Learning latent tree models with small query complexity

LUC DEVROYE[1,a], GÁBOR LUGOSI[2,3,b] and PIOTR ZWIERNIK[2,c]

[1]*School of Computer Science, McGill University, Montreal, Canada*[a]*lucdevroye@gmail.com*
[2]*Department of Economics and Business, Pompeu Fabra University, Barcelona, Spain;*
Barcelona School of Economics[b]*gabor.lugosi@upf.edu,* [c]*piotr.zwiernik@utoronto.ca*
[3]*Department, ICREA, Pg. Lluís Companys 23, 08010 Barcelona, Spain*

We consider the problem of structure recovery in a graphical model of a tree where some variables are latent. Specifically, we focus on the Gaussian case, which can be reformulated as a well-studied problem: recovering a semi-labeled tree from a distance metric. We introduce randomized procedures that achieve query complexity of optimal order. Additionally, we provide statistical analysis for scenarios where the tree distances are noisy. The Gaussian setting can be extended to other situations, including the binary case and non-paranormal distributions.

Keywords: Latent tree models; phylogenetics; query complexity; probabilistic graphical models; structure learning; phylogenetics

1. Introduction

First discussed by Judea Pearl as tree-decomposable distributions to generalize star-decomposable distributions such as the latent class model (Pearl, 1988, Section 8.3), latent tree models are probabilistic graphical models defined on trees, where only a subset of variables is observed. Mourad et al. (2013), Zhang (2004), Zhang and Poon (2017) extended our theoretical understanding of these models. We refer to Zwiernik (2018) for more details and references.

Latent tree models have found wide applications in various fields. They are used in phylogenetic analysis, network tomography, computer vision, causal modelling, and data clustering. They also contain other well-known classes of models like hidden Markov models, the Brownian motion tree model, the Ising model on a tree, and many popular models used in phylogenetics. In generic high-dimensional problems, latent tree models can be useful in various ways. They share many computational advantages of observed tree models but are more expressible. Latent tree models have been used for hierarchical topic detection (Côme et al., 2021) and clustering.

In phylogenetics, latent tree models have been used to reconstruct the tree of life from the genetic material of surviving species. They have also been used in bioinformatics and computer vision. Machine-learning methods for models with latent variables attract substantial attention from the research community. Some other applications include latent tree models and novel algorithms for high-dimensional data (Chen, Chen and Zhang (2019)), and the design of low-rank tensor completion methods (Zhang et al. (2022)).

Structure learning

The problem of structure or parameter learning for latent tree models has been extensively studied. A seminal work in this field is by Choi et al. (2011), where two consistent and computationally efficient algorithms for learning latent trees were proposed. The main idea is to use the link between a broad

class of latent tree models and tree metrics studied extensively in phylogenetics, a link first established by Pearl (1988) for binary and Gaussian distributions. This has been extended to symmetric discrete distributions by Choi et al. (2011). The proof in (Zwiernik, 2018, Section 2.3) makes it clear that the only essential assumption is that the conditional expectation of every node given its neighbour is a linear function of the neighbour.

From the statistical perspective, testing the corresponding algebraic restrictions on the correlation matrix was studied by Shiers et al. (2016). Sturma, Drton and Leung (2022) revisited this problem. For the machine learning perspective, see Aizenbud et al. (2023), Anandkumar et al. (2011), Huang et al. (2020), Jaffe et al. (2021), Kandiros et al. (2023), Zhou, Wang and Guo (2020). Most of these papers build on the idea of local recursive grouping as proposed by Choi et al. (2011). In particular, they all start by computing all distances between all observed nodes in the tree.

Our contributions

Our work is motivated by novel applications where the dimension n is so large that computing all the distances—or all correlations between observed variables—is impossible. We propose a randomized algorithm that queries the distance oracle and show that the expected query time for our algorithm is $O(\Delta n \log_\Delta(n))$, where n is the number of observed variables and Δ is the maximal degree of the underlying tree. A special case of our problem is the problem of phylogenetic tree recovery. In this case, our approach resembles other phylogenetic tree recovery methods that try to minimize query complexity (see Afshar et al. (2020) for references) and seems to be the first such method that is asymptotically optimal; see King, Zhang and Zhout (2003) for the matching lower bound. More importantly, our algorithm can deal with general latent tree models and in this context, it is again the first such algorithm. As we show in the last section, our algorithm can be easily adjusted to the case of noisy oracles, which is relevant in statistical practice.

2. Preliminaries and basic results

2.1. Trees and semi-labeled trees

A *tree* $T = (V, E)$ is a connected undirected graph with no cycles. In particular, for any two $u, v \in V$ there is a unique path between them, which we denote by $\overline{uv}$. A vertex of T with only one neighbour is called a *leaf*. A vertex of T that is not a leaf is called an *inner vertex* or *internal node*. An edge of T is *inner* if both ends are inner vertices; otherwise, it is called *terminal*. A connected subgraph of T is called a *subtree* of T. A rooted tree (T, ρ) is simply a tree $T = (V, E)$ with one distinguished vertex $\rho \in V$.

A tree T is called a semi-labeled tree with labeled nodes $W \subseteq V$ if every vertex of T of degree ≤ 2 lies in W[1]. We say that T is a phylogenetic tree if T has no degree-2 nodes and W is exactly the set of leaves of T. If $v \in W$, then we say that v is *regular* and we depict it by a solid vertex. If $|W| = n$ then we typically label the vertices in W with $[n] = \{1, \dots, n\}$. A vertex that is not labeled is called *latent*. An example semi-labeled tree is shown in Figure 1.

By definition, all latent nodes are internal and have degree ≥ 3. Therefore, if $|W| = n$, then $|T| \leq 2n + 1$.

[1] This differs slightly from the regular definition of semi-labeled trees (or X-trees) in phylogenetics, where regular nodes can get multiple labels; see Semple and Steel (2003).

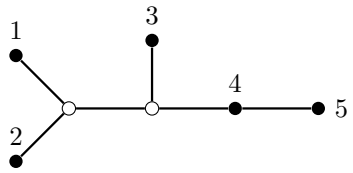


Figure 1. A semi-labeled tree with five regular nodes labeled with $\{1, 2, 3, 4, 5\}$ and two latent nodes.

2.2. Tree metrics

Consider metrics on the discrete set W induced by semi-labeled trees in the following sense. Let $T = (V, E)$ be a tree and suppose that $\ell : E \to \mathbb{R}_+$ is a map that assigns positive lengths to the edges of T. For any pair $u, v \in V$ by $\overline{uv}$ we denote the path in T joining u and v. We now define the map $d_{T,\ell} : V \times V \to \mathbb{R}$ by setting, for all $u, v \in V$,

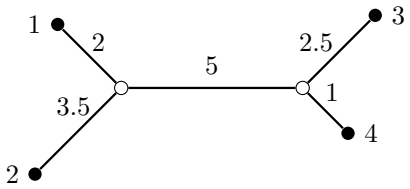
$$d_{T,\ell}(u, v) = \begin{cases} \sum_{e \in \overline{uv}} \ell(e), & \text{if } u \neq v, \\ 0, & \text{otherwise.} \end{cases}$$

Suppose now we are interested only in the distances between the regular vertices.

Definition 2.1. A function $d : [n] \times [n] \to \mathbb{R}$ is called a *tree metric* if there exists a semi-labeled tree $T = (V, E)$ with n regular nodes W and a (strictly) positive length assignment $\ell : E \to \mathbb{R}_+$ such that for all $i, j \in W$

$$d(i, j) \;=\; d_{T,\ell}(i, j).$$

Example 2.2. Consider a quartet tree with edge lengths as indicated on the left in Figure 2. The distance between vertices 1 and 3 is $d(1, 3) = 2 + 5 + 2.5 = 9.5$ and the whole distance matrix is given on the right in Figure 2, where the dots indicate that this matrix is symmetric.



$$\begin{bmatrix} 0 & 5.5 & 9.5 & 8 \\ \cdot & 0 & 11 & 9.5 \\ \cdot & \cdot & 0 & 3.5 \\ \cdot & \cdot & \cdot & 0 \end{bmatrix},$$

Figure 2. A metric on a quartet tree.

It is easy to describe the set of all possible tree metrics.

Definition 2.3. We say that a map $d : [n] \times [n] \to \mathbb{R}$ satisfies the *four-point condition* if for every four (not necessarily distinct) elements $i, j, k, l \in [n]$,

$$d(i, j) + d(k, l) \quad \leq \quad \max \begin{cases} d(i, k) + d(j, l) \\ d(i, l) + d(j, k). \end{cases}$$

Since the elements $i, j, k, l \in [n]$ in Definition 2.3 need not be distinct, every such map is a metric on $[n]$ given that $d(i,i) = 0$ and $d(i,j) = d(j,i)$ for all $i, j \in [n]$. The following fundamental theorem links tree metrics with the four-point condition.

Theorem 2.4 (Tree-metric theorem, Buneman (1974)). *Suppose that $d : [n] \times [n] \to \mathbb{R}$ is such that $d(i,i) = 0$ and $d(i,j) = d(j,i)$ for all $i, j \in [n]$. Then, d is a tree metric on $[n]$ if and only if it satisfies the four-point condition. Moreover, a tree metric uniquely determines the defining semi-labeled tree and edge lengths.*

Note that the assumption about strictly positive lengths of each edge in Definition 2.1 is crucial for uniqueness in Theorem 2.4.

2.3. Recovering a tree from a tree metric

Our problem is to recover the semi-labeled tree T based on a few queries of the distance matrix $D = [d(u,v)]$, which contains the distances between the regular nodes of T (Abdi (1990), Felsenstein (2004), Warnow (1999), Wu, K.-Chao and Tang (1999)). We consider the *query complexity*, which measures the number of queries of the distance matrix, also called queries of the distance oracle. Trivially, we can reconstruct the tree with query complexity $\binom{n}{2}$ (Batagelj, Pisanski and Simoes-Pereira (1990), Boesch (1968), Culberson and Rudnicki (1989), Gusfield (1997), Hakimi and Yau (1964), Nakhleh et al. (2005), Waterman et al. (1977)).

To provide a more refined analysis of the query complexity, we introduce the maximum degree of the tree:

$$\Delta := \max_{i \in V} \text{degree}(i).$$

When the query complexity is jointly measured in terms of n and Δ, a lower bound for both worst-case and expected query complexity is $\Omega(\Delta n \log_\Delta(n))$ (King, Zhang and Zhout (2003)). For another proof of the lower bound for the expected complexity, see Bastide and Groenland (2024).

When T is a phylogenetic tree (W is the set of leaves of T), the distance queries are sometimes referred to as "additive queries" (Waterman et al. (1977)). This case has been extensively studied in the literature; see Jansson (2016) for a recent overview. When the maximum degree Δ is bounded, Hein (1989) demonstrated that the problem can be solved using $O(n \log(n))$ distance queries. When Δ is unbounded, Culberson and Rudnicki (1989) proposed an algorithm that was claimed to achieve a query complexity matching the lower bound for trees where all edge weights are equal to 1. However, as noted by Reyzin and Srivastava (2007), the algorithm's actual runtime is $O(n^{3/2}\sqrt{\Delta})$.

Complementing this work on phylogenetic recovery, Kannan, Lawler and Warnow (1996) proposed an algorithm with $O(\Delta n \log(n))$ query complexity for the noisy-ultrametric model, a special computational framework that is not directly related to "additive queries". In the same computational model, Brodal et al. (2001) presented an algorithm matching the lower bound of $O(\Delta n \log_\Delta(n))$; see also Kao, Lingas and Östlin (1999) for related results. For additional references on this and other specialized computational models, we refer to Afshar et al. (2020). Further discussions and generalizations of phylogenetic tree reconstruction can be found in Buneman (1971), Csürös (2002), Daskalakis, Mossel and Roch (2006, 2009), Gąsieniec et al. (1999), Gronau, Moran and Snir (2012).

Our objective is to develop a randomized algorithm with expected query complexity $O(\Delta n \log_\Delta(n))$ for the general semi-labeled tree reconstruction problem, regardless of how Δ varies with n. Similar to the seminal work of Hein (1989), we focus on pairwise distance queries. This problem generalizes the phylogenetic tree reconstruction problem, as it allows internal non-latent nodes to have degree two, and

regular nodes are not necessarily leaves. Consequently, methods designed specifically for phylogenetic tree reconstruction are no longer directly applicable.

3. The new algorithm

The proposed algorithm uses a randomized version of divide-and-conquer. We will use the notion of a *bag* $B \subseteq V$. The algorithm maintains a queue, consisting of sets of bags. Initially, there is only one bag, containing all regular nodes, that is, $B = W$. The procedure takes a bag B. One node in this set, denoted by $\rho(B)$, is marked as a representative, which can be thought of as a root. A set of edges that jointly form a tree is called a *skeleton* and is typically denoted by the mnemonic S. Our algorithm starts with an empty skeleton and incrementally constructs the skeleton of the sought tree, which we call the tree *induced* by B.

Algorithm 1: Outline of our algorithm

```
Let κ = Δ ;
Pick a node u ∈ W, and set ρ(W) ← u (note: W is now a bag);
Make an empty queue Q ;
Add W to Q ;
Set S ← ∅ ;
while |Q| > 0 do
    Remove bag B from the front of Q ;
    if |B| ≤ κ then
        Query all (|B| choose 2) distances between nodes in B;
        Find S*, the full skeleton for the tree induced by B ;
        S ← S ∪ S*;
    else
        Apply procedure BIGSPLIT (B). This procedure outputs a skeleton S* (connecting nodes
          from B and possibly latent nodes) and bags B_1, ..., B_k where B_i overlaps with the
          nodes of the skeleton in ρ(B_i) only, and are non-overlapping otherwise (i.e.,
          (B_i \ ρ(B_i)) ∩ (B_j \ ρ(B_j)) = ∅) ;
        S ← S ∪ S* ;
        Add B_1, ..., B_k to the rear of Q ;
Return the skeleton S ;
```

The procedure BIGSPLIT takes a bag B and a random set of nodes in it, $u_1, \ldots, u_\kappa$, and forms the subtree that connects $u_1, \ldots, u_\kappa$ and $\rho(B)$. The edges of this subtree give the skeleton that is the output. The remaining nodes of B are collected in bags that "hang" from the skeleton. The representatives of these bags are precisely those nodes where the bags overlap the skeleton. Note that the skeleton may contain latent nodes not originally in B. Within BIGSPLIT, all representatives of the hanging bags have their distances to all nodes in their bags queried, so for all practical purposes, the newly discovered latent nodes act as regular nodes. The bags become smaller as the algorithm proceeds, which leads to a logarithmic number of rounds. The main result of the paper is the following theorem, whose proof is given in Section 4 below.

Theorem 3.1. *Given a distance oracle D between the regular vertices of a semi-labeled tree T, Algorithm 1 with parameter $\kappa = \Delta$ correctly recovers the induced tree with expected query complexity $O(\Delta n \log_\Delta(n))$.*

The procedure BIGSPLIT uses two sub-operations, called BASIC and EXPLODE that we describe next.

3.1. The "basic" operation

Let B be a bag with representative $\rho = \rho(B)$, and let $\alpha \in B$ be a distinct regular vertex. In our basic step, we query $d(v, \alpha)$ for all $v \in B$, and set

$$D(v) = d(v, \alpha) - d(v, \rho). \tag{1}$$

We group all nodes v according to the different values $D(v)$ that are observed. Ordering the sets in this partition of B from small to large value of $D(\cdot)$, we obtain bags $B_1, \ldots, B_k$. It is clear that $\rho \in B_k$ and $\alpha \in B_1$. Within each B_i, we let u_i be the node of B_i closest to α. If

$$d(u_i, \alpha) + d(u_i, \rho) = d(\alpha, \rho),$$

then u_i (a regular node) is on the path from α to ρ in the induced tree. We set $\rho(B_i) = u_i$. If, however,

$$d(u_i, \alpha) + d(u_i, \rho) > d(\alpha, \rho),$$

then we know that there must be a latent node w_i that connects the (α, ρ) path to the nodes in B_i. In fact, for all $v \in B_i$, we have

$$d(v, w_i) = \frac{1}{2}\left(d(v, \alpha) + d(v, \rho) - d(\alpha, \rho)\right). \tag{2}$$

These values can be stored for further use. So, we add w_i to B_i and define $\rho(B_i) = w_i$. In this manner, we have identified S^*, the part of the final skeleton that connects α with ρ:

$$(\rho(B_1), \rho(B_2)), \ldots, (\rho(B_{k-1}), \rho(B_k)).$$

Algorithm 2: BASIC(B, α)

```
Input: a bag B, and α ∈ B;
Set ρ = ρ(B);
for v ∈ B do
    Compute D(v) in (1);
Assign all v ∈ B to bags B_1, ..., B_k according to decreasing values of D(v);
for i = 1, ..., k do
    u_i = arg min_{v ∈ B_i} d(v, α);
    if d(ρ, u_i) + d(u_i, α) = d(ρ, α) then
        ρ(B_i) = u_i
    else
        Identify latent node w_i;
        Add w_i to B_i;
        Set ρ(B_i) = w_i;
        Update the distance oracle by calculating d(w_i, v) for all v ∈ B_i using (2);
Return B_1, ..., B_k and the skeleton (ρ(B_1), ρ(B_2)), ..., (ρ(B_{k-1}), ρ(B_k));
```

The query complexity of Algorithm 2 is bounded by $2|B| - 3$.

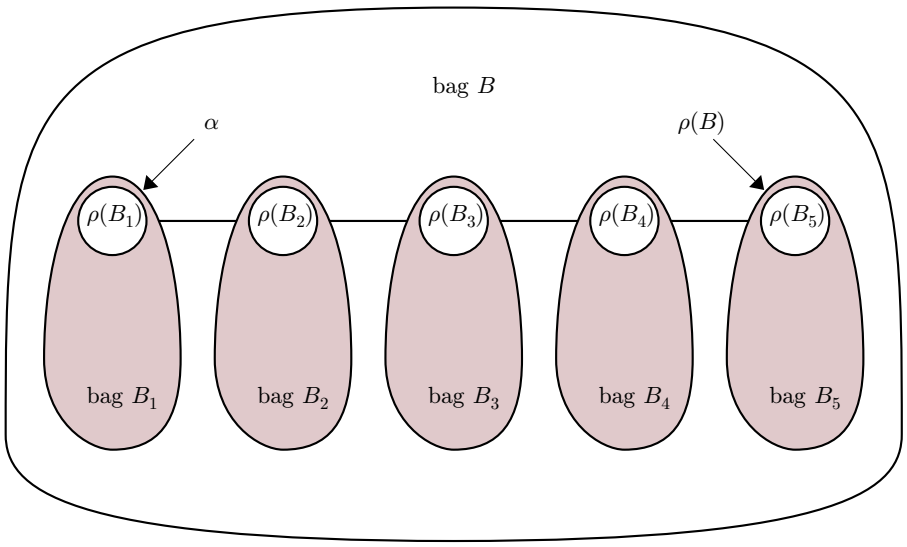


Figure 3. In a basic operation, a bag B with root $\rho(B)$ and query node α is decomposed into several smaller bags, with root nodes on the skeleton.

3.2. The operation "explode"

Another fundamental operation, called *explode*, decomposes a bag B into smaller bags $B_1, \ldots, B_k$—all having the same representative $\rho(B)$—according to the different subtrees of $\rho(B)$ that are part of the tree induced by B. For arbitrary nodes $u \neq v \in B$, $u, v \neq \rho(B)$, we note that u and v are in the same subtree if and only if

$$d(u, v) < d(u, \rho(B)) + d(\rho(B), v).$$

Thus, in query time $O(|B|)$, we can determine the nodes that are in the same subtree as u. Therefore, we can partition all nodes of $B \setminus \{\rho(B)\}$ into disjoint subtrees of $\rho(B)$ (without constructing these trees yet) in query time at most $|B|\Delta$ by peeling off each set in the partition in turn. These sets are output as bags denoted by $B_1, \ldots, B_k$.

Algorithm 3: EXPLODE(B)

Input: a bag B ;
while $|B| > 1$ **do**
  Take $u \in B$, $u \neq \rho(B)$;
  Set $B^* = \{v \in B, v \neq \rho(B) : d(u, v) < d(u, \rho(B)) + d(v, \rho(B))\} \cup \{\rho(B)\}$;
  Set $\rho(B^*) = \rho(B)$;
  Output B^* ;
  Set $B \leftarrow \{\rho(B)\} \cup (B \setminus B^*)$;

The query complexity of Algorithm 3 is bounded by $(|B| - 1)\Delta$.

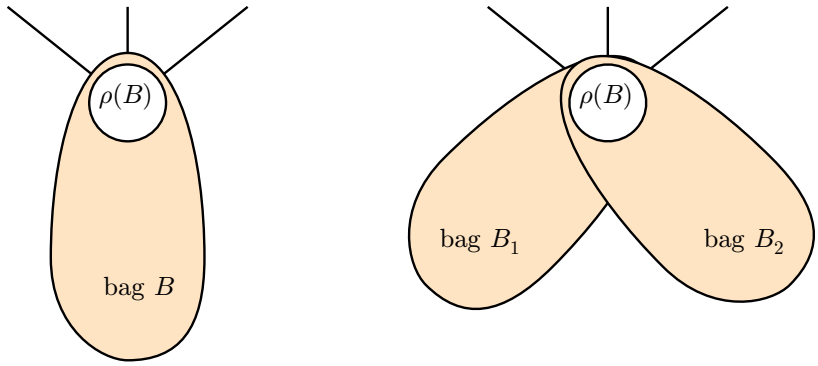


Figure 4. In the explode operation, a bag B with root $\rho(B)$ is decomposed into several smaller bags, each with root $\rho(B)$. All other nodes of B end up in only one of the smaller bags.

3.3. The procedure bigsplit

We are finally ready to provide the details of the procedure BIGSPLIT, which takes as input a bag B and distinct nodes $u_1, \ldots, u_k$ in B not equal to $\rho(B)$.

Algorithm 4: BIGSPLIT(B)

Let $C = \{B\}$;
Let S be a skeleton. Initially, $S = \emptyset$;
Let $M = \max_{B' \in C} |B'|$;
while $M > |B|/\sqrt{\Delta}$ **do**
  Sample uniformly at random and without replacement nodes $u_1, \ldots, u_\kappa$ from $B \setminus \{\rho(B)\}$;
  Let $\mathcal{B}$ be a collection of bags. Initially, $\mathcal{B} = \{B\}$;
  Let S be a skeleton. Initially, $S = \emptyset$;
  for $i = 1$ *to* κ **do**
    Find the bag B^* in $\mathcal{B}$ to which u_i belongs ;
    if $u_i \neq \rho(B^*)$ **then**
      Apply BASIC(B^*, u_i), which outputs a skeleton S^* and bags $B_1, \ldots, B_k$;
      $\mathcal{B} \leftarrow \mathcal{B} \setminus \{B^*\}$;
      $\mathcal{B} \leftarrow \mathcal{B} \cup \cup_{j=1}^{k} \{B_j\}$;
      $S \leftarrow S \cup S^*$;
  Let C be a collection of bags. Initially, C is empty;
  for *all* $B \in \mathcal{B}$ **do**
    EXPLODE(B), which leaves output $B_1, \ldots, B_\ell$;
    Add $B_1, \ldots, B_\ell$ to C;
  Let $M = \max_{B' \in C} |B'|$;
Output S ;
Output all bags in C ;

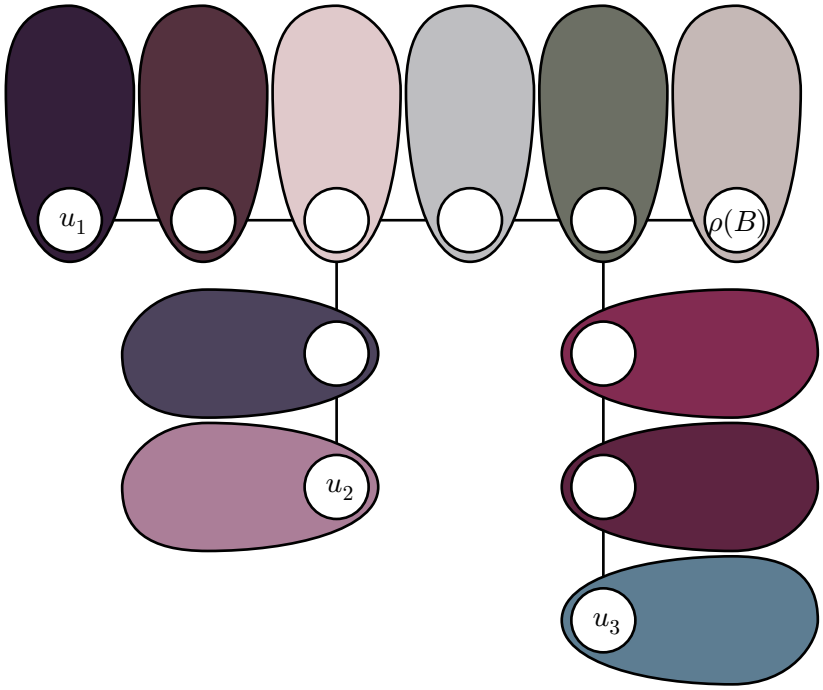


Figure 5. Illustration of BIGSPLIT(B). Each bag anchored on the skeleton is further exploded into smaller bags (not shown).

4. The query complexity

4.1. Proof of Theorem 3.1

For a bag B, and nodes $u_1, \ldots, u_\kappa$ taken uniformly at random without replacement from $B \setminus \{\rho(B)\}$, let M be the size of the largest bag output by BIGSPLIT (B), where $|B| \geq \kappa + 1$, and $\kappa = \Delta$. Lemma 4.1 below shows that

$$\mathbb{P}\left\{M > \frac{|B|}{\sqrt{\Delta}}\right\} \leq \frac{1}{2}.$$

The while loop in BIGSPLIT is repeated until $M \leq |B|/\sqrt{\Delta}$. Thus, we can visualize the algorithm in rounds, starting with W. In other words, in the r-th round, we apply BIGSPLIT to all bags that have been through a BIGSPLIT $r-1$ times. In every round, all bags of the previous round are reduced in size by a factor of $1/\sqrt{\Delta}$. Therefore, there are $\leq \log_{\sqrt{\Delta}}(n) = 2\log_\Delta(n)$ rounds. The query complexity of one BIGSPLIT without the EXPLODE operations is at most $(\kappa+1)(2|B|-1)$. The total query complexity due to all EXPLODE operations is at most $\Delta(|B|-1)$, for a grand total bounded by

$$\kappa + 1 + (|B| - 1) \times (2 + 2\kappa + \Delta).$$

Then the (random) query complexity for splitting bag B is bounded by

$$X(\kappa+1)+X(|B|-1)\times(2+2\kappa+\Delta)=X(\Delta+1)+X(|B|-1)\times(2+3\Delta)\,, \tag{3}$$

where X is geometric $(1/2)$. The expected value of this is $2(\Delta+1)+2(|B|-1)(2+3\Delta)$. Summing over all bags B that participate in one round yields an expected value bound of $O(\Delta)$ times the sum of $|B|-1$ over all participating B. But the bags do not overlap, except possibly for their representatives. Hence the sum of all values $(|B|-1)$ is at most n, as each proper item in a bag is a regular node. As $\kappa=\Delta$, the expected cost of one round of splitting is at most $2(\Delta+1)+n(4+6\Delta)$.

There is another component of the query complexity due to the part in which we construct the induced tree for a bag B when $|B|\le\kappa$. A bag B dealt with in this manner is called final. So the total query complexity becomes the sum of $\binom{|B|}{2}$ computed over all final bags of size at least two. Let the sizes of the final bags be denoted by n_i. Noting that $\sum_i(n_i-1)\le n$, we see that the query complexity due to the final bags is at most

$$\sum_i\frac{n_i(n_i-1)}{2}\le\max_i n_i\sum_i\frac{n_i-1}{2}\le\frac{\kappa n}{2}<\Delta n. \tag{4}$$

The overall expected query complexity does not exceed

$$n\Delta+(\Delta+n(4+6\Delta))\times 2\log_\Delta(n). \tag{5}$$

This finishes the proof of Theorem 3.1.

□

4.2. The main technical lemma

Lemma 4.1. *For a bag B, and random nodes $u_1,\ldots,u_\kappa$ taken uniformly at random without replacement from $B\setminus\{\rho(B)\}$, let M be the size of the largest bag output by* BIGSPLIT *(B), where $|B|\ge\kappa+1$, and $\kappa=\Delta$. Then*

$$\mathbb{P}\left\{M\ge 1+\frac{|B|}{\sqrt{\Delta}}\right\}\le\frac{1}{2}.$$

Proof. Consider $\rho(B)$ as the root of the tree induced by B, on which we perform DFS (depth-first-search). Note that each of the bags left after BIGSPLIT $(B,u_1,\ldots,u_\kappa)$ corresponds to a subtree with root on the skeleton output by BIGSPLIT and having root degree one. We list the regular nodes in DFS order and separate this list into $\kappa+1$ sublists, all separated by $\rho(B),u_1,\ldots,u_\kappa$. Call the sizes of the sublists $N_0,N_1,\ldots,N_\kappa$. Note that the bags consist of regular nodes, except possibly their representatives on the skeleton. Each bag is either contained in a sublist or a sublist plus one of the nodes $\rho(B),u_1,\ldots,u_\kappa$. Thus,

$$M\le\max_i N_i+1.$$

If we number the nodes by DFS order, starting at $\rho(B)$, then picking k nodes uniformly at random without replacement from all nodes, $\rho(B)$ excepted, shows that the N_i's correspond to the cardinalities of the intervals defined by the selected nodes. Therefore, $N_0,\ldots,N_\kappa$ are identically distributed. In addition, for an arbitrary integer k,

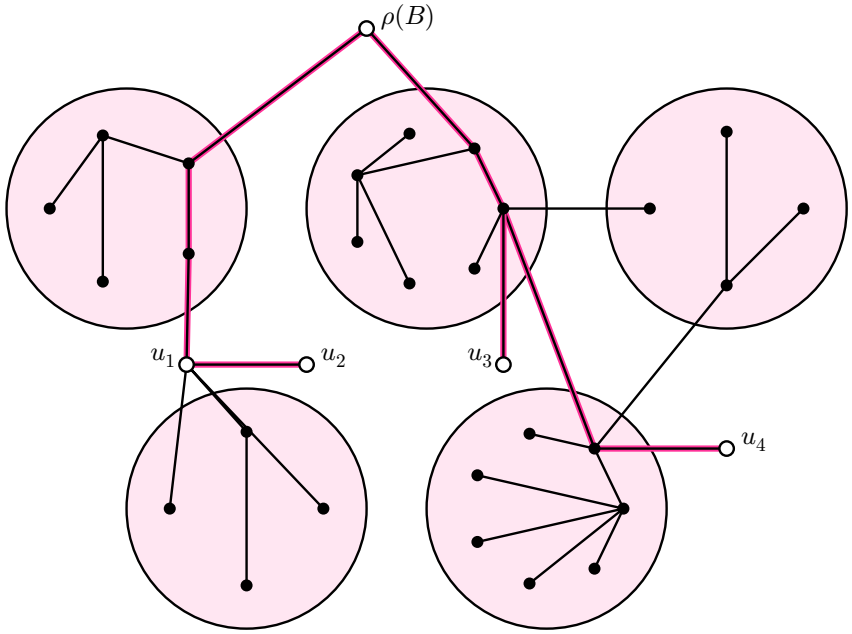


Figure 6. Illustration of BIGSPLIT(B). The skeleton induced by $\{\rho(B), u_1, u_2, u_3, u_4\}$ is shown in thick lines. The remainder of the tree induced by B is shown in thin lines. The nodes traversed by DFS between visits of $\{\rho(B), u_1, u_2, u_3, u_4\}$ are grouped.

$$\begin{aligned}
\mathbb{P}\{N_0 \geq k\} &= \frac{|B|-1-k}{|B|-1} \times \frac{|B|-2-k}{|B|-2} \times \cdots \times \frac{|B|-\kappa-k}{|B|-\kappa} \\
&\leq \left(1 - \frac{k}{|B|-1}\right)^{\kappa} \\
&\leq \exp\left(-\frac{k\kappa}{|B|-1}\right).
\end{aligned}$$

Thus, by the union bound,

$$\begin{aligned}
\mathbb{P}\left\{M \geq 1 + \frac{|B|}{\sqrt{\Delta}}\right\} &\leq (\kappa+1)\exp\left(-\frac{|B|\kappa}{(|B|-1)\sqrt{\Delta}}\right) \\
&\leq (\kappa+1)\exp\left(-\frac{\kappa+1}{\sqrt{\Delta}}\right) \\
&\quad (\text{as } |B| \geq \kappa+1) \\
&\leq (\Delta+1)\exp\left(-\frac{\Delta+1}{\sqrt{\Delta}}\right)
\end{aligned}$$

$$\leq 4\exp\left(-\frac{4}{\sqrt{3}}\right)$$

(as the expression decreases for $\Delta \geq 3$)

$$= 0.39728\ldots$$

$$< \frac{1}{2}.$$

□

4.3. More refined probabilistic bounds

In this section we refine Theorem 3.1 by offering more detailed distributional estimates for the query complexity of Algorithm 1.

Theorem 4.2. *The query complexity Z of Algorithm 1 with parameter $\kappa = \Delta$ has*

$$\mathbb{E}\{Z\} \;\leq\; 19\Delta n \log_\Delta(n).$$

Furthermore, if $\log_\Delta(n) \to \infty$ as $n \to \infty$, then $\mathbb{P}\{Z \geq 2\mathbb{E}\{Z\}\} = o(1)$. Finally, if $\log_\Delta(n) = O(1)$, then $Z/(\Delta n) \overset{st.}{\leq} 6X + o_p(1)$, where $\overset{st.}{\leq}$ denotes stochastic domination, X is a geometric $(1/2)$ random variable, and $o_p(1)$ is a random variable tending to 0 in probability.

Proof. With the choice $\kappa = \Delta$, using (3) for the complexity due to bag splitting and (4) for the complexity due to the final treatment of bags, the algorithm's query complexity can be bounded by the random variable

$$Z = \Delta n + \sum_k \sum_{\text{bags } B \text{ split in round } k} (\Delta + X_B(2+3\Delta)(|B|-1)),$$

where all X_B are i.i.d. geometric $(1/2)$ random variables. Using (5), we have

$$\mathbb{E}\{Z\} \leq \Delta n + (\Delta + (4+6\Delta)n) \times 2\log_\Delta(n) \leq 19\Delta n \log_\Delta(n).$$

As argued in the proof of Theorem 3.1 in each round the bags get reduced by a factor of $1/\sqrt{\Delta}$. Thus, given all bags B in all levels, we see that the variance $\mathbb{V}\{Z\}$ is not more than

$$\sum_k \sum_{\text{bags } B \text{ split in round } k} \mathbb{V}\{X_B\}(2+3\Delta)^2(|B|-1)^2$$

$$\leq (2+3\Delta)^2 \sum_k \frac{n}{\Delta^{k/2}} \sum_{\text{bags } B \text{ split in round } k} 2(|B|-1)$$

$$\leq 2n(2+3\Delta)^2 \sum_k \frac{n}{\Delta^{k/2}}$$

$$\leq \frac{32(n\Delta)^2}{1-1/\sqrt{\Delta}}$$

$$\leq 90(n\Delta)^2.$$

There are two cases: if $\log_\Delta(n) \to \infty$, then by Chebyshev's inequality,

$$\mathbb{P}\{Z > 2\mathbb{E}\{Z\}\} \to 0\,,$$

where we used the fact that $\mathbb{E}(Z) = \Omega(\Delta n \log_\Delta(n))$ (King, Zhang and Zhout (2003)). If on the other hand, $\log_\Delta(n) \le K$ for a large constant K, i.e., $\Delta \ge n^{1/K}$, then

$$Z \;\le\; \Delta n + \Delta + X(2+3\Delta)n + Z' \;\le\; 6n\Delta X + Z',$$

where Z' is defined as Z with level $k = 0$ excluded. Arguing as above, we have

$$\mathbb{V}\{Z'\} \le \frac{32(n\Delta)^2}{\sqrt{\Delta}-1} = o\left((\mathbb{E}\{Z\})^2\right),$$

so that by Chebyshev's inequality,

$$\frac{Z}{n\Delta} \le 6X + o_p(1),$$

where $o_p(1)$ denotes a quantity that tends to zero in probability as $n \to \infty$. In other words, the sequence of random variables $Z/(n\Delta)$ is tight. □

Remark 4.3. For fixed $\Delta \ge 3$, the complexity grows as $n \log n$. In many cases, this is exponential in the diameter of the tree. For example, in a complete Δ-ary tree, the number of leaves is at least $n/2$, while the diameter is about $2\log_\Delta n$. Reconstruction of such trees is impossible without performing at least $n/4$ distance computations.

5. Graphical models on semi-labeled trees

We present a family of probabilistic models over partially observed trees for which the distance-based Algorithm 1 recovers the underlying tree. Although the Gaussian and the binary tree models (also known as the Ising tree model) discussed in Section 5.1 are probably the most interesting, with little effort we can generalize this, which we do in Section 5.2 and Section 5.3.

Given a tree $T = (V, E)$ and a random vector $Y = (Y_v)_{v\in V}$ with values in the product space $\mathcal{Y} = \prod_{v\in V} \mathcal{Y}_v$, consider the underlying graphical model over T, that is, the family of density functions that factorize according to the tree

$$f_Y(y) \;=\; \prod_{uv\in E} \psi_{uv}(y_u, y_v) \qquad \text{for } y \in \mathcal{Y}, \tag{6}$$

where ψ_{uv} are non-negative potential functions (Lauritzen (1996)). The underlying *latent tree model* over the semi-labeled tree T with the labelling set $W = [n]$ is a model for $X = (X_1, \ldots, X_n)$, which is the sub-vector of $Y = (X, H)$ associated to the regular vertices. The density of X is obtained from the joint density of Y by marginalizing out the latent variables H

$$f(x) \;=\; \int_{\mathcal{H}} f_Y(x, h)\mathrm{d}h.$$

Note that in the definition of a semi-labeled tree, we required that all nodes of T of degree ≤ 2 are regular. The restriction complies with this definition; c.f. Section 11.1 in Zwiernik (2018).

5.1. The binary/Gaussian case

We single out two simple examples where Y is jointly Gaussian or when Y is binary. In the second case, the model on a tree coincides with the binary Ising model on T. In both cases, we get a useful path product formula for the correlations, which states that the correlation between any two regular nodes can be written as the product of edge correlations for edges on the path joining these nodes

$$\rho_{ij} = \prod_{uv \in \overline{ij}} \rho_{uv} \qquad \text{for all } i, j \in [n]. \tag{7}$$

The advantage of this representation is that (7) gives a direct translation of correlation structures in these models to tree metrics via $d(i, j) := -\log|\rho_{ij}|$ for all $i \neq j$; see also (Pearl, 1988, Section 8.3.3). To make this explicit we formulate the following proposition.

Proposition 5.1. *Consider a latent tree model over a semi-labeled tree T. Whenever the correlations $\Sigma = [\rho_{ij}]$ in the underlying tree model admit the path-product formula (7), Algorithm 1, applied to the distance oracle defined by $d(i, j) = -\log|\rho_{ij}|$ for all $i \neq j$, recovers the underlying tree.*

Note that our algorithm recovers all the edge lengths, so we can also find the absolute values of the correlations between the latent variables. In the Gaussian case, this yields parameter identification.

Remark 5.2. In the Gaussian latent tree model over a semi-labeled tree T, Algorithm 1 recovers the underlying tree and the model parameters up to sign swapping of the latent variables.

It turns out that the basic binary/Gaussian setting can be largely generalized. We discuss three such generalizations:

(1) General Markov models
(2) Linear models
(3) Non-paranormal distributions

We briefly describe these models for completeness. The former two are dealt with in Section 11.2 in Zwiernik (2018).

5.2. General Markov models and linear models

By general Markov model, we mean a generalization of the binary latent tree models, where each $\mathcal{Y}_v = \{0, \dots, d-1\}$ for some finite $r \geq 1$. Denoting by P^{uv} the $d \times d$ matrix representing the joint distribution of (Y_u, Y_v), and by P^{vv} the diagonal $d \times d$ matrix with the marginal distribution of Y_v on the diagonal, we can define for any two nodes u, v

$$\tau_{uv} := \frac{\det(P^{uv})}{\sqrt{\det(P^{uu})\det(P^{vv})}}. \tag{8}$$

It turns out that for these new quantities, an equation of type (7) still holds, namely, $\tau_{ij} = \prod_{uv \in \overline{ij}} \tau_{uv}$, so we again obtain the tree distance $d(i, j) = -\log|\tau_{ij}|$.

Proposition 5.3. *In a general Markov model over a semi-labeled tree T we can recover T using Algorithm 1 from the distances $d(i, j) = -\log|\tau_{ij}|$ with τ_{ij} defined in (8).*

More generally, suppose that Y_v are now potentially vector-valued, all in $\mathbb{R}^k$, and it holds for every edge uv of T that $\mathbb{E}[Y_u|Y_v]$ is an affine function of Y_v. Let

$$\Sigma_{uv} \;=\; \mathrm{cov}(Y_u, Y_v) \;=\; \mathbb{E}Y_u Y_v^\top - \mathbb{E}Y_u \mathbb{E}Y_v^\top.$$

In this case, defining τ_{uv} as

$$\tau_{uv} \;:=\; \det(\Sigma_{uu}^{-1/2}\Sigma_{uv}\Sigma_{vv}^{-1/2}), \tag{9}$$

we reach the same conclusion as for general Markov models; see Section 11.2.3 in Zwiernik (2018) for details.

Proposition 5.4. *In a linear model over a semi-labeled tree T we can recover recover T using Algorithm 1 from the distances $d(i,j) = -\log|\tau_{ij}|$ with τ_{ij} defined in (9).*

5.3. Non-paranormal distributions

Suppose that $Z = (Z_v)$ is a Gaussian vector with underlying latent tree $T = (V, E)$. Suppose that Y is a monotone transformation of Z, so that $Y_v = f_v(Z_v)$ for strictly increasing functions f_v, $v \in V$. The conditional independence structure of Y and Z are the same. The problem is that the correlation structure of Y may be quite complicated and may not satisfy the product path formula in (7). Suppose however that we have access to the Kendall-τ coefficients $K = [\kappa_{ij}]$ for X with

$$\kappa_{ij} \;=\; \mathrm{corr}(\mathrm{sgn}(X_i - X_i'), \mathrm{sgn}(X_j - X_j')), \tag{10}$$

where X' is an independent copy of X. Then we can use the fact that for all $i \neq j$

$$\mathrm{corr}(\mathrm{sgn}(X_i - X_i'), \mathrm{sgn}(X_j - X_j')) \;=\; \mathrm{corr}(\mathrm{sgn}(Z_i - Z_i'), \mathrm{sgn}(Z_j - Z_j')) \;=:\; \kappa_{ij}^Z.$$

As observed by Liu et al. (2012), since Z is Gaussian, we have a simple formula that relates κ_{ij}^Z to the correlation coefficient $\rho_{ij} = \mathrm{corr}(Z_i, Z_j)$, for all $i \neq j$

$$\rho_{ij} \;=\; \sin(\tfrac{\pi}{2}\kappa_{ij}^Z) \;=\; \sin(\tfrac{\pi}{2}\kappa_{ij}).$$

Applying a simple transformation to the oracle K gives us access to the underlying Gaussian correlation pattern, which now can be used as in the Gaussian case.

Proposition 5.5. *In a non-paranormal distribution over a semi-labeled tree T we can recover T using Algorithm 1 from the distances*

$$d(i,j) \;:=\; -\log|\sin(\tfrac{\pi}{2}\kappa_{ij})|,$$

where κ_{ij} is the Kendall-τ coefficient defined in (10).

6. Statistical guarantees

In this section, we illustrate how the results developed in this paper can be applied in a more realistic scenario when the entries of the covariance matrix cannot be measured exactly. In particular, suppose

a random sample of size N is observed from a zero-mean distribution with covariance matrix Σ. Assume that $\Sigma_{ii} = 1$ for all $i = 1, \dots, n$ and the correlations $\Sigma_{ij} = \rho_{ij}$ satisfy parametrization (7) for some semilabeled tree. In this case $d(i, j) = -\log|\rho_{ij}|$ for all $i \neq j$ forms a tree metric.

Denote by $\widehat{\rho}_{ij}$ a suitable estimator of the correlations based on the sample and let $\widehat{d}(i, j) = -\log|\widehat{\rho}_{ij}|$ be the corresponding plug-in estimator of the distances. Since the $\widehat{d}(i, j)$ do not form a tree metric, we cannot apply directly Algorithm 1. The algorithm uses distances at five places:

1. In Algorithm 2, to compute $D(v)$.
2. In Algorithm 2, to decide if $\rho(B_i)$ is latent or not.
3. In the case when $\rho(B_i)$ is latent, Algorithm 2 also uses the distances to calculate $d(\rho(B_i), v)$ for all $v \in B_i$.
4. In Algorithm 3, to group nodes in B according to the connected components obtained by removing $\rho(B)$.
5. When recovering the skeleton of a small subtree in the case when $|B| \leq \kappa$.

In order to adapt the algorithm to the "noisy" distance oracle, we first propose the noisy versions of BASIC and EXPLODE. The procedure BASIC.noisy is outlined in Algorithm 5, and EXPLODE.noisy in Algorithm 6. The performance of the procedure depends on the following quantities

$$\ell := \min_{i \neq j} d(i, j), \qquad u := \max_{i \neq j} d(i, j), \tag{11}$$

where the minimum and maximum are taken over all regular nodes.

The algorithms have an additional input parameter $\epsilon > 0$ that is an upper bound for the noise level. More precisely, the algorithms work correctly whenever $\max_{i \neq j} |\widehat{d}(i, j) - d(i, j)| \leq \epsilon$.

Algorithm 5: BASIC.noisy(B, α, ϵ)

Input: a bag B, $\alpha \in B$, and $\epsilon > 0$;
for $v \in B$ **do**
  Compute $\widehat{D}(v) = \widehat{d}(v, \alpha) - \widehat{d}(v, \rho)$;
Order $v \in B$ according to the decreasing value of $\widehat{D}(v)$;
If $|\widehat{D}(u) - \widehat{D}(v)| \leq 4\epsilon$, assign u, v to the same bag;
Denote the resulting bags by $B_1, \dots, B_k$;
for $i = 1, \dots, k$ **do**
  $u_i = \arg\min_{v \in B_i} \widehat{d}(\alpha, v)$;
  if $|\widehat{d}(\rho, u_i) + \widehat{d}(u_i, \alpha) - \widehat{d}(\rho, \alpha)| \leq 3\epsilon$ **then**
    $\rho(B_i) = u_i$
  else
    Identify latent node w_i;
    Add w_i to B_i;
    Set $\rho(B_i) = w_i$;
    Calculate $\widehat{d}(w_i, v)$ for all $v \in B_i$ using (2);
Return $B_1, \dots, B_k$ and the skeleton $(\rho(B_1), \rho(B_2)), \dots, (\rho(B_{k-1}), \rho(B_k)))$;

The next simple fact is used repeatedly below.

Lemma 6.1. *Suppose* $\max_{i \neq j} |\widehat{d}(i, j) - d(i, j)| \leq \epsilon$ *for all regular* $i \neq j$ *and let* $a \in \mathbb{R}^{\binom{n}{2}}$. *Then* $|a^\top(\widehat{d} - d)| \leq \|a\|_1 \epsilon$. *In particular,*

(i) *if* $a^\top d = 0$ *then* $|a^\top \widehat{d}| \leq \|a\|_1 \epsilon$;

(ii) *if* $a^\top d \geq \eta$ *then* $a^\top \widehat{d} \geq \eta - \|a\|_1 \epsilon$.

Algorithm 6: EXPLODE.noisy(B, ϵ)

Input: a bag B ;
while $|B| > 1$ **do**
  Take $u \in B$, $u \neq \rho(B)$;
  Set $B^* = \{v \in B, v \neq \rho(B) : d(u, \rho(B)) + d(v, \rho(B)) - d(u, v) > 3\epsilon\} \cup \{\rho(B)\}$;
  Set $\rho(B^*) = \rho(B)$;
  Output B^* ;
  Set $B \leftarrow \{\rho(B)\} \cup (B \setminus B^*)$;

In what follows, we condition on the random event

$$\mathcal{E}(\epsilon) = \left\{ \max_{i \neq j} |\widehat{d}(i,j) - d(i,j)| \leq \epsilon \text{ for all regular } i \neq j \right\} \tag{12}$$
$$= \left\{ \max_{i \neq j} \left| \log \left| \frac{\widehat{\rho}_{ij}}{\rho_{ij}} \right| \right| \leq \epsilon \text{ for all regular } i \neq j \right\}.$$

Proposition 6.2. *Suppose that in the current round, the event* $\mathcal{E}(\epsilon)$ *holds with* $\epsilon < \ell/4$. *Then Algorithm 5 and Algorithm 6 applied to the noisy distances gives the same output as Algorithm 2 and Algorithm 3 applied to their noiseless versions.*

Proof. In the first part, Algorithm 5 computes $\widehat{D}(v)$ for all v and it uses this information to produce bags $B_1, \ldots, B_k$. The bags in Algorithm 2 are obtained by grouping nodes based on the increasing values of $D(v)$. By Lemma 6.1(i), if $D(u) = D(v)$ then $|\widehat{D}(u) - \widehat{D}(v)| \leq 4\epsilon$. Moreover, if $D(u) > D(v)$ then it must be that $D(u) - D(v) \geq 2\ell$ and so, by Lemma 6.1(ii), $\widehat{D}(u) - \widehat{D}(v) > 2\ell - 4\epsilon > 4\epsilon$. This shows that this step of Algorithm 5 provides the same bags as Algorithm 2. In the second part of the algorithm, we decide whether or not the corresponding path nodes $\rho(B_i)$ are regular. Let $\hat{u}_i = \arg\min_{v \in B_i} \widehat{d}(\alpha, v)$ and $u_i = \arg\min_{v \in B_i} d(\alpha, v)$. If $\hat{u}_i \neq u_i$ then $d(\rho, \hat{u}_i) - d(\rho, u_i) \geq \ell$. By Lemma 6.1(ii),

$$\widehat{d}(\rho, \hat{u}_i) - \widehat{d}(\rho, u_i) \geq \ell - 2\epsilon \geq 2\epsilon,$$

which contradicts the fact that $\hat{u}_i = \arg\min_{v \in B_i} \widehat{d}(\alpha, v)$. We conclude that $\hat{u}_i = u_i$. Now consider the problem of deciding whether $\rho(B_i) = u_i$. Suppose first that $d(\rho, u_i) + d(\alpha, u_i) - d(\rho, \alpha) = 0$ (i.e., $\rho(B_i) = u_i$). By Lemma 6.1(i),

$$|\widehat{d}(\rho, u_i) + \widehat{d}(\alpha, u_i) - \widehat{d}(\rho, \alpha)| \leq 3\epsilon.$$

If $d(\rho, u_i) + d(\alpha, u_i) - d(\rho, \alpha) > 0$ then, since D is a tree metric, it must be $d(\rho, u_i) + d(\alpha, u_i) - d(\rho, \alpha) \geq 2\ell$. Then, by Lemma 6.1(ii) and by the fact that $\epsilon < \ell/4$

$$\widehat{d}(\rho, u_i) + \widehat{d}(\alpha, u_i) - \widehat{d}(\rho, \alpha) \geq 2\ell - 3\epsilon > 5\epsilon.$$

This shows the correctness of the second part of Algorithm 5.
Since $d(v, \alpha) + d(v, \rho) - d(\alpha) \geq 2\ell$, we get $\widehat{d}(v, \alpha) + \widehat{d}(v, \rho) - \widehat{d}(\alpha) \geq 2\ell - 3\epsilon$
We can similarly show that Algorithm 6 gives the same output as Algorithm 3. □

The problem with applying Proposition 6.2 recursively to each round is that the event $\mathcal{E}(\epsilon)$ only bounds the noise for distances between the n originally regular nodes. As the procedure progresses,

new nodes are made regular; if $\rho(B_i)$ is latent we make it "regular" by updating distances $\widehat{d}(w_i, v)$ for all $v \in B_i$ using (2).

Lemma 6.3. *Suppose that event $\mathcal{E}(\epsilon)$ holds. After R rounds of the algorithm, we have $|\widehat{d}(i,j) - d(i,j)| \le (1 + \frac{R}{2})\epsilon$ for all $i, j \in B$ that are regular in the R-th round.*

Proof. Consider the first run of Algorithm 5. In this case, $\rho(B)$ and α are both regular. If w_i is identified as a latent node, we calculate $\widehat{d}(w_i, v)$ for all $v \in B_i$ using (2). By the triangle inequality, $|\widehat{d}(w_i, v) - d(w_i, v)| \le \frac{3}{2}\epsilon$ and this bound is sharp. This establishes the case $R = 1$. Suppose the lemma bound holds up to the $(R-1)$-st round. In the R-th round, $\rho(B)$ may be a latent node added in the previous call but α is still sampled from the originally regular nodes. If $\rho(B_i)$ is identified as a latent node, we calculate $\widehat{d}(\rho(B_i), v)$ for all $v \in B_i$. Since v is also among the originally regular nodes, we conclude $|\widehat{d}(v, \alpha) - d(v, \alpha)| \le \epsilon$. By induction, $|\widehat{d}(v, \rho(B)) - d(v, \rho(B))| \le (1 + \frac{R-1}{2})\epsilon$ and $|\widehat{d}(\alpha, \rho(B)) - d(\alpha, \rho(B))| \le (1 + \frac{R-1}{2})\epsilon$. By the triangle inequality,

$$|\widehat{d}(\rho(B_i), v) - d(\rho(B_i), v)| \;\le\; \tfrac{1}{2}(\epsilon + (1 + \tfrac{R-1}{2})\epsilon + (1 + \tfrac{R-1}{2})\epsilon) \;=\; (1 + \tfrac{R}{2})\epsilon.$$

The result now follows by induction. □

It is generally easy to show that the event $\mathcal{E}(\epsilon)$ holds with probability at least $1 - \eta$ as long as the sample size N is large enough. Let $\delta = 1 - e^{-\epsilon}$ and suppose that the following event holds

$$\mathcal{E}'(\delta) \;:=\; \left\{ |\widehat{\rho}_{ij} - \rho_{ij}| \le |\rho_{ij}|\delta \quad \text{for all regular } i \ne j \right\}.$$

Since $\delta < 1$, under the event $\mathcal{E}'$ the signs of $\widehat{\rho}_{ij}$ and ρ_{ij} are the same. It is easy to see that $\mathcal{E}' \subseteq \mathcal{E}$. Indeed, under $\mathcal{E}'$, $\frac{\hat{\rho}_{ij}}{\rho_{ij}} > 0$ and for all $1 - \delta \le x \le 1 + \delta$ we have $\log(1-\delta) \le \log(x) \le \log(1+\delta)$. It follows that

$$\left| \log \frac{\hat{\rho}_{ij}}{\rho_{ij}} \right| \;\le\; \max\{\log(1+\delta), |\log(1-\delta)|\} \;=\; -\log(1-\delta) \;=\; \epsilon.$$

It is then enough to bound the probability of the event $\mathcal{E}'$. To illustrate how this can be done without going into unnecessary technicalities, suppose $\max_i \mathbb{E}X_i^4 \le \kappa$ for some $\kappa > 0$. In this case $\mathrm{var}(X_i X_j) \le \kappa$, and therefore one may use the median-of-means estimator (see, e.g., Lugosi and Mendelson (2019)) to estimate $\rho_{ij} = \mathbb{E}[X_i X_j]$. We get the following result.

Theorem 6.4. *Suppose a random sample of size N is generated from a mean zero distribution with covariance matrix Σ satisfying $\Sigma_{ii} = 1$ and suppose $\max_i \mathbb{E}X_i^4 \le \kappa$ for some $\kappa > 0$. Let ℓ, u be as defined in (11). Fix $\eta \in (0,1)$ and suppose*

$$\epsilon \;\le\; \frac{\ell}{4(1 + \log_\Delta(n))} \qquad \text{and} \qquad N \;\ge\; \frac{64\kappa \log(n/\eta)}{(1 - 2e^{-\epsilon})} e^{2u}$$

then the noisy version of the Algorithm 1 correctly recovers the underlying semi-labeled tree with probability $1 - \eta$.

Proof. Suppose the event $\mathcal{E}(\epsilon)$ holds. Like in the proof of Theorem 3.1 we modify the procedure by repeating BIGSPLIT.noisy until the largest bag is bounded in size by $|B|/\sqrt{\Delta}$. With this modification, the algorithm stops after each round. With our choice of ϵ, by Lemma 6.3, we are guaranteed that all computed distances satisfy $|\widehat{d}(u,v) - d(u,v)| \le \epsilon$ in the first $2\log_\Delta(n)$ rounds. By Proposition 6.2,

all these subsequent calls of BIGSPLIT.noisy return the same answer as BIGSPLIT applied to noiseless distances. The proof now follows if we can show that, with probability at least $1-\eta$, event $\mathcal{E}(\epsilon)$ holds. We show that $\mathcal{E}'(\delta)$ with $\delta = 1-e^{-\epsilon}$ holds, which is a stronger condition. Recall that $\mathbb{E}X_i^2 = 1$, $\mathbb{E}X_i^4 \le \kappa$ and, in consequence, $\mathrm{var}(X_iX_i) \le \kappa$. We want to estimate $\mathbb{E}(X_iX_j) = \rho_{ij}$. By Theorem 2 in Lugosi and Mendelson (2019), the median-of-means estimator $\widehat{\rho}_{ij}$ (with an appropriately chosen number of blocks that depends on η only) satisfies that with probability at least $1-2\eta/\binom{n}{2}$

$$|\widehat{\rho}_{ij} - \rho_{ij}| \le \sqrt{\frac{32\kappa \log(\binom{n}{2}/\eta)}{N}}.$$

Thus, we get that with probability at least $1-\eta$, all $\widehat{\rho}_{ij}$ satisfy simultaneously that $|\widehat{\rho}_{ij} - \rho_{ij}| \le |\rho_{ij}|\delta$ as long as

$$N \ge \frac{64\kappa \log(n/\eta)}{\delta^2 \min_{i\neq j} \rho_{ij}^2} = \frac{64\kappa \log(n/\eta)}{e^{-2u}\delta^2}.$$

The inequality $\epsilon < \frac{\ell}{4(1+\log_\Delta(n))}$ is equivalent to $\delta < 1 - e^{-\frac{\ell}{4(1+\log_\Delta(n))}}$. Thus, we require

$$N \ge \frac{64\kappa \log(n/\eta)}{e^{-2u}(1-2e^{-\frac{\ell}{4(1+\log_\Delta(n))}})}.$$

□

Note that the required sample size N grows exponentially with the diameter u of the tree. While this seems undesirable, in the setup of this section, such a dependence is easily seen to be necessary, as the correlations along any path decrease exponentially, and any estimator of the correlations needs to have accuracy at the same scale.

Funding

The authors thank the referees for their insightful comments. Luc Devroye was supported by NSERC grant A3456. Piotr Zwiernik and Gábor Lugosi acknowledge the support of Ayudas Fundación BBVA a Proyectos de Investigación Científica 2021 and the Spanish Ministry of Economy and Competitiveness grant PID2022-138268NB-I00, financed by MCIN/AEI/10.13039/501100011033, FSE+MTM2015-67304-P, and FEDER, EU. Piotr Zwiernik was also supported by NSERC grant RGPIN-2023-03481.

References

ABDI, H. (1990). Additive-tree representations. In *Trees and Hierarchical Structures: Proceedings of a Conference held at Bielefeld, FRG, Oct. 5-9th, 1987. Lecture Notes in Biomathematics, vol. 84* 43–59. Springer.

AFSHAR, R., GOODRICH, M. T., MATIAS, P. and OSEGUEDA, M. C. (2020). Reconstructing biological and digital phylogenetic trees in parallel. In *28th Annual European Symposium on Algorithms (ESA 2020)* 3:1–3:24. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

AIZENBUD, Y., JAFFE, A., WANG, M., HU, A., AMSEL, N., NADLER, B., CHANG, J. T. and KLUGER, Y. (2023). Spectral top-down recovery of latent tree models. *Inf. Inference* **12** iaad032. https://doi.org/10.1093/imaiai/iaad032

ANANDKUMAR, A., CHAUDHURI, K., HSU, D. J., KAKADE, S. M., SONG, L. and ZHANG, T. (2011). Spectral methods for learning multivariate latent tree structure. In *NIPS'11: Proceedings of the 24th International Conference on Neural Information Processing Systems* **24** 2025–2033.

BASTIDE, P. and GROENLAND, C. (2024). Optimal distance query reconstruction for graphs without long induced cycles. *arXiv* 2306.05979.

BATAGELJ, V., PISANSKI, T. and SIMOES-PEREIRA, J. M. S. (1990). An algorithm for tree-realizability of distance matrices. *Int. J. Comput. Math.* **34** 171–176.

BOESCH, F. T. (1968). Properties of the distance matrix of a tree. *Quarterly Appl. Math.* **26** 607–609.

BRODAL, G. S., FAGERBERG, R., PEDERSEN, C. N. and ÖSTLIN, A. (2001). The complexity of constructing evolutionary trees using experiments. In *Automata, Languages and Programming: 28th International Colloquium, ICALP 2001 Crete, Greece, July 8–12, 2001 Proceedings 28* 140–151. Springer.

BUNEMAN, P. (1971). The Recovery of Trees from Measures of Dissimilarity. In *Mathematics in the Archaeological and Historical Sciences* (F. HODSON, D. KENDALL and P. TAUTU, eds.) 387–395. Edinburgh University Press.

BUNEMAN, P. (1974). A note on the metric properties of trees. *J. Comb. Th. Ser. B* **17** 48–50.

CHEN, P., CHEN, Z. and ZHANG, N. L. (2019). A novel document generation process for topic detection based on hierarchical latent tree models. In *Symbolic and Quantitative Approaches to Reasoning with Uncertainty: 15th European Conference, ECSQARU 2019, Belgrade, Serbia, September 18-20, 2019, Proceedings 15* 265–276. Springer.

CHOI, M. J., TAN, V. Y. F., ANANDKUMAR, A. and WILLSKY, A. S. (2011). Learning latent tree graphical models. *J. Mach. Learn. Res.* **12** 1771–1812.

CÔME, E., JOUVIN, N., LATOUCHE, P. and BOUVEYRON, C. (2021). Hierarchical clustering with discrete latent variable models and the integrated classification likelihood. *Adv. Data An. Class.* **15** 957–986.

CSÜRÖS, M. (2002). Fast Recovery of Evolutionary Trees with Thousands of Nodes. *J. Comput. Biol.* **9** 277–297.

CULBERSON, J. C. and RUDNICKI, P. (1989). A fast algorithm for constructing trees from distance matrices. *Inf. Process. Lett.* **30** 215–220.

DASKALAKIS, C., MOSSEL, E. and ROCH, S. (2006). Optimal Phylogenetic Reconstruction. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing (STOC)* 159–168.

DASKALAKIS, C., MOSSEL, E. and ROCH, S. (2009). Phylogenies Without Branch Bounds: Contracting the Short, Pruning the Deep. In *Proceedings of the Thirteenth Annual International Conference on Computational Molecular Biology (RECOMB)* 451–465.

FELSENSTEIN, J. (2004). *Inferring Phylogenies*. Sinauer Associates, Inc.

GĄSIENIEC, L., JANSSON, J., LINGAS, A. and ÖSTLIN, A. (1999). On the Complexity of Constructing Evolutionary Trees. *J. Comb. Opt. (JOCO)* **3** 183–197.

GRONAU, I., MORAN, S. and SNIR, S. (2012). Fast and Reliable Reconstruction of Phylogenetic Trees with Indistinguishable Edges. *Random Struct Algorithms* **40** 350–384. https://doi.org/10.1002/rsa.20372

GUSFIELD, D. M. (1997). *Algorithms on Strings, Trees and Sequences*. Cambridge University Press.

HAKIMI, S. L. and YAU, S. S. (1964). Distance matrix of a graph and its realizability. *Quarterly Appl. Math.* **22** 305–317.

HEIN, J. J. (1989). An optimal algorithm to reconstruct trees from additive distance data. *Bull. Math. Biol.* **51** 597–603.

HUANG, F., NARESH, N. U., PERROS, I., CHEN, R., SUN, J. and ANANDKUMAR, A. (2020). Guaranteed scalable learning of latent tree models. In *Uncertainty in Artificial Intelligence* 883–893. PMLR.

JAFFE, A., AMSEL, N., AIZENBUD, Y., NADLER, B., CHANG, J. T. and KLUGER, Y. (2021). Spectral neighbor joining for reconstruction of latent tree models. *SIAM J. Math. Data Sci.* **3** 113–141.

JANSSON, J. (2016). Phylogenetic tree construction from a distance matrix. In *Encyclopedia of Algorithms, Second Edition* (M.-Y. KAO, ed.) 1564–1567. Springer Science.

KANDIROS, V., DASKALAKIS, C., DAGAN, Y. and CHOO, D. (2023). Learning and testing latent-tree Ising models efficiently. In *The Thirty Sixth Annual Conference on Learning Theory* 1666–1729. PMLR.

KANNAN, S. K., LAWLER, E. L. and WARNOW, T. J. (1996). Determining the evolutionary tree using experiments. *J. Alg.* **21** 26–50.

KAO, M.-Y., LINGAS, A. and ÖSTLIN, A. (1999). Balanced randomized tree splitting with applications to evolutionary tree constructions. In *STACS 99: 16th Annual Symposium on Theoretical Aspects of Computer Science Trier, Germany, March 4–6, 1999 Proceedings 16* 184–196. Springer.

KING, V., ZHANG, L. and ZHOUT, Y. (2003). On the complexity of distance-based evolutionary tree reconstruction. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms* 444–453. SIAM.

LAURITZEN, S. L. (1996). *Graphical Models*. Clarendon Press, Oxford, United Kingdom.

LIU, H., HAN, F., YUAN, M., LAFFERTY, J. and WASSERMAN, L. (2012). High-dimensional semiparametric Gaussian copula graphical models. *Ann. Statist.* **40** 2293–2326.

LUGOSI, G. and MENDELSON, S. (2019). Mean estimation and regression under heavy-tailed distributions: A survey. *Foundations of Computational Mathematics* **19** 1145–1190.

MOURAD, R., SINOQUET, C., ZHANG, N. L., LIU, T. and LERAY, P. (2013). A survey on latent tree models and applications. *J. Artif. Intell. Res.* **47** 157–203.

NAKHLEH, L., WARNOW, T., RINGE, D. and EVANS, S. N. (2005). A Comparison of Phylogenetic Reconstruction Methods on an Indo-European Dataset. *Trans. Philol. Soc.* **103** 171–192. https://doi.org/10.1111/j.1467-968X.2005.00155.x

PEARL, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. *The Morgan Kaufmann Series in Representation and Reasoning*. Morgan Kaufmann, San Mateo, CA.

REYZIN, L. and SRIVASTAVA, N. (2007). On the longest path algorithm for reconstructing trees from distance matrices. *Information Processing Letters* **101** 98–100.

SEMPLE, C. and STEEL, M. A. (2003). *Phylogenetics*. *Oxford Lecture Series in Mathematics and Its Applications* **24**. Oxford University Press.

SHIERS, N., ZWIERNIK, P., ASTON, J. A. D. and SMITH, J. Q. (2016). The correlation space of Gaussian latent tree models and model selection without fitting. *Biom.* **103** 531–545.

STURMA, N., DRTON, M. and LEUNG, D. (2022). Testing many and possibly singular polynomial constraints. *arXiv:2208.11756*.

WARNOW, T. (1999). Some combinatorial optimization problems in phylogenetics. In *Graph Theory and Combinatorial Biology, Bolyai Society Mathematical Studies Volume 7* 363–413. Bolyai János Matematikai Társulat.

WATERMAN, M. S., SMITH, T. F., SINGH, M. and BEYER, W. A. (1977). Additive evolutionary trees. *J. Theo. Biol.* **64** 199–213.

WU, B. Y., K. CHAO, M. and TANG, C. Y. (1999). Approximation and exact algorithms for constructing minimum ultrametric trees from distance matrices. *J. Combin. Optim.* **3** 199–211.

ZHANG, N. L. (2004). Hierarchical latent class models for cluster analysis. *J. Mach. Learn. Res.* **5** 697–723.

ZHANG, N. and POON, L. (2017). Latent tree analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence* **31** 4891-4897.

ZHANG, H., ZHAO, J., WANG, X. and XUAN, Y. (2022). Low-voltage distribution grid topology identification with latent tree model. *IEEE Trans. Smart Grid* **13** 2158–2169.

ZHOU, C., WANG, X. and GUO, J. (2020). Learning mixed latent tree models. *J. Mach. Learn. Res.* **21** 1–35.

ZWIERNIK, P. (2018). Latent Tree Models. In *Handbook of Graphical Models* (M. Drton, M. H. Maathuis, S. L. Lauritzen and M. J. Wainwright, eds.) 283–306. CRC Press.