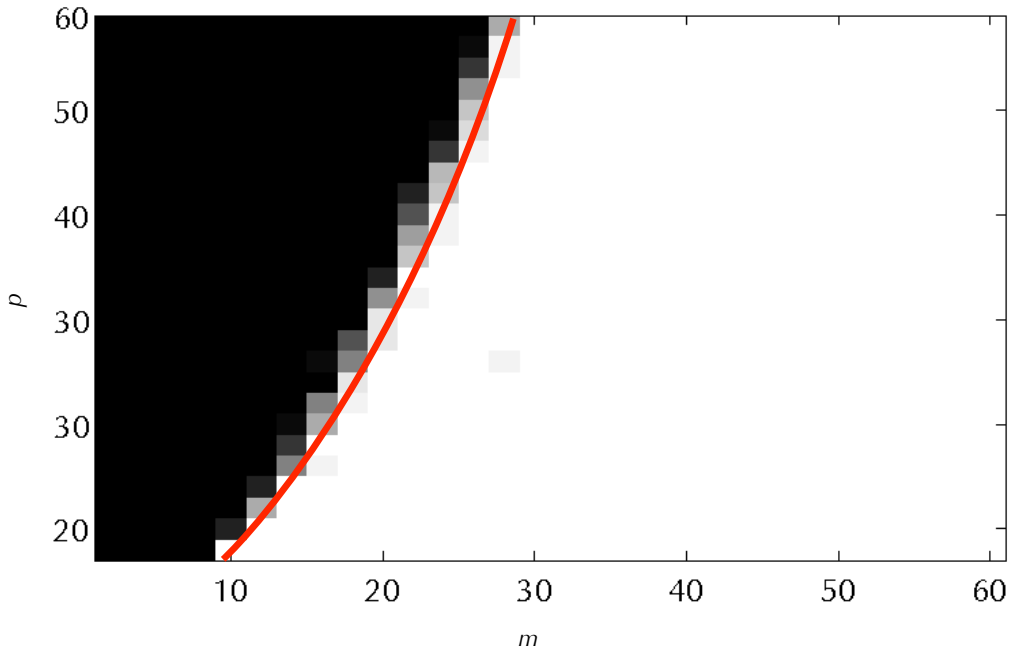


Figure 4.3: Phase transition plot. The (i, j)-th pixel shows (an approximation) to the probability of success of the optimization problem (P_1) in recovering a distributed sparse $X \in \mathbb{R}^{i \times i}$ with sketch-size j. The (red) solid line shows the boundary of the phase transition regime and is approximately the curve $p = \frac{1}{14}m^2$

the diagonal entries were all chosen to be 8 (these numbers were chosen to make X diagonally dominant, but neither the theory nor the experiments hinge on this property). We then generate A and B to be 21×40 binary random matrices such that each column contains exactly four 1's at coordinates chosen uniformly at random without replacement and the remaining entries are set to 0 (note that $\lceil \log(40) \rceil = 2$). The matrix on the right is a perfect reconstruction of X from the measurement AXB^T. We used the CVX toolbox [79, 65] to solve the optimization problem (P_1).

Figure 4.3 is what is known now as the "phase transition diagram". Each coordinate $(i, j) \in \{10, 12, \ldots, 60\} \times \{2, 4, \ldots, 60\}$ in the figure corresponds to an experiment with

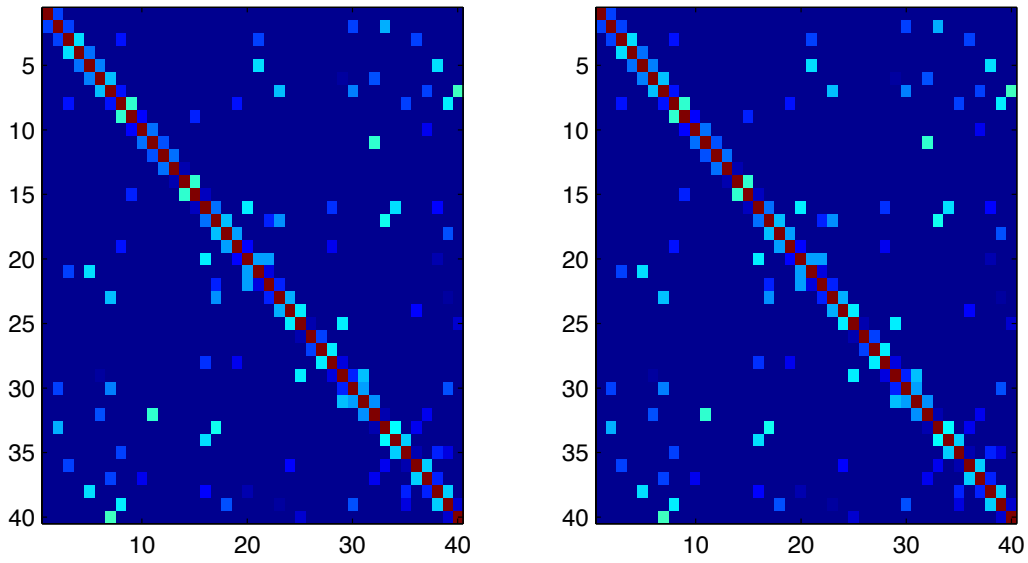


Figure 4.4: The matrix on the left is a 40×40 sparse matrix and the matrix on the right is a perfect reconstruction with $m = 21$.

$p = i$ and $m = j$. The value at the coordinate (i, j) was generated as follows. We first generate a random 4-distributed sparse $X \in \mathbb{R}^{i \times i}$ by taking a diagonal matrix and picking 3 other coordinates uniformly at random in each row. We then symmetrize the matrix and discard it (and re-sample) if the matrix has more than 4 non-zeros per row/column after symmetrization. The diagonal and off-diagonal values are picked at random as above. We then pick a random $A \in \mathbb{R}^{j \times i}$ as described above with $2 \times \lceil \log(i) \rceil$ 1's per column in randomly chosen locations. The optimization problem (P_1) was solved using the CVX toolbox [79, 65]. The solution X^* was compared to X in the $\|\cdot\|_\infty$ norm (upto numerical precision errors). This was repeated 40 times and the average number of successes (error of zero) was calculated and is reported in the (i, j)-th pixel. In the figure, the white region denotes success during each trial and the black region denotes failure in every single trial and there is a sharp *phase transition* between successes and failures. In fact, the curve that borders this phase transition region roughly looks like the curve

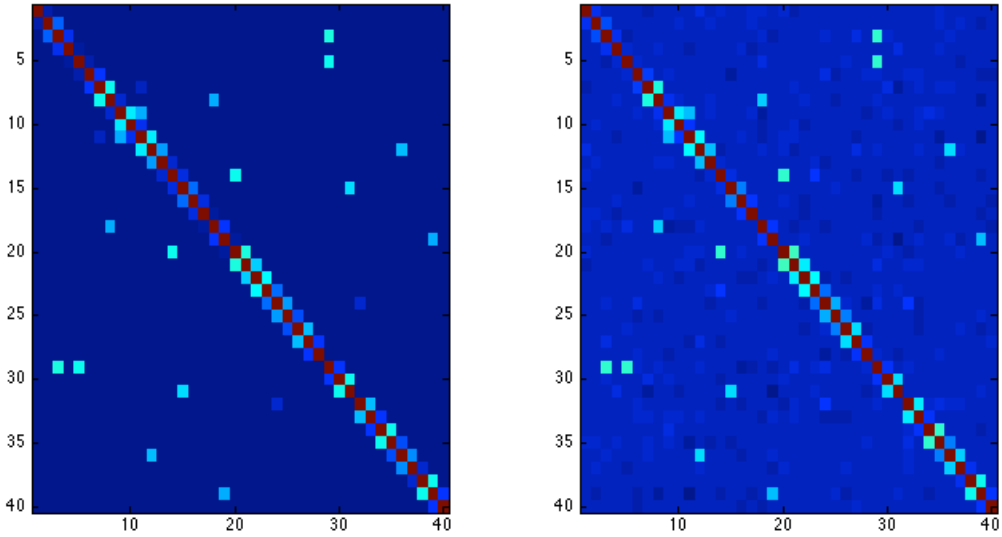


Figure 4.5: The matrix on the left is a 40×40 distributed sparse matrix. The matrix on the right was reconstructed using $n = 2100$ samples and with sketches of size $m = 21$.

$p = \frac{1}{14}m^2$ which is what our theory predicts (upto constants and log factors).

We also ran some preliminary tests on reconstructing a covariance matrix from sketches of samples drawn from the original distribution as in Section 4.1.3. We generated a 2−distributed sparse covariance matrix $\Sigma_\xi \in \mathbb{R}^{40\times 40}$ as above (picking the entries of the matrix so that it is diagonally dominant to ensure positive semidefiniteness) and generated n samples $\{\xi_i\}_{i=1}^n$ from $\mathcal{N}(\mathbf{0}, \Sigma_\xi)$, where we chose $n = 2100$. We set $m = 21$ (note that $\left\lceil\sqrt{2 \times 40 \times \log(40)}\right\rceil = 21$) and generated $A \in \{0,1\}^{21\times 40}$ as before. This gave us the observations $Z_i = A\xi_i, i = 1, \ldots, n$. To factor in the "noise", instead of (P_1), we solved the following optimization problem: *minimize* $||\Sigma||_1$ *subject to* $||A\Sigma A^T - \widehat{\Sigma}_Z^{(n)}||_F^2 \leq \kappa^*$, where κ^* was picked by cross-validation as in [12]. That is, we divided the observations into two sets of sizes n_1 and n_2 as suggested by [12]. We let $\widehat{\Sigma}_{Z,n_2}$ be the sample covariance matrix of the observations (Z_i's) in the latter set and we let $\Sigma_{n_1}^{*,\kappa}$ be the solution of the above optimization problem (with parameter κ) using only the observations of the first

set. This was repeated N times and the "risk" for κ, $R(\kappa)$, was calculated by averaging $\left\|A\Sigma_{n_1}^{*,\kappa}A^T - \widehat{\Sigma}_{Z,n_2}\right\|_F^2$ over the N trials. Then κ^* (whose numerical value turned out to be 2.635) was chosen to be the κ that minimizes $R(\kappa)$. Figure 4.5 shows a representative result which is encouraging. The matrix on the left is a 40×40 distributed sparse covariance matrix as described above and the matrix on the right is a reconstruction using $n = 2100$ sketches of size $m = 21$ each.

4.3 Preliminaries and Notation

Before we prove our theoretical results, we will establish some notation and some preliminary concepts that will be used through the rest of the chapter. We will conclude this section with the statement of Lemma 4.7 which is a novel result concerning tensor products of bipartite graphs, which may be of independent interest.

For any $p \in \mathbb{N}$, we define $[p] := \{1, 2, \ldots, p\}$. Henceforth, unless otherwise stated, we will deal with the graph $G = ([p], E)$. We also assume that all the graphs that we consider here include all the self loops, i.e., $\{i, i\} \in E$ for all $i \in [p]$. For any $S \subset [p]$, the set of its neighbors is denoted as $N(S)$, i.e.,

$$N(S) = \{j \in [p] : i \in S, \{i, j\} \in E\}.$$

For any vertex $i \in [p]$, as before, the degree $\deg(i)$ is defined as $\deg(i) := |N(i)|$.

Definition 4.3 (Bounded degree graphs and regular graphs)**.** *A graph $G = ([p], E)$ is*

said to be a **bounded degree graph** *with (maximum) degree d if for all $i \in [p]$,*

$$deg(i) \leq d$$

*The graph is said to be $d-$***regular** *if $deg(i) = d$ for all $i \in [p]$.*

We will be interested in another closely related combinatorial object. Given $p, m \in \mathbb{N}$, a **bipartite graph** $G = ([p], [m], E)$ is a graph with the *left set* $[p]$ and *right set* $[m]$ such that the edge set E only has pairs $\{i, j\}$ where i is the left set and j is in the right set. A bipartite graph $G = ([p], [m], E)$ is said to be $\delta-$**left regular** if for all i in the left set $[p]$, $\deg(i) = \delta$. Given two sets $A \subset [p], B \subset [m]$, we define the set

$$E(A : B) := \{(i, j) \in E : i \in A, j \in B\},$$

which we will find use for in our analysis. This set is sometimes known as the *cut set*. Finally for a set $A \subset [p]$ (resp. $B \subset [m]$), we define $N(A) := \{j \in [m] : i \in A, \{i, j\} \in E\}$ (resp. $N_R(B) := \{i \in [p] : j \in B, \{i, j\} \in E\}$). This distinction between N and N_R is made only to reinforce the meaning of the quantities which is otherwise clear in context.

Definition 4.4. *(Tensor graphs) Given two bipartite graphs $G_1 = ([p], [m], E_1)$ and $G_2 = ([p], [m], E_2)$, we define their* **tensor graph** *$G_1 \otimes G_2$ to be the bipartite graph $([p] \times [p], [m] \times [m], E_1 \otimes E_2)$ where $E_1 \otimes E_2$ is such that $\{(i, i'), (j, j')\} \in E_1 \otimes E_2$ if and only if $\{i, j\} \in E_1$ and $\{i', j'\} \in E_2$.*

Notice that if the adjacency matrices of G_1, G_2 are given respectively by $A^T, B^T \in \mathbb{R}^{p \times m}$, then the adjacency matrix of $G_1 \otimes G_2$ is $(A \otimes B)^T \in \mathbb{R}^{p^2 \times m^2}$.

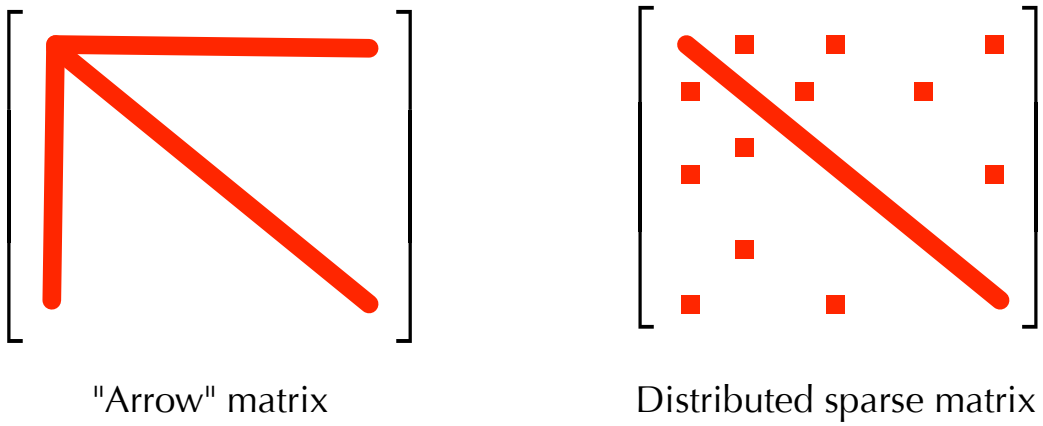


Figure 4.6: Two matrices with $\mathcal{O}(p)$ non-zeros. The "arrow" matrix is impossible to recover by covariance sketching while the distributed sparse matrix is.

As mentioned earlier, we will be particularly interested in the situation where $B = A$. In this case, write the tensor product of a graph $G = ([p], [m], E)$ with itself as $G^{\otimes} = ([p] \times [p], [m] \times [m], E^{\otimes})$. Here $E^{\otimes}$ is such that $\{(i, i'), (j, j')\} \in E^{\otimes}$ if and only if $\{i, j\}$ and $\{i', j'\}$ are in E.

Throughout this chapter, we write $\|\cdot\|$ to denote norms of vectors. For instance, $\|x\|_1$ and $\|x\|_2$ respectively stand for the ℓ_1 and ℓ_2 norm of x. Furthermore, for a matrix X, we will often write $\|X\|$ to denote $\|\mathrm{vec}(X)\|$ to avoid clutter. Therefore, the Frobenius norm of a matrix X will appear in this chapter as $\|X\|_2$.

4.3.1 Distributed Sparsity

As promised, we will now argue that distributed sparsity is important. Towards this end, let us turn our attention to Figure 4.6 which shows two matrices with $\mathcal{O}(p)$ non-zeros. Suppose that the non-zero pattern in X looks like that of the matrix on the left (which we dub as the "arrow" matrix). It is clear that it is impossible to recover this X from

AXB^T *even if* we know the non-zero pattern in advance. For instance, if $v \in \ker(A)$, then the matrix $\tilde{X}$, with v added to the first column of X is such that $AXB^T = A\tilde{X}B^T$ and $\tilde{X}$ is also an arrow matrix and hence indistinguishable from X. Similarly, one can "hide" a kernel vector of B in the first row of the arrow matrix. In other words, it is impossible to uniquely recover X from AXB^T.

In what follows, we will show that if the sparsity pattern of X is more *distributed*, as in the right side of Figure 4.6, then one can recover X and do so efficiently. In fact, our analysis will also reveal what that the size of the sketch Y needs to be able to perform this task and we will see that this is very close to being optimal.
In order to make things concrete, we will now define these notions formally.

Definition 4.5 ($d-$distributed sets and $d-$distributed sparse matrices)**.** *We say that a subset* $\Omega \subset [p] \times [p]$ *is* $d-$**distributed** *if the following hold.*

1. *For* $i = 1, 2, \ldots, p$, $(i, i) \in \Omega$.

2. *For all* $k \in [p]$, *the cardinality of the sets* $\Omega_k := \{(i, j) \in \Omega : i = k\}$ *and* $\Omega^k := \{(i, j) \in \Omega : j = k\}$ *is no more than* d.

The set of all $d-$*distributed subsets of* $[p] \times [p]$ *will be denoted as* $\mathfrak{W}_{p,d}$. *We say that a matrix* $X \in \mathbb{R}^{p \times p}$ *is* $d-$**distributed sparse** *if there exists an* $\Omega \in \mathfrak{W}_{d,p}$ *such that* $supp(X) := \{(i, j) \in [p] \times [p] : X_{ij} \neq 0\} \subset \Omega$.

While the theory we develop here is more generally applicable, the first point in the above definition makes the presentation easier. Notice that this forces the number of off-diagonal non-zeros in each row or column of a $d-$distributed sparse matrix X to be

at most $d-1$. This is not a serious limitation since a more careful analysis along these lines can only improve the bounds we obtain by at most a constant factor.

Examples:

- Any diagonal matrix is $d-$distributed sparse for $d = 1$. Similarly a tridiagonal matrix is $d-$distributed sparse with $d = 3$.
- The adjacency matrix of a bounded degree graph with maximum degree $d-1$ is $d-$distributed sparse
- As shown in Proposition 4.1, *random sparse matrices* are $d-$distributed sparse with $d = \mathcal{O}(\log p)$. While we implicitly assume that d is constant with respect to p in what follows, all arguments work even if d grows logarithmically in p as is the case here.

Given a matrix $X \in \mathbb{R}^{p\times p}$, as mentioned earlier, we write $\text{vec}(X)$ to denote the $\mathbb{R}^{p^2}$ vector obtained by stacking the columns of X. Suppose $x = \text{vec}(X)$. It will be useful in what follows to remember that x was actually derived from the matrix X and hence we will employ slight abuses of notation as follows: We will say x is $d-$distributed sparse when we actually mean that the matrix X is. Also, we will write (i, j) to denote the index of x corresponding to X_{ij}, i.e.,

$$x_{ij} := x_{(i-1)p+j} = X_{ij}.$$

We finally make two remarks:

- Even if X were distributed sparse, the observed vector $Y = AXB^T$ is usually unstructured (and dense).

- While the results in this chapter focus on the regime where the maximum number of non-zeros d per row/column of X is a constant (with respect to p), they can be readily extended to the case when d grows poly-logarithmically in p. Extensions to more general scalings of d is an interesting avenue for future work.

4.3.2 Random Bipartite Graphs, Weak Distributed Expansion and the Choice of the Sketching Matrices

As alluded to earlier, we will choose the sensing matrices A, B to be the adjacency matrices of certain random bipartite graphs. The precise definition of this notion follows.

Definition 4.6 (Uniformly Random $\delta-$left regular bipartite graph)**.** *We say that $G = ([p], [m], E)$ is a uniformly random $\delta-$left regular bipartite graph if the edge set E is a random variable with the following property: for each $i \in [p]$ one chooses δ vertices $j_1, j_2, \ldots, j_\delta$ chosen uniformly and independently at random (with replacement) from $[m]$ such that $\{\{i, j_k\}\}_{k=1}^{\delta} \subset E$.*

Remarks:

- Note that since we are sampling with replacement, it follows that the bipartite graph thus constructed may not be simple. If for instance there are two edges from the left node i to the right node i, the corresponding entry $A_{ij} = 2$.

- It is in fact possible to work with a sampling without replacement model in Definition 4.6 (where the resulting graph is indeed simple) and obtain qualititatively the same results. We work with a "sampling with replacement" model for the ease of exposition.

The probabilistic claims in this chapter are made with respect to this probability distribution on the space of all bipartite graphs.

In past work [8, 89] , the authors show that a random graph generated as above is, for suitable values of ϵ, δ, a $(k, \delta, \epsilon)-$expander. That is, for all sets $S \subset [p]$ such that $|S| \leq k$, the size of the neighborhood $|N(S)|$ is no less than $(1-\epsilon)\delta\,|S|$. If A is the adjacency matrix of such a graph, then it can then be shown that this implies that ℓ_1 minimization would recover a $k-$sparse vector x if one observes the sketch Ax (actually, [8] shows that these two properties are equivalent). Notice that in our context, the vector that we need to recover is $\mathcal{O}(p)$ sparse and therefore, our random graph needs to be a $(\mathcal{O}(p), \delta, \epsilon)-$expander. Unfortunately, this turns out to not be true of $G_1 \otimes G_2$ when G_1 and G_2 are randomly chosen as directed above.

However, we prove that if G_1 and G_2 are picked as in Definition 4.6, then their tensor graph $G_1 \otimes G_2$, satisfies what can be considered a *weak distributed expansion* property. This roughly says that the neighborhood of a $d-$distributed $\Omega \subset [p] \times [p]$ is large enough. Moreover, we show that this is in fact sufficient to prove that with high probability X can be recovered from AXB^T efficiently. The precise statement of these combinatorial claims follows.

Lemma 4.7. *Suppose that $G_1 = ([p], [m], E_1)$ and $G_2 = ([p], [m], E_2)$ are two independent uniformly random $\delta-$left regular bipartite graphs with $\delta = \mathcal{O}(\log p)$ and $m = \Omega(\sqrt{dp}\log p)$. Let $\Omega \in \mathfrak{W}_{d,p}$ be fixed. Then there exists an $\epsilon \in \left(0, \frac{1}{4}\right)$ such that $G_1 \otimes G_2$ has the following properties with probability exceeding $1 - p^{-c}$, for some $c > 0$.*

1. $|N(\Omega)| \geq p\delta^2(1-\epsilon)$.

2. *For any* $(i, i') \in ([p] \times [p]) \setminus \Omega$ *we have* $|N(i, i') \cap N(\Omega)| \leq \epsilon\delta^2$.

3. *For any* $(i, i') \in \Omega$, $|N(i, i') \cap N(\Omega \setminus (i, i'))| \leq \epsilon\delta^2$.

Moreover, all these claims continue to hold when G_2 *is the same as* G_1.

Remarks:

- Part 1 of Lemma 4.7 says that if Ω is a $d-$distributed set, then the size of the neghborhood of Ω is large. This can be considered a *weak distributed expansion property*. Notice that while it is reminiscent of the *vertex expansion* property of expander graphs, it is easy to see that it does not hold if Ω is not distributed. Furthermore, we call it "weak" because the lower bound on the size of the neighborhood is $\delta^2 p(1-\epsilon)$ as opposed to $\delta^2 |\Omega| (1-\epsilon) = \delta^2 dp(1-\epsilon)$ as one typically gets for standard expander graphs. It will become clear that this is one of the key combinatorial facts that ensures that the necessary theoretical guarantees hold for covariance sketching.

- Parts 2 and 3 say that the number of collisions between the edges emanating out of a single vertex with the edges emanating out of a distributed set is small. Again, this combinatorial property is crucial for the proof of Theorem 4.1 to work.

As stated earlier, we are particularly interested in the challenging case when $G_1 = G_2$ (or equivalently, their adjacency matrices A, B are the same). The difficultly, loosely speaking, stems from the fact that since we are not allowed to pick G_1 and G_2 separately, we have much less independence. In Appendix C.1, we will first prove Lemma 4.7 in the case when $G_1 = G_2$. We will then follow this up with the simple modifications to handle the case when G_1 and G_2 are drawn independently.

4.4 Proof of Theorem 4.1

In this section, we will prove the main theorem. To reduce clutter in our presentation, we will employ the following notational shortcuts. When the context is clear, the ordered pair (i, i') will simply be written as ii' and the set $[p] \times [p]$ will be written as $[p]^2$. Sometimes, we will also write $\mathcal{A}$ to mean $B \otimes A$ and if $S \subset [p] \times [p]$, we write $(B \otimes A)_S$ or $\mathcal{A}_S$ to mean the submatrix of $B \otimes A$ obtained by appending the columns $\{B_i \otimes A_j \mid (i, j) \in S\}$.

Proof of Theorem 4.1. We will consider an arbitrary ordering of the set $[p] \times [p]$ and we will order the edges in $E_1 \otimes E_2$ lexicographically based on this ordering, i.e., the first δ^2 edges $e_1, \ldots, e_{\delta^2}$ in $E_1 \otimes E_2$ are those that correspond to the first element as per the ordering on $[p] \times [p]$ and so on. Now, one can imagine that the graph $G_1 \otimes G_2$ is formed by including these edges sequentially as per the ordering on the edges. This allows us to partition the edge set into the set E_M of edges that do not collide with any of the previous edges as per the ordering and the set $E_M^c := (E_1 \otimes E_2) \setminus E_M$. (We note that a similar proof technique was adopted in Berinde et al. [8]).

As a first step towards proving the main theorem, we will show that the operator $B \otimes A$ preserves the ℓ_1 norm of a matrix X as long as X is distributed sparse. Berinde et al., [8] call a similar property RIP-1, taking cues from the restricted isometry property that has become popular in literature [16]. The proposition below can also be considered to be a restricted isometry property but the operator in our case is only constrained to behave like an isometry for distributed sparse vectors.

Proposition 4.2 (ℓ_1-RIP)**.** *Suppose that $X \in \mathbb{R}^{p \times p}$ is a $d-$distributed sparse matrix. Also suppose that A and B are the adjacency matrices of two independent uniformly*

random $\delta-$left regular bipartite graphs. Then there exists an $\epsilon > 0$ such that

$$(1-2\epsilon)\delta^2 \|X\|_1 \leq \left\|AXB^T\right\|_1 \leq \delta^2 \|X\|_1, \tag{4.6}$$

with probability exceeding $1-p^{-c}$ for some $c > 0$. Furthermore, this result continues to hold when $A = B$.

Proof. The upper bound follows (deterministically) from the fact that the induced (matrix) ℓ_1-norm of $B \otimes A$, i.e., the maximum column sum of $B \otimes A$, is precisely δ^2. To prove the lower bound, we need the following lemma.

Lemma 4.8. *For any $X \in \mathbb{R}^{p\times p}$,*

$$\left\|AXB^T\right\|_1 \geq \delta^2 \|X\|_1 - 2 \sum_{jj' \in [m]^2} \sum_{ii' \in [p]^2} \mathbb{1}_{\{ii',jj'\} \in E_M^c} |X_{ii'}| \tag{4.7}$$

Proof. In what follows, we will denote the indicator function $\mathbb{1}_{\{ii',jj'\}\in S}$ by $\mathbb{1}_S^{\{ii'jj'\}}$. We

begin by observing that

$$\begin{aligned}
\left\|AXB^T\right\|_1 &= \|\mathcal{A}\,\mathrm{vec}(X)\|_1 \\
&= \sum_{jj'\in[m]^2}\left|\sum_{ii'\in[p]^2}\mathcal{A}_{\{ii'jj'\}}X_{ii'}\right| \\
&= \sum_{jj'\in[m]^2}\left|\sum_{ii'\in[p]^2}\mathbb{1}_{E_1\otimes E_2}^{\{ii'jj'\}}\,X_{ii'}\right| \\
&= \sum_{jj'\in[m]^2}\left|\sum_{ii'\in[p]^2}\mathbb{1}_{E_M}^{\{ii'jj'\}}X_{ii'}+\sum_{ii'\in[p]^2}\mathbb{1}_{E_M^c}^{\{ii'jj'\}}X_{ii'}\right| \\
&\geq \sum_{jj'\in[m]^2}\left|\sum_{ii'\in[p]^2}\mathbb{1}_{E_M}^{\{ii'jj'\}}X_{ii'}\right|-\left|\sum_{ii'\in[p]^2}\mathbb{1}_{E_M^c}^{\{ii'jj'\}}X_{ii'}\right| \\
&\overset{(a)}{\geq} \sum_{jj'\in[m]^2}\left(\sum_{ii'\in[p]^2}\mathbb{1}_{E_M}^{\{ii'jj'\}}\left|X_{ii'}\right|-\sum_{ii'\in[p]^2}\mathbb{1}_{E_M^c}^{\{ii'jj'\}}\left|X_{ii'}\right|\right) \\
&= \sum_{ii'\in[p]^2,jj'\in[m]^2}\mathbb{1}_{E_1\otimes E_2}^{\{ii'jj'\}}\left|X_{ii'}\right|-2\sum_{ii'\in[p]^2jj'\in[m]^2}\mathbb{1}_{E_M^c}^{\{ii'jj'\}}\left|X_{ii'}\right|,
\end{aligned}$$

where (a) follows after observing that the first (double) sum has only one term and applying triangle inequality to the second sum. Since $\sum_{ii'\in[p]^2,jj'\in[m]^2}\mathbb{1}_{E_1\otimes E_2}^{\{ii'jj'\}}\left|X_{ii'}\right| = \delta^2\left\|X\right\|_1$, this concludes the proof of the lemma. □

Now, to complete the proof of Proposition 4.2, we need to bound the sum in the LHS of (4.7). Notice that

$$\sum_{ii'jj':(ii'jj')\in E_2^{\otimes}}\left|X_{ii'}\right| = \sum_{ii'}\left|X_{ii'}\right|r_{ii'} = \sum_{ii'\in\Omega}\left|X_{ii'}\right|r_{ii'}.$$

where $r_{ii'}$ is the number of collisions of edges emanating from ii' with all the previous edges as per the ordering we defined earlier. Since Ω is $d-$distributed, from the third part of Lemma 4.7, we have that for all $ii'\in\Omega$, $r_{ii'}\leq\epsilon\delta^2$ with probability exceeding

$1 - p^{-c}$ and therefore,

$$\sum_{ii' \in \Omega} |X_{ii'}| \, r_{ii'} \leq \epsilon \delta^2 \, \|X\|_1 \, .$$

This concludes the proof. Observe that since both Lemma 4.8 and Lemma 4.7 continue to hold even when $A = B$, Proposition 4.2 also holds in this case. □

Next, we will use the fact that $B \otimes A$ behaves as an approximate isometry (in the ℓ_1 norm) to prove what can be considered a nullspace property [41, 23]. This will tell us that the nullspace of $B \otimes A$ is "smooth" with respect to distributed support sets and hence ℓ_1 minimization as proposed in (P_1) will find the right solution.

Proposition 4.3 (Nullspace Property)**.** *Suppose that $A, B \in \{0,1\}^{m \times p}$ are adjacency matrices of (independent) random bipartite $\delta-$left regular graph with $\delta = \mathcal{O}(\log p)$ and $m = \mathcal{O}(\sqrt{dp} \log p)$ and that $\Omega \in \mathfrak{W}_{d,p}$ is fixed. Then, with probability exceeding $1 - p^{-c}$, for any $V \in \mathbb{R}^{p \times p}$ such that $AVB^T = 0$, we have*

$$\|V_\Omega\|_1 \leq \frac{\epsilon}{1 - 3\epsilon} \|V_{\Omega^c}\|_1 \, . \tag{4.8}$$

for some $\epsilon \in (0, \frac{1}{4})$ and for some $c > 0$. Furthermore, this result holds even when $A = B$.

Proof. Let V be any symmetric matrix such that $AVB^T = 0$. Let $v = \mathrm{vec(V)}$ and note that $\mathrm{vec}\left(AVB^T\right) = (B \otimes A)\, v = 0$. Let Ω be a d-distributed set. As indicated in Section 4.3, we define $N(\Omega) \subseteq [m]^2$ to be the set of neighbors of Ω with respect to the graph $G_1 \otimes G_2$. Let $(B \otimes A)^{N(\Omega)}$ denote the submatrix of $B \otimes A$ that contains only those rows corresponding to $N(\Omega)$ (and all columns). We will slightly abuse notation and use v_Ω to denote the vectorization of the projection of V onto the set Ω, i.e., $v_\Omega = \mathrm{vec}(V_\Omega)$,

where

$$[V_\Omega]_{i,j} = \begin{cases} V_{i,j} & (i,j) \in \Omega \\ \\ 0 & \text{otherwise} \end{cases}.$$

Now, observe that the following chain of inequalities are true.

$$\begin{aligned} 0 &= \left\|(B \otimes A)^{N(\Omega)} v\right\|_1 \\ &= \left\|(B \otimes A)^{N(\Omega)} (v_\Omega + v_{\Omega^c})\right\|_1 \\ &\geq \left\|(B \otimes A)^{N(\Omega)} v_\Omega\right\|_1 - \left\|(B \otimes A)^{N(\Omega)} v_{\Omega^c}\right\|_1 \\ &= \|(B \otimes A) v_\Omega\|_1 - \left\|(B \otimes A)^{N(\Omega)} v_{\Omega^c}\right\|_1 \\ &\geq (1-2\epsilon)\delta^2 \|V_\Omega\|_1 - \left\|(B \otimes A)^{N(\Omega)} v_{\Omega^c}\right\|_1, \end{aligned}$$

where the last inequality follows from Proposition 4.2. Resuming the chain of inequalities, we have:

$$\begin{aligned} 0 &\geq (1-2\epsilon)\delta^2 \|V_\Omega\|_1 - \sum_{ii' \in \Omega^c} \left\|(B \otimes A)^{N(\Omega)} v_{\{ii'\}}\right\|_1 \\ &\geq (1-2\epsilon)\delta^2 \|V_\Omega\|_1 - \sum_{\substack{ii', jj' : (ii', jj') \in E_1 \otimes E_2, \\ jj' \in N(\Omega), ii' \in \Omega^c}} |V_{ii'}| \\ &= (1-2\epsilon)\delta^2 \|V_\Omega\|_1 - \sum_{ii' \in \Omega^c} |(E_1 \otimes E_2)(ii' : N(\Omega))| \, |V_{ii'}| \\ &\overset{(a)}{\geq} (1-2\epsilon)\delta^2 \|V_\Omega\|_1 - \sum_{ii' \in \Omega^c} \epsilon\delta^2 \ |V_{ii'}| \\ &\geq (1-2\epsilon)\delta^2 \|V_\Omega\|_1 - \epsilon\delta^2 \|V\|_1, \end{aligned}$$

where (a) follows from the second part of Lemma 4.7. Writing $\|V\|_1 = \|V_\Omega\|_1 + \|V_{\Omega^c}\|_1$

and rearranging, we get the required result. Notice that since Lemma 4.7 holds even when $A = B$, this result continues to hold in this case $\square$

Now, we can use this to conclude the proof of Theorem 4.1. Let Ω be the support of X and notice that Ω is $d-$distributed. Now, suppose that there exists an $\tilde{X} \neq X$ such that $A\tilde{X}B^T = Y$. Observe that $A\left(\tilde{X} - X\right)B^T = 0$. Now, consider

$$\begin{aligned}
\|X\|_1 &\leq \left\|X - \tilde{X}_\Omega\right\|_1 + \left\|\tilde{X}_\Omega\right\|_1 \\
&= \left\|\left(X - \tilde{X}\right)_\Omega\right\|_1 + \left\|\tilde{X}_\Omega\right\|_1 \\
&\overset{(a)}{\leq} \frac{\epsilon}{1-3\epsilon}\left\|\left(X - \tilde{X}\right)_{\Omega^c}\right\| + \left\|\tilde{X}_\Omega\right\|_1 \\
&= \frac{\epsilon}{1-3\epsilon}\left\|-\tilde{X}_{\Omega^c}\right\| + \left\|\tilde{X}_\Omega\right\|_1 \\
&< \left\|\tilde{X}\right\|_1
\end{aligned}$$

where (a) follows from Proposition 4.3, and the last line follows from the fact that $\epsilon < \frac{1}{4}$, again from Proposition 4.3. Therefore, the unique solution of (P_1) is X with probability exceeding $1 - p^{-c}$, for some $c > 0$. As before, since Proposition 4.3 holds when $A = B$, the result of Theorem 4.1 continues to hold in this case. $\square$

4.5 Discussion

In this chapter we have introduced the notion of distributed sparsity for matrices. We have shown that when a matrix is X distributed sparse, and A, B are suitable random binary matrices, then it is possible to recover X from under-determined linear measurements of the form $Y = AXB^T$ via ℓ_1 minimization. We have also shown that

this recovery procedure is robust in the sense that if X is equal to a distributed sparse matrix plus a perturbation, then our procedure returns an approximation with accuracy proportional to the size of the perturbation. Our results follow from a new result about the properties of tensor products of random bipartite graphs. We also describe three interesting applications where our results would be directly applicable.

In future work, we plan to investigate the statistical behavior and sample complexity of estimating a distributed sparse matrix (and its exact support) in the presence of various sources of noise (such as additive Gaussian noise, and Wishart noise). We expect an interesting trade-off between the sketching dimension and the sample complexity.

Bibliographic note. The material in this chapter is based on joint work with Parikshit Shah, Badri Bhaskar, and Rob Nowak. Preliminary results appeared in [29, 30]. An extended version is under review by the IEEE Transactions on Information Theory and a preprint is available at [31].

Chapter 5

Inferring Species Trees from Multiple Loci

5.1 Introduction

In this chapter, we consider the problem of estimating the common evolutionary history, more precisely the *species tree*, of a set of n species using sequence data from multiple genes or loci. It is well known that the estimated genealogical history of a gene (*gene tree*) may be topologically distinct from the species tree that encapsulates it, possibly confounding phylogenetic analysis [98]. The subject of this chapter is an important source of such gene tree incongruence, known as *incomplete lineage sorting* (ILS), where two lineages fail to coalesce in their most recent common ancestral population. That failure may lead one of the lineages to first coalesce with a more distantly related population thereby producing a gene tree whose topology differs from the species tree that we are trying to estimate. Several species tree reconstruction methods have recently been developed that address ILS. See for instance [106, 93] and references therein. Many such methods rely on a statistical model known as the *multispecies coalescent* which, roughly speaking, generates gene trees by performing independent coalescent processes in each ancestral population and then assembling these together. This process is illustrated

in Figure 5.1 below and explained in a little more detail in Section 5.2.2. For more background on phylogenetic inference and coalescent theory see, e.g., [59, 67, 121].

The accuracy of multiloci reconstruction methods has been evaluated empirically, for instance, in [92, 95]. The focus of this chapter is the mathematical characterization of the performance of such methods. Prior theoretical work has focused mainly on statistical consistency under the multispecies coalescent; see e.g., [95, 35, 101, 94]. That is, assuming access to either correct gene trees or correct pairwise distances (or coalescence times) for each gene, a method is *statistically consistent* if it is guaranteed to converge on the correct species tree as the number of genes, m, tends to infinity. [116] studies the rates of convergence (in m) for several such methods. For instance, letting $f > 0$ denote the smallest branch length in the species tree, in the limit $f \to 0$, it was shown that the GLASS algorithm [101], which is an agglomerative clustering method in which the dissimilarity between each pair of species is taken to be the *minimum* of the coalescent times among the m genes, needs the number of genes m to scale as f^{-1}. On the other hand, m needs to scale as f^{-2} for the STEAC algorithm [95], which is also an agglomerative clustering method which instead uses the *average* of the coalescent times across the m genes as the measure of dissimilarity.

In reality, however, one has to estimate gene trees and coalescent times from finite, say, length-k molecular sequences. Taking into account the resulting estimation errors at the gene level is key to mathematically quantify and compare the performance of different methods (see e.g., [103, 137, 70]). Intuitively, for instance, the "minimum" used in GLASS may be significantly more sensitive to estimation errors than the "average" used in STEAC. We make progress towards this goal by performing the first full data requirement analysis of some species tree reconstruction methods.

Our contribution is two-fold. First it is known that, in order to reconstruct a single gene tree correctly with high probability, it is both necessary [129] and sufficient [50] for the sequence length k to scale as f^{-2}. Therefore, in light of this and the results in [116], one might expect that the total amount of data required, mk, must scale as f^{-3} and f^{-4} for GLASS and STEAC respectively. We show that, by a crucial modification of STEAC, one obtains an algorithm that is guaranteed to reconstruct the species tree exactly with high probability as long as m scales like f^{-2} and $k \geq 1$. In particular, it suffices for the overall sample complexity, mk, to scale like f^{-2} (which is much smaller than f^{-3} and f^{-4} in the regime of interest, where $f \ll 1$). Secondly, unlike GLASS, STEAC only works under the restrictive molecular clock assumption [121], where the mutation rates and population sizes are constant across the populations represented by the branches of the species tree. We extend the previous data requirement result beyond the molecular clock by devising a novel STEAC-like species tree reconstruction algorithm which we call METAL (Metric algorithm for Estimation of Trees based on Aggregation of Loci). This algorithm is a distance based method where the distances are defined by concatenating the molecular sequences corresponding to all the loci (genes).

5.2 Preliminaries and Notation

We will begin with a description of our modeling assumptions and introduce some notation that will be used throughout this chapter.

5.2.1 The Species Tree

At the heart of the model is an unknown *species tree* $S = (V, E)$ which represents the evolutionary history of n isolated populations; these isolated populations are represented by the size n leaf set L of this tree. The goal is to learn the structure of S. We assume that each branch $e \in E$ of the species tree corresponds to t_e generations of evolution and we assume that each generation in this branch has a population of size N_e. As is standard in coalescent theory, we will assign each branch $e \in E$, a length $\tau_e > 0$ in coalescent time units defined as $\tau_e \triangleq t_e/N_e$. The smallest branch length, $f \triangleq \min_e \tau_e$, will play an important role in our analysis and in particular, we will be interested in the case where f is very small. For a pair of vertices $X, Y \in V$, we will use $\pi^S_{XY} \subset E$ to denote the unique path connecting X and Y in S and τ_{XY} will denote the length of this path. Notice that $\{\tau_{AB}\}_{A,B\in L}$ forms a metric on the set L and such a metric that can be written as a sum of path lengths on a tree is called an *additive metric* (see e.g., [121]) with respect to that tree. If we additionally assume that the population sizes in each branch are equal to some constant N, then $\{\tau_{AB}\}_{A,B\in L}$ forms an ultrametric with respect to S, i.e., for any three leaves A, B, C such that S restricted to A, B, C has the topology $((A, B), C)$[1], we have that

$$\tau_{AB} \leq \tau_{AC} = \tau_{BC}.$$

We will let $\Delta \triangleq \max_{A,B\in L} \tau_{AB}$ denote the diameter of the species tree. Finally, To each branch $e \in E$, we will also associate a mutation rate, μ_e and we will let $\mu_L \triangleq \min_{e\in E} \mu_e$ and $\mu_U \triangleq \max_{e\in E} \mu_e$ denote the smallest and largest mutation rates, respectively.

[1]We will sometimes find it useful to represent trees in the so called Newick Format. For instance, the Newick representations of the trees labelled Gene 1 and Gene 2 in Figure 5.1 are $((A, B), C)$ and $(A, (B, C))$, respectively.

5.2.2 The Multispecies Coalescent and the Gene Trees

Following [111], we assume that a *multispecies coalescent* (MSC) process produces m (independent) random genealogies $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(m)}$ based on S. These encode, say, the evolutionary history of m different genes or loci on the genome and will be referred to as *gene trees* henceforth.

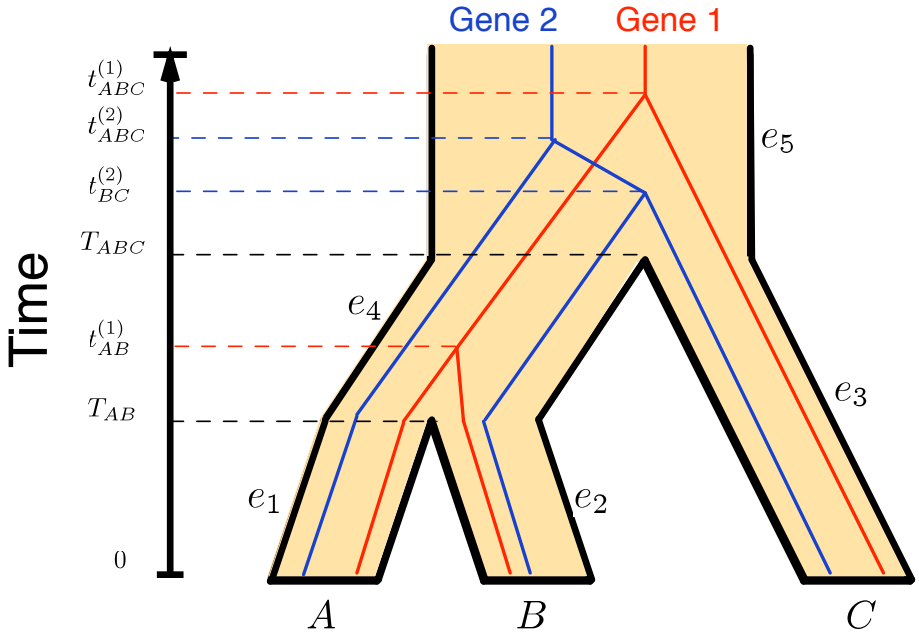


Figure 5.1: A species tree (the thick, shaded tree) and two samples from the multispecies coalescent. Notice that while the topology of Gene 1 agrees with the species tree, the topology of Gene 2 does not.

It is easier to understand the MSC constructively and in the case where the population size N_e in each branch $e \in E$ is a constant N. Consider the 3 species example of Figure 5.1, where the thick, shaded tree is the species tree S with edges $\{e_i\}_{i=1}^5$. As is standard in coalescent theory, we will think of time as running backwards, that is, time (in coalescent time units) starts at 0 at the leaves and increases towards the root of the tree. By T_{AB} (resp. T_{ABC}), we mean the time when the parent population of A and B (resp. the

parent population of A, B, and C) branch (or speciate). Let us first consider one random draw from the MSC, i.e., the case of one particular gene, Gene 1. A, B, and C each have a copy (or allele) of Gene 1 and the MSC describes the evolutionary history of the lineages corresponding to these alleles. From time 0 until T_{AB}, the lineages corresponding to A and B are in isolated populations and hence do not "coalesce". However, once these lineages reach the parent population of A and B (represented by the branch e_4), they have a chance to coalesce. According to the MSC, the coalescence happens after a random time drawn according to the Exp(1) distribution, that is,

$$\mathbb{P}\left[t_{AB}^{(1)} - T_{AB} \geq x\right] = 1 - e^{-x}, \qquad x \geq 0. \tag{5.1}$$

Now, the coalesced A-B lineage and the lineage corresponding to C do not interact until time T_{ABC}, which is when they find themselves in a common population. They then coalesce at a random time $t_{ABC}^{(1)}$ which is again such that $t_{ABC}^{(1)} - T_{ABC} \sim \text{Exp}(1)$. This gives us a random gene tree with the topology $((A, B), C)$. To contrast with this, consider the case of Gene 2. Here, the lineages corresponding to the alleles in A and B do not coalesce in e_4 (since the randomly drawn coalescence time was more than the length of e_4). So, at time T_{ABC}, there are three lineages present in the branch e_5. When there are multiple lineages in the same population, according to the MSC, each pair independently coalesces again after a random time period drawn according to the Exp(1) distribution. In this case, the genealogies of B and C alleles coalesce (at time $t_{BC}^{(2)}$) before A and B, thus giving us a second random tree with topology $(A, (B, C))$. Notice that while the genealogy (evolutionary history) of Gene 1 agrees with that of the species, the genealogy of Gene 2 does not. This is an example of incomplete lineage sorting which, as

mentioned earlier, is a fundamental road block for learning the tree of life.

We refer the reader to [111] for more details on the multispecies coalescent but, we will state the model here for the sake of completeness. Before we proceed, we will record a simple fact about the exponential distribution: If $X_1, \ldots, X_p \overset{\text{iid}}{\sim} \text{Exp}(1)$, then $\min_{i\in\{1,\ldots,p\}} X_i \sim \text{Exp}(p)$. This follows since

$$\mathbb{P}\left(\min_{i\in\{1,\ldots,p\}} X_i \geq t\right) = \prod_{i=1}^{p} P(X_i \geq t) = e^{-pt}. \tag{5.2}$$

The density of the likelihood of a gene tree $\mathcal{G}^{(i)} = \left(\mathcal{V}^{(i)}, \mathcal{E}^{(i)}\right)$ can now be written down as follows. We will focus our attention on the branch $e \in E$ of the species tree and for the gene tree $\mathcal{G}^{(i)}$, let $I_e^{(i)}$ and $O_e^{(i)}$ be the number of lineages entering and leaving the branch e respectively. For instance, consider Gene 1 in Figure 5.1. Here, two lineages enter the branch e_4 and one lineage leaves it. On the other hand, in the case of Gene 2 in Figure 5.1, two lineages enter the branch e_4 and two lineages leave it. Let $t_{e,s}^{(i)}, s = \left\{1, 2, \ldots, I_e^{(i)} - O_e^{(i)} + 1\right\}$ be the $s-$th coalescent time corresponding to $\mathcal{G}^{(i)}$ in the branch e. Recall that each pair of lineages in a population can coalesce at a random time drawn according to the Exp(1) distribution independently of each other. Therefore, after the $(s-1)$-th coalescent event at time $t_{e,s-1}^{(i)}$, there are $I_e^{(i)} - s + 1$ surviving lineages in branch e and the likelihood that the $s-$th coalescence time in branch e is $t_{e,s}^{(i)}$ corresponds to the event that the minimum of $\binom{I_e^{(i)}-s+1}{2}$ random variables distributed according to Exp(1) has the value $t_{e,s}^{(i)} - t_{e,s-1}^{(i)}$. Therefore using (5.2), the density of the likelihood of $\mathcal{G}^{(i)}$ can be written as

$$\prod_{e\in E} \prod_{s=1}^{I_e^{(i)}-O_e^{(i)}+1} \exp\left\{-\binom{I_e^{(i)} - s + 1}{2}\left[t_{e,s}^{(i)} - t_{e,s-1}^{(i)}\right]\right\}, \tag{5.3}$$

ÂňâĂć where, for convenience, we let $t^{(i)}_{e,0}$ and $t^{(i)}_{e,I^{(i)}_e - O^{(i)}_e + 1}$ be respectively the divergence times of the population in e and of its parent population.

5.2.3 Observation Model and The Inference Problem

Much of the prior work on understanding the theoretical complexity of learning species trees from multiple loci (or gene trees) has focused on the case where exact gene trees are available. However, in reality one needs to estimate these gene trees from molecular sequences and indeed there has been a recent thrust towards understanding the effect of errors in estimating the gene trees (see e.g., [103, 137, 70]). Our approach will be to take this error into account explicitly and in fact bypass the reconstruction of gene trees altogether.

We model the sample generation process according to the standard Jukes-Cantor (JC) model (see e.g., [121]). That is, given a gene tree $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we will associate to each $\tilde{e} \in \mathcal{E}$, a probability $p_{\tilde{e}}$ (whose dependence on the length of $\tilde{e}$ we will make explicit below). Then, the JC model assigns a character from $\{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}\}$ uniformly at random to the root of $\mathcal{G}$. Moving away from the root, with probability $p_{\tilde{e}}$, each edge $\tilde{e}$ changes the state of its ancestor to one of the other three, chosen uniformly at random. The states at the leaves of $\mathcal{G}$ are assembled into a length n vector to get the first sample; this process is repeated k times to generate the data set. Notice that k models the number of sites or the sequence length of each gene.

Now, we will define $p_{\tilde{e}}$. To each edge $\tilde{e}$ of the random gene tree $\mathcal{G}$ is associated a random length $\sigma_{\tilde{e}}$ according to the MSC. Also, given an edge $e \in E$ of the species tree, we will write $\sigma_{e \cap \tilde{e}}$ to denote the length of the portion $\tilde{e}$ that overlaps with e. This lets

us define the effective (mutation rate adjusted) branch lengths, $\delta_{\tilde{e}} = \sum_{e \in E} \mu_e \sigma_{e \cap \tilde{e}}$. As before, for any two vertices $X, Y \in \mathcal{V}$, $\pi^{\mathcal{G}}_{XY}$ denotes the path joining X and Y in $\mathcal{G}$ and σ_{XY} (resp. δ_{XY}) denotes the length of this path under σ (resp. under δ). Now, for an edge $\tilde{e} \in \mathcal{E}$, we define $p_{\tilde{e}} \triangleq \frac{3}{4}(1 - e^{-\frac{4}{3}\delta_{\tilde{e}}})$. Notice that this definition implies that the probability p_{XY} of disagreement between the characters at vertices X and Y satisfies, $p_{XY} = \frac{3}{4}(1 - e^{-\frac{4}{3}\delta_{XY}})$.

The goal then, is to learn the structure of S given the data $\{\chi^{ij}\}_{i \in [m], j \in [k]}$ which is an $n \times m \times k$ array composed of the characters $\{\mathtt{A}, \mathtt{T}, \mathtt{G}, \mathtt{C}\}$, where $\{\chi^{ij}\}_{j \in [k]}$ is the data generated from the random gene tree $\mathcal{G}^{(i)}$ according to the Jukes-Cantor model.

The Jukes-Cantor model was chosen because it lends itself to easy presentation. Since the techniques developed here are *distance-based*, all our results can be generalized to the more realistic Generalized Time-Reversible (GTR) model [131] using spectral techniques as in [115, 100].

5.3 Main Results

We now state the main results of the chapter. First, we will deal with the case where the strong molecular clock [121] assumption holds. We will then turn our attention to the more general case that does away with this assumption.

5.3.1 The Molecular Clock Assumption Holds

Assuming that the molecular clock hypothesis holds is often unrealistic; it is equivalent to believing that all extant and ancestral populations have the same population size and that the mutations happen at the same rate through time and across populations. It

has however proven to be a useful abstraction for developing powerful methods. In our setting, this is equivalent to assuming that for all $e \in E$, $\mu_e = \mu > 0$, and $N_e = N$, both constants independent of e.

In order to infer the species tree from samples, we will begin by defining a distance measure on the leaves. For each pair of leaves $A, B \in L$, we define

$$\widehat{p}_{AB} = \frac{1}{mk} \sum_{i \in [m], j \in [k]} \mathbb{1}\{\chi_A^{ij} \neq \chi_B^{ij}\}, \tag{5.4}$$

which can be thought of as the normalized hamming distance between the concatenated molecular sequences corresponding to species A and B. Our first result, which is proved in Section D.1, is that, in expectation, $\{\widehat{p}_{AB}\}_{A,B \in L}$ is not only a metric on L, but is in fact an ultrametric with respect to S.

Theorem 5.1. *$\{\mathbb{E}[\widehat{p}_{AB}]\}_{A,B \in L}$ forms an ultrametric with respect to the true species tree S. In fact, for any triple $A, B, C \in L$ with the topology $((A, B), C)$ in S, we have*

$$\mathbb{E}[\widehat{p}_{AC}] = \mathbb{E}[\widehat{p}_{BC}] > \mathbb{E}[\widehat{p}_{AB}] + \frac{3e^{-\frac{4}{3}\mu\tau_{AC}}\mu}{8\mu + 3} f. \tag{5.5}$$

This result inspires the following procedure for reconstructing S: Use $\{\widehat{p}_{AB}\}_{A,B \in L}$ as a dissimilarity measure for L and use a standard algorithm that accepts a dissimilarity measure and returns an ultrametric tree (see e.g., [59, 121] for background on distance based methods). For the sake of simplicity, we may assume that we use the UPGMA algorithm [126], the standard method for bottom-up agglomerative clustering, in order to produce an ultrametric tree. Then, recalling that μ denotes the (common) mutation rate across the populations represented by the species tree S, and Δ denotes diameter of

S, we have the following performance guarantee.

Theorem 5.2. *Given an $\epsilon > 0$, using UPGMA on L with the dissimilarity measure $\{\widehat{p}_{AB}\}_{A,B\in L}$ results in the correct tree S being output with probability no less than $1-\epsilon$ as long as the number of genes m, and the sequence length k satisfy*

$$m \geq C_1(\mu, \Delta, n, \epsilon) \times f^{-2} \quad \textit{and} \quad k \geq 1, \tag{5.6}$$

where $C_1(\mu, \Delta, n, \epsilon) = \frac{16\, e^{\frac{8}{3}\mu\Delta}(8\mu+3)^2}{9\mu^2} \log\left(\frac{8\binom{n}{3}}{\epsilon}\right)$.

Theorem 5.2, which is proved in Section D.2, tells us that the above procedure succeeds with high probability as long as we get molecular sequences of length at least one from at least $\mathcal{O}(f^{-2})$ genes. That is, a total sequence length of $mk = \mathcal{O}(f^{-2})$ suffices for reliable learning.

Notice that the procedure we propose is similar to the STEAC algorithm [95] except instead of using the average coalescent time as the distance measure, we use (5.4), which can be considered as the normalized hamming distance. It turns out that this modification is crucial to obtaining our improved sample complexity result.

5.3.2 The Molecular Clock Assumption Does Not Hold

We will now consider the more general case where the strong molecular clock assumption does not hold. That is, we will assume that each branch e of the species tree has a (possibly) distinct mutation rate μ_e and population size N_e.

First, we observe that $\{\mathbb{E}[\widehat{p}_{AB}]\}_{A,B\in L}$ as defined above is no longer an ultrametric with respect to S and therefore, the above procedure (and for a similar reason, the STEAC

algorithm) cannot be used to recover the species tree. In such situations, one usually turns to distance methods that rely on the 4-point condition (see e.g., [121]). However, it is not immediately clear how to define a metric that satisfies the 4-point condition in our setting. Our next result, which is arguably the most important contribution of this chapter, shows that this can be done. As before, we will first consider an idealized measure of dissimilarity as follows:

$$d_{AB} = -\frac{3}{4}\log\left(1 - \frac{4}{3}\mathbb{E}\left[\widehat{p}_{AB}\right]\right), \quad A, B \in L,$$

where $\widehat{p}_{AB}$ is as defined in (5.4). Our next result, which parallels Theorem 5.1, shows that this "idealized" dissimilarity measure is actually an *additive metric* with respect to S. Recall that this means that the four point condition holds, i.e., for a quadruple of leaves A, B, C, D that are such that the topology of S restricted to these 4 leaves is $((A, B), (C, D))$ or $(((A, B), C), D)$, the above distances satisfy

$$d_{AB} + d_{CD} \leq d_{AC} + d_{BD} = d_{AD} + d_{BC}.$$

See [121], for instance, for more information about tree metrics.

Theorem 5.3. *The set of dissimilarities $\{d_{AB}\}_{A,B\in L}$ forms an additive metric with respect to S. In fact, suppose the leaves $A, B, C, D \in L$ are such that either $((A, B), (C, D))$ or $(((A, B), C), D)$ holds with respect to S, then*

$$d_{AC} + d_{BD} = d_{AD} + d_{BC} > d_{AB} + d_{CD} + \alpha_{\text{add}}, \tag{5.7}$$

where $\alpha_{\text{add}} = \frac{3}{4}\log\left(\frac{8}{3}\mu_L(1 - e^{-f}) + 1\right) > 0$ and $\mu_L \triangleq \min_{e\in E}\mu_e$ is the smallest mutations

rate, as defined in Section 5.2.1.

It is somewhat surprising that this result is true. It tells us that if one ignores the fact that there are multiple loci and pretends as though all samples came from a single gene tree, then the gene tree estimated from this "concatenated molecular sequence" has the same topology as S. Furthermore, this result is also interesting since phylogenetic mixtures are known to cause problems for distance-based methods [130]. We prove Theorem 5.3 in Section D.3.

In light of this, we propose the following algorithm to reconstruct S. First, we define the following sample-based *corrected* measure of dissimilarity (with $\widehat{p}_{AB}$ as defined in (5.4))

$$\widehat{d}_{AB} \triangleq -\frac{3}{4}\log\left(1-\frac{4}{3}\widehat{p}_{AB}\right). \tag{5.8}$$

Now, use any quartet-test based algorithm (like Neighbor Joining [120]) which returns an additive tree using $\{\widehat{d}_{AB}\}_{A,B\in L}$ defined as in (5.8) as the input dissimilarity measure. We call this algorithm METAL (for Metric algorithm for Estimation of Trees based on Aggregation of Loci).

Recall that μ_U and μ_L are respectively the maximum and minimum mutation rates, and Δ is the diameter of the species tree S (c.f. Section 5.2.1). We then have the following result.

Theorem 5.4. *For any $\epsilon > 0$, METAL succeeds in reconstructing (the unrooted version*

of) S with probability at least $1-\epsilon$ as long as m and k satisfy

$$k \geq 1 \text{ and } m \geq \frac{e^{\frac{8\mu_U\Delta}{3}}(8\mu_U+3)^2(24+8\alpha_{\text{add}})^2}{162\alpha_{\text{add}}^2}\log\left(\frac{16\binom{n}{4}}{\epsilon}\right) \tag{5.9}$$

where $\alpha_{\text{add}} = \frac{3}{4}\log\left(\frac{8}{3}\mu_L(1-e^{-f})+1\right)$.

In the limit as $f \to 0$*, the right side above approaches*

$$C_2(\mu_U,\mu_L,\Delta,n,\epsilon) \times f^{-2}, \text{ where } C_2(\mu_U,\mu_L,\Delta,n,\epsilon) = \frac{8e^{\frac{8\mu_U\Delta}{3}}(8\mu_U+3)^2}{9\mu_L^2}\log\left(\frac{16\binom{n}{3}}{\epsilon}\right).$$

Remark. *Following [50], the diameter* Δ *can be replaced by the (often much smaller)* depth[1] *of the tree by employing a distance method that uses only those distances that are "small enough".*

We prove Theorem 5.4 using arguments that are similar in spirit to those in the proof of Theorem 5.2. We refer the reader to Section D.4 for the exact details.

Theorem 5.4 tells us that as long as m scales like $\mathcal{O}(f^{-2})$ and $k \geq 1$, the species tree can be reconstructed (upto the location of the root) reliably. It should be noted here that we assume that for each population/branch $e \in E$, the mutation rate μ_e is constant across gene trees; generalizing this analysis to the case where the mutation rates are allowed to change is an interesting avenue for future work.

[1] *The depth of an edge e is the length (under τ) of the shortest path between two leaves crossing e; the depth of a tree is the maximum edge depth.*

5.4 Discussion

Irrespective of the sequence length k of each gene, the number of genes m required needs to satisfy $m \in \Omega(f^{-1})$ for consistent species tree estimation. To see this, consider the species tree in Figure 5.1. Given m gene trees drawn according to the MSC based on this species tree, the probability that none of them have a coalescent event in branch e_4 is given by $e^{-m\tau_{e_4}}$ (this is the probability that m independent exponentials are bigger than τ_{e_4}). Therefore, if $m < \tau_{e_4}^{-1}$, then with probability greater than e^{-1}, none of the m the gene trees have a coalescence event in e_4, that is, there is no evidence for the existence of this branch from the sample. This argument can also be formalized by observing that any algorithm that is able to estimate S reliably should be able to perform a reliable hypothesis test between two shifted exponential distributions. Therefore, this result follows from the fact that $D_{\mathrm{KL}}\left(p(x;\tau_{AB}+f)\|p(x;\tau_{AB})\right) = f$, where $p(x;a) = e^{-(x-a)}\mathbb{1}\left\{x \geq a\right\}$ and $D_{KL}\left(\cdot\|\cdot\right)$ is the Kullback-Liebler divergence [**?**].

On the other hand, we know from [129] that even without the confounding effect of the multispecies coalescent, a total sequence length $(m \times k)$ of at least $\Omega(f^{-2})$ is needed for consistent estimation. These two together imply that there is a constant $C > 0$ such that m needs to satisfy the following for consistent estimation of the species tree

$$m \geq C \max\left\{f^{-1}, \frac{f^{-2}}{k}\right\}. \tag{5.10}$$

As mentioned earlier, the results in this chapter show that $m \in \mathcal{O}(f^{-2})$ is achievable irrespective of the value of k, i.e., in particular, a total data set size of $mk \in \mathcal{O}(f^{-2})$ is achievable. Prior to this, to the best of our knowledge, the best complexity bounds were provably attained by GLASS [101] (as shown in [116]) which requires that $m \geq \mathcal{O}(f^{-1})$

and $k \geq \mathcal{O}(f^{-2})$, i.e., a total data set size of $mk \in \mathcal{O}(f^{-3})$.

This raises two very interesting open questions. (A) What is the precise tradeoff between m and k for reliable recovery of S and in particular, is it possible to devise an algorithm that recovers S given $m \in o(f^{-2})$ when the sequence length, k, is moderate, say, $\mathcal{O}(f^{-1})$? (B) Is there a procedure that attains all points (values of m and k) in this tradeoff, as opposed to the current situation where it appears as though GLASS meets the lower bounds for large k and METAL meets the lower bound for small k?

It will also be interesting to investigate phylogeny reconstruction using actively selected pairwise distances by techniques similar to those in Chapter 3. In fact, [77] provides an ideal "population version" algorithm for doing this and the statistical performance of this algorithm may be analyzed using, say, the techniques presented in this chapter. This could be especially relevant in the modern era of genome sequencing, where it has become extremely cost effective to obtain a large number of "short reads" (short genome sequences) of the genome, which are then, using a computationally intensive procedure, *assembled* into a full genome before performing any analysis. Therefore, if one is able to identify a (statistically efficient) technique for computing a reasonable (dis)similarity measure from the reads without assembly, then it might be very attractive to perform phylogenomic analysis using as few pairwise (dis)similarity computations as possible.

Bibliographic note. The material presented in this chapter is based on joint work with Rob Nowak and Sebastien Roch. A preliminary version [28] is to appear in the Proceedings of the IEEE Symposium on Information Theory (2014). The full version is under review and a preprint is available at [27].

Appendix A

Proofs from Chapter 2

Proposition 2.4. *Suppose that the unknown routing tree is a balanced $\ell-$ary tree (where each non-leaf node has ℓ children) with n leaves (end hosts) and suppose that the similarity matrix satisfies a Margin Condition of order δ. Algorithm 2 can recover the proper DFS ordering using no more than $p(\ell)n\log_\ell n$, where $p(\ell) \triangleq \left(\frac{\ell+1}{2} - \frac{1}{\ell}\right)$.*

Proof. We let $f_\ell(n)$ denote the number of probes required by Algorithm 2 to discover the topology of a balances $\ell-$ary tree. Observe that it satisfies the following recursive inequality.

$$\begin{aligned} f_\ell(n) &\leq n + \frac{\ell-1}{\ell}n + \cdots + \frac{2}{\ell}n + \ell f_\ell\left(\frac{n}{\ell}\right) \\ &= \frac{n}{\ell}p(\ell) + \ell f_\ell\left(\frac{n}{\ell}\right). \end{aligned}$$

Now, we can use the recursion equation to obtain an upper bound of $p(\ell)n\log_\ell n$ on $f_\ell(n)$. This concludes the proof. □

Now, we prove Theorem 2.2.

Theorem 2.2. *Using Algorithm 4 and Algorithm 1, the logical topology for a balanced $\ell-$ary tree with n end hosts can be reconstructed using no more than $n((\ell+9)\log_2 n + 1)$*

pairwise similarities, provided the similarity matrix satisfies the Monotonicity Condition.

Proof. As above, we define $f_\ell(n)$ denote the number of probes required to discover the topology of a balances $\ell-$ary tree. Let $s_m(n) \leq m^2 \log_m n$ be the number of similarities required to find the exact split by agglomerative clustering. Observe that these satisfy the following recursive inequality.

$$\begin{aligned} f_\ell(n) &\leq (n + s_m(n)) + \left(\frac{\ell-1}{\ell}n + s_m\left(\frac{\ell-1}{\ell}n\right)\right) + \cdots + \left(\frac{2}{\ell}n + s_m\left(\frac{2}{\ell}n\right)\right) + \ell f_\ell\left(\frac{n}{\ell}\right) \\ &= np(\ell) + \sum_{i=2}^{\ell} s_m\left(\frac{i}{\ell}n\right) + \ell f_\ell\left(\frac{n}{\ell}\right) \end{aligned}$$

The above recursion can be simplified to give the following upper bound.

$$\begin{aligned} f_\ell(n) &\leq np(\ell)\log_\ell n + \left(\sum_{j=1}^{\log_\ell n} \ell^j\right) m^2 \log_m n \\ &\leq np(\ell)\log_\ell n + nm^2 \log_m n. \end{aligned}$$

Given that we control the agglomerative clustering procedure Algorithm 4, we can reduce the total pairwise measurements needed by setting $m = 3$ (since $m \geq 3$). This results in the total pairwise measurements needed by Algorithm 4 and Algorithm 1 to resolve DFS ordering to be no more than $n((\ell + 9)\log_2 n)$. Combined with Proposition 2.3, we see that a total of $n((\ell + 9)\log_2 n + 1)$ pairwise similarities that satisfy the monotonic condition suffice. □

Appendix B

Proofs from Chapter 3

B.1 Proof of Proposition 3.2

Defining the randomly chosen subset of items of size $n_V = |S_V|$, we now bound the size of this subset, such that the two properties hold with high probability.

First, we show that if n_V is large enough, the set of items S_V contains at least one item in each of the top-most split subclusters, C_L and C_R.

Lemma B.1. *If $n_V \geq \frac{\log(6/\delta_C)}{\log(1/(1-\eta))}$, then with probability $> 1 - \frac{\delta_C}{3}$, S_V contains at least one item in each of the top-most split subclusters C_L and C_R.*

Proof. Suppose we draw n_V items uniformly at random from C, then

$$\begin{aligned} P(\text{all items} \in C_L \text{ or all items} \in C_R) &\leq \left(\frac{|C_L|}{|C|}\right)^{n_V} + \left(\frac{|C_R|}{|C|}\right)^{n_V} \\ &\leq 2\,(1-\eta)^{n_V} \end{aligned}$$

where η denotes the balance factor. Therefore, $2(1-\eta)^{n_V} \leq \delta_C/3$ if

$$n_V \geq \frac{\log(6/\delta_C)}{\log(1/(1-\eta))}.$$

□

Second, the cluster count values $c_{i,j}$ must reveal whether two items x_i, x_j are in the same cluster (i.e., $i, j \in C_L$ or $i, j \in C_R$) or whether items x_i, x_j are in differing clusters (i.e., $i \in C_L$ and $j \in C_R$, or $i \in C_R$ and $j \in C_L$), given that the similarity $s_{i,j}$ does not have a violation. Define $\Omega_{i,j}$ to be the event that the similarity between items x_i, x_j does not have a violation, then we can state the following lemma.

Lemma B.2. *With probability greater than* $1 - \frac{\delta_C}{3}$*, comparing the outlier count values* $c_{i,j}$ *to a set threshold* $\gamma = \gamma' n_V$*, where* $\gamma' \in (0, 1)$ *is a constant, will identify whether the items* x_i, x_j *are in same clusters or different, given that the similarity value* $s_{i,j}$ *does not have a violation, if*

$$n_V \geq \frac{\log(6/\delta_C)}{2\min\left(\left(\gamma' - 1 + (1-q)^2\right)^2, \left((1-q)^2\eta - \gamma'\right)^2\right)}$$

and the violation probability q *and balance factor* η *satisfy* $(1-(1-q)^2) < \gamma' < (1-q)^2\eta$.

For n_V randomly chosen items, the expected outlier count conditioned on the choice of x_i, x_j and the similarity value $s_{i,j}$ not having a violation can be stated as,

$$\begin{aligned} E\left[c_{i,j} \mid (x_i, x_j \in C_L \text{ or } x_i, x_j \in C_R), \Omega_{i,j}\right] &\geq (1-q)^2 \eta n_V \\ E\left[c_{i,j} \mid (x_i \in C_R, x_j \in C_L \text{ or } x_i \in C_L, x_j \in C_R), \Omega_{i,k}\right] &\leq \left(1 - (1-q)^2\right) n_V. \end{aligned}$$

Since $\gamma = \gamma' n_V$, we see that $(1-(1-q)^2)n_V < \gamma < (1-q)^2\eta n_V$. Therefore, the outlier counts will resolve whether the two items are in the same or differing clusters if, with high probability, the observed cluster count $c_{i,j}$ does not deviate from the conditional expectation by too much. Using Hoeffding's Inequality, we have,

$P\left(c_{i,j} \leq \gamma \,|\, (x_i, x_j \in C_L \text{ or } x_i, x_j \in C_R), \Omega_{i,j}\right) \leq \exp\left(-2n_V\left((1-q)^2\eta - \gamma'\right)^2\right)$ and $P\left(c_{i,j} > \gamma \,|\, (x_i \in C_R, x_j \in C_L \text{ or vice versa}), \Omega_{i,j}\right) \leq \exp\left(-2n_V\left(\gamma' - 1 + (1-q)^2\right)^2\right)$. Both these probabilities are $\leq \delta_C/6$ if

$$n_V \geq \frac{\log(6/\delta_C)}{2\min\left(\left(\gamma' - 1 + (1-q)^2\right)^2, \left((1-q)^2\eta - \gamma'\right)^2\right)}$$

Therefore, combining Lemma B.1 and Lemma B.2, the two properties claimed in Proposition 3.2 hold with probability $> 1 - 2\delta_C/3$, if

$$\begin{aligned} n_V &= \frac{c}{2}\log n \\ &\geq \max\left(\frac{\log(6/\delta_C)}{\log(1/(1-\eta))}, \frac{\log(6/\delta_C)}{2\min\left(\left(\gamma' - 1 + (1-q)^2\right)^2, \left((1-q)^2\eta - \gamma'\right)^2\right)}\right) \end{aligned}$$

B.2 Proof of Proposition 3.3

Define $\Phi_{i,j}$ as the event that both similarities $s_{i,k}, s_{j,k}$ do not have a violation and the cluster counts $c_{i,k}, c_{j,k}$ are correct, given the threshold property in Lemma B.2. Therefore, we can state the probabilities,

$$\begin{aligned} P\left(\Phi_{i,j}\right) &\geq \left(1 - \frac{\delta_C}{3}\right)^2 (1-q)^2 \\ P\left(\Phi_{i,j}^C\right) &\leq 1 - \left(1 - \frac{\delta_C}{3}\right)^2 (1-q)^2 \end{aligned}$$

Then defining the conditional expectations of the agreement counts, $a_{i,j}$ from Equation 3.4,

$$\begin{aligned} E\left[a_{i,j} \mid x_j \in C_j, x_i \notin C_j\right] &= P\left(\Phi_{i,j}^{C}\right) n_A + P\left(\Phi_{i,j}\right) 0 \\ &\leq \left(1-(1-q)^2\left(1-\frac{\delta_C}{3}\right)^2\right) n_A \\ E\left[a_{i,j} \mid x_i, x_j \in C_j\right] &\geq P\left(\Phi_{i,j}\right) n_A \\ &\geq (1-q)^2\left(1-\frac{\delta_C}{3}\right)^2 n_A \end{aligned}$$

For two items not in the same cluster and two items in the same cluster, respectively.

Thus, comparing the agreement counts, $a_{i,j}$, to a threshold of $\frac{1}{2}n_A$ will reveal the correct placement of x_i with high probability. We bound the probability that two items in the different clusters have agreement counts, $a_{i,j}$, greater than the threshold, and the probability that two items in the same cluster have agreement counts, $a_{i,j}$, less than the threshold by a small number $\delta_C/6$

$$\begin{aligned} P\left(a_{i,j} \geq \frac{1}{2} \mid x_i \notin C_j, x_j \in C_j\right) &\leq \frac{\delta_C}{6} \\ P\left(a_{i,j} < \frac{1}{2} \mid x_i, x_j \in C_j\right) &\leq \frac{\delta_C}{6}. \end{aligned}$$

Using Chernoff's Bound, we obtain,

$$n_A = \frac{c}{2}\log n \quad \geq \quad \frac{\log \frac{6}{\delta_C}}{2\left(\left(1-\frac{2\delta_C}{3}\right)^2(1-q)^2-\frac{1}{2}\right)}.$$

Thus, the placement of an item x_i can be correctly recovered with probability $> 1-\delta_C/3$, if c is large enough (as stated above). The probability that all $x_i \in C$ are correctly recovered is $> 1 - N\delta_C/3$. Thus, the statement of Proposition 3.3 holds with probability

$> 1 - N\delta_C/3$.

B.3 Proof of Proposition 3.4

Consider a tree structure of n items with balance factor $\eta \leq 1/2$. After ℓ levels, the number of items in the largest cluster are bounded by $(1-\eta)^{\ell} N$. If L denotes the maximum number of levels, then there can only be 1 item in the largest cluster after L levels, we have $1 \leq (1-\eta)^{L} N$, or $L \leq \frac{\log N}{\log(\frac{1}{1-\eta})}$.

Appendix C

Proofs from Chapter 4

C.1 Proof of Lemma 4.7

Lemma 4.7. *Suppose that $G_1 = ([p], [m], E_1)$ and $G_2 = ([p], [m], E_2)$ are two independent uniformly random $\delta-$left regular bipartite graphs with $\delta = \mathcal{O}(\log p)$ and $m = \Omega(\sqrt{dp}\log p)$. Let $\Omega \in \mathfrak{W}_{d,p}$ be fixed. Then there exists an $\epsilon \in \left(0, \frac{1}{4}\right)$ such that $G_1 \otimes G_2$ has the following properties with probability exceeding $1 - p^{-c}$, for some $c > 0$.*

1. *$|N(\Omega)| \geq p\delta^2(1-\epsilon)$.*
2. *For any $(i, i') \in ([p] \times [p]) \setminus \Omega$ we have $|N(i, i') \cap N(\Omega)| \leq \epsilon\delta^2$.*
3. *For any $(i, i') \in \Omega$, $|N(i, i') \cap N(\Omega \setminus (i, i'))| \leq \epsilon\delta^2$.*

Moreover, all these claims continue to hold when G_2 is the same as G_1.

Proof. We will first prove this lemma for the case when $G_1 = G_2$. With a few minor modifications, one can readily get a proof for the easier case when G_1 and G_2 are drawn independently. The details of these modifications are outlined in Section C.1.1.

Let $\mathcal{E}_1, \mathcal{E}_2$ and, $\mathcal{E}_3$ respectively denote the events that the implications (1), (2) and,

(3) are true. Notice that

$$\begin{aligned}\mathbb{P}\left(\mathcal{E}_1^c \cup \mathcal{E}_2^c \cup \mathcal{E}_3^c\right) &\leq \mathbb{P}(\mathcal{E}_1^c) + \mathbb{P}(\mathcal{E}_2^c) + \mathbb{P}(\mathcal{E}_3^c) \\ &= \mathbb{P}(\mathcal{E}_1^c) + \mathbb{P}(\mathcal{E}_2^c \mid \mathcal{E}_1)\mathbb{P}(\mathcal{E}_1) + \mathbb{P}(\mathcal{E}_2^c \mid \mathcal{E}_1^c)\mathbb{P}(\mathcal{E}_1^c) + \mathbb{P}(\mathcal{E}_3^c \mid \mathcal{E}_1)\mathbb{P}(\mathcal{E}_1) \\ &\qquad + \mathbb{P}(\mathcal{E}_3^c \mid \mathcal{E}_1^c)\mathbb{P}(\mathcal{E}_1^c) \\ &\leq 3\mathbb{P}(\mathcal{E}_1^c) + \mathbb{P}(\mathcal{E}_2^c \mid \mathcal{E}_1) + \mathbb{P}(\mathcal{E}_3^c \mid \mathcal{E}_1)\end{aligned}$$

Our strategy will be to upper bound $\mathbb{P}(\mathcal{E}_1^c), \mathbb{P}(\mathcal{E}_2^c \mid \mathcal{E}_1)$, and $\mathbb{P}(\mathcal{E}_3^c \mid \mathcal{E}_1)$. Suppose the bounds were p_1, p_2 and, p_3 respectively, then it is easy to see that

$$\mathbb{P}\left\{\mathcal{E}_1 \cap \mathcal{E}_2 \cap \mathcal{E}_3\right\} \geq 1 - \max\{3p_1, p_2, p_3\}. \tag{C.1}$$

Part 1. We will first show that $\mathbb{P}(\mathcal{E}_1^c)$ is small. Since Ω is $d-$distributed, the "diagonal" set $\mathcal{D} := \{(1,1), \ldots, (p,p)\}$ is a subset of Ω. Now, notice that for $j \neq j' \in [m], i \in [p]$,

$$\mathbb{P}\left[(j,j') \in N((i,i))\right] = \frac{\delta(\delta-1)}{m(m-1)} \tag{C.2}$$

This implies that

$$\mathbb{P}\left[(j,j') \notin N(\mathcal{D})\right] = \left(1 - \frac{\delta(\delta-1)}{m(m-1)}\right)^{|\mathcal{D}|}$$

Therefore, we can bound the expected value of $|N(\Omega)|$ as follows.

$$\begin{aligned}
\mathbb{E}\Big[\,|N(\Omega)|\,\Big] &\geq \mathbb{E}\Big[\,|N(\mathcal{D})|\,\Big] \\
&= \sum_{jj' \in [m]\times[m]} \mathbb{P}\left[(j,j') \in N(\mathcal{D})\right] \\
&\geq \sum_{\substack{jj' \in [m]\times[m], \\ j \neq j'}} \mathbb{P}\left[(j,j') \in N(\mathcal{D})\right] \\
&= \sum_{\substack{jj' \in [m]\times[m], \\ j \neq j'}} \left(1 - \left(1 - \frac{\delta(\delta-1)}{m(m-1)}\right)^{|\mathcal{D}|}\right) \\
&= m(m-1)\left(1 - \left(1 - \frac{\delta(\delta-1)}{m(m-1)}\right)^{|\mathcal{D}|}\right) \\
&\geq m(m-1)\left(\frac{|\mathcal{D}|\,\delta(\delta-1)}{m(m-1)} - \frac{|\mathcal{D}|^2\,\delta^2(\delta-1)^2}{m^2(m-1)^2}\right) \\
&= |\mathcal{D}|\,\delta^2\left(1 - \left(\frac{1}{\delta} + \frac{(\delta-1)^2\,|\mathcal{D}|}{m(m-1)}\right)\right) \\
&= p\delta^2\,(1-\epsilon')\,.
\end{aligned}$$

Where in the last step, we set $\epsilon' = \frac{1}{\delta} + \frac{(\delta-1)^2|\mathcal{D}|}{m(m-1)}$.

To complete the proof, we must bound the probability that the random quantity $|N(\Omega)|$ cannot be much smaller than $p\delta^2(1-\epsilon')$. As a first step, we define the random variables $\chi_{jj'} := \mathbb{1}_{\{(j,j') \in N(\mathcal{D})\}}$ and notice that the following chain of inequalities hold

$$|N(\Omega)| \geq |N(\mathcal{D})| \geq \sum_{\substack{jj' \in [m]\times[m] \\ j \neq j'}} \chi_{jj'}.$$

Therefore, we have that

$$\mathbb{P}\left[|N(\Omega)| < p\delta^2(1-\epsilon'-\epsilon'')\right] \leq \mathbb{P}\left[\sum_{\substack{jj'\in[m]\times[m]\\ j\neq j'}} \chi_{jj'} < p\delta^2(1-\epsilon'-\epsilon'')\right].$$

Also, since by above, $\mathbb{E}\left[\sum_{j\neq j'}\chi_{jj'}\right] \geq p\delta^2(1-\epsilon')$, we have that

$$\mathbb{P}\left[|N(\Omega)| < p\delta^2(1-\epsilon)\right] \leq \mathbb{P}\left[\sum_{\substack{jj'\in[m]\times[m]\\ j\neq j'}} \chi_{jj'} < \mathbb{E}\left[\sum_{\substack{jj'\in[m]\times[m]\\ j\neq j'}} \chi_{jj'}\right] - p\delta^2\epsilon''\right].$$

Now, notice that the sum $\sum_{j\neq j'}\chi_{jj'}$ has $m(m-1)$ terms and each term in the sum is dependent on no more than $2m-4$ terms. Therefore, one way to bound the required quantity is to extract independent sub-sums from the above sum and bound the deviation of each of those from their means (which is the corresponding sub-sum of the mean). A principled way of doing this is suggested by the celebrated Hajnal-Szemerédi theorem [71, 90]. Consider a graph on the vertex set $[m]\times[m]\setminus\{(1,1),\ldots,(m,m)\}$ where there is an edge between vertices (j,j') and (j_1,j_1') if $j=j_1$ and/or $j'=j_1'$, i.e., exactly when the random variables $\chi_{jj'}$ and $\chi_{j_1j_1'}$ are dependent. Since this graph has degree $\Theta(m)$, Hajnal-Szemeridi theorem tells us that this graph can be equitable colored with $\Theta(m)$ colors. In other words, the above sum can be partitioned into $\Theta(m)$ sub-sums such that each sub-sum has $\Theta(m)$ elements and the random variables in each of them are independent. Along with this and the fact that $m(m-1) > p\delta^2$, we can use the union

bound and write

$$\mathbb{P}\left[\sum_{\substack{jj'\in[m]\times[m]\\ j\neq j'}}\chi_{jj'} < \mathbb{E}\left[\sum_{\substack{jj'\in[m]\times[m]\\ j\neq j'}}\chi_{jj'}\right] - p\delta^2\epsilon''\right]$$
$$\leq \Theta(m)\ \mathbb{P}\left[\sum_{jj'\in C_1}\chi_{jj'} < \mathbb{E}\left[\sum_{jj'\in C_1}\chi_{jj'}\right] - |C_1|\,\epsilon''\right],$$

where C_1 is one of the "colors". Notice that $|C_1| = \Theta(m)$.

Finally, using Chernoff bounds, we have

$$\Theta(m)\ \mathbb{P}\left[\frac{1}{|C_1|}\sum_{jj'\in C_1}\chi_{jj'} < \frac{1}{|C_1|}\mathbb{E}\left[\sum_{jj'\in C_1}\chi_{jj'}\right] - \epsilon''\right] \leq \Theta(m)\ \exp\left\{-2\epsilon''^2\Theta(m)\right\}.$$

Finally, since ϵ' can be made as small as possible, setting $\epsilon := \epsilon' + \epsilon''$ yeilds $p_1 < p^{-c_1}$ for some $c_1 > 0$. This technique of generating large deviation bounds when one has limited dependence is not new, see [108].

Part 2: Now, we bound $\mathbb{P}(\mathcal{E}_2^c \mid \mathcal{E}_1)$. Associated to a fixed i one can imagine δ independent random trials that determine the outgoing edges from i. In a similar way there are δ independent random trials associated to the outgoing edges of i'. Let us fix (i, i') and investigate the outgoing edge (in the tensor graph) determined by the first trial of i and the first trial of i'. The probability that this edge emanating from the vertex $(i, i') \in [p]^2 \setminus \{(1,1), \ldots, (p,p)\}$ hits an arbitrary vertex $(j, j') \in [m]^2$ is given by $1/m^2$. The probability that this edge lands in $N(\Omega)$ is, therefore, given by $|N(\Omega)|/m^2$. Since there are δ^2 edges that are incident on (i, i'), the expected size of overlap between $N(i, i')$ and $N(\Omega)$ is upper bounded by

$$\delta^2\frac{|N(\Omega)|}{m^2}.$$

Again, to show concentration, we employ similar arguments as before and define indicator random variables $\chi_1 \ldots, \chi_{\delta^2}$ each of which corresponds to one of the edges emanating from the vertex (i, i') and then observing that the sum $\sum_{k=1}^{\delta^2} \chi_k$ is precisely equal to the random quantity $|N(i, i') \cap N(\Omega)|$. To conclude that this random quantity concentrates, we first observe that, as above, the δ^2 dependent terms can be divided up into $\Theta(\delta)$ with $\Theta(\delta)$ elements each such that in each set the terms are independent. Therefore, we have

$$\begin{aligned}
\mathbb{P}\left[|N(i,i') \cap N(\Omega)| > \delta^2 \frac{|N(\Omega)|}{m^2}(1+\epsilon')\right] &= \mathbb{P}\left[\sum_{k=1}^{\delta^2} \chi_k > \delta^2 \frac{|N(\Omega)|}{m^2}(1+\epsilon')\right] \\
&\leq \Theta(\delta)\ \mathbb{P}\left[\sum_{k \in C_1} \chi_k > \Theta(\delta) \frac{|N(\Omega)|}{m^2}(1+\epsilon')\right] \\
&\leq \Theta(\delta) \exp\left\{-\delta \frac{|N(\Omega)|}{m^2}\epsilon'\right\},
\end{aligned}$$

where C_1 is one of the colors.

Therefore, since conditioned on $\mathcal{E}_1$, $|N(\Omega)| > \delta^2 p(1-\epsilon)$ if we pick $m = \delta\sqrt{dp}$, and $\delta = \Theta(\log p)$ there is a $c_2' = c_2'(\epsilon') > 2$ such that, $|N(i,i') \cap N(\Omega)| > \delta^2 \frac{|N(\Omega)|}{m^2}(1+\epsilon')$, with probability not exceeding $p^{-c_2'}$. Setting $\epsilon = (1+\epsilon')\,|N(\Omega)|\,/m^2$, picking m as prescribed, and taking union bound over $(i, i') \in \Omega^c$, we get $p_2 \leq p^{-c_2}$ for some $c_2 > 0$.

Part 3: Next, we bound $\mathbb{P}(\mathcal{E}_3^c \mid \mathcal{E}_1)$. Notice that the proof is very similar to that of part 2 when $i \neq i'$. So, here we will consider the quantity $|N(i,i) \cap N(\Omega \setminus \{(i,i)\})|$.

As explained above to each left node i we associate δ random trials that determine its outgoing edges. Correspondingly, if we fix a left node (i, i) in the tensor graph, and think of its outgoing edges they are determined by the outcome of δ^2 product trials. Let $\mathbb{1}_k(j)$ be the indicator function of the event that in the k^{th} trial of i the outgoing edge is incident

on j. The probability that the edge associated to the (k,l) trial associated to (i,i) is incident on (j,j') is the random variable $\mathbb{1}_k(j)\mathbb{1}_l(j')$. Note that $\mathbb{1}_k(j)\mathbb{1}_k(j') = \mathbb{1}_k(j)$ if $j = j'$ and 0 otherwise.

Note that

$$\begin{aligned}|N(i,i) \cap N(\Omega \setminus (i,i))| &= \sum_{k=1}^{\delta}\sum_{l=1}^{\delta}\sum_{(j,j')\in N(\Omega\setminus(i,i))} \mathbb{1}_k(j)\mathbb{1}_l(j') \\ &\leq \delta + \sum_{k\neq l}\sum_{(j,j')\in N(\Omega\setminus(i,i))} \mathbb{1}_k(j)\mathbb{1}_l(j')\end{aligned}$$

When $k \neq l$, the trials corresponding to $\mathbb{1}_k(j), \mathbb{1}_l(j')$ are independent, and hence $\mathbb{E}\mathbb{1}_k(j)\mathbb{1}_l(j') = \frac{1}{m^2}$. We define $\chi_{k,l} := \sum_{(j,j')\in N(\Omega\setminus(i,i))} \mathbb{1}_k(j)\mathbb{1}_l(j')$ and note that $\mathbb{E}(\chi_{k,l}) = \frac{N(\Omega\setminus(i,i))}{m^2}$. We also note that $\chi_{k,l}$ is binary valued, and

$$|N(i,i) \cap N(\Omega \setminus (i,i))| \leq \delta + \sum_{k\neq l}\chi_{k,l}.$$

Therefore,

$$\begin{aligned}\mathbb{E}(|N(i,i) \cap N(\Omega \setminus (i,i))|) &\leq \delta + (\delta^2 - \delta)\frac{N(\Omega \setminus (i,i))}{m^2} \\ &\leq \delta + (\delta^2 - \delta)\frac{\delta^2 dp}{m^2} \\ &\leq \delta^2\left(\frac{1}{\delta} + \left(1 - \frac{1}{\delta}\right)\frac{\delta^2 dp}{m^2}\right) \\ &\leq \delta^2\epsilon.\end{aligned}$$

Next we need to prove that the quantity of interest $\sum_{k\neq l}\chi_{k,l}$ concentrates about its mean. To that end we note that these binary valued variables are such that any particular $\chi_{k,l}$ is dependent on at most $2\delta - 2$ other variables. Using Chernoff concentration bounds

in conjunction with the Hajnal-Szemeredi based coloring argument explained in part 1 of this proof, followed by a union bound over $i \in [p]$ we obtain the required probability bounds $p_3 < p^{-c_3}$ for some $c_3 > 0$.

Substituting the bounds for p_1, p_2, p_3 back into (C.1) concludes the proof for the case when $A = B$. We will next outline the simple modifications to the above proof technique that need to be done to handle the case when $A \neq B$.

C.1.1 The case when $G_1 \neq G_2$

As stated earlier, the case where $G_1 \neq G_2$ (i.e., $A \neq B$) follows from a simple modification of the proof technique presented above. Part 1. Since A and B are independent, instead of equation (C.2), we have the following expression for $j \neq j' \in [m], i \in [p]$,

$$\mathbb{P}\left[(j, j') \in N((i, i))\right] = \frac{\delta^2}{m^2}. \tag{C.3}$$

Then, proceeding as before, we get the following expression as a lower bound for the expected size of $|N(\Omega)|$

$$\begin{aligned}
\mathbb{E}\left[|N(\Omega)|\right] &\geq \sum_{\substack{jj' \in [m]\times[m] \\ j \neq j'}} \mathbb{E}\left[\chi_{jj'}\right] \\
&= m(m-1)\left(1 - \left(1 - \frac{\delta^2}{m^2}\right)^{|\mathcal{D}|}\right) \\
&\geq p\delta^2\left(1 - \tilde{\epsilon}'\right),
\end{aligned}$$

where, as before, $\chi_{jj'} = \mathbb{1}\{(j,j') \in N(\mathcal{D})\}$ but $\tilde{\epsilon}'$ is given by $\frac{1}{m} + \frac{|\mathcal{D}|\delta^2}{m^2} - \frac{|\mathcal{D}|\delta^2}{m^3}$. The rest of the proof of Part 1 follows as above with $\tilde{\epsilon}'$ playing the role the definition of ϵ'.

Part 2. The proof of Part 2 follows exactly as before.

Part 3. The proof of Part 3 in the case when $A \neq B$ becomes simpler as well. As before, for both G_1 and G_2, to each left node i we associate δ random trials that determine its outgoing edges.

Correspondingly, if we fix a left node (i,i) in the tensor graph, and think of its outgoing edges, they are determined by the outcome of δ^2 product trials. Let $\mathbb{1}_k^1(j)$ (resp. $\mathbb{1}_k^2(j)$) be the indicator function of the event that in the k^{th} trial of i from graph G_1 (resp. graph G_2) the outgoing edge is incident on j. The probability that the edge associated to the (k,l) trial associated to (i,i) is incident on (j,j') is the random variable $\mathbb{1}_k^1(j)\mathbb{1}_l^2(j')$.

Therefore,

$$|N(i,i) \cap N(\Omega \setminus (i,i))| = \sum_{k=1}^{\delta}\sum_{l=1}^{\delta} \sum_{(j,j') \in N(\Omega\setminus(i,i))} \mathbb{1}_k^1(j)\mathbb{1}_l^2(j')$$

Since G_1 and G_2 are independent, the trials corresponding to $\mathbb{1}_k^1(j)$ and $\mathbb{1}_l^2(j')$ are now independent irrespective of the value of k and l. Hence $\mathbb{E}\mathbb{1}_k^1(j)\mathbb{1}_l^2(j') = \frac{1}{m^2}$. We define $\chi_{k,l} := \sum_{(j,j') \in N(\Omega\setminus(i,i))} \mathbb{1}_k^1(j)\mathbb{1}_l^2(j')$ and note that $\mathbb{E}(\chi_{k,l}) = \frac{N(\Omega\setminus(i,i))}{m^2}$. We therefore have

Therefore, apart from the extra δ term, the following chain of inequalities follows

much like Part 3 above

$$\begin{aligned}\mathbb{E}\left(|N(i,i)\cap N(\Omega\setminus(i,i))|\right) &= \sum_{k,l}\chi_{k,l}\\ &= \frac{N(\Omega\setminus(i,i))}{m^2}\\ &\leq \delta^2\frac{\delta^2 dp}{m^2}\\ &\leq \delta^2\epsilon\end{aligned}$$

Now, to conclude the proof, we need to show that the probability that the random sum $\sum_{k.l}\chi_{k.l}$ makes a large excursion from its mean is small. The proof of this concentration result follows exactly as above. □

C.2 Proof of Proposition 4.1

Proposition 4.1. *Consider a random matrix $X\in\mathbb{R}^{p\times p}$ such that $X_{ij}\overset{iid}{\sim}\mathrm{Ber}(\gamma)$ where $p\gamma=\Delta=\Theta(1)$, then for any $\epsilon>0$, X is $d-$distributed sparse with probability at least $1-\epsilon$, where*

$$d=\Delta\left(1+\frac{2\log(2p/\epsilon)}{\Delta}\right).$$

Proof. Let $X_i, i=1,\ldots,p$ denote the sparsity of the $i-$th column and let $X_i, i=p+1,\ldots,2p$, denote the sparsity of the $i-$th row. Notice that the $2p$ random variables $X_1,X_2,\ldots,X_{2p}$ are (dependent) $\mathrm{Bin}(p,\gamma)$ random variables. With the choice of d as

indicated in the theorem, we have the following,

$$\begin{aligned}\mathbb{P}(X_1 > d) &= \mathbb{P}\left(X_1 > \Delta\left(1 + \frac{\log(2p/\epsilon)}{\Delta}\right)\right) \\ &\overset{(a)}{\leq} \exp\left\{-\frac{\beta^2\Delta}{2+\beta}\right\}, \quad \beta = \frac{2\log(2p/\epsilon)}{\Delta} \\ &\overset{(b)}{\leq} \exp\left\{-\frac{\beta\Delta}{2}\right\} \\ &= \frac{\epsilon}{2p}\end{aligned}$$

where (a) follows from the multiplicative form of the Chernoff Bound [4] and (b) follows as long as $\beta > 2$. The rest of the proof follows from a simple application of the union bound. □

C.3 Proof of Theorem 4.2

Theorem 4.2. *Suppose that X is a $p \times p$ matrix. Furthermore, suppose that the hypotheses of Theorem 4.1 hold and let X^* be the solution to the optimization program (P_1). Then, there exists a $c > 0$ and an $\epsilon \in (0, 1/4)$ such that the following holds with probability exceeding $1 - p^{-c}$.*

$$\|X^* - X\|_1 \leq \frac{2 - 4\epsilon}{1 - 4\epsilon}\left(\min_{\Omega \in \mathfrak{W}_{d,p}} \|X - X_\Omega\|_1\right). \tag{C.4}$$

Proof. Since X^* is the optimum of the optimization program (P_1), we have that $\|X\|_1 \geq \|X^*\|_1$. Let Ω^* be such that $\|X - X_{\Omega^*}\|_1 = \min_{\Omega \in \mathfrak{W}_{d,p}} \|X - X_X\|$. We can proceed as

follows

$$\begin{aligned}
\|X\|_1 &\geq \|X^*\|_1 \\
&= \|(X + X^* - X)_\Omega\|_1 + \|(X + X^* - X)_{\Omega^c}\|_1 \\
&\geq \|X_\Omega\|_1 - \|(X^* - X)_\Omega\|_1 + \|(X^* - X)_{\Omega^c}\|_1 - \|X_{\Omega^c}\|_1 \\
&= \|X\|_1 - 2\|X_{\Omega^c}\|_1 + \|X^* - X\|_1 - 2\|(X - X^*)_\Omega\|_1 \\
&\geq \|X\|_1 - 2\|X_{\Omega^c}\|_1 + \left(1 - \frac{2\epsilon}{1 - 2\epsilon}\right)\|X^* - X\|_1
\end{aligned}$$

where in the last step, we have used the fact that since X^* is a feasible point in (P_1), $AX^*B^T = AXB^T$ and therefore, we can apply the result of Proposition 4.3 to $X^* - X$. This completes the proof. □

Appendix D

Proofs from Chapter 5

D.1 Proof of Theorem 5.1

Recall that for any pair of leaves $A, B \in L$, we define

$$\widehat{p}_{AB} = \frac{1}{mk} \sum_{i\in[m],j\in[k]} \mathbb{1}\{\chi_A^{ij} \neq \chi_B^{ij}\}. \tag{D.1}$$

Theorem 5.1. $\{\mathbb{E}[\widehat{p}_{AB}]\}_{A,B\in L}$[2] *forms an ultrametric with respect to the true species tree* S*. In fact, for any triple* $A, B, C \in L$ *with the topology* $((A, B), C)$ *in* S*, we have*

$$\mathbb{E}[\widehat{p}_{AC}] = \mathbb{E}[\widehat{p}_{BC}] > \mathbb{E}[\widehat{p}_{AB}] + \frac{3e^{-\frac{4}{3}\mu\tau_{AC}}\mu f}{8\mu + 3}. \tag{D.2}$$

Proof. Suppose that $A, B, C \in L$ are three arbitrary leaves of the species tree with the topology $((A, B), C)$. By definition, we have that

$$\mathbb{E}[\widehat{p}_{AC}] = \mathbb{E}\left[\frac{3}{4}\left(1 - e^{-\frac{4}{3}\delta_{AC}}\right)\right],$$

where δ_{AC} is the distance between A and C on a random gene tree drawn according to the multispecies coalescent. Notice that it satisfies $\delta_{AC} = \mu\tau_{AC} + 2\mu Z$ with $Z \sim \text{Exp}(1)$.

[2] *Unless otherwise noted, expectations will be with respect all the randomness present.*

Therefore, we have

$$\begin{aligned}\mathbb{E}\left[\widehat{p}_{AC}\right] - \mathbb{E}\left[\widehat{p}_{AB}\right] &= -\frac{3}{4}e^{-\frac{4}{3}\mu\tau_{AC}}\mathbb{E}\left[e^{-\frac{8}{3}\mu Z}\right] + -\frac{3}{4}e^{-\frac{4}{3}\mu\tau_{AB}}\mathbb{E}\left[e^{-\frac{8}{3}\mu Z}\right] \\ &\overset{(a)}{=} \frac{3\left(e^{-\frac{4}{3}\mu\tau_{AB}} - e^{-\frac{4}{3}\mu\tau_{AC}}\right)}{4(\frac{8}{3}\mu+1)} \\ &\overset{(b)}{\geq} \frac{3e^{-\frac{4}{3}\mu\tau_{AC}}\mu f}{(8\mu+3)},\end{aligned}$$

where (a) follows from the fact that if $X \sim \text{Exp}(1)$, for any $\alpha > 0$, $\mathbb{E}[e^{-\alpha X}] = (\alpha+1)^{-1}$ and (b) follows from observing that for any $\alpha > 0$ and $x < y$, we have

$$\frac{e^{-\alpha x}}{\alpha} - \frac{e^{-\alpha y}}{\alpha} = \int_x^y e^{-\alpha t}\, dt \geq (y-x)e^{-\alpha y}$$

Proceeding similarly, It can be seen that $\mathbb{E}\left[\widehat{p}_{AC}\right] = \mathbb{E}\left[\widehat{p}_{BC}\right]$. This concludes the proof. □

D.2 Proof of Theorem 5.2

We now prove Theorem 5.2 which guarantees that S can be reliably recovered by using a standard distance-based algorithm like UPGMA or bottom-up agglomerative clustering with $\{\widehat{p}_{AB}\}_{A,B\in L}$ as a dissimilarity measure for L.

Theorem 5.2. *Given an $\epsilon > 0$, using UPGMA on L with the dissimilarity measure $\{\widehat{p}_{AB}\}_{A,B\in L}$ results in the correct tree S being output with probability no less than $1-\epsilon$ as long as the number of genes m, and the sequence length k satisfy*

$$m \geq C_1(\mu, \Delta, n, \epsilon) \times f^{-2} \quad \textit{and} \quad k \geq 1, \tag{D.3}$$

where $C_1(\mu, \Delta, n, \epsilon) = \frac{16\, e^{\frac{8}{3}\mu\Delta}(8\mu+3)^2}{9\mu^2} \log\left(\frac{8\binom{n}{3}}{\epsilon}\right)$.

Proof. Recall that the algorithm we propose to recover the tree uses $\{\widehat{p}_{AB}\}_{A,B\in L}$ as a dissimilarity measure and uses an agglomerative clustering algorithm. Therefore, this procedure errs if for any triple of leaves A, B, C which have the topology $((A, B), C)$ with respect to S, either $\widehat{p}_{AB} > \widehat{p}_{AC}$ or $\widehat{p}_{AB} > \widehat{p}_{BC}$. Letting $\binom{L}{3}$ denote the set of all unordered triples in L, we can use the union bound and over-estimate the error as follows

$$\mathbb{P}\left[\text{Error}\right] \tag{D.4}$$

$$= \mathbb{P}\left[\bigcup_{((A,B),C)\in\binom{L}{3}} \left\{\text{The triple } ((A, B), C) \text{ is such that } \widehat{p}_{AB} > \widehat{p}_{AC} \text{ or } \widehat{p}_{AB} > \widehat{p}_{BC}\right\}\right]$$

$$\leq \sum_{((A,B),C)\in\binom{L}{3}} \mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{AC}\right] + \mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{BC}\right]. \tag{D.5}$$

We will now upper bound the term $\mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{AC}\right]$, the other term will satisfy the same upper bound. Defining $\alpha_{\text{um}} = \frac{3e^{-\frac{4}{3}\Delta}\mu f}{(8\mu+3)}$, for an arbitrary triple $((A, B), C)$ we have

$$\begin{aligned}
\mathbb{P}\left[\widehat{p}_{AB} - \widehat{p}_{AC} > 0\right] &= \mathbb{P}\left[\widehat{p}_{AB} - \mathbb{E}\left[p_{AB}\right] - \widehat{p}_{AC} + \mathbb{E}\left[p_{AC}\right] > \mathbb{E}\left[p_{AC}\right] - \mathbb{E}\left[p_{AB}\right]\right] \\
&\overset{(a)}{\leq} \mathbb{P}\left[\widehat{p}_{AB} - \mathbb{E}\left[p_{AB}\right] - \widehat{p}_{AC} + \mathbb{E}\left[p_{AC}\right] > \alpha_{\text{um}}\right] \\
&\leq \mathbb{P}\left[\widehat{p}_{AB} - \mathbb{E}\left[p_{AB}\right] > \frac{\alpha_{\text{um}}}{2}\right] + \mathbb{P}\left[\mathbb{E}\left[p_{AC}\right] - \widehat{p}_{AC} > \frac{\alpha_{\text{um}}}{2}\right],
\end{aligned} \tag{D.6}$$

where (a) follows from Theorem 5.1. Let us first look at the first term in (D.6). The

second one will follow similarly.

$$
\begin{aligned}
&\mathbb{P}\left[\widehat{p}_{AB} - \mathbb{E}[p_{AB}] > \alpha_{\text{um}}/2\right] \\
&\quad \overset{(a)}{=} \mathbb{E}\left[\mathbb{P}\left(\widehat{p}_{AB} - \mathbb{E}\left[p_{AB}\right] > \frac{\alpha_{\text{um}}}{2}\middle|\{\delta_{AB}^{(i)}\}_{i\in[m]}\right)\right] \\
&\quad \leq \mathbb{E}\left[\mathbb{P}\left(\widehat{p}_{AB} - \frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} > \frac{\alpha_{\text{um}}}{4}\middle|\{\delta_{AB}^{(i)}\}_{i\in[m]}\right)\right. \\
&\qquad\qquad \left. + \mathbb{P}\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} - \mathbb{E}\left[p_{AB}\right] > \frac{\alpha_{\text{um}}}{4}\middle|\{\delta_{AB}^{(i)}\}_{i\in[m]}\right)\right] \\
&\quad = \mathbb{E}\left[\mathbb{P}\left(\widehat{p}_{AB} - \frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} > \frac{\alpha_{\text{um}}}{4}\middle|\{\delta_{AB}^{(i)}\}_{i\in[m]}\right)\right] + \mathbb{P}\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} - \mathbb{E}\left[p_{AB}\right] > \frac{\alpha_{\text{um}}}{4}\right),
\end{aligned}
\tag{D.7}
$$

where in (a), $\delta_{AB}^{(i)}$ is the random distance between leaves A and B in gene tree $\mathcal{G}^{(i)}$. The two terms in the last equation can now be upper bounded using Hoeffding's inequality:

$$
\mathbb{E}\left[\mathbb{P}\left[\frac{1}{mk}\sum_{i=1}^{m}\sum_{j=1}^{k} X_{AB}^{ij} - \frac{1}{m}\sum_{i=1}^{m} p_{AB}^{(i)} > \frac{\alpha_{\text{um}}}{4}\middle|\left\{d_{AB}^{(i)}\right\}\right]\right] \leq e^{-mk\alpha_{\text{um}}^2/16}, \tag{D.8}
$$

$$
\mathbb{P}\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} - \mathbb{E}\left[p_{AB}\right] > \frac{\alpha_{\text{um}}}{4}\right) \leq e^{-m\alpha_{\text{um}}^2/16}, \tag{D.9}
$$

where by X_{AB}^{ij}, we mean the random variable $\mathbb{1}\left\{\chi_A^{ij} \neq \chi_B^{ij}\right\}$. Substituting these in (D.5),

we have

$$\begin{aligned}\mathbb{P}\left[\text{Error}\right] &\leq \sum_{((AB)C)\in\binom{L}{3}} \mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{AC}\right] + \mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{BC}\right] \\ &\leq \sum_{((AB)C)\in\binom{L}{3}} 4\left(e^{-mk\alpha_{\text{um}}^2/16} + e^{-m\alpha_{\text{um}}^2/16}\right) \\ &\leq \binom{n}{3} 4\left(e^{-mk\alpha_{\text{um}}^2/16} + e^{-m\alpha_{\text{um}}^2/16}\right)\end{aligned}$$

Therefore, the probability of error can be made less than ϵ if we pick m and k as shown in (5.6) or (D.3).

Sketch of an alternate proof. Now, we give a brief proof sketch for an alternative and direct technique to prove the result of Theorems 5.2 (upto constants). This idea would apply to the proof of Theorem 5.2.

Instead of (D.7), we may proceed as follows:

$$\begin{aligned}\mathbb{P}\left[\widehat{p}_{AB} - \mathbb{E}[p_{AB}] > \alpha_{\text{um}}/2\right] &\overset{(a)}{=} \mathbb{P}\left[\frac{1}{m}\sum_{i\in[m]}\left(\frac{1}{k}\sum_{j\in[k]} X_{AB}^{ij}\right) - \mathbb{E}[p_{AB}] > \alpha_{\text{um}}/2\right] \\ &\overset{(b)}{\leq} \exp\left\{-\frac{m\alpha_{\text{um}}^2}{4\left(\sigma^2 + \frac{\alpha_{\text{um}}}{3}\right)}\right\}, \end{aligned} \tag{D.10}$$

where in (a), as before, X_{AB}^{ij} is the random variable $\mathbb{1}\left\{\chi_A^{ij} \neq \chi_B^{ij}\right\}$. In (b), we use Bernstein's inequality with

$$\sigma^2 \triangleq \text{Var}\left(\frac{1}{k}\sum_{j=1}^{k} X_{AB}^{ij}\right).$$

Therefore, as before, the probability of error is upper-bounded (say, by substituting into

(D.5)) as follows:

$$\begin{aligned}
\mathbb{P}\left[\text{Error}\right] &\leq \sum_{((AB)C)\in\binom{L}{3}} \mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{AC}\right] + \mathbb{P}\left[\widehat{p}_{AB} > \widehat{p}_{BC}\right] \\
&\leq \sum_{((AB)C)\in\binom{L}{3}} 4\exp\left\{-\frac{m\alpha_{\text{um}}^2}{4\left(\sigma^2 + \frac{\alpha_{\text{um}}}{3}\right)}\right\} \\
&\leq \binom{n}{3} 4\exp\left\{-\frac{m\alpha_{\text{um}}^2}{4\left(\sigma^2 + \frac{\alpha_{\text{um}}}{3}\right)}\right\}.
\end{aligned}$$

In other words, as long as $m \geq 4\left(\frac{\sigma^2}{\alpha_{\text{um}}^2} + \frac{1}{3\alpha_{\text{um}}}\right)\log\left(\frac{4\binom{n}{3}}{\epsilon}\right)$, the probability of error can be made smaller than ϵ. In order to complete the proof, all we need to do is upper bound σ^2. For this, we use the conditional variance formula. That is, for any $i \in [m]$,

$$\begin{aligned}
\sigma^2 &\triangleq \text{Var}\left(\frac{1}{k}\sum_{j=1}^{k} X_{AB}^{ij}\right) \\
&\overset{(a)}{=} \mathbb{E}\left[\text{Var}\left(\frac{1}{k}\sum_{j=1}^{k} X_{AB}^{ij}\middle| p_{AB}^{(i)}\right)\right] + \text{Var}\left(\mathbb{E}\left[\frac{1}{k}\sum_{j=1}^{k} X_{AB}^{ij}\middle| p_{AB}^{(i)}\right]\right) \\
&\overset{(b)}{=} \mathbb{E}\left[\frac{1}{k}\, p_{AB}^{(i)}(1 - p_{AB}^{(i)})\right] + \text{Var}\left(p_{AB}^{(i)}\right) \\
&\overset{(c)}{\leq} \frac{1}{4k} + \text{Var}\left[\frac{3}{4}\left(1 - e^{-\frac{4}{3}\delta_{AB}^{(i)}}\right)\right] \\
&\overset{(d)}{=} \frac{1}{4k} + \frac{9}{16}\left(1 + e^{-\frac{8}{3}\mu\tau_{AB}}\text{Var}\left[e^{-\frac{8}{3}\mu Z}\right]\right) \\
&\leq \frac{1}{4k} + \frac{9}{16}\left(1 + e^{-\frac{8}{3}\mu\tau_{AB}}\mathbb{E}\left[e^{-\frac{16}{3}\mu Z}\right]\right) \\
&= \frac{1}{4k} + \frac{9}{16}\left(1 + \frac{e^{-\frac{8}{3}\mu\tau_{AB}}}{\frac{16}{3}\mu + 1}\right),
\end{aligned}$$

where in (a), we have used the conditional variance formula, conditioning on $p_{AB}^{(i)} \triangleq \frac{3}{4}\left(1 - \delta_{AB}^{(i)}\right)$ with $\delta_{AB}^{(i)}$ denoting the random (gene tree) distance between A and B in $\mathcal{G}^{(i)}$.

(b) follows since conditioned on $p_{AB}^{(i)}$, $\frac{1}{k}\sum_{j=1}^{k} X_{AB}^{ij} \sim \text{Bin}(k, p_{AB}^{(i)})$ and $\mathbb{E}\left[X_{AB}^{ij}\middle|p_{AB}^{(i)}\right] = p_{AB}^{(i)}$. In (c), we have simply used the inequality of arithmetic and geometric means [1]. In (d), as before, we use the fact that $\delta_{AB} = \mu\tau_{AB} + 2\mu Z$ with $Z \sim \text{Exp}(1)$. This, upper bound on σ^2 can be used above to get the desired result (upto constants). □

D.3 Proof of Theorem 5.3

In what follows, for two random variables X and Y, we will use the notation $X \overset{d}{=} Y$ to mean that X and Y have the same distribution. Recall that we define $d_{AB} = -\frac{3}{4}\log\left(1 - \frac{4}{3}\mathbb{E}\left[\widehat{p}_{AB}\right]\right)$ and Theorem 5.3, which we will prove now, tells us that these distances form an additive metric with respect to S.

Theorem 5.3. *The set of dissimilarities $\{d_{AB}\}_{A,B\in L}$ forms an additive metric with respect to S. In fact, suppose the leaves $A, B, C, D \in L$ are such that either $((A,B),(C,D))$ or $(((A,B),C),D)$ holds with respect to S, then*

$$d_{AC} + d_{BD} = d_{AD} + d_{BC} > d_{AB} + d_{CD} + \alpha_{\text{add}},$$

where $\alpha_{\text{add}} = \frac{3}{4}\log\left(\frac{8}{3}\mu_L(1 - e^{-f}) + 1\right) > 0$.

Proof. We will first show that for any 4 leaves $A, B, C, D \in L$ that are such that either $((A,B),(C,D))$ or $(((A,B),C,D))$ holds with respect to S, then $d_{AC} + d_{BD} > d_{AB} + d_{CD} + \alpha_{\text{add}}$. Using similar techniques, we will next establish that $d_{AC} + d_{BD} = d_{AB} + d_{CD}$.

[1] If $a_1 \ldots a_n$ are positive numbers, then $\frac{1}{n}\sum_{i\in[n]} a_i \geq (a_1 a_2 \cdots a_n)^{1/n}$

We begin by observing that by definition,

$$d_{AC} + d_{BD} - d_{AB} - d_{CD} = -\frac{3}{4}\log\left(1 - \frac{4}{3}\mathbb{E}\left[\widehat{p}_{AC}\right]\right) - \frac{3}{4}\log\left(1 - \frac{4}{3}\mathbb{E}\left[\widehat{p}_{BD}\right]\right) + \frac{3}{4}\log\left(1 - \frac{4}{3}\mathbb{E}\left[\widehat{p}_{AB}\right]\right) + \frac{3}{4}\log\left(1 - \frac{4}{3}\mathbb{E}\left[\widehat{p}_{AB}\right]\right) \quad \text{(D.11)}$$

$$= \frac{3}{4}\log\left(\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]}\right), \quad \text{(D.12)}$$

where the expectations in the last equation are with respect to the multispecies coalescent and the δ's are the random gene tree distances as defined in Section 5.2.3.

We will prove this theorem by lower bounding the quantity $\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]}$ appropriately. Towards this end, we note that for any 4 leaves of the species tree A, B, C, D, there are only 2 possible topologies with respect to S upto relabeling: (a) $((A, B), (C, D))$ and (b) $(((A, B), C), D)$. We will consider each case separately and bound the above quantity in what follows.

Case (a): $((A, B), (C, D))$ In order to tackle the first case, we will use the notation from Figure D.1a below, which shows the species tree S restricted to the leaves A, B, C, D. Let o_1, o_2 and o_3 be the common ancestors of (A, B), (C, D) and (A, C) respectively. Let $\mathcal{E}_{AB}$ be the event that the lineages corresponding to A and B coalesce in the segment (o_1, o_3) of the tree in Figure D.1a and let $\overline{\mathcal{E}_{AB}}$ be the event that this does not occur. Similarly, we define the events $\mathcal{E}_{CD}$ and $\overline{\mathcal{E}_{CD}}$. To reduce notational clutter, for $w, v \in S$, we will write μ_{wv} to denote $\sum_{e \in \pi^S_{wv}} \mu_e \tau_e$. Now, for leaves $X, Y \in L$, let Z_{XY} denote the random quantity $\frac{1}{2}(\delta_{XY} - \mu_{XY})$, i.e., it is the effective (mutation rate adjusted) coalescent time after the lineages corresponding to X and Y find themselves in a common population.

By the memoryless property of the exponential distribution, it is easy to check that $Z_{AB} - \mu_{o_1 o_3}$ conditioned on $\overline{\mathcal{E}_{AB}}$ has the same distribution as $Z_{CD} - \mu_{o_2 o_3}$ conditioned on $\overline{\mathcal{E}_{CD}}$. Let Z denote be a random variable with this common distribution. Also observe that Z_{AC} and Z_{BD} have the same distribution as Z. This is depicted diagrammatically in Figure D.1a.

Now, using the fact that by definition, $\delta_{AB} = \mu_{AB} + 2Z_{AB}$, we have

$$\begin{aligned}
\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right] &= e^{-\frac{4}{3}\mu_{AB}}\mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\right] \\
&= e^{-\frac{4}{3}\mu_{AB}}\left\{\mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\middle|\mathcal{E}_{AB}\right]\mathbb{P}\left(\mathcal{E}_{AB}\right) + \mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\middle|\overline{\mathcal{E}_{AB}}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\} \\
&\overset{(a)}{\geq} e^{-\frac{4}{3}\mu_{AB}}\left\{e^{-\frac{8}{3}\mu_{o_1 o_3}}\mathbb{P}\left(\mathcal{E}_{AB}\right) + e^{-\frac{8}{3}\mu_{o_1 o_3}}\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\} \\
&= e^{-\frac{4}{3}\left(\mu_{AB}+2\mu_{o_1 o_3}\right)}\left\{\mathbb{P}\left(\mathcal{E}_{AB}\right) + \mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\}, \qquad \text{(D.13)}
\end{aligned}$$

where (a) follows from the fact that conditioned on $\mathcal{E}_{AB}$, $Z_{AB} \leq \mu_{o_1 o_3}$ and that conditioned on $\overline{\mathcal{E}_{AB}}$, $Z_{AB} \overset{d}{=} Z + \mu_{o_1 o_3}$. Similarly, we get the following lower bound corresponding to the leaves C, D.

$$\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right] \geq e^{-\frac{4}{3}\left(\mu_{CD}+2\mu_{o_2 o_3}\right)}\left\{\mathbb{P}\left(\mathcal{E}_{CD}\right) + \mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{CD}}\right)\right\} \qquad \text{(D.14)}$$

On the other hand, notice that $\delta_{AC} = \mu_{AC} + 2Z_{AC} \overset{d}{=} \mu_{AC} + 2Z$ and $\delta_{BD} = \mu_{BD} + 2Z_{BD} \overset{d}{=} \mu_{BD} + 2Z$. Therefore, we have

$$\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right] = e^{-\frac{4}{3}\mu_{AC}}\mathbb{E}\left[e^{-\frac{8}{3}Z}\right], \qquad \text{and} \qquad \mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right] = e^{-\frac{4}{3}\mu_{BD}}\mathbb{E}\left[e^{-\frac{8}{3}Z}\right], \qquad \text{(D.15)}$$

Âě From equations (D.13) - (D.15), we have

$$\begin{aligned}
&\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]} \times \frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]} \\
&\qquad \geq \frac{e^{-\frac{4}{3}\left(\mu_{AB}+2\mu_{o_1o_3}\right)}\left\{\mathbb{P}\left(\mathcal{E}_{AB}\right)+\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\}}{e^{-\frac{4}{3}\mu_{AC}}\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]} \\
&\qquad\qquad \times \frac{e^{-\frac{4}{3}\left(\mu_{CD}+2\mu_{o_2o_3}\right)}\left\{\mathbb{P}\left(\mathcal{E}_{CD}\right)+\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\}}{e^{-\frac{4}{3}\mu_{BD}}\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]} \qquad \text{(D.16)} \\
&\qquad \overset{(a)}{=} \frac{\left\{\mathbb{P}\left(\mathcal{E}_{AB}\right)+\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\}\left\{\mathbb{P}\left(\mathcal{E}_{CD}\right)+\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\mathbb{P}\left(\overline{\mathcal{E}_{CD}}\right)\right\}}{\left(\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]\right)^2} \\
&\qquad = \left[\frac{\mathbb{P}\left(\mathcal{E}_{AB}\right)}{\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]}+\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right] \times \left[\frac{\mathbb{P}\left(\mathcal{E}_{CD}\right)}{\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]}+\mathbb{P}\left(\overline{\mathcal{E}_{CD}}\right)\right] \qquad \text{(D.17)}
\end{aligned}$$

where in (a), we have used the fact that $\mu_{AB}+\mu_{CD}+2\mu_{o_1o_3}+2\mu_{o_2o_3}=\mu_{AC}+\mu_{BD}$ and in the last step we divide each term in the numerator by $\mathbb{E}\left[e^{-\frac{8}{3}Z}\right]$.

Next, observe that Z stochastically dominates the random variable $\mu_L\tilde{Z}$, where $\tilde{Z}\sim$ Exp(1). Therefore, we have

$$\mathbb{E}\left[e^{-\frac{8}{3}Z}\right] \leq \mathbb{E}\left[e^{-\frac{8}{3}\mu_L\tilde{Z}}\right] = \frac{1}{\frac{8}{3}\mu_L+1}. \qquad \text{(D.18)}$$

Substituting this in (D.17) gives us

$$\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]} \geq \left[\frac{8}{3}\mu_L\mathbb{P}\left(\mathcal{E}_{AB}\right)+1\right] \times \left[\frac{8}{3}\mu_L\mathbb{P}\left(\mathcal{E}_{CD}\right)+1\right] \qquad \text{(D.19)}$$

Finally, we observe that the probability that the event $\mathcal{E}_{AB}$ occurs is given by $1-e^{-\tau_{o_1o_3}}$, where $\tau_{o_1o_3}$ is the length of the path (o_1,o_3) in the species tree; this follows from the memoryless property of the exponential distribution. Since $\tau_{o_1o_3}\geq f$, we have that

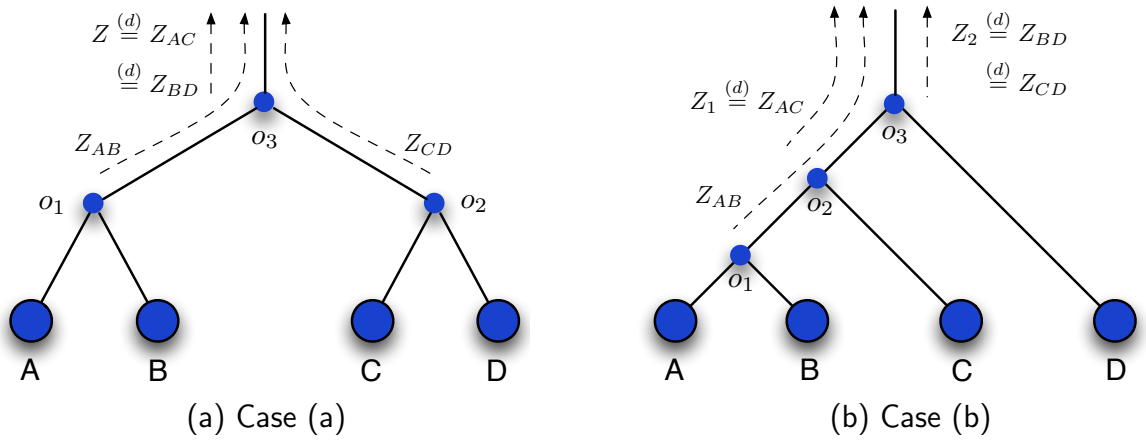


Figure D.1: Pictures showing the random variables and internal nodes used in Proof of Theorem 5.3

$\mathbb{P}(\mathcal{E}_{AB}) \geq 1 - e^{-f}$, and similarly $\mathbb{P}(\mathcal{E}_{CD}) \geq 1 - e^{-f}$. Substituting this in (D.19), we get the following lower bound

$$\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]} \geq \left[\frac{8}{3}\mu_L\left(1 - e^{-f}\right) + 1\right]^2 \tag{D.20}$$

Next, we consider Case (b).

Case (b) : $(((A, B), C), D)$ Here, we will write o_1, o_2, o_3 to denote the most recent common ancestors of (A, B), (A, C) and (A, D) respectively. Again we will use notation from the previous case for random variables of the form $Z_{XY}, X, Y \in L$. In this case, we let $\mathcal{E}_{AB}$ denote the event that the lineages corresponding to A and B coalesce in the branch (o_1, o_2) in Figure D.1b. Again, from the memoryless property, it can be seen that the random variable $Z_{AB} - \mu_{o_1 o_2}$ conditioned on $\overline{\mathcal{E}_{AB}}$ and the random variable Z_{AC} have the same distribution; we let Z_1 denote a random variable with this common distribution. Similarly Z_{CD} and Z_{BD} have the same distribution and we let Z_2 denote a random

variable with this distribution.

Reasoning as before, we see that since $\delta_{AB} = \mu_{AB} + 2Z_{AB}$,

$$\begin{aligned}\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right] &= e^{-\frac{4}{3}\mu_{AB}}\left\{\mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\middle|\mathcal{E}_{AB}\right]\mathbb{P}\left(\mathcal{E}_{AB}\right) + \mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\middle|\overline{\mathcal{E}_{AB}}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\} \\ &\overset{(a)}{\geq} e^{-\frac{4}{3}\mu_{AB}}\left\{e^{-\frac{8}{3}\mu_{o_1o_2}}\mathbb{P}\left(\mathcal{E}_{AB}\right) + e^{-\frac{8}{3}\mu_{o_1o_2}}\mathbb{E}\left[e^{-\frac{8}{3}Z_1}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\} \\ &= e^{-\frac{4}{3}\left(\mu_{AB}+2\mu_{o_1o_2}\right)}\left\{\mathbb{P}\left(\mathcal{E}_{AB}\right) + \mathbb{E}\left[e^{-\frac{8}{3}Z_1}\right]\mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right\}, \end{aligned} \tag{D.21}$$

where, as before, (a) follows from the fact that conditioned on $\mathcal{E}_{AB}$, $Z_{AB} \leq \mu_{o_1o_2}$ and that conditioned on $\overline{\mathcal{E}_{AB}}$, $Z_{AB} \overset{d}{=} Z_1 + \mu_{o_1o_2}$. On the other hand, we have

$$\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right] = e^{-\frac{4}{3}\mu_{CD}}\mathbb{E}\left[e^{-\frac{8}{3}Z_2}\right] \tag{D.22}$$

$$\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right] = e^{-\frac{4}{3}\mu_{AC}}\mathbb{E}\left[e^{-\frac{8}{3}Z_1}\right] \tag{D.23}$$

$$\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right] = e^{-\frac{4}{3}\mu_{BD}}\mathbb{E}\left[e^{-\frac{8}{3}Z_2}\right]. \tag{D.24}$$

Âě Therefore, from (D.21)-(D.24), we have that

$$\begin{aligned}\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]} &\geq e^{-\frac{4}{3}\left(\mu_{AB}+\mu_{CD}+2\mu_{o_1o_2}-\mu_{AC}-\mu_{BD}\right)}\left(\frac{1}{\mathbb{E}\left[e^{-\frac{8}{3}Z_1}\right]}\mathbb{P}\left(\mathcal{E}_{AB}\right) + \mathbb{P}\left(\overline{\mathcal{E}_{AB}}\right)\right) \\ &= \frac{\mathbb{P}\left[\mathcal{E}_{AB}\right]}{\mathbb{E}\left[e^{-\frac{8}{3}Z_1}\right]} + \mathbb{P}\left[\overline{\mathcal{E}_{AB}}\right] \end{aligned} \tag{D.25}$$

where the second step follows from the fact that $\mu_{AB} + \mu_{CD} + 2\mu_{o_1o_2} = \mu_{AC} + \mu_{BD}$. Finally, as in case (a), we use the bounds $\mathbb{E}\left[e^{-\frac{8}{3}Z_1}\right] \leq \frac{1}{\frac{8}{3}\mu_L+1}$ and that $\mathbb{P}\left[\mathcal{E}_{AB}\right] \geq 1 - e^{-f}$

to get the following lower bound.

$$\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]} \geq \frac{8}{3}\mu_L(1-e^{-f})+1 \tag{D.26}$$

Âě

Since $\left(\frac{8}{3}\mu_L(1-e^{-f})+1\right) \geq 1$, from (D.20) and (D.26), we have that for any 4 leaves A, B, C, D such that the species tree S restricted to these four leaves satisfies either $((A,B),(C,D))$ or $(((A,B),C),D)$, then

$$\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]} \geq \frac{8}{3}\mu_L(1-e^{-f})+1 \tag{D.27}$$

Substituting this lower bound in (D.12), we get the result that for any 4 leaves $A, B, C, D \in L$ that are such that $((A,B),(C,D))$ or $(((A,B),C,D))$ holds with respect to S, we have that $d_{AC}+d_{BD} > d_{AB}+d_{CD}+\alpha_{\text{add}}$, where $\alpha_{\text{add}} = \frac{3}{4}\log\left(\frac{8}{3}\mu_L(1-e^{-f})+1\right)$.

To conclude the proof, we will next establish the "equality part" of the theorem. As in (D.12), notice that the following holds.

$$\begin{aligned} d_{AC}+d_{BD}-d_{AB}-d_{CD} &= -\frac{3}{4}\log\left(1-\frac{4}{3}\mathbb{E}\left[\widehat{p}_{AC}\right]\right)-\frac{3}{4}\log\left(1-\frac{4}{3}\mathbb{E}\left[\widehat{p}_{BD}\right]\right) \\ &\quad +\frac{3}{4}\log\left(1-\frac{4}{3}\mathbb{E}\left[\widehat{p}_{AB}\right]\right)+\frac{3}{4}\log\left(1-\frac{4}{3}\mathbb{E}\left[\widehat{p}_{AB}\right]\right) \qquad \text{(D.28)} \\ &= \frac{3}{4}\log\left(\frac{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AB}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{CD}}\right]}{\mathbb{E}\left[e^{-\frac{4}{3}\delta_{AC}}\right]\mathbb{E}\left[e^{-\frac{4}{3}\delta_{BD}}\right]}\right), \qquad \text{(D.29)} \end{aligned}$$

Again, we will divide this proof into two cases as above.

Case (a): $((A,B),(C,D))$ Observe that the following hold with μ_{XY} and Z as defined before (cf. Fig D.1a).

$$\delta_{AD} = \mu_{AD} + 2Z,$$

$$\delta_{BC} = \mu_{BC} + 2Z,$$

$$\delta_{AC} = \mu_{AC} + 2Z,$$

$$\delta_{BD} = \mu_{BD} + 2Z$$

Substituting these in (D.29) and observing that $\mu_{AD} + \mu_{BC} = \mu_{AC} + \mu_{BD}$ tells us that $d_{AC} + d_{BD} = d_{AD} + d_{BC}$ in case (a).

Case (b): $(((A,B),C),D)$ In this case, observe that the following hold again with μ_{XY} and Z_1 and Z_2 as defined earlier (cf. Fig 2(b)):

$$\delta_{AD} = \mu_{AD} + 2Z_2,$$

$$\delta_{BC} = \mu_{BC} + 2Z_1$$

$$\delta_{AC} = \mu_{AC} + 2Z_1,$$

$$\delta_{BD} = \mu_{BD} + 2Z_2 d$$

Again, substituting these in (D.29) and observing that $\mu_{AD} + \mu_{BC} = \mu_{AC} + \mu_{BD}$ tells us that $d_{AC} + d_{BD} = d_{AD} + d_{BC}$ in case (b) as well. This concludes the proof.

□

D.4 Proof of Theorem 5.4

We will now prove the last main result in our chapter that shows that Theorem 5.3 can be used to design a tree reconstruction algorithm when one only has access to molecular data and also provides sample complexity results for this algorithm. Recall that we propose the following measure of dissimilarity from the samples

$$\widehat{d}_{AB} \triangleq -\frac{3}{4}\log\left(1-\frac{4}{3}\widehat{p}_{AB}\right). \tag{D.30}$$

where $\widehat{p}_{AB}$ is as defined in (5.4).

In light of Theorem 5.3, we proposed the following tree reconstruction procedure, which we call METAL: use any distance algorithm (like Neighbor Joining [120]) which returns an additive tree using $\{\widehat{d}_{AB}\}_{A,B\in L}$ as the dissimilarity measure. We then have the following result.

Theorem 5.4. *For any $\epsilon > 0$, the METAL algorithm succeeds in reconstructing (the unrooted version of) S with probability at least $1-\epsilon$ as long as m and k satisfy*

$$k \geq 1 \text{ and } m \geq \frac{e^{\frac{8\mu_U\Delta}{3}}(8\mu_U+3)^2(24+8\alpha_{\text{add}})^2}{162\alpha_{\text{add}}^2}\log\left(\frac{16\binom{n}{4}}{\epsilon}\right) \tag{D.31}$$

where $\alpha_{\text{add}} = \frac{3}{4}\log\left(\frac{8}{3}\mu_L(1-e^{-f})+1\right)$.

In the limit as $f \to 0$, the right side above approaches

$$C_2(\mu_U,\mu_L,\Delta,n,\epsilon) \times f^{-2}, \text{ where } C_2(\mu_U,\mu_L,\Delta,n,\epsilon) = \frac{8e^{\frac{8\mu_U\Delta}{3}}(8\mu_U+3)^2}{9\mu_L^2}\log\left(\frac{16\binom{n}{3}}{\epsilon}\right).$$

Proof. Notice that the above algorithm makes an error only if there exists a set of four

leaves A, B, C, D such that $\tau_{AB}+\tau_{CD} \leq \tau_{AC}+\tau_{BD} = \tau_{AD}+\tau_{BC}$, but the 4-point condition is not satisfied by $\widehat{d}$, that is:

$$\widehat{d}_{AB} + \widehat{d}_{CD} - \widehat{d}_{AC} - \widehat{d}_{BD} > 0 \quad \text{or} \quad \widehat{d}_{AB} + \widehat{d}_{CD} - \widehat{d}_{AD} - \widehat{d}_{BC} > 0$$

Therefore, using the union bound, the probability of error can be upper bounded as follows:

$$\begin{aligned}\mathbb{P}\left(\text{Error}\right) \leq \sum_{\substack{A,B,C,D\in L:\\ \tau_{AB}+\tau_{CD}\leq\tau_{AC}+\tau_{BD}=\tau_{AD}+\tau_{BC}}} & \mathbb{P}\left[\hat{d}_{AB} + \hat{d}_{CD} - \hat{d}_{AC} - \hat{d}_{BD} > 0\right] \\ & + \mathbb{P}\left[\hat{d}_{AB} + \hat{d}_{CD} - \hat{d}_{AD} - \hat{d}_{BC} > 0\right] \quad \text{(D.32)}\end{aligned}$$

We will bound the first term inside the summation of (D.32) and the second one will follow similarly. Setting $\alpha_{\text{add}} \triangleq \frac{3}{4}\log\left(\frac{8}{3}\mu_L(1-e^{-f})+1\right)$, observe that for a quadruple of leaves A, B, C, D such that $\tau_{AB} + \tau_{CD} \leq \tau_{AC} + \tau_{BD} = \tau_{AD} + \tau_{BC}$, we have

$$\begin{aligned}&\mathbb{P}\left[\widehat{d}_{AB} + \widehat{d}_{CD} - \widehat{d}_{AC} - \widehat{d}_{BD} > 0\right] \\ &\qquad = \mathbb{P}\Bigg[\widehat{d}_{AB} - d_{AB} + \widehat{d}_{CD} - d_{CD} - \widehat{d}_{AC} + d_{AC} - \widehat{d}_{BD} + d_{BD} \\ &\qquad\qquad > d_{AC} + d_{BD} - d_{AB} - d_{CD}\Bigg] \\ &\qquad \leq \mathbb{P}\left[\widehat{d}_{AB} - d_{AB} + \widehat{d}_{CD} - d_{CD} - \widehat{d}_{AC} + d_{AC} - \widehat{d}_{BD} + d_{BD} > \alpha_{\text{add}}\right],\end{aligned}$$

where the second inequality follows from Theorem 5.3 which says that $d_{AC} + d_{BD} -$

$d_{AB} - d_{CD} > \alpha_{\text{add}}$. We will again use the union bound to get

$$\mathbb{P}\left[\widehat{d}_{AB} + \widehat{d}_{CD} - \widehat{d}_{AC} - \widehat{d}_{BD} > 0\right] \tag{D.33}$$

$$\leq \mathbb{P}\left[\widehat{d}_{AB} - d_{AB} + \widehat{d}_{CD} - d_{CD} - \widehat{d}_{AC} + d_{AC} - \widehat{d}_{BD} + d_{BD} > \alpha_{\text{add}}\right]$$

$$\leq \mathbb{P}\left[\widehat{d}_{AB} - d_{AB} > \frac{\alpha_{\text{add}}}{4}\right] + \mathbb{P}\left[\widehat{d}_{CD} - d_{CD} > \frac{\alpha_{\text{add}}}{4}\right]$$

$$+\mathbb{P}\left[d_{AC} - \widehat{d}_{AC} > \frac{\alpha_{\text{add}}}{4}\right] + \mathbb{P}\left[d_{BD} - \widehat{d}_{BD} > \frac{\alpha_{\text{add}}}{4}\right]. \tag{D.34}$$

To proceed, we will focus our attention on the first term in (D.34). The remaining terms will follow similarly. For notational clarity, let us define the function $\ell(x) \triangleq -\frac{3}{4}\log\left(1 - \frac{4}{3}x\right)$ and let $p_{AB}^{(i)}$ (resp. p_{AB}) denote the random quantity $\ell^{-1}\left(\delta_{AB}^{(i)}\right)$ (resp. $\ell^{-1}\left(\delta_{AB}\right)$), where, as usual, $\delta_{AB}^{(i)}$ and δ_{AB} are the distances between A and B on random gene trees drawn according to the MSC. Now, observe that, by definition, $\widehat{d}_{AB}$ and d_{AB} are equal to $\ell(\widehat{p}_{AB})$ and $\ell(\mathbb{E}\left[p_{AB}\right])$ respectively.

Our strategy will be to first show that with high probability $\widehat{p}_{AB}$ is close to $\frac{1}{m}\sum_{i=1}^{m} p_{AB}^{(i)}$ which is in turn close to $\mathbb{E}\left[p_{AB}\right]$. We will then use the fact that $\ell(x)$ is a well-behaved function to obtain an upper bound on the the first term of (D.34).

Conditioned on a particular realization of the MSC process $\left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}$, let $\mathcal{E}_1(\xi)$ and $\mathcal{E}_2(\xi)$ denote the events that $\left|\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} - \mathbb{E}p_{AB}\right| > \xi$ and $\left|\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} - \widehat{p}_{AB}\right| > \xi$,

respectively. Now, notice we can bound the first term in (D.34) as follows.

$$\begin{aligned}
\mathbb{P}\left[\widehat{d}_{AB} - d_{AB} > \frac{\alpha_{\text{add}}}{4}\right] &= \mathbb{P}\left[\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}p_{AB}) > \frac{\alpha_{\text{add}}}{4}\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\mathbb{P}\left[\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}p_{AB}) > \frac{\alpha_{\text{add}}}{4}\middle|\left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}\right]\right] \\
&\overset{(b)}{\leq} \mathbb{E}\left[\mathbb{P}\left(\mathcal{E}_1(\xi)\middle|\left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}\right)\right] + \mathbb{E}\left[\mathbb{P}\left(\mathcal{E}_2(\xi)\middle|\left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}\right)\right] \\
&\quad + \mathbb{E}\left[\mathbb{P}\left[\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}p_{AB}) > \frac{\alpha_{\text{add}}}{4}\middle|\left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}, \mathcal{E}_1(\xi)^c, \mathcal{E}_2(\xi)^c\right]\right],
\end{aligned} \tag{D.35}$$

where in (a) we condition on $\left\{\delta_{AB}^{(i)}\right\}$, a particular realization of the MSC. In (b) we use the following fact: for any three events $\mathcal{E}_a, \mathcal{E}_b, \mathcal{E}_c$, the following inequality holds

$$\begin{aligned}
\mathbb{P}(\mathcal{E}_a) &= \mathbb{P}(\mathcal{E}_a|\mathcal{E}_b \cup \mathcal{E}_c)\mathbb{P}\left(\mathcal{E}_b \cup \mathcal{E}_c\right) + \mathbb{P}(\mathcal{E}_a|\mathcal{E}_b^c \cap \mathcal{E}_c^c)\mathbb{P}\left(\mathcal{E}_b^c \cap \mathcal{E}_c^c\right) \\
&\leq \mathbb{P}(\mathcal{E}_b \cup \mathcal{E}_c) + \mathbb{P}(\mathcal{E}_a|\mathcal{E}_b^c \cap \mathcal{E}_c^c) \\
&\leq \mathbb{P}\left(\mathcal{E}_b\right) + \mathbb{P}\left(\mathcal{E}_c\right) + \mathbb{P}(\mathcal{E}_a|\mathcal{E}_b^c \cap \mathcal{E}_c^c),
\end{aligned}$$

where we identify $\mathcal{E}_a, \mathcal{E}_b$, and $\mathcal{E}_c$ with the events $\widehat{d}_{AB} - d_{AB} > \frac{\alpha_{\text{add}}}{4}, \mathcal{E}_1(\xi)$, and $\mathcal{E}_2(\xi)$ respectively. Our goal now is to pick a value of ξ so that the third term in (D.35) is 0. Towards this end, we will use the following result that we prove in Section **??**.

Claim D.1. *For any $\xi > 0$, conditioned on a particular realization $\left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}$ of the MSC process, and the events $\mathcal{E}_1(\xi)^c$ and $\mathcal{E}_2(\xi)^c$, the following inequality holds*

$$|\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}p_{AB})| \leq \frac{2\xi}{\frac{e^{-4\mu_U\Delta/3}}{\frac{8}{3}\mu_U+1} - \frac{8\xi}{3}}. \tag{D.36}$$

Now, Claim D.1 tells us that if we make the following choice for ξ

$$\xi = \xi_0 \triangleq \frac{9\alpha_{\text{add}} e^{-\frac{4}{3}\mu_U \Delta}}{(24+\alpha_{\text{add}})(8\mu_U + 3)}, \tag{D.37}$$

then conditioned on the events $\mathcal{E}_1(\xi_0)^c$ and $\mathcal{E}_2(\xi_0)^c$, we have that

$$\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}p_{AB}) \leq \frac{\alpha_{\text{add}}}{4}$$

Therefore, we have

$$\mathbb{P}\left[\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}p_{AB}) > \frac{\alpha_{\text{add}}}{4} \middle| \left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}, \mathcal{E}_1(\xi_0)^c, \mathcal{E}_2(\xi_0)^c\right] = 0. \tag{D.38}$$

Using this in (D.35), we have

$$\begin{aligned}\mathbb{P}\left[\widehat{d}_{AB} - d_{AB} > \frac{\alpha_{\text{add}}}{4}\right] &\leq \mathbb{E}\left[\mathbb{P}\left(\mathcal{E}_1(\xi_0) \middle| \left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}\right)\right] + \mathbb{E}\left[\mathbb{P}\left(\mathcal{E}_2(\xi_0) \middle| \left\{\delta_{AB}^{(i)}\right\}_{i\in[m]}\right)\right] \\ &\leq e^{-2m\xi_0^2} + e^{-2mk\xi_0^2},\end{aligned} \tag{D.39}$$

where the second inequality comes from applying Hoeffding's inequality to each term, as in (D.8) and (D.9). Since this upper bound is independent of the choice of the pair of leaves, we can use (D.39) and (D.34) in (D.32) to get

$$\begin{aligned}\mathbb{P}\left[\text{Error}\right] &\leq \sum_{\substack{A,B,C,D\in L: \\ \tau_{AB}+\tau_{CD}\leq\tau_{AC}+\tau_{BD}=\tau_{AD}+\tau_{BC}}} 8\left(e^{-2m\xi_0^2} + e^{-2mk\xi_0^2}\right) \\ &\leq 8\binom{n}{4}\left(e^{-2m\xi_0^2} + e^{-2mk\xi_0^2}\right).\end{aligned} \tag{D.40}$$

Now, if we pick m and k as in (D.31) (also (5.9)), we see that the right side above is less than ϵ, which concludes the proof. The limit as $f \to 0$ can also be readily computed by observing that $\alpha_{\text{add}} \to 2\mu_L f$ as $f \to 0$. □

D.4.1 Proof of Claim D.1

We will begin by using the fact that $\ell(x)$ satisfies the following Lipschitz property: for any $0 \leq x \leq y \leq B$, we have

$$\begin{aligned}\ell(y) - \ell(x) &= -\frac{3}{4}\log\left(1 - \frac{4}{3}y\right) + \frac{3}{4}\log\left(1 - \frac{4}{3}x\right) \\ &= \int_x^y \frac{1}{1 - \frac{4}{3}t}\, dt \\ &\leq \frac{(y-x)}{1 - \frac{4}{3}B}. \end{aligned} \tag{D.41}$$

From this, we have that

$$\left|\ell\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)}\right) - \ell\left(\mathbb{E}[p_{AB}]\right)\right| \leq \frac{\xi}{1 - \frac{4}{3}\left(\mathbb{E}\left[p_{AB}\right] + \xi\right)}, \quad \text{conditioned on } \mathcal{E}_1(\xi), \tag{D.42}$$

where we have chosen the B (of (D.41)) to be $\mathbb{E}\left[p_{AB}\right] + \xi$, since conditioned on $\mathcal{E}_1(\xi)$, we have that

$$\frac{1}{m}\sum_{i=1}^{m} p_{AB}^{(i)} \leq \mathbb{E}\left[p_{AB}\right] + \xi. \tag{D.43}$$

Similarly, conditioned on $\mathcal{E}_2(\xi)^c$ and $\mathcal{E}_1(\xi)^c$, we have

$$\begin{aligned}\left|\ell\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)}\right) - \ell\left(\widehat{p}_{AB}\right)\right| &\leq \frac{\xi}{1-\frac{4}{3}\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} + \xi\right)} \\ &\leq \frac{\xi}{1-\frac{4}{3}\left(\mathbb{E}\left[p_{AB}\right] + 2\xi\right)},\end{aligned} \tag{D.44}$$

where in the first inequality we have chosen B (of (D.41)) to be $\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} + \xi$, since conditioned on $\mathcal{E}_2(\xi)^c$, we have that $\widehat{p}_{AB} \leq \frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)} + \xi$, and the second inequality follows from (D.43). Therefore from (D.42) and (D.44), we have that the following inequality holds conditioned on $\mathcal{E}_1(\xi)^c$ and $\mathcal{E}_2(\xi)^c$:

$$\begin{aligned}|\ell(\widehat{p}_{AB}) - \ell(\mathbb{E}\left[p_{AB}\right])| &\leq \left|\ell\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)}\right) - \ell\left(\mathbb{E}[p_{AB}]\right)\right| + \left|\ell\left(\frac{1}{m}\sum_{i\in[m]} p_{AB}^{(i)}\right) - \ell\left(\widehat{p}_{AB}\right)\right| \\ &\leq \frac{2\xi}{1-\frac{4}{3}\left(\mathbb{E}\left[p_{AB}\right] + 2\xi\right)}\end{aligned} \tag{D.45}$$

Finally, to conclude the proof of the claim, we bound $\mathbb{E}[p_{AB}]$ using the properties of the multispecies coalescent. Notice that, by definition, the random distance δ_{AB} is equal to $\mu_{AB} + 2Z_{AB}$, where μ_{AB} and Z_{AB} are as defined in Section D.3. Therefore,

$$\begin{aligned}\mathbb{E}\left[p_{AB}\right] &= \mathbb{E}\left[\frac{3}{4}(1 - e^{-\frac{4}{3}\delta_{AB}})\right] \\ &= \frac{3}{4}\left(1 - e^{-\frac{4}{3}\mu_{AB}}\mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\right]\right)\end{aligned} \tag{D.46}$$

Next, we observe that the random variable Z_{AB} is stochastically dominated by the

random variable $\mu_U Z$, where $Z \sim \text{Exp}(1)$. This implies that

$$\begin{aligned}\mathbb{E}\left[e^{-\frac{8}{3}Z_{AB}}\right] &\geq \mathbb{E}\left[e^{-\frac{8}{3}\mu_U Z}\right] \\ &= \frac{1}{\frac{8}{3}\mu_U + 1}.\end{aligned}$$

Using this and the fact that $\mu_{AB} \leq \mu_U \Delta$ in (D.46), we have

$$\mathbb{E}\left[p_{AB}\right] \leq \frac{3}{4}\left(1 - \frac{e^{-\frac{4}{3}\mu_U \Delta}}{\frac{8}{3}\mu_U + 1}\right).$$

Substituting this in (D.45) concludes the proof.

Remark. *As in the case of the proof of Theorem 5.2, we can prove Theorem 5.4 (upto small variations in the constants) using Bernstein's inequality and the conditional variance formula.*

Bibliography

[1] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor, *Graph sketches: sparsification, spanners, and subgraphs*, Proceedings of the 31st symposium on Principles of Database Systems, ACM, 2012, pp. 5–14.

[2] David Alderson, John Doyle, Ramesh Govindan, and Walter Willinger, *Toward an optimization-driven framework for designing and generating realistic internet topologies*, ACM SIGCOMM Computer Communication Review **33** (2003), no. 1, 41–46.

[3] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David Woodruff, *Efficient sketches for earth-mover distance, with applications*, in FOCS, 2009.

[4] Dana Angluin and Leslie G. Valiant, *Fast probabilistic algorithms for hamiltonian circuits and matchings*, Journal of Computer and system Sciences **18** (1979), no. 2, 155–193.

[5] M.F. Balcan and P. Gupta, *Robust Hierarchical Clustering*, Proceedings of the Conference on Learning Theory (COLT), July 2010.

[6] Moulinath Banerjee and Thomas Richardson, *On a dualization of graphical gaussian models: A correction note*, Scandinavian Journal of Statistics **30** (2003), no. 4, 817–820.

[7] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont, *Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data*, The Journal of Machine Learning Research **9** (2008), 485–516.

[8] R. Berinde, A.C. Gilbert, P. Indyk, H. Karloff, and M.J. Strauss, *Combining geometry and combinatorics: A unified approach to sparse signal recovery*, Communication, Control, and Computing, 2008 46th Annual Allerton Conference on, sept. 2008, pp. 798 –805.

[9] S Bernstein, *The theory of probabilities*, 1946.

[10] Dimitris Bertsimas and John N Tsitsiklis, *Introduction to linear optimization*, Athena Scientific Belmont, MA, 1997.

[11] P. J. Bickel and E. Levina, *Covariance regularization by thresholding*, Annals of Statistics **36** (2008), no. 6, 2577–2604.

[12] P. J. Bickel and E. Levina, *Regularized estimation of large covariance matrices*, Annals of Statistics **36** (2008), no. 1, 199–227.

[13] B. Bollobás, *Random graphs*, vol. 73, Cambridge university press, 2001.

[14] CAIDA, *The Skitter Project*, http://www.caida.org/, 2007.

[15] E.J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, Information Theory, IEEE Transactions on **52** (2006), no. 2, 489–509.

[16] E.J. Candes and T. Tao, *Decoding by linear programming*, Information Theory, IEEE Transactions on **51** (2005), no. 12, 4203–4215.

[17] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright, *Robust principal component analysis?*, Journal of the ACM (JACM) **58** (2011), no. 3, 11.

[18] R. Castro, M. Coates, and R. Nowak, *Likelihood Based Hierarchical Clustering*, IEEE Transactions on Signal Processing, vol. 52, August 2004, pp. 2308–2321.

[19] Venkat Chandrasekaran, Benjamin Recht, PabloA. Parrilo, and AlanS. Willsky, *The convex geometry of linear inverse problems*, Foundations of Computational Mathematics **12** (2012), 805–849 (English).

[20] Joseph T Chang, *Full reconstruction of markov models on evolutionary trees: identifiability and consistency*, Mathematical biosciences **137** (1996), no. 1, 51–73.

[21] Sanjay Chaudhuri, Mathias Drton, and Thomas S Richardson, *Estimation of a covariance matrix with zeros*, Biometrika **94** (2007), no. 1, 199–216.

[22] Herman Chernoff et al., *A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations*, The Annals of Mathematical Statistics **23** (1952), no. 4, 493–507.

[23] Albert Cohen, Wolfgang Dahmen, and Ronald Devore, *COMPRESSED SENSING AND BEST k-TERM APPROXIMATION*, Journal of the American Mathematical Society **22** (2009), no. 1, 211–231.

[24] Graham Cormode and S Muthukrishnan, *Combinatorial algorithms for compressed sensing*, Structural Information and Communication Complexity (2006), 280–294.

[25] Michael Costanzo, Anastasia Baryshnikova, Jeremy Bellay, Yungil Kim, Eric D Spear, Carolyn S Sevier, Huiming Ding, Judice LY Koh, Kiana Toufighi, Sara

Mostafavi, et al., *The genetic landscape of a cell*, science **327** (2010), no. 5964, 425–431.

[26] David R Cox, Nanny Wermuth, et al., *Linear dependencies represented by chain graphs*, Statistical Science **8** (1993), no. 3, 204–218.

[27] Gautam Dasarathy, Robert Nowak, and Sebastien Roch, *Data requirement for phylogenetic inference from multiple loci: A new distance method*, arXiv preprint arXiv:1404.7055 (2014).

[28] ———, *New sample complexity bounds for phylogenetic inference from multiple loci*, Proceedings of IEEE International Symposium on Information Theory (to appear), IEEE, 2014.

[29] Gautam Dasarathy, Parikshit Shah, Badri Narayan Bhaskar, and Robert Nowak, *Covariance sketching*, Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on, IEEE, 2012, pp. 1026–1033.

[30] ———, *Sketching sparse covariance matrices and graphs*, NIPS workshop on Randomized Methods in Machine Learning, 2013.

[31] ———, *Sketching sparse matrices*, arXiv preprint arXiv:1303.6544 (2013).

[32] Sanjoy Dasgupta and Philip M Long, *Performance guarantees for hierarchical clustering*, Journal of Computer and System Sciences **70** (2005), no. 4, 555–569.

[33] William HE Day, David S Johnson, and David Sankoff, *The computational complexity of inferring rooted phylogenies by parsimony*, Mathematical biosciences **81** (1986), no. 1, 33–42.

[34] Daniel Defays, *An efficient algorithm for a complete link method*, The Computer Journal **20** (1977), no. 4, 364–366.

[35] James H. Degnan, Michael DeGiorgio, David Bryant, and Noah A. Rosenberg, *Properties of consensus methods for inferring species trees from gene trees*, Systematic Biology **58** (2009), no. 1, 35–54.

[36] Arthur P Dempster, *Covariance selection*, Biometrics (1972), 157–175.

[37] J. DeRisi, V. Iyer, and P. Brown, *Exploring the metabolic and genetic control of gene expression on a genomic scale*, Science, vol. 278, October 1997, pp. 680–686.

[38] Peter J Diggle and Arūnas P Verbyla, *Nonparametric estimation of covariance structure in longitudinal data*, Biometrics (1998), 401–415.

[39] 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, pp. 2210–2220.

[40] David L Donoho and Michael Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via ?1 minimization*, Proceedings of the National Academy of Sciences **100** (2003), no. 5, 2197–2202.

[41] David L Donoho and Xiaoming Huo, *Uncertaintly Principles and Ideal Atomic Decomposition*, IEEE Transactions on Information Theory **47** (2001), no. 7, 2845–2862.

[42] David Leigh Donoho, *Compressed sensing*, Information Theory, IEEE Transactions on **52** (2006), no. 4, 1289–1306.

[43] Mathias Drton and Thomas S Richardson, *A new algorithm for maximum likelihood estimation in gaussian graphical models for marginal independence*, Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2002, pp. 184–191.

[44] Marco F Duarte and Richard G Baraniuk, *Kronecker compressive sensing*, Image Processing, IEEE Transactions on **21** (2012), no. 2, 494–504.

[45] M.F. Duarte and R.G. Baraniuk, *Kronecker product matrices for compressive sensing*, Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on, march 2010, pp. 3650 –3653.

[46] N. Duffield and F. L. Presti, *Network tomography from measured end-to-end delay covariance*, vol. 12, 2004, pp. 978–992.

[47] N. Duffield, F. L. Presti, V. Paxson, and D. Towsley, *Network loss tomography using striped unicast probes*, vol. 14, 2006, pp. 697–710.

[48] N.G. Duffield, J. Horowitz, and F. L. Presti, *Adaptive Multicast Topology Inference*, in Proceedings of IEEE INFOCOM '01, 2001, pp. 1636–1645.

[49] Rick Durrett, *Probability: theory and examples*, vol. 3, Cambridge university press, 2010.

[50] Peter L Erdos, Michael A Steel, László A Székely, and Tandy J Warnow, *A few logs suffice to build (almost) all trees (i)*, Random Structures and Algorithms **14** (1999), no. 2, 153–184.

[51] B. Eriksson, G. Dasarathy, P. Barford, and R. Nowak, *Toward the practical use of network tomography for internet topology discovery*, Proceedings of IEEE INFOCOM 2010 Conference (San Diego, CA), March 2010.

[52] Brian Eriksson, Gautam Dasarathy, Paul Barford, and Robert Nowak, *Efficient network tomography for internet topology discovery*, IEEE/ACM Transactions on Networking (TON) **20** (2012), no. 3, 931–943.

[53] Brian Eriksson, Gautam Dasarathy, Aarti Singh, and Robert Nowak, *Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities*, Proceedings of the International Conference on AI & Statistics (AISTATS), vol. 15, 2011, pp. 260–268.

[54] ———, *Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities*, arXiv preprint arXiv:1102.3887 (2011).

[55] Dave Evans, *The internet of things: How the next evolution of the internet is changing everything*, Cisco Internet Business Solutions Group (IBSG).

[56] J. Fan, Y. Fan, and J. Lv, *High dimensional covariance matrix estimation using a factor model*, Journal of Econometrics **147** (2007), no. 1, 43.

[57] Joseph Felsenstein, *Cases in which parsimony or compatibility methods will be positively misleading*, Systematic Biology **27** (1978), no. 4, 401–410.

[58] ———, *Evolutionary trees from dna sequences: a maximum likelihood approach*, Journal of molecular evolution **17** (1981), no. 6, 368–376.

[59] ———, *Inferring phylogenies*, vol. 2, Sinauer Associates Sunderland, 2004.

[60] W. Fitch and E. Margoliash, *Construction of Phylogenetic Trees*, Science, vol. 155, pp. 279–284.

[61] Les R Foulds and Ronald L Graham, *The steiner problem in phylogeny is np-complete*, Advances in Applied Mathematics **3** (1982), no. 1, 43–49.

[62] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, *Sparse inverse covariance estimation with the graphical lasso*, Biostatistics **9** (2008), no. 3, 432–441.

[63] Anna C Gilbert and Kirill Levchenko, *Compressing network graphs*, Proceedings of the LinkKDD workshop at the 10th ACM Conference on KDD, Citeseer, 2004.

[64] M. Girvan and M. Newman, *Community Structure in Social and Biological Networks*, Proceedings of the National Academy of Sciences, vol. 99, pp. 7821–7826.

[65] M. Grant and S. Boyd, *Graph implementations for nonsmooth convex programs*, Recent Advances in Learning and Control (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110.

[66] Rémi Gribonval and Morten Nielsen, *Sparse representations in unions of bases*, Information Theory, IEEE Transactions on **49** (2003), no. 12, 3320–3325.

[67] RC Griffiths and Simon Tavaré, *Ancestral inference in population genetics*, Statistical Science (1994), 307–319.

[68] S. Guha, R. Rastogi, and K. Shim, *ROCK: A Robust Clustering Algorithm for Categorical Attributes*, Information Systems, vol. 25, July 2000, pp. 345–366.

[69] Mehmet H. Gunes and Kamil Sarac, *Resolving IP aliases in building traceroute-based Internet maps*, Technical Report, 2006.

[70] Matthew W Hahn et al., *Bias in phylogenetic tree reconciliation methods: implications for vertebrate genome evolution*, Genome Biol **8** (2007), no. 7, R141.

[71] András Hajnal and Endre Szemerédi, *Proof of a conjecture of erdos*, Combinatorial theory and its applications **2** (1970), 601–623.

[72] D. Harrison and D.L. Rubinfeld, *Hedonic prices and the demand for clean air*, Journal of Environment, Economics, and Management, vol. 5, 1978, pp. 81–102.

[73] John A Hartigan, *Statistical theory in clustering*, Journal of classification **2** (1985), no. 1, 63–76.

[74] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer, 2001.

[75] Jarvis Haupt, Waheed U Bajwa, Gil Raz, and Robert Nowak, *Toeplitz compressed sensing matrices with applications to sparse channel estimation*, Information Theory, IEEE Transactions on **56** (2010), no. 11, 5862–5875.

[76] Larry V Hedges, Ingram Olkin, Mathematischer Statistiker, Ingram Olkin, and Ingram Olkin, *Statistical methods for meta-analysis*, Academic Press New York, 1985.

[77] Jotun J Hein, *An optimal algorithm to reconstruct trees from additive distance data*, Bulletin of Mathematical Biology **51** (1989), no. 5, 597–603.

[78] Wassily Hoeffding, *Probability inequalities for sums of bounded random variables*, Journal of the American statistical association **58** (1963), no. 301, 13–30.

[79] CVX Research, Inc., *CVX: Matlab software for disciplined convex programming, version 2.0 beta*, September 2012.

[80] Anil K Jain, M Narasimha Murty, and Patrick J Flynn, *Data clustering: a review*, ACM computing surveys (CSUR) **31** (1999), no. 3, 264–323.

[81] W.B. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, Contemporary mathematics **26** (1984), no. 189-206, 1–1.

[82] Iain M Johnstone, *On the distribution of the largest eigenvalue in principal components analysis*, Annals of statistics (2001), 295–327.

[83] Iain M Johnstone and Arthur Yu Lu, *Sparse principal components analysis*, Unpublished manuscript **7** (2004).

[84] S. Jokar, *Sparse recovery and kronecker products*, Information Sciences and Systems (CISS), 2010 44th Annual Conference on, march 2010, pp. 1 –4.

[85] Sadegh Jokar and Volker Mehrmann, *Sparse solutions to underdetermined kronecker product systems*, Linear Algebra and its Applications **431** (2009), no. 12, 2437–2447.

[86] A. B. Kahn, *Topological sorting of large networks*, Communications of the ACM, vol. 5, 1962, pp. 558–562.

[87] Daniel M. Kane, Jelani Nelson, Ely Porat, and David P. Woodruff, *Fast moment estimation in data streams in optimal space*, Proceedings of the 43rd annual ACM

symposium on Theory of computing (New York, NY, USA), STOC '11, ACM, 2011, pp. 745–754.

[88] G. Karypis, E. Han, and V. Kumar, *CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling*, IEEE Computer, vol. 32, 1999, pp. 68–75.

[89] MAmin Khajehnejad, Alexandros G Dimakis, Weiyu Xu, and Babak Hassibi, *Sparse recovery of nonnegative signals with minimal expansion*, Signal Processing, IEEE Transactions on **59** (2011), no. 1, 196–208.

[90] H. A. Kierstead and A. V. Kostochka, *A short proof of the Hajnal-Szemeredi Theorem on equitable colouring*, Comb. Probab. Comput. **17** (2008), no. 2, 265–270.

[91] S. Lauritzen, *Graphical models*, Clarendon Press, Oxford, 1996.

[92] Adam D Leaché and Bruce Rannala, *The accuracy of species tree estimation under simulation: a comparison of methods*, Systematic Biology **60** (2011), no. 2, 126–137.

[93] Liang Liu, Lili Yu, Laura Kubatko, Dennis K Pearl, and Scott V Edwards, *Coalescent methods for estimating phylogenetic trees*, Molecular Phylogenetics and Evolution **53** (2009), no. 1, 320–328.

[94] Liang Liu, Lili Yu, and Dennis Pearl, *Maximum tree: a consistent estimator of the species tree*, Journal of Mathematical Biology **60** (2010), 95–106, 10.1007/s00285-009-0260-0.

[95] Liang Liu, Lili Yu, Dennis K Pearl, and Scott V Edwards, *Estimating species phylogenies using coalescence times among sequences*, Systematic Biology **58** (2009), no. 5, 468–477.

[96] R. Castro M. Coates and R. Nowak, *Maximum Likelihood Network Topology Identification from Edge-Based Unicast Measurements*, June 2002.

[97] P. Madadevan, C. Hubble, D. Krioukov, B. Huffaker, and A. Vahdat, *Orbis: Rescaling degree correlations to generate annotated internet topologies*, Proceedings of ACM SIGCOMM Conference (Kyoto, Japan), August 2007.

[98] Wayne P Maddison, *Gene trees in species trees*, Systematic biology **46** (1997), no. 3, 523–536.

[99] Nicolai Meinshausen and Peter Bühlmann, *High-dimensional graphs and variable selection with the lasso*, The Annals of Statistics **34** (2006), no. 3, 1436–1462.

[100] Elchanan Mossel and Yuval Peres, *Information flow on trees*, The Annals of Applied Probability **13** (2003), no. 3, 817–844.

[101] Elchanan Mossel and Sebastien Roch, *Incomplete lineage sorting: consistent phylogeny estimation from multiple loci*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **7** (2010), no. 1, 166–171.

[102] S Muthukrishnan, *Data streams: Algorithms and applications*, Now Publishers Inc, 2005.

[103] Luay Nakhleh, *Computational approaches to species phylogeny inference and gene tree reconciliation*, Trends in ecology & evolution **28** (2013), no. 12, 719–728.

[104] Jerzy Neyman, *Molecular studies of evolution: a source of novel statistical problems*, Statistical decision theory and related topics (1971), 1–27.

[105] J. Ni, H. Xie, S. Tatikonda, and Y. R. Yang, *Efficient and Dynamic Routing Topology Inference from End-to-End Measurements*, IEEE/ACM Transactions on Networking, vol. 18, February 2010, pp. 123–135.

[106] Richard Nichols, *Gene trees and species trees are not the same*, Trends in Ecology & Evolution **16** (2001), no. 7, 358–364.

[107] J. Pearl and M. Tarsi, *Structuring Causal Trees*, Journal of Complexity, vol. 2, 1986, pp. 60–77.

[108] Sriram V. Pemmaraju, *Equitable colorings extend Chernoff-Hoeffding bounds*, Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms (Philadelphia, PA, USA), SODA '01, Society for Industrial and Applied Mathematics, 2001, pp. 924–925.

[109] M. Qiu, C. Xue, Z. Shao, Q. Zhuge, M. Liu, and E. Sha, *Efficent Algorithm of Energy Minimization for Heterogeneous Wireless Sensor Network*, Embedded and Ubiquitous Computing, Lecture Notes in Computer Science, 2006, pp. 25–34.

[110] R. Ramasubramanian, D. Malkhi, F. Kuhn, M. Balakrishnan, and A. Akella, *On The Treeness of Internet Latency and Bandwidth*, Proceedings of ACM SIGMETRICS Conference (Seattle, WA), 2009.

[111] Bruce Rannala and Ziheng Yang, *Bayes estimation of species divergence times and ancestral population sizes using dna sequences from multiple loci*, Genetics **164** (2003), no. 4, 1645–1656.

[112] Sylvia Ratnasamy and Steven McCanne, *Inference of multicast routing trees and*

bottleneck bandwidths using end-to-end measurements, INFOCOM'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 1, IEEE, 1999, pp. 353–360.

[113] P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu, *High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence*, Electronic Journal of Statistics **5** (2011).

[114] Sebastien Roch, *A short proof that phylogenetic tree reconstruction by maximum likelihood is hard*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **3** (2006), no. 1, 92.

[115] ———, *Toward extracting all phylogenetic information from matrices of evolutionary distances*, Science **327** (2010), no. 5971, 1376–1379.

[116] ———, *An analytical comparison of multilocus methods under the multispecies coalescent: the three-taxon case.*, Pacific Symposium on Biocomputing, World Scientific, 2013, pp. 297–306.

[117] Adam J Rothman, Elizaveta Levina, and Ji Zhu, *Generalized thresholding of large covariance matrices*, Journal of the American Statistical Association **104** (2009), no. 485, 177–186.

[118] Mark Rudelson and Roman Vershynin, *On sparse reconstruction from fourier and gaussian measurements*, Communications on Pure and Applied Mathematics **61** (2008), no. 8, 1025–1045.

[119] Karen Sachs, Omar Perez, Dana Pe'er, Douglas A Lauffenburger, and Garry P

Nolan, *Causal protein-signaling networks derived from multiparameter single-cell data*, Science Signalling **308** (2005), no. 5721, 523.

[120] Naruya Saitou and Masatoshi Nei, *The neighbor-joining method: a new method for reconstructing phylogenetic trees.*, Molecular biology and evolution **4** (1987), no. 4, 406–425.

[121] Charles Semple and Mike A Steel, *Phylogenetics*, vol. 24, Oxford University Press, 2003.

[122] R. Sherwood, A. Bender, and N. Spring, *DisCarte: A Disjunctive Internet Cartographer*, Proceedings of ACM SIGCOMM (Seattle, WA), August 2008.

[123] R. Sherwood and N. Spring, *Touring the internet in a TCP sidecar*, IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, 2006, pp. 339–344.

[124] M. Shih and A. Hero, *Hierarchical inference of unicast network topologies based on end-to-end measurements*, IEEE Transactions on Signal Processing, vol. 55, 2007, pp. 1708–1718.

[125] Robin Sibson, *Slink: an optimally efficient algorithm for the single-link cluster method*, The Computer Journal **16** (1973), no. 1, 30–34.

[126] Robert R Sokal and Charles D Michener, *A statistical method for evaluating systematic relationships*, University of Kansas, 1958.

[127] N. Spring, R. Mahajan, and D. Wetherall, *Measuring ISP Topologies with Rocketfuel*, Proceedings of ACM SIGCOMM '02 (Pittsburgh, PA), August 2002.

[128] R. K. Srivastava, R. P. Leone, and A. D. Shocker, *Market Structure Analysis: Hierarchical Clustering of Products Based on Substitution-in-Use*, The Journal of Marketing, vol. 45, pp. 38–48.

[129] M. A. Steel and L. A. Székely, *Inverting random functions. II. Explicit bounds for discrete maximum likelihood estimation, with applications*, SIAM J. Discrete Math. **15** (2002), no. 4, 562–575 (electronic).

[130] Mike Steel, *A basic limitation on inferring phylogenies by pairwise sequence comparisons*, Journal of Theoretical Biology **256** (2009), no. 3, 467 – 472.

[131] Simon Tavaré, *Some probabilistic and statistical problems in the analysis of dna sequences*, Lectures on mathematics in the life sciences **17** (1986), 57–86.

[132] Joel A Tropp, *Greed is good: Algorithmic results for sparse approximation*, Information Theory, IEEE Transactions on **50** (2004), no. 10, 2231–2242.

[133] Y. Tsang, M. Yildiz, P. Barford, and R. Nowak, *Network radar: tomography from round trip time measurements*, IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, 2004, pp. 175–180.

[134] Roman Vershynin, *Introduction to the non-asymptotic analysis of random matrices*, arXiv preprint arXiv:1011.3027 (2010).

[135] Ulrike Von Luxburg and Shai Ben-David, *Towards a statistical theory of clustering*, Pascal workshop on statistics and optimization of clustering, 2005.

[136] Martin J Wainwright, *Sharp thresholds for high-dimensional and noisy sparsity*

recovery using ℓ_1-constrained quadratic programming (lasso), Information Theory, IEEE Transactions on **55** (2009), no. 5, 2183–2202.

[137] Jimmy Yang and Tandy Warnow, *Fast and accurate methods for phylogenomic analyses*, BMC bioinformatics **12** (2011), no. Suppl 9, S4.

[138] Bin Yao, Ramesh Viswanathan, Fangzhe Chang, and Daniel Waddington, *Topology Inference in the Presence of Anonymous Routers*, In IEEE INFOCOM, 2003, pp. 353–363.

[139] H. Yu and M. Gerstein, *Genomic Analysis of the Hierarchical Structure of Regulatory Networks*, Proceedings of the National Academy of Sciences, vol. 103, 2006, pp. 14,724–14,731.

[140] Ming Yuan and Yi Lin, *Model selection and estimation in the gaussian graphical model*, Biometrika **94** (2007), no. 1, 19–35.