Doctoral Thesis

A study on structure recovery and the broadcasting problem

Vasiliki Velona

Thesis advisors:

Gábor Lugosi and Juanjo Rué

Στη μητέρα μου.

Abstract

In the first part of the thesis, we study the problem of recovering the structure underlying large Gaussian graphical models or, more generally, partial correlation graphs. In high-dimensional problems it is often too costly to store the entire sample covariance matrix. We propose a new input model in which one can query single entries of the covariance matrix. We prove that it is possible to recover the support of the inverse covariance matrix with low query and computational complexity. Our algorithms work in a regime when this support is represented by tree-like graphs and, more generally, for graphs of small treewidth. Our results demonstrate that for large classes of graphs, the structure of the corresponding partial correlation graphs can be determined much faster than even computing the empirical covariance matrix.

In the second part of the thesis, we study the broadcasting problem when the underlying tree is a random recursive tree. The root of the tree has a random bit value assigned. Every other vertex has the same bit value as its parent with probability $1 - q$ and the opposite value with probability q, where $q \in [0, 1]$. The broadcasting problem consists in estimating the value of the root bit upon observing the unlabeled tree, together with the bit value associated with every vertex. In a more difficult version of the problem, the unlabeled tree is observed but only the bit values of the leaves are observed. When the underlying tree is a uniform random recursive tree or a linear preferential attachment tree, in both variants of the problem we characterize the values of q for which the optimal reconstruction method has a probability of error bounded away from $1/2$. We also show that the probability of error is bounded by a constant times q. Two simple reconstruction rules are analyzed in detail. One of them is the simple majority vote, the other is the bit value of the centroid of the tree. We also analyze a third reconstruction rule which is more complex but works for all q where reconstruction is theoretically possible.

Acknowledgements

I am thankful to my advisors, Gábor and Juanjo, for giving me the opportunity to work on this thesis. I enjoyed interacting with both of them and I learned a lot. I would also like to thank Gábor for spending a lot of hours discussing with me on the mathematics of this thesis and passing on his enthusiasm. These hours were valuable to me.

I thank Anna Ben-Hamou and Cecilia Holmgren for providing comments on my thesis. Moreover, I thank Anna for hosting me in Paris for a research stay. This was done through a scholarship from the *Ferran Sunyer i Balaguer* foundation and I acknowledge their support as well.

I was lucky to work with my co-authors Jakub, Louigi, Luc, Piotr. I learned valuable things from all of them and I feel grateful for that.

Pompeu Fabra University supported and hosted me while working on this thesis. My time there was enjoyable.

A number of friends and colleagues in Barcelona made everyday happier. A non-exhaustive list has Christoph, Max, Julia, Adrian, Xinghua, Rahil, Sofia, Gerel, Malecky, Analia, Milena, Lauri, Claudia. Special thanks go to Pilar.

Finally, I am thankful to my family and friends in Greece for their constant love and support that was vital to keep me going.

Contents

Chapter 1

Introduction

This thesis contains a study of two problems of combinatorial statistics. The first one is *structure learning* for partial correlation graphs and the second one is the *broadcasting problem* on certain families of random recursive trees. In the next sections we explain each of these problems and provide an overview of the results of the thesis, accompanied with an overview of related work.

1.1 Structure recovery for graphical models and partial correlation graphs

Consider a Gaussian random vector $X = (X_1, \dots, X_n)$ and its covariance matrix Σ. Let K be the inverse covariance matrix and $(K_{ij})_{1 \le i,j \le n}$ be its entries. The matrix K encodes conditional independence relations. In particular, its entries satisfy that

$$K_{ij} = 0 \quad \Longleftrightarrow \quad X_i \perp\!\!\!\perp X_j \mid X_{[n]\setminus\{i,j\}} \,, \tag{1.1.1}$$

where the notation $X_i \perp\!\!\!\perp X_j \mid X_{[n]\setminus\{i,j\}}$ denotes that X_i is conditionally independent of X_j given $X_{[n]\setminus\{i,j\}}$ and $[n]$ is the set $\{1, \dots, n\}$; see, for instance, Lauritzen [60]. The set of Gaussian vectors that satisfy the same independence relations of this type is called Gaussian *graphical model*. We are therefore interested in learning the graph corresponding to the zeros of K, that is, a graph with n vertices where an edge ij exists if an only if $K_{ij} \neq 0$ (we call this the *partial correlation graph*). We assume that we have access to entries σ_{ij} of the covariance matrix Σ, through queries to a *covariance oracle*. The covariance oracle takes a pair of indices $i, j \in [n]$ as an input and outputs the corresponding entry σ_{ij} of Σ. Assuming that n is very big, we would like to store the least possible number of entries σ_{ij}. Our problem can be

formalized as follows.

- Given access to entries of the covariance matrix Σ, learn which entries of Σ^{-1} are non-zero using only a small fraction of all the entries of Σ.

We call the above problem *structure recovery*. Learning the partial correlation graph of Gaussian graphical models is the primary motivation of this work, but the results are presented in a more general framework. In the Gaussian setting, the partial correlation graph encodes the conditional independence relations. Therefore, we refer to the problem generally as *structure recovery for partial correlation graphs* and we do not assume Gaussianity, even though we use the Gaussian setting to motivate some results and tools. It should be noted that there are interesting interpretations of K also in the non-Gaussian setting and we give relevant references in Section 1.1.1.

We propose a series of algorithms for structure recovery, assuming that the partial correlation graphs satisfy certain sparsity conditions. The assumed sparsity is related to how much the graph resembles a tree. We give an algorithm for trees, one for graphs with small 2-connected components, and one for graphs of small *treewidth* (all these notions are precisely defined below). These algorithms are actually able to estimate K itself and not only learn the partial correlation graph. It should be stressed that the proposed algorithms invert a symmetric positive definite matrix and the analysis can be detached from its statistical connection and impact.

When an algorithm takes as input an entry σ_{ij}, we say that it performs a *covariance query* (or just a *query*) and we refer to the total number of queries as *query complexity*. The motivation for the use of covariance queries is that Σ might be too large to even store. In fact, our goal is to learn the partial correlation graph using $o(n^2)$ queries, since $\Theta(n^2)$ time is needed just to write down and store the covariance matrix–this is the starting point for a big part of the relevant literature and we provide references in detail in Section 1.1.1.

1.1.1 Related work

Learning the graph structure underlying probabilistic graphical models is a problem with a long history; see Drton and Maathuis [32] for a recent exposition. In the classical setting, when the number n of variables is reasonably small, this can be done by using stepwise selection procedures based on information criteria like BIC, AIC, or using the likelihood function; see Højsgaard, Edwards, and Lauritzen [44, Section 4.4] for a discussion.

In high-dimensional scenarios the methods proposed for Gaussian graphical models have become particularly successful. Here the graphs are encoded by ze-

ros in the inverse covariance matrix (or *precision matrix*) K. Specifically, an edge is present in the graph if and only if the corresponding element of K is not zero and so LASSO-type learning procedures can be applied Banerjee, Ghaoui, and d'Aspremont [7], Yuan and Lin [92]. The link between the entries of K and coefficients obtained by linearly regressing one variable against the rest gave rise to the so-called neighbor selection methods, see Meinshausen and Bühlmann [68]. In all these theoretical developments, the sample complexity required for learning the underlying graph is well understood. On the other hand, in these studies either computational issues played a secondary role or the computational budget was relatively large as all known methods require computing the sample covariance; see, for example, Cai, Liu, and Zhou [18], Dasarathy, Singh, Balcan and Park [25]. Hsieh, Sustik, Dhillon, Ravikumar, Poldrack [45] perform a careful analysis of the optimization objective used in earlier methods, which leads to a divide-and-conquer algorithm that can be applied to large data sets. More recently, Zhang, Fattahi, and Sojoudi [94] devised another scalable procedure based on thresholding the sample covariance matrix followed by a novel optimization procedure. However, the computational complexity of these approaches is still of at least quadradic order as a function of the number of variables.

In a growing number of applications, the number of variables n is so large that a computational cost of order n^2 becomes prohibitive. This means that even writing down or storing the covariance matrix (or an estimate of it) is not practical, rendering all aforementioned approaches unfeasible. This requires a different approach to structure recovery, which addresses the computational issues much more carefully. This computationally conscious approach has become more popular in the recent years where, in selected scenarios, it was possible to study the trade-off between statistical accuracy and computational complexity, see Chandrasekaran, and Jordan [20], Rudi, Camoriano, Rosasco [83]. Examples where this approach is useful occur in some applications in biology, such as the problem of reconstructing gene regulatory networks from large scale gene expression data. Hwang, Lee, and Bang [47] give an extensive discussion of computational challenges of massive amounts of gene expression data and note that issues of computational complexity made researchers rely on pairwise notions of dependence; see, for example, Chan, Stumpf, and Babtie [19], Zhang, Zhao, He, Lu, Cao, Liu, Hao, Liu, and Chen [95]. Scalable algorithms are also of interest in phylogenetics, where the problem is to reconstruct the evolutionary relationships between tens to hundreds of thousands of DNA sequences Price, Dehal, and Arkin [79], Brown and Truszkowski [14, 13]. Another example leading to large networks is building human brain functional connectivity networks using functional MRI data. In this setting, the data are

usually aggregated to obtain a data set with a moderate number of variables that can be processed with current algorithms Huang, Li, Sun, Ye, Fleisher, Wu, Chen, and Reiman [46].

In the case where the true underlying graph is a tree, the Chow-Liu algorithm Chow and Liu [22] is a widely used computationally efficient algorithm to search for the tree that maximizes the likelihood function. The method was originally proposed for categorical variables but it works in a much more general context with the Gaussian likelihood or any other modular criterion such as BIC, AIC as discussed by Edwards, De Abreu, and Labouriau [34]. In our setting the Chow-Liu algorithm is equivalent to computing a maximum-weight spanning tree in the complete graph with edge weights given by the absolute values of the correlations between any two variables. Although the Chow-Liu algorithm is relatively efficient and it has good statistical properties, the computational cost is of order $\Omega(n^2)$, which may be prohibitive in large-scale applications.

Similar ideas to this thesis have appeared in Jagadish and Sen [48], but the results did not go further than trees and the actual algorithms are different than the ones presented here. In a different context, similar ideas for Bayesian networks appear in Bello and Honorio [9].

In our main result for structure recovery, the proposed algorithm works under the assumption of small treewidth. Graphs with bounded treewidth form an important class of sparse graphs that have played a central role in graph algorithms. The class of graphs with small treewidth includes series-parallel graphs, outerplanar graphs, Halin graphs, Apollonian networks, and many others, see Bodlaender [11] for a general reference. Treewidth is known to be an essential parameter in inference and structure recovery for graphical models Chandrasekaran, Srebro, and Harsha [21], Kwisthout, Bodlaender, and van der Gaag [59], Wainwright and Jordan [90]. Moreover, bounded treewidth graphs have long been of interest in machine learning due to the low computational cost of inference in such models Chandrasekaran, Srebro, and Harsha [21], Karger and Srebro [54], Kwisthout, Bodlaender, and van der Gaag [59]. Current heuristics of treewidth estimation in real-world data have indicated small treewidth in various cases of interest Abu-Ata and Dragan [1], Adcock, Sullivan, and Mahoney [2], Maniu, Senellart, and Jog [67].

The main motivation of this work was learning Gaussian graphical models but our results are interesting in a much broader family of distributions. Vanishing partial correlations correspond to conditional independence in the Gaussian case but also in the non-paranormal case of Liu, Lafferty, and Wasserman [62], Liu, Han, Yuan, Lafferty, and Wasserman [61]. In general, partial correlation graphs inform only about linear dependences but there are still interesting situations when

much more is implied by vanishing partial correlations Rossell and Zwiernik [82]. In the tree case, conditional independence is implied not only for Gaussian and non-paranormal data but also for binary variables, or more generally, in situations where the dependence of adjacent variables in the tree is linear; see Zwiernik [96] for more details.

1.1.2 Overview of the results on structure recovery

In the biggest part of this study, we abstract away from statistical considerations. In particular, we propose the following input model for our analysis. The entries σ_{ij} can be accessed through queries to a *covariance oracle*. The covariance oracle takes a pair of indices $i, j \in [n]$ as an input and outputs the corresponding entry σ_{ij} of the matrix Σ. This is an idealized scenario that makes the main ideas more transparent. In practice, of course, these covariances are not exactly available as they are often estimated from data. This setup is meaningful in applications in which one may estimate, relatively easily and accurately, the covariance between any given pair of variables. Importantly, one does not need to estimate the entire covariance matrix. In the last section of Chapter 2, we discuss conditions under which the idealized covariance oracle may be replaced by a noisy version.

The *query complexity* of an algorithm is the number of entries of the covariance matrix Σ queried during the execution of the algorithm. The main findings of this work show that, in many nontrivial cases, the graph underlying the graphical model of X can be recovered with only $\mathcal{O}(n\, polylog(n))$ queries using randomized algorithms. The computational complexity of the proposed algorithms is also quasi-linear. This is a significant decrease in complexity compared to the quadratic complexity of any recovery algorithm that uses the entire (estimated) covariance matrix as a starting point.

Of course a so stated problem cannot be solved in full generality and the algorithms need to rely on the sparsity of K induced by bounds on related parameters of the underlying graph such as maximum degree and treewidth. We propose randomized procedures that recover the correct graph and have low query and computational complexity with high probability. This work is devoted to a careful analysis of three main cases: trees, graphs with small 2-connected components, and graphs with small treewidth. Our main result is an algorithm for each of the three cases, which recovers the correct graph with query and computational complexity $\mathcal{O}(n\, polylog(n))$.

Formulating simplified versions of the main results, we use the notation $\mathcal{O}_\alpha$ to denote that the complexity order contains a factor depending on parameters α.

For a given graph G over vertex set $[n] = \{1, \ldots, n\}$, we denote by $\mathcal{M}(G)$ the set of covariance matrices Σ satisfying $K_{ij} = 0$ for all $ij \notin E(G)$. By $\mathcal{G}(\Sigma)$ we denote the partial correlation graph associated to the covariance matrix Σ. Moreover, we say that $\Sigma \in \mathcal{M}(G)$ is generic to signify that it belongs to a dense open subset Γ of $\mathcal{M}(G)$ where certain desirable properties - to be defined - hold. Our first result studies computationally efficient ways to learn a tree.

Theorem 2.2.8 (Simplified version)**.** *Suppose $\mathcal{G}(\Sigma)$ is a tree T with n vertices and maximum degree $\Delta(T) \leq d$. Then there is an algorithm that outputs the correct tree and, with probability at least $1 - \epsilon$, works in time and query complexity $\mathcal{O}_{\epsilon,d}(n \log^2 n)$.*

We also show that these bounds are essentially optimal and the dependence on the maximum degree is essential.

Our second result is for graphs with small 2-connected components and small degree of the block-cut tree; see Section 1.3.1 for formal definitions.

Theorem 2.2.10 (Simplified version)**.** *Let $\mathcal{G}(\Sigma)$ be a graph of n vertices whose largest 2-connected component has size at most b and whose maximum degree of the block-cut tree is at most d. If $\Sigma \in \mathcal{M}(G)$ is generic, then there is an algorithm that outputs the correct graph and, with probability at least $1 - \epsilon$, works in time and query complexity $\mathcal{O}_{\epsilon,d,b}(n \log^2 n)$.*

Our main result is a randomized algorithm that is able to recover efficiently the partial correlation graph as long as it has bounded *treewidth* and maximum degree. (In fact, the algorithm remains efficient when both parameters grow slowly with n.)

Theorem 2.4.4 (Simplified version)**.** *Let G be a graph with n vertices, treewidth at most k, and maximum degree at most d. If $\Sigma \in \mathcal{M}(G)$ is generic, then there is an algorithm that outputs the correct concentration graph and, with probability at least $1 - \epsilon$, works in time and query complexity $\mathcal{O}_{\epsilon,k,d}(n \log^5 n)$.*

The algorithm we propose not only reconstructs the partial correlation graph but it also computes the inverse covariance matrix K. Since there are at most kn edges in a graph with treewidth k, there is no contradiction with the stated computational complexity.

The case when Σ is observed with error leads to additional complications. Solving this problem in full generality is beyond the scope of this work. In order to present the main ideas and some bottlenecks, in the final section of Chapter 2 we study the problem of recovering tree models when only a noisy covariance oracle is available.

1.2 The broadcasting problem

Moving to the second problem under study, we consider a *broadcasting process* on a tree to be the propagation of a message (let us say a bit value in $\{0,1\}$) from one node to all the rest, possibly corrupted. Our goal is to estimate the initial message. We assume that the tree is created dynamically at times $0,1,\ldots,n$, through a random process.

Consider random recursive trees defined as follows. At time i (beginning from time 0), the vertex with label i enters the system and attaches to a vertex j (then j is called the *parent* of i and i is called the *child* of j) with probability:

$$\mathbb{P}\{i \sim j\} = \frac{1}{i}\,.$$

This is called the *uniform attachment model* (we denote it shortly by UA) or *the uniform random recursive tree*, since every recursive tree, that is, a tree whose labels are nondecreasing when moving at any path beginning from vertex 0, has the same probability to appear. This model has been extensively studied, hence we refer the reader to the book Drmota [31] for an overview and references on this model.

If

$$\mathbb{P}\{i \sim j\} = \frac{D_j^+(i)+\beta}{\sum_{j<i}(D_j^+(i)+\beta)}\,,$$

where $\beta > 0$ is a parameter and $D_j^+(i)$ is the outdegree of vertex j at time i, then this is the *linear preferential attachment model* with parameter β. This is defined in a similar way as the classical preferential model (or *plane-oriented recursive tree*), where, instead of the outdegrees, one uses the total degrees. The results are formulated and proved in the linear preferential attachment model but all the proofs can be immediately transferred to the latter case as well. For convenience, we refer to the model as preferential attachment (or just PA), but we mean the linear preferential attachment model. This model is also well studied and we refer the reader again to Drmota [31].

We now consider that a bit value is passed from parents to their children, beginning from vertex 0 and proceeding recursively as follows: with probability $1-q$ the child vertex maintains the bit of its parent, otherwise it *flips*.

We study two problems, the first one is called *root-bit reconstruction* and the second one is called *reconstruction problem from leaf-bits*. In the former, we are given access to the shape of the tree and the bit values at the vertices but we do not know the time labels of the vertices, and we want to estimate the bit value of the root

vertex 0. In the second problem, we want to answer the same question but only observing the shape and the bit values of the leaves, that is, all the vertices of degree equal to one. In a concise way:

- Given access to a random unlabelled tree produced by either uniform attachment or preferential attachment and the bit values of the vertices, estimate the bit value of vertex zero.
- In a more difficult variant, answer the same question given only the bits of vertices with degree one (the *leaves*).

One estimates the value of the root bit B_0 by a value $\widehat{b} \in \{0,1\}$. The probability of error (or risk) is denoted by

$$R(n,q) = \mathbb{P}\left\{\widehat{b} \neq B_0\right\} .$$

We study the optimal risk

$$R^*(n,q) = \inf R(n,q) , \tag{1.2.1}$$

where the infimum is taken over all estimators $\widehat{b}$. In particular, we are interested in

$$R^*(q) = \limsup_{n\to\infty} R^*(n,q) .$$

If $R^*(q) < \frac{1}{2}$, we say that *reconstruction is possible*. We would like to know for which q reconstruction is possible, as well as upper bound $R^*(q)$ with a function of q.

We analyze three different estimators $\hat{b}$ in both models, for all parameters β. The first one is the majority rule, that is, the bit value of the majority of the observed vertices. The second estimator returns the bit value of the *centroid* vertex defined later. The third estimator requires the existence of a particular structure in the tree with positive probability and, under conditioning on that event, the bit value of the root can be guessed correctly with non-trivial probability.

1.2.1 Related work

The broadcasting problem on trees has a long and rich history. The form studied here was proposed by Evans, Kenyon, Peres, and Schulman [37]. We refer to this paper for the background of the problem and related literature. In the broadcasting problem of [37], a bit is transmitted from each node to its children recursively, beginning from the root vertex. Each time the bit is transmitted between two nodes,

the value of the bit is flipped with some probability. The authors study the problem of reconstructing the bit value of the root, based on the bit values of all vertices at distance k from the root. They establish a sharp threshold for the probability of reconstruction as k goes to infinity, depending on the tree's *branching number*. Some older papers are Kesten and Stigum [55], where the known as *Kesten-Stigum bound* was established for reconstruction in the case of complete d-ary trees, and Bleher, Ruiz, and Zagrebnov [10], where its converse was shown for the symmetric case and two colours. In fact, for optimal reconstruction in this case it is enough to know only the census at level k, i.e., the number of nodes of each colour Mossel and Peres [78]. What is more, it can be shown that any recursive algorithm is suboptimal comparing to majority in the class of *periodic trees* Mossel [73]. In the general case, the Kesten-Stigum threshold for reconstruction holds when only census information is available Mossel and Peres [78]. However, in many instances it is possible to reconstruct under this threshold when also the configuration is observable and the channel is binary asymmetric or q-ary symmetric Mossel [74]. In fact, in Sly [85] the author showed the converse direction of the Kesten-Stigum threshold for the 3-ary symmetric channel, when d is large, and also that the bound is not tight for $q \geq 5$. A recent paper Jain, Koehler, Liu, and Mossel [49] resolves a more general conjecture made in Evans, Kenyon, Peres, and Schulman [37], that any recursive algorithm (notably also *recursive* majority) is suboptimal, i.e., there is a noise level where reconstruction is possible information-theoretically but the algorithm is not better than a random guess.

The authors in Janson and Mossel [52] studied the *perturbed reconstruction problem*, where the observed vertices at the k-th level get some extra noise. In Moitra, Mossel, and Sandon [71], the authors examine circuit reconstruction algorithms and establish related complexity results. In Makur, Mossel, and Polyanskiy [66], the authors study the broadcasting problem in random directed acyclic graphs, equipped with a boolean function at each node. In that case, they give an example where reconstruction is possible with layers of size $\Theta(\log(k))$.

The broadcasting problem is closely related to various problems in different contexts. It is connected to the phylogenetic problem, i.e., the reconstruction of phylogenetic trees from molecular data on its leaves (see Mossel [75], Daskalakis, Mossel, and Polyanskiy [26, 27]). It also has ties with spin glass theory in physics Mézard and Montanari [69], Bleher, Ruiz, and Zagrabnov [10]. Moreover, it has ties with the community detection problem in the stochastic block model Mossel, Neyman, and Sly [77]. That happens since, intuitively, a neighbourhood of a node in that model resembles a Galton Watson tree, and the labels of its child-nodes have occurred by applying a noisy channel between them and the parent node.

In Mossel [76], the author gives a survey of results on recovering the initial bit in trees with given root and observable bits in some fixed distance from the root. In Mossel [73], the author studies the n-level broadcasting problem in periodic trees, when using recursive algorithms. The author finds a threshold value that does not depend on the shape of the initial tree, but only on its number of leaves.

As far as we know, the broadcasting problem has not been studied for random recursive trees. In the vast majority of the literature on the broadcasting problem, the location of the root is assumed to be known. Of course, in this case the reconstruction problem is meaningful only if the bit values near the root are not observed. The types of trees that are generally considered are such that, even if the root is not identified, it is easy to locate. In the problems that we consider, the trees are random recursive trees where localizing the root is a nontrivial issue. Hence, both the root-bit reconstruction problem and the problem of reconstruction from leaf bits are meaningful. The structure of the tree plays an important role in the solution of both problems.

The problem of localizing the root in different models of random recursive trees (the *inference-of-the-root* problem) has been studied by Haigh [40], Shah and Zaman [84], Bubeck, Devroye, and Lugosi [15]. For diverse results on closely related problems, see Curien, Duquesne, Kortchemski, and Manolescu [24], Bubeck, Mossel, and Rácz [17], Bubeck, Eldan, Mossel, and Rácz [16], Khim and Loh [56], Jog and Loh [53], Lugosi and Pereira [64], and Devroye and Reddad [80].

1.2.2 Overview of the results on broadcasting

We first present our main findings for the uniform attachment model. One of the main results is that the trivial lower bound $R^*(q) \geq q/2$ is tight, up to a constant factor (to see the lower bound, note that with probability q vertex 1 has different bit than the root and then they are statistically indistinguishable).

Theorem 1.2.1. *Consider the root-bit reconstruction problem in a uniform random recursive tree. Then*

$$\frac{q}{2} \leq R^*(q) \leq q$$

for all $q \in [0,1]$. *In the reconstruction problem from leaf bits,*

$$\frac{q}{2} \leq R^*(q) \leq 13q$$

for all $q \in [0,1]$.

Our other main result is that for the random recursive tree, we characterize the values of q for which $R^*(q) < 1/2$.

Theorem 1.2.2. *Consider the broadcasting problem in a uniform random recursive tree.*

1. *In the root-bit reconstruction problem $R^*(q) < 1/2$ if and only if $q \in [0,1)$.*
2. *In the reconstruction problem from leaf bits, $R^*(q) < 1/2$ if and only if $q \in [0,1/2) \cup (1/2,1)$.*

Note that in the reconstruction problem from leaf bits, one obviously has $R^*(1/2) = 1/2$. This follows from the fact that, when $q = 1/2$, the bit values on the vertices of the tree are independent unbiased coin tosses. With probability tending to one, the root of the tree is not a leaf and therefore its bit value is not observed. In all other cases (except when $q = 1$), an asymptotic probability of error strictly smaller than $1/2$ is achievable.

Perhaps the conceptually simplest method is the *majority* rule that simply counts the number of observed vertices with both bit values and decides according to the majority. Denote by $\widehat{b}_{\text{maj}}$ the majority. (In case of a voting tie we may arbitrarily define $\widehat{b}_{\text{maj}} = 0$.) This simple method has surprisingly good properties. Indeed, we prove the following bound.

Theorem 1.2.3. *Consider the broadcasting problem in a uniform random recursive tree. Denote the probability of error of the majority vote by*

$$R^{maj}(n,q) = \mathbb{P}\left\{\widehat{b}_{maj} \neq B_0\right\} .$$

For both the root-bit reconstruction problem and the reconstruction problem from leaf bits, the following hold.

1. *There exists $c > 0$ such that*

$$\limsup_{n\to\infty} R^{maj}(n,q) \leq cq \quad \text{for all } q \in [0,1] .$$

2. $$\limsup_{n\to\infty} R^{maj}(n,q) < 1/2 \quad \text{if } q \in [0,1/4)$$

and

$$\limsup_{n\to\infty} R^{maj}(n,q) = 1/2 \quad \text{if } q \in [1/4,1/2] .$$

A quite different approach is based on the idea that, if one is able to identify a vertex that is close to the root, then the bit value associated to that vertex is correlated to that of the root bit, giving rise to a meaningful guess of the root bit. The possibilities and limitations of identifying the root vertex have been thoroughly studied in recent years; see Section 1.2.1 for references.

A simple and natural candidate for an estimate of the root is the *centroid* of the tree. In order to define the centroid of a tree T, we need some notation. The *neighborhood* of a vertex v, that is, the set of vertices in T connected to v, is denoted by $N(v)$.

Define $\phi : V(T) \to \mathbb{R}^+$ by

$$\phi(v) = \max_{u \in N(v)} \left| V\left(T^v_{u\downarrow}\right)\right| ,$$

where $T^v_{u\downarrow}$ is the subgraph of T which contains the vertices whose unique path to v passes from u. Moreover, define a *centroid* of T by

$$v^* = \arg\min_{v \in V(T)} \phi(v) .$$

It is well known that a tree can have at most two centroids. In fact, $\phi(v^*) \leq \frac{|V(T)|}{2}$ and there are at most two vertices that attain the minimum value. If there are two of them, then they are connected with an edge (Harary [43]).

Equipped with this notion, now we may define an estimator $\widehat{b}_{\text{cent}}$ of the root bit in a natural way: (1) in the root-bit reconstruction problem, $\widehat{b}_{\text{cent}} = B_{v^*}$ is the bit value of an arbitrary centroid v^* of T; (2) in the reconstruction problem from leaf bits, let v^* be a centroid of T, let v° be a leaf closest to v^*, and let $\widehat{b}_{\text{cent}} = B_{v^\circ}$ be the associated bit value.

We call this estimator the *centroid rule*.

Theorem 1.2.4. *Consider the broadcasting problem in a uniform random recursive tree. Denote the probability of error of the centroid rule by*

$$R^{cent}(n,q) = \mathbb{P}\left\{\widehat{b}_{cent} \neq B_0\right\} .$$

For the root-bit reconstruction problem,

$$\limsup_{n\to\infty} R^{cent}(n,q) \leq q \quad \textit{for all } q \in [0,1]$$

and

$$\limsup_{n\to\infty} R^{cent}(n,q) \leq \frac{\log 2}{2} \approx 0.34 \quad \textit{for all } q \leq 1/2\,.$$

For the reconstruction problem from leaf bits,

$$\limsup_{n\to\infty} R^{cent}(n,q) \leq 13q \quad \textit{for all } q \in [0,1]\,.$$

Moreover,

$$\limsup_{n\to\infty} R^{cent}(n,q) < 1/2 \quad \textit{for all } q < 1/2\,.$$

Clearly, Theorem 1.2.4 implies Theorem 1.2.1. In order to prove Theorem 1.2.2, we need to construct an estimator of the root bit that performs better than random guessing when $q \in (1/2, 1)$. This is given by a more involved construction that will be precisely defined in the main body of the thesis.

We also show analogues of all the aforementioned theorems for the preferential attachment model. We defer their statement to the relevant sections of the thesis.

1.3 Basic definitions and notation

In this section we give a bulk of basic definitions that will be used throughout the thesis, for easy future reference.

1.3.1 Graph-theoretic definitions

A *graph* $G(V, E)$ is a pair of finite sets $V = V(G)$ and $E = E(G)$ called *vertices* (or *nodes*) and *edges*, where E is a set of subsets of V of size two. We typically write uv instead of $\{u, v\}$ to denote an edge and our graphs are simple, that is, $u \neq v$. A *subgraph* of G is a graph $G' = (V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$. For $V' \subseteq V$, denote by $G[V']$ the graph $(V', \{uv \in E | u, v \in V'\})$, called the *induced subgraph* of G on V'. If $S \subset V$ we write $G \setminus S$ to denote $G[V \setminus S]$. A *path* between u and v is a sequence of edges v_0v_1, v_1v_2,...,$v_{k-1}v_k$ with $v_0 = u$ and $v_k = v$. We allow for *empty paths* that consist of a single vertex. Two vertices $u, v \in V$ are *connected* if there is a path between u and v. For $v \in V$, the set $N(v) = \{u \in V | uv \in E\}$ is the *neighborhood* of v, $\deg(v) \stackrel{\text{def.}}{=} |N(v)|$ is its *degree*, $\Delta(G) \stackrel{\text{def.}}{=} \max_{v\in V} \deg(v)$ denotes the *maximum degree* of G, and $|V|$ is the *size* of G.

A graph on $n \geq 3$ vertices is a *cycle* if there is an ordering of its vertices $v_1, \ldots, v_n$, such that $E = \{v_1v_2, \ldots, v_{n-1}v_n, v_nv_1\}$. A graph is *connected* if all $u, v \in V$ are *connected*. A *tree* is a connected graph with no cycles.

A *connected component* of G is a maximal, with respect to inclusion, connected subgraph of G. A set $S \subseteq V$ *separates* $A, B \subseteq V$ *in* G if any path from A to B contains a vertex in S. Then S is called a *separator of A and B* in G. When S is of minimum size, it will be called a *minimal separator.* Note that we allow A and B to intersect, in which case $A \cap B$ needs to be contained in every separator of A and B. Denote by $\mathcal{C}^S$ the set of connected components of the graph $G \setminus S$. If S separates two disjoint sets A and B, then for every $u \in A \setminus S$ and $v \in B \setminus S$, u, v lie in two different connected components of $G \setminus S$. We will call a separator S *balanced* if all $C \in \mathcal{C}^S$ are small in some appropriate sense that will be made precise each time.

A graph is *2-connected* if for any vertex v, $G \setminus v$ is connected. If $V' \subseteq V$ is maximal, with respect to inclusion, such that $G[V']$ is 2-connected, then $G[V']$ is a *2-connected component* or *block* of G. For a given graph G, let $\mathcal{B}$ be the set of 2-connected components of G and let A be the set of *cut-vertices*, that is, vertices that belong to more than one 2-connected components. The *block-cut tree* $\mathrm{bc(G)}$ of G is a bipartite graph on $A \cup \mathcal{B}$ where an edge between $a \in A$ and $B \in \mathcal{B}$ exists if $a \in B$. A block-cut tree is a tree (see [43, Theorem 4.4]). In Figure 2.1 in Section 2.1, we provide an example of a graph and its block-cut tree.

Two graphs $G_1(V_1, E_1), G_2(V_1, E_2)$ are *equivalent under graph isomorphism* if there exists a bijection $\phi : V_1 \to V_2$ such that $(\{a, b\} \in E_1) \Leftrightarrow (\{\phi(a), \phi(b)\} \in E_2)$. If $G_1 = G_2$, we say that ϕ is an automorphism of G_1.

A graph is *rooted* if we distinguish one of its vertices as the *root*. A rooted tree with root ρ is called *d-ary* if all vertices are either leaves or have degree $d + 1$, apart from ρ which has degree d. Moreover, if all vertices up to distance $k - 1$ from ρ have degree $d + 1$ and the rest are leaves, we say that the tree is a *complete d-ary tree* of *depth k*. Here the distance of two vertices is considered to be the number of edges in the smallest path that connects them. The depth of a rooted tree is the largest distance from the root to another node.

The outdegree of a node in a rooted tree is its total degree minus 1, apart from the root vertex, whose degree is considered equal to its outdegree.

1.3.2 General notation

Let functions $g(n), f(n) : \mathbb{R}^+ \to \mathbb{R}^+$. We will say that $g(n)$ is $\mathcal{O}(f(n))$ when $\limsup_{n\to\infty} \frac{g(n)}{f(n)} < \infty$ and $\Omega(f(n))$ when $\liminf_{n\to\infty} \frac{g(n)}{f(n)} > 0$. Moreover, we will say that $g(n)$ is $o(f(n))$ when $\limsup_{n\to\infty} \frac{g(n)}{f(n)} = 0$. If the variable with respect to which we take the limits is unclear, we make it precise with a subscript (e.g. a function $f(n)$ could be $o_n(1)$).

We will denote by $|S|$ the *size* of a set S, that is, the number of elements in it.

Unless specified otherwise, the size of a graph will be consider the number of its vertices.

The notation $\stackrel{\text{def.}}{=}$ denotes definition. The acronym i.i.d. stands for *independent identically distributed*. The acronym a.s. stands for *almost surely*. The notation $\stackrel{d}{\Rightarrow}$ stands for *convergence in distribution* and $d_{TV}(X, Y)$ stands for the *total variation distance* between the random variables X, Y. In general, calligraphic characters will refer to random variables and plain characters to a fixed realization. For instance, $\mathcal{T}_n$ will refer to a random tree of $n+1$ vertices, when viewed as a random variable. In parallel, T_n will refer to a tree graph of $n+1$ vertices that is a realization of $\mathcal{T}_n$. Moreover, n-subscripts denote time when it is not otherwise specified.

1.4 Thesis outline

In Chapter 2 we provide proofs for the results concerning structure learning in graphical models and partial correlation graphs. In Chapter 3 we provide proofs for the results concerning the broadcasting problem.

Chapter 2

Structure learning in graphical models and partial correlation graphs

The results in this chapter are joint work with Gábor Lugosi, Jakub Truszkowski, and Piotr Zwiernik.

2.1 Preliminaries

Let $\Sigma = [\sigma_{ij}]$ be an $n \times n$ symmetric positive definite matrix and let $K \stackrel{\text{def.}}{=} \Sigma^{-1}$. For a given graph G over vertex set $[n] = \{1, \ldots, n\}$, denote by $\mathcal{M}(G)$ the set of covariance matrices Σ satisfying $K_{ij} = 0$ for all $ij \notin E(G)$. If Σ is a covariance matrix of a Gaussian random vector X then the condition $\Sigma \in \mathcal{M}(G)$ can be equivalently formulated through a set of conditional independence statements because of the equivalence (see [60])

$$K_{ij} = 0 \quad \iff \quad X_i \perp\!\!\!\perp X_j \mid X_{[n]\setminus\{i,j\}}. \tag{2.1.1}$$

For a given Σ, the *partial correlation graph* $\mathcal{G}(\Sigma) = ([n], E)$ is the graph with $E = \{ij | K_{ij} \neq 0\}$. The problem we deal with is the following:

- We are given access to individual entries of Σ. We want to learn the support of K observing only a small fraction of Σ.

Given a vector $x \in \mathbb{R}^n$ and a subset $A \subset [n]$ denote by x_A the subvector of x with entries x_i for $i \in A$. Similarly, for sets $A, B \subseteq [n]$ and a matrix $M \in \mathbb{R}^{n \times n}$, let $M_{A,B}$ denote the restriction of M to rows in A and columns in B. Write M_A for $M_{A,A}$.

If Σ is the covariance of X then $\Sigma_{A,B} = \mathrm{cov}(X_A, X_B)$. In this article we extensively use the following result of Seth Sullivant, Kelli Talaska, and Jan Draisma, which translates zero restrictions on a positive definite matrix K in terms of minors of Σ.

Theorem 2.1.1. [87, Theorem 2.15] *Let G be a connected graph with vertex set $[n]$. We have $\mathrm{rank}(\Sigma_{A,B}) \leq r$ for all $\Sigma \in \mathcal{M}(G)$ if and only if there is a set $S \subseteq [n]$ with $|S| \leq r$ such that S separates A and B in G. Consequently, $\mathrm{rank}(\Sigma_{A,B}) \leq \min\{|S| :$ S separates A and $B\}$. Moreover, there exists a dense open subset Γ of $\mathcal{M}(G)$ such that equality holds for all matrices in Γ.*

We call Γ a *generic* set. By a slight abuse of terminology, we call covariance matrices Σ in Γ, as well as the corresponding random vectors X, generic.

In this paper we assume that $\mathcal{G}(\Sigma)$ is connected or, equivalently by Theorem 2.1.1, that Σ has no zero entries. Without this assumption the problem quickly becomes impossible to solve. For example, whether $\mathcal{G}(\Sigma)$ has zero or one edge can only be decided after seeing the entire covariance matrix.

Assumption 1. The graph $\mathcal{G}(\Sigma)$ is connected.

Moreover, we assume that the genericity condition of Theorem 2.1.1 holds.

Assumption 2. The matrix $\Sigma \in \mathcal{M}(G)$ is always to be generic, or equivalently, for every $A, B \subseteq V$, $\mathrm{rank}(\Sigma_{A,B}) = \min\{|S| : S \text{ separates } A \text{ and } B\}$.

This assumption gives us the following important result that translates small sets of covariance queries into information about the underlying concentration graph $\mathcal{G}(\Sigma)$.

Lemma 2.1.2. *Under Assumption 2, $\mathrm{rank}(\Sigma_{AC,BC}) = \mathrm{rank}(\Sigma_{A,B})$ if and only if C is a subset of a minimal separator of A and B in $\mathcal{G}(\Sigma)$.*

Here and throughout we use the convention of writing $A \cup B$ as AB in subindices.

Proof. By Assumption 2, $\mathrm{rank}(\Sigma_{AC,BC})$ is the size of a minimal separator of $A \cup C$ and $B \cup C$, and $\mathrm{rank}(\Sigma_{A,B})$ is the size of a minimal separator of A and B. Since $\mathrm{rank}(\Sigma_{AC,BC}) = \mathrm{rank}(\Sigma_{A,B})$, there is a minimal separator of $A \cup C$ and $B \cup C$ that is also a minimal separator for A and B. By construction, this separator contains C. ∎

The following characterisation of graphical models over trees will be useful. This result is well known for Gaussian tree models (see, for example, Zwiernik [96]), but it is actually a purely algebraic result that holds in generality.

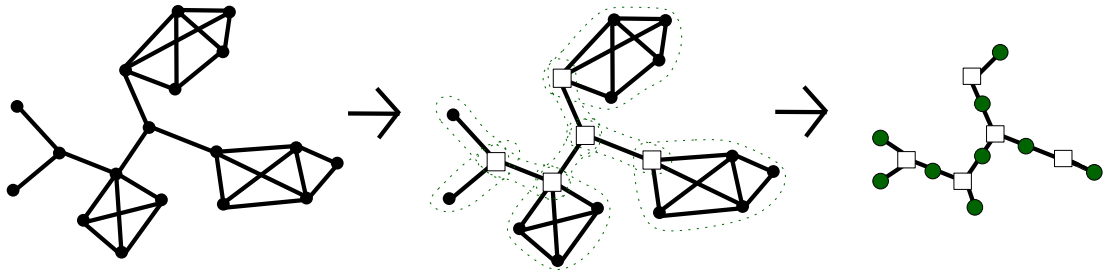


Figure 2.1: A graph, its cut-vertices (in squares) and its 2-connected components (circled by dotted curves), and the corresponding block-cut tree.

Lemma 2.1.3. *If $\Sigma \in \mathcal{M}(T)$ for a tree T, then for every $i, j \in V$, the normalized entries $\rho_{ij} = \sigma_{ij}/\sqrt{\sigma_{ii}\sigma_{jj}}$ for $i, j \in V$ satisfy the product formula*

$$\rho_{ij} = \prod_{uv \in \overline{ij}} \rho_{uv}, \tag{2.1.2}$$

where $\overline{ij}$ denotes the unique *path between i and j in T. Also, if the normalized entries ρ_{ij} in Σ satisfy (2.1.2) for some tree T then $\Sigma \in \mathcal{M}(T)$.*

For the sequel, recall the notation in Section 1.3.1.

2.2 Recovery of tree-like structures

In this section we discuss in detail procedures for learning trees and graphs with small 2-connected components. A graph is *2-connected* if for any vertex v, $G \setminus v$ is connected. If $V' \subseteq V$ is maximal, with respect to inclusion, such that $G[V']$ is 2-connected, then $G[V']$ is a *2-connected component* or *block* of G. Note that the requirement of small 2-connected components does not imply that the graph is sparse, that is, a linear number of edges.

For a given graph G, let $\mathcal{B}$ be the set of 2-connected components of G and let A be the set of *cut-vertices*, that is, vertices that belong to more than one 2-connected components. The *block-cut tree* bc(G) of G is a bipartite graph on $A \cup \mathcal{B}$ where an edge between $a \in A$ and $B \in \mathcal{B}$ exists if $a \in B$. A block-cut tree is a tree by [43, Theorem 4.4]. See Figure 2.1 for an example of a graph and its block-cut tree.

We will propose two learning procedures, one for recovering trees and one for recovering graphs with small 2-connected components. Both of them are divide-and-conquer type algorithms. First, we determine a cut vertex that splits the graph into relatively small pieces, then we identify the pieces and proceed recursively.

Therefore, the starting point of our analysis is to identify, at each step of the algorithm, a cut vertex that is balanced.

2.2.1 Centrality and balanced separators

Let $\mathcal{C}^v$ be the set of connected components of $G \setminus v$ and

$$c(v) \stackrel{\text{def.}}{=} \frac{1}{|V|-1} \max_{C \in \mathcal{C}^v} |C|. \tag{2.2.1}$$

Denote by v^* a vertex that attains the minimum such value, that is,

$$v^* = \operatorname*{argmin}_{v \in V} c(v) \, .$$

If G is a tree, v^* is called a *centroid*. It is a well-known fact (see [43, Theorem 4.3], for instance) that a tree can have at most two centroids and $c(v^*) \leq \frac{1}{2} \cdot \frac{|V|}{|V|-1}$.

In the first phase we efficiently find vertices with $c(v) \leq \alpha$ for a fixed $\alpha < 1$. To that end, we introduce a measure of vertex centrality, called *s-centrality* and denoted by $s(v)$. This can be used as a surrogate for $c(v)$ and its minimizer can be approximated efficiently. For $v \in V$, s-centrality is defined as

$$s(v) \stackrel{\text{def.}}{=} \frac{1}{(|V|-1)^2} \sum_{C \in \mathcal{C}^v} |C|^2. \tag{2.2.2}$$

Moreover, $v^\circ \stackrel{\text{def.}}{=} \operatorname{argmin}_{v \in V} s(v)$.

Lemma 2.2.1. *Let graph G. For every vertex $v \in G$, we have $s(v) \leq c(v) \leq \sqrt{s(v)}$. Moreover, $s(v^\circ) \leq c(v^*)$.*

Proof. Let $v \in V$ and $\mathcal{C}^v = \{C_1, \ldots, C_m\}$. The first inequality follows from

$$s(v) \;\leq\; \frac{1}{(|V|-1)^2} \sum_{j=1}^{m} |C_j| \max_i |C_i| \;=\; \frac{|V|-1}{(|V|-1)^2} \max_{i \in [m]} |C_i| \;=\; c(v) \, .$$

To show $c(v) \leq \sqrt{s(v)}$, consider the vector $p = (p_1, \ldots, p_m)$ with $p_i = \frac{|C_i|}{|V|-1}$. Then $c(v) = \|p\|_\infty$ and $s(v) = \|p\|_2^2$. The second inequality simply follows from the fact that $\|p\|_\infty \leq \|p\|_2$ for every $p \in \mathbb{R}^m$. To show the last inequality note that $s(v^\circ) \leq s(v^*)$ by the optimality of v° and $s(v^*) \leq c(v^*)$ by the first inequality that we proved. ∎

The procedure `sCentral` outlined in Algorithm 1 finds, with high probability, a vertex $\hat{v}$ with $s(\hat{v})$ close to $s(v^\circ)$. For each vertex $v \in V$ the algorithm approximates $s(v)$ by randomly sampling a few pairs u, w of vertices in $V \setminus \{v\}$ and checking if v separates u and w. By Lemma 2.1.3, this can be accomplished by checking if $\Sigma_{uv}\Sigma_{vw} = \Sigma_{uw}\Sigma_{vv}$, or equivalently, if $\det(\Sigma_{uv,vw}) = 0$.

The algorithm outputs a vertex with smallest approximate value of $s(v)$.

Algorithm 1: `sCentral`(V)

Parameter: κ;
$\hat{s}(v) := 0$ for all $v \in V$;
for *all* $v \in V$ **do**
  for $i = 1$ **to** κ **do**
    Pick u, w uniformly at random in $V \setminus \{v\}$;
    if $\det(\Sigma_{uv,vw}) \neq 0$ **then**
      $\hat{s}(v) := \hat{s}(v) + \frac{1}{\kappa}$;
Return $\arg\min_v \hat{s}(v)$;

Proposition 2.2.2. *Let $G(V, E)$ be a graph. The time and query complexity of computing $\hat{v} =$ `sCentral`(V) (in Algorithm 1) are both $\mathcal{O}(|V|\kappa)$. Moreover, for any $\delta > 0$,*

$$\mathbb{P}\left(s(\hat{v}) \geq s(v^\circ) + 2\delta\right) \;\leq\; 2|V| \exp\left(-2\delta^2\kappa\right).$$

Proof. The time and query complexity are obtained in a straightforward way. For the second statement note that, for every $v \in V$, $\kappa\hat{s}(v)$ is a binomial random variable with mean $\kappa s(v)$. Hence, by Hoeffding's inequality and the union bound, we obtain

$$\mathbb{P}[\max_v |\hat{s}(v) - s(v)| \geq \delta] \;\leq\; 2|V| \exp\left(-2\delta^2\kappa\right).$$

Let $\hat{v}$ be the output of Algorithm 1. We obtain

$$\begin{aligned}
\mathbb{P}[s(\hat{v}) \geq s(v^\circ) + 2\delta] &\leq \mathbb{P}[s(\hat{v}) - \hat{s}(\hat{v}) + \hat{s}(v^\circ) - s(v^\circ) \geq 2\delta] \quad (\text{since } \hat{s}(\hat{v}) \leq \hat{s}(v^\circ)) \\
&\leq \mathbb{P}[\max_v \{|\hat{s}(v) - s(v)|\} \geq \delta] \\
&\leq 2|V| \exp\left(-2\delta^2\kappa\right).
\end{aligned}$$

■

We now show that, with high probability, Algorithm 1 finds a vertex $\hat{v}$ with low centrality $c(\hat{v})$.

Proposition 2.2.3. *For any graph $G(V,E)$, if $s(v) < s(v^\circ) + 2\delta$, then*

$$c(v) \;<\; \sqrt{s(v^\circ)+2\delta} \;\leq\; \sqrt{c(v^*)+2\delta}.$$

In particular, if $G(V,E)$ is a tree with $|V| \geq 4$, and $\delta < \frac{1}{6}$, then $c(v) < 1$.

Proof. It follows from Lemma 2.2.1 and the fact that for trees

$$c(v^*) \leq \frac{1}{2} \cdot \frac{|V|}{|V|-1} \leq \frac{2}{3}\,.$$

■

Proposition 2.2.2 and Proposition 2.2.3 imply the following, upon fixing $\delta = 1/8$.

Corollary 2.2.4. *If $G(V,E)$ is a tree, $|V| \geq 4$, and $\hat{v} = \texttt{sCentral}(V)$ (in Algorithm 1), then*

$$\mathbb{P}\left(c(\hat{v}) > \sqrt{\frac{11}{12}}\right) \;\leq\; 2|V|\exp(-\kappa/32).$$

In Proposition 2.2.3 we used the fact that for trees $c(v^*) \leq \frac{2}{3}$ if $|V| \geq 4$. In general, for graphs with small 2-connected components we rely on the following.

Lemma 2.2.5. *Suppose $G(V,E)$ is a connected graph. Let $d \geq 2$ be a bound on the maximum degree of the block-cut tree of G, and let b be a bound on the size of the largest 2-connected component of G. If $|V| > db$, then*

$$c(v^*) \;\leq\; 1 - \frac{1}{2d}\,.$$

Proof. If $|V| > db$ then $b < |V|$ and therefore G has a cut vertex. Let C^* be the largest connected component in $\mathcal{C}^{v^*}$ and let B^* be the 2-connected component of G such that $v^* \in B^*$ and $B^* \setminus \{v^*\} \subseteq C^*$; see a depiction in Figure 2.2. We can assume that there exists a cut vertex of G in $B^* \setminus \{v^*\}$, because otherwise $B^* = C^* \cup \{v^*\}$ and then clearly $|V| \leq db$. For each cut vertex $v \neq v^*$ in B^*, let k_v be the number of all vertices in the union of all the connected components of $\mathcal{C}^v$ excluding the component containing B^* (grey blobs in Figure 2.2). By construction, $\sum_{v \neq v^*} k_v = |C^*| - |B^*|$ and there are at most $d-1$ such vertices. If $\overline{v} = \arg\max_{v \neq v^*} k_v$, then $k_{\overline{v}} \geq \frac{1}{d-1}(|C^*| - |B^*|)$. Denote by $\overline{C}$ the largest connected component of $G \setminus \{\overline{v}\}$. By optimality of v^*, $\overline{C}$ must be equal to the connected component containing the complement of C^*. Thus we have

$$|C^*| \;\leq\; |\overline{C}| \;\leq\; (|V| - 1 - |C^*|) + |B^*| + (|C^*| - |B^*| - k_{\overline{v}})$$

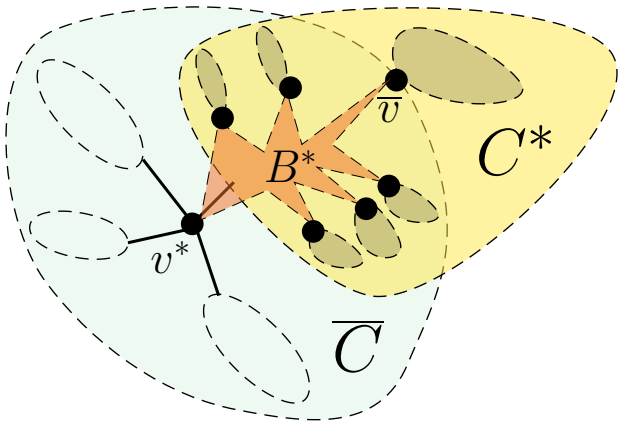


Figure 2.2: Illustration of the proof of Lemma 2.2.5

$$\leq \quad (|V| - 1 - |C^*|) + |B^*| + \frac{d-2}{d-1}(|C^*| - |B^*|)$$

Now simple algebra and the fact that $|B^*| \leq b$ allow us to rewrite this inequality as

$$c(v^*) \;\leq\; 1 - \frac{1}{d}\left(1 - \frac{b}{|V|-1}\right).$$

To conclude the proof, notice that $\frac{b}{|V|-1} \leq \frac{1}{2}$ if $|V| \geq db + 1$. ∎

Proposition 2.2.3 and Lemma 2.2.5 imply that whenever $s(v) < s(v^\circ) + 2\delta$ and $\delta < \frac{1}{4d}$ then $c(v) < 1$. By choosing $\delta = \frac{1}{8d}$, the following corollary follows from Proposition 2.2.2.

Corollary 2.2.6. *Let $G = (V, E)$ be a graph with $|V| > db$ and let $\hat{v} =$ `sCentral`(V). Then*

$$\mathbb{P}\left(c(\hat{v}) > \frac{\sqrt{4d-1}}{\sqrt{4d}}\right) \quad \leq \quad 2|V| \exp\left(-\frac{\kappa}{32d^2}\right).$$

2.2.2 Recovering a tree

In this section we present procedure `ReconstructTree` (Algorithm 2), which efficiently recovers the structure of the tree $T = \mathcal{G}(\Sigma)$. We start the procedure by running `ReconstructTree`$([n])$. The algorithm updates an edge set $\hat{E}$ that is initiated as $\hat{E} = \emptyset$. At each call, if $|V| > 1$, V gets partitioned into sets $V_1, \ldots, V_m$ by procedure `ComponentsTree` (Algorithm 3) and the edge set $\hat{E}$ gets updated. Then, `ReconstructTree` recurses into all the generated sets.

`ComponentsTree`(V) picks a central vertex $w =$ `sCentral`(V). Then it sorts, in descending order, the absolute values of the pairwise correlations $\rho_{uw} = \frac{\Sigma_{uw}}{\sqrt{\Sigma_{uu}\Sigma_{ww}}}$, where $u \in V \setminus \{w\}$, and places them in an ordered list B. For every vertex u in the list, the algorithm checks whether there exists an already known neighbour v of w that separates u from w, or equivalently, if $\det(\Sigma_{uw,vw}) \neq 0$; see Section 2.2.1. If this is the case, then it adds u to the connected component where v belongs after removing w. Otherwise, a new connected component is registered that corresponds to the vertex u and the edge uw is added in $\hat{E}$. In the end, `ComponentsTree` returns the vertex sets of all such connected components $V_1, \ldots, V_m$. The edges between w and each of the m neighbours in the m connected components are added to $\hat{E}$.

Algorithm 2: `ReconstructTree`(V)

$\hat{E} := \emptyset$;
if $|V| > 1$ **then**
  $V_1, \ldots, V_m \leftarrow$ `ComponentsTree`(V);
  for i from 1 to m **do**
    `ReconstructTree`(V_i);
Return $\hat{E}$;

Algorithm 3: `ComponentsTree`(V)

`// uses a global` $\hat{E}$ `when is called by ReconstructTree`
$w \leftarrow$ `sCentral`(V);
$N := \emptyset$;
Sort $|\rho_{uw}|$ for $u \in V \setminus \{w\}$ in decreasing order and put them in list B;
for every u in the order of B **do**
  $t := true$;
  for *all* $v \in N$ **do**
    if $\det(\Sigma_{uw,vw}) \neq 0$ **then**
      $V_v \leftarrow V_v \cup \{u\}$; $t := false$;
  if t=*true* **then**
    $\hat{E} \leftarrow \hat{E} \cup \{uw\}$;
    $N \leftarrow N \cup \{u\}$, $V_u \leftarrow \{u\}$;
Return all V_u for $u \in N$;

Proposition 2.2.7. *Algorithm 2 is correct; that is, if $\mathcal{G}(\Sigma)$ is a tree T, then algorithm* `ReconstructTree`$([n])$ *returns $\hat{E} = E(T)$.*

Proof. After every call of `ComponentsTree`(V) it holds that $\bigcup_{i=1}^{m} V_i = V \setminus \{w\}$, hence `ReconstructTree`$([n])$ always terminates.

For every $w \in [n]$, either $\{w\}$ is one of the components $V_1, \dots, V_m$ that are returned by `ComponentsTree`(V) (call such a vertex terminal) or this does not happen and w is the output of `sCentral`(V) in one of the calls of the subroutine `ComponentsTree` (call such a vertex central).

Initially the algorithm picks a vertex $w =$ `sCentral`$([n])$, which induces the partition of $[n] \setminus \{w\}$, $\mathcal{C}^w = \{V_1, \dots, V_m\}$. The vertices $u \in [n] \setminus \{w\}$ are examined in descending order with respect to $|\rho_{uw}|$. Let $v \in N(w)$ be adjacent to w and let u be any other other vertex in the same connected component $C \in \mathcal{C}^w$ as v. Then v separates u and w and, in particular, by Lemma 2.1.3 we have $|\rho_{vw}| > |\rho_{uw}|$. This shows that, for any $C \in \mathcal{C}^w$, the vertex v in C which is a neighbour of w comes earlier in the order specified in the algorithm than any other vertex in C. Hence, $v \in N$ (c.f. Algorithm 3) and for all other $u \in C$ it holds that $\det(\Sigma_{uw,vw}) \neq 0$. This shows that in the first call of `ComponentsTree` the algorithm:

(i) adds to $\hat{E}$ the edges between the central vertex w and its neighbours in T,

(ii) assigns all vertices to their connected components in $\mathcal{C}^w$.

Since each $G[V_i]$ is a tree, the same argument can be applied to subsequent calls of `ReconstructTree`. Hence, by induction, these two properties hold at any call of the algorithm `ComponentsTree`. In particular, $\hat{E} \subseteq E(T)$. To show the opposite inclusion first note that if $uv \in E(T)$ and u or v is central, then $uv \in \hat{E}$ by (i). Moreover, if u, v are both terminal, then there is some call of `ComponentsTree` that places them in different sets V_i. Then by (ii), there is no edge uv in E. ∎

The subroutine `sCentral` is a probabilistic component of the algorithm that is essential to obtain good complexity bounds.

Theorem 2.2.8. *Suppose $\mathcal{G}(\Sigma)$ is a tree $T([n], E)$ with maximum degree $\Delta(T) \leq d$. Fix $\epsilon < 1$ and define $\kappa = \lceil 32 \log\left(\frac{2n^2}{\epsilon}\right) \rceil$ to be the parameter of Algorithm 1. Then, with probability at least $1 - \epsilon$, Algorithm 2 requires time and queries of the order*

$$\mathcal{O}\left(n \log(n) \max\left\{\log\left(\frac{n}{\epsilon}\right), d\right\}\right).$$

Proof. First we analyze the complexity of one call of `ComponentsTree`(V). By Proposition 2.2.2, the call of `sCentral`(V) takes time and queries of the order $\mathcal{O}(|V|\kappa)$. We then query $|V|$ pairwise correlations and sort them, which takes time $\mathcal{O}(|V| \log |V|)$. Partitioning V into sets V_i takes time and queries both of order

$\mathcal{O}(d|V|)$ since $|N| \leq d$. For all calls of $\texttt{ReconstructTree}(V_i)$ in a recursion level (i.e., distance from the first `ReconstructTree` call in the recursion tree), it holds that $V_i \cap V_j = \emptyset$. Hence, in each recursion level the time complexity is of order $\mathcal{O}(n \log n + n\kappa + nd)$ and the query complexity is of order $\mathcal{O}(n\kappa + nd)$.

Assume first that $\hat{v} = \texttt{sCentral}(V)$ satisfies $c(\hat{v}) \leq \alpha := \sqrt{11/12}$ in each call with $|V| \geq 4$. In this case the recursion depth is at most $\log_{1/\alpha}(n) + 4$ and, overall, the algorithm has time complexity $\mathcal{O}\big(n \log(n)(\log(n) + \kappa + d)\big)$ and query complexity $\mathcal{O}\big(n \log(n)(\kappa + d)\big)$. Since $\kappa = \mathcal{O}(\log(n/\epsilon) + \log d)$, the announced bounds follow.

It remains to show that, with the given choice of κ, with probability at least $1 - \epsilon$ we get that $c(\hat{v}) \leq \alpha$ in each call with $|V| \geq 4$. By Corollary 2.2.4, in a single call the probability that $c(\hat{v}) > \alpha$ is at most $2|V| \exp(-\kappa/32)$, which is further bounded by $2n \exp(-\kappa/32)$. As $\texttt{ReconstructTree}([n])$ runs, the procedure `sCentral` is called at most n times, which is the total number of available vertices. From the union bound, the probability that in at least one call $\hat{v} = \texttt{sCentral}(V)$ satisfies $c(\hat{v}) > \alpha$ is at most $2n^2 \exp(-\kappa/32)$. Demanding the latter to be at most ϵ, we obtain the indicated value for κ and the desired result. ∎

2.2.3 Graphs with small blocks

We now present an algorithm that recovers concentration graphs with small 2-connected components and small maximum degree of the block-cut tree. The procedure `ReconstructSB` (Algorithm 4) takes as input a vertex set V and, like in the tree case, updates the global variable $\hat{E}$, which is initially set as $\hat{E} := \emptyset$. If V is small enough, that is, $|V| \leq db$, then the algorithm reconstructs the induced graph over V by directly inverting the matrix Σ_V. Otherwise it calls `ComponentsSB`, which first finds a vertex $w = \texttt{sCentral}(V)$ and returns sets $C \cup \{w\}$ for all $C \in \mathcal{C}^w$. This part of the algorithm is similar to $\texttt{ComponentsTree}(V)$, but the edges incident with w are not recovered at this stage.

Algorithm 4: $\texttt{ReconstructSB}(V)$

```
if |V| > db then
    V_1, ..., V_m ← ComponentsSB(V);
    for i from 1 to m do
        ReconstructSB(V_i);
else
    Ê ← Ê ∪ E(G(Σ_{V,V}));
```

Algorithm 5: `ComponentsSB`(V)

$w \leftarrow$ `sCentral`(V);
$N := \emptyset$; `// contains one vertex from each` $C \in \mathcal{C}^w$
for all $u \in V \setminus \{w\}$ **do**
  if there exists $v \in N$, $\det(\Sigma_{uw,vw}) \neq 0$ **then**
    $V_v \leftarrow V_v \cup \{u\}$;
  else
    $V_u \leftarrow \{u\}$;
    $N \leftarrow N \cup \{u\}$;
return $V_u \cup \{w\}$ *for all* $u \in N$;

Proposition 2.2.9. *Algorithm 4 is correct; that is, if* $\Sigma \in \mathcal{M}(G)$ *and* Σ *is generic then* `ReconstructSB`$([n])$ *returns* $\hat{E}$ *equal to* $E(G)$.

Proof. Assume we are on the first call of `ReconstructSB`. If $n \leq db$ then the algorithm outputs $\mathcal{G}(\Sigma)$, which, by definition, is the correct graph. If $n > db$, then G contains a cut vertex, so with probability 1 a cut vertex w will be eventually found by `sCentral`$([n])$ at some call of `ComponentsSB`. Let $C_1, \ldots, C_m$ be the connected components of $G \setminus \{w\}$. The sets V_i produced by this call of `sCentral`$([n])$ correspond to the sets $C_1 \cup \{w\}, \ldots, C_m \cup \{w\}$. This is clear by Lemma 2.1.2: a vertex u belongs to the same connected component as v in $G \setminus \{w\}$ if and only if w does not separate u and v in G. Note also that for any $A, B \subset V_i$ any minimal separator of A and B is contained in V_i. In particular, by Theorem 2.1.1,

(i) For every V_i, the edge-set of $G[V_i]$ is the same as the edge-set of the graph of the marginal distribution, $\mathcal{G}(\Sigma_{V_i})$.

By induction, statement (i) holds for every call of `ReconstructSB`. ∎

Theorem 2.2.10. *Let* $G([n], E)$ *be a graph whose maximum degree of the block-cut tree is bounded by* d*, and let* b *be a bound on the size of the largest 2-connected component. Fix* $\epsilon < 1$ *and define* $\kappa = \lceil 32d^2 \log\left(\frac{2dn}{\epsilon}\right) \rceil$ *to be the parameter of Algorithm 1. If* $\Sigma \in \mathcal{M}(G)$*,* Σ *is generic and* $G(\Sigma)$ *connected, then with probability at least* $1 - \epsilon$*, Algorithm 4 runs with query and time complexity of order*

$$\mathcal{O}\left(d^4 n \log(n)\left(\log\left(\frac{n}{\epsilon}\right) + b^2\right)\right) \quad \textit{and} \quad \mathcal{O}\left(d^4 n \log(n)\left(\log\left(\frac{n}{\epsilon}\right) + db^3\right)\right)$$

Proof. First we analyze the complexity of one call of `ComponentsSB`(V). If $|V| \leq db$ then $\mathcal{G}(\Sigma_{V,V})$ is obtained. Since matrix inversion takes at most cubic time,

the time and queries required are $\mathcal{O}(d^3b^3)$ and $\mathcal{O}(d^2b^2)$ respectively. If $|V| > db$ in $\texttt{ReconstructSB}(V)$, then $\texttt{ComponentsSB}(V)$ is called, which calls $\texttt{sCentral}(V)$. By Proposition 2.2.2, the call of $\texttt{sCentral}(V)$ takes time and queries of the order $\mathcal{O}(|V|\kappa)$. The latter provides the splitting vertex w and then V is split into at most d sets $U_i = V_i \cup \{w\}$. This last step takes $\mathcal{O}(|V|d)$ queries and time. Hence, each call $\texttt{ReconstructSB}(V)$ requires

$$\mathcal{O}\Big(|V|\kappa + |V|d + d^2b^2\Big) \quad \text{and} \quad \mathcal{O}\Big(|V|\kappa + |V|d + d^3b^3\Big)$$

queries and time, respectively.

Let $U_1, \ldots, U_r$ be the sets on which the algorithm recurses on the i-th level of the recursion tree. By construction, these sets are not disjoint and, for each $v \in V$, at most d copies of it are created during the algorithm. Hence, in each recursion level there are at most nd vertices, implying that $\sum_{i=1}^{r} |U_i| \leq nd$ and that $r \leq nd$. Using the complexity bounds for a single call of `ReconstructSB`, we get that any recursion level in the recursion tree requires

$$\mathcal{O}\Big(nd\kappa + nd^3b^2\Big) \quad \text{and} \quad \mathcal{O}\Big(nd\kappa + nd^4b^3\Big)$$

queries and time, respectively.

Assume first that $\hat{v} = \texttt{sCentral}(V)$ satisfies $c(\hat{v}) \leq \alpha \stackrel{\text{def.}}{=} \sqrt{4d-1}/\sqrt{4d}$ in each call. In this case the recursion depth is at most $\log_{1/\alpha}(n)$ and overall the algorithm requires

$$\mathcal{O}\left(\frac{dn\log(n)}{\log(1/\alpha)}\left(\kappa + d^2b^2\right)\right) \quad \text{and} \quad \mathcal{O}\left(\frac{dn\log(n)}{\log(1/\alpha)}\left(\kappa + d^3b^3\right)\right)$$

queries and time, respectively. Since $\kappa = \mathcal{O}(d^2\log(n/\epsilon))$ and $1/\log(1/\alpha) \leq 20d$, we obtain the expressions in the statement of the theorem.

It remains to show that with the given choice of κ, with probability at least $1-\epsilon$, we get that $c(\hat{v}) \leq \alpha$ in each call. By Corollary 2.2.6, in a single call the probability that $c(\hat{v}) > \alpha$ is at most $2|V|\exp\left(-\frac{\kappa}{32d^2}\right)$. As $\texttt{ReconstructSB}([n])$ runs, the procedure `sCentral` is called at most dn times. From the union bound, the probability that in at least one call $\hat{v} = \texttt{sCentral}(V)$ satisfies $c(\hat{v}) > \alpha$ is at most $2dn\exp(-\kappa/(32d^2))$. This is at most ϵ for the indicated value of κ. ∎

2.2.4 A lower bound

In this section we show that the result of Theorem 2.2.8 is optimal up to logarithmic factors, in the sense that one cannot reconstruct trees with maximum degree d with less than $\Omega(dn)$ covariance queries. We should note that this lower bound is implied by the results in King, Li, and Zu [57]. We give an alternative proof that fits directly in our context.

Let $\mathcal{X}$ be the class of $n \times n$ covariance matrices whose concentration graph is a tree. We write $T(\Sigma)$ for the tree induced by $\Sigma \in \mathcal{X}$. We also denote by $\mathcal{X}_d$ the class of covariance matrices whose concentration graph is a tree of maximum degree bounded by d. In our construction we use the characterization of the class $\mathcal{X}$ given in Lemma 2.1.3.

We first prove that any algorithm that recovers the correct tree (without any restriction on the maximum degree) needs to access the covariance oracle $\Omega(n^2)$ times.

In order to formalize such a statement, let $\mathcal{A}_k$ be the class of all randomized algorithms that query the covariance oracle at most k times. An algorithm $A \in \mathcal{A}_k$ outputs the tree $\mathcal{T}(A)$. The probability of error of algorithm A for $\Sigma \in \mathcal{X}$ is denoted by

$$P(A, \Sigma) = \mathbb{P}\left\{\mathcal{T}(A) \neq T(\Sigma)\right\} ,$$

where the probability is with respect to the randomization of the algorithm A. The quantity of interest is the *minimax risk*

$$R(\mathcal{A}_k, \mathcal{X}) = \inf_{A \in \mathcal{A}_k} \sup_{\Sigma \in \mathcal{X}} P(A, \Sigma) .$$

$R(\mathcal{A}_k, \mathcal{X})$ expresses the worst-case probability of error of the best algorithm that takes at most k covariance queries.

Theorem 2.2.8 implies that there exists a constant $c > 0$ such that, for every $\epsilon > 0$, we have $R(\mathcal{A}_k, \mathcal{X}_d) \leq \epsilon$ whenever $k > cn \log(n)\,(d + \log(n/\epsilon))$. In this section we prove that this upper bound is tight up to logarithmic factors.

We start with the case $d = n - 1$ (i.e., no restriction on the maximum degree) since this simpler case already contains the main ideas. The lower bound for $R(\mathcal{A}_k, \mathcal{X}_d)$ follows by a small adjustment.

Theorem 2.2.11. *For all $k \leq \binom{n}{2}$,*

$$R(\mathcal{A}_k, \mathcal{X}) \geq \frac{1}{2} - \frac{k}{(n-1)^2} .$$

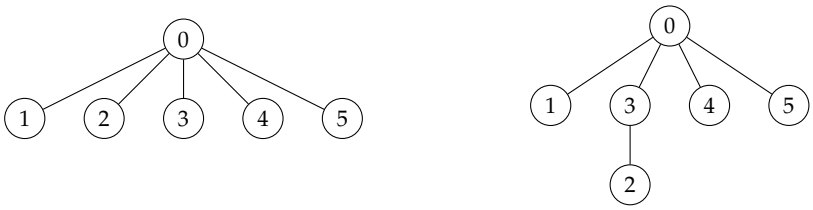


Figure 2.3: An illustration of the construction in the proof of Theorem 2.2.11, with $n = 6$, $\{I, J\} = \{2, 3\}$, and $B = 0$ (left), $B = 1$ (right).

In particular, $R(\mathcal{A}_k, \mathcal{X}) \geq 1/2 - o(1)$ *whenever* $k = o(n^2)$.

Proof. In order to prove the lower bound, we define a probability distribution $\mathcal{D}$ on the set $\mathcal{X}$ and write

$$R(\mathcal{A}_k, \mathcal{X}) \geq \inf_{A \in \mathcal{A}_k} \mathbb{E}_{\Sigma \sim \mathcal{D}} P(A, \Sigma) \, .$$

Next we specify how a random symmetric matrix Σ, distributed according to $\mathcal{D}$, is generated. Σ is defined by a collection of independent random variables: let B be a Bernoulli random variable with parameter $1/2$, let $U_1, \ldots, U_{n-1}$ be independent random variables, uniformly distributed on $[0, 1]$, and let I, J be different indices in $[n-1]$, uniformly distributed over all $(n-1)(n-2)$ such pairs. Then $\Sigma = \Sigma(B, U_1, \ldots, U_{n-1}, I, J)$ is defined as follows; see Figure 2.3. (We index the n columns and rows from 0 to $n-1$.)

• $\Sigma_{i,i} = 1$ for all $i = 0, \ldots, n-1$.

• Regardless of B, I, J, we have $\Sigma_{0,i} = U_i$ for all $i = 1, \ldots, n-1$.

• If $B = 0$, then $\Sigma_{i,j} = U_i U_j$ for all $i, j \in \{1, \ldots, n-1\}$, $i \neq j$. Note that in this case, by Lemma 2.1.3, the concentration graph is a star with vertex 0 as a center (and therefore indeed $\Sigma \in \mathcal{X}$).

• If $B = 1$, then $\Sigma_{i,j} = U_i U_j$ for all $i, j \in \{1, \ldots, n-1\}$ such that $i \neq j$ and $\{i, j\} \neq \{I, J\}$. Moreover, $\Sigma_{I,J} = \min(U_I, U_J) / \max(U_I, U_J)$. In this case, again by Lemma 2.1.3, the concentration graph is a tree in which vertex 0 has degree $n-2$, every vertex $i \notin \{I, J\}$ has degree 1 and is attached to vertex 0, and vertices $0, I, J$ form a path such that, if $U_I < U_J$, then J is the middle vertex and if $U_I > U_J$, then I is the middle vertex.

Regardless of what the algorithm A is, it is unable to distinguish between $\Sigma(0, U_1, \ldots, U_{n-1}, I, J)$ and $\Sigma(1, U_1, \ldots, U_{n-1}, I, J)$ before the entry $\Sigma_{I,J}$ is queried. (No other entry of Σ provides any information about $\Sigma_{I,J}$.) In other words, if B,

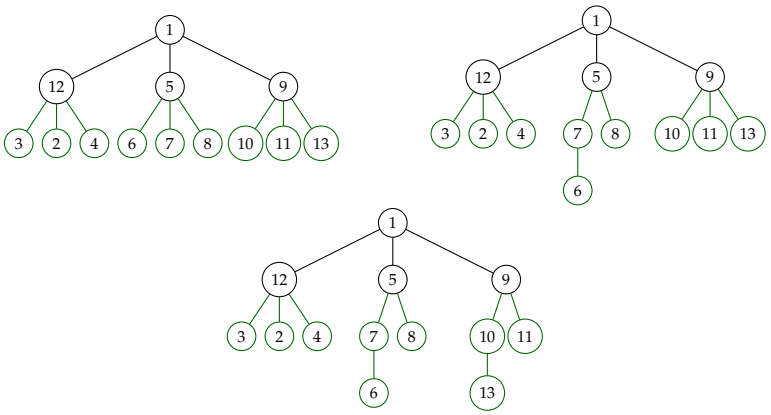


Figure 2.4: Three trees made from a ternary tree, by removing the parent edge of at most one leaf from each 3-ary branch and attaching it to another leaf of the same branch.

$U_1, \ldots, U_{n-1}$, and I, J are fixed and $\Sigma = \Sigma(B, U_1, \ldots, U_{n-1}, I, J)$, then

$$P(A, \Sigma) \;\geq\; \frac{1}{2}\,\mathbb{E}\left(\mathbb{1}_{(I,J) \text{ is not queried}} | (I, J)\right).$$

Thus, for any algorithm A,

$$\begin{aligned}
\mathbb{E}_{\Sigma\sim\mathcal{D}} P(A, \Sigma) &\geq \frac{1}{2\binom{n-1}{2}} \sum_{\{i,j\}\subset[n-1]:i\neq j} \mathbb{E}\left(\mathbb{1}_{(i,j) \text{ is not queried}} | (I, J) = (i, j)\right) \\
&\geq \frac{\binom{n-1}{2} - k}{2\binom{n-1}{2}} \quad \text{(by symmetry)}
\end{aligned}$$

proving the theorem. ∎

For the class of covariance matrices $\mathcal{X}_d$ whose concentration graph is a tree with maximum degree bounded by d, we have the following bound. Its proof is similar to that of Theorem 2.2.11. To avoid repetitions, we do not detail the proof. The only difference is that the class of trees that support the distribution $\mathcal{D}$ now includes the complete d-ary tree of height h of $n = (d^{h+1} - 1)/(d - 1)$ vertices and its modifications such that, in each d-ary branch at the leaf level, one can remove a leaf and attach it to another one of the same branch (see Figure 2.4 for such instances, made from a ternary tree).

Theorem 2.2.12.

$$R(\mathcal{A}_k, \mathcal{X}_d) \geq \frac{1}{2}\,(1 - o(1))$$

whenever $k = o(nd)$.

2.3 Separators in bounded treewidth graphs

In the next two sections, we deal with the main results of the paper. We show that a large and important class of sparse concentration graphs can be reconstructed efficiently with $O(n \, polylog \, n)$ correlation queries. The class includes all graphs with bounded treewidth and bounded maximum degree.

The algorithm we propose is a divide-and-conquer algorithm. The main idea is that, once one finds a small set of vertices (a separator) whose removal decomposes the graph into small connected components, and these components are identified, one may recurse in these components. The nontrivial task is to find such separators efficiently.

Our algorithm starts by taking a random sample W of the vertices, of size proportional to the treewidth of G. Then we find a separator of W of size at most $k+1$ that splits the vertices of W into two sets of comparable size. We argue that, with high probability, such a separator exists. We also prove that this separator is a balanced separator of the entire vertex set. Removal of this separator decomposes the graph into connected components of significantly reduced size. We identify these components using a linear number of queries. Then the algorithm recurses into each of the components. In this section we discuss the first splitting of the graph. How to subsequently recurse into the smaller subsets is described in detail in Section 2.4.

A *tree decomposition* of a graph $G(V, E)$ is a tree T with vertices $B_1, \ldots, B_m$, where $B_i \subseteq V$ satisfy

1. The union of all sets B_i equals V.
2. If B_i and B_j both contain v, then all vertices B_k of T in the unique path between B_i and B_j contain v as well.
3. For every edge uv in G, there is B_i that contains both u and v.

The *width* of a tree decomposition is the size of its largest set B_i minus one. The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum width among all possible tree decompositions of G.

A key property of bounded-treewidth graphs is that they have small "balanced" separators. To precisely define a balanced separator, we generalize the notion of

centrality (2.2.1) for any set $S \subset V$ by writing

$$c(S) \stackrel{\text{def.}}{=} \frac{1}{|V \setminus S|} \max_{C \in \mathcal{C}^S} |C| \,, \tag{2.3.1}$$

where recall that $\mathcal{C}^S$ is the set of connected components of the graph induced by $V \setminus S$. We will call a separator S *balanced* when $c(S) \leq 0.93$. We start by noting that every graph with bounded treewidth has a small balanced separator.

Proposition 2.3.1. (see e.g. Bodlaender [11, Theorem 19]) *If* $\mathrm{tw}(G) \leq k$ *then* G *has a separator* S *such that* $|S| \leq k+1$ *and*

$$c(S) \leq \frac{1}{2} \cdot \frac{|V| - k}{|V| - (k+1)} \,.$$

Remark. By [11, Lemma 11], if $\mathrm{tw}(G) \leq k$ then $\mathrm{tw}(H) \leq k$ for every subgraph H of G. In particular, one can recursively split a graph into subgraphs of small treewidth using small balanced separators.

2.3.1 Finding a separator of two sets

Let $G(V, E)$ be a graph and let $\Sigma \in \mathcal{M}(G)$ be generic. We give an algorithm that finds a minimal separator of two subsets $A, B \subset V$. By Assumption 2, the size of such a minimal separator is $r \stackrel{\text{def.}}{=} \mathrm{rank}(\Sigma_{A,B})$. Denote by $\mathcal{S}(A, B)$ the set of all minimal separators of A and B in G. Denote by U the set of all vertices that lie in some minimal separator in $\mathcal{S}(A, B)$.

Lemma 2.3.2. *A vertex* $v \in V$ *lies in* U *if and only if* $\mathrm{rank}(\Sigma_{Av,Bv}) = r$.

Proof. This follows immediately from Lemma 2.1.2. ∎

Lemma 2.3.2 together with Lemma 2.1.2 give a simple and efficient procedure to find an element in $\mathcal{S}(A, B)$, detailed in Algorithm 6.

Proposition 2.3.3. *Let* $G(V, E)$ *be a graph and let* $\Sigma \in \mathcal{M}(G)$ *be generic. For any* $A, B \subset V$ *with* $M = \max\{|A|, |B|\}$, *Algorithm 6 finds a minimal separator of* A *and* B *with query complexity* $\mathcal{O}(|V|M^2)$ *and computational complexity* $\mathcal{O}(|V|M^3)$.

Proof. By Lemma 2.3.2, the first loop finds the set U of all vertices that lie in a minimal separator of A and B. This loop has $\mathcal{O}(|V|M^2)$ and $\mathcal{O}(|V|M^3)$ query and computational complexity, respectively.

Algorithm 6: `ABSeparator`(A, B)

$U := \emptyset$;
$r := \text{rank}(\Sigma_{A,B})$;
forall $v \in V$ **do**
  if $\text{rank}(\Sigma_{Av,Bv}) = r$ **then**
    $U \leftarrow U \cup \{v\}$;
$C \leftarrow \{v_0\}$ for an arbitrary $v_0 \in U$;
forall $u \in U \setminus \{v_0\}$ **do**
  if $\text{rank}(\Sigma_{ACu,BCu}) = r$ **then**
    $C \leftarrow C \cup \{u\}$;
return C;

We now take an arbitrary vertex $v_0 \in U$ and show that the second loop of the algorithm finds a minimal separator of A and B that contains v_0. Start with $C = \{v_0\}$ and note that, since $v_0 \in U$, there exists $S \in \mathcal{S}(A, B)$ containing v_0. In each iteration of the second loop we add $u \in U \setminus \{v_0\}$ to C if $\text{rank}(\Sigma_{ACu,BCu}) = r$. Therefore, by Lemma 2.1.2, we update $C \leftarrow C \cup \{u\}$ if and only if there exists (not necessarily unique) $S \in \mathcal{S}(A, B)$ such that $C \cup \{u\} \subseteq S$. By Assumption 2, $|S| = r$ and so, if $|C \cup \{u\}| = r$ then $S = C \cup \{u\}$ and the rank condition will *not* be satisfied for the subsequent vertices in the loop (showing correctness of the algorithm). If $|C \cup \{u\}| < r$ then $C \cup \{u\}$ is a strict subset of S and all the remaining vertices in $S \setminus (C \cup \{u\})$ appear in the second loop *after* u. We conclude correctness of the algorithm.

Since $|U| \leq |V|$ and $r \leq M$, the number of queries and computational complexity of the second loop are $\mathcal{O}(|V|M^2)$ and $\mathcal{O}(|V|M^3)$, respectively, which concludes the proof. ∎

2.3.2 Balanced separators in G

In Section 2.3.1 we provided an efficient procedure that finds a separator for a given pair of sets $A, B \subset V$. In this section we show how to construct such a pair of small sets so that the obtained separator is, with high probability, a balanced separator for the entire graph G.

Our approach to finding a balanced separator is to base the search on a random subset $W \subset V$ of size m which can be handled within our computational budget. To argue why our randomization works and guide the choice of the parameter m we rely on VC-theory initiated by Vapnik and Chervonenkis [89]. Let $\mathcal{F}_S$ be the set

of all connected components in $\mathcal{C}^S$ and their complements in $V \setminus S$. Let

$$\mathcal{F}_k \stackrel{\text{def.}}{=} \bigcup_{S:|S|\leq k} \mathcal{F}_S \,, \tag{2.3.2}$$

that is, $C \in \mathcal{F}_k$ if it is a connected component of $G \setminus S$ for some S with at most k elements, or C is a union of all but one such components.

Definition 2.3.4. We will call a set $W \subseteq V$ a *δ-sample* for $\mathcal{F}_k$ if for all sets $C \in \mathcal{F}_k$,

$$\frac{|C|}{|V|} - \delta \;\leq\; \frac{|W \cap C|}{|W|} \;\leq\; \frac{|C|}{|V|} + \delta \,. \tag{2.3.3}$$

We now present conditions that assure that a uniformly random sample W from the vertex set V is a δ-sample with high probability. A subset $W \subset V$ is *shattered* by $\mathcal{F}_k$ if $W \cap \mathcal{F}_k \;=\; \{W \cap C : \; C \in \mathcal{F}_k\}$ is the set of all subsets of W. Define the *VC-dimension* of $\mathcal{F}_k$, denoted by $\mathrm{VC}(\mathcal{F}_k)$, to be the maximal size of a subset shattered by $\mathcal{F}_k$. The following follows from the classical Vapnik-Chervonenkis inequality (see Devroye and Lugosi [28] for a version that implies the constants shown here):

Theorem 2.3.5. *Suppose that* $\mathrm{VC}(\mathcal{F}_k) = r$, $\delta > 0$, *and* $\tau \leq 1/2$. *A set W obtained by sampling m vertices from V uniformly at random, with replacement, is a δ-sample of $\mathcal{F}_k$ with probability at least* $1 - \tau$ *if*

$$m \;\geq\; \max\left(\frac{10r}{\delta^2}\log\left(\frac{8r}{\delta^2}\right), \frac{2}{\delta^2}\log\left(\frac{2}{\tau}\right)\right) \,. \tag{2.3.4}$$

A key property of the set $\mathcal{F}_k$ is that its VC-dimension is bounded by a linear function of the treewidth k.

Lemma 2.3.6 (Feige and Mahdian [38])**.** *Let $G(V, E)$ be a graph and, for a fixed k, let $\mathcal{F}_k$ for be the set defined in (2.3.2). Then* $\mathrm{VC}(\mathcal{F}_k) \leq 11 \cdot k$.

Remark. The statement of this lemma in Feige and Mahdian [38] uses a universal constant. Their proof however allows one to specify this constant to be 11.

We now show that a δ-sample admits a balanced separator.

Proposition 2.3.7. *Let $G(V, E)$ be such that* $\mathrm{tw}(G) \leq k$, *and let $W \subset V$ be a δ-sample of $\mathcal{F}_{k+1}$ satisfying $|W| \geq 6(k+1)$. If $\delta \leq \frac{1}{24}$, then we can partition W into two sets A, B such that $|A|, |B| \leq \frac{2|W|}{3}$ and a minimal separator S of A and B has at most $k+1$ elements. Moreover, for any such partition,* $\max(|A \setminus S|, |B \setminus S|) \leq \frac{4}{5}|W \setminus S|$.

In the latter proposition we refer to $\mathcal{F}_{k+1}$ and not $\mathcal{F}_k$, in order to be consistent with Proposition 2.3.1.
Before we prove this result we formulate two useful lemmas.

Lemma 2.3.8. *Let $U = \bigcup_{i=1}^{d} C_i$ be a partition of a set U into disjoint sets such that $|C_1| \geq \cdots \geq |C_d| > 0$. If $|C_1| \leq \alpha|U|$ for $\alpha \geq \frac{2}{3}$, then there exists $t \in \{1, \ldots, d\}$ such that*

$$(1-\alpha)|U| \leq \sum_{i=1}^{t} |C_i| \leq \alpha|U| \,.$$

Proof. Obviously there exists t such that the upper bound is true. Let us consider the largest such t. Then $\sum_{i=t+1}^{d} |C_i| \leq \frac{|U|}{3}$. If also $\sum_{i=1}^{t-1} |C_i| < \frac{|U|}{3}$, then $|C_t| > \frac{|U|}{3}$. But the latter implies $\sum_{i=1}^{t-1} |C_i| > \frac{|U|}{3}$, by the ordering of the $|C_i|$, a contradiction. ∎

For the second lemma, let

$$\lambda \stackrel{\text{def.}}{=} 1 - \frac{k+1}{|V|} \tag{2.3.5}$$

and note that $\frac{|V \setminus S|}{|V|} \geq \lambda$ for all S such that $|S| \leq k+1$.

Lemma 2.3.9. *Suppose $W \subset V$ is a δ-sample for $\mathcal{F}_{k+1}$ and let $S \subset V$ with $|S| \leq k+1$. If $C \in \mathcal{C}^S$, then*

$$\frac{\lambda \cdot c(S) - \delta}{\lambda + 2\delta} \;\leq\; \frac{|W \cap C|}{|W \setminus S|} \;\leq\; \frac{\lambda \cdot c(S) + \delta}{\lambda - 2\delta}. \tag{2.3.6}$$

Proof. Using (2.3.3), we get

$$\frac{|W \cap C|}{|W \setminus S|} \;\leq\; \left(\frac{|C|}{|V|} + \delta\right) \cdot \frac{|W|}{|W \setminus S|} \;\leq\; \left(c(S) \cdot \frac{|V \setminus S|}{|V|} + \delta\right) \cdot \frac{|W|}{|W \setminus S|} \,.$$

To bound the last expression, let $\bar{C} = V \setminus (C \cup S)$. Since $C, \bar{C} \in \mathcal{F}_{k+1}$, we get

$$\frac{|W \setminus S|}{|W|} \;=\; \frac{|C \cap W|}{|W|} + \frac{|\bar{C} \cap W|}{|W|} \;\geq\; \left(\frac{|C|}{|V|} + \frac{|\bar{C}|}{|V|} - 2\delta\right) \;=\; \frac{|V \setminus S|}{|V|} - 2\delta \,.$$

A similar argument gives an upper bound for $\frac{|W \setminus S|}{|W|}$, which after taking reciprocals gives

$$\frac{1}{\frac{|V \setminus S|}{|V|} + 2\delta} \;\leq\; \frac{|W|}{|W \setminus S|} \;\leq\; \frac{1}{\frac{|V \setminus S|}{|V|} - 2\delta} \,. \tag{2.3.7}$$

This gives the upper bound in (2.3.6) because

$$\frac{|W \cap C|}{|W \setminus S|} \;\leq\; \frac{c(S)\frac{|V\setminus S|}{|V|} + \delta}{\frac{|V\setminus S|}{|V|} - 2\delta} \;\leq\; \frac{\lambda c(S) + \delta}{\lambda - 2\delta},$$

where the last inequality follows by the fact that the middle expression is a decreasing function of $\frac{|V\setminus S|}{|V|}$ and $\frac{|V\setminus S|}{|V|} \geq \lambda$. This establishes the upper bound in (2.3.6). The lower bound follows by similar arguments. ∎

Proof. [Proof of Proposition 2.3.7] Let S^* be a minimizer of $c(S)$ among all $S \subset V$ such that $|S| \leq k+1$. By Proposition 2.3.1, $c(S^*) \leq \frac{1}{2} \cdot \frac{|V|-k}{|V|-(k+1)}$, which is further bounded by $11/20$ if $|V| \geq 6(k+1)$. By Lemma 2.3.9, if $C \in \mathcal{C}^{S^*}$ then

$$\frac{|W \cap C|}{|W \setminus S^*|} \;\leq\; \frac{\frac{11}{20}\lambda + \delta}{\lambda - 2\delta}.$$

The right-hand side is an increasing function of δ and the maximum for $\delta \leq \frac{1}{24}$ is $(\frac{11}{20}\lambda + \frac{1}{24})/(\lambda - \frac{1}{12})$, which is bounded by $\frac{2}{3}$ because $\lambda \geq \frac{5}{6}$ (recall the definition of λ in equation (2.3.5) and that $|V| \geq |W| \geq 6(k+1)$). This shows that $W \setminus S^*$ can be partitioned into disjoint subsets $W \cap C$ for $C \in \mathcal{C}^{S^*}$, all of size at most $\frac{2}{3}|W \setminus S^*|$. By Lemma 2.3.8, we can group these sets into two groups A' and B' where each has size at most $\frac{2}{3}|W \setminus S^*|$. To show the first claim let A, B be any two sets partitioning W that satisfy $A \setminus S^* = A'$, $B \setminus S^* = B'$. We next show that there is a choice of A, B that gives $\max(|A|, |B|) \leq \frac{2}{3}|W|$. This is done by allocating the elements of $W \cap S^*$ in a balanced way between A' and B' so that both A' and B' get at most $\frac{2}{3}$ of the elements in $W \cap S^*$. This can be always done if $W \cap S^*$ has at least two elements. If $W \cap S^*$ is empty, the statement is trivial. If $|W \cap S^*| = 1$ we consider two cases (i) $|A'| < |B'|$ and (ii) $|A'| = |B'|$. In case (i) we allocate the element in $W \cap S^*$ to A'. In that case

$$|A| \;=\; |A'| + 1 \;\leq\; |B'| \;=\; |B| \;\leq\; \frac{2}{3}|W \setminus S^*| \;\leq\; \frac{2}{3}|W|.$$

In case (ii), we again allocate the element in $W \cap S^*$ to A' and use the fact that $|A'| = |B'| = \frac{1}{2}|W \setminus S^*|$, which gives

$$|B| \;\leq\; |A| \;=\; |A'| + 1 \;=\; \frac{1}{2}|W \setminus S^*| + 1 \;=\; \frac{1}{2}|W| + \frac{1}{2} \;\leq\; \frac{2}{3}|W|,$$

where the last inequality holds always if $|W| \geq 3$. This proves the first claim.

To show the second claim, assume $\max(|A|,|B|) \leq \frac{2}{3}|W|$. Since $|W| \geq 6(k+1)$ we have $|W \setminus S| \geq 5(k+1) \geq 5|W \cap S|$. Now

$$\max(|A \setminus S|, |B \setminus S|) \leq \frac{2}{3}|W| \leq \frac{2}{3}(|W \setminus S| + |W \cap S|) \leq \frac{2}{3}\left(1+\frac{1}{5}\right)|W \setminus S|,$$

which completes the argument. ∎

2.3.3 Separating and splitting

We now present the procedure `Separator` that finds a balanced separator in G. When the separator is found, decomposing the graph into connected components is straightforward and is given in the procedure `Components`.

The algorithm starts by choosing a sample $W \subset V$. Then it looks for a partition of W into two sets A, B such that $|A|, |B| \leq \frac{2}{3}|W|$ and the rank of $\Sigma_{A,B}$ is small. In Proposition 2.3.10 we argue why such a partition exists with high probability. Then the algorithm uses ideas of Section 2.3.1 to efficiently find a minimal separator S of A and B in G; by construction $|S| = \operatorname{rank}(\Sigma_{A,B})$. At this moment a purely deterministic part of the process begins. Given S, the algorithm decomposes V into connected components in $\mathcal{C}^S$. This is done using rank conditions like in the tree-like case.

Algorithm 7: `Separator`

Pick a set W by taking m vertices uniformly at random, where m satisfies (2.3.4) with $r = 11k$ and $\delta = 1/24$;
Search exhaustively through all partitions of W into sets A, B with $|A|, |B| \leq \frac{2}{3}|W|$, minimizing $\operatorname{rank}(\Sigma_{A,B})$;
If no such balanced partition exists, output any partition A, B of W;
$S \leftarrow$ `ABSeparator`(A, B);
return S

The next proposition shows that Algorithm 7 outputs a balanced separator with high probability.

Proposition 2.3.10. *Let $G(V,E)$ be a graph with $\operatorname{tw}(G) \leq k$, and $|V| \geq 6(k+1)$ vertices and let $\Sigma \in \mathcal{M}(G)$ be generic. Let $\tau \in (0,1)$. Then, with probability at least $1-\tau$, Algorithm 7 finds a separator S in G such that $|S| \leq k+1$ and $|C| \leq 0.93|V|$ for each connected component $C \in \mathcal{C}^S$.*

Proof. The procedure starts by choosing a sample $W \subset V$. The size of the sample m is chosen so that, with probability at least $1-\tau$, W is a δ-sample for $\mathcal{F}_{k+1}$. A

Algorithm 8: `Components(V)`

$S \leftarrow$ `Separator(V)`;
$r \leftarrow |S|$;
$R \leftarrow \emptyset$; `// will contain one vertex from each` $C \in \mathcal{C}^S$
for $v \in V \setminus S$ **do**
  $notFound \leftarrow True$;
  for $u \in R$ **do**
    if $\operatorname{rank}(\Sigma_{uS,vS}) = r + 1$ **then**
      $C_u \leftarrow C_u \cup \{v\}$;
      $notFound \leftarrow False$;
  if $notFound$ **then**
    $C_v \leftarrow \{v\}$; `// create new component` $C^v \in \mathcal{C}^S$
    $R \leftarrow R \cup \{v\}$;
return S and all C_v for $v \in R$;

sufficient condition for m follows by Theorem 2.3.5 and Lemma 2.3.6. Note that this condition also assures that $|W| \geq 6(k+1)$.

Since $\delta = 1/24$, by Proposition 2.3.7, we can partition W into two sets A, B such that $|A|, |B| \leq \frac{2|W|}{3}$ and any minimal separator S of A and B has at most $k+1$ elements. Moreover, for any such partition $|A \setminus S|, |B \setminus S| \leq \frac{4|W \setminus S|}{5}$. Now we only need to show that for each connected component $C \in \mathcal{C}^S$ of G, $|C| \leq \frac{93}{100}|V|$. Indeed, if C^* is the maximal component in $\mathcal{C}^S$ then $\frac{|W \cap C^*|}{|W \setminus S|} \leq \frac{4}{5}$. Since C^* lies in $\mathcal{F}_{k+1}$, we get

$$\frac{|C^* \cap W|}{|W \setminus S|} \geq \left(\frac{|C^*|}{|V \setminus S|} \frac{|V \setminus S|}{|V|} - \delta \right) \frac{|W|}{|W \setminus S|} \overset{(2.3.7)}{\geq} \frac{c(S)\frac{|V \setminus S|}{|V|} - \delta}{\frac{|V \setminus S|}{|V|} + 2\delta}.$$

The expression on the right-hand side is an increasing function of $\frac{|V \setminus S|}{|V|}$ and $\frac{|V \setminus S|}{|V|} \geq \lambda$, which gives that

$$\frac{|C^* \cap W|}{|W \setminus S|} \geq \frac{\lambda \frac{|C^*|}{|V \setminus S|} - \delta}{\lambda + 2\delta}$$

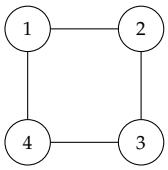
Since $\frac{|C^* \cap W|}{|W \setminus S|} \leq \frac{4}{5}$, $\delta \leq \frac{1}{24}$, and $\lambda \geq \frac{5}{6}$, we get that $\frac{|C^*|}{|V \setminus S|} \leq \frac{93}{100}$.

■

2.4 Recovery of bounded treewidth graphs

In this section we present an algorithm for reconstructing graphs with bounded treewidth. Let $G([n], E)$ be a graph and let $\Sigma \in \mathcal{M}(G)$ be generic. To recover G from Σ we follow a similar divide-and-conquer strategy as in the previous sections. First, a balanced separator S is chosen and then the algorithm recurses into all the components in $\mathcal{C}^S$. There is however a complication. If $C \in \mathcal{C}^S$, then $G[C \cup S]$ is not equal to the graph of Σ_{CS}, unless for *each* $C \in \mathcal{C}^S$ the set of vertices in S that are linked to C by an edge is a clique of G; see Frydenberg [39, Theorem 3.3]. The condition holds, in particular, when S is a clique, but in our case there is no way to assure that in general. If it does not hold, then the graph of $\mathcal{G}(\Sigma_{CS})$ is strictly bigger than the subgraph $G[C \cup S]$. The next example illustrates this phenomenon.

Example. Consider the four-cycle given below together with the corresponding covariance and precision matrices



$$\Sigma = \begin{bmatrix} 7 & -2 & 1 & -2 \\ -2 & 7 & -2 & 1 \\ 1 & -2 & 7 & -2 \\ -2 & 1 & -2 & 7 \end{bmatrix} \qquad K = \frac{1}{24}\begin{bmatrix} 4 & 1 & 0 & 1 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 1 & 0 & 1 & 4 \end{bmatrix}$$

Then $\{1, 3\}$ separates 2 and 4 but the graph $\mathcal{G}(\Sigma_{123})$ is the complete graph over $\{1, 2, 3\}$ because

$$\begin{bmatrix} 7 & -2 & 1 \\ -2 & 7 & -2 \\ 1 & -2 & 7 \end{bmatrix}^{-1} = \frac{1}{96}\begin{bmatrix} 15 & 4 & -1 \\ 4 & 16 & 4 \\ -1 & 4 & 15 \end{bmatrix}.$$

An easy way around this problem is by noting that Gaussian graphical models are closed under conditioning and this probabilistic statement has a useful algebraic counterpart. The graph of the conditional covariance is obtained from G by removing the vertices in the conditioning set and all the incident edges. This means that the edges in $G[C]$ can be recovered from the conditional covariance matrix

$$\Sigma_{C|S} \overset{\text{def.}}{=} \Sigma_{C,C} - \Sigma_{C,S}\Sigma_{S,S}^{-1}\Sigma_{S,C}. \tag{2.4.1}$$

More concretely, we have the following basic result.

Lemma 2.4.1. *If S separates C from the rest of the graph, then $K_{C,C} = (\Sigma_{C|S})^{-1}$.*

This result can be argued by standard properties of the Gaussian distribution, as K_C is the inverse of the conditional covariance matrix $\Sigma_{C|[n]\setminus C}$; see, for example, equation (C.3) in Lauritzen [60]. If S separates C from the remaining vertices then, by conditional independence, this conditional covariance matrix is equal to $\Sigma_{C|S}$. We should stress however that Lemma 2.4.1 holds without assuming Gaussianity; the statement is purely algebraic.

This result shows that in order to keep information about the induced subgraph $G[C]$, it is important to keep the information about the separating set. To see how this is done, it is helpful to study the situation in Figure 2.5. Suppose that S separates G into several components one of which is C. We then recurse our algorithm on C by conditioning on S. In the next step we use the matrix $\Sigma_{C|S}$ to find a balanced separator S' of $G[C]$. We then recurse on the corresponding components C_1, C_2, C_3, C_4. Note that in the next step it is not enough to condition on S' to study $G[C_2]$ because it is connected to the rest of the graph through S. Therefore, in this recursive call we need to work with the conditional covariance matrix $\Sigma_{C_2|SS'}$.

The dependence on separating sets requires a modification of the algorithms that we use to decompose the graph. Instead of working on the covariance matrix, they should be working on the conditional covariance matrix. Note however, that rank queries for $\Sigma_{A,B|S}$ with $A, B \subset C$ are equivalent to rank queries on $\Sigma_{AS,BS}$. Indeed, by the Guttman rank additivity formula (see e.g. [93, Section 0.9])

$$\operatorname{rank}(\Sigma_{AS,BS}) \;=\; \operatorname{rank}(\Sigma_S) + \operatorname{rank}(\Sigma_{A,B|S}) \;=\; |S| + \operatorname{rank}(\Sigma_{A,B|S}). \tag{2.4.2}$$

Therefore, the algorithms $\texttt{ABSeparator}(A, B)$, $\texttt{Separator}$, $\texttt{Components}$ have their simple modifications $\texttt{ABSeparator}(V, A, B, S)$, $\texttt{Separator}(V, S)$, $\texttt{Components}(V, S)$, where a set S disjoint from V is added to both the row and the column set in all the rank queries. For completeness we explicitly provide these algorithms in Section 2.5.

In Algorithm 9 we present the complete algorithm which relies on routine $\texttt{Reconstruct}$, which is then called recursively in Algorithm 10. With a fixed bound k on the treewidth of G, the main algorithm returns the precision matrix $K = \Sigma^{-1}$.

Algorithm 9: *Main algorithm*

$\widehat{K} \leftarrow 0 \in \mathbb{R}^{n \times n}$
Fix m satisfying (2.3.4) with $r = 11k$ and $\delta = \frac{1}{24}$
$\texttt{Reconstruct}([\texttt{n}], \emptyset)$

At each call of $\texttt{Reconstruct}(V, S)$, if the input vertex set V is larger than the fixed threshold m, then $\texttt{Separator}(V)$ (called during the procedure $\texttt{Components}$)

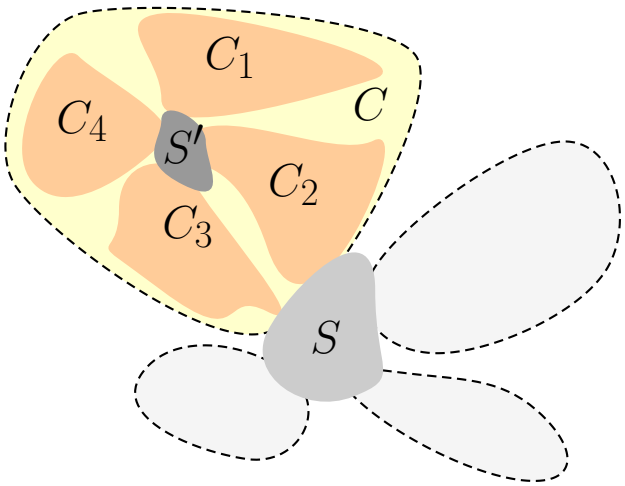


Figure 2.5: Components found during subsequent recursive calls to `Reconstruct`. The algorithm needs to keep track of the separators found at all levels, as C_2 and C_3 are connected by both S' and S.

finds a balanced separator S' of $G[V]$. Then the procedure `Components` finds all connected components C_i in $G[V] \setminus S'$. Subsequently, `Reconstruct` recurses in all these components replacing S with $S \cup S'$ as in Figure 2.5.

Most edges in K are only reconstructed in the final recursive calls. Consider the situation in Figure 2.5. Suppose that `Reconstruct`(C, S) makes a recursive call to `Reconstruct`$(C_1, S \cup S')$. If $|C_1| \leq m$, then `Reconstruct`$(C_1, S \cup S')$ computes K_{C_1}, which by Lemma 2.4.1 is equal to the inverse of $\Sigma_{C_1|SS'}$. The matrices $K_{C_1S'}$ and $K_{S'S'}$ can be computed in a similar way as described in the lemma below.

Lemma 2.4.2. *Suppose $C \in \mathcal{C}^S$ and that C is further decomposed into S' and the connected components $\{C_1, \ldots, C_d\}$ (as in Figure 2.5). Let K denote Σ^{-1}. The submatrix K_C has a block structure with $K_{C_i,C_j} = 0$ for $i \neq j$ and*

$$K_{C_i,S'} = -K_{C_i}\Sigma_{C_i,S'|S}\Sigma^{-1}_{S'|S}, \qquad K_{S'} = \left(\mathbb{I}_{|S'|} - \sum_{i=1}^{d} K_{S',C_i}\Sigma_{C_i,S'|S}\right)\Sigma^{-1}_{S'|S},$$

where $\mathbb{I}_m$ denotes the $m \times m$ identity matrix.

Proof. There are no direct links between C_i and C_j in G and so $K_{C_i,C_j} = 0$ for $i \neq j$. Lemma 2.4.1 gives the identity $K_C\Sigma_{C|S} = \mathbb{I}_{|C|}$. Taking the C_i-rows of K_C and the S'-columns of $\Sigma_{C|S}$ we get from this identity that

$$K_{C_i}\Sigma_{C_i,S'|S} + K_{C_i,S'}\Sigma_{S',S'|S} \;=\; 0,$$

which implies the first equality. Taking the S'-rows and S'-columns we get

$$\sum_{i=1}^{d} K_{S',C_i}\Sigma_{C_i,S'|S} + K_{S'}\Sigma_{S'|S} \;=\; \mathbb{I}_{|S'|},$$

which implies the second formula. ∎

Algorithm 10: `Reconstruct`(V, S)

```
if |V| > m then
    C_1, ..., C_d, S' ← Components(V, S);          // see Section 2.5
    for i from 1 to d do
        K̂_{C_i} ← Reconstruct(C_i, S ∪ S');
        K̂_{C_i,S'} ← −K̂_{C_i} Σ_{C_i,S'|S} Σ^{-1}_{S'|S}
    K̂_{S'} ← (I_{|S'|} − Σ_{i=1}^{d} K̂_{S',C_i} Σ_{C_i,S'|S}) Σ^{-1}_{S'|S};
else
    return K̂_V
```

Theorem 2.4.3. `Reconstruct`$([n], \emptyset)$ *correctly recovers* K.

Proof. The correctness of `Components`$([n], \emptyset)$ was already shown in Proposition 2.3.10. Using the discussion given in the beginning of this section, we can easily adjust this proof for any call of `Components`(V, S). This, together with Lemma 2.4.1, implies that in each call of `Components`(V, S):

(i) If u, v are put in different components components then $K_{uv} = 0$.

(ii) It holds that $K_{C_i} = (\Sigma_{C_i|SS'})^{-1}$.

By (i) all (zero) entries in K_{C_i,C_j} are correctly recovered. By (ii) all the entries of K_{C_i} obtained by inverting $\Sigma_{C_i|SS'}$ are correct. By Lemma 2.4.2, also $K_{C_i,S'}$ and $K_{S',S'}$ are correctly recovered. ∎

Theorem 2.4.4. *Let $G([n], E)$ be a graph with treewidth $\mathrm{tw}(G) \leq k$ and maximum degree $\Delta(G) \leq d$. Let $\Sigma \in \mathcal{M}(G)$ be generic with $G(\Sigma)$ connected and let m in* `Separator` *satisfy (2.3.4) with $r \geq 11k$, $\delta \leq \frac{1}{24}$, and $\tau \leq \frac{1}{3}$. Then, with probability at least $1 - \frac{1}{n^8}$, the query complexity of* `Reconstruct` *is of the order*

$$\mathcal{O}\left((2^{\mathcal{O}(k \log k)} + dk \log n)k^2 n \log^3 n\right),$$

and the time complexity is of the order

$$\mathcal{O}\left((2^{\mathcal{O}(k\log k)} + dk\log n)k^3 n\log^4 n\right).$$

Before we prove this result, a number of remarks are in order.

Remark. The choice of $\tau \leq 1/3$ in Theorem 2.4.4 is arbitrary; any other choice would only change the constant factors in the complexity bounds and the probability of exceeding the stated query and time complexity. Moreover, we note that the total probability of error can be made arbitrarily small. In the following proof we show that the recursion depth is $\mathcal{O}(\log n)$ with high probability. Then, executing the algorithm $\mathcal{O}(\log \frac{1}{\epsilon})$ times (each time stopping if it does not finish in the time limit stated in Theorem 2.4.4) we get at least one timely execution of the algorithm with probability $1-\epsilon$, regardless of n. The complexity only changes by a factor of $\mathcal{O}(\log \frac{1}{\epsilon})$.

Remark. As it is seen from the proof, the assumption of bounded degree can be relaxed. It can be substituted by the assumption that removing $\mathcal{O}(k\log n)$ vertices decomposes the graph into at most a polylogarithmic number of connected components. In other words, one may weaken the bounded-degree assumption by suitable assumptions on the *fragmentation* of the graph. We refer to Hajiaghayi and Hajiaghayi [41] and Hajiaghayi and Nishimura [42] for more information on this notion. An example of a graph with unbounded degree for which our reconstruction method works is the wheel graph (i.e., the graph formed by connecting a central vertex to all vertices of a cycle of $n-1$ vertices). This graph has treewidth 3, maximum degree $n-1$ but low fragmentation.

Remark. The problem of computing the treewidth of a graph given its adjacency matrix is NP-hard Arnborg, Corneil, and Proskurowski [4] (see Bodlaender, Drange, Dregi, Fomin, Lokshtanov, and Pilipczuk [12] for an account on the history of the problem and references on current results). Hence it is not surprising to have an exponential dependence on treewidth. Note however that in our case we do not have access to edge queries but to separation queries, in contrast to the traditional setting. We are not aware of algorithmic results in this setting. We expect that similar hardness results should hold but proving such hardness seems non-trivial and outside of the scope of this study.

Proof. [Proof of Theorem 2.4.4] We refer to all operations in $\texttt{Reconstruct}([n],\emptyset)$, excluding operations in subsequent calls $\texttt{Reconstruct}(C_i,S')$, as the zeroth recursion level of the algorithm. Similarly, the operations of all $\texttt{Reconstruct}(C_i,S')$ for $i=1,\ldots,d$ apart from their subsequent calls are called the first recursion level. We extend this definition iteratively to the t-th recursion level for $t>1$.

Assume initially that `Components` never fails, that is, $|S'| \leq k+1$ and for each connected component $C \in \mathcal{C}^S$ it holds that $|C| \leq 0.93|V|$, as stated in Proposition 2.3.10. We will bound the total probability of failure later in the proof. Since the algorithm recurses on sets C_i of size at most $0.93|V|$, the recursion depth (maximal number of recursion levels) is at most $\log_{100/93} n$, which is of order $\mathcal{O}(\log(n))$. Moreover, in each call of `Reconstruct`(V, S) always $|S| = \mathcal{O}(k \log n)$.

We start with the analysis of `Components`(V, S). Assume first that $|V| > m$ and write $s = |S|$ and $s' = |S'|$. Finding a balanced partition A, B of W in `Separator` is achieved by exhaustively searching all $< 2^m$ balanced partitions and computing the rank of the associated matrices $\Sigma_{ASS',BSS'}$, which gives query complexity $\mathcal{O}(2^m(m+s)^2)$ and time complexity $\mathcal{O}(2^m(m+s)^3)$ for this step. Taking into account that $m = \mathcal{O}(k \log k)$ and $s = \mathcal{O}(k \log n)$, we obtain complexity

$$\mathcal{O}(2^m s^2) \text{ and } \mathcal{O}(2^m s^3)$$

for queries and time, respectively. Then, given a balanced split, the procedure `ABSeparator`(V, A, B, S) finds a separator of A and B in $\mathcal{O}(|V|(m+s)^2)$ queries and $\mathcal{O}(|V|(m+s)^3)$ time. This bound can be obtained by a simple modification of Proposition 2.3.3, which gives bounds for `ABSeparator`(A, B). Hence we obtain complexity

$$\mathcal{O}(|V|s^2) \quad \text{and} \quad \mathcal{O}(|V|s^3)$$

for queries and time, respectively.

Since removing at most s' vertices of degree at most d splits the graph into at most ds' connected components, splitting V into the connected components C_i requires $\mathcal{O}(|V|s'ds)$ queries and $\mathcal{O}(|V|s'ds^3)$ time, and since $s = \mathcal{O}(k \log n)$, this gives query and time complexity bounds

$$\mathcal{O}(|V|dk^2 \log n) \text{ and } \mathcal{O}(|V|dk^4 \log^3 n).$$

In the case where $|V| \leq m$, we obtain queries and time of the order $\mathcal{O}(s^2)$ and $\mathcal{O}(s^3)$ respectively. These terms are dominated by the terms that appeared for earlier steps of the algorithm and will be ignored in what follows. Overall, *at each recursion level* this part of the algorithm requires

$$\mathcal{O}\left(n2^m k^2 \log^2 n + nk^2 \log^2 n + ndk^2 \log n\right)$$

queries and

$$\mathcal{O}(n2^m k^3 \log^3 n + nk^3 \log^3 n + ndk^4 \log^3 n)$$

time, since the vertex sets V are disjoint and there can be up to $O(n)$ calls to `Separator` at the bottom levels of recursion. Using the fact that $m = \mathcal{O}(k \log k)$ and simplifying, we obtain:

$$\mathcal{O}\left(2^{\mathcal{O}(k \log k)} k^2 n \log^2 n + dk^2 n \log^2 n\right) \tag{2.4.3}$$

queries and

$$\mathcal{O}(2^{\mathcal{O}(k \log k)} k^3 n \log^3 n + dk^4 n \log^3 n) \tag{2.4.4}$$

time.

After calling `Components`(V, S), `Reconstruct` obtains $\widehat{K}_{C_i}$ (we focus on a fixed recursion level so we can ignore the recursive call of `Reconstruct`(V, S)) and computes the matrices $\widehat{K}_{C_i,S'}$. After these matrices are computed for all components C_i, the algorithm computes $\widehat{K}_{S',S'}$. For these computations we need to calculate the conditional covariance matrices $\Sigma_{C_i,S'|S}$ and $\Sigma_{S'|S}$. The time to compute each $\Sigma_{C_i,S'|S} = \Sigma_{C_i,S'} - \Sigma_{C_i,S}\Sigma_S^{-1}\Sigma_{S,S'}$ requires $\mathcal{O}(|C_i|s + s^2)$ and $\mathcal{O}(|C_i|s^2 + s^3)$ queries and time, respectively, hence

$$\mathcal{O}(|V|s + s^2 dk \log n) \text{ and } \mathcal{O}(|V|s^2 + s^3 dk \log n),$$

for all C_i. Computing $\widehat{K}_{C_i,S'} = \widehat{K}_{C_i}\Sigma_{C_i,S'|S}\Sigma_{S'}^{-1}$ requires only $\mathcal{O}(k^2)$ additional queries (we use $s' \leq k + 1$). The time complexity of this computation is dominated by the time needed to compute $\widehat{K}_{C_i}\Sigma_{C_i,S'|S}$. A naive method of computing $\widehat{K}_{C_i}\Sigma_{C_i,S'|S}$ would require $\mathcal{O}(|C_i|^2 k)$ time, which is too time-consuming for our purposes. However, we can take advantage of the fact that the subgraph $G[C_i]$ has treewidth at most k (by the remark following Proposition 2.3.1) and the number of edges in such a graph is at most $|C_i|k$. This implies that there are at most $k|C_i|$ non-zero entries in $\widehat{K}_{C_i}$ and so the multiplication takes time $\mathcal{O}(k^2|C_i|)$. Considering all C_i, we obtain time complexity

$$\mathcal{O}(k^2|V|)$$

for this step. Finally, to compute $\widehat{K}_{S'}$ we need additional $\mathcal{O}(sk)$ queries for $\Sigma_{S,S'}$ and $\mathcal{O}(|V|k^2 + dk^3)$ time to compute $\widehat{K}_{S'}$. However, since $s = \mathcal{O}(k \log n)$, these terms are clearly dominated by complexity of the preceding steps in the algorithm and so they will be ignored in what follows. Overall, at each recursion level this part of the algorithm requires

$$\mathcal{O}\left(ns + ns^2 dk \log n\right) \qquad \text{and} \qquad \mathcal{O}(ns^2 + ns^3 dk \log n + k^2 n)$$

queries and time respectively. Using the fact that $m = \mathcal{O}(k \log k)$ and $s = \mathcal{O}(k \log n)$ and simplifying, we obtain:

$$\mathcal{O}\left(kn \log n + dk^3 n \log^3 n\right) \tag{2.4.5}$$

queries and

$$\mathcal{O}(k^2 n \log^2 n + dk^4 n \log^4 n) \tag{2.4.6}$$

time.

The total complexity for a fixed recursive level are obtained by combining (2.4.3)–(2.4.6). Taking into account the recursion depth $\mathcal{O}(\log(n))$ we get the stated overall complexity bounds.

Let I_i be the indicator variable that the i-th call of `Components` succeeds. This happens with $\mathbb{P}(I_i = 1) = 1 - \tau \geq 2/3$. Let $\alpha = 0.93$. Consider a given recursion path from the root to a leaf in the recursion tree. There are at most $\log_{1/\alpha} n$ calls with $I_i = 1$ in such a recursion path. Since the I_i are independent, we can use Hoeffding's inequality to bound the probability that we have less than $\log_{1/\alpha} n$ successes in $N = 3 \log_{1/\alpha} n$ calls:

$$\mathbb{P}\left(\sum_{i=1}^{N} I_i < \frac{1}{3}N\right) \;\leq\; e^{-\frac{2}{9}N} \;=\; n^{-\frac{2}{3\log(1/\alpha)}} \;\leq\; n^{-9}.$$

This argument implies that a fixed path from the root to a leaf in the recursion tree is logarithmic with high probability. There are at most n such paths. Hence, by the union bound and the condition for n, the probability that there exists one of them with more than logarithmic length is bounded by $1/n^8$. ∎

2.5 Modified algorithms

Here we present the modified versions of Algorithms 6, 7, and 8 that use conditional covariance information, as described in Section 2.4.

Algorithm 11: `ABSeparator`(V, A, B, S)

$U := \emptyset$;
$r = \text{rank}(\Sigma_{AS,BS})$;
forall $v \in V$ **do**
  if $\text{rank}(\Sigma_{ASv,BSv}) = r$ **then**
    $U \leftarrow U \cup \{v\}$;
$C \leftarrow \{v_0\}$ for some $v_0 \in U$;
forall $u \in U \setminus \{v_0\}$ **do**
  if $\text{rank}(\Sigma_{ASCu,BSCu}) = r$ **then**
    $C \leftarrow C \cup \{u\}$;
return C;

Algorithm 12: `Separator`(V, S)

Pick a set $W \subset V$ by taking m vertices uniformly at random, where m satisfies (2.3.4) with $r = 11k$ and $\delta = 1/24$;
Search exhaustively through all partitions of W into sets A, B with $|A|, |B| \leq \frac{2}{3}|W|$, minimizing $\text{rank}(\Sigma_{AS,BS})$;
If no balanced partition exists, output any partition A, B of W;
$S' \leftarrow$ `ABSeparator`(V, A, B, S);
return S'

Algorithm 13: `Components`(V, S)

$S' \leftarrow$ `Separator`(V, S);
$r := |S'| + |S|$;
$R \leftarrow \emptyset$; `// will contain one vertex from each` $C \in \mathcal{C}^{S'}$
for $v \in V \setminus S'$ **do**
  notFound $\leftarrow$ *True*;
  for $u \in R$ **do**
    if $\text{rank}(\Sigma_{uSS',vSS'}) = r + 1$ **then**
      $C_u \leftarrow C_u \cup \{v\}$;
      notFound $\leftarrow$ *False*;
  if *notFound* **then**
    create $C_v = \{v\}$;
    $R \leftarrow R \cup \{v\}$;
return S' *and all* C_j;

2.6 Tree reconstruction using noisy covariance oracles

In this paper we focus on the noiseless setting when entries of a matrix $\Sigma \in \mathcal{M}(G)$ may be accessed by a learner and these values are available exactly, without error – this is our "covariance oracle." In other words, we have shown that in many cases it is possible to invert the positive definite matrix Σ after seeing a tiny, adaptively chosen, fraction of its entries.

In many learning problems, the entries of Σ are not available exactly. This is the case in statistical problems when Σ is an unknown covariance matrix of a random vector X and its entries may be estimated from data. In this section we discuss how the results of this work may be extended to situations when covariances may not be observed exactly, for example, due to statistical fluctuations.

Here we limit ourselves to the study of the case when the underlying graph G is a tree. We show how Algorithm 2 may be modified to handle noise and establish sufficient conditions that guarantee correct recovery. Along similar lines, one may also modify the other algorithms introduced in this paper (for recovery of tree-like graphs and graphs of bounded treewidth). However, the details are somewhat cumbersome and go beyond the scope of this work.

In order to simplify the presentation, we assume that all diagonal elements of Σ are equal to 1, that is, Σ is a correlation matrix with entries σ_{ij} for $i \neq j$. The extension of the general case is straightforward, at the price of visually more complicated formulas.

In the discussion that follows, we assume that the noisy covariance oracle, when queried for the (i, j)-th entry of Σ, returns a value $\widehat{\sigma}_{ij} \in [-1, 1]$ satisfying

$$\max_{ij} \left|\widehat{\sigma}_{ij} - \sigma_{ij}\right| < \epsilon \tag{2.6.1}$$

for some $\epsilon \in (0, 1)$. We assume that $\widehat{\sigma}_{ii} = 1$ for all i. In a statistical setting when Σ is the covariance matrix of a random vector X and one may obtain independent samples of X, it is easy to construct such a noisy covariance oracle. We discuss this in more detail at the end of the section.

In the noise-free case we required that the graph $\mathcal{G}(\Sigma)$ is connected and generic, or, equivalently, Σ does not have any entry in $\{-1, 0, 1\}$. In the presence of noise, because of problems of identifiability, we need stronger assumptions on the entries of Σ corresponding to edges of the graph. In particular, we assume that there exist constants $0 < \delta < \gamma < 1$ such that

$$\delta \leq \left|\sigma_{ij}\right| \leq \gamma \qquad \text{for all } ij \in E\,. \tag{2.6.2}$$

By the product formula (2.1.2), this implies that for any two distinct $i, j \in V$,

$$\left|\sigma_{ij}\right| \geq \delta^D \,,$$

where D is the diameter of the graph. Although the diameter of G played no role in the noise-free setting, in the noisy case the recovery guarantees crucially depend on D. In particular, our recovery guarantees are only meaningful when D is logarithmic in $1/\epsilon$. To see why this happens, note that under assumption (2.6.2), $|\sigma_{ij}| \leq \gamma^{d(i,j)}$ where $d(i,j)$ denotes the distance of vertex i and vertex j in the tree. This value becomes indistinguishable from zero by the noisy covariance oracle unless $d(i,j) < \log(1/\epsilon)/\log(1/\gamma)$.

Remark. The assumption that the diameter of G is small is relatively mild. Many important real life examples of complex networks have small diameter – these are the so-called small-world networks. For example the diameter of the world wide web, with way over billion nodes Van den Bosch, Bogers, and Kunder [88], is around 19 Albert, Jeong, and Barabási [3], while social networks with over six billion individuals are believed to have a diameter of around six Milgram [70]. Small-world networks have also found applications in brain study Smith Bassett and Bullmore [8].

Next we show how Algorithm 2 may be modified so that it tolerates noise of magnitude ϵ – in the sense of (2.6.1).

Algorithm 2 uses the covariance oracle in order to check whether $\det\left(\Sigma_{ij,jk}\right) = 0$ for triples of vertices i, j, k, or equivalently, whether $\sigma_{ij}\sigma_{jk} = \sigma_{ik}$. Also, the algorithm sorts all the correlations σ_{uw} for a fixed w. We show that both of these steps can be correctly executed with a noisy covariance oracle if

$$\epsilon \;\leq\; \frac{1}{8}\delta^D(1-\gamma^2)\,. \tag{2.6.3}$$

Under (2.6.3), we may choose a value τ such that

$$\tau > 3\epsilon \quad \text{and} \quad \tau \leq \delta^D\left(1-\gamma^2\right) - 3\epsilon\,.$$

In order to test whether $\sigma_{ij}\sigma_{jk} = \sigma_{ik}$, we use the decision

$$\begin{array}{lll} \text{if} & \left|\widehat{\sigma}_{ij}\widehat{\sigma}_{jk} - \widehat{\sigma}_{ik}\right| < \tau & \text{accept} \\ \text{if} & \left|\widehat{\sigma}_{ij}\widehat{\sigma}_{jk} - \widehat{\sigma}_{ik}\right| > \tau & \text{reject}\,. \end{array}$$

To see why the decision is correct, observe first that for all $i, j \in [n]$,

$$\left|\sigma_{ij}\sigma_{jk} - \widehat{\sigma}_{ij}\widehat{\sigma}_{jk}\right| = \left|\sigma_{jk}\left(\sigma_{ij} - \widehat{\sigma}_{ij}\right) + \widehat{\sigma}_{ij}\left(\sigma_{jk} - \widehat{\sigma}_{jk}\right)\right| \leq 2\epsilon\,. \tag{2.6.4}$$

Hence, if $\sigma_{ij}\sigma_{jk} = \sigma_{ik}$, then

$$\left|\widehat{\sigma}_{ij}\widehat{\sigma}_{jk} - \widehat{\sigma}_{ik}\right| \leq 3\epsilon < \tau$$

and the decision is correct.

We now treat the case where $\sigma_{ij}\sigma_{jk} \neq \sigma_{ik}$. We consider two cases.
Case I: There exists a vertex m that separates all i, j, k, that is,

$$\sigma_{im}\sigma_{mk} = \sigma_{ik} \quad \text{and} \quad \sigma_{jm}\sigma_{mk} = \sigma_{jk}\,.$$

In this case

$$\begin{aligned}\left|\sigma_{ij}\sigma_{jk} - \sigma_{ik}\right| &= \left|\sigma_{im}\sigma_{jm}^2\sigma_{km} - \sigma_{im}\sigma_{km}\right| \\ &= |\sigma_{ik}|\left|\sigma_{jm}^2 - 1\right| \\ &\geq \delta^D\left(1-\gamma^2\right)\,. \end{aligned} \tag{2.6.5}$$

But then, arguing as above,

$$\begin{aligned}\left|\widehat{\sigma}_{ij}\widehat{\sigma}_{jk} - \widehat{\sigma}_{ik}\right| &\geq \left|\sigma_{ij}\sigma_{jk} - \sigma_{ik}\right| - 3\epsilon \\ &\geq \delta^D\left(1-\gamma^2\right) - 3\epsilon > \tau\end{aligned}$$

and the decision is once again correct. It remains to consider
Case II: Either i separates j and k or k separates i and j. Without loss of generality, assume the latter, so that $\sigma_{ij} = \sigma_{ik}\sigma_{jk}$. Then

$$\begin{aligned}\left|\sigma_{ij}\sigma_{jk} - \sigma_{ik}\right| &= \left|\sigma_{ik}\sigma_{jk}^2 - \sigma_{ik}\right| \\ &= |\sigma_{ik}|\left|\sigma_{jk}^2 - 1\right| \\ &\geq \delta^D\left(1-\gamma^2\right)\,.\end{aligned}$$

Hence, similarly to Case I, we have

$$\left|\widehat{\sigma}_{ij}\widehat{\sigma}_{jk} - \widehat{\sigma}_{ik}\right| \geq \left|\sigma_{ij}\sigma_{jk} - \sigma_{ik}\right| - 3\epsilon$$

$$\geq \quad \delta^D \left(1-\gamma^2\right) - 3\epsilon > \tau \,,$$

which proves correctness of the testing procedure.

The only other ingredient of Algorithm 2 that uses the covariance oracle performs queries of the form $|\sigma_{uw}| < |\sigma_{vw}|$. In the presence of a noisy covariance oracle satisfying (2.6.1), we may use the following rule:

$$\begin{aligned} \text{if } |\widehat{\sigma}_{uw}| < |\widehat{\sigma}_{vw}| - 2\epsilon \quad & \text{accept} \\ \text{if } |\widehat{\sigma}_{vw}| < |\widehat{\sigma}_{uw}| - 2\epsilon \quad & \text{reject} \\ \text{otherwise} \quad & \text{do either} \,. \end{aligned}$$

We now show that the decision rule accepts and rejects correctly. In the first two cases, the decision is clearly correct. Hence, we only need to examine the case when

$$|\widehat{\sigma}_{uw}| \geq |\widehat{\sigma}_{vw}| - 2\epsilon \quad \text{and} \quad |\widehat{\sigma}_{vw}| \geq |\widehat{\sigma}_{uw}| - 2\epsilon$$

happen simultaneously. In such case,

$$||\widehat{\sigma}_{uw}| - |\widehat{\sigma}_{vw}|| \leq 2\epsilon \,.$$

which implies

$$||\sigma_{uw}| - |\sigma_{vw}|| \leq 4\epsilon \,.$$

In terms of the ordering of correlations, it is enough that for a central vertex w in Algorithm 3 the following holds: if

(i) w does not separate u and v and

(ii) v is the neighbor of w

then $|\widehat{\sigma}_{vw}| > |\widehat{\sigma}_{uw}|$. Indeed, note that the only thing that matters is that, at each step of the algorithm, all vertices in the same connected component of $G \setminus w$ are sorted after the unique neighbour of w that belongs in that component. But if (i) and (ii) hold then

$$||\sigma_{uw}| - |\sigma_{vw}|| \;=\; |\sigma_{vw}|(1-|\sigma_{uv}|) \geq \delta(1-\gamma) > 4\epsilon \,,$$

where the last inequality follows by (2.6.3).

This concludes the proof of correctness of the modified procedure under condition (2.6.3).

We close this section by noting that a noisy covariance oracle satisfying (2.6.1) may easily be constructed when Σ is the covariance matrix of a zero-mean random

vector $X = (X_1, \ldots, X_n)$. All one needs is that, for each pair of indices $i, j \in [n]$ queried by the algorithm, one may obtain N i.i.d. samples of the pair (X_i, X_j). For example, if all components of X have a bounded fourth moment, say $\mathbb{E}X_i^4 \leq \kappa$ for all $i \in [n]$ for some $\kappa > 0$, then $\mathrm{Var}(X_i X_j) \leq \kappa$, and therefore one may use robust mean estimators (see, e.g., [63, Theorem 2]) to estimate $\sigma_{ij} = \mathbb{E}[X_i X_j]$ in a way that $\widehat{\sigma}_{ij}$ satisfy (2.6.3) (simultaneously, for all $i, j \in [n]$), with probability at least $1 - \eta$, whenever

$$N \geq 32 \left(\frac{\kappa}{\epsilon}\right)^2 \log \frac{n}{\eta} \,.$$

Chapter 3

Broadcasting on random recursive trees

The results in this chapter are joint work with Louigi Addario-Berry, Luc Devroye, and Gábor Lugosi.

3.1 Preliminaries

Let T_n be a rooted tree on $n+1$ vertices, whose vertices are labeled in the set $\{0, 1, \ldots, n\}$ and the root has label 0. The *parent* p_i of a vertex $i \in \{1, \ldots, n\}$ is the unique vertex on the path between the root and vertex i that is connected to i by an edge (accordingly, vertex i is called a *child* of p_i). Each vertex is assigned a bit value $B_i \in \{-1, 1\}$ generated by the following random mechanism: the root bit obtains a bit uniformly at random, while all other vertices have the same bit value as their parent with probability $1 - q$ and the opposite value with probability q, where $q \in [0, 1]$. In other words, for $i \in \{1, \ldots, n\}$

$$B_i = B_{p_i} Z_i \,,$$

where $Z_1, \ldots, Z_n$ are independent random variables taking values in $\{-1, 1\}$ with $\mathbb{P}\{Z_i = -1\} = q$.

In the *broadcasting problem* one wants to guess the bit value of the root, upon observing the bit values of a subset of vertices in the tree. We first consider the variant of the problem where one observes the tree T_n without labels, together with the bit value associated with every vertex. Note that since the vertex labels

are not observed, the identity of the root is not known. We call this the *root-bit reconstruction problem*. Then we consider a more difficult version of the problem, where the unlabeled tree is observed but only the bit values of the *leaves* (that is, vertices that are not parents of any vertex) are observed. We refer to this variant as the problem of *reconstruction from leaf bits*.

We will consider these problems when the underlying tree is a *random recursive tree*. Such trees are grown starting from the root vertex 0, by adding vertices recursively one-by-one according to some simple random rule. The two random rules that will be studied in the sequel are the *uniform attachment* model and the *linear preferential attachment* model (we will also refer to the model simply as *preferential attachment*). The two models will be defined in the respective sections.

Both reconstruction problems that we defined are binary classification problems, where one guesses the value of the root bit B_0 by an estimate $\widehat{b}$. The probability of error (or risk) is denoted by

$$R(n,q) = \mathbb{P}\left\{\widehat{b} \neq B_0\right\} .$$

We study the optimal risk

$$R^*(n,q) = \inf R(n,q) , \tag{3.1.1}$$

where the infimum is taken over all estimators $\widehat{b}$. In particular, we are interested in

$$R^*(q) = \limsup_{n\to\infty} R^*(n,q) .$$

Clearly, $R^*(n,q) \leq 1/2$ for all n and q and a principal question of interest is for what values of q one has $R^*(q) < 1/2$ and how $R^*(q)$ depends on q in both problems and under the various random attachment models.

We assume, for simplicity, that the generating mechanism of the tree and the value q are known to the statistician.

3.2 Broadcasting on uniform attachment trees

In the *uniform attachment tree*, for each $i \in \{1,\dots,n\}$, vertex i attaches with an edge to a vertex picked uniformly at random among vertices $0, 1, \dots, i-1$.

Before discussing root-bit estimators, we make an easy observation.

Lemma 3.2.1. *In the root-bit reconstruction problem and the reconstruction problem*

from leaf bits on a uniform attachment tree, $R^*(q) \geq q/2$. In particular, $R^*(1) = 1/2$. Moreover, $R^*(0) = 0$.

Proof. With probability q, the bit values of vertex 0 and vertex 1 are different. Since these two vertices are statistically indistinguishable after their labels are removed, on this event any classification rule has probability of error $\frac{1}{2}$. ∎

We can recover the maximum likelihood estimator of B_0, using existing results for the *inference of the root* problem that was described in Chapter 1. To describe such a classification rule with minimal probability of error, we first have to introduce some definitions.

A recursive labeling of a rooted tree $T = T_n$ on $n+1$ vertices is a labeling of the vertices of the tree with integers in $\{0, 1, \ldots, n\}$ such that every vertex has a distinct label, and the labels on every path starting from the origin are increasing. (Thus, the root has label 0.)

Write $V(T)$ for the set of vertices of a tree T. Given vertices $u, v \in V(T)$, we denote by $T^v_{u\downarrow}$ the subtree of T that contains all vertices whose path to v includes u.

For a vertex $v \in V(T)$, we denote by $\mathrm{Aut}(v, T)$ the number of vertices equivalent to v under graph isomorphism. Formally,

$$\mathrm{Aut}(v, T) = |\{w \in V(T) : \exists \text{ graph automorphism } \phi : T \to T \text{ where } \phi(v) = w\}|$$

Let $u_1, \ldots, u_j$ be the children of v and consider the subtrees $T^0_{u_1\downarrow}, \ldots, T^0_{u_j\downarrow}$. These subtrees belong to rooted graph isomorphism classes $S_1, \ldots, S_m$. For $i \in [m]$, let ℓ_i be the number of representatives of S_i, formally $\ell_i \stackrel{\text{def.}}{=} \left|\left\{k \in [j] : T^0_{u_k\downarrow} \in S_i\right\}\right|$. Moreover, $\overline{\mathrm{Aut}}\left(T^0_{v\downarrow}\right) \stackrel{\text{def.}}{=} \prod_{i=1}^m \ell_i!$.

It is shown in Bubeck, Devroye, and Lugosi [15, Proposition 1] that, given a tree T on $n+1$ vertices, for any node $u \in T$, the number of recursive labelings of T such that u has label 0 equals

$$\frac{n!}{\overline{\mathrm{Aut}}\left(T^u_{u\downarrow}\right) \cdot \prod_{v \in V(T)\setminus(L(T,u)\cup\{u\})} \left(|T^u_{v\downarrow}| \cdot \overline{\mathrm{Aut}}\left(T^u_{v\downarrow}\right)\right)},$$

where $L(T, u)$ is the set of leaves of $T^u_{u\downarrow}$. Since every recursive tree is equally likely in the uniform attachment model, as a consequence we have that, given an unlabeled tree T, the likelihood of a vertex u being the root is proportional to the function

$$\lambda(u) \;=\; \frac{1}{\mathrm{Aut}(u, T) \prod_{v \in V\setminus L(T,u)} \left(\left|T^u_{v\downarrow}\right| \cdot \overline{\mathrm{Aut}}\left(T^u_{v\downarrow}\right)\right)}. \tag{3.2.1}$$

Hence we immediately obtain the following.

Corollary 3.2.2. *For the root-bit reconstruction problem on a uniform random recursive tree T, the following estimator b^* of the root bit B_0 minimizes the probability of error:*

$$b^* = \begin{cases} 1 & \text{if } \sum_{u \in V(T): B_u = 1} \lambda(u) > \sum_{u \in V(T): B_u = 0} \lambda(u) \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $\mathbb{P}\{b^* \neq B_0\} = R(n, q)$.

The analysis of the optimal rule described above seems difficult. Instead, we analyze various other classification methods.

3.3 The majority rule

In this section, we analyze the majority rule. We denote by $\widehat{b}^{\mathrm{maj}}$ the majority vote among all bit values. Our result can be summarized as follows.

Theorem 3.3.1. *Consider the broadcasting problem in a uniform attachment tree. Denote the probability of error of the majority vote by*

$$R^{maj}(n, q) = \mathbb{P}\left\{\widehat{b}_{maj} \neq B_0\right\} .$$

For both the root-bit reconstruction problem and the reconstruction problem from leaf bits, the following hold.

There exists $c > 0$ such that

$$\limsup_{n \to \infty} R^{maj}(n, q) \leq cq \quad \text{for all } q \in [0, 1] .$$

Moreover,

$$\limsup_{n \to \infty} R^{maj}(n, q) < 1/2 \quad \text{if } q \in [0, 1/4)$$

and

$$\limsup_{n \to \infty} R^{maj}(n, q) = 1/2 \quad \text{if } q \in [1/4, 1/2] .$$

Observe that the number of vertices in the uniform random recursive tree T_n with bit value B_0 is distributed as the number of black balls in a Pólya urn of black and white balls with random replacements defined as follows: initially, there is one

black ball in the urn. For $i = 1, 2, \ldots$, at time i, a uniformly random ball is selected from the urn. The ball is returned to the urn together with a new ball whose color is decided according to a Bernoulli(q) coin toss. If the value is 1 (which happens with probability q), the color of the new ball is the opposite of the selected one. Otherwise the new ball has the same color as that of the selected ball.

Such randomized urn processes have been thoroughly studied. In particular, early results can be traced back to Wei [91] and depend on results by Athreya and Karlin [6] concerning random multi-type trees. More recently, Janson [50] and Knape and Neininger [58] proved general limit laws that may be used to analyze the probability of error of the majority rule.

Instead of using these limit laws, our starting point is a decomposition of the uniform random recursive tree defined below. This methodology allows us to prove the first inequality of Theorem 3.3.1 in an elementary way. Moreover, this decomposition may be used to treat the case of the reconstruction problem from leaf bits in a straightforward fashion. The same technique will also prove useful in analyzing the majority vote in the linear preferential attachment model.

In Sections 3.3.3 and 3.3.4 we use Janson's limit theorems to derive qualitative results on the probability of error of the majority rule.

In this entire section we assume without loss of generality that $q \leq 1/2$. The conclusions of the theorem hold trivially for $q \geq \frac{1}{2}$.

3.3.1 A decomposition of the URRT

It is convenient to decompose our random process as follows. First, a uniform attachment tree $\mathcal{T}_n$ is generated in the standard way, without attached bit values. Then we identify all nodes apart from the root as follows:

- with probability $2q$, they are *marked*. Then there is a coin flip ξ that takes values uniformly at random in $\{-1, 1\}$ and determines if a marked node takes the same bit value as its parent or not (in which case we say it *flips*).
- with probability $1 - 2q$ they are *not marked*. These nodes do not flip and thus have the same bit value as their parent.

The root and marked nodes become roots of subtrees that are disjoint and shatter the uniform recursive tree into many pieces. Each of the subtrees consists of nodes of the same bit value and the roots have the bit value of their original parent if $\xi = 1$ and different otherwise (if $\xi = -1$). We recall that nodes are numbered 0 through n, where 0 is the root. The node variables are, for node i:

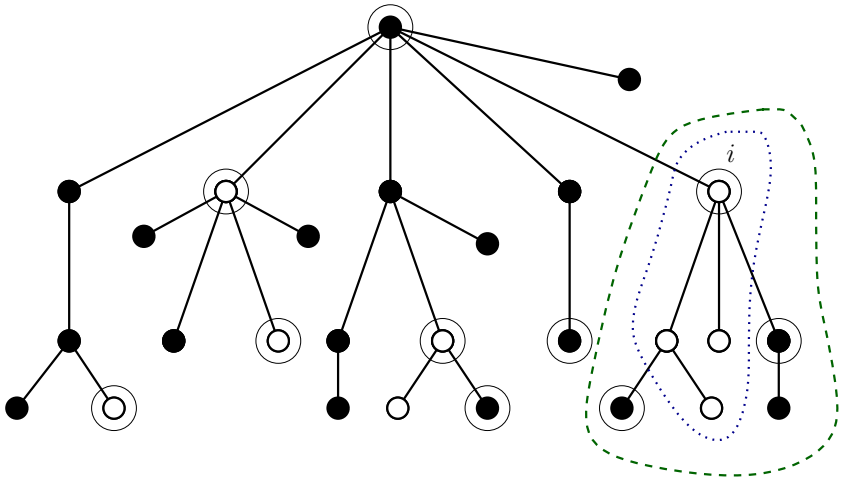


Figure 3.1: Illustration of the decomposition of a tree. The vertices enclosed by a circle are marked. The subtree that is enclosed by a dotted curve is $\widetilde{T}_i$. The subtree that is enclosed by a dashed curve is $T^0_{i\downarrow}$.

- $p_i \in \{0, \ldots, i-1\}$: the uniform random index of its parent
- $m_i \in \{0, 1\}$: a Bernoulli$(2q)$ random variable (1 indicates marking)
- $\xi_i \in \{-1, 1\}$: a Rademacher random variable used for flipping bit values ($\mathbb{P}[\xi_i = 1] = \frac{1}{2}$).

For each $i \in \{1, \ldots, n\}$, p_i, m_i, and ξ_i are independent. For our convenience, we will consider bit values in $\{-1, 1\}$ instead of $\{0, 1\}$. Let B_i be the bit value in $\{-1, 1\}$ of node i and assume $B_0 = 1$. We set

$$B_i = \begin{cases} B_{p_i}, & \text{if } m_i = 0 \text{ (no marking) or if } m_i = 1, \xi_i = +1 \text{ (no flipping)} \\ -B_{p_i}, & \text{if } m_i = 1 \text{ and } \xi_i = -1\,. \end{cases}$$

Formally, $B_i = (m_i \xi_i + (1 - m_i))\, B_{p_i}$. Note that

- The shape of $\mathcal{T}_n$ depends only upon $p_1, \ldots p_n$.
- The decomposition of the tree into subtrees depends upon $p_1, \ldots p_n$ and $m_1, \ldots, m_n$.
- The bit counting algorithm (that outputs the majority) uses $\xi_1, \ldots, \xi_n$ as well as the two other sequences.

Let $\widetilde{T}_i$ be the maximal size subtree of $T^0_{i\downarrow}$ with root i and homogeneous bit values, such that all its vertices apart from i are unmarked (i can be either marked or unmarked). See Figure 3.3.1 for an illustration. We write $N_i = |\widetilde{T}_i|$.

3.3.2 Linear upper bound for the probability of error

Here we prove the first part of Theorem 3.3.1, that is, that there exists a universal constant c such that

$$\limsup_{n\to\infty} R^{\mathrm{maj}}(n,q) \le cq \quad \text{for all } q \in [0,1]\,. \tag{3.3.1}$$

Taking $c \ge 8$, we may assume that $q \le 1/8$. We will first consider that we have access to all bit values.

The difference between the number of nodes of value 1 and those of value -1 is given by

$$\Delta \stackrel{\mathrm{def}}{=} N_0 + \sum_{i=1}^{n} N_i B_{p_i} \xi_i m_i.$$

In this formula, we only count subtrees corresponding to vertices with $m_i = 1$, and add the vertex count N_i to the $B_{p_i}\xi_i$ side. As the ξ_i's are independent of the rest of the variables, we have

$$\mathbb{E}\left[\Delta\right] = \mathbb{E}\left[N_0\right]\,. \tag{3.3.2}$$

Also, by first conditioning on everything but the ξ_i's, we have

$$\mathbb{E}\left[\Delta^2\right] = \mathbb{E}\left[N_0^2\right] + \sum_{i=1}^{n} \mathbb{E}\left[N_i^2 B_{p_i}^2 m_i\right] = \mathbb{E}\left[N_0^2\right] + 2q\sum_{i=1}^{n} \mathbb{E}\left[N_i^2\right]\,.$$

So,

$$\mathrm{Var}\left[\Delta\right] = \mathrm{Var}\left[N_0\right] + 2q\sum_{i=1}^{n} \mathbb{E}\left[N_i^2\right]\,.$$

By Chebyshev's inequality,

$$\begin{aligned}\mathbb{P}\left\{\widehat{b}_{\mathrm{maj}} \neq B_0\right\} &\le \mathbb{P}\left\{\Delta \le 0\right\} \le \frac{\mathrm{Var}\left[\Delta\right]}{\left(\mathbb{E}\left[\Delta\right]\right)^2} \\ &= \frac{\mathrm{Var}\left[N_0\right]}{\left(\mathbb{E}\left[N_0\right]\right)^2} + 2q\frac{\sum_{i=1}^{n}\mathbb{E}\left[N_i^2\right]}{\left(\mathbb{E}\left[N_0\right]\right)^2}\,.\end{aligned}$$

In Lemmas 3.3.5, 3.3.6, and 3.3.7, stated and proved in Section 3.3.5, we establish

bounds for the first and second moments of N_i. These bounds imply (3.3.1) as follows.

Let us write $\zeta(\alpha) = \sum_{i=1}^{\infty} 1/i^{\alpha}$ for the Riemann zeta function. Also, we write $\widetilde{\zeta}(\alpha) = \sum_{i=1}^{\infty} (\log i)/i^{\alpha}$. Note that both functions are finite and decreasing for $\alpha > 1$. By Lemmas 3.3.5 and 3.3.7,

$$\begin{aligned}
&\frac{\mathrm{Var}\left[N_0\right]}{\left(\mathbb{E}\left[N_0\right]\right)^2} \\
&\quad \leq \; 2qe^4(4+e)\zeta(2-4q) + 2qe^4 n^{-(1-4q)} + 12e^5 q^2 \widetilde{\zeta}(2-4q) + 4e^4 q^2 n^{-(1-4q)} \log n \\
&\quad \leq \; c_1 q + c_2 q^2 + o_n(1)
\end{aligned}$$

with $c_1 = 2e^4(4+e)\zeta(3/2)$ and $c_2 = 12e^5\widetilde{\zeta}(3/2)$, where we used the fact that ζ and $\widetilde{\zeta}$ are decreasing functions and that $q \leq 1/8$.

On the other hand, by Lemmas 3.3.5 and 3.3.6,

$$\frac{\sum_{i=1}^{n} \mathbb{E}\left[N_i^2\right]}{\left(\mathbb{E}\left[N_0\right]\right)^2} \leq e^4(4+e)\zeta(2-4q) + n^{-(1-4q)}e^3 \leq \frac{c_1}{2} + o_n(1)\,.$$

Hence, for all $q \leq 1/8$,

$$\mathbb{P}\left\{\widehat{b}_{\mathrm{maj}} \neq B_0\right\} \leq 2c_1 q + c_2 q^2 + o_n(1)\,,$$

proving (3.3.1).

In Section 3.3.6, we will see that similar estimates hold for the first and second moments of the leaf counts in N_i. Therefore, a similar computation holds also for the reconstruction problem from leaf bits (in Section 3.3.6 we provide a more detailed discussion on that).

3.3.3 Majority is better than random guessing for $q < 1/4$

In this Section, we show the second part of Theorem 3.3.1, that is,

$$\limsup_{n\to\infty} R^{\mathrm{maj}}(n,q) < \frac{1}{2} \quad \text{for all } q < \frac{1}{4}\,. \tag{3.3.3}$$

To this end, we will apply the limit theorems for Pólya urns with randomized replacements that can be found in Janson [50].

Consider first the model when bit values are observed at every vertex of the

tree. Recall that the number of vertices with bit value B_0 may be represented by the number of white balls in a Pólya urn of white and black balls, initialized with one white ball. At each time, a random ball is drawn. The drawn ball is returned to the urn, together with another ball whose color is the same as the drawn one with probability $1-q$ and has opposite color with probability q. The asymptotic distribution of the balls is determined by the eigenvalues and eigenvectors of the transpose of the matrix of the expected number of returned balls. In this case, the matrix is simply

$$\begin{pmatrix} 1-q & q \\ q & 1-q \end{pmatrix},$$

whose eigenvalues are 1 and $1-2q$. If $q < 1/4$, by [50, Theorem 3.24],

$$\frac{\Delta - \mathbb{E}\Delta}{n^{1-2q}}$$

converges, in distribution, to a random variable whose distribution is symmetric about zero and has a positive density at 0. Since

$$\frac{\mathbb{E}\Delta}{n^{1-2q}} \geq \frac{1}{e\Gamma(2-2q)}$$

by (3.3.2) and the calculations in Lemmas 3.3.3 and 3.3.5 below, it follows that

$$\begin{aligned} \limsup_{n\to\infty} \mathbb{P}\left\{\widehat{b}_{\text{maj}} \neq B_0\right\} &\leq \limsup_{n\to\infty} \mathbb{P}\left\{\Delta \leq 0\right\} \\ &= \limsup_{n\to\infty} \mathbb{P}\left\{\frac{\Delta - \mathbb{E}\Delta}{n^{1-2q}} \leq -\frac{\mathbb{E}\Delta}{n^{1-2q}}\right\} \\ &< \frac{1}{2}, \end{aligned}$$

proving (3.3.3).

The majority rule in the leaf-bit reconstruction problem may also be studied using Pólya urns with random replacements. In this case the urn has four colors, corresponding to: (1) leaf vertices whose bit value equals B_0, (2) leaf vertices whose bit value equals $1-B_0$, (3) internal vertices whose bit value equals B_0, (4) internal vertices whose bit value equals $1-B_0$.

Initially, there is one ball of type (1) and no balls of any other type in the urn. When a ball of type (1) is drawn, it is replaced by a ball of type (3). With probability $1-q$, an additional ball of type (1) is added to the urn, and with probability q a ball

of type (2) is added, etc. The resulting replacement matrix is

$$\begin{pmatrix} -q & q & 1 & 0 \\ q & -q & 0 & 1 \\ 1-q & q & 0 & 0 \\ q & 1-q & 0 & 0 \end{pmatrix}$$

The eigenvalues of this matrix are $1, 1-2q, -1, -1$, and once again [50, Theorem 3.24] applies. Reasoning as previously and using Lemma 3.3.8, we have that for $q < 1/4$,

$$\limsup_{n\to\infty} \mathbb{P}\left\{\widehat{b}_{\text{maj}} \neq B_0\right\} \;<\; \frac{1}{2}\,.$$

3.3.4 Majority is not better than random guessing for $q \geq 1/4$

In this section, we show the third part of Theorem 3.3.1, that is,

$$\limsup_{n\to\infty} R^{\text{maj}}(n,q) = \frac{1}{2} \quad \text{for all } q \in [1/4, 1/2]\,. \tag{3.3.4}$$

The case $q > \frac{1}{4}$ follows from the decomposition of $\mathcal{T}_n$ that we have introduced and the following lemma:

Lemma 3.3.2 (Rogozin, 1961 [81])**.** *Let $\eta_1, \ldots, \eta_n$ be i.i.d. Bernoulli$\left(\frac{1}{2}\right)$ random variables. Then for any $\alpha_1, \ldots, \alpha_n$, all nonzero,*

$$\sup_x \;\; \mathbb{P}\left\{\sum_{i=1}^n \eta_i \alpha_i = x\right\} \leq \frac{\gamma}{\sqrt{n}}$$

for some universal constant γ, uniformly over all choices of $\alpha_1, \ldots \alpha_n$.

Indeed,

$$\begin{aligned} \mathbb{P}\left\{\widehat{b}_{\text{maj}} \neq B_0\right\} \;&\geq\; \mathbb{P}\left\{\Delta < 0\right\} = \mathbb{P}\left\{\sum_{i=1}^n N_i B_{p_i} m_i \xi_i < -N_0\right\} \\ &=\; \frac{1}{2}\mathbb{P}\left\{\left|\sum_{i=1}^n N_i B_{p_i} m_i \xi_i\right| > N_0\right\} \quad \text{(by symmetry)} \\ &\geq\; \frac{1}{2}\mathbb{E}\left[\left(1 - \frac{2\gamma(N_0+1)}{\sqrt{\sum_{i=1}^n m_i}}\right)_+\right]. \end{aligned}$$

The inequality above follows by first conditioning on all but the ξ_i's and using Lemma 3.3.2. The latter expression is further lower bounded by

$$\begin{aligned}
&\frac{1}{2}\left(\mathbb{E}\left[\left(1-\frac{2\gamma(N_0+1)}{\sqrt{qn}}\right)_+\right]-\mathbb{P}\left\{\sum_{i=1}^{n} m_i < qn\right\}\right)\\
&\geq \frac{1}{2}\left(1-\frac{2\gamma\mathbb{E}\left[N_0+1\right]}{\sqrt{qn}}\right)_+ - \mathbb{P}\left\{\text{Binomial}(n,2q) < qn\right\} \text{ (by Jensen's inequality)}\\
&= \frac{1}{2} - o_n\left(1\right) ,
\end{aligned}$$

since $\mathbb{E}\left[N_0\right] = o\left(\sqrt{n}\right)$ when $q > \frac{1}{4}$ by Lemma 3.3.5.

In the "critical" case $q = 1/4$, we may once again use the Pólya urn representation and the limit theorems in Janson [50]. Indeed, by working as in Section 3.3.3, [50, Theorem 3.23] applies and it implies that

$$\frac{\Delta - \mathbb{E}\Delta}{n^{1/2}\log n}$$

converges in distribution to a normal random variable. Since

$$\frac{\mathbb{E}\Delta}{n^{1/2}\log n} = o\left(1\right)$$

by Lemmas 3.3.3 and 3.3.5, we have

$$\begin{aligned}
\limsup_{n\to\infty}\mathbb{P}\left\{\widehat{b}_{\text{maj}} \neq B_0\right\} &\geq \limsup_{n\to\infty}\mathbb{P}\left\{\Delta < 0\right\}\\
&= \limsup_{n\to\infty}\mathbb{P}\left\{\frac{\Delta-\mathbb{E}\Delta}{n^{1/2}\log n} \leq -\frac{\mathbb{E}\Delta}{n^{1/2}\log n}\right\} = \frac{1}{2}\,.
\end{aligned}$$

A similar computation can be performed for the case when only leaf-bits are observed.

3.3.5 The study of N_i

We will estimate the first and second moments of the random variables $(N_i)_{0\leq i\leq n}$ (recall their definition from subsection 3.3.1). We begin with some auxiliary lemmas.

Lemma 3.3.3. *For all $i \geq 0$ and constant $\alpha \geq 0$,*

$$\prod_{t=i}^{n-1}\left(1+\frac{\alpha}{t+1}\right)=\frac{\Gamma(\alpha+n+1)}{\Gamma(n+1)}\cdot\frac{\Gamma(i+1)}{\Gamma(\alpha+i+1)}\,.$$

Proof.

$$\prod_{t=i}^{n-1}\left(1+\frac{\alpha}{t+1}\right)=\frac{\prod_{t=0}^{n-1}\left(\frac{\alpha+1+t}{1+t}\right)}{\prod_{t=0}^{i-1}\left(\frac{\alpha+1+t}{1+t}\right)}\,. \tag{3.3.5}$$

Also,

$$\prod_{t=0}^{n-1}\left(\frac{\alpha+1+t}{1+t}\right)=\frac{\Gamma(\alpha+n+1)}{\Gamma(\alpha+1)\,\Gamma(n+1)}\,,$$

implying that (3.3.5) equals

$$\frac{\Gamma(\alpha+n+1)}{\Gamma(\alpha+1)\,\Gamma(n+1)}\cdot\frac{\Gamma(\alpha+1)\,\Gamma(i+1)}{\Gamma(\alpha+i+1)}=\frac{\Gamma(\alpha+n+1)}{\Gamma(n+1)}\cdot\frac{\Gamma(i+1)}{\Gamma(\alpha+i+1)}\,.$$

■

Lemma 3.3.4. *For $n \geq 1$ and $\alpha \in [0,1]$,*

$$\left(\frac{n+1}{e}\right)^{\alpha}\leq\frac{\Gamma(\alpha+n+1)}{\Gamma(n+1)}\leq(n+1)^{\alpha}\,.$$

Proof. If Gamma $(n+1)$ denotes a Gamma random variable with parameters $(n+1,1)$, then

$$\begin{aligned}\frac{\Gamma(\alpha+n+1)}{\Gamma(n+1)} &= \frac{\int_0^\infty x^{\alpha+n}e^{-x}dx}{\int_0^\infty x^{n}e^{-x}dx}\\ &= \mathbb{E}\left[\text{Gamma}\,(n+1)^{\alpha}\right]\\ &\leq \left(\mathbb{E}\left[\text{Gamma}\,(n+1)\right]\right)^{\alpha}=(n+1)^{\alpha}\,,\end{aligned}$$

by Jensen's inequality. We show the lower bound by induction on n. For $n=1$ it holds for all $\alpha\in[0,1]$, since $\left(\frac{2}{e}\right)^{\alpha}\leq 1\leq\Gamma(2+\alpha)$. For larger n, we have:

$$\frac{\Gamma(\alpha+n+1)}{\Gamma(n+1)}=\frac{n+\alpha}{n}\cdot\frac{\Gamma(\alpha+n)}{\Gamma(n)}\geq\frac{n+\alpha}{n}\left(\frac{n}{e}\right)^{\alpha}\geq\left(\frac{n+1}{e}\right)^{\alpha}\,,$$

where the first inequality follows by induction hypothesis and the second since $\frac{n+\alpha}{n}\geq\left(\frac{n+1}{n}\right)^{\alpha}$. ■

Lemma 3.3.5. *For all $i \geq 0$ and $q \leq \frac{1}{2}$,*

$$e^{-1}\left(\frac{n+1}{i+1}\right)^{1-2q} \leq \mathbb{E}\left[N_i\right] \leq e\left(\frac{n+1}{i+1}\right)^{1-2q} .$$

Proof. The statement follows immediately by Lemmas 3.3.3 and 3.3.4 by noting that

$$\mathbb{E}\left[N_i\right] = \prod_{t=i}^{n-1}\left(1+\frac{1-2q}{t+1}\right) . \tag{3.3.6}$$

To see that (3.3.6) holds, define $Y_i = 1$ and, for $t \in \{i, \ldots, n-1\}$, let

$$Y_{t+1} = Y_t + \beta_{1-2q}\beta_{Y_t/(t+1)} .$$

where each appearance of β_x denotes an independent Bernoulli(x) random variable. Clearly, Y_t is distributed as the number of vertices counted by N_i and which have label at most t. Hence N_i has the same distribution as Y_n. For all $t \geq 1$, by conditioning on Y_t we see that

$$\mathbb{E}\left[Y_{t+1}\right] = \mathbb{E}\left[Y_t\right]\left(1+\frac{1-2q}{t+1}\right) ,$$

from which (3.3.6) is immediate. ∎

Lemma 3.3.6. *For all $i \geq 0$ and $q \leq \frac{1}{2}$,*

$$\mathbb{E}\left[N_i^2\right] \leq \left(\frac{n+1}{i+1}\right)^{2-4q} e^{2(1-2q)}\left(4+e\right) + e\left(1-2q\right) .$$

Proof. As in Lemma 3.3.5, we use the random variables $(Y_t)_{i\leq t\leq n}$, where Y_t is distributed as the number of vertices with label at most t that are counted by N_i. Then $N_i \stackrel{d}{=} Y_n$. Let us set $\alpha = 1-2q$, $f(t) = (1-2q)\,\frac{\mathbb{E}[Y_t]}{t+1}$, and $x_t = \mathbb{E}\left[Y_t^2\right]$. Then it is easy to confirm the recurrence

$$x_i = 1, \;\; x_{t+1} = x_t\left(1+\frac{2\alpha}{t+1}\right) + f(t) , \qquad i \leq t \leq n .$$

The solution is given by

$$x_n \;=\; x_i\prod_{t=i}^{n-1}\left(1+\frac{2\alpha}{t+1}\right) + \sum_{s=i+1}^{n-1}\prod_{t=s}^{n-1}\left(1+\frac{2\alpha}{t+1}\right) f\left(s-1\right) + f\left(n-1\right) .$$

Combining the Lemmas 3.3.3, 3.3.4, 3.3.5, and the bound $f(t) \leq \alpha \left(\frac{t+1}{i+1}\right)^{\alpha} \frac{e}{t+1}$,

$$\begin{aligned}
x_n &\leq x_i \left(\frac{n+1}{i+1}\right)^{2\alpha} e^{2\alpha} + \sum_{s=i+1}^{n-1} \left(\frac{n+1}{s+1}\right)^{2\alpha} e^{2\alpha+1} \alpha \left(\frac{s}{i+1}\right)^{\alpha} \frac{1}{s} + \alpha e \\
&= \left(\frac{n+1}{i+1}\right)^{2\alpha} e^{2\alpha} \left(1 + \sum_{s=i+1}^{n-1} \frac{s^{\alpha} \cdot e\alpha \, (i+1)^{\alpha}}{s \, (s+1)^{2\alpha}}\right) + \alpha e \\
&\leq \left(\frac{n+1}{i+1}\right)^{2\alpha} e^{2\alpha} \left(1 + \frac{e\alpha}{i+1} + \sum_{s=i+2}^{n-1} \frac{e\alpha \, (i+1)^{\alpha}}{s^{1+\alpha}}\right) + \alpha e \\
&\leq \left(\frac{n+1}{i+1}\right)^{2\alpha} e^{2\alpha} \left(4 + e\alpha \, (i+1)^{\alpha} \int_{i+1}^{\infty} \frac{1}{s^{1+\alpha}} ds\right) + \alpha e \\
&= \left(\frac{n+1}{i+1}\right)^{2\alpha} e^{2\alpha} \left(4 + \frac{e\alpha \, (i+1)^{\alpha}}{\alpha \, (i+1)^{\alpha}}\right) + \alpha e \\
&\leq \left(\frac{n+1}{i+1}\right)^{2\alpha} e^{2\alpha} \, (4+e) + \alpha e \, .
\end{aligned}$$

Replacing α by $1 - 2q$, we have

$$\mathbb{E}\left[N_i^2\right] \leq \left(\frac{n+1}{i+1}\right)^{2-4q} e^{2(1-2q)} \, (4+e) + e \, (1-2q) \, .$$

■

Recall the notation $\zeta(\alpha) = \sum_{i=1}^{\infty} 1/i^{\alpha}$ and $\widetilde{\zeta}(\alpha) = \sum_{i=1}^{\infty} (\log i)/i^{\alpha}$.

Lemma 3.3.7. $\mathrm{Var}(N_0)$ *is bounded by*

$$2qe^2(4+e)(n+1)^{2-4q}\zeta(2-4q) + 2nqe^2 + 12e^3q^2(n+1)^{2-4q}\widetilde{\zeta}(2-4q) + 4e^2q^2n\log n \, .$$

Proof. Knowing the parent selectors $p_1, \ldots, p_n$ and the coin flips $\xi_1, \ldots, \xi_n$, we have that N_0 is a function of the independent random variables $m_1, \ldots, m_n$. Note that resampling one of them, say m_i, does not change the value of N_i. Moreover, resampling m_i can change N_0 by at most N_i: if before resampling we had $m_i = 0$ and $\widetilde{T}_i \subset \widetilde{T}_0$, and after resampling we have $m_i = 1$, then N_0 decreases by N_i; also, if before resampling we had $m_i = 1$ and after resampling we have $m_i = 0$, then $\widetilde{T}_i$ might become a subtree of $\widetilde{T}_0$ and then N_0 increases by N_i. Hence, by the Efron-Stein inequality ([35, 86]),

$$\mathrm{Var}(N_0 | p_1, \ldots, p_n, \xi_1, \ldots, \xi_n) \leq \sum_{i=1}^{n} 2q(1-2q)\mathbb{E}\left[N_i^2 | p_1, \ldots, p_n, \xi_1, \ldots, \xi_n\right] \, .$$

Writing $Z_0 = \mathbb{E}\left[N_0 | p_1, \ldots, p_n, \xi_1, \ldots, \xi_n\right]$, we have

$$\mathrm{Var}(N_0) = \mathbb{E}\,\mathrm{Var}(N_0 | p_1, \ldots, p_n, \xi_1, \ldots, \xi_n) + \mathrm{Var}(Z_0) \leq 2q \sum_{i=1}^{n} \mathbb{E}N_i^2 + \mathrm{Var}(Z_0)\ .$$

The first term on the right-hand side may be bounded using Lemma 3.3.6, by

$$\begin{aligned} 2q \sum_{i=1}^{n} \mathbb{E}N_i^2 &\leq 2qe^2 \sum_{i=1}^{n} \left(\left(\frac{n+1}{i+1} \right)^{2-4q} (4+e) + 1 \right) \\ &\leq 2qe^2(4+e)(n+1)^{2-4q}\zeta(2-4q) + 2nqe^2\ . \end{aligned}$$

To bound $\mathrm{Var}(Z_0)$, let δ_i be the distance between the root and node i in $\widetilde{T}_0$. These distances are a function of $p_1, \ldots, p_n$ only and, therefore, we have

$$Z_0 = \sum_{v} (1-2q)^{\delta_v} = 1 + \sum_{j=1}^{n} (1-2q)^{\delta_j}\ .$$

We define

$$Z_j = \sum_{v \in T_{j\downarrow}^0} (1-2q)^{\delta_v - \delta_j}\,, \quad 0 \leq j \leq n\ .$$

Let Z_i' denote the modification of Z_i when the random variable p_i is replaced by an independent copy p_i' and the other values $p_1, \ldots p_{i-1}, p_{i+1}, \ldots, p_n$ are kept unchanged. Define similarly the variables δ_i'. Observe that if p_j is replaced by p_j', then

$$Z_0 - Z_0' = Z_j \left((1-2q)^{\delta_j} - (1-2q)^{\delta_j'} \right)$$

whose absolute value is at most

$$Z_j\,(1-2q)^{\min\left(\delta_j, \delta_j'\right)} \left(1 - (1-2q)^{|\delta_j - \delta_j'|} \right) \leq \begin{cases} 0, & \text{if } \delta_j = \delta_j' \\ Z_j 2q|\delta_j - \delta_j'|, & \text{else} \end{cases}$$

Therefore, by the Efron-Stein inequality,

$$\begin{aligned} \mathrm{Var}\left[Z_0\right] &\leq \frac{1}{2} \sum_{j=1}^{n} \mathbb{E}\left[Z_j^2 4q^2 \left(\delta_j - \delta_j' \right)^2 \right] \\ &= 2q^2 \sum_{j=1}^{n} \mathbb{E}\left[Z_j^2\right] \mathbb{E}\left[\left(\delta_j - \delta_j' \right)^2 \right] \qquad \text{(by independence)} \end{aligned}$$

By Jensen's inequality, $\mathbb{E}\left[Z_j^2\right] \leq \mathbb{E}\left[N_j^2\right]$. Moreover,

$$\mathbb{E}\left[\left(\delta_j - \delta_j'\right)^2\right] = 2\mathrm{Var}\left[\delta_j\right] \leq 2\log j \tag{3.3.7}$$

by well-known properties of uniform random recursive trees (Devroye [29]). Therefore,

$$\begin{aligned}
\mathrm{Var}\left[Z_0\right] &\leq 4q^2 \sum_{j=1}^{n} \mathbb{E}\left[Z_j^2\right] \log j \\
&\leq 4q^2 \sum_{j=1}^{n} \mathbb{E}\left[N_j^2\right] \log j \\
&\leq 4q^2 \sum_{j=1}^{n} \left(\left(\frac{n+1}{j+1}\right)^{2-4q} e^2\left(4+e\right) + e^2 \right) \log j \\
&\qquad \text{(by Lemma 3.3.6)} \\
&\leq 12e^3 q^2 (n+1)^{2-4q} \sum_{j=1}^{n} \frac{\log j}{(j+1)^{2-4q}} + 4e^2 q^2 \log(n!) \\
&\leq 12e^3 q^2 (n+1)^{2-4q} \widetilde{\zeta}(2-4q) + 4e^2 q^2 n \log n \,.
\end{aligned}$$

■

3.3.6 Majority of the leaf bits

We have proved Theorem 3.3.1 for the root-bit reconstruction problem. It remains to show the analogous statements for the reconstruction problem from leaf bits, that is, for the case when $\widehat{b}^{\mathrm{maj}}$ denotes the majority vote among the bit values observed on the leaves only. We can show this with a slight adaptation of the proof presented in the previous section.

Recall that N_i is the maximum number of unmarked vertices in a subtree rooted at i in $T_{i\downarrow}^0$ (i is included and can be marked or not marked). Let $\overline{N}_i$ be the number of them that are leaves. It suffices to show that the first and second moments of $\overline{N}_i$ satisfy inequalities analogous to those of Lemmas 3.3.5, 3.3.6, and 3.3.7.

The next lemma establishes the desired analogues of Lemmas 3.3.5 and 3.3.6. This suffices to prove (3.3.1) by the same argument as before. (The corresponding extension of Lemma 3.3.7 is straightforward and is omitted.)

Lemma 3.3.8. *For all $i \leq n$,*

$$\frac{1}{24e}\left(\frac{n+1}{i+1}\right)^{1-2q} - \frac{i}{4ne} \leq \mathbb{E}\left[\overline{N}_i\right] \leq e\left(\frac{n+1}{i+1}\right)^{1-2q}$$

and

$$\mathbb{E}\left[\overline{N}_i^2\right] \leq \left(\frac{n+1}{i+1}\right)^{2-4q} e^{2(1-2q)}\left(4+e\right) + e\left(1-2q\right) .$$

Proof. The upper bounds for the expectation and the second moment clearly hold by the fact that $\overline{N}_i \leq N_i$ and by Lemma 3.3.5.

Recall from the proof of Lemma 3.3.5 that for $t \in \{i, \ldots, n-1\}$, Y_t denotes the number of vertices that are counted by N_i and whose label is at most t. Similarly, define $\overline{Y}_t$ as the number of leaves in the same subtree. Then, $\overline{Y}_n$ is distributed as $\overline{N}_i$. For $t \in \{i+1, \ldots, n\}$, we have

$$\mathbb{E}\left[\overline{Y}_t \middle| \overline{Y}_{t-1}, Y_{t-1}\right] = \overline{Y}_{t-1} + \frac{1-2q}{t}\left(Y_{t-1} - \overline{Y}_{t-1}\right),$$

since given $\overline{Y}_{t-1}, Y_{t-1}$, with probability $\frac{1-2q}{t}\left(Y_{t-1} - \overline{Y}_{t-1}\right)$ the number of leaves increases by one ($1-2q$ is the probability that the new vertex is unmarked). Hence $a_t \stackrel{\text{def.}}{=} \mathbb{E}\overline{Y}_t$ satisfies, for $t \in \{i+1, \ldots, n\}$,

$$a_t = a_{t-1}\left(1 - \frac{1-2q}{t}\right) + f(t) ,$$

where $f(t) = \frac{1-2q}{t}\mathbb{E}Y_{t-1}$. Solving the recurrence we have

$$\begin{aligned}
a_n &\geq \sum_{j=i}^{n-1} f(j+1) \prod_{k=j+1}^{n}\left(1 - \frac{(1-2q)}{k}\right) \\
&\geq \sum_{j=i}^{n-1} \frac{1-2q}{e\,(j+1)}\left(\frac{j+1}{i+1}\right)^{1-2q}\frac{j}{n} && \text{(by Lemma 3.3.5)} \\
&\geq \frac{1-2q}{2ne\,(i+1)^{1-2q}} \int_{j=i}^{n-1} x^{1-2q} dx \\
&\geq \frac{1}{4ne\,(i+1)^{1-2q}}\left((n-1)^{2-2q} - i^{2-2q}\right) \\
&\geq \frac{1}{24e}\left(\frac{n+1}{i+1}\right)^{1-2q} - \frac{i}{4ne} .
\end{aligned}$$

■

3.4 The centroid rule

In this section, we analyze the *centroid rule*, that is, our guess for B_0 will be the bit value of a particular vertex called the *centroid*. In order to define the centroid of a tree T, we need some notation.

Recall that the *neighborhood* of a vertex v, that is, the set of vertices in T connected to v, is denoted by $N(v)$. Define $\phi : V(T) \to \mathbb{R}^+$ by

$$\phi(v) \stackrel{\text{def.}}{=} \max_{u \in N(v)} \left| V\left(T^v_{u\downarrow}\right) \right|$$

and define a *centroid* of T by

$$v^* \stackrel{\text{def.}}{=} \arg\min_{v \in V(T)} \phi(v) \, .$$

It is well-known that a tree can have at most two centroids and if there are two of them, then they are connected with an edge. Moreover, $\phi(v^*) \leq \frac{|V(T)|}{2}$ (see, for instance, Harary [43]).

We can now define an estimator $\widehat{b}_{\text{cent}}$ of the root bit as follows: (1) in the root-bit reconstruction problem, $\widehat{b}_{\text{cent}} = B_{v^*}$ is the bit value of an arbitrary centroid v^* of T; (2) in the reconstruction problem from leaf bits, let v^* be a centroid of T, let v° be a leaf closest to v^*, and let $\widehat{b}_{\text{cent}} = B_{v^\circ}$ be the associated bit value.

We call $\widehat{b}_{\text{cent}}$ the *centroid rule*.

Theorem 3.4.1. *Consider the broadcasting problem in a uniform random recursive tree. Denote the probability of error of the centroid rule by*

$$R^{cent}(n,q) = \mathbb{P}\left\{\widehat{b}_{cent} \neq B_0\right\} \, .$$

For the root-bit reconstruction problem,

$$\limsup_{n\to\infty} R^{cent}(n,q) \leq q \quad \textit{for all } q \in [0,1]$$

and

$$\limsup_{n\to\infty} R^{cent}(n,q) \leq \frac{\log 2}{2} \approx 0.34 \quad \textit{for all } q \leq 1/2 \, .$$

For the reconstruction problem from leaf bits,

$$\limsup_{n\to\infty} R^{cent}(n,q) \le 13q \quad \textit{for all } q \in [0,1]\,.$$

Moreover,

$$\limsup_{n\to\infty} R^{cent}(n,q) < 1/2 \quad \textit{for all } q < 1/2\,.$$

In the rest of this section, we prove Theorem 3.4.1.

Assume first that we observe all bit values. First notice that, with high probability, the centroid of a uniform random recursive tree is unique:

Lemma 3.4.2. *If $\mathcal{T}_n$ is a uniform random recursive tree on $n+1$ vertices, then*

$$\mathbb{P}\{\mathcal{T}_n \textit{ has two centroids}\} = \begin{cases} 0 & \textit{if } n \textit{ is even} \\ \frac{4}{n+3} & \textit{if } n \textit{ is odd.} \end{cases}$$

Proof. Recall that the number of recursive trees on $n+1$ vertices equals $n!$ and each of them are equally likely.

Any tree with an odd number of vertices has a unique centroid, so the first half of the statement follows trivially. For odd n, if the tree has two centroid vertices L, R, then there exist two disjoint subtrees of $\frac{1}{2}(n+1)$ vertices, each containing one of the centroids (there exists a *central edge* LR). Call these subtrees *left* and *right* subtree. The left subtree contains vertex L and the right subtree contains vertex R. We may assume, without loss of generality, that the label of L is smaller than the label of R. Then vertex 0 belongs to the left subtree. Moreover, the two subtrees correspond to unique recursive trees of $\frac{1}{2}(n+1)$ vertices, after suitable relabelling that respects the relative ordering of the labels.

To count the number of recursive trees with two centroids, note that there are $\binom{n+1}{\frac{n-1}{2}}$ ways of choosing the labels in the left subtree, excluding the label of L. Then there are are $\frac{n-1}{2}+2$ remaining labels. The label of vertex R is smaller than all its descendants and larger than the label of L. Hence L has the smallest available label and R has the second smallest available label. Once the labels in the left subtree (and therefore in the right subtree) are fixed, there are $\left(\left(\frac{n-1}{2}\right)!\right)^2$ ways of selecting the recursive trees that correspond to each. Hence,

$$\mathbb{P}\{T_n \text{ has two centroids}\} \;=\; \frac{\binom{n+1}{\frac{n-1}{2}} \cdot \left(\left(\frac{n-1}{2}\right)!\right)^2}{n!} \;=\; \frac{4}{n+3}\,.$$

■

Let D_n (or D when it is clear from the context) be the edge distance between the root and v^* in T_n. In Moon [72], it is shown that the probability that the root is a centroid is asymptotically positive. In particular, Moon proves

$$\liminf_{n\to\infty} \mathbb{P}\{\delta_n = 0\} \to 1 - \log 2 \,,$$

where δ_n is the distance between the root and the closest centroid to the root. Hence, for all $q \le 1/2$,

$$\begin{aligned}
\limsup_{n\to\infty} \mathbb{P}\left\{\widehat{b}_{\text{cent}} \neq B_0\right\} &\le \frac{1}{2} - \frac{1}{2}\liminf_{n\to\infty} \mathbb{P}\{D_n = 0\} && (3.4.1)\\
&= \frac{1}{2} - \frac{1}{2}\liminf_{n\to\infty} \mathbb{P}\{\delta_n = 0\} \\
&\qquad \text{(by Lemma 3.4.2)} \\
&= \frac{1}{2} - \frac{1}{2}\left(1 - \log 2\right) \,, && (3.4.2)
\end{aligned}$$

proving the second statement of Theorem 1.2.4.

Given D, the number of changes of the bit value on the path between the root and v^* is Binomial(D, q). Thus,

$$\begin{aligned}
\mathbb{P}\left\{\widehat{b}_{\text{cent}} \neq B_0\right\} &= \mathbb{E}\left[\mathbb{1}_{\{\text{Binomial}(D,q) \text{ is odd}\}}\right] \\
&= \frac{1 - \mathbb{E}\left[(-1)^{\text{Binomial}(D,q)}\right]}{2} \\
&= \frac{1 - \mathbb{E}\left[(1-2q)^{D}\right]}{2} \\
&\le q\mathbb{E}D \,. && (3.4.3)
\end{aligned}$$

We can bound the expectation with direct computation, as in the following Lemma. This implies a linear bound for the risk.

Lemma 3.4.3. *Let $d(i, j)$ denote the distance of vertices i and j in T_n (a random recursive tree). The distance $D_n = d(v^*, 0)$ between the centroid and the root satisfies for all positive integers t*

$$\mathbb{P}\{D_n \ge t\} \le 2(t+1)2^{-t} + o_n(1) \,.$$

In particular,

$$\mathbb{E}D_n \le \frac{9}{2} + o_n(1) \,.$$

Proof. The centroid v^* satisfies $\left|V(T^0_{v^*\downarrow})\right| \geq \frac{n}{2}$ and therefore

$$\mathbb{P}\{v^* = i\} \leq \mathbb{P}\left\{\left|V(T^0_{i\downarrow})\right| \geq \frac{n}{2}\right\} .$$

In order to bound this probability, note that $\left|V(T^0_{i\downarrow})\right|$ evolves according to the number of white balls in a Pólya urn process. Initially the urn contains one white and i black balls. At each time $t = 1, \ldots, n-i$, a ball is drawn randomly from the urn and it is returned in the urn together with an additional ball of the same color. If W_t denotes the number of white balls at time t, then $\left|V(T^0_{i\downarrow})\right|$ has the same distribution as W_{n-i}. By elementary identities for the distribution of Pólya urns (see, e.g., Mahmoud [65, Theorem 3.1]), for any $k \in \{1, \ldots, n\}$,

$$\mathbb{P}\left\{W_{n-i} = k\right\} = \frac{i}{n}\prod_{j=1}^{i-1}\frac{n-k-j+1}{n-j} .$$

Since $k \geq n/2$, when $j > 1$ each term in the product is at most $\frac{1}{2}$. When $j = 1$, the corresponding term is bounded by $\frac{1}{2}\left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right)$. Then,

$$\mathbb{P}\{v^* = i\} \leq \sum_{k \geq n/2}\mathbb{P}\left\{W_{n-i} = k\right\} \leq i2^{-i}\left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) .$$

Since each vertex whose distance to the root is at least t must have index at least t, we immediately obtain

$$\mathbb{P}\{D \geq t\} \leq \mathbb{P}\left\{\exists i \geq t : v^* = i\right\} \leq \sum_{i \geq t} i2^{-i}\left(1 + o_n\left(1\right)\right) = 2(t+1)2^{-t}\left(1 + o_n\left(1\right)\right) .$$

Hence,

$$\mathbb{E}D = \sum_{t \geq 1}\mathbb{P}\{D \geq t\} \leq 3 + \left(1 + o_n\left(1\right)\right)\sum_{t \geq 4} 2(t+1)2^{-t} = \frac{9}{2} + o_n(1) .$$

■

The above lemma implies that $\mathbb{P}\left\{\widehat{b}_{\text{cent}} \neq B_0\right\} \leq \frac{9q}{2}$. This constant is in fact tighter, by the following result of Moon.

Theorem 3.4.4. ([72, Theorem 2.1]) Let δ_n be the depth of the centroid that is closest

to the root. Then for any $n \geq 0$,

$$\mathbb{E}\left[\delta_n\right] = \begin{cases} \frac{n}{n+2} & \text{for } n \text{ odd} \\ \frac{n-1}{n+2} & \text{for } n \text{ even.} \end{cases}$$

It follows that in the root-bit reconstruction problem, the centroid rule satisfies

$$\limsup_{n\to\infty} R^{\text{cent}}(n,q) \leq q \quad \text{for all } q \in [0,1] \,,$$

which is the first statement of Theorem 3.4.1.

To complete the proof of Theorem 3.4.1, it remains to consider the reconstruction problem from leaf bits. Recall that in this case the centroid rule localizes a leaf vertex that is closest to a centroid and guesses the root bit B_0 by the bit value at this leaf.

The missing step for proving the linear upper bound for the asymptotic probability of error is the following lemma, stating that in a uniform random recursive tree, the expected distance of the nearest leaf to the root is bounded.

Lemma 3.4.5. *Consider the uniform random recursive tree $\mathcal{T}_n$ and let*

$$\Delta_n \stackrel{\text{def.}}{=} \min_{i\text{: vertex } i \text{ is a leaf}} d(i,0) \,,$$

where $d(i,0)$ denotes the distance between the root and vertex i. Then, for all n,

$$\mathbb{E}\Delta_n \leq 11 + \mathcal{O}\left(n^{-1-3\log(3/e)}\right) \,.$$

In particular,

$$\limsup_{n\to\infty} \mathbb{E}\Delta_n \leq 11 \,.$$

Proof. We write Δ for Δ_n and start with the decomposition

$$\mathbb{E}\Delta \leq 2 + 3(\log n)\mathbb{P}\{\Delta > 2\} + \sum_{i > 3\log n} \mathbb{P}\{\Delta \geq i\} \,.$$

To bound $\mathbb{P}\{\Delta > 2\}$, we show that with probability at least

$$1 - \frac{3}{\log n}\left(1 + o_n\left(1\right)\right) \,,$$

T_n has a leaf at depth 2. Let A_i be the event that i is a leaf, and B_i the event that $d\left(i,0\right) = 2$. Let $X = \sum_{i=\lceil 2n/3 \rceil}^{n} \mathbb{1}_{A_i \cap B_i}$ be the number of leaves at distance 2 from the

root, among the vertices $\lceil 2n/3 \rceil, \ldots, n$. We bound the mean and variance as follows.

First, note that $A_i = \bigcap_{j=i+1}^{n} \{p_j \neq i\}$ and $B_i = \bigcup_{j=1}^{i-1} \{p_i = j,\ p_j = 0\}$. Then A_i and B_i are independent and

$$\mathbb{P}\{A_i\} = \prod_{j=i+1}^{n} \left(1 - \frac{1}{j}\right) = \frac{i}{n} \quad \text{and} \quad \mathbb{P}\{B_i\} = \sum_{j=1}^{i-1} \left(\frac{1}{i} \cdot \frac{1}{j}\right) = \frac{H_{i-1}}{i},$$

where H_i denotes the sum $\sum_{j=1}^{i} 1/j$. Thus,

$$\mathbb{E}X = \sum_{i=\lceil 2n/3 \rceil}^{n} \left(\frac{i}{n} \cdot \frac{H_{i-1}}{i}\right) = (1 + o_n(1)) \frac{\log n}{3}.$$

We now turn to the calculation of $\mathbb{E}\left\{X^2\right\}$. For $2n/3 \leq i < k \leq n$ we have

$$\mathbb{P}\{A_k | A_i\} = \prod_{l=k+1}^{n} \mathbb{P}\{p_l \neq k | p_l \neq i\} = \prod_{l=k+1}^{n} \left(1 - \frac{1}{l-1}\right) = \frac{k-1}{n-1},$$

so

$$\begin{aligned} \mathbb{P}\{A_k \cap A_i\} &= \mathbb{P}\{A_i\}\,\mathbb{P}\{A_k\}\,\frac{(k-1)\,n}{k\,(n-1)} \\ &= \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right)\mathbb{P}\{A_i\}\,\mathbb{P}\{A_k\}. \end{aligned} \tag{3.4.4}$$

Moreover, $\mathbb{P}\{B_i \cap B_k | A_i \cap A_k\} = \mathbb{P}\{B_i \cap B_k | p_k \neq i\}$, which is equal to

$$\begin{aligned} &\sum_{j=1}^{i-1} \mathbb{P}\{p_i = p_k = j, p_j = 0 | p_k \neq i\} \\ &+ \sum_{j=1}^{i-1} \sum_{\substack{l=1 \\ l \neq j}}^{k-1} \mathbb{P}\{p_i = j, p_j = 0\}\,\mathbb{P}\{p_k = l, p_l = 0 | p_k \neq i\} \\ = \quad &\frac{1}{k-1} \cdot \frac{H_{i-1}}{i} + \sum_{j=1}^{i-1} \sum_{\substack{l=2 \\ l \neq j}}^{k-1} \mathbb{P}\{p_i = j, p_j = 0\}\,\mathbb{P}\{p_k = l, p_l = 0 | p_k \neq i\}. \end{aligned}$$

Since $k \geq 2n/3$, we have

$$\mathbb{P}\{p_k = l, p_l = 0 | p_k \neq i\} = \frac{1}{k-1} \cdot \frac{1}{l} = \left(1 + o\left(\frac{1}{n}\right)\right)\mathbb{P}\{p_k = l, p_l = 0\}.$$

To handle the $j = l$ term, we note that

$$\sum_{j=2}^{i-1} \mathbb{P}\left\{p_i = j, p_j = 0\right\} \mathbb{P}\left\{p_k = j, p_j = 0\right\} = \frac{1}{k \cdot i} \sum_{j=1}^{i-1} \frac{1}{j^2} = \mathcal{O}(1) \cdot \frac{1}{k \cdot i}.$$

It follows that

$$\mathbb{P}\left\{B_i \cap B_k | A_i \cap A_k\right\} = \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) \mathbb{P}\left\{B_i\right\} \mathbb{P}\left\{B_k\right\} + \frac{H_{i-1} - \mathcal{O}(1)}{i(k-1)}, \quad (3.4.5)$$

Then $\mathbb{E}\left[X^2\right]$ is equal to

$$\begin{aligned}
& \sum_{2n/3 \le i \le n} \sum_{2n/3 \le k \le n} \mathbb{P}\left\{A_i \cap B_i \cap A_k \cap B_k\right\} \\
= & \sum_{2n/3 \le i \le n} \mathbb{P}\left\{A_i \cap B_i\right\} + 2 \sum_{2n/3 \le i < k \le n} \mathbb{P}\left\{B_i \cap B_k | A_i \cap A_k\right\} \mathbb{P}\left\{A_i \cap A_k\right\} \\
= & \sum_{2n/3 \le i \le n} \mathbb{P}\left\{A_i \cap B_i\right\} + 2 \sum_{2n/3 \le i < k \le n} \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) \mathbb{P}\left\{A_i \cap B_i\right\} \mathbb{P}\left\{A_k \cap B_k\right\} \\
& + 2 \sum_{2n/3 \le i < k \le n} \mathbb{P}\left\{A_i \cap A_k\right\} \frac{H_{i-1} - \mathcal{O}(1)}{i(k-1)} \\
& \text{(by (3.4.4),(3.4.5) and independence of } A_i, B_i) \\
\le & \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) \left((\mathbb{E}X)^2 + \sum_{2n/3 \le i \le n} \left(\mathbb{P}\left\{A_i \cap B_i\right\} - \mathbb{P}\left\{A_i \cap B_i\right\}^2\right)\right) + o_n(1) \\
\le & (\mathbb{E}X)^2 + \frac{1}{3} \log n \left(1 + o_n(1)\right)
\end{aligned}$$

Recalling that $\mathbb{E}X = \left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right) \frac{\log n}{3}$, it follows that

$$\mathbb{P}\left\{X = 0\right\} \le \frac{\mathrm{Var}\left\{X\right\}}{\left(\mathbb{E}\left\{X\right\}\right)^2} \le \frac{3\left(1 + o_n(1)\right)}{\log n}.$$

It remains to bound $\sum_{i > 3 \log n} \mathbb{P}\{\Delta \ge i\}$. For that, it is enough to bound $\sum_{i > 3 \log n} \mathbb{P}\{d(n,0) \ge i\}$, where $d(n,0)$ is the depth of vertex n. By standard results on uniform random recursive trees (see Devroye [29]), $d(i,0)$ is distributed as $\sum_{j=1}^{i} Y_j$, where the Y_j are independent Bernoulli random variables with $\mathbb{P}\{Y_j = 1\} = 1/j$. We will employ a Chernoff bound for sums of independent Bernoulli

variables in order to bound $\mathbb{P}\{d(n,0) \geq i\}$. Since for all $1 \leq k \leq n$,

$$\mathbb{E}\left[e^{\lambda Y_k}\right] \;=\; 1 + \frac{1}{k}\left(e^{\lambda} - 1\right) \;\leq\; \exp\left(\frac{1}{k}\left(e^{\lambda} - 1\right)\right) ,$$

by independence we have that

$$\begin{aligned} \mathbb{P}\{d(n,0) \geq i\} &\leq \exp(-\lambda i) \cdot \prod_{k \leq n} \exp\left(\frac{1}{k}\left(e^{\lambda} - 1\right)\right) \\ &= \exp\left(H_n\left(e^{\lambda} - 1\right) - \lambda i\right) \\ &= \exp\left(i - H_n - i \log \frac{i}{H_n}\right) , \end{aligned} \tag{3.4.6}$$

if we set $\lambda = \log\left(\frac{i}{H_n}\right)$. By (3.4.6), for all $i > 3H_n$,

$$\mathbb{P}\{\Delta \geq i\} \leq \mathbb{P}\{d(n,0) \geq i\} \leq \exp\left(i - H_n - i \log \frac{i}{H_n}\right) \leq \frac{1}{n} e^{-i \log(3/e)} .$$

Thus,

$$\sum_{i > 3 \log n} \mathbb{P}\{\Delta \geq i\} = \mathcal{O}\left(n^{-1-3\log(3/e)}\right) .$$

Collecting terms, the proof of the lemma is complete. ∎

Let $\widetilde{v}_1$ be a leaf vertex that is closest to the centroid v^* and $\widetilde{v}_2$ be a leaf vertex that is closest to the root. Then the distance of $\widetilde{v}_1$ to the root is bounded as follows:

$$d(\widetilde{v}_1, 0) \leq d(\widetilde{v}_1, v^*) + d(0, v^*) \leq d(0, \widetilde{v}_2) + 2d(0, v^*) = \Delta + 2D ,$$

where $D = d(v^*, 0)$. Hence, by Lemmas 3.4.2, 3.4.4, and 3.4.5, we conclude that

$$\limsup_{n \to \infty} \mathbb{E} d(\widetilde{v}, 0) \leq 13 .$$

Combining the above display with the derivation in (3.4.3), we complete the proof of the third statement in Theorem 3.4.1.

The fourth statement of Theorem 3.4.1 follows with a similar argument as the second statement.

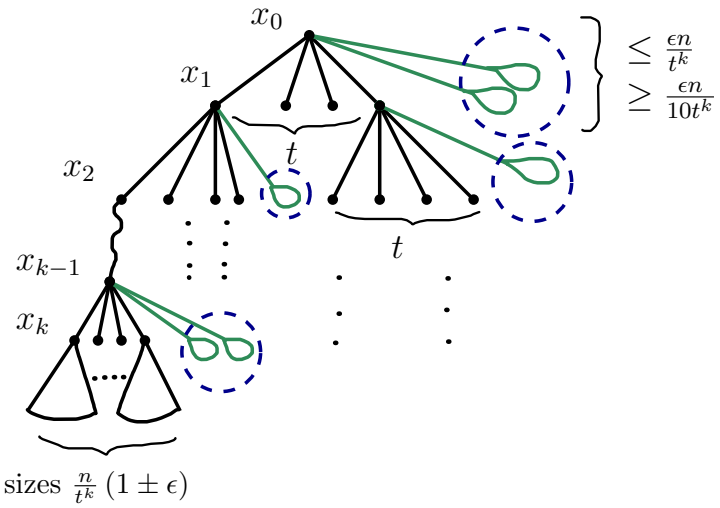


Figure 3.2: A depiction of condition (ii) in the event $E_{t,k}$ that is described in Definition 3.5.2.

3.5 Reconstruction is possible in UA

In this section we prove the following theorem.

Theorem 3.5.1. *Consider the broadcasting problem in a uniform random recursive tree.*

1. *In the root-bit reconstruction problem $R^*(q) < 1/2$ if and only if $q \in [0, 1)$.*
2. *In the reconstruction problem from leaf bits, $R^*(q) < 1/2$ if and only if $q \in [0, 1/2) \cup (1/2, 1)$.*

To that end, we will define an event $E_{t,k}$ and a vertex x_0 given $E_{t,k}$ (see Definition 3.5.2). Our procedure will be the following:

- If $E_{t,k}$ happens, then return the colour of x_0.
- Otherwise, flip a coin.

We will show that $\mathbb{P}\{E_{t,k}\}$ is bounded away from zero and, conditioning on $E_{t,k}$, the colour of x_0 is the correct one with probability larger than $\frac{1}{2}$ (for large n).

Recall the definitions of Aut and $\overline{\text{Aut}}$ from Section 3.2.

Definition 3.5.2. (see also Figure 3.2) Fix integers $t, k > 3$ such that $k \leq t$ and let $\epsilon \in (0, \frac{1}{2t^k})$. Let $E_{t,k}$ denote the event that the following conditions are satisfied:

(i) T_n contains a complete rooted t-ary subtree D of height k (we denote its root-vertex by x_0 and its leaves by $L(D)$).

(ii) Let T be an arbitrary subtree of T_n which is maximal subject to the constraint that $|T \cap D| = 1$. Let v be the unique vertex in $T \cap D$. If $v \in D \setminus L(D)$, then T has at most $\frac{\epsilon n}{t^k}$ vertices and at least $\frac{\epsilon n}{10t^k}$ vertices. If $v \in L(D)$, then T has at most $(1+\epsilon)\frac{n}{t^k}$ vertices and at least $(1-\epsilon)\frac{n}{t^k}$ vertices.

(iii) Let T_1, T_2 be two different subtrees of T_n that are maximal subject to the constraint $|T_1 \cap D| = |T_2 \cap D| = 1$. If T_1 and T_2 intersect D on vertices of the same depth (in D), then they are different as unlabelled rooted trees.

(iv) For all $v \in D \setminus L(D)$, $\mathrm{Aut}(v, T_n) = \overline{\mathrm{Aut}}\left(T^{x_0}_{v\downarrow}\right) = 1$.

We now present the proof of the theorem. The auxiliary Lemmas that it uses are proved subsequently.

Proof. (Theorem 3.5.1) We begin with the first part of Theorem 3.5.1, where we observe all bit values.

Recall that x_0 is the root vertex of D. Consider large enough t, k that are fixed and $k \leq t$. Also, fix $\epsilon \in (0, \frac{1}{2t^k})$. Let p_i be the probability that a vertex at distance i from the root 0 has the same bit value as the root, B_0. Then p_i follows the recurrence

$$p_i \;=\; (1-q)p_{i-1} + q(1-p_{i-1})$$

with initial conditions $p_0 = 1$. Solving the recurrence, we find

$$p_i \;=\; \frac{1}{2}\left(1 + (1-2q)^i\right)\;. \tag{3.5.1}$$

Let $\overline{D}$ denote the set $D \setminus L(D)$. Then we have

$$\begin{aligned}
&\mathbb{P}\left\{B_{x_0} = B_0 \middle| E_{t,k}\right\} \\
&\geq \sum_{i=0}^{k-1} \mathbb{P}\left\{B_{x_0} = B_0 \middle| E_{t,k},\ 0 \in \overline{D},\ d(0,x_0) = i\right\} \mathbb{P}\left\{0 \in \overline{D},\ d(0,x_0) = i \middle| E_{t,k}\right\} \\
&\geq \exp\left(-\frac{k}{t^k}\right)\left(1 - \frac{1}{t^{k-1}}\right)^2 \frac{\sum_{i=0}^{k-1} p_i t^i \left(\prod_{j=1}^{i} \frac{1}{t^j - 1}\right)}{\sum_{i=0}^{k-1} t^i \left(\prod_{j=1}^{i} \frac{1}{t^j-1}\right)} + o_n(1) \\
&\qquad\qquad \text{(by Lemma 3.5.3 below)} \\
&= \exp\left(-\frac{k}{t^k}\right)\left(1 - \frac{1}{t^{k-1}}\right)^2 \frac{\sum_{i=0}^{k-1} \frac{1}{2}\left(1 + (-1)^i (2q-1)^i\right) t^i \prod_{j=1}^{i} \frac{1}{t^j-1}}{\sum_{i=0}^{k-1} t^i \prod_{j=1}^{i} \frac{1}{t^j-1}} + o_n(1)\;. \\
&\qquad\qquad \text{(by equation (3.5.1))}
\end{aligned}$$

For large t,

$$\sum_{i=0}^{k} t^i \prod_{j=1}^{i} \frac{1}{t^j - 1} = 1 + \frac{t}{t-1} + \frac{t^2}{(t-1)(t^2-1)} + \cdots = 2 + \mathcal{O}\left(\frac{1}{t}\right)$$

and

$$\begin{aligned} \sum_{i=0}^{k} (-1)^i \left((2q-1)t\right)^i \prod_{j=1}^{i} \frac{1}{t^j - 1} &= 1 - \frac{(2q-1)t}{t-1} + \frac{(2q-1)^2 t^2}{(t-1)(t^2-1)} + \ldots \\ &= 1 - (2q-1) + \mathcal{O}\left(\frac{1}{t}\right), \end{aligned}$$

and therefore $\liminf_{n\to\infty} \mathbb{P}\left\{B_{x_0} = B_0 \middle| E_{t,k}\right\}$ is lower bounded by

$$\exp\left(-\frac{k}{t^k}\right)\left(1 - \frac{1}{t^{k-1}}\right)^2 \left(\frac{1}{2} + \frac{1}{2} \cdot \frac{1 - (2q-1) + \mathcal{O}\left(\frac{1}{t}\right)}{2 + \mathcal{O}\left(\frac{1}{t}\right)}\right),$$

which in turn is equal to

$$\frac{2-q}{2} + \mathcal{O}\left(\frac{1}{t}\right) > \frac{1}{2}$$

for large t. Since $\liminf_{n\to\infty} \mathbb{P}\{E_{t,k}\} > 0$ by Lemma 3.5.4 below, there exists a choice of the parameters t and k such that the procedure that guesses B_{x_0} if the event $E_{t,k}$ occurs and guesses a random bit otherwise is positively correlated with B_0. This completes the proof of the first statement.

For the second statement of Theorem 3.5.1, we need to show a similar statement as before, when only the leaf bits are observed. The problem here is that even when the tree T_n has the structure described in Definition 3.5.2, the root of the complete t-ary subtree D is not a leaf and therefore its bit value is not observable. However, the root of a random recursive tree is attached to a leaf with probability that is bounded away from zero, as n goes to infinity. In fact, the number of leaves of the root converges in distribution to a Poisson $Po(1)$. This follows by representing the random recursive tree as a uniformly random permutation, where the subtrees hanging from the root represent the cycles of the permutation; then we can apply standard results for uniformly random permutations (see for instance Arratia, Barbour, and Tavaré [5, Theorem 1.3]).

Choose t and k as previously. Let $E'_{t,k}$ be the event that the four conditions listed in Definition 3.5.2 are satisfied and moreover a leaf v of T_n is attached to the root

of the subtree D. Then the following procedure has a probability of error bounded away from $1/2$:

- If $E'_{t,k}$ does not occur then flip a coin.
- If $E'_{t,k}$ occurs, then:
  - ◇ If $q < \frac{1}{2}$, then set $\hat{b} = B_v$.
  - ◇ If $q > \frac{1}{2}$, then set $\hat{b} = 1 - B_v$.

Since $\liminf_{n\to\infty} \mathbb{P}\{E'_{r,k}\} > 0$ and B_{x_0} is positively correlated with B_0, we have that

$$\liminf_{n\to\infty} \mathbb{P}\left\{\hat{b} = B_0\right\} > \frac{1}{2},$$

as desired. ■

It remains to show the two Lemmas that are used in the previous proof.

Lemma 3.5.3. *Let $t, k > 3$ with $k \leq t$ and let $\epsilon \leq \frac{1}{2t^k}$. Let $\overline{D}$ denote the set $D \setminus L(D)$. Then for all $i = 0, 1, \ldots, k-1$, $\liminf_{n\to\infty} \mathbb{P}\left\{0 \in \overline{D},\ d(0, x_0) = i | E_{t,k}\right\}$ is lower bounded by*

$$\exp\left(-\frac{k}{t^k}\right)\left(1 - \frac{1}{t^{k-1}}\right)^2 \frac{r^i \prod_{j=1}^{i}\left(\frac{1}{t^j - 1}\right)}{\sum_{m<k} t^m \prod_{j=1}^{m}\left(\frac{1}{t^j-1}\right)}.$$

Proof. We first lower bound $\mathbb{P}\left\{0 \in \overline{D} | E_{t,k}\right\}$. Notice that under the event $E_{t,k}$, if $0 \notin \overline{D}$, then either $T^0_{1\downarrow}$ contains at least $\left(1 - \frac{1+\epsilon}{t^k}\right) n$ vertices or it contains at most $(1+\epsilon)\frac{n}{t^k}$ vertices. By standard results of the theory of Pólya urns (Eggenberger and Pólya [36]), $\left|T^0_{1\downarrow}\right|$ converges, in distribution, to a uniform random variable on $[0, 1]$.

Hence,

$$\begin{aligned}\mathbb{P}\left\{0 \in \overline{D} | E_{t,k}\right\} &= 1 - \frac{2(1+\epsilon)}{t^k} + o_n(1) \\ &\geq 1 - \frac{1}{t^{k-1}} + o_n(1),\end{aligned}$$

by the assumptions on ϵ, t. It remains to derive a lower bound for

$$\mathbb{P}\left\{d(0, x_0) = i \middle| 0 \in \overline{D},\ E_{t,k}\right\} = \sum_{v \in \overline{D}: d(v, x_0) = i} \mathbb{P}\left\{0 = v \middle| 0 \in \overline{D},\ E_{t,k}\right\}.$$

Recall the definition of the function $\lambda(u)$ from (3.2.1) and that, given an unlabeled tree, the probability that vertex u is the root is proportional to $\lambda(u)$. Hence, defining for $i = 0, 1, \ldots, k-1$

$$W_i = \sum_{v \in \overline{D}: d(v,x_0)=i} \frac{\lambda(v)}{\lambda(x_0)} ,$$

we have that

$$\mathbb{P}\left\{ d(0,x_0) = i \middle| 0 \in \overline{D},\ E_{t,k} \right\} = \frac{W_i}{\sum_{j=0}^{k-1} W_j} .$$

Under the event $E_{t,k}$, for all $u \in \overline{D}$ we have $\mathrm{Aut}\,(u, T_n) = 1$ and $\overline{\mathrm{Aut}}\left(T_{u\downarrow}^{x_0}\right) = 1$. Hence, if $x_i \in \overline{D}$ has depth i in D and $x_0 x_1 \ldots x_i$ is the path in D that connects it to the root of D, then for all $j = 1, \ldots, i-1$,

$$\begin{aligned}
\frac{\lambda(x_{j+1})}{\lambda(x_j)} &= \frac{\left|T_{x_{j+1}\downarrow}^{x_j}\right|}{\left|T_{x_j\downarrow}^{x_{j+1}}\right|} \\
&\geq \frac{\frac{n}{t^{j+1}}(1-\epsilon)}{n - \frac{n}{t^{j+1}}(1-\epsilon)} \quad \left(\text{since } \left|T_{x_{j+1}\downarrow}^{x_j}\right| \geq t^{k-j-1} \cdot \frac{n}{t^k}(1-\epsilon)\, n\right) \\
&= \frac{1}{t^{j+1}-1}\left(1 - \frac{\epsilon t^{j+1}}{t^{j+1}-1+\epsilon}\right) \\
&\geq \frac{1}{t^{j+1}-1}\left(1 - \frac{1}{t^k}\right) ,
\end{aligned}$$

since $\epsilon \leq \frac{1}{2t^k}$. Thus,

$$\begin{aligned}
\frac{\lambda(x_i)}{\lambda(x_0)} &\geq \left(1 - \frac{1}{t^k}\right)^k \prod_{j=1}^{i} \left(\frac{1}{t^j - 1}\right) \\
&\geq \left(1 - \frac{k}{t^k}\right) \prod_{j=1}^{i} \left(\frac{1}{t^j - 1}\right) \\
&\geq \left(1 - \frac{1}{t^{k-1}}\right) \prod_{j=1}^{i} \left(\frac{1}{t^j - 1}\right) ,
\end{aligned}$$

since $k \leq t$. Similarly,

$$\frac{\lambda(x_{j+1})}{\lambda(x_j)} \leq \frac{\frac{n}{t^{j+1}}(1+\epsilon) + t^k}{n - \frac{n}{t^{j+1}}(1+\epsilon) - t^k}$$

$$\leq \left(\frac{1}{t^{j+1}-1}\right)\left(1+\frac{1}{t^k}\right)+o_n(1)$$

and

$$\begin{aligned}\frac{\lambda(x_i)}{\lambda(x_0)} &\leq \left(1+\frac{1}{t^k}\right)^k \prod_{j=1}^{i}\left(\frac{1}{t^j-1}\right)+o_n(1)\\ &\leq \exp\left(\frac{k}{t^k}\right)\prod_{j=1}^{i}\left(\frac{1}{t^j-1}\right)+o_n(1)\,.\end{aligned}$$

Putting these estimates together, we obtain the statement of the lemma. ∎

Finally, we show that the probability of $E_{t,k}$ is bounded away from zero.

Lemma 3.5.4. *Let $t,k>3$. Then $\liminf_{n\to\infty}\mathbb{P}\{E_{t,k}\}>0$.*

Proof. Fix t and k. After the insertion of $M \stackrel{\text{def.}}{=} \frac{t^{k+1}-1}{t-1}$ vertices, the probability that the uniform random recursive tree $\mathcal{T}_M$ is isomorphic to a complete t-ary tree D of height k is positive and depends only on t and k. Let us call this event E_I. This event clearly implies condition (i) in Definition 3.5.2.

In what follows, we work on the conditional probability space defined by the event E_I. Let $u_1,\ldots,u_{t^k}$ be the vertices of height k in D and $v_1,\ldots,v_{M'}$ be the rest of the vertices in D (note for the sequel that $M' \stackrel{\text{def.}}{=} M-t^k$). For every such vertex v_i (or u_j accordingly), we define $\overline{T}^{x_0}_{v_i\downarrow}$ to be the maximal subtree of $T^{x_0}_{v_i\downarrow}$ that intersects D only at v_i. Then the vector $\left(\left|\overline{T}^{x_0}_{v_1\downarrow}\right|,\ldots,\left|\overline{T}^{x_0}_{v_m\downarrow}\right|,\left|\overline{T}^{x_0}_{u_1\downarrow}\right|,\ldots,\left|\overline{T}^{x_0}_{u_{t^k}\downarrow}\right|\right)$ behaves as a standard Pólya urn with M colors, initialized with one ball of each color. As n goes to infinity, the proportions of the balls of each color converge to a Dirichlet distribution $\mathrm{Dir}(1,\ldots,1)$.

We define Ω as the set that contains all $(x_1,\ldots,x_M)\in\mathbb{R}^M$ that satisfy the conditions

(i) $\sum_{i=1}^{M} x_i = 1$

(ii) $x_{t^k+1},\ldots,x_M \in \left(\frac{\epsilon}{t^k}-\frac{\epsilon}{t^{3k}},\frac{\epsilon}{t^k}\right)$

(iii) $x_1,\ldots,x_{t^k-1} \in \left(t^{-k}\left(1-\frac{M'\epsilon}{t^k}\right)-\frac{\epsilon}{t^{3k}},t^{-k}\left(1-\frac{M'\epsilon}{t^k}\right)+\frac{\epsilon}{t^{3k}}\right)$.

The set Ω satisfies condition (ii) in Definition 3.5.2. Let us verify this. The proportions $x_{t^k+1},\ldots,x_M$ do so obviously. For the proportions $x_1,\ldots,x_{t^k-1}$, we have that $\frac{M'}{t^k}+\frac{1}{t^{2k}}\leq 1$ and $\frac{1}{t^{2k}}-\frac{M'}{t^k}\leq 1$. Hence each one of them belongs in the right range

$\left(\frac{1}{t^k}(1-\epsilon), \frac{1}{t^k}(1+\epsilon)\right)$. It remains to verify that $x_{t^k} = 1 - \sum_{i \neq t^k} x_i$ is also necessarily in the same range. For that, it is enough to notice that x_{t^k} is at most

$$1 - \frac{M'\epsilon}{t^k} + \frac{M'\epsilon}{t^{3k}} - \frac{t^k - 1}{t^k}\left(1 - \frac{M'\epsilon}{t^k}\right) + \frac{\epsilon(t^k-1)}{t^{3k}} \leq \frac{1}{t^k}(1+\epsilon)$$

and at least

$$1 - \frac{M'\epsilon}{t^k} - \frac{t^k - 1}{t^k}\left(1 - \frac{M'\epsilon}{t^k}\right) - \frac{\epsilon(t^k-1)}{t^{3k}} \geq \frac{1}{t^k}(1-\epsilon) \ .$$

Combining the previous two paragraphs, we have that

$$\mathbb{P}\left\{(ii) \,|E_I\right\} \ \geq \ \int_\Omega dDir(1,\ldots,1) > 0 \, .$$

Therefore, conditions (i) and (ii) jointly hold with probability bounded away from zero.

Conditioning on event E_I, condition (iii) of Definition 3.5.2 clearly holds with probability converging to one, since t, k are fixed.

Finally, we check condition (iv), assuming that $(i), (ii), (iii)$ hold. We abbreviate by A the event that conditions $(i), (ii), (iii)$ hold. Let $v \in \overline{D}$ and $S_1, \ldots, S_k$ be the maximal rooted subtrees of T_n that are contained in $\overline{T}^{x_0}_{v\downarrow}$ and whose roots are connected with an edge to v. Denote by n_v the number of vertices of the subtree $\overline{T}^{x_0}_{v\downarrow}$. By property (ii), we have that $n_v = \Omega(n)$.

We call an S_iS_j-*conflict* the event where $S_i \cong S_j$ as rooted unlabelled trees. Moreover, we denote by $C_i^{(n_v)}$ the number of indices j such that $|S_j| = i$. We would like to show that

$$\liminf_{n\to\infty} \mathbb{P}\left\{\text{there is no } S_iS_j\text{-conflict} \,|A\right\} > 0 \, .$$

To this end, it suffices that the quantity

$$\liminf_{n_v\to\infty}\left(\mathbb{P}\left\{\forall i \leq \sqrt{n_v},\ C_i^{(n_v)} \leq 1 | A\right\} - \mathbb{P}\left\{\exists S_iS_j\text{-conflict where } |S_i| > \sqrt{n_v} | A\right\}\right)$$

is strictly positive.

Claim 3.5.5. *For any $j > \sqrt{n_v}$,*

$$\mathbb{P}\left\{C_j^{(n_v)} \geq 2 | A\right\} \leq \mathbb{P}\left\{C_{\sqrt{n_v}}^{(n_v)} \geq 2\right\} + \mathcal{O}\left(n_v^{-3/2}\right) \ .$$

Proof. The multiset $\{|S_1|, \ldots, |S_k|\}$ is distributed as the multiset of cycle lengths of a uniformly random permutation of $\left|T_{v\downarrow}^{x_0}\right| - 1$. In view of Arratia, Barbour, and Tavaré [5, Lemma 1.2],

$$\mathbb{P}\left\{C_j^{(n_v)} = m | A\right\} = \frac{1}{j^m m!} \sum_{\ell=0}^{\lfloor n_v/j \rfloor - m} \frac{(-1)^\ell}{j^\ell \ell!}\,. \tag{3.5.2}$$

Then

$$\begin{aligned}
\mathbb{P}\left\{C_j^{(n_v)} \geq 2 | A\right\} &= \sum_{m\geq 2} \frac{1}{j^m m!} \sum_{\ell=0}^{\lfloor n_v/j \rfloor - m} \frac{(-1)^\ell}{j^l \ell!} \\
&< \sum_{m\geq 2} \left(\frac{1}{\sqrt{n_v}^m m!} \left(\sum_{\ell=0}^{\lfloor \sqrt{n_v} \rfloor - m} \frac{(-1)^\ell}{j^\ell \ell!} - \sum_{\ell=\lfloor n_v/j \rfloor - m+1}^{\lfloor \sqrt{n_v} \rfloor - m} \frac{(-1)^\ell}{j^\ell \ell!} \right) \right) \\
&= \mathbb{P}\left\{C_{\sqrt{n_v}}^{(n_v)} \geq 2 | A\right\} + \sum_{m\geq 2} \frac{1}{\sqrt{n_v}^m m!} \sum_{\ell=\lfloor n_v/j \rfloor - m+1}^{\lfloor \sqrt{n_v} \rfloor - m} \frac{(-1)^{\ell+1}}{j^\ell \ell!} \\
&\leq \mathbb{P}\left\{C_{\sqrt{n_v}}^{(n_v)} \geq 2 | A\right\} + \frac{1}{n_v} \sum_{m\geq 2} \frac{1}{m!} \sum_{\ell=1}^{\lfloor \sqrt{n_v} \rfloor} \frac{1}{j^\ell \ell!} \\
&\leq \mathbb{P}\left\{C_{\sqrt{n_v}}^{(n_v)} \geq 2 | A\right\} + \frac{e}{n_v} \left(\frac{1}{\sqrt{n_v}} + \frac{1}{n_v} + \ldots \right) \\
&= \mathbb{P}\left\{C_{\sqrt{n_v}}^{(n_v)} \geq 2 | A\right\} + \mathcal{O}\left(n_v^{-3/2}\right)\,,
\end{aligned}$$

and the claim follows. ∎

Let $(Z_1, \ldots, Z_{n_v})$ be a vector of independent Poisson variables Z_i with mean $\frac{1}{i}$. It is known (see for instance [5, Lemma 1.4]) that

$$\mathrm{d}_{TV}\left(\left(C_1^{(n_v)}, \ldots, C_b^{(n_v)}\right), (Z_1, \ldots, Z_b)\right) \leq \frac{2b}{n_v + 1}, \tag{3.5.3}$$

where d_{TV} denotes the total variation distance. Then,

$$\begin{aligned}
\mathbb{P}\left\{\forall i \leq \sqrt{n_v},\ C_i^{(n_v)} \leq 1\right\} &\geq \prod_{i\leq \sqrt{n_v}} \mathbb{P}\left\{\mathrm{Poisson}\left(\frac{1}{i}\right) \leq 1\right\} - \frac{2\sqrt{n_v}}{n_v+1} \quad \text{(by (3.5.3))} \\
&= \prod_{i\leq \sqrt{n_v}} \exp\left(-\frac{1}{i}\right)\left(1 + \frac{1}{i}\right) - \frac{2\sqrt{n_v}}{n_v+1}
\end{aligned}$$

$$\begin{aligned} &\geq \exp\left(-\log\left(\sqrt{n_v}+1\right)\right)\left(\sqrt{n_v}+1\right)-\frac{2\sqrt{n_v}}{n_v+1} \\ &= 1-\frac{2\sqrt{n_v}}{n_v+1} \end{aligned}$$

and

$$\begin{aligned} &\mathbb{P}\left\{\exists S_i S_j\text{-conflict with } |S_i|>\sqrt{n_v}|A\right\} \\ &\leq \sum_{k>\sqrt{n_v}} \mathbb{P}\left\{C_k^{(n_v)}\geq 2|A\right\} \\ &\leq \sum_{k>\sqrt{n_v}} \mathbb{P}\left\{C_{\sqrt{n_v}}^{(n_v)}\geq 2|A\right\}+\mathcal{O}\left(n_v^{-1/2}\right) \quad \text{(by Claim 3.5.5)} \\ &\leq n_v \sum_{m=2}^{m=\sqrt{n_v}} \frac{1}{\sqrt{n_v}^m m!} \sum_{\ell=0}^{\lfloor\sqrt{n_v}\rfloor-m} \frac{(-1)^\ell}{\sqrt{n_v}^\ell \ell!}+\mathcal{O}\left(n_v^{-1/2}\right) \quad \text{(by (3.5.2))} \\ &\leq \mathcal{O}\left(n_v^{-1}\right)+\mathcal{O}\left(n_v^{-1/2}\right). \end{aligned}$$

By independence and since t, k are fixed the claim then holds for all $v \in \overline{D}$ with constant probability. We may now conclude that for large n, for all $v \in \overline{D}$ we have $\overline{\text{Aut}}\left(T_{v\downarrow}^{x_0}\right)=1$ with positive probability.

Finally, the constraints on the subtree sizes from (ii) imply that any automorphism of T_n restricts to an automorphism of D. It follows that when (ii) holds, for any $v \in D \setminus L(D)$, any automorphism ϕ of T_n with $\phi(v) \neq v$ must permute the set of subtrees of T_n which intersect $L(D)$ in exactly one vertex. It follows that if (i),(ii) and (iii) all hold, then no such automorphism can exist, i.e., $\text{Aut}(v, T_n)=1$. ∎

3.6 Linear preferential attachment

In this section we consider *linear preferential attachment trees*. In this model,

$$\mathbb{P}\{i \sim j\}=\frac{D_j^+(i-1)+\beta}{\sum_{k=0}^{i-1} D_k^+(i-1)+\beta},$$

where $\beta>0$ is a parameter and $D_j^+(i-1)$ denotes the outdegree of vertex j at time $i-1$. We will extend the results of the previous section in the linear preferential model.

3.6.1 The majority rule in preferential attachment

Just like in the case of uniform attachment, the asymptotic probability of error is bounded by a constant multiple of q both in the root-bit reconstruction problem and in the reconstruction problem from leaf bits. Interestingly, the break-down point of the majority rule is not at $q = 1/4$ anymore. The critical value depends on the parameter β and it is given by

$$\gamma(\beta) = \min\left(\frac{\beta+1}{4\beta}, \frac{1}{2}\right) .$$

Note that this value is always larger than $1/4$ and therefore the majority rule has a better break-down point than in the case of uniform attachment, for all values of β. Moreover, when $\beta \leq 1$, the majority vote has a nontrivial probability of error for all values of $q < 1/2$.

Theorem 3.6.1. *Consider the broadcasting problem in the linear preferential attachment model with parameter $\beta > 0$. For both the root-bit reconstruction problem and the reconstruction problem from leaf bits, there exists a constant c such that*

$$\limsup_{n\to\infty} R^{maj}(n,q) \leq cq \quad \textit{for all } q \in [0,1] .$$

Moreover,

$$\limsup_{n\to\infty} R^{maj}(n,q) < 1/2 \quad \textit{if } q \in [0,\gamma(\beta)) ,$$

and

$$\limsup_{n\to\infty} R^{maj}(n,q) = 1/2 \quad \textit{if } q \in [\gamma(\beta),1/2] .$$

The proof of the linear bound follows exactly the same steps as the corresponding proof of Theorem 3.3.1, only here Lemmas 3.6.2, 3.6.3 take the role of Lemmas 3.3.5, 3.3.6, 3.3.8. Note that the bound on $\mathrm{Var}(\delta_j)$ in (3.3.7) that is used in the proof of Lemma 3.3.7, is similar in the preferential attachment model (see for instance [30, Theorem 2.7, Section 7]). Hence we omit this proof.

For the other two assertions, the proof follows the same steps as in Section 3.3.4, and Section 3.3.3, only now the replacement matrix encodes the expected change of the *weight* of each of the four categories of nodes. The weight of a set A of vertices

is defined by $\beta|A| + \sum_{v\in A} D_v^+$. We obtain the following matrix:

$$\begin{pmatrix} -\beta q & \beta(1-q) & \beta q & \beta q \\ \beta+1 & 1 & 0 & 0 \\ \beta q & \beta q & -\beta q & \beta(1-q) \\ 0 & 0 & \beta+1 & 1 \end{pmatrix}$$

The eigenvalues of the transpose of this matrix are $\beta+1, \beta+1-2\beta q, -\beta, -\beta$ and then [50, Theorems 3.23, 3.24] can be immediately applied as before, in combination with Lemmas 3.6.2 and 3.6.3.

All that is left is to prove analogues of Lemmas 3.3.5, 3.3.6, 3.3.8 in the preferential attachment model.

The difference with respect to uniform attachment is that, in the preferential attachment model, knowing N_i at time $n-1$ is not enough to determine the probability that N_i increases in the next time step. This is because the vertices counted by N_i do not only have connections between them but also with other external vertices. So we introduce the *weight* w_j, for $j \geq i$. Recall that $\widetilde{T}_i$ denotes the maximal size subtree of $T^0_{i\downarrow}$ with root i and all other vertices unmarked. Also $N_i = |\widetilde{T}_i|$. As in Section 3.3.6, Y_j denotes the number of vertices $u \in \widetilde{T}_i$, such that $u \leq j$. Moreover, $\mathcal{Y}_j$ is the set of vertices $u \in \widetilde{T}_i$ such that $u \leq j$. Then

$$w_j \overset{\text{def.}}{=} \sum_{v\in\mathcal{Y}_j} \left(D_v^+(j) + \beta\right) = \beta \cdot Y_j + \sum_{v\in\mathcal{Y}_j} D_v^+(j) \ . \tag{3.6.1}$$

Similarly to Lemmas 3.3.3 and 3.3.4, it is easy to see that for any positive $a, b < 1$,

$$e^{-1}\left(\frac{n+1-\alpha}{i+1-\alpha}\right)^b \leq \prod_{j=i}^{n-1}\left(1+\frac{b}{j+1-\alpha}\right) \leq e\left(\frac{n+1-\alpha}{i+1-\alpha}\right)^b \ . \tag{3.6.2}$$

Recall that in order to show the linear upper bound for the risk, we may assume that $q < 1/8$ (otherwise a linear bound holds trivially).

Lemma 3.6.2. *Let $r = 1 - \frac{2\beta q}{\beta+1}$, $r_1 = \frac{1}{\beta+1}$, and assume that $q < 1/8$. Then for any $i \leq n$,*

$$\frac{3\beta}{8(\beta+1)e}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r - \frac{3\beta}{4e(\beta+1)} \leq \mathbb{E}[N_i] \leq \frac{\beta e}{1+\beta}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r + \frac{1}{\beta+1}$$

and

$$\mathbb{E}\left[N_i^2\right] \leq \frac{4}{(1+\beta)^2}\left(\beta e + \beta e^2(1+\beta) + re^2(1+\beta)^2\right)\left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r}.$$

Proof. We have

$$\mathbb{E}\left[w_n | w_{n-1}\right] = w_{n-1}\left(1 + \frac{2q + (1+\beta)(1-2q)}{n(\beta+1)-1}\right),$$

since if $\mathcal{Y}_n$ is chosen by the new vertex n, then with probability $2q$ we have $w_n = w_{n-1} + 1$ (n is marked) and with probability $1-2q$ we have $w_n = w_{n-1} + 1 + \beta$ (n is unmarked). Taking expectations and expanding the resulting recurrence, we have

$$\mathbb{E}\left[w_n\right] = \beta \prod_{j=i}^{n-1}\left(1 + \frac{r}{j+1-r_1}\right) \leq \beta e\left(\frac{n+1-r_1}{i+1-r_1}\right)^r \tag{3.6.3}$$

by (3.6.2) and the fact that $w_i = \beta$. Similarly,

$$\mathbb{E}\left[w_n\right] \geq \beta e^{-1}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r. \tag{3.6.4}$$

For the second moment, we use a similar argument as in for the first moment and obtain

$$\begin{aligned}
\mathbb{E}\left[w_n^2 | w_{n-1}^2\right] &= w_{n-1}^2 + \frac{(1-2q)\,w_{n-1}}{(\beta+1)\,n-1}\left(2(1+\beta)\,w_{n-1} + (1+\beta)^2\right) \\
&\quad + \frac{2qw_{n-1}}{(\beta+1)\,n-1}\left(2w_{n-1}+1\right) \\
&\leq w_{n-1}^2\left(1 + \frac{2r}{n-r_1}\right) + \frac{w_{n-1}(\beta+1)\,r}{n-r_1}.
\end{aligned}$$

Taking expectations and setting $f(j) = r(\beta+1)\frac{\mathbb{E}\left[w_{j-1}\right]}{j-r_1}$, we obtain the following recurrence for $a_n \stackrel{\text{def}}{=} \mathbb{E}\left[w_n\right]$:

$$\begin{aligned}
a_n &\leq a_{n-1}\left(1 + \frac{2r}{n-r_1}\right) + f(n) \\
&\leq \beta \prod_{j=i}^{n-1}\left(1 + \frac{2r}{j+1-r_1}\right) + \sum_{j=i}^{n-2} f(j+1) \prod_{k=j+1}^{n-1}\left(1 + \frac{2r}{k+1-r_1}\right) + f(n) \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(since } w_i = \beta\text{)}
\end{aligned}$$

$$
\begin{aligned}
&\leq \beta e \left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r} + \sum_{j=i}^{n-1} \frac{r\beta e^2(1+\beta)}{j+1-r_1}\left(\frac{j+1-r_1}{i+1-r_1}\right)^{r}\left(\frac{n+1-r_1}{j+1-r_1}\right)^{2r} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{(by (3.6.2) and (3.6.3))} \\
&= \left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r}\left(\beta e + r\beta e^2(1+\beta)(i+1-r_1)^r \sum_{j=i}^{n-1}(j+1-r_1)^{-r-1}\right) \\
&\leq \beta e \left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r} + \left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r} + r\beta e^2(1+\beta)(i+1-r_1)^r \\
&\qquad \cdot \left(\int_i^n (x+1-r_1)^{-r-1}\,dx + \frac{1}{(i+1-r_1)^{r+1}}\right) \\
&\leq \left(\beta e + \beta e^2(1+\beta) + re^2(1+\beta)^2\right)\left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r}.
\end{aligned}
$$

By (3.6.3) and $Y_n = \frac{1}{1+\beta} + \frac{w_n}{1+\beta}$, we have

$$
\mathbb{E}[Y_n] \leq \frac{\beta e}{1+\beta}\left(\frac{n+1-r_1}{i+1-r_1}\right)^{r} + \frac{1}{\beta+1}. \tag{3.6.5}
$$

Moreover,

$$
\mathbb{E}[Y_n | Y_{n-1}, w_{n-1}] = Y_{n-1} + \frac{(1-2q)\,w_{n-1}}{(\beta+1)(n-r_1)}.
$$

Taking expectations and expanding the resulting recurrence we obtain the following

$$
\begin{aligned}
\mathbb{E}[Y_n] &= \frac{(1-2q)}{\beta+1}\sum_{j=i}^{n-1}\frac{\mathbb{E}[w_j]}{j+1-r_1} \\
&\geq \frac{(1-2q)}{\beta+1}\sum_{j=i}^{n-1}\frac{\beta e^{-1}\left(\frac{j+1-r_1}{i+1-r_1}\right)^r}{j+1-r_1} \quad \text{by (3.6.4)} \\
&= \frac{\beta(1-2q)}{e(\beta+1)(i+1-r_1)^r}\sum_{j=i}^{n-1}(j+1-r_1)^{r-1} \\
&\geq \frac{\beta(1-2q)}{e(\beta+1)(i+1-r_1)^r}\int_i^{n-1}(x+1-r_1)^{r-1}\,dx \\
&\geq \frac{3\beta}{4e(\beta+1)(i+1-r_1)^r}\left((n-r_1)^r - (i+1-r_1)^r\right) \\
&\qquad\qquad\qquad \left(\text{since } q < \frac{1}{8} \text{ and } \frac{1-2q}{r} \geq \frac{3}{4}\right)
\end{aligned}
$$

$$\geq \frac{3\beta}{8e\left(\beta+1\right)}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r - \frac{3\beta}{4e\left(\beta+1\right)}$$

The upper bound for the second moment follows by $Y_n = \frac{1}{1+\beta} + \frac{w_n}{1+\beta}$, hence $\mathbb{E}\left[Y_n^2\right] \leq \frac{4\mathbb{E}\left[w_n^2\right]}{(1+\beta)^2}$, and the previous computations. ∎

Denote by $\overline{Y}_j$ the number of leaf vertices in $\mathcal{Y}_j$.

Lemma 3.6.3. *Let $r = 1 - \frac{2\beta q}{\beta+1}$, $r_1 = \frac{1}{\beta+1}$, and assume that $q < 1/8$. For any $i \leq n$,*

$$\frac{\beta}{8e\left(\beta+1\right)}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r - \frac{3\beta}{8e\left(\beta+1\right)} \leq \mathbb{E}\left[\overline{N}_i\right] \leq \frac{\beta e}{1+\beta}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r + \frac{1}{\beta+1}$$

and

$$\mathbb{E}\left[\overline{N}_i^2\right] \leq \frac{4}{\left(1+\beta\right)^2}\left(\beta e + \beta e^2(1+\beta) + re^2(1+\beta)^2\right)\left(\frac{n+1-r_1}{i+1-r_1}\right)^{2r}.$$

Proof. The upper bounds clearly hold by the fact that $\overline{Y}_j \leq Y_j$ and Lemma 3.6.2. Let us denote by $\overline{w}_j$ the weight of the set of leaves in $\mathcal{Y}_j$ (recall the weight function defined in (3.6.1)). Notice that $\overline{w}_n = \beta\overline{Y}_n$. Hence,

$$\begin{aligned}
\mathbb{E}\left[\overline{Y}_n|\overline{Y}_{n-1}, w_{n-1}, \overline{w}_{n-1}\right] &= \overline{Y}_{n-1} + \frac{1-2q}{\left(\beta+1\right)\left(n-r_1\right)}\left(w_{n-1} - \overline{w}_{n-1}\right) \\
&= \overline{Y}_{n-1} + \frac{1-2q}{\left(\beta+1\right)\left(n-r_1\right)}\left(w_{n-1} - \beta\overline{Y}_{n-1}\right) \\
&= \overline{Y}_{n-1}\left(1 - \frac{\beta\left(1-2q\right)}{\left(\beta+1\right)\left(n-r_1\right)}\right) + \frac{w_{n-1}\left(1-2q\right)}{\left(\beta+1\right)\left(n-r_1\right)}.
\end{aligned}$$

We can assume that $i \leq n-2$, since otherwise the result can be confirmed immediately. Let $f\left(n\right) = \frac{1-2q}{(\beta+1)(n-r_1)}\mathbb{E}\left[w_{n-1}\right]$. Then, $a_n \stackrel{\text{def}}{=} \mathbb{E}\left[\overline{Y}_n\right]$ satisfies

$$\begin{aligned}
a_n &= a_{n-1}\left(1 - \frac{\beta\left(1-2q\right)}{\left(\beta+1\right)\left(n-r_1\right)}\right) + f\left(n\right) \\
&\geq \sum_{j=i}^{n-2} f\left(j+1\right) \prod_{k=j+1}^{n-1}\left(1 - \frac{\beta\left(1-2q\right)}{\left(\beta+1\right)\left(k+1-r_1\right)}\right) \\
&\geq \sum_{j=i}^{n-2} \frac{\beta\left(1-2q\right)}{e\left(\beta+1\right)\left(j+1-r_1\right)}\left(\frac{j+1-r_1}{i+1-r_1}\right)^r \frac{j+1-r_1}{n+1-r_1} \qquad \text{(by (3.6.4))}
\end{aligned}$$

$$\geq \frac{\beta(1-2q)}{e(\beta+1)(n+1-r_1)}(i+1-r_1)^{-r}\int_i^{n-2}(x+1-r_1)^r\,dx$$
$$\geq \frac{3\beta}{8e(\beta+1)}\left(\frac{1}{3}\left(\frac{n+1-r_1}{i+1-r_1}\right)^r-1\right).$$

■

3.6.2 The centroid rule in PA

For the performance of the centroid rule, we have the following analog of Theorem 3.4.1 for linear preferential attachment trees. The proof parallels the arguments of Section 3.4.

Theorem 3.6.4. *Consider the broadcasting problem in the linear preferential attachment model with fixed parameter $\beta > 0$. For both the root-bit reconstruction problem and the reconstruction problem from leaf bits, there exists a constant c such that*

$$\limsup_{n\to\infty} R^{cent}(n,q) \leq cq \quad \textit{for all } q \in [0,1].$$

In particular, $c \leq \frac{\beta}{\beta+1}$ in the root-bit reconstruction problem and $c \leq 2+\frac{2\beta}{\beta+1}+\frac{3(\beta+1)}{\beta}e^{\frac{3\beta+1}{\beta+1}}$ in the reconstruction problem from leaf bits. Moreover,

$$\limsup_{n\to\infty} R^{cent}(n,q) < 1/2 \quad \textit{for all } q \leq 1/2.$$

To show the theorem, we work as in Section 3.4. For brevity, we omit overlapping arguments and we only fill in the missing points. Recall that the estimator $\widehat{b}_{\text{cent}}$ is the bit value of the centroid v^* of the tree. In case there are two centroids we pick one uniformly at random. However, the probability of this event tends to zero, see Wagner and Durant [33, Lemma 15].

We can bound the expectation of D (that is, the distance between the root and v^*) with direct computation.

Lemma 3.6.5. *Consider the broadcasting problem in the linear preferential attachment model with fixed parameter $\beta > 0$. Let D be the distance between the centroid v^* and the root. Then for all positive integers $t > 2$,*

$$\mathbb{P}\{D \geq t\} \leq 8(t+1)2^{-t}$$

and

$$\mathbb{E}D \leq 9\,.$$

Proof. We follow the argument of Lemma 3.4.3 and keep the same notation. Here again we have an urn, but a ball is picked proportionally to its weight (defined in (3.6.1)). Assume $i > 2$ and let $r_1 = \frac{1}{\beta+1}$. Then

$$\begin{aligned}
\mathbb{P}\left\{W_{n-i} = k\right\} &= \binom{n-i}{k-1} \frac{\prod_{j=1}^{k-1}(j\beta + j - 1)\prod_{j=i}^{n-k}(j\beta + j - 1)}{\prod_{j=i+1}^{n}(j\beta + j - 1)} \\
&= \binom{n-i}{k-1} \frac{\prod_{j=1}^{k-1}(j - r_1)\prod_{j=i}^{n-k}(j - r_1)}{\prod_{j=i+1}^{n}(j - r_1)} \\
&\leq \frac{(i - r_1)\prod_{j=0}^{k-2}(n-i-j)\prod_{j=1}^{n-k}(j - r_1)}{\prod_{j=1}^{n}(j - r_1)} \\
&\leq \frac{(i - r_1)\prod_{j=1}^{n-k}(j - r_1)}{\prod_{j=1}^{n-i-k+2}(j - r_1)\prod_{j=n-i+2}^{n}(j - r_1)} \\
&= \frac{(i - r_1)\prod_{j=n-i-k+3}^{n-k}(j - r_1)}{\prod_{j=n-i+2}^{n}(j - r_1)} \\
&= \frac{i - r_1}{n-i+2-r_1}\left(\frac{n-i+3-k-r_1}{n-i+3-r_1}\cdots\frac{n-k-r_1}{n-r_1}\right) \\
&\leq \frac{i - r_1}{n-i+2-r_1}\left(\frac{n-k-r_1}{n-r_1}\right)^{i-2} \\
&\qquad \text{(since } \frac{n-k-r_1-x}{n-r_1-x} \text{ is decreasing with } x\text{)} \\
&\leq \frac{i - r_1}{n-i+2-r_1}\left(\frac{n/2-r_1}{n-r_1}\right)^{i-2} \qquad \text{(since } k \geq n/2\ \text{)} \\
&= \frac{i - r_1}{n-i+2-r_1}\left(\frac{1}{2}\right)^{i-2} \\
&\leq \frac{2i}{n}\left(\frac{1}{2}\right)^{i-2} \qquad \text{(since } i \leq (n+2)/2\ \text{)}
\end{aligned}$$

Hence

$$\mathbb{P}\{v^* = i\} \leq \sum_{k \geq n/2} \mathbb{P}\left\{W_{n-i} = k\right\} \leq 4i2^{-i}$$

and

$$\mathbb{P}\{D \geq t\} \leq \mathbb{P}\left\{\exists i \geq t : v^* = i\right\} \leq 4 \sum_{i \geq t} i 2^{-i} = 8(t+1)2^{-t} .$$

Consequently,

$$\mathbb{E}D = \sum_{t \geq 1} \mathbb{P}\{D \geq t\} \leq 3 + 4 \sum_{t \geq 4} 2(t+1)2^{-t} = 9 .$$

■

However, tighter results are already existent.

Theorem 3.6.6. (Wagner and Durant [33, Theorem 9, Theorem 11]) Let δ_n be the depth of the centroid closest to the root and L_n be its label at time n. Then

$$\lim_{n\to\infty} \mathbb{E}\left[\delta_n\right] = \frac{\beta}{\beta+1} \text{ and } \lim_{n\to\infty} \mathbb{P}\left\{L_n = 0\right\} = 1 - \beta\left(2^{1/(1+\beta)} - 1\right) .$$

We may combine the above theorem and equation (3.4.1) as follows.

$$\begin{aligned} \limsup_{n\to\infty} \mathbb{P}\left\{\widehat{b}_{\text{cent}} \neq B_0\right\} &\leq \frac{1}{2} - \frac{1}{2} \liminf_{n\to\infty} \mathbb{P}\{D = 0\} \\ &= \frac{1}{2} - \frac{1}{2} \liminf_{n\to\infty} \mathbb{P}\{\delta_n = 0\} \\ &= \frac{1}{2} - \frac{1}{2}\left(1 - \beta\left(2^{1/(1+\beta)} - 1\right)\right) < \frac{1}{2} . \end{aligned}$$

The rest follows directly by combining Theorem 3.6.6 and equation (3.4.3).
To show Theorem 3.6.4 in the case of reconstruction from leaf-bits, we prove the following lemma.

Lemma 3.6.7. $\mathbb{P}\left\{\Delta > 2\right\} \leq \beta(2\beta+1)(1+\beta)^{-\frac{1}{\beta+1}} e^{\frac{\beta-1}{\beta+1}} n^{-\frac{1}{\beta+1}} + \mathcal{O}\left(\frac{1}{n}\right)$.

Proof. Denote by N_1 the set of vertices $i \leq \lceil n/2 \rceil$ at distance one from the root. For vertex u such that $\lceil n/2 \rceil < u \leq n$, we write Y_u for the indicator that u attaches to a vertex in N_1 (say it attaches to u_1) and also an independent Bernoulli $\left(\frac{D^+_{u_1}(\lceil n/2 \rceil)}{D^+_{u_1}(u-1)}\right)$ coin flip is successful. We add the last condition so that

$$\mathbb{P}\{Y_u Y_v = 1\} = \mathbb{P}\{Y_u = 1\}\mathbb{P}\{Y_v = 1\} ,$$

for any u, v such that $v > u > \lceil n/2 \rceil$. We write X_u for the indicator that u is connected with an edge to N_1 and is a leaf. Then, $X_u = Y_u Z_u$, where Z_u is the

indicator that no vertex $t > u$ attaches to u. Moreover,

$$\mathbb{P}\{Z_v = 1 | Y_u Y_v = 1\} = \mathbb{P}\{Z_v = 1 | Y_v = 1\}$$

when $v > u$, and

$$\mathbb{P}\{X_u X_v = 1\} = \mathbb{P}\{Z_u = 1 | Z_v Y_u Y_v = 1\} \mathbb{P}\{Z_v = 1 | Y_u Y_v = 1\} \mathbb{P}\{Y_u Y_v = 1\} .$$

Combining the previous observations, we have that for $v > u$ the covariance $\mathrm{Cov}\left(X_u X_v\right)$ is equal to

$$\mathbb{P}\{Z_v = 1 | Y_u Y_v = 1\} \mathbb{P}\{Y_u Y_v = 1\} \left(\mathbb{P}\{Z_u = 1 | Z_v Y_u Y_v = 1\} - \mathbb{P}\{Z_u = 1 | Y_u = 1\}\right) .$$

But

$$\begin{aligned} \mathbb{P}\left\{Z_u = 1 | Y_u Y_v Z_v = 1\right\} &= \frac{u}{u + 1 - \frac{1}{\beta+1}} \cdots \frac{v-2}{v - 1 - \frac{1}{\beta+1}} \cdot \frac{v - \frac{\beta}{\beta+1}}{v} \cdots \frac{n - 1 - \frac{\beta}{\beta+1}}{n-1} \\ &\leq \frac{u}{u + \frac{\beta}{\beta+1}} \cdots \frac{v-2}{v - 2 + \frac{\beta}{\beta+1}} \cdot \frac{v}{v + \frac{\beta}{\beta+1}} \cdots \frac{n-1}{n - 1 + \frac{\beta}{\beta+1}} \end{aligned}$$

and

$$\mathbb{P}\left\{Z_u = 1 | Y_u = 1\right\} = \frac{u}{u + 1 - \frac{1}{\beta+1}} \cdots \frac{n-1}{n - \frac{1}{\beta+1}} = \frac{u}{u + \frac{\beta}{\beta+1}} \cdots \frac{n-1}{n - 1 + \frac{\beta}{\beta+1}} .$$

Therefore, for $w\left(N_1\right) = \sum_{i \in N_1} \left(D_i^+ \left(\lceil n/2 \rceil\right) + \beta\right)$, we have

$$\begin{aligned} \mathrm{Cov}\left(X_u X_v\right) &\leq \left(1 - \frac{v-1}{v - 1 + \frac{\beta}{\beta+1}}\right) \cdot \mathbb{E}\left\{\frac{w\left(N_1\right)}{\left(\beta+1\right) u - 1}\right\}^2 \\ &\leq \frac{2}{n} \cdot \mathbb{E}\left\{\frac{w\left(N_1\right)}{\left(\beta+1\right) u - 1}\right\}^2 \\ &\leq \frac{8}{n^3 \left(\beta+1\right)^2} \cdot \mathbb{E}\left\{w\left(N_1\right)\right\}^2 , \end{aligned}$$

since $v > u \geq n/2 + 1$. Moreover,

$$\mathbb{E} X_u = \left(\frac{u}{u + \frac{\beta}{\beta+1}} \cdots \frac{n-1}{n - 1 + \frac{\beta}{\beta+1}}\right) \cdot \mathbb{E}\left\{\frac{w\left(N_1\right)}{\left(\beta+1\right) u - 1}\right\}$$

$$\geq \quad e^{-\frac{\beta}{\beta+1}} \cdot \mathbb{E}\left\{\frac{w\left(N_1\right)}{\left(\beta+1\right)n}\right\} .$$

Then, by Chebyshev's inequality and the previous bounds,

$$\begin{aligned}\mathbb{P}\left\{\sum_{i>\lceil n/2\rceil} X_i = 0\right\} &\leq \frac{\sum\limits_{i\geq\lceil n/2\rceil}\mathrm{Var}(X_i) + \sum\limits_{\substack{i\neq j\\ i\geq\lceil n/2\rceil}}\mathrm{Cov}(X_iX_j)}{\left(\sum\limits_{i\geq\lceil n/2\rceil}\mathbb{E}X_i\right)^2}\\ &\leq \frac{e^{\frac{2\beta}{\beta+1}}\left(\beta+1\right)}{\mathbb{E}\left\{w\left(N_1\right)\right\}} + \mathcal{O}\left(\frac{1}{n}\right) .\end{aligned}$$

Moreover $\mathbb{E}\left\{w\left(N_1\right)\right\} \geq \frac{(1+\beta)^{\frac{\beta+2}{\beta+1}}}{e\beta(2\beta+1)} \cdot n^{\frac{1}{\beta+1}}$, which concludes the proof. To see that, notice that its expectation satisfies the recurrence

$$a_n \;=\; \alpha_{n-1}\left(1+\frac{1/\left(\beta+1\right)}{n-1/\left(\beta+1\right)}\right)+b_n$$

where $b_n \;=\; \frac{\beta\mathbb{E}[w(0)]_n}{n(1+\beta)-1}$, hence

$$\begin{aligned}a_n &\geq b_1\cdot\prod_{i=1}^{n-1}\left(1+\frac{1/\left(\beta+1\right)}{i+1-1/\left(\beta+1\right)}\right)\\ &\geq \frac{b_1}{e}\left(\frac{n+1-\frac{1}{\beta+1}}{2-\frac{1}{\beta+1}}\right)^{\frac{1}{\beta+1}} \quad \text{by (3.6.2)}\\ &\geq \frac{b_1}{e}\left(\frac{\beta+1}{2\beta+1}\right)^{\frac{1}{\beta+1}} n^{\frac{1}{\beta+1}}\\ &= \frac{(1+\beta)^{\frac{\beta+2}{\beta+1}}}{e\beta(2\beta+1)}\cdot n^{\frac{1}{\beta+1}}\end{aligned}$$

■

By Lemma 3.6.7 and [31, Theorem 6.50],

$$\mathbb{E}\Delta \;=\; \sum_{i=0}^{n-1}\mathbb{P}\left\{\Delta > i\right\}$$

$$\begin{aligned} &\leq 2+\beta(2\beta+1)(1+\beta)^{-\frac{1}{\beta+1}}e^{\frac{\beta-1}{\beta+1}}+\sum_{i>n^{1/(\beta+1)}}\mathbb{P}\left\{\Delta>i\right\}+o_n\left(1\right)\\ &= 2+\beta(2\beta+1)(1+\beta)^{-\frac{1}{\beta+1}}e^{\frac{\beta-1}{\beta+1}}+o_n\left(1\right). \end{aligned}$$

As in Section 3.4 and using Theorem 3.6.6, Lemma 3.6.7, we have that, if $\widetilde{v}$ is a leaf vertex that is closest to the centroid v^*, then

$$\limsup_{n\to\infty}\mathbb{E}d(\widetilde{v},0)\leq\mathbb{E}[\Delta+2D]\leq 2+\frac{2\beta}{\beta+1}+\beta(2\beta+1)(1+\beta)^{-\frac{1}{\beta+1}}e^{\frac{\beta-1}{\beta+1}}.$$

This completes the proof of the first part of Theorem 3.6.4 for the reconstruction problem from leaf bits. The second part follows from the fact that the root is the centroid of the tree with probability bounded away from zero, combined with the fact that the expected distance of the nearest leaf is bounded, as shown above.

3.7 Reconstruction is possible in PA

In this section, we show an equivalent of Theorem 3.5.1 in the case of linear preferential attachment.

Theorem 3.7.1. *Consider the broadcasting problem in the linear preferential attachment model.*

1. *In the root-bit reconstruction problem $R^*(q) < 1/2$ if and only if $q \in [0,1)$.*
2. *In the reconstruction problem from leaf bits, $R^*(q) < 1/2$ if and only if $q \in [0,1/2)\cup(1/2,1)$.*

Let us denote by $[n]$ the set $\{0,\dots,n\}$ and by Π_n the set of all permutations of $[n]$. We call *history* a pair (T_n,π) where T_n is a recursive tree and $\pi\in\Pi_n$. Intuitively, a history is a recursive tree on which we have applied a vertex relabelling according to π. We assume that we observe $\mathcal{T}_n$ after applying a uniformly random permutation on its vertex labels. We will denote the outcome of this process by $\tilde{t}_n$.

Given $\tilde{t}_n$ and a vertex $u\in\tilde{t}_n$, we define the set $hist(\tilde{t}_n,u)$ as the set that contains all histories (T_n,π) such that

(i) T_n and $\tilde{t}_n$ are isomorphic as unlabelled rooted trees, where T_n has root 0 and $\tilde{t}_n$ has root u. In other words, T_n is compatible to $\tilde{t}_n$ with root u.

(ii) π gives a graph isomorphism between T_n and $\tilde{t}_n$ that maps 0 to u (that is, $\pi(0)=u$).

We denote by $recur(\tilde{t}_n, u)$ the set of recursive trees that are compatible with $\tilde{t}_n$, if we assume that u is the root.

In the proof of [23, Theorem 8], the authors show that

$$\sum_{(T_n,\pi)\in hist(\tilde{t}_n,u)} \mathbb{P}\{T_n\} \;=\; |Aut(\tilde{t}_n)| \left(\frac{1}{|Aut(\tilde{t}_n,u)|} \sum_{t_n\in recur(\tilde{t}_n,u)} \mathbb{P}\{\mathcal{T}_n = t_n\} \right) ,$$

where $Aut(\tilde{t}_n)$ is the set of graph automorphisms of $\tilde{t}_n$ and $Aut(\tilde{t}_n, u)$ is the set of vertices that are equivalent to u in $\tilde{t}_n$ under graph automorphism. Notice that the second factor is equal to the likelihood of vertex u being the root vertex and the first factor only depends on the topology of $\tilde{t}_n$. Hence, maximizing the expression

$$\sum_{(T_n,\pi)\in hist(\tilde{t}_n,u)} \mathbb{P}\{T_n\} \tag{3.7.1}$$

is equivalent to maximizing the maximum likelihood function (in the problem where one guesses the tree's root vertex).

Well-known tree models such as uniform attachment, preferential attachment [1], and the *diffusion model* can be shown to satisfy the condition of *shape exchangeability* [23, Theorem 4]. This condition requires that, given $\tilde{t}_n$, any compatible recursive tree has the same probability. It is a well-known fact that the number of histories beginning from a vertex u in a tree of $n+1$ vertices is equal to

$$n! \prod_{i\in[n]\setminus u} \frac{1}{|T^u_{i\downarrow}|} \,. \tag{3.7.2}$$

It is easy to show this claim, but one can also confirm in [23, Proposition 5]. This implies that, in the aforementioned models, the maximum likelihood estimator for root finding is the minimizer of $\prod_{i\in[n]} |T^u_{i\downarrow}|$. (Note that, as a consequence, the $\overline{\mathrm{Aut}}\left(T^u_{v\downarrow}\right)$ factor in 3.2.1 can actually be dropped.)

The model of linear preferential attachment that we study does not satisfy shape-exchangeability. However, it *almost* does, and that will help us prove theorem 3.7.1.

Lemma 3.7.2. *Assume the linear preferential attachment model. Then for any two neigh-*

[1] Let us make precise that the preferential attachment model in [23] uses the total degrees of the vertices and not the outdegrees, so it is not identical to the model that we study here.

bouring vertices $u, u' \in \tilde{t}_n$, we have

$$\frac{\lambda(u)}{\lambda(u')} \;=\; \frac{(deg(u)-1+\beta)(deg(u')-2+\beta)}{(deg(u)-2+\beta)(deg(u')-1+\beta)} \cdot \frac{\left|T^{u'}_{u\downarrow}\right|}{\left|T^{u}_{u'\downarrow}\right|}\,, \tag{3.7.3}$$

where $\lambda(u)$ is the likelihood that u is the root vertex.

Proof. By [23, Theorem 8], we have that

$$\lambda(u) \;=\; \sum_{(T_n,\pi)\in hist(\tilde{t}_n,u)} \mathbb{P}\left\{T_n\right\}$$

and a similar statement also holds for $\lambda(u')$, recalling (3.7.1). Observe that all recursive trees that are compatible to $\tilde{t}_n$ with root a fixed vertex u have the same probability, since the set of outdegrees of $\tilde{t}_n$ remains the same. Hence

$$\lambda(u) \;=\; \frac{n!p^*(u)}{\prod_{i\in[n]\setminus u}|T^u_{i\downarrow}|}\,,$$

where $p^*(u)$ is the probability of an arbitrary $T_n \in recur(\tilde{t}_n, u)$ and using (3.7.2).

Now one needs to observe that $\prod_{i\in[n]}|T^u_{i\downarrow}|$ and $\prod_{i\in[n]}|T^{u'}_{i\downarrow}|$ only defer in $|T^{u'}_{u\downarrow}|$ and $|T^u_{u'\downarrow}|$. Moreover, $p^*(u)$ is equal to

$$\frac{\prod\limits_{\substack{i\notin\{u,u'\}:\\ deg(u)>1}} (deg(i)-2+\beta)_{deg(u)-1}}{\prod_{i=2}^{n}\left(i\,(1+\beta)-1\right)} \cdot (deg(u)-1+\beta)_{deg(u)-1}\left(deg(u')-2+\beta\right)_{deg(u')-1}$$

where the notation $(a)_k$ stands for the falling factorial $a(a-1)\ldots(a-k+1)$ and $deg(\,\cdot\,)$ denotes the total degree of a vertex. A similar expression holds for the value $p^*(u')$. The proof is concluded upon noticing that the first factor is the same in both $p^*(u)$ and $p^*(u')$. ∎

We would like to repeat the proof of section 3.5, this time for the linear preferential attachment model. We first show an analogue of Lemma 3.5.3, using the same notation. We will consider the event $\tilde{E}_{t,k}$ that is similar to the event $E_{t,k}$ defined in Definition 3.5.2.

Definition 3.7.3. We define $\tilde{E}_{t,k}$ as the event where all conditions in Definition 3.5.2 are satisfied, with one difference: the quantities $\overline{Aut}(\,\cdot\,)$ are not restricted.

Lemma 3.7.4. *Assume the linear preferential attachment model with parameter $\beta > 0$. Fix $k, t > 3$ with $k \leq t$ and let $\epsilon \leq \frac{1}{2t^k}$. Then for all $i = 0, 1, \ldots, k-1$,*

$$\liminf_{n\to\infty} \mathbb{P}\left\{0 \in \overline{D},\ d\left(0, x_0\right) = i | \tilde{E}_{t,k}\right\}$$

is lower bounded by

$$\left(\left(1 - \frac{1+\epsilon}{t^k}\right)^{\frac{\beta}{\beta+1}} - \left(\frac{1+\epsilon}{t^k}\right)^{\frac{\beta}{\beta+1}}\right) \exp\left(-\frac{k}{t^k}\right)\left(1 - \frac{1}{t^{k-1}}\right) \frac{t^i \prod_{j=1}^{i}\left(\frac{1}{t^j - 1}\right)}{\sum_{m<k} t^m \prod_{j=1}^{m}\left(\frac{1}{t^j-1}\right)}\ .$$

Proof. Given that the event $\tilde{E}_{t,k}$ is true, if $0 \notin \overline{D}$ then either $T^0_{1\downarrow}$ contains at least $n\left(1 - \frac{1+\epsilon}{t^k}\right)$ vertices or it contains at most $(1+\epsilon)\, n/t^k$ vertices. The vector $\left(w\left(T^1_{0\downarrow}\right), w\left(T^0_{1\downarrow}\right)\right)$ behaves as a Pólya urn with diagonal replacement matrix

$$\begin{pmatrix} 1+\beta & 0 \\ 0 & 1+\beta \end{pmatrix}$$

and initial vector $(1+\beta, \beta)$. Hence $\frac{1}{n}\left|T^0_{1\downarrow}\right|$ converges in distribution to a random variable $Beta\left(\frac{\beta}{1+\beta}, 1\right)$ (see [65, Theorem 3.2] or [51, Theorem 1.4] for the more general setting). We have that

$$\mathbb{P}\left\{Beta\left(\frac{\beta}{\beta+1}, 1\right) \leq \frac{1+\epsilon}{t^k}\right\} = \left(\frac{1+\epsilon}{t^k}\right)^{\frac{\beta}{\beta+1}}$$

and

$$1 - \mathbb{P}\left\{Beta\left(\frac{\beta}{\beta+1}, 1\right) \leq 1 - \frac{1+\epsilon}{t^k}\right\} = 1 - \left(1 - \frac{1+\epsilon}{t^k}\right)^{\frac{\beta}{\beta+1}}$$

consequently

$$\mathbb{P}\left\{0 \in \overline{D} | \tilde{E}_{t,k}\right\} \ = \ \left(1 - \frac{1+\epsilon}{t^k}\right)^{\frac{\beta}{\beta+1}} - \left(\frac{1+\epsilon}{t^k}\right)^{\frac{\beta}{\beta+1}} + o_n\left(1\right)\ .$$

It remains to derive a lower bound for

$$\mathbb{P}\left\{d(0, x_0) = i \middle| 0 \in \overline{D},\ E_{t,k}\right\} = \sum_{v \in \overline{D} : d(v, x_0) = i} \mathbb{P}\left\{0 = v \middle| 0 \in \overline{D},\ E_{t,k}\right\}\ .$$

Recall that $\lambda(u)$ denotes the likelihood that vertex u is the root. As in Section 3.5, we define for $i = 0, 1, \ldots, k-1$,

$$W_i = \sum_{v \in \overline{D}: d(v,x_0)=i} \frac{\lambda(v)}{\lambda(x_0)} .$$

We have that

$$\mathbb{P}\left\{d(0,x_0) = i \middle| 0 \in \overline{D},\ \tilde{E}_{t,k}\right\} = \frac{W_i}{\sum_{j=0}^{k-1} W_j} .$$

Assume that $x_i \in \overline{D}$ has depth i in D and $x_0 x_1 \ldots x_i$ is the path in D that connects it to the root of D. By Lemma 3.7.3 and working as in the proof of Lemma 3.5.3, we have for all $j = 1, \ldots, i-1$,

$$\begin{aligned} \frac{\lambda(x_{j+1})}{\lambda(x_j)} &\geq \frac{1}{t^j - 1}\left(1 - \frac{1}{t^k}\right) \cdot \frac{(deg(x_{j+1}) - 1 + \beta)(deg(x_j) - 2 + \beta)}{(deg(x_{j+1}) - 2 + \beta)(deg(x_j) - 1 + \beta)} \\ &= \left(1 + \mathcal{O}\left(\frac{1}{\gamma(n)}\right)\right) \cdot \frac{1}{t^j - 1}\left(1 - \frac{1}{t^k}\right) , \end{aligned}$$

since we have conditioned on $\tilde{E}_{t,k}$, where $\frac{1}{\gamma(n)}$ is a random variable defined as follows:

$$\frac{1}{\gamma(n)} \stackrel{def}{=} \max_{i \in D} \left| 1 - \frac{deg_n(i) - 1 + \beta}{deg_n(i) - 2 + \beta} \right| .$$

Moreover,

$$\frac{\lambda(x_i)}{\lambda(x_0)} \geq \left(1 + \mathcal{O}\left(\frac{1}{\gamma(n)}\right)\right) \cdot \left(1 - \frac{1}{t^{k-1}}\right) \prod_{j=1}^{i} \left(\frac{1}{t^j - 1}\right) .$$

Similarly,

$$\frac{\lambda(x_{j+1})}{\lambda(x_j)} \leq \left(1 + \mathcal{O}\left(\frac{1}{\gamma(n)}\right)\right) \cdot \left(\frac{1}{t^j - 1}\right)\left(1 + \frac{1}{t^k}\right)$$

and

$$\frac{\lambda(x_i)}{\lambda(x_0)} \leq \left(1 + \mathcal{O}\left(\frac{1}{\gamma(n)}\right)\right) \cdot \exp\left(\frac{k}{t^k}\right) \prod_{j=1}^{i} \left(\frac{1}{t^j - 1}\right) .$$

Given that $\tilde{E}_{t,k}$ and $0 \in \overline{D}$ holds, we have $\frac{1}{\gamma(n)} \to 0$ almost surely. Putting these estimates together, we obtain the statement of the lemma. ∎

Lemma 3.7.5. *Let $t, k > 3$. Then $\liminf_{n\to\infty} \mathbb{P}\left\{\tilde{E}_{t,k}\right\} > 0$.*

Proof. Let $M = (t-1)^{-1}(t^{k+1}-1)$ and $M' = M - t^k$. First, observe that $\mathcal{T}_M$ is a t-ary tree of depth k with positive probability. This implies condition (i) of Definition 3.5.2.

Conditioning on the above event, we continue with condition (ii) of definition 3.5.2. We use again the set Ω that contains all $(x_1, \ldots, x_M) \in \mathbb{R}^M$ such that

(i) $\sum_{i=1}^{M} x_i = 1$

(ii) $x_{t^k+1}, \ldots, x_M \in \left(\frac{\epsilon}{t^k} - \frac{\epsilon}{t^{3k}}, \frac{\epsilon}{t^k}\right)$

(iii) $x_1, \ldots, x_{t^k-1} \in \left(t^{-k}\left(1 - \frac{M'\epsilon}{t^k}\right) - \frac{\epsilon}{t^{3k}}, t^{-k}\left(1 - \frac{M'\epsilon}{t^k}\right) + \frac{\epsilon}{t^{3k}}\right)$.

Ω is an open set inside the $(M-1)$-simplex that satisfies condition (ii) of Definition 3.5.2 (if we view each x_i as the proportion of vertices contained in the i-th subtree). Now let $u_1, \ldots, u_{t^k}$ be the vertices of height k in D and $v_1, \ldots, v_{M'}$ be the rest of the vertices in D. For every such vertex v_i (or u_j accordingly), we define $\overline{T}^{x_0}_{v_i\downarrow}$ to be the maximal subtree of $T^{x_0}_{v_i\downarrow}$ that intersects D in only at v_i. Then the random vector

$$\left(w\left(\overline{T}^{x_0}_{v_1\downarrow}\right), \ldots, w\left(\overline{T}^{x_0}_{v_{M'}\downarrow}\right), w\left(\overline{T}^{x_0}_{u_1\downarrow}\right), \ldots, w\left(\overline{T}^{x_0}_{u_{t^k}\downarrow}\right)\right)$$

behaves as a Pólya urn with M colors and diagonal replacement matrix $(1+\beta)\cdot I_M$, initialized with $t+\beta$ for the first M' colours and with $1+\beta$ for the rest t^k. As n goes to infinity, the vector

$$\frac{1}{n}\left(w\left(\overline{T}^{x_0}_{v_1\downarrow}\right), \ldots, w\left(\overline{T}^{x_0}_{v_{M'}\downarrow}\right), w\left(\overline{T}^{x_0}_{u_1\downarrow}\right), \ldots, w\left(\overline{T}^{x_0}_{u_{t^k}\downarrow}\right)\right)$$

converges to a distribution with positive density in the $(M-1)$-simplex. The same does the vector

$$\frac{1}{n}\left(\left|\overline{T}^{x_0}_{v_1\downarrow}\right|, \ldots, \left|\overline{T}^{x_0}_{v_{M'}\downarrow}\right|, \left|\overline{T}^{x_0}_{u_1\downarrow}\right|, \ldots, \left|\overline{T}^{x_0}_{u_{t^k}\downarrow}\right|\right) .$$

Consequently, $\mathbb{P}\{(ii)|(i)\} \geq \mathbb{P}\{\Omega|(i)\} > 0$.

Condition (iii) holds asymptotically almost surely, just by the convergence of $\frac{1}{n}\left(\left|\overline{T}^{x_0}_{v_1\downarrow}\right|, \ldots, \left|\overline{T}^{x_0}_{v_{M'}\downarrow}\right|, \left|\overline{T}^{x_0}_{u_1\downarrow}\right|, \ldots, \left|\overline{T}^{x_0}_{u_{t^k}\downarrow}\right|\right)$ (the probability that the proportions in two different coordinates is the same tends to zero).

The first part of condition (iv) also holds with positive probability, arguing as in the proof of Lemma 3.5.4. ∎

The rest of the proof of Theorem 3.7.1 follows through as in Section 3.5.

Bibliography

[1] Muad Abu-Ata and Feodor F Dragan. Metric tree-like structures in real-world networks: an empirical study. *Networks*, 67(1):49–68, 2016.

[2] Aaron B Adcock, Blair D Sullivan, and Michael W Mahoney. Tree-like structure in large social and information networks. In *2013 IEEE 13th International Conference on Data Mining*, pages 1–10. IEEE, 2013.

[3] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Internet: Diameter of the world-wide web. *nature*, 401(6749):130, 1999.

[4] Stefan Arnborg, Derek G Corneil, and Andrzej Proskurowski. Complexity of finding embeddings in a k-tree. *SIAM Journal on Algebraic Discrete Methods*, 8(2):277–284, 1987.

[5] Richard Arratia, Andrew D. Barbour, and Simon Tavaré. *Logarithmic Combinatorial Structures: a Probabilistic Approach*, volume 1. European Mathematical Society, 2003.

[6] Krishna B. Athreya and Samuel Karlin. Limit theorems for the split times of branching processes. *Journal of Mathematics and Mechanics*, 17(3):257–277, 1967.

[7] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9(Mar):485–516, 2008.

[8] Danielle Smith Bassett and ED Bullmore. Small-world brain networks. *The neuroscientist*, 12(6):512–523, 2006.

[9] Kevin Bello and Jean Honorio. Computationally and statistically efficient learning of causal bayes nets using path queries. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances*

in Neural Information Processing Systems, volume 31. Curran Associates, Inc., 2018.

[10] Pavel M Bleher, Jean Ruiz, and Valentin A Zagrebnov. On the purity of the limiting Gibbs state for the Ising model on the Bethe lattice. *Journal of Statistical Physics*, 79(1-2):473–482, 1995.

[11] Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[12] Hans L Bodlaender, Pål Grønås Drange, Markus S Dregi, Fedor V Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.

[13] Daniel G Brown and Jakub Truszkowski. Fast error-tolerant quartet phylogeny algorithms. In *Annual Symposium on Combinatorial Pattern Matching*, pages 147–161. Springer, 2011.

[14] Daniel G Brown and Jakub Truszkowski. Fast phylogenetic tree reconstruction using locality-sensitive hashing. In *International Workshop on Algorithms in Bioinformatics*, pages 14–29. Springer, 2012.

[15] Sébastien Bubeck, Luc Devroye, and Gábor Lugosi. Finding Adam in random growing trees. *Random Structures & Algorithms*, 50(2):158–172, 2017.

[16] Sébastien Bubeck, Ronen Eldan, Elchanan Mossel, and Miklós Rácz. From trees to seeds: on the inference of the seed from large trees in the uniform attachment model. *Bernoulli*, 23(4A):2887–2916, 2017.

[17] Sébastien Bubeck, Elchanan Mossel, and Miklós Rácz. On the influence of the seed graph in the preferential attachment model. *IEEE Transactions on Network Science and Engineering*, 2(1):30–39, 2015.

[18] T Tony Cai, Weidong Liu, and Harrison H Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2):455–488, 2016.

[19] Thalia E Chan, Michael PH Stumpf, and Ann C Babtie. Network inference and hypotheses-generation from single-cell transcriptomic data using multivariate information measures. *bioRxiv*, 2016.

[20] Venkat Chandrasekaran and Michael I Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181–E1190, 2013.

[21] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of inference in graphical models. *arXiv preprint arXiv:1206.3240*, 2012.

[22] C Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

[23] Harry Crane and Min Xu. Inference on the history of a randomly growing tree. *arXiv preprint arXiv:2005.08794*, 2020.

[24] Nicolas Curien, Thomas Duquesne, Igor Kortchemski, and Ioan Manolescu. Scaling limits and influence of the seed graph in preferential attachment trees. *arXiv preprint arXiv:1406.1758*, 2014.

[25] Gautamd Dasarathy, Aarti Singh, Maria-Florina Balcan, and Jong H Park. Active learning algorithms for graphical model selection. In *Artificial Intelligence and Statistics*, pages 1356–1364, 2016.

[26] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Optimal phylogenetic reconstruction. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, pages 159–168. ACM, 2006.

[27] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Evolutionary trees and the Ising model on the Bethe lattice: a proof of Steel's conjecture. *Probability Theory and Related Fields*, 149(1-2):149–189, 2011.

[28] L. Devroye and G. Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, New York, 2000.

[29] Luc Devroye. Applications of the theory of records in the study of random trees. *Acta Informatica*, 26(1-2):123, 1988.

[30] Michael Drmota. The height of increasing trees. *Annals of Combinatorics*, 12(4):373–402, 2009.

[31] Michael Drmota. *Random trees: an Interplay between Combinatorics and Probability*. Springer-Verlag Wien, 2009.

[32] Mathias Drton and Marloes H Maathuis. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4:365–393, 2017.

[33] Kevin Durant and Stephan Wagner. On the centroid of increasing trees. *Discrete Mathematics & Theoretical Computer Science*, 21 4, 2019.

[34] David Edwards, Gabriel CG De Abreu, and Rodrigo Labouriau. Selecting high-dimensional mixed graphical models using minimal aic or bic forests. *BMC Bioinformatics*, 11(1):18, 2010.

[35] B. Efron and C. Stein. The jackknife estimate of variance. *The Annals of Statistics*, 9:586–596, 1981.

[36] F. Eggenberger and G. Pólya. über die statistik verketteter vorgäge. *Zeitschrift für Angewandte Mathematik und Mechanik*, 3:279—289, 1923.

[37] William Evans, Claire Kenyon, Yuval Peres, and Leonard Schulman. Broadcasting on trees and the Ising model. *The Annals of Applied Probability*, 10(2):410–433, 2000.

[38] Uriel Feige and Mohammad Mahdian. Finding small balanced separators. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, pages 375–384. ACM, 2006.

[39] Morten Frydenberg. Marginalization and collapsibility in graphical interaction models. *The Annals of Statistics*, pages 790–805, 1990.

[40] John Haigh. The recovery of the root of a tree. *Journal of Applied Probability*, 7(1):79–88, 1970.

[41] Mohammad Taghi Hajiaghayi and Mahdi Hajiaghayi. A note on the bounded fragmentation property and its applications in network reliability. *European Journal of Combinatorics*, 24(7):891–896, 2003.

[42] MohammadTaghi Hajiaghayi and Naomi Nishimura. Subgraph isomorphism, log-bounded fragmentation, and graphs of (locally) bounded treewidth. *Journal of Computer and System Sciences*, 73(5):755–768, 2007.

[43] Frank Harary. *Graph theory*. Addison-Wesley, Reading, MA, 1969.

[44] Søren Højsgaard, David Edwards, and Steffen Lauritzen. *Graphical models with R*. Springer Science & Business Media, 2012.

[45] Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. Big & quic: Sparse inverse covariance estimation for a million variables. In *Advances in neural information processing systems*, pages 3165–3173, 2013.

[46] Shuai Huang, Jing Li, Liang Sun, Jieping Ye, Adam Fleisher, Teresa Wu, Kewei Chen, Eric Reiman, and Alzheimer's Disease NeuroImaging Initiative. Learning brain connectivity of alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 50(3):935–949, 2010.

[47] Byungjin Hwang, Ji Hyun Lee, and Duhee Bang. Single-cell rna sequencing technologies and bioinformatics pipelines. *Experimental & Molecular Medicine*, 50(8):96, 2018.

[48] M. Jagadish and Anindya Sen. Learning a bounded-degree tree using separator queries. In Sanjay Jain, Rémi Munos, Frank Stephan, and Thomas Zeugmann, editors, *Algorithmic Learning Theory*, pages 188–202, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[49] Vishesh Jain, Frederic Koehler, Jingbo Liu, and Elchanan Mossel. Accuracy-memory tradeoffs and phase transitions in belief propagation. In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 1756–1771, Phoenix, USA, 25–28 Jun 2019. PMLR.

[50] Svante Janson. Functional limit theorems for multitype branching processes and generalized pólya urns. *Stochastic Processes and their Applications*, 110(2):177–245, 2004.

[51] Svante Janson. Limit theorems for triangular urn schemes. *Probability Theory and Related Fields*, 134(3):417–452, 2006.

[52] Svante Janson and Elchanan Mossel. Robust reconstruction on trees is determined by the second eigenvalue. *Annals of Probability*, 32(3B):2630–2649, 07 2004.

[53] Varun Jog and Po-Ling Loh. Analysis of centrality in sublinear preferential attachment trees via the crump-mode-jagers branching process. *IEEE Transactions on Network Science and Engineering*, 4(1):1–12, 2016.

[54] David Karger and Nathan Srebro. Learning markov networks: Maximum bounded tree-width graphs. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 392–401. Society for Industrial and Applied Mathematics, 2001.

[55] Harry Kesten and Bernt Stigum. Additional limit theorems for indecomposable multidimensional Galton-Watson processes. *The Annals of Mathematical Statistics*, 37(6):1463–1481, 1966.

[56] Justin Khim and Po-Ling Loh. Confidence sets for the source of a diffusion in regular trees. *IEEE Transactions on Network Science and Engineering*, 4(1):27–40, 2016.

[57] Valerie King, Li Zhang, and Yunhong Zhou. On the complexity of distance-based evolutionary tree reconstruction. In *SODA*, volume 3, pages 444–453, 2003.

[58] Margarete Knape and Ralph Neininger. Pólya urns via the contraction method. *Combinatorics, Probability and Computing*, 23(6):1148–1186, 2014.

[59] Johan Kwisthout, Hans L Bodlaender, and Linda C van der Gaag. The necessity of bounded treewidth for efficient inference in bayesian networks. In *ECAI*, volume 215, pages 237–242, 2010.

[60] Steffen L Lauritzen. *Graphical models*. Clarendon Press, 1996.

[61] Han Liu, Fang Han, Ming Yuan, John Lafferty, and Larry Wasserman. High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

[62] Han Liu, John Lafferty, and Larry Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10(Oct):2295–2328, 2009.

[63] Gábor Lugosi and Shahar Mendelson. Mean estimation and regression under heavy-tailed distributions: A survey. *Foundations of Computational Mathematics*, 19(5):1145–1190, 2019.

[64] Gábor Lugosi and Alan S. Pereira. Finding the seed of uniform attachment trees. *Electronic Journal of Probability*, 24:1–15, 2019.

[65] Hosam Mahmoud. *Pólya Urn Models*. Chapman and Hall/CRC, 2008.

[66] A. Makur, E. Mossel, and Y. Polyanskiy. Broadcasting on random directed acyclic graphs. *IEEE Transactions on Information Theory*, 66(2):780–812, 2020.

[67] Silviu Maniu, Pierre Senellart, and Suraj Jog. An experimental study of the treewidth of real-world graph data (extended version). *arXiv preprint arXiv:1901.06862*, 2019.

[68] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[69] Marc Mézard and Andrea Montanari. Reconstruction on trees and spin glass transition. *Journal of Statistical Physics*, 124(6):1317–1350, Sep 2006.

[70] Stanley Milgram. The small world problem. *Psychology today*, 2(1):60–67, 1967.

[71] Ankur Moitra, Elchanan Mossel, and Colin Sandon. The circuit complexity of inference. *arXiv preprint arXiv:1904.05483*, 2019.

[72] John W Moon. On the centroid of recursive trees. *Australasian Journal of Combinatorics*, 25:211–220, 2002.

[73] Elchanan Mossel. Recursive reconstruction on periodic trees. *Random Structures & Algorithms*, 13(1):81–97, 1998.

[74] Elchanan Mossel. Reconstruction on trees: beating the second eigenvalue. *Annals of Applied Probability*, pages 285–300, 2001.

[75] Elchanan Mossel. Phase transitions in phylogeny. *Transactions of the American Mathematical Society*, 356(6):2379–2404, 2004.

[76] Elchanan Mossel. Survey-information flow on trees. *DIMACS series in discrete mathematics and theoretical computer science*, 63:155–170, 2004.

[77] Elchanan Mossel, Joe Neeman, and Allan Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, editors, *Proceedings of The 27th Conference on Learning Theory*, volume 35 of *Proceedings of Machine Learning Research*, pages 356–370, Barcelona, Spain, 13–15 Jun 2014. PMLR.

[78] Elchanan Mossel and Yuval Peres. Information flow on trees. *The Annals of Applied Probability*, 13(3):817–844, 2003.

[79] Morgan N Price, Paramvir S Dehal, and Adam P Arkin. Fasttree 2–approximately maximum-likelihood trees for large alignments. *PloS one*, 5(3):e9490, 2010.

[80] Tommy Reddad and Luc Devroye. On the discovery of the seed in uniform attachment trees. *Internet Mathematics*, page 7593, 2019.

[81] B.A. Rogozin. An estimate for concentration functions. *Theory of Probability & Its Applications*, 6(1):94–97, 1961.

[82] David Rossell and Piotr Zwiernik. Dependence in elliptical partial correlation graphs. *arXiv preprint arXiv:2004.13779*, 2020.

[83] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Advances in Neural Information Processing Systems*, pages 1657–1665, 2015.

[84] Devavrat Shah and Tauhid R. Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181, 2011.

[85] Allan Sly. Reconstruction for the Potts model. *Annals of Probability*, 39(4):1365–1406, 07 2011.

[86] J.M. Steele. An Efron-Stein inequality for nonsymmetric statistics. *Annals of Statistics*, 14:753–758, 1986.

[87] Seth Sullivant, Kelli Talaska, and Jan Draisma. Trek separation for Gaussian graphical models. *The Annals of Statistics*, pages 1665–1685, 2010.

[88] Antal Van den Bosch, Toine Bogers, and Maurice De Kunder. Estimating search engine index size variability: a 9-year longitudinal study. *Scientometrics*, 107(2):839–856, 2016.

[89] V.N. Vapnik and A.Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[90] Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[91] L.J. Wei. The generalized polya's urn design for sequential medical trials. *The Annals of Statistics*, 7(2):291–296, 1979.

[92] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[93] Fuzhen Zhang, editor. *The Schur complement and its applications*, volume 4 of *Numerical Methods and Algorithms*. Springer-Verlag, New York, 2005.

[94] Richard Y Zhang, Salar Fattahi, and Somayeh Sojoudi. Large-scale sparse inverse covariance estimation via thresholding and max-det matrix completion. *arXiv preprint arXiv:1802.04911*, 2018.

[95] Xiujun Zhang, Xing-Ming Zhao, Kun He, Le Lu, Yongwei Cao, Jingdong Liu, Jin-Kao Hao, Zhi-Ping Liu, and Luonan Chen. Inferring gene regulatory networks from gene expression data by path consistency algorithm based on conditional mutual information. *Bioinformatics*, 28(1):98–104, 2011.

[96] Piotr Zwiernik. Latent tree models. In *Handbook of graphical models*, Chapman & Hall/CRC Handb. Mod. Stat. Methods, pages 265–288. CRC Press, Boca Raton, FL, 2019.