Recoverability Landscape of Tree Structured Markov Random Fields under Symmetric Noise

Ashish Katiyar
Amazon

Soumya Basu
Google

Vatsal Shah
Amazon

Constantine Caramanis
UT Austin

Abstract

We study the problem of learning tree-structured Markov random fields (MRF) on discrete random variables with common support when the observations are corrupted by a k-ary symmetric noise channel with unknown probability of error. For Ising models (support size $= 2$), past work has shown that graph structure can only be recovered up to the leaf clusters (a leaf node, its parent, and its siblings form a leaf cluster) and exact recovery is impossible. No prior work has addressed the setting of support size of 3 or more, and indeed this setting is far richer. As we show, when the support size is 3 or more, the structure of the leaf clusters may be partially or fully identifiable. We provide a precise characterization of this phenomenon and show that the extent of recoverability is dictated by the joint PMF of the random variables. In particular, we provide necessary and sufficient conditions for exact recoverability. Furthermore, we present a polynomial time, sample efficient algorithm that recovers the exact tree when this is possible, or up to the unidentifiability as promised by our characterization, when full recoverability is impossible. Finally, we demonstrate the efficacy of our algorithm experimentally.

1 INTRODUCTION

Markov Random Fields (MRFs) provide a useful framework for modeling high dimensional probability distributions via an associated dependency graph $\mathbf{G}$, which captures the conditional independence relationships between random variables. Here, the nodes correspond to the random variables; edges represent the conditional independence relationships between these nodes. Any random variable conditioned on the random variables with which it shares an edge is independent of all the remaining random variables.

This 'Markov' property has encouraged the adoption of MRFs in a wide variety of fields such as computer vision, finance, biology, and social networks. Here, MRFs model various inference tasks via popular algorithms such as loopy belief propagation, message passing, etc. For a deeper understanding of these underlying ideas and applications, we refer the reader to Lauritzen [1996], Koller and Friedman [2009], Wainwright and Jordan [2008], Pearl [2014].

A special class of graphical model where the underlying graph is tree-structured is suited for applications where sample efficient learning, and time-efficient inference are required with strong theoretical guarantees. As a result, the problem of learning tree-structured graphical models from data has been well-studied since the 1960s. In the seminal work Chow and Liu [1968], the authors propose the Chow-Liu algorithm, which shows that the maximum weight spanning tree of the empirical mutual information between all the pairs of random variables corresponds to the maximum-likelihood tree estimate. In practice, it is rare to observe the random variables without noise, as sources of noise are ubiquitous, e.g. errors in sensors, incorrect human labeling. In Nikolakakis et al. [2019], the authors present numerous motivating examples from social science, epidemiology, biology, differential privacy, and finance, where noise is present in the observations. Unfortunately, in the face of corruption by unequal noise in the nodes, the Chow-Liu algorithm breaks down. This occurs as the noise in the random variables alters the order of the pairwise mutual information. The noise also destroys the tree structure by adding fictitious edges. Moreover, as noise is unknown, the structure of a noisy graphical model could possibly originate from different tree structures. This brings the recoverability of the original tree structure into question.

In this paper, we focus on learning the underlying

tree-structured graphical model on non-noisy discrete random variables using samples that are corrupted by a k-ary symmetric noise channel (where k is the size of the common support of all the random variables). Our work reveals a rich recoverability landscape for MRFs under symmetric noise. We discover that when $k \geq 3$, for a fixed underlying tree structure, the recoverability is determined by the pairwise PMF of the non-noisy random variables. This is in contrast to the Gaussian graphical model and Ising model results (Katiyar et al. [2019], Katiyar et al. [2020], Tandon et al. [2021]) where, for a fixed tree structure, edges within a *leaf cluster* (a leaf node, its parent, and its siblings) are never recoverable irrespective of the probability distribution of the non-noisy random variables. We completely characterize the recoverability for $k \geq 2$ by providing the necessary and sufficient conditions for the identifiability of the edges within a *leaf cluster*.

Our contributions can be summarized as follows:

1. **Identifiability Characterization**: In *Theorem 1*, we completely characterize the recoverability of tree-structured MRF on support size k when the observations come from unknown k-ary symmetric channel noise where each node has a different error probability. We show the identifiability depends on the PMF of the non-noisy random variables, which is unobserved. This dependence can then be translated to the PMF of the noisy random variables, which is observed, that provides the characterization.
   We show that for the special class of *Symmetric Graphical Models* (as defined in *Section 4.4*), for any k, the nodes within a *leaf cluster* are unidentifiable. On the other direction, we show for the class of Perturbed Symmetric Graphical Models (details in *Section 4.4*) for $k \geq 4$, the exact tree is identifiable.

2. **Algorithm**: We develop an algorithm that recovers the class of candidate trees that can explain the noisy observations. In the identifiable setting, this corresponds to recovering the exact tree. The algorithm is iterative where we recover one edge from the candidate tree per iteration. *(Section 5)*.

3. **Sample Complexity Analysis:** We provide novel sample complexity lower bounds and upper bounds (*Section 6*). Our upper bounds are shown to have orderwise tight dependence on underlying graph parameters, size of the graph, edge parameters (related to underlying conditional PMF), and noise parameters. The lower bound proof relies on a novel construction of a class of graphical models including perturbed symmetric graphical models where part of the *leaf clusters* are identifiable.

4. **Experiments:** We demonstrate the efficacy of our algorithm via extensive numerical experiments for a variety of trees with different structures, edge parameters, corruption, and support sizes.

2 RELATED WORK

We divide the related work into three main categories: **Learning Generic Graphical Models from Non-Noisy Samples:** There exists a rich literature on the problem of learning graphical models on discrete random variables which assume access to non-noisy samples: Bresler et al. [2014b, 2008], Bresler [2015], Bresler et al. [2014a], Lee et al. [2007], Klivans and Meka [2017], Wu et al. [2019], Ravikumar et al. [2010]. However, these models do not provide guarantees in the face of noise in the samples.

Learning Tree-Structured Graphical Models: The special class of tree-structured graphical models has also been extensively studied beginning with the classical Chow-Liu algorithm was proposed in Chow and Liu [1968]. Chow-Liu algorithm's error exponents for Gaussian graphical models and graphical models on discrete random variables were analyzed in Tan et al. [2010] and Tan et al. [2011] respectively. Results in Tan et al. [2011] were further refined in Tandon et al. [2020] under additional assumptions of homogeneity and zero external field in tree-structured Ising models. In Bresler and Karzand [2016] the authors approximate the distribution of generic Ising models using tree-structured Ising models. More recently, in Daskalakis and Pan [2020], the authors provide an algorithm to learn tree-structured Ising models providing total variation distance guarantees. In Bhattacharyya et al. [2020], the authors provide finite sample guarantees for the Chow-Liu algorithm. As these algorithms assume access to non-noisy samples, no performance guarantees can be established when the samples have noise.

Robust Estimation of Graphical Models: Robust estimation of graphical models has been studied in multiple prior works but they are unable to resolve our setting. The algorithms in Goel et al. [2019], Lindgren et al. [2019], Hamilton et al. [2017] learn graphical models on discrete random variables without the tree structure assumption but assume access to error probabilities. This is complementary to our setting as we have the tree structure constraint but do not require the knowledge of the error probabilities.

Recovering Ising models robust to a noise model where a fraction of samples are arbitrarily corrupted has been studied in Prasad et al. [2020] and Diakonikolas et al. [2021]. While we allow all the samples to be noisy, we restrict the noise to a k-ary symmetric noise channel. On the contrary these papers allow the noise to be arbitray, however, they require access to a fraction of clean samples. Thus, the techniques proposed in these papers are inapplicable in our setting.

In Tandon et al. [2020], Nikolakakis et al. [2019, 2020], the authors study the recovery of trees using noisy sam-

ples. Critically, they operate in the restricted regime where the Chow-Liu algorithm converges to the correct tree. While these results are insightful in their own right, their assumptions are generally violated in our setting making their results inapplicable.

For Gaussian graphical models and Ising models, the unidentifiability properties are established in Katiyar et al. [2019] and Katiyar et al. [2020], respectively. In Tandon et al. [2021] the authors extend the results in Katiyar et al. [2019, 2020], providing better sample complexity results and a more efficient algorithm. The critical limitation of these results is that they do not extend to discrete random variables with support sizes larger than 2 and therefore fail to capture the nuanced identifiability properties demonstrated in our setting.

Finally, our problem can be posed as the latent tree graphical model estimation problem, where the noisy nodes are observed and non-noisy nodes are latent. Results for learning latent tree graphical models in Pearl and Tarsi [1986], Chang [1996], Choi et al. [2011], and *independently and concurrently* in Casanellas et al. [2021], can be used to recover the underlying tree barring the nodes within leaf clusters. Importantly, these models do not assume any structure on the noise, and thereby, contrived noise models make it impossible to recover nodes within a leaf cluster. As a result they fail to uncover the possibility of identifiability within a leaf cluster when we consider the natural k-ary symmetric channel noise model.

3 PROBLEM SETUP

Let $\mathbf{X} = [X_1, X_2 \ldots X_n]$ be the vector of random variables with a common support set, $\mathcal{S} = \{s_1, s_2, \ldots s_k\}$ such that their graphical model structure is a tree T^*. The vanilla learning problem is to recover the tree T^* from i.i.d samples of X_i.

In this paper, we consider the problem of recovering T^* but we do not get to observe samples of X_i. Instead, the samples of X_i pass through a k-ary symmetric noise channel and we observe the output denoted by X_i', that is,

$$X_i' = \begin{cases} X_i & \text{w.p. } 1 - q_i, \\ U_i & \text{w.p. } q_i, \end{cases} \quad (1)$$

where q_i is the probability of error for X_i and U_i is a discrete random variable independent of $\mathbf{X}$ and U_j $\forall j \neq i$, distributed uniformly on $\mathcal{S}$. Note that q_i can be unequal for all X_i. The vector of the noisy random variables is denoted by $\mathbf{X}' = [X_1', X_2' \ldots X_n']$. Due to the noise in X_i, the graphical model of the nodes in $\mathbf{X}'$ is no longer given by T^*. In general, *the graphical model on the noisy random variables can be a complete graph*.

Matrix PMF and Distance Notation: We denote the joint PMF matrix for random variables (X_a, X_b), and (X_a', X_b') by the matrix $P_{a,b}$ and $P_{a',b'}$ respectively, such that:

$$(P_{a,b})_{i,j} = P(X_a = s_i, X_b = s_j),$$
$$(P_{a',b'})_{i,j} = P(X_a' = s_i, X_b' = s_j).$$

The conditional PMF of X_a conditioned on X_b is denoted by the matrix $P_{a|b}$ while the marginal distribution of random variables X_a and X_a' are denoted using diagonal matrices P_a and $P_{a'}$ respectively such that:

$$(P_{a|b})_{i,j} = P(X_a = s_i | X_b = s_j),$$
$$(P_a)_{i,i} = P(X_a = s_i), (P_{a'})_{i,i} = P(X_a' = s_i).$$

The information distance metric between two nodes proposed in Lake [1994], is defined as follows:

$$d_{i,j} = -\log \frac{|det(P_{i,j})|}{\sqrt{det(P_i)det(P_j)}}. \quad (2)$$

We require the following assumptions that are natural and standard in this line of literature (c.f. Chang [1996], Choi et al. [2011]).

Assumption 1. *The probability mass at every support for each non-noisy random variable is bounded away from 0 :* $(P_a)_{i,i} \geq p_{min} > 0$.

Assumption 2. *The distance $d_{i,j}$ between adjacent non-noisy random variables is bounded:* $0 < d_{min} < d_{i,j} < d_{max}$.

Assumption 3. *The probability of error is upper bounded away from 1:* $q_i \leq q_{max} < 1$.

Assumption 1 ensures that the probability mass at any support is not arbitrarily small for any random variable. The bounds on the distance in Assumption 2 ensure that no adjacent random variables are duplicates or independent. Assumption 3 ensures that the noisy observations are not independent of the underlying random variables. Our sample complexity lower bounds in Section 6 show that the problem becomes infeasible if these assumptions are not satisfied.

Lastly, we also formally define a *leaf cluster* as follows:

Definition 1. *The **leaf cluster** of any leaf node is the set containing that leaf node, its parent node and all its sibling leaf nodes.*

4 IDENTIFIABILITY RESULTS

In this section, we prove that the identifiability of the underlying tree is determined by the joint PMF of leaf parent pairs. We do not assume access to d_{min} and d_{max}. It is straightforward to extend it to a setting when d_{min} and d_{max} are known by ruling out any tree

which violates the distance constraints. The proof is divided in 3 parts - (i) prove that the only potential unidentifiability is within the leaf clusters of the tree, (ii) analyze the existence of valid probability of error for a tree on three nodes, (iii) extend the analysis to a generic tree and arrive at the necessary and sufficient condition for identifiability.

4.1 Potential unidentifiability is limited to leaf clusters

For any tree T^*, Katiyar et al. [2019] defined the equivalence class $\mathcal{T}_{T^*}$ to be the set of all the trees obtained by different permutations of nodes within a leaf cluster, and showed that in the Gaussian graphical model setting, $\mathcal{T}_{T^*}$ can be recovered. We show here that with a few new proof ideas, essentially the same is true for graphical models on discrete random variables with general support size k:

Lemma 1. *Suppose the random variables in* **X** *form a tree graphical model T^*. Given samples from noisy random variables X_i', it is possible to recover the equivalence class $\mathcal{T}_{T^*}$.*

Proof Idea. The proof of this lemma is similar in spirit to Katiyar et al. [2019] and so we defer the details to Appendix A. The proof depends on categorizing groups of 4 nodes as a *non-star* when 2 of the nodes lie in one subtree and the remaining 2 nodes lie in a disjoint subtree. The key new element we need for this categorization in the discrete setting for general k, is the information distance metric $d_{i,j}$ as defined in (2).
Remarks: (i) Lemma 1 is not limited to the k-ary symmetric noise channel and holds for any noise channel such that when conditioned on X_i, X_i' is independent of X_j $\forall j \in [n] \neq i$ and X_i and X_i' are not independent. This result was independently and concurrently derived in Casanellas et al. [2021]. (ii) If there are no restrictions on the noise channel, recovering $\mathcal{T}_{T^*}$ is the best we can do. That is, for every tree in $\mathcal{T}_{T^*}$, it is possible to construct a noise model that can produce the noisy observation. This analysis along with the proof of Lemma 1 is included in Appendix A.

4.2 Error Estimation for a Tree on 3 Nodes

Additional Notation for k-ary Symmetric Channel: For each random variable X_a, we define a $k \times k$ error matrix E_a as follows:

$$E_a = (1 - q_a)I + \frac{q_a}{k}O,$$

where O is a matrix of all ones. Recall that k is the common support size for all the random variables and q_a is the probability of error of X_a.
Note that $P_{a',b'}$ and $P_{a,b}$ are related as follows:

$$P_{a',b'} = E_a P_{a,b} E_b. \tag{3}$$

It is also easy to see that:

$$P_{a'} = (1 - q_a)P_a + \frac{q_a}{k}I. \tag{4}$$

Error Estimation: Suppose there exist 3 nodes such that $X_1 \perp X_3 | X_2$ and we observe X_1', X_2' and X_3' through a k-ary symmetric channel as defined in Equation (1). The conditional independence relationship gives us:

$$P_{1,3} = P_{1,2}P_2^{-1}P_{2,3}. \tag{5}$$

From Equation (3), we have $P_{1',3'} = E_1 P_{1,3} E_3$, $P_{1',2'} = E_1 P_{1,2} E_2$, $P_{2',3'} = E_2 P_{2,3} E_3$. From Equation (4), we have $P_{2'} = (1 - q_2)P_2 + \frac{q_2}{k}I$. By substituting these in Equation (5) we get the following quadratic equation with matrix coefficients in noise parameter q_2 (details in Appendix B):

$$\frac{q_2^2}{k^2}(O - kI) - \frac{q_2}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + \\ P_{2',3'}P_{1',3'}^{-1}P_{1',2'} - P_{2'} = 0, \tag{6}$$

where the 0 on the RHS is a $k \times k$ matrix of all 0s. The key insight here is that, Equation (6) depends only on the noisy observations. Therefore, in the absence of the knowledge of conditional independence relation, it can be used as a test to check if the noisy observations can potentially be explained by $X_1 \perp X_3 | X_2$. Precisely, for a graph on 3 nodes (X_1, X_2, X_3), X_2 is a potential middle node if it can satisfy Equation (6) for some noise parameter $q_2 \in [0, q_{max}]$. In other words, X_2 is a potential middle node if the following holds, with $\|\cdot\|_F$ as the Forbenius norm of a matrix:

$$\min_{0 \leq x \leq q_{max}} \| \frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) \\ + P_{2',3'}P_{1',3'}^{-1}P_{1',2'} - P_{2'} \|_F = 0. \tag{7}$$

This is equivalent to k^2 quadratic equations corresponding to each element of the matrix having a common root which lies between 0 and q_{max}. These equations need not be unique.

4.3 Extension to a generic tree

Before presenting the identifiability result, we first establish some notation. Let $\mathcal{L}$ be the set containing all the leaf nodes of the tree-structured graphical model T^*. Now, consider the subset of leaf nodes with the following property: the leaf node X_2, its parent node X_1, and any arbitrary node X_3 from the graph have a solution to Equation (7). We label this subset $\mathcal{L}^{sub} \subseteq \mathcal{L}$. $\mathcal{T}_{T^*}^{sub} \subseteq \mathcal{T}_{T^*}$ represents the equivalence class where only leaves in $\mathcal{L}^{sub}$ can exchange positions with their parents.
The next theorem completely characterizes the identifiability of the underlying tree for a k-ary symmetric noise channel.

Theorem 1. *Suppose the random variables in* $\mathbf{X}$ *form a tree-structured graphical model* T^*. *Let* $\mathbf{X}'$ *be the observed noisy output after passing* $\mathbf{X}$ *through a* k-*ary symmetric channel. Then, we show that for any leaf node* $X_2 \in \mathcal{L}^{sub}$ *and its parent node* X_1, *equation* (7) *remains unchanged for any arbitrary third node* X_3 *from the graph. Using* $\mathbf{X}'$, *we can recover* $\mathcal{T}_{T^*}^{sub}$. *Moreover, for every tree* $\tilde{T} \in \mathcal{T}_{T^*}^{sub}$, *there exist random variables* $\tilde{\mathbf{X}}$ *and a* k-*ary symmetric channels such that the graphical model of* $\tilde{\mathbf{X}}$ *is* $\tilde{T}$ *and the* k-*ary channel output is* $\mathbf{X}'$.

Proof Idea: As the unidentifiability is only between the nodes within a *leaf cluster*, the key idea is to study a subset of 3 nodes comprising of a leaf parent pair and an arbitrary third node. It is clear that, Equation (7) has a solution when the parent node is the middle node. Whenever Equation (7) does not have a solution for a given node being a candidate center node, we can rule out the possibility of that node being a parent node. We further show that when the solution exists for a leaf node as a candidate center node, we can construct a tree where the parent node exchanges position with the leaf node. The details are presented in Appendix C.

4.4 Examples

Theorem 1 naturally gives rise to the questions - (i) is it possible that Equation (7) never has a solution, or (ii) is it possible that Equation (7) always has a solution, irrespective of the joint PMF of the random variables. In this section we demonstrate that neither of that is the case by considering 2 classes of graphical models - symmetric and perturbed symmetric graphical models. We prove that symmetric graphical models are unidentifiable, whereas perturbed symmetric graphical models are unidentifiable for $k = 3$ but are identifiable for $k \geq 4$. Finally, we show that our analysis recovers the existing results for $k = 2$. Note that we do not assume access to q_{max} and analyse the solution to Equation (7) with the constraint $0 < x < 1$. Extension to the setting of $0 < x < q_{max}$ is straightforward where we reject any solution $x > q_{max}$.

Symmetric graphical models: Symmetric graphical models are a class of graphical models where the marginals of all the random variables are uniform on the support and the conditional PMF matrix $P_{a|b}$ for random variables X_a, X_b that have an edge between them, takes the following form:

$$P_{a|b} = P_{b|a} = \alpha_{a,b} I + (1 - \alpha_{a,b}) \frac{O}{k}.$$

Recall that O is the matrix of all ones. The bounds on the distance in Assumption 2 enforces $\exp(-d_{max}/(k-1)) < \alpha_{a,b} < \exp(-d_{min}/(k-1))$.

Theorem 2. *Suppose the random variables in* $\mathbf{X}$ *form a tree graphical model* T^*. *Let* X_2 *be any leaf node and*

X_1 *be its parent node. If* $P_1 = P_2 = \frac{I}{k}$ *and* $P_{2|1} = \alpha_{2,1} I + (1 - \alpha_{2,1}) \frac{O}{k}$ *such that* $\exp(-d_{max}/(k-1)) < \alpha_{2,1} < \exp(-d_{min}/(k-1))$, *then Equation* (7) *has a solution.*

The proof is included in Appendix D. Since, Equation (7) has a solution for every leaf node X_2 as the candidate center node, using Theorem 1, we conclude that symmetric graphical models are unidentifiable.

Perturbed symmetric graphical models: We first define a $k \times k$ perturbation matrix $\Delta_{a,b}$. For a given offset $c_{a,b} \in \{1, 2, \ldots, k-1\}$, the term in the (i,j) position of $\Delta_{a,b}$ is:

$$\Delta_{a,b}(i,j) = \begin{cases} \delta_{a,b}, & \text{for } j = ((i-1+c_{a,b})\%k) + 1 \\ 0, & \text{o/w.} \end{cases}$$

In the perturbed symmetric model, the marginals continue to be uniform on the support but the conditional PMF matrix $P_{a|b}$ for adjacent X_a and X_b is modified to:

$$P_{a|b} = (\alpha_{a,b} - \delta_{a,b}) I + (1 - \alpha_{a,b}) \frac{O}{k} + \Delta_{a,b}.$$

Here $\alpha_{a,b}$ and $\delta_{a,b}$ are chosen such that Assumption 2 is satisfied. We find that perturbed symmetric graphical models are unidentifiable for $k = 3$ but become identifiable for $k \geq 4$.

Theorem 3. *Suppose the random variables in* $\mathbf{X}$ *form a tree graphical model* T^*. *Let* X_2 *be any leaf node and* X_1 *be its parent node. Suppose* $P_1 = P_2 = \frac{I}{k}$ *and* $P_{2|1} = (\alpha_{a,b} - \delta_{a,b}) I + (1 - \alpha_{a,b}) \frac{O}{k} + \Delta_{a,b}$ *such that* $|\delta_{a,b}| > 0, \alpha_{a,b} \neq \delta_{a,b}$, *and* $\alpha_{a,b}, \delta_{a,b}$ *are such that the distance assumptions in 2 are satisfied. Then, equation* (7) *has a solution for* $k = 3$, *but does not have a solution for* $k \geq 4$.

Proof Idea. The proof for $k \geq 4$ relies on lower bounding the Frobenius norm of the quadratic away from 0. In conjunction with Theorem 1, this implies that the exact tree is identifiable when $k \geq 4$. For $k = 3$, we explicitly calculate the solution to Equation (7). Note that, for $k = 3$ the class of symmetric and perturbed symmetric graphical models together comprise all the joint PMF matrices that are circulant. In fact, for $k = 3$, when the marginals are uniformly distributed, the joint PMF matrix being circulant is a necessary and sufficient condition for unidentifiability. These details are presented in Appendix E.

Unidentifiability when $k = 2$**:** We now discuss the unidentifiability for $k = 2$.

Lemma 2. *Suppose the random variables in* $\mathbf{X}$ *have support size* $k = 2$ *and they form a tree graphical model* T^*. *The random variables in* $\mathbf{X}$ *pass through a binary*
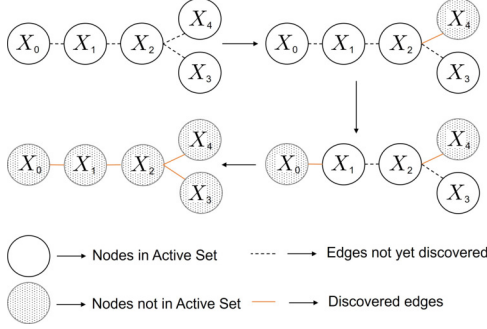
Figure 1: At each iteration, the algorithm uses the GetLeafParent routine to find a leaf parent pair from the active set of nodes, registers the edge between them, and removes the leaf node from the active set of nodes.

symmetric channel with positive probability of error and we observe $\mathbf{X}'$. *For any 3 nodes* (X_1, X_2, X_3), *Equation* (7) *always has a valid solution.*

The proof of Lemma 2 is in Appendix F. Corollary 1 recovers the unidentifiability results of Katiyar et al. [2020].

Corollary 1. *When the random variables in* $\mathbf{X}$ *have a support size of 2 and all the parents of leaf nodes have non-zero noise, we have* $\mathcal{T}_{T^*}^{sub} = \mathcal{T}_{T^*}$.

5 ALGORITHM

In this section, we present the algorithm to recover a tree from $\mathcal{T}_{T^*}^{sub}$ given samples corrupted by a k-ary symmetric noise channel as inputs.

Key Idea: The algorithm to recover the tree is an iterative one. During an iteration, we have an active set of nodes which are guaranteed to form a subtree. At each iteration, we find a leaf parent pair in the subtree, record that edge, and remove the leaf node from the active set of nodes. The algorithm to recover the tree structure is presented in Algorithm 1. A toy example illustrating the steps of the algorithm in the completely identifiable setting is presented in Figure 1.

Algorithm 1 Recover Tree Structure

Input: Pairwise noisy distributions, $P'_{i,j} \ \forall i, j \in [n]$
Output: List of edges, *Edges*
 1: **procedure** FindTree($P'_{i,j} \ \forall i, j \in [n]$)
 2:     $ActiveSet \leftarrow \{1, \dots n\} \ Edges \leftarrow \{\} \ Parents \leftarrow \{\}$
 3:     **while** $|ActiveSet| > 2$ **do**
 4:         $leaf, parent \leftarrow$ GetLeafParent($P'_{i,j}, \dots$
 5:                     $ActiveSet, Edges, Parents$)
 6:         $ActiveSet \leftarrow ActiveSet \setminus leaf$
 7:         $Edges \leftarrow Edges \cup (leaf, parent)$
 8:         $Parents \leftarrow Parents \cup parent$
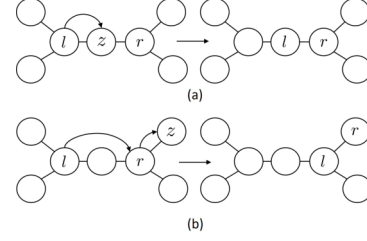 9: $Edges \leftarrow Edges \cup (ActiveSet[0], ActiveSet[1])$
10: **return** *Edges*

Figure 2: (a) If the node z lies between l and r, l becomes z, hence getting closer to r. (b) If the node r lies between l and z, both l and r shift towards the right with l becoming r and r becoming z.

Finding a leaf parent pair: We next describe the algorithm to find a leaf parent pair. We maintain two nodes - a left node l, and a right node r. The idea is to move both the nodes towards the right side till r is a leaf node and l is its parent node. In order to do this we consider a third node z and perform the following operations:

1. If the center node in (l, r, z) is z, we shift node l to node z,

2. If the center node in (l, r, z) is r, we shift node l to node r and node r to node z.

This is illustrated in Figure (2). Finding the center node can be done by checking the feasibility of Equation (7) for different candidate center nodes.

If Equation (7) has a solution for more than one nodes, we use an alternative method which uses the 3 nodes in conjunction with different 4th nodes. These 4 nodes are categorized as star/non-star to arrive at the center node. While doing the test for the center node, we only consider the nodes with pairwise distances smaller than $4d_{max} + 3\eta_{max}$. Here η_{max} is an upper bound on the distance between a clean and noisy node. For a given p_{min} and q_{max} from Assumption 1 and 3 respectively, $\eta_{max} = (1 - k) \log(1 - q_{max}) - 0.5k \log p_{min}$ (details in Appendix G). This makes it easy to adapt the algorithm for the finite sample setting.

Finite sample algorithm: The finite sample version of the algorithm uses the empirical estimate of the joint PMF of random variables to test for the center node given a set of three nodes. We only perform the test for nodes that whose empirical distance is small to avoid a sample complexity exponential in the diameter of the graph. For the test of center node by checking for existence of a solution to Equation (7) using empirical PMF estimates, we need the following additional assumption:

Assumption 4. *When Equation* (7) *does not have a*

solution, we have the following inequality:

$$\min_{0 \leq x \leq q_{max}} \| \frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'}P_{1',3'}^{-1}P_{1',2'} - P_{2'} \|_F > t_0.$$

This assumption ensures that when Equation (7) does not have a solution for a leaf node X_2 as a center node, it can be detected in the presence of perturbations due to finite samples. In Appendix G, we provide the details of the algorithm including finding the center node, and necessary modifications for executing the algorithm using finite samples. In addition, we also include the pseudocode and the proof of correctness of the algorithm.

Insights into the input parameters of the algorithm: The algorithm in its vanilla form requires d_{min}, d_{max}, q_{max}, p_{min} and t_0 in addition to the noisy samples as inputs. While the dependence on the knowledge of q_{max} is necessary, it is possible to obtain estimates of bounds of d_{min} and d_{max} using the noisy samples. This comes at the cost of higher sample complexity. Dependence on t_0 can also be avoided at the cost of higher time complexity. This is detailed as follows:

- The upper bound on d_{max} is denoted by $\tilde{d}_{max}$. It is defined as $\tilde{d}_{max} = \max_i \min_{j \neq i} d_{i'j'}$. This bound can potentially be loose by $2\eta_{max}$.
- If the ground truth is such that $d_{min} - 2\eta_{max} > 0$ then a lower bound on d_{min}, denoted by $\tilde{d}_{min}$, can be defined as $\tilde{d}_{min} = \min_i \min_{j \neq i} d_{i'j'} - 2\eta_{max}$. This bound can also be loose by $2\eta_{max}$.
- If p_{min} and q_{max} are such that $p_{min} > q_{max}$ then a valid lower bound on p_{min} is $\min_i (P_{a'})_{i,i} - q_{max}$ which can potentially be loose by q_{max}.
- In the absence of the knowledge of t_0, we can use the star/non-star test for finding the center node among 3 nodes as long as no 2 nodes belong to the same *leaf cluster*. This increases the time complexity of finding the center node from $\mathcal{O}(1)$ to $\mathcal{O}(n)$. Once we get nodes within the same *leaf cluster*, the potential center node with the minimum objective function in Equation (7) is chosen as the center node.

6 SAMPLE COMPLEXITY RESULTS

In this section, we provide both the sample complexity upper bounds and sample complexity lower bounds for recovering the tree using our algorithm in presence of corrupted samples.

Theorem 4 (Sample Complexity Upper Bound). *Suppose the random variables in $\mathbf{X}$ form a tree graphical*

model T^* and we observe $\mathbf{X}'$ such that Assumptions 1, 2, 3 and 4 are satisfied. Then there exists N_0, where

$$N_0 = \mathcal{O}\left(\max \left\{ \frac{k^4 \exp(8d_{\max})}{(1-q_{max})^{6(k-1)}(kp_{min})^{5k}(1-\exp(-2d_{min}))^2}, \right. \right.$$
$$\left. \left. \frac{k^5 \exp(16d_{\max})}{t_0^2(1-q_{max})^{12(k-1)}(kp_{min})^{10k}} \right\} \log\left(\frac{2nk(n-1)}{\delta} \right) \right)$$

such that when $N > N_0$, the finite sample Algorithm 1 correctly recovers $\mathcal{T}_{T^*}^{sub}$ with probability at least $1 - \delta$.

In the unidentifiable setting, since Equation (7) always has a solution, our algorithm finds more than one candidate center nodes and therefore resorts to the star/non-star test for finding the center node. In the sample complexity, the second term in the max comes from the quadratic test and therefore it can be dropped. As a result, since we have an easier learning problem of learning only $\mathcal{T}_{T^*}$, the sample complexity has better dependence on d_{max}, q_{max} and p_{min}. The resulting sample complexity becomes:

$$\mathcal{O}\left(\frac{k^4 \exp(8d_{\max})}{(1-q_{max})^{6(k-1)}(kp_{min})^{5k}(1-\exp(-2d_{min}))^2} \right) \log \frac{2nk(n-1)}{\delta}.$$

Theorem 5 (Sample Complexity Lower Bound). *Suppose the random variables in $\mathbf{X}$ form a tree graphical model T^* and we observe $\mathbf{X}'$ such that Assumptions 1, 2, 3 and 4 are satisfied. Then any algorithm that correctly recovers $\mathcal{T}_{T^*}^{sub}$ with probability at least $1 - \delta$ requires N samples where*

$$N = \Omega\left(\frac{d_{max}\exp(d_{max})}{(1-\exp(-d_{min}))(1-q_{max})}k(1-\delta)\log n \right).$$

Furthermore, for $k \geq 4$, $0 < t_0 \leq \frac{k}{10}\exp(-2\frac{d_{\max}}{k-1})$, we additionally have

$$N = \Omega\left(\frac{\max\limits_{d \in \{d_{\max}, d_{\min}\}} e^{-\frac{2d}{k-1}}\left(1 - e^{-\frac{d}{k-1}}\right)k(1-\delta)\log(n)}{t_0^2} \right).$$

We note that our lower bounds on sample complexity show our certain dependence on the problem parameters cannot be improved orderwise. Firstly, we see the dependence on the graph size scales as $\Theta(\log(n))$ which is standard in graphical model learning. We observe that the sample complexity scales as $\exp(\Theta(d_{\max}))$ as a function of the d_{max}. Furthermore, for small enough t_0 and support size 4 or more, the dependence t_0 scales as $\Theta(\frac{1}{t_0^2})$ highlighting the significance of Assumption 4 in the recovery of MRFs.

Our lower bound proof for t_0 dependence in the (partially) identifiable case uses a family of $(n+1)$ star graphs with n edges each, where one graph is a perturbed symmetric graphical model (Section 4.4), and for the other graphs we select one edge each and replace the conditional PMF with the one from a symmetric model. Thus, the equivalence class $\mathcal{T}_{T^*}^{sub}$ for each graph

in the family is unique. For the lower bounds in the unidentifiable scenario, we generalize the construction in Tandon et al. [2021] to $k > 2$ support size by introducing a novel class of graphical models called *Partially Mixing Graphical Models* (PMGM). Our derivation for KL divergence for PMGM, and perturbed symmetric graphical models used in the lower bound proofs can be of independent interest. These details can be found in Appendix I.

7 EXPERIMENTS

In this section, we present the experiments demonstrating the efficacy of our algorithm. We first demonstrate the performance of our algorithm for the $k = 2$ setting and demonstrate that our algorithm considerably outperforms the algorithm in Tandon et al. [2021]. Next, we showcase the performance of our algorithm for the $k = 4$ setting with the perturbed symmetric model. As discussed in Section 4.4, the exact tree is identifiable in this scenario.

7.1 Support size, $k = 2$ (Unidentifiable):

In this part, we compare the performance of our algorithm for chain and star graphs to that of SGA proposed in Tandon et al. [2021]. We use the exact same settings as in Tandon et al. [2021] and demonstrate that we outperform SGA.

For chain graphs, the nodes are labeled X_1 to X_{12} from left to right. The star graphs have X_1 as the center node and $X_2, \ldots X_{12}$ are leaf nodes connected to X_1.

Setting: (i) Number of nodes $= 12$. (ii) Correlation of all the adjacent nodes $= \rho$. (iii) Alternate nodes have maximum noise ($q_i = 0$ if $i \% 2 = 0$, $q_i = q_{max}$ if $i \% 2 = 1$). (iv) Assume access to ρ.

For both, chain graphs and star graphs, we vary ρ in $\{0.6, 0.8\}$ and q_{max} in $\{0, 0.4\}$.

We would like to point out that q_{max} is defined differently in our setting and in SGA; q_{max} in our setting is twice the SGA's q_{max}. The final results are presented in Figures 3a and 3b respectively.

7.2 Support size, $k = 4$ (Identifiable):

We present the performance of our algorithm for chain graphs for the perturbed symmetric model.

Setting: (i) Number of nodes $= 7$. (ii) Distance of all the adjacent nodes $= d$. (iii) Error probability is uniformly samples from $[0, q_{max}]$. *Note that the error probability has not been adversarially chosen but still causes Chow-Liu algorithm to fail, thereby demonstrating the fragility of Chow-Liu algorithm to noise.* (iv) Assume access to q_{max}, d_{min} but not to d_{max}, t_0.

While fixing $d = 0.7$, $q_{max} = 0.2$, we vary δ over

$\{0.01, 0.03, 0.05\}$ to study its impact on the performance of the algorithm (see, Figure 4).

We observe, on expected lines, that our algorithm converges faster for larger δ. The Chow-Liu algorithm consistently converges to the incorrect tree.

We also perform extensive experiments with more graph structures as well as evaluate the impact of the probability of error, number of nodes, distance between adjacent nodes and present the results in Appendix J.

8 CONCLUSION AND FUTURE WORK

In this paper, we highlight the nuances of the recoverability of tree structured Markov Random Fields when the samples are corrupted by symmetric noise. We develop an algorithm that is robust to the symmetric noise and provide the sample complexity results. This paper lays the foundation for further research in two key directions: (1) generalization of the symmetric noise model, and (2) identifiability of non-tree structured MRFs. Firstly, when the error samples are independent but non-symmetric our methods can be extended easily. Further, one may generalize to a setting where noisy observation could come from different distributions, each with a corresponding unknown error probability. Here, Eq.(6) needs to be generalized to recover multiple unknown error probabilities. Finally, identifiability of non-tree structured MRFs requires generalizing star and non-star detection in presence of cycles. See Appendix K for more discussions.

Ashish Katiyar, Soumya Basu, Vatsal Shah, Constantine Caramanis
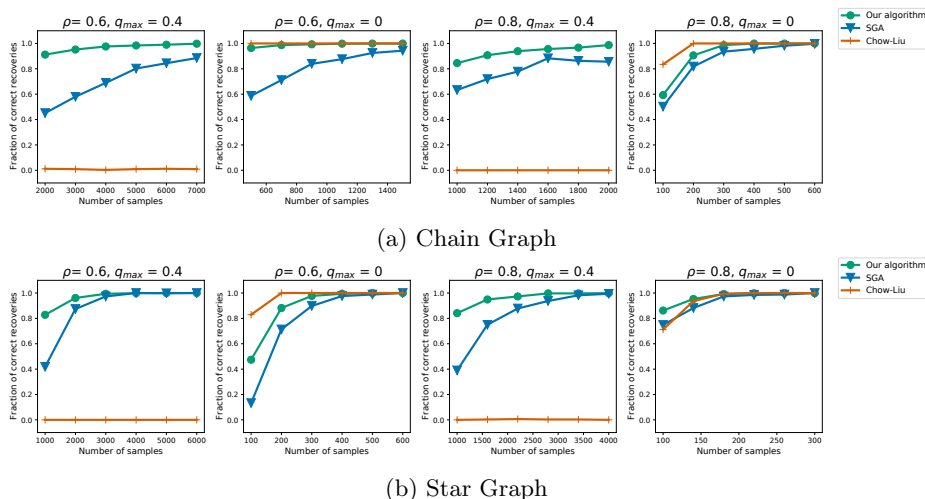

(a) Chain Graph

(b) Star Graph

Figure 3: For both chain and star graphs, our algorithm outperforms SGA for 4 different settings - (i) $\rho_{max} = 0.6, q_{max} = 0.4$, (ii) $\rho_{max} = 0.6, q_{max} = 0.0$, (iii) $\rho_{max} = 0.8, q_{max} = 0.4$, (iv) $\rho_{max} = 0.8, q_{max} = 0.0$
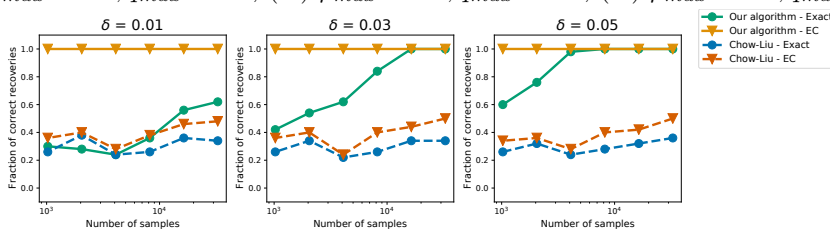

.

Figure 4: Comparing the performance of our algorithm and Chow-Liu over different values of $\delta_{i,j} \in \{0.01, 0.03, 0.05\}$ given $d_{min} = d_{max} = 0.7$, $q_{max} = 0.2$, # of nodes= 7. For both algorithms, we provide results for two cases: i) when the exact underlying tree is recovered, ii) when a tree from the equivalence class is recovered.

References

Rajendra Bhatia. *Perturbation bounds for matrix eigenvalues*. SIAM, 2007.

Arnab Bhattacharyya, Sutanu Gayen, Eric Price, and NV Vinodchandran. Near-optimal learning of tree-structured distributions by Chow-Liu. *arXiv preprint arXiv:2011.04144*, 2020.

Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 771–782. ACM, 2015.

Guy Bresler and Mina Karzand. Learning a tree-structured ising model in order to make predictions. *arXiv preprint arXiv:1604.06749*, 2016.

Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356. Springer, 2008.

Guy Bresler, David Gamarnik, and Devavrat Shah. Hardness of parameter estimation in graphical models. In *Advances in Neural Information Processing Systems*, pages 1062–1070, 2014a.

Guy Bresler, David Gamarnik, and Devavrat Shah. Structure learning of antiferromagnetic ising models. In *Advances in Neural Information Processing Systems*, pages 2852–2860, 2014b.

Guy Bresler, Mina Karzand, et al. Learning a tree-structured ising model in order to make predictions. *Annals of Statistics*, 48(2):713–737, 2020.

Marta Casanellas, Marina Garrote-López, and Piotr Zwiernik. Robust estimation of tree structured models. *arXiv preprint arXiv:2102.05472*, 2021.

Joseph T Chang. Full reconstruction of markov models on evolutionary trees: identifiability and consistency. *Mathematical biosciences*, 137(1):51–73, 1996.

Myung Jin Choi, Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. Learning latent tree graphical models. *Journal of Machine Learning Research*, 12(May):1771–1812, 2011.

C Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

Constantinos Daskalakis and Qinxuan Pan. Tree-structured ising models can be learned efficiently. *arXiv preprint arXiv:2010.14864*, 2020.

Ilias Diakonikolas, Daniel M Kane, Alistair Stewart, and Yuxin Sun. Outlier-robust learning of ising models under dobrushin's condition. In *Conference on Learning Theory*, pages 1645–1682. PMLR, 2021.

Surbhi Goel, Daniel M Kane, and Adam R Klivans. Learning ising models with independent failures. *arXiv preprint arXiv:1902.04728*, 2019.

Linus Hamilton, Frederic Koehler, and Ankur Moitra. Information theoretic properties of markov random fields, and their algorithmic applications. In *Advances in Neural Information Processing Systems*, pages 2463–2472, 2017.

Ashish Katiyar, Jessica Hoffmann, and Constantine Caramanis. Robust estimation of tree structured gaussian graphical models. In *International Conference on Machine Learning*, pages 3292–3300, 2019.

Ashish Katiyar, Vatsal Shah, and Constantine Caramanis. Robust estimation of tree structured ising models. *arXiv preprint arXiv:2006.05601*, 2020.

Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354. IEEE, 2017.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

James A Lake. Reconstructing evolutionary trees from dna and protein sequences: paralinear distances. *Proceedings of the National Academy of Sciences*, 91 (4):1455–1459, 1994.

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of markov networks using l_1-regularization. In *Advances in neural Information processing systems*, pages 817–824, 2007.

Erik M Lindgren, Vatsal Shah, Yanyao Shen, Alexandros G Dimakis, and Adam Klivans. On robust learning of ising models. In *NeurIPS Workshop on Relational Representation Learning*, 2019.

Konstantinos E Nikolakakis, Dionysios S Kalogerias, and Anand D Sarwate. Learning tree structures from noisy data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1771–1782, 2019.

Konstantinos E. Nikolakakis, Dionysios S. Kalogerias, and Anand D. Sarwate. Information thresholds for non-parametric structure learning on tree graphical models, 2020.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

Judea Pearl and Michael Tarsi. Structuring causal trees. *Journal of Complexity*, 2(1):60–77, 1986.

Adarsh Prasad, Vishwak Srinivasan, Sivaraman Balakrishnan, and Pradeep Ravikumar. On learning ising models under huber's contamination model. *Advances in neural information processing systems*, 33:16327–16338, 2020.

Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al. High-dimensional ising model selection using l1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

Leon Sot. Simple identity involving q-pochhammer symbol. Mathematics Stack Exchange. URL https://math.stackexchange.com/q/2081765. URL:https://math.stackexchange.com/q/2081765 (version: 2017-01-03).

Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. Learning gaussian tree models: Analysis of error exponents and extremal structures. *IEEE Transactions on Signal Processing*, 58(5):2701–2714, 2010.

Vincent YF Tan, Animashree Anandkumar, Lang Tong, and Alan S Willsky. A large-deviation analysis of the maximum-likelihood learning of markov tree structures. *IEEE Transactions on Information Theory*, 57(3):1714–1735, 2011.

Anshoo Tandon, Vincent YF Tan, and Shiyao Zhu. Exact asymptotics for learning tree-structured graphical models with side information: Noiseless and noisy samples. *arXiv preprint arXiv:2005.04354*, 2020.

Anshoo Tandon, Aldric HJ Yuan, and Vincent YF Tan. Sga: A robust algorithm for partial recovery of tree-structured graphical models with noisy samples. *arXiv preprint arXiv:2101.08917*, 2021.

Joel A Tropp. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015.

Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference.* Now Publishers Inc, 2008.

Shanshan Wu, Sujay Sanghavi, and Alexandros G Dimakis. Sparse logistic regression learns all discrete pairwise graphical models. In *Advances in Neural Information Processing Systems*, pages 8071–8081, 2019.
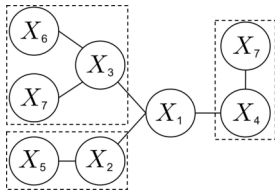
Figure 5: The equivalence class of this tree is given by all the permutations of nodes in the leaf clusters within the dotted regions. Nodes (X_1, X_4, X_6, X_7) form a non-star. Nodes (X_1, X_2, X_3, X_4) form a star.
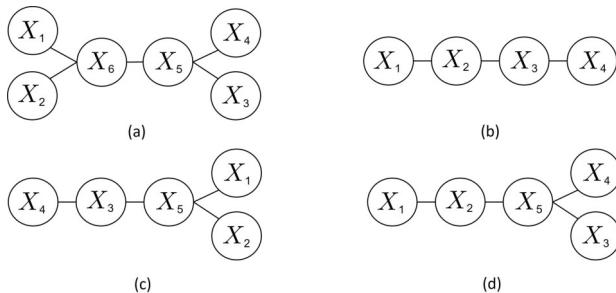

Figure 6: Four possible configurations of (X_1, X_2, X_3, X_4) when they form a non-star such that (X_1, X_2) form a pair.

A   Proof of Lemma 1

This proof relies on the classification of a set of 4 nodes as star/non-star. This graph theoretic concept was originally introduced in Katiyar et al. [2019] where it was used to analyze Gaussian graphical models. Any set of 4 nodes is classified as a non-star if the tree can be split into two subtrees with each subtree containing exactly 2 nodes. The nodes in the same subtree form a pair. If the 4 nodes do not form a non-star, they form a star. Figure 5 provides an example of the equivalence class and star/non-star classification. Once a set of 4 nodes is classified as a star/non-star, Lemma 1 follows directly from the proof of Theorem 2 in Katiyar et al. [2019]. The key idea is that the categorization of any set of 4 nodes as star/non-star completely defines all the splits of the tree into two subtrees with each subtree having at least 2 nodes. All of these splits can be combined to recover the equivalence class $\mathcal{T}_{T^*}$.

Next, we see how to categorize a set of 4 nodes as star non-star. In Katiyar et al. [2019] and Katiyar et al. [2020], the classification of a set of 4 nodes as star/non-star was done using pairwise correlations between random variables. Unfortunately, for random variables on support sizes larger than 2, correlation cannot be used to perform this classification. This is where we use the information distance metric $d_{i,j}$ as defined in Equation (2).

A set of 4 nodes (X_1, X_2, X_3, X_4) forms a non-star with (X_1, X_2) forming a pair if:

$$d_{1',3'} + d_{2',4'} = d_{1',4'} + d_{2',3'} \neq d_{1',2'} + d_{3',4'}.$$

The set forms a star if:

$$d_{1',3'} + d_{2',4'} = d_{1',4'} + d_{2',3'} = d_{1',2'} + d_{3',4'}.$$

Next, we see why these conditions for star/non-star classification are correct.

Non-Star condition: When any 4 nodes (X_1, X_2, X_3, X_4) form a non-star such that (X_1, X_2) form a pair, the 4 nodes can have one of the four configurations as shown in Figure 6. There exist more configurations with X_1 and X_2 exchanging positions or X_3 and X_4 exchanging positions. Since X_1 and X_2 always occur interchangeably, the results continue to hold for the configurations where X_1 and X_2 exchange positions. Same argument holds for X_3 and X_4.

Note that the distances $d_{i,j}$ are additive along the paths connecting X_i and X_j. Therefore for all the cases, it is easy to see that:
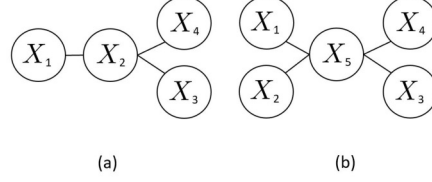
$$d_{1,3} + d_{2,4} = d_{1,4} + d_{2,3}.$$

Figure 7: Two possible configurations of (X_1, X_2, X_3, X_4) when they form a star.

Therefore we have that :

$$d_{1,3} + d_{2,4} + d_{1,1'} + d_{2,2'} + d_{3,3'} + d_{4,4'} = d_{1,4} + d_{2,3} + d_{1,1'} + d_{2,2'} + d_{3,3'} + d_{4,4'},$$

$$d_{1',3'} + d_{2',4'} = d_{1',4'} + d_{2',3'} (\text{As } d_{i',j'} = d_{i,i'} + d_{i,j} + d_{j,j'}).$$

Furthermore, one can see that

$$d_{1,3} + d_{2,4} - (d_{1,2} + d_{3,4}) \geq 2d_{min}.$$

Adding and subtracting the noise distances again, we get that

$$d_{1',3'} + d_{2',4'} - (d_{1',2'} + d_{3',4'}) \geq 2d_{min}.$$

Star condition: When the 4 nodes form a star, they can have either of the two configurations in Figure 7. All the nodes are allowed to exchange positions with each other. Using the distance additivity for this setting, it is easy to see that, for both the cases,

$$d_{1,3} + d_{2,4} = d_{1,4} + d_{2,3} = d_{1,2} + d_{3,4}.$$

Furthermore using $d_{i',j'} = d_{i,i'} + d_{i,j} + d_{j,j'}$, we get that

$$d_{1',3'} + d_{2',4'} = d_{1',4'} + d_{2',3'} = d_{1',2'} + d_{3',4'}.$$

This concludes the proof that the distances between noisy random variables can be used to classify a set of 4 nodes as star/non-star thereby proving that the only unidentifiability could possibly be within a leaf cluster.

B   Obtaining Equation (6)

From Equation (3), we have $P_{1',3'} = E_1 P_{1,3} E_3$, $P_{1',2'} = E_1 P_{1,2} E_2$, $P_{2',3'} = E_2 P_{2,3} E_3$. From Equation (4), we have $P_{2'} = (1 - q_2)P_2 + \frac{q_2}{k}I$. Substituting these in Equation (5), we get:

$$P_{2',3'} P_{1',3'}^{-1} P_{1',2'} = E_2 \frac{1}{(1 - q_2)} \left(P_{2'} - \frac{q_2}{k}I \right) E_2 \tag{8}$$

$$P_{2',3'} P_{1',3'}^{-1} P_{1',2'} = E_2 \frac{1}{(1 - q_2)} \left(P_{2'} - \frac{q_2}{k}I \right) E_2 \tag{9}$$

$$P_{2',3'} P_{1',3'}^{-1} P_{1',2'} = \frac{E_2}{1 - q_2} (1 - q_2) \left(P_{2'} - \frac{q_2}{k}I \right) \frac{E_2}{1 - q_2}$$

$$\left(\frac{E_2}{1 - q_2} \right)^{-1} P_{2',3'} P_{1',3'}^{-1} P_{1',2'} \left(\frac{E_2}{1 - q_2} \right)^{-1} = (1 - q_2) \left(P_{2'} - \frac{q_2}{k}I \right) \tag{10}$$

Note that:

$$\frac{E_2}{1 - q_2} = I + \frac{q_2 O}{k(1 - q_2)}$$

$$\left(\frac{E_2}{1 - q_2} \right)^{-1} = I - \frac{q_2 O}{k} \tag{11}$$

Substituting this back in Equation (10)

$$\left(I - \frac{q_2 O}{k}\right) P_{2',3'} P_{1',3'}^{-1} P_{1',2'} \left(I - \frac{q_2 O}{k}\right) = (1 - q_2)\left(P_{2'} - \frac{q_2}{k}I\right)$$

$$\frac{q_2^2}{k^2}(OP_{2',3'}P_{1',3'}^{-1}P_{1',2'}O - kI) - \frac{q_2}{k}(OP_{2',3'}P_{1',3'}^{-1}P_{1',2'} + P_{2',3'}P_{1',3'}^{-1}P_{1',2'}O - kP_{2'} - I)$$

$$+ P_{2',3'}P_{1',3'}^{-1}P_{1',2'} - P_{2'} = 0 \tag{12}$$

To simplify this, we observe that:

$$OP_{2',3'} = OP_{1',3'} = OP_3'$$
$$P_{2',3'}O = P_{2'}O$$
$$P_{1',2'}O = P_{1',3'}O = OP_1' \tag{13}$$
$$OP_{1',2'} = OP_{2'}$$

Substituting these back in Equation (12), we get:

$$\frac{q_2^2}{k^2}(O - kI) - \frac{q_2}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'}P_{1',3'}^{-1}P_{1',2'} - P_{2'} = 0 \tag{14}$$

C   Proof of Theorem 1

Proof. Note that a graphical model on any subset of 3 nodes comprising of a leaf node X_2, it's parent X_1 and an arbitrary third node X_3 always forms a tree and satisfies $X_2 \perp X_3 | X_1$. However, due to the unidentifiability between X_2 and X_1, we don't know a priori whether $X_2 \perp X_3 | X_1$ or $X_1 \perp X_3 | X_2$. Therefore, we attempt to estimate the probability of error for both the cases using an equation equivalent to Equation (7). All the cases for which the equation has a feasible solution can explain the noisy observations.

Clearly, the case corresponding to the ground truth $X_2 \perp X_3 | X_1$ has a solution.

We denote the error estimated for node X_a which enforces $X_b \perp X_c | X_a$ by $\tilde{q}_a^{b,c}$ and we also define the matrix $\tilde{E}_a^{b,c}$ as:

$$\tilde{E}_a^{b,c} = (1 - \tilde{q}_a^{b,c})I + \frac{\tilde{q}_a^{b,c}}{k}O.$$

Now we see what happens when we check whether node X_2 is the middle node by solving Equation (8) when the ground truth has node 1 in the middle. That is, we try to estimate $\tilde{q}_2^{1,3}$ when $X_2 \perp X_3 | X_1$.

In the current setting, we have:

$$P_{2,3} = P_{2,1}P_1^{-1}P_{1,3}.$$

We also have:

$$P_{2,3}' = E_2 P_{2,3} E_3, \quad P_{1,3}' = E_1 P_{1,3} E_3, \quad P_{1,2}' = E_1 P_{1,2} E_2.$$

Substituting these in Equations (8) and (6), we get:

$$E_2 P_{2,1} P_1^{-1} P_{1,2} E_2 = \tilde{E}_2^{1,3} \frac{1}{(1 - \tilde{q}_2^{1,3})}\left(P_{2'} - \frac{\tilde{q}_2^{1,3}}{k}I\right)\tilde{E}_2^{1,3} \tag{15}$$

$$\text{s.t.} 0 \le \tilde{q}_2^{1,3} < 1,$$

$$\frac{(\tilde{q}_2^{1,3})^2}{k^2}(O - kI) - \frac{\tilde{q}_2^{1,3}}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I)$$

$$+ E_2 P_{2,1} P_1^{-1} P_{1,2} E_2 - P_{2'} = 0 \text{ s.t. } 0 \le \tilde{q}_2^{1,3} < 1. \tag{16}$$

Note that this equation does not depend on the random variable X_3. Therefore, whether a leaf node and its parent are unidentifiable depends solely on the joint distribution of the parent node X_1 and the noisy leaf node X_2'. When this equation does not have a solution, we can conclude that X_2 is a leaf node. Thus any tree in $\mathcal{T}_{T^*}$ which has X_1 as a leaf node can be ruled out.

Now, let us focus on the case when Equation (16) has a solution. We aim to obtain $\tilde{\mathbf{X}}$ whose graphical model is $\tilde{T}$. In order to do that, we assign the probability of error $\tilde{q}_i$ which resulted in each of the observed noisy random variable X'_i as follows:

$$\tilde{q}_1 = 0, \tilde{q}_2 = \tilde{q}_2^{1,3}, \tilde{q}_i = q_i \ \forall i \notin \{1, 2\}. \tag{17}$$

Therefore we have that $\tilde{X}_i = X_i \ \forall i \notin \{1, 2\}$. Note that, by construction, this results in $\tilde{X}_1 \perp X_i | \tilde{X}_2 \ \forall i \notin \{1, 2\}$. We next prove that for any pair of nodes such that $X_{k_1} \perp X_{k_2} | X_1$ and $k_1, k_2 \notin \{1, 2\}$, we have that $X_{k_1} \perp X_{k_2} | \tilde{X}_2$. This is equivalent to proving that $P_{k_1,k_2} = P_{k_1,\tilde{2}} P_{\tilde{2}}^{-1} P_{\tilde{2},k_2}$ where $P_{k_1,\tilde{2}}, P_{\tilde{2}}$ and $P_{\tilde{2},k_2}$ are the joint PMF matrix of X_{k_1} and $\tilde{X}_2$, diagonal marginal of $\tilde{X}_2$, and the joint PMF matrix of $\tilde{X}_2$ and X_{k_2} respectively. We have that:

$$P_{k_1,2} = P_{k_1,1} P_1^{-1} P_{1,2}, \ \ P_{2,k_2} = P_{2,1} P_1^{-1} P_{k_2,1}.$$

Substituting these in $P_{k_1,k_2} = P_{k_1,1} P_1^{-1} P_{1,k_2}$, we get:

$$P_{k_1,k_2} = P_{k_1,2} P_{1,2}^{-1} P_1 P_{2,1}^{-1} P_{2,k_2}.$$

Note that $P_{k_1,2} E_2 = P_{k_1,\tilde{2}} \tilde{E}_2^{1,3} = P_{k_1,2'}$. Using this along with Equation (15) we get $P_{k_1,k_2} = P_{k_1,\tilde{2}} P_{\tilde{2}}^{-1} P_{\tilde{2},k_2}$.

The above analysis of ruling out the trees with X_1 as a leaf node when Equation (16) does not have a solution and constructing $\tilde{\mathbf{X}}$ when Equation (16) has a solution, holds true for every pair of parent and leaf nodes. Thus any tree in $\mathcal{T}_{T^*} \setminus \mathcal{T}_{T^*}^{sub}$ can be ruled out. Furthermore, for any tree $\tilde{T} \in \mathcal{T}_{T^*}^{sub}$ in which leaf nodes $\mathcal{L}_{\tilde{T}} \subseteq \mathcal{L}^{sub}$ exchange positions with their parents, we can define the probability of error for $\tilde{q}_i$ for every node $\tilde{X}_i \in \tilde{\mathbf{X}}$ as follows:

$$\tilde{q}_i = \tilde{q}_i^{p_i,3} \ \forall i \in \mathcal{L}_{\tilde{T}},$$
$$\tilde{q}_{p_i} = c_i \ \forall i \in \mathcal{L}_{\tilde{T}},$$
$$\tilde{q}_i = q_i \ \text{otherwise},$$

where X_{p_i} is the parent node of X_i and $0 \le c_i \le q_{max}$. It is straightforward to see that the graphical model of $\tilde{\mathbf{X}}$ is $\tilde{T}$. $\qquad \square$

D Proof of Theorem 2

We first present a simple equation that helps in working with symmetric and perturbed symmetric models:

$$\left(\alpha_1 I + (1 - \alpha_1)\frac{O}{k}\right)\left(\alpha_2 I + (1 - \alpha_2)\frac{O}{k}\right) = \left(\alpha_1 \alpha_2 I + (1 - \alpha_1 \alpha_2)\frac{O}{k}\right). \tag{18}$$

When X_2 is a leaf node, X_1 is its parent node and X_3 is an arbitrary third node, $X_3 \perp X_2 | X_1$. This gives us:

$$P_{2,3} = P_{2,1} P_1^{-1} P_{1,2}.$$

Substituting this in $P_{2',3'} P_{1',3'}^{-1} P_{1',2'}$ while noting that $P_{a',b'} = E_a P_{a,b} E_b$, we get that:

$$P_{2',3'} P_{1',3'}^{-1} P_{1',2'} = E_2 P_{2,1} P_1^{-1} P_{1,2} E_2. \tag{19}$$

Now, using $P_1 = I/k$, $P_{2|1} = \alpha_{1,2} I + (1 - \alpha_{1,2})\frac{O}{k}$, $E_2 = (1 - q_2)I + q_2\frac{O}{k}$ and Equation 18, we get that:

$$P_{2',3'} P_{1',3'}^{-1} P_{1',2'} = E_2 P_{2,1} P_1^{-1} P_{1,2} E_2 = \frac{1}{k}\left((1 - q_2)^2 \alpha_{1,2}^2 I + (1 - (1 - q_2)^2 \alpha_{1,2}^2)\frac{O}{k}\right).$$

With these expressions, along with $P_{2'} = \frac{I}{k}$, we now look at the quadratic in Equation (7).

$$\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(O P_{2'} + P_{2'} O - k P_{2'} - I) + P_{2',3'} P_{1',3'}^{-1} P_{1',2'} - P_{2'}$$
$$= \frac{x^2}{k^2}(O - kI) - \frac{2x}{k}(O/k - I) + \frac{1}{k}\left((1 - q_2)^2 \alpha_{1,2}^2 I + (1 - (1 - q_2)^2 \alpha_{1,2}^2)\frac{O}{k}\right) - \frac{I}{k}$$
$$= \frac{(x - 1)^2 - (1 - q_2)^2 \alpha_{2,1}^2}{k}(O - kI).$$

Thus, Equation 7 has a solution $x = 1 - (1 - q_2)\alpha_{1,2}$. $\qquad \square$

E   Proof of Theorem 3

Using Equation (19), and recalling that $P_1 = P_{1'} = P_2 = P_{2'} = \frac{I}{k}$, we have that:

$$\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'}P_{1',3'}^{-1}P_{1',2'} - P_{2'} \tag{20}$$

$$= \frac{x^2}{k^2}(O - kI) - \frac{2x}{k^2}(O - kI) + E_2 P_{2,1} P_1^{-1} P_{1,2} E_2 - \frac{I}{k} \tag{21}$$

$$= \left(\frac{x-1}{k}\right)^2 (O - kI) - \frac{O}{k^2} + kE_2 P_{2,1} P_{1,2} E_2. \tag{22}$$

Substituting $E_2 = (1 - q_2)I + q_2\frac{O}{k}$ and $P_{2,1} = (\alpha_{a,b} - \delta_{a,b})I + (1 - \alpha_{a,b})\frac{O}{k} + \Delta_{a,b}$, we get:

$$E_2 P_{2,1} P_{1,2} E_2 = \left((1 - q_2)I + q_2\frac{O}{k}\right)\left((\alpha_{a,b} - \delta_{a,b})I + (1 - \alpha_{a,b})\frac{O}{k} + \Delta_{a,b}\right) \tag{23}$$

$$\left((\alpha_{a,b} - \delta_{a,b})I + (1 - \alpha_{a,b})\frac{O}{k} + \Delta_{a,b}^T\right)\left((1 - q_2)I + q_2\frac{O}{k}\right). \tag{24}$$

Now we have:

$$E_2 P_{2,1} = \left((1 - q_2)I + q_2\frac{O}{k}\right)\left((\alpha_{a,b} - \delta_{a,b})I + (1 - \alpha_{a,b})\frac{O}{k} + \Delta_{a,b}\right)$$

$$= (1 - q_2)(\alpha_{a,b} - \delta_{a,b})I + (1 - q_2)(1 - \alpha_{a,b})\frac{O}{k} + (1 - q_2)\Delta_{a,b}$$

$$+ q_2(\alpha_{a,b} - \delta_{a,b})\frac{O}{k} + q_2(1 - \alpha_{a,b})\frac{O}{k} + q_2\delta_{a,b}\frac{O}{k}$$

$$= (1 - q_2)(\alpha_{a,b} - \delta_{a,b})I + (1 - (1 - q_2)\alpha_{a,b})\frac{O}{k} + (1 - q_2)\Delta_{a,b}$$

Define $\alpha'_{a,b} \triangleq (1 - q_2)\alpha_{a,b}, \delta'_{a,b} \triangleq (1 - q_2)\delta_{a,b}$ and $\Delta'_{a,b} = (1 - q_2)\Delta_{a,b}$, we get:

$$E_2 P_{2,1} = (\alpha'_{a,b} - \delta'_{a,b})I + (1 - \alpha'_{a,b})\frac{O}{k} + \Delta'_{a,b}.$$

Noting that $P_{1,2}E_2 = (E_2 P_{2,1})^T$, we get:

$$E_2 P_{2,1} P_{1,2} E_2 = \frac{1}{k^2}\left(((\alpha'_{a,b} - \delta'_{a,b})^2 + (\delta'_{a,b})^2)I + \frac{O}{k}(1 - (\alpha'_{a,b})^2) + (\alpha'_{a,b} - \delta'_{a,b})((\Delta'_{a,b})^T + \Delta'_{a,b})\right)$$

This gives us:

$$Q^2(x) = \left\| \left(\frac{x-1}{k}\right)^2 (O - kI) - \frac{O}{k^2} + kE_b P_{b,a} P_{a,b} E_b \right\|_F^2$$

$$= \left\| \left(\frac{x-1}{k}\right)^2 (O - kI) + ((\alpha'_{a,b} - \delta'_{a,b})^2 + \delta'^2_{a,b})\frac{I}{k} - \alpha'^2_{a,b}\frac{O}{k^2} + \frac{(\alpha'_{a,b} - \delta'_{a,b})}{k}(\Delta'^T_{a,b} + \Delta'_{a,b}) \right\|_F^2$$

Each diagonal element (total k) of the matrix is $\left(\frac{x-1}{k}\right)^2 - \frac{(x-1)^2}{k} + \frac{(\alpha'_{a,b} - \delta'_{a,b})^2 + \delta'^2_{a,b}}{k} - \frac{\alpha'^2_{a,b}}{k^2}$.

Each element at the positions of the support $(\Delta'_{a,b} + \Delta'^T_{a,b})$ (total $2k$) is $\left(\frac{x-1}{k}\right)^2 - \frac{\alpha'^2_{a,b}}{k^2} + \frac{\delta'_{a,b}(\alpha'_{a,b} - \delta'_{a,b})}{k}$.

Every remaining element (total $k^2 - 3k$) is $\left(\frac{x-1}{k}\right)^2 - \frac{\alpha'^2_{a,b}}{k^2}$. To simplify the above equation, we define $\gamma = (1 - x)^2 - \alpha'^2_{a,b}$, $e = \delta'_{a,b}(\alpha'_{a,b} - \delta'_{a,b})$. Each diagonal element is $\frac{\gamma}{k^2} - \frac{\gamma}{k} - \frac{2e}{k}$.

Each element at the positions of the support $(\Delta'_{a,b} + \Delta'^T_{a,b})$ (total $2k$) is $\frac{\gamma}{k^2} + \frac{e}{k}$.

Every remaining element (total $k^2 - 3k$) is $\frac{\gamma}{k^2}$. Thus, we get:

$$Q^2(x) = k\left(\frac{\gamma}{k^2} - \frac{\gamma}{k} - \frac{2e}{k}\right)^2 + 2k\left(\frac{\gamma}{k^2} + \frac{e}{k}\right)^2 + (k^2 - 3k)\frac{\gamma^2}{k^4}$$

$$= \frac{1}{k^3}((k-1)\gamma + 2ke)^2 + \frac{2}{k^3}(\gamma + ke)^2 + \frac{k-3}{k^3}\gamma^2$$

$Q^2(x)$ is minimized for $\gamma = -\frac{2ke}{k-1}$. Substituting this, we get:

$$Q^2(x) \geq \frac{2(k-3)e^2k^2}{k-1}.$$

When $k > 4$, $Q^2(x) \geq 0$. This completes the proof that when $k > 4$, Equation (7) does not have a solution.

Next we look at the case when $k = 3$. For $k = 3$, when $\gamma = -3e$, we get $Q^2(x) = 0$. The only thing that remains is to check that $\gamma = -3e$ corresponds to a valid solution of x.

$$(1-x)^2 - \alpha'^2_{a,b} = \gamma$$
$$(1-x)^2 - \alpha'^2_{a,b} + 3e = 0$$
$$(1-x)^2 = \alpha'^2_{a,b} - 3\delta'_{a,b}(\alpha'_{a,b} - \delta'_{a,b})$$

Note that $\alpha'^2_{a,b} - 3\delta'_{a,b}(\alpha'_{a,b} - \delta'_{a,b}) \geq \frac{\alpha'^2_{a,b}}{4}$. Also note that for $P_{2|1}$ to be a valid PMF, we need that $\alpha > \delta, 0 < \alpha < 1$. Under these constraints, it is easy to see that $\alpha'^2_{a,b} - 3\delta'_{a,b}(\alpha'_{a,b} - \delta'_{a,b}) \leq 1$. Therefore $(1-x)^2 = \alpha'^2_{a,b} - 3\delta'_{a,b}(\alpha'_{a,b} - \delta'_{a,b})$ has a solution for $0 \leq x \leq 1$. This concludes the proof that for $k = 3$, solution to Equation (7) always exists. In other words, for $k = 3$ the joint PMF matrix being circulant is a sufficient condition for unidentifiability.

Next we go on to prove that for $k = 3$, the joint PMF matrix being circulant is also a necessary condition for unidentifiability. In order to arrive at this, note that, from Equation 9, a solution exists for Equation (7) if and only if it exists for:

$$P_{2',3'}P_{1',3'}^{-1}P_{1',2'} = \tilde{E}_2^{1,3} \frac{1}{(1-\tilde{q}_2^{1,3})}\left(P_{2'} - \frac{\tilde{q}_2^{1,3}}{k}I\right)\tilde{E}_2^{1,3} \text{ s.t. } 0 \leq \tilde{q}_2^{1,3} < 1. \tag{25}$$

Recall that $\tilde{E}_2^{1,3} = (1-\tilde{q}_2^{1,3})I + \tilde{q}_2^{1,3}\frac{O}{k}$ We would like to prove that if Equation (25) has a solution then the matrix $P_{2,1}$ is circulant. Since $P_{2'} = \frac{I}{k}, P_1 = \frac{I}{k}, P_{2',3'}P_{1',3'}^{-1}P_{1',2'} = E_2P_{2,1}P_1^{-1}P_{1,2}E_2$, we have that for some $0 \leq \tilde{q}_2^{1,3} < 1$:

$$9P_{2,1}P_{1,2} = E_2^{-1}\tilde{E}_2^{1,3}\tilde{E}_2^{1,3}E_2^{-1}. \tag{26}$$

Note that $E_2^{-1} = ((1-q_2)I + q_2\frac{O}{k})^{-1} = \frac{1}{1-q_2}(I + \frac{q_2}{1-q_2}\frac{O}{k})^{-1} = \frac{1}{1-q_2}(I - \frac{\frac{q_2}{1-q_2}\frac{O}{k}}{1+\frac{q_2}{1-q_2}})$ (using Woodbury Matrix Identity). Simplifying, we get:

$$E_2^{-1} = \frac{1}{1-q_2}(I - q_2\frac{O}{k}) = \frac{1}{1-q_2}I + (1 - \frac{1}{1-q_2})\frac{O}{k}.$$

Now, using Equation (18), we get:

$$E_2^{-1}\tilde{E}_2^{1,3} = \frac{1-\tilde{q}_2^{1,3}}{1-q_2}I + \left(1 - \frac{1-\tilde{q}_2^{1,3}}{1-q_2}\right)\frac{O}{k}$$

Again, using Equation (18), we get:

$$E_2^{-1}\tilde{E}_2^{1,3}\tilde{E}_2^{1,3}E_2^{-1} = (E_2^{-1}\tilde{E}_2^{1,3})^2 = \left(\frac{1-\tilde{q}_2^{1,3}}{1-q_2}\right)^2I + \left(1 - \left(\frac{1-\tilde{q}_2^{1,3}}{1-q_2}\right)^2\right)\frac{O}{k}.$$

We note that in Equation (26), the RHS has equal off-diagonal elements and equal diagonal elements. Before proceeding further, for the ease of notation, we define $M = 3P_{1,2}$ and M_i is the ith column of M. Since Equation (26) has a solution, we have the following properties of M:

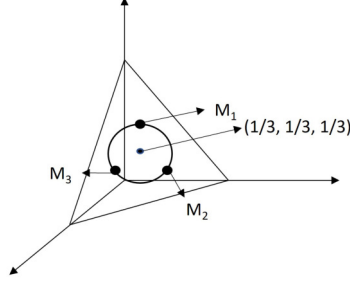1. M is doubly stochastic (as $P_1 = P_2 = I/3$),

Figure 8: Position of the three column vectors of matrix M for unidentifiability.

2. $||M_i||_2 = ||M_j||_2 \; \forall i, j \in \{1, 2, 3\}$ (as the diagonal elements of $M^T M$ are equal),

3. $< M_i, M_j >$ is equal $\forall i \neq j \in \{1, 2, 3\}$ (as the off-diagonal elements of $M^T M$ are equal).

These properties can hold true only if the columns of M are circulant. In order to see this, note that:

1. A necessary condition for property 1 is that M_1, M_2 and M_3 lie on the probability simplex.

2. For property 2 to hold, M_1, M_2 and M_3 lie on a circle on the plane of the probability simplex with center at $(1/3, 1/3, 1/3)$.

3. For property 3 to hold, M_1, M_2 and M_3 lie on an equilateral triangle of this circle.

This can be visualized in Figure (8). In order to see that they would be circulant, note that once we are given the vector M_1, vectors M_2 and M_3 are also determined. Given that we know that circulated versions of M_1 satisfy 1, 2 and 3, vectors M_2 and M_3 have to be the circulated M_1.

F   Proof of Lemma 2

We first analyze what happens to the solution of Equation (7) for 3 nodes (X_1, X_2, X_3) such that no 2 nodes are independent conditioned on the third. That is, their marginal distribution is not tree structured. We perform this analysis for general support size $k > 2$. In this case, there exists another node, say X_4, such that $X_1 \perp X_2 \perp X_3 | X_4$. This analysis is going to be useful in the proof of Lemma 2 as well as the algorithm design.

Lemma 3. *Consider any three nodes (X_1, X_2, X_3) in a tree graphical model whose marginals are not tree structured. Then there exists a node X_4 such that $X_1 \perp X_2 \perp X_3 | X_4$. Solving Equation (7) outputs X_2 as a potential center node among (X_1, X_2, X_3) if and only if it outputs X_2 as a potential center node among (X_4, X_2, X_3)*

Proof. In this setting, we would like to estimate the probability of error of X_2 using Equation (7). We have that:

$$P_{2,3} = P_{2,4} P_4^{-1} P_{4,3},$$
$$P_{1,3} = P_{1,4} P_4^{-1} P_{4,3},$$
$$P_{1,2} = P_{1,4} P_4^{-1} P_{1,2}$$

Using these expressions coupled with Equation (3) and substituting them in Equation (6) we get the following quadratic equation:

$$\frac{(\tilde{q}_2^{1,3})^2}{k^2} (O - kI) - \frac{\tilde{q}_2^{1,3}}{k} (OP_2' + P_2'O - kP_2' - I) + E_2 P_{2,4} P_4^{-1} P_{4,2} E_2 - P_2' = 0. \tag{27}$$

This is the same equation with X_1 replaced by X_4. $\qquad\square$

Next we go on to prove Lemma 2

Proof. First, let us look at the case when (X_1, X_2, X_3) form a tree. If $X_1 \perp X_3 | X_2$, solution to Equation (7) exists and it recovers the true error for X_2. We see what happens when $X_2 \perp X_3 | X_1$. We consider the case when there is no noise in X_2 and X_3. This is analysis is sufficient, as even if there was independent noise in X_2 and X_3, we would have had $X_2' \perp X_3' | X_2$. Thus we can assume that X_2 and X_3 already have the noise factored in.

For this case, we know that Equation (7) boils down to Equation (16) with $E_2 = I$. Using basic algebra, we see that all the quadratic equations corresponding to the different matrix components are equal to the following:

$$\frac{(\tilde{q}_2^{1,3})^2}{4} - \frac{(\tilde{q}_2^{1,3})}{2} + \frac{(P_{2,1})_{0,0}(P_{2,1})_{1,0}}{(P_{2,1})_{0,0} + (P_{2,1})_{1,0}} + \frac{(P_{2,1})_{0,1}(P_{2,1})_{1,1}}{(P_{2,1})_{0,1} + (P_{2,1})_{1,1}} = 0 \text{ s.t. } 0 \le \tilde{q}_2^{1,3} < 1 \qquad (28)$$

Since the entries of $P_{2,1}$ are positive and sum up to 1, the smallest root of this equation is 0 (when one of $(P_{2,1})_{0,0}, (P_{2,1})_{1,0}$ and one of $(P_{2,1})_{0,1}, (P_{2,1})_{1,1}$ are 0) and the largest root is 1 (when all entries of $P_{2,1}$ are 1/4). Since $P_{2,1}$ is full rank, we can conclude that Equation (28) has a solution.

Next, consider the case when (X_1, X_2, X_3) do not form a tree. There exists a node X_4 such that $X_1 \perp X_2 \perp X_3 | X_4$. Using the above result, we know that Equation (7) has a solution when we estimate the probability of error of X_2 which enforces $X_4 \perp X_3 | X_2$. Using Lemma 3, we conclude that Equation (7) has a solution which enforces $X_1 \perp X_3 | X_2$. □

G Algorithm Details

In this section, we provide the details of the algorithm to recover the tree upto unidentifiability. When we have access to t_0 (Assumption 4), we can recover $\mathcal{T}_{T^*}^{sub}$. In the absence of the knowledge of t_0, the algorithm returns one tree from $\mathcal{T}_{T^*}^{sub}$. We discuss the details after presenting the pseudocode. Also, if we have prior knowledge that the tree is identifiable only upto $\mathcal{T}_{T^*}$ (for instance, when $k = 2$ or for symmetric models), we can gain in runtime by $\mathcal{O}(n)$.

Obtaining η_{max} We first prove that $\eta_{max} = (1 - k) \log(1 - q_{max}) - 0.5k \log p_{min}$. First note that for any node X_i, we have that:

$$P_{i'|i} = (1 - q_i)I + q_i \frac{O}{k}.$$

Note that:

$$d_{i',i} = -\log\left(\frac{|det(P_{i',i})|}{\sqrt{det(P_{i'})det(P_i)}}\right) = -\log\left(det(P_{i'|i})\sqrt{\frac{det(P_i)}{det(P_{i'})}}\right).$$

Using the matrix determinant lemma, we get $det(P_{i'|i}) = (1 - q_i)^{k-1}$. Also $det(P_{i'}) < 1$ and $det(P_i) \ge p_{min}^k$. This gives us:

$$d_{i',i} \le (1 - k) \log(1 - q_i) - 0.5k \log p_{min} \triangleq \eta_{max}$$

Neighborhood Vectors   We define for each node X_i, a neighborhood vector $N(X_i)$, which is the array of nodes X_j sorted by $d_{i',j'}$ in ascending order and only contains nodes such that $d_{i',j'}$ is smaller than a threshold t_{real}. This is given as follows:

$$N(X_i) = sort(X_j : d_{i',j'} \le t_{real}, \text{ key } = d_{i',j'}) \qquad (29)$$

The threshold is $t_{real} = 4d_{max} + 3\eta_{max}$.

G.1 Pseudocode and runtime analysis

We first provide the pseudocode for the two building blocks - FINDCENTER and QUADRATICERROR. FINDCENTER returns the center node among 3 nodes as long as no 2 nodes are in the same leaf cluster. Otherwise it returns the nodes that belong to the same leaf cluster. QUADRATICERROR is used by the LEAFCLUSTERRESOLUTION routine to find the parent node within a leaf cluster. Using these, we present the FINDLEAFPARENT subroutine that returns a leaf parent pair given an active set of nodes that form a subtree.

G.1.1 QUADRATICERROR

In this subroutine, we test if Equation (7) has a solution. Note that the quadratic in Equation (7) with matrix coefficients is equivalent to having k^2 quadratic equations. Equation (7) has a solution if all the k^2 quadratic equations have a common root in $[0, q_{max}]$. Since we are working with the finite sample empirical estimates of the PMFs, we do not get an exact solution. To work in the finite sample domain, we find the mean of the root of all the k^2 quadratic equations and use that as an estimate of the common root. We return the Frobenius norm of the quadratic with the estimated root plugged in.

Algorithm 2 Find the Error of the quadratic in Equation (7)

Input - Pairwise noisy distributions, a set of 3 nodes, test center node among the three nodes.
Output - Error of the quadratic in Equation (7).

1: **procedure** QUADRATICERROR($P_{i',j'}, NodeTriplet, TestCenter$)
2:   $A, B, C \leftarrow$ Matrix Quadratic Coefficients from Equation (7) for given $NodeTriplet, TestCenter$.
3:   $MeanRoot \leftarrow 0$
4:   **for** i_1 in $1 \ldots k$ **do**
5:     **for** i_2 in $1 \ldots k$ **do**
6:       $MeanRoot \leftarrow MeanRoot + \frac{root(A[i_1,i_2]x^2 + B[i_1,i_2]x + C[i_1,i_2])}{k^2}$
   return $\|A(MeanRoot)^2 + B(MeanRoot) + C\|_F$

G.1.2 FINDCENTER

The key idea is based on the observation that for any 3 nodes (X_1, X_2, X_3), if X_2 is the center node, then any set of 4 nodes (X_1, X_2, X_3, j) which forms a non-star, never has (X_2, j) as a pair. Thus we can scan through all the nodes j and rule out the nodes that pair with j. This procedure could potentially detect a leaf node as the center node if its parent is the center node. However, this is as expected since using the star/non-star procedure, it is impossible to differentiate between leaf and parent nodes.

Algorithm 3 Recover Center Node in the Unidentifiable setting

---

Input - Pairwise noisy distributions and 3 nodes
Output - Candidate Center Nodes

1: **procedure** FINDCENTER($P_{i',j'}, NodeTriplet$)
2:   $x \leftarrow NodeTriplet[0], y \leftarrow NodeTriplet[1], z \leftarrow NodeTriplet[2]$
3:   $CenterCand \leftarrow \{x, y, z\}$
4:   **for** $j \in N(x) \cap N(y) \cap N(z)$ **do**
5:     **if** (x, y, z, j)- Non-star and $pair(j) \in CenterCand$ **then**
6:       $CenterCand \leftarrow CenterCand \setminus pair(j)$
   return $CenterCand$

G.1.3 GETLEAFPARENT

This routine finds a leaf parent pair given an active set of nodes that form a subtree. We maintain two nodes - a left node l, and a right node r. The idea is to move both the nodes towards the right side till r is a leaf node and l is its parent node. In order to do this we consider a third node z and perform the following operations:

1. If the center node in (l, r, z) is z, we shift node l to node z,

2. If the center node in (l, r, z) is r, we shift node l to node r and node r to node z.

Selecting nodes l, r and z: When the GETLEAFPARENT subroutine is called for the first time, node r is randomly initialized. For any subsequent calls to GETLEAFPARENT, node r is initialized to one of the nodes that was detected as a parent node in the previous iterations and is still in the active set. l is initialized to the node closest to r in terms of $d_{i',j'}$. z is obtained by iterating through $N(X_i) \setminus l$ in the increasing order of distance.

When for a given (l, r, z), there are more than one candidate center nodes, we conclude that they belong to the same leaf cluster. We check if we have already discovered the right node in one of the previous iterations if we

have, we return the leaf parent pair. Otherwise, we attempt to find the parent node in that leaf cluster using the LEAFCLUSTERRESOLUTION routine.

Algorithm 4 Find a leaf parent pair.

Input - Pairwise noisy distributions and Active nodes
Output - Leaf Node and its parent in the subtree of Active Nodes.

1: **procedure** GETLEAFPARENT($P_{i',j'}$, $ActiveSet$, $Edges$, $Parents$)
2:     **if** $|ActiveSet \cap Parents| > 0$ **then**
3:         $r \leftarrow ActiveSet \cap Parents[0]$
4:     **else**
5:         $r \leftarrow ActiveSet[0]$
6:     $l \leftarrow N(r)[0] \cap ActiveSet$
7:     $i \leftarrow 1, visited \leftarrow \{l, r\}$
8:     **while** $i < len(N(r))$ **do**
9:         $z \leftarrow N(r)[i]$
10:         **if** $z \in visited$ or $z \notin ActiveSet$ **then**
11:             $i \leftarrow i + 1$
12:             continue
13:         $visited \leftarrow visited \cup z$
14:         $C \leftarrow$ FINDCENTER($P_{i',j'}, (l, r, z)$)
15:         **if** $|C| == 1$ **then**
16:             $l\_r\_order = True$
17:         **if** $C == z$ **then**
18:             $l \leftarrow z$
19:         **else if** $C == r$ **then**
20:             $l \leftarrow r, r \leftarrow z, i \leftarrow 0$
21:         **else if** $|C| > 1$ **then**
22:             **if** $l\_r\_order == True$ and $r, l \in C$ **then**
23:                 break
24:             $r, l \leftarrow$ LEAFCLUSTERRESOLUTION($C, Parents, ActiveSet$)
25:             break
    return r, l

G.1.4 LEAFCLUSTERRESOLUTION

When we have more than one nodes from the same leaf cluster, we find the parent node of that leaf cluster. If one of the nodes has been detected as a parent node in an earlier iteration, it is selected as the parent node. Otherwise, we perform the following operation on every subset of two nodes X_{i_1}, X_{i_2} in C:

1. Consider a third node $X_{i_3} \in X_{i_1} \cap X_{i_2}$.

2. Check if X_{i_3} also belongs to the same leaf cluster as X_{i_1} and X_{i_2}.

   (a) If X_{i_3} is not in the same leaf cluster, record the value $Q^2(x)$ in Equation (7), for two cases - (i) if X_{i_1} is the center node, (ii) if X_{i_2} is the center node.

   (b) If X_{i_3} is in the same leaf cluster, record the value $Q^2(x)$ in Equation (7), for three cases - (i) if X_{i_1} is the center node, (ii) if X_{i_2} is the center node, (iii) if X_{i_3} is the center node.

Select the center node with the lowest value of the residual $Q^2(x)$ as the parent node. Note that in order to check if 3 nodes are in the same leaf cluster, we attempt to find the center node using the star/non-star subroutine. If we cannot eliminate the possibility of any node being a center node, all the nodes are in the same leaf cluster.

Algorithm 5 Find the parent node in a leaf cluster

Input - Nodes of the leaf cluster, parents.
Output - A parent leaf pair from the leaf cluster.

1: **procedure** LEAFCLUSTERRESOLUTION($P_{i',j'}$, C, $Parents$)
2:     **if** $|C \cap Parents| > 0$ **then**
3:         $l \leftarrow C \cap Parents[0]$ **return** $C \setminus \{l\}[0]$, l
4:     $MinError \leftarrow \infty$
5:     **for** $(X_{i_1}, X_{i_2}) \in C$ **do**
6:         **for** $X_{i_3} \in N(X_{i_1}) \cap N(X_{i_2})$ **do**
7:             **if** $X_{i_3} \in$ FINDCENTER($P_{i',j'}$, $(X_{i_1}, X_{i_2}, X_{i_3})$) and $d_{X'_{i_3}, X'_{i_1}}, d_{X'_{i_3}, X'_{i_2}} \leq d_{max} + 2\eta_{max}$ **then**
8:                 $CandidateParent \leftarrow (X_{i_1}, X_{i_2}, X_{i_3})$
9:             **else** $CandidateParent \leftarrow (X_{i_1}, X_{i_2})$
10:         **for** $X_i \in CandidateParent$ **do**
11:             $err \leftarrow$ QUADRATICERROR($(X_{i_1}, X_{i_2}, X_{i_3}), X_i$)
12:             **if** $err < MinError$ **then**
13:                 $MinError \leftarrow err, l \leftarrow X_i$
14:     $r \leftarrow C \setminus \{l\}[0]$
15: **return** r, l

G.1.5   Runtime Analysis

Following are the runtime for constant k:

1. QUADRATICERROR: $\mathcal{O}(1)$.

2. FINDCENTER: $\mathcal{O}(n)$ as in the worst case, the intersection of the neighborhood can contain $\mathcal{O}(n)$ nodes. The star/non-star test is $\mathcal{O}(1)$.

3. LEAFCLUSTERRESOLUTION: The for loop on line 6 can execute n times in the worst case calling FINDCENTER in each iteration. Thus the total time complexity is $\mathcal{O}(n^2)$.

4. FINDLEAFPARENT: In the worst case LEAFCLUSTERRESOLUTION is called $\mathcal{O}(n)$ times thereby making the sample complexity $\mathcal{O}(n^3)$.

5. FINDTREE: This calls FINDLEAFPARENT $\mathcal{O}(n)$ times. Thus the sample complexity of the algorithm is $\mathcal{O}(n^4)$.

Note that when we know apriori that all the nodes within leaf clusters are unidentifiable, we only use the LEAFCLUSTERRESOLUTION subroutine to check if the parent node was already selected in the previous iteration (lines 1-5). We do not use the QUADRATICERROR subroutine, thereby making it LEAFCLUSTERRESOLUTION an $\mathcal{O}(1)$ operation. In that case, FINDLEAFPARENT is now dominated by FINDCENTER and becomes an $\mathcal{O}(n^2)$ making FINDTREE an $\mathcal{O}(n^3)$ operation (a gain of $\mathcal{O}(n)$)

G.1.6   Recovering $\mathcal{T}_{T^*}^{sub}$

Once we recover a tree from $\mathcal{T}_{T^*}^{sub}$, we can obtain the complete set $\mathcal{T}_{T^*}^{sub}$ by considering all the parent leaf pairs within every cluster along with an arbitrary third node. We call the function QUADRATICERROR with this triplet and only $TestCenter$ node with $err < t_0/2$ is a candidate parent node. This operation does not increase the time complexity as it is an $\mathcal{O}(n^3)$ operation in the worst case.

G.1.7   Modifications for the unidentifiable setting

If we know apriori that the nodes within a leaf cluster are unidentifiable, we do not hope to achieve anything from the QUADRATICERROR subroutine. Therefore, we do not execute any for loops in the LEAFCLUSTERRESOLUTION subroutine, thereby making it an $\mathcal{O}(1)$ operation. Therefore, the GETLEAFPARENT subroutine becomes an $\mathcal{O}(n^2)$ operation making FINDTREE an $\mathcal{O}(n^3)$ operation.
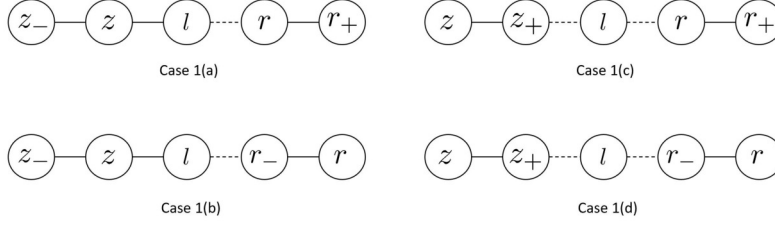
Figure 9: All the possible when node z lies to the left of node l

G.2 Proof of correctness

G.2.1 Proof of correctness of FINDLEAFPARENT subroutine

We first prove that while no two nodes among (l, r, z) are in the same leaf cluster, the subroutine FINDCENTER returns C such that $|C| \leq 1$. For the next part, we assume that no two nodes among (l, r, z) are in the same leaf cluster.

Notation: For any node, the adjacent node on its left is denoted with subscript $-$ and the adjacent node on the right is denoted by subscript $+$. l^{t+1}, r^{t+1} and z^{t+1} are the selection of nodes l, r and z in the next iteration respectively.

We have already proved the correctness of the star/non-star routine in the proof of Lemma 1. Recall from the functionality of FINDCENTER that when we consider nodes (l, r, z) with another node j, if (l, r, z, j) forms a non-star, we eliminate the node that pairs with node j from the candidate center nodes.

With this in mind, we enumerate all the possible configurations of nodes (l, r, z) such that no two of these nodes are in the same leaf cluster. For each case, we present two nodes which, when considered with (l, r, z) would eliminate different nodes from (l, r, z). This is equivalent to proving that $|C| \leq 1$.

Claim: $d_{r,l}, d_{r,z} \leq d_{max} + \eta_{max}$
We first show that this holds true in the initialization of l, r, z. When r is an internal node, we have that:

$$d_{r,l} \leq d_{r,l'} \leq d_{r,r'_-} \leq d_{max} + \eta_{max}, d_{r,z} \leq d_{r,z'} \leq d_{r,r'_+} \leq d_{max} + \eta_{max}.$$

When r is a leaf node, since l, z are not in the same leaf cluster as r, $l \neq z \neq r_-$. Therefore, we have that:

$$d_{r,l} \leq d_{r,l'} \leq d_{r,r'_-} \leq d_{max} + \eta_{max}, d_{r,z} \leq d_{r,z'} \leq d_{r,r_-} \leq d_{max} + \eta_{max}.$$

Now, we assume that $d_{r,l'}, d_{r,z'} \leq d_{max} + \eta_{max}$ is true at the beginning of any iteration and prove that it will continue to hold true at the end of every iteration.

Case 1: We first enumerate all the cases when node z lies to the left of node l. These are presented in Figure 9.

Case 1(a): Node z lies to the left of node l and is adjacent to it and r_+ exists.
In the case there exists a node z_- to the left of z such that there is an edge between z and z_-. (If such a node did not exist, node l and z would have been in the same leaf cluster.)

$$
\begin{aligned}
d_{r',z'_-} =& d_{r,r'} + d_{r,z} + d_{z,z'_-} \\
\leq & \eta_{max} + (d_{max} + \eta_{max}) + (d_{max} + \eta_{max}) \\
=& 2d_{max} + 3\eta_{max} \\
d_{l',z'_-} =& d_{l,l'} + d_{l,z} + d_{z,z'_-} \\
\leq & \eta_{max} + (d_{max} + \eta_{max}) + (d_{max} + \eta_{max}) \\
=& 2d_{max} + 3\eta_{max} \\
d_{z',r'_+} =& d_{z',z} + d_{z,r} + d_{r,r'_+} \\
\leq & 2d_{max} + 3\eta_{max} \\
d_{l',r'_+} =& d_{l,l'} + d_{l,r} + d_{r,r'_+} \\
\leq & 2d_{max} + 3\eta_{max}
\end{aligned}
$$
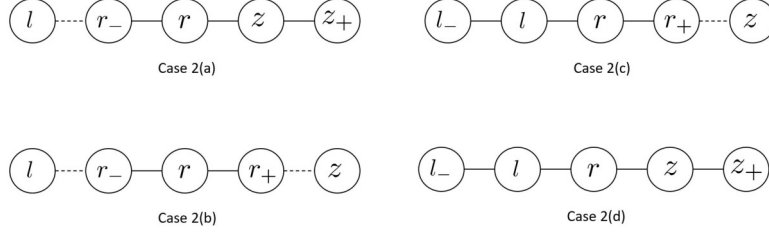
Figure 10: All the possible when node z lies to the right of node r

Thus $z_-, r_+ \in N(r) \cap N(l) \cap N(z)$. z_- eliminates z and r_+ eliminates r. In this case, nodes l and r do not change in this iteration. Therefore, $d_{l^{t+1}, r^{t+1}} = d_{l,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1}, r^{t+1}} \leq d_{r'_+, r} \leq d_{max} + \eta_{max}$.

Case 1(b): Node z lies to the left of node l and is adjacent to it and r_+ does not exists.
When r_+ does not exist, it is easy to see that $\exists r_- \neq l, z$. The first 2 inequalities continue to hold true. We also have:

$$d_{z', r'_-} = d_{z', z} + d_{z, r_-} + d_{r_-, r'_-}$$
$$\leq d_{max} + 3\eta_{max}$$
$$d_{l', r'_-} \leq d_{max} + 3\eta_{max}$$

Thus $r_-, z_- \in N(r) \cap N(z) \cap N(l)$. r_- eliminates r and z_- eliminates z. In this case, nodes l and r do not change in this iteration. Therefore, $d_{l^{t+1}, r^{t+1}} = d_{l,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1}, r^{t+1}} \leq d_{r'_-, r} \leq d_{max} + \eta_{max}$.

Case 1(c): Node z lies to the left of node l and there exists a node between l and z. Also, r_+ exists.
We consider the nodes z_+ and r_+.

$$d_{r', z'_+} = dr', r + d_{r, z_+} + d_{z_+, z'_+} \leq d_{max} + 3\eta_{max},$$
$$d_{l', z'_+} = dl', l + d_{l, z_+} + d_{z_+, z'_+} \leq d_{max} + 3\eta_{max}.$$

For d_{z', r'_+} and d_{l', r'_+}, Case 1(a) calculations are valid.
Thus $r_+, z_+ \in N(r) \cap N(z) \cap N(l)$. r_+ eliminates r and z_+ eliminates z. In this case, nodes l and r do not change in this iteration. Therefore, $d_{l^{t+1}, r^{t+1}} = d_{l,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1}, r^{t+1}} \leq d_{r'_+, r} \leq d_{max} + \eta_{max}$.

Case 1(d): Node z lies to the left of node l and there exists a node between l and z. r_+ does not exist.
In this case, we have $z'_+, r'_- \in N(r) \cap N(l) \cap N(z)$. The derivation comes from Case 1(b) and 1(c). r_- eliminates r and z_+ eliminates z. In this case, nodes l and r do not change in this iteration. Therefore, $d_{l^{t+1}, r^{t+1}} = d_{l,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1}, r^{t+1}} \leq d_{r'_-, r} \leq d_{max} + \eta_{max}$.

**Case 2:** We next enumerate all the cases when node z lies to the right of node r. These are presented in Figure 10.

Case 2(a): z lies to the right of r and there exists at least one node between l and r and but no node between r and z.

$$d_{l', z'_+} = d_{l', l} + d_{l, r} + d_{r, z'_+} \leq 3d_{max} + 3\eta_{max},$$
$$d_{r', z'_+} \leq 2d_{max} + 2\eta_{max},$$
$$d_{r'_-, z'} \leq 2d_{max} + 2\eta_{max},$$
$$d_{l', r'_-} = d_{l', l} + d_{l, r_-} + dr_-, r'_- \leq d_{max} + 3\eta_{max}.$$

Thus $r_-, z_+ \in N(r) \cap N(z) \cap N(l)$. r_- eliminates l and z_+ eliminates z. In this case, $l^{t+1} = r, r^{t+1} = z$ Therefore, $d_{l^{t+1}, r^{t+1}} = d_{z,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1}, r^{t+1}} \leq d_{z'_+, z} \leq d_{max} + \eta_{max}$.

Case 2(b): z lies to the right of r and there exists at least one node between l and r and also between r and z.
Nodes of interest - r_+, r_-. d_{l', r'_-} is the same as case 2(a).

$$d_{l', r'_+} = d_{l', r} + d_{r, r'_+} \leq 2(d_{max} + \eta_{max})$$
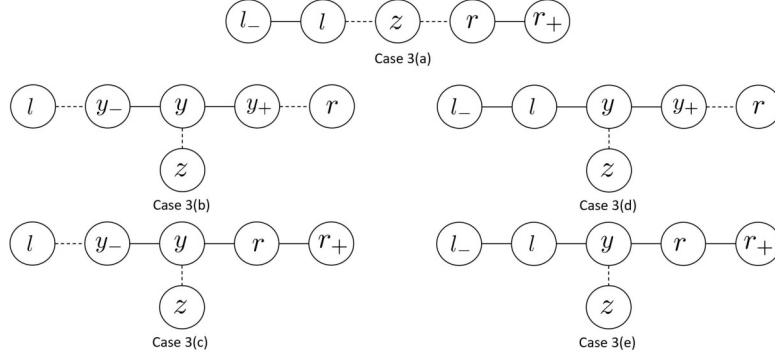
Figure 11: All the possible when node z does not lie to the left of l or right of r

Similarly, $d_{z',r'_-} \leq 2(d_{max} + \eta_{max})$, $d_{z',r'_+} \leq d_{max} + 3\eta_{max}$. Thus $r_-, r_+ \in N(r) \cap N(z) \cap N(l)$. r_- eliminates l and r_+ eliminates z. In this case, $l^{t+1} = r, r^{t+1} = z$ Therefore, $d_{l^{t+1},r^{t+1}} = d_{z,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1},r^{t+1}} \leq d_{z'_-,z} \leq d_{max} + \eta_{max}$.

Case 2(c): z lies to the right of r and there exists at least one node between r and z but no node between r and l.
This is symmetric to Case 2(a). Thus $l_-, r_+ \in N(r) \cap N(z) \cap N(l)$. l_- eliminates l and r_+ eliminates z. In this case, $l^{t+1} = r, r^{t+1} = z$ Therefore, $d_{l^{t+1},r^{t+1}} = d_{z,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1},r^{t+1}} \leq d_{z'_-,z} \leq d_{max} + \eta_{max}$.

Case 2(d): z lies to the right of r and no nodes exist between r and z or r and l.
Since all the nodes are within a radius of 3, it is easy to see that $l_-, r_+ \in N(r) \cap N(z) \cap N(l)$. l_- eliminates l and r_+ eliminates z. In this case, $l^{t+1} = r, r^{t+1} = z$ Therefore, $d_{l^{t+1},r^{t+1}} = d_{z,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1},r^{t+1}} \leq d_{z'_+,z} \leq d_{max} + \eta_{max}$.

Case 3(a): z lies between l and r. Consider l_- and r_+.

$$d_{l'_-,r'} = d_{l'_-,l} + d_{l,r} + d_{r,r'} \leq 2d_{max} + 3\eta_{max}$$
$$d_{l'_-,z'} = d_{l'_-,l} + d_{l,z} + d_{z,z'} \leq 2d_{max} + 3\eta_{max}$$
$$d_{l',r'_+} = d_{l',l} + d_{l,r} + d_{r,r'_+}s \leq 2d_{max} + 3\eta_{max}$$
$$d_{z',r'_+} = d_{z',z} + d_{z,r} + d_{r,r'_+} \leq 2d_{max} + 3\eta_{max}$$

Thus $l_-, r_+ \in N(r) \cap N(z) \cap N(l)$. l_- eliminates l and r_+ eliminates r. In this case, $l^{t+1} = z, r^{t+1} = r$ Therefore, $d_{l^{t+1},r^{t+1}} = d_{z,r} \leq d_{max} + \eta_{max}$. Also, $d_{z^{t+1},r^{t+1}} \leq d_{r'_+,r} \leq d_{max} + \eta_{max}$.

If l_- does not exist, we use l_+. Similarly, if r_+ does not exist, we use r_-.

Case 3(b): Nodes l, r, z form a Y-shape, that is, there exists a node y such that $l \perp r \perp z | y$. There exists at least one node between l and y as well as between y and r.
Consider nodes y_-, y_+.

$$d_{y'_-,z'} = d_{z,z'} + d_{y,z} + d_{y,y'_-} \leq 2d_{max} + 3\eta_{max}$$
$$d_{y'_-,l'} = d_{l',l} + d_{l,y_-} + d_{y_-,y'_-} \leq d_{max} + 3\eta_{max}$$
$$d_{y'_-,r'} = d_{r',r} + d_{r,y_-} + d_{y_-,y'_-} \leq d_{max} + 3\eta_{max}$$
$$d_{y'_+,z'} = d_{z,z'} + d_{y,z} + d_{y,y'_+} \leq 2d_{max} + 3\eta_{max}$$
$$d_{y'_+,l'} = d_{l',l} + d_{l,y_+} + d_{y_+,y'_+} \leq d_{max} + 3\eta_{max}$$
$$d_{y'_+,r'} = d_{r',r} + d_{r,y_+} + d_{y_+,y'_+} \leq d_{max} + 3\eta_{max}$$

Thus $y_-, y_+ \in N(r) \cap N(z) \cap N(l)$. y_- eliminates l and y_+ eliminates r. If z is also eliminated, $l^{t+1} = l, r^{t+1} = r$ and $d_{z^{t+1},r^{t+1}} \leq d_{r'_-,r} \leq d_{max} + \eta_{max}$. If z is not eliminated, $l^{t+1} = z, r^{t+1} = r$, $d_{l^{t+1},r^{t+1}} = d_{r,z} \leq d_{max} + \eta_{max}$ $d_{z^{t+1},r^{t+1}} \leq d_{r'_-,r} \leq d_{max} + \eta_{max}$.

Case 3(c): Nodes l, r, z form a Y-shape, that is, there exists a node y such that $l \perp r \perp z | y$. There exists at least one node between l and y but no node between y and r.
Consider nodes y_-, r_+. Analysis for y_- is the same as in case 3(b).

$$d_{r'_+, z'} = d_{z, z'} + d_{r, z} + d_{r, r'_+} \leq 2d_{max} + 3\eta_{max}$$

$$d_{r'_+, l'} = d_{l', l} + d_{l, r} + d_{r, r'_+} \leq 2d_{max} + 3\eta_{max}$$

Thus $y_-, r_+ \in N(r) \cap N(z) \cap N(l)$. y_- eliminates l and r_+ eliminates r. If z is also eliminated, $l^{t+1} = l, r^{t+1} = r$ and $d_{z^{t+1}, r^{t+1}} \leq d_{r'_+, r} \leq d_{max} + \eta_{max}$. If z is not eliminated, $l^{t+1} = z, r^{t+1} = r$, $d_{l^{t+1}, r^{t+1}} = d_{r, z} \leq d_{max} + \eta_{max}$ $d_{z^{t+1}, r^{t+1}} \leq d_{r'_+, r} \leq d_{max} + \eta_{max}$.

Case 3(d): Nodes l, r, z form a Y-shape, that is, there exists a node y such that $l \perp r \perp z | y$. There exists at least one node between r and y but no node between y and l.
Consider nodes l_-, y_+. Analysis for y_+ is the same as Case 3(b).

$$d_{l'_-, z'} = d_{z, z'} + d_{z, y} + d_{y, l'_-} \leq d_{r, z'} + d_{y, l'_-} \leq 3d_{max} + 3\eta_{max}$$

$$d_{l'_-, r'} = d_{r', r} + d_{r, l} + d_{l, l'_-} \leq 2d_{max} + 3\eta_{max}$$

Thus $y_+, l_- \in N(r) \cap N(z) \cap N(l)$. y_+ eliminates r and l_- eliminates l. If z is also eliminated, $l^{t+1} = l, r^{t+1} = r$ and $d_{z^{t+1}, r^{t+1}} \leq d_{r', r} \leq d_{max} + \eta_{max}$. If z is not eliminated, $l^{t+1} = z, r^{t+1} = r$, $d_{l^{t+1}, r^{t+1}} = d_{r, z} \leq d_{max} + \eta_{max}$ $d_{z^{t+1}, r^{t+1}} \leq d_{r'_-, r} \leq d_{max} + \eta_{max}$.

Case 3(e): Nodes l, r, z form a Y-shape, that is, there exists a node y such that $l \perp r \perp z | y$. There exists no nodes between r and y and between y and l.
Consider nodes l_-, r_+. Analysis follows from Cases 3(c) and 3(d). Thus $r_+, l_- \in N(r) \cap N(z) \cap N(l)$. r_+ eliminates r and l_- eliminates l. If z is also eliminated, $l^{t+1} = l, r^{t+1} = r$ and $d_{z^{t+1}, r^{t+1}} \leq d_{r'_+, r} \leq d_{max} + \eta_{max}$. If z is not eliminated, $l^{t+1} = z, r^{t+1} = r$, $d_{l^{t+1}, r^{t+1}} = d_{r, z} \leq d_{max} + \eta_{max}$ $d_{z^{t+1}, r^{t+1}} \leq d_{r'_+, r} \leq d_{max} + \eta_{max}$.

Thus at each iteration, we visit one node and remove it from the set of nodes that get visited in subsequent iterations until we get (l, r, z) such that at least 2 of the nodes are in the same leaf cluster. Note that the maximum distance in the above analysis is $3d_{max} + 3\eta_{max}$. However our threshold for the neighborhood set is $4d_{max} + 3\eta_{max}$. The extra d_{max} is there to account for the fact that in the unidentifiable case, a parent node from a leaf cluster may have been confused with a leaf node. In that case, the leaf node is retained in the active set while the parent node is removed from the active set for the subsequent iterations. In order to account for that, we add a factor of d_{max} to the neighborhood threshold.

Proof of correctness of LeafClusterResolution   From the above analysis, we know that LeafClusterResolution is called with nodes in C belonging in the same leaf cluster. The idea is to check if any on the nodes in C are such that when they act as the center node, Equation (7) has a solution. In order to do this, we consider 2 nodes in C at a time and scan through all the nodes in their common neighborhood as the third node. We check if the third node is also in the same leaf cluster in which case we also see if the error for this node as the parent node is small. If it is not in the same leaf cluster, we just use it as the third node needed for Equation (7). We first show that the routine to check if X_{i_3} is in the same leaf cluster as (X_{i_1}, X_{i_2}) is correct:
If X_{i_3} is in the same leaf cluster as (X_{i_1}, X_{i_2}), it is easy to see that any star/non-star test on $(X_{i_1}, X_{i_2}, X_{i_3}, j)$ always returns a non-star. When X_{i_3} is not in the same leaf cluster as (X_{i_1}, X_{i_2}), then there exists a node $X_{i_3^+}$ adjacent to X_{i_3} either away from the path connecting X_{i_3} to (X_{i_1}, X_{i_2}) or on that path such that $(X_{i_1}, X_{i_2}, X_{i_3}, X_{i_3^+})$ forms a non-star where $(X_{i_3}, X_{i_3^+})$ forms a pair. It is easy to see that $d_{X'_1, (X_{i_3^+})'}, d_{X'_2, (X_{i_3^+})'} \leq 2d_{max} + 3\eta_{max}$.
Therefore, $X_{i_3^+} \in N(X_{i_1}) \cap N(X_{i_2}) \cap N(X_{i_3})$. Thus it is ruled out from being a parent candidate.
Now it is easy to see that if any leaf node is identifiable, it will have a non-zero error for Equation (7). For an unidentifiable leaf node, both the leaf and parent have a solution to Equation (7) and one of them is randomly selected as the parent node.

Any subsequent calls with nodes from the same leaf cluster always select the correct parent in line (2).

From the correctness of LeafClusterResolution, we conclude that FindLeafParent subroutine is correct. Once we have the correctness of GetLeafParent, the correctness of FindTree is easy to understand. We prove this by induction on the number of nodes.

Base Case (n=2): Line 9 recovers the lone edge.

Inductive Case: Let us assume that the algorithm works for all $n < k$. For $n = k + 1$, by the correctness of GETLEAFPARENT, the algorithm correctly recovers one leaf parent pair and adds that edge to the edge set. Once the leaf node is removed, the algorithm is effectively running on k nodes and by the inductive assumption that is correct.

This completes the proof of correctness of the algorithm.

G.3 Modification for finite sample domain

In this section we present the necessary modifications needed to execute the algorithm using finite samples.
Classifying 4 nodes as star/non-star using finite samples: Let us denote $\kappa_{i',j'} = \exp{-d_{i',j'}}$, $\kappa_{max} = \exp(-d_{min})$. We denote the finite sample estimate of $\kappa_{i',j'}$ by $\hat{\kappa_{i',j'}}$

In the infinite sample setting, a set of 4 nodes (X_1, X_2, X_3, X_4) forms a non-star with (X_1, X_2) forming a pair if:

$$\frac{\sqrt{\kappa_{1',3'}\kappa_{2',4'}\kappa_{1',4'}\kappa_{2',3'}}}{\kappa_{1',2'}\kappa_{3',4'}} \leq \kappa_{max}^2$$

$$\frac{\sqrt{\kappa_{1',2'}\kappa_{3',4'}\kappa_{1',4'}\kappa_{2',3'}}}{\kappa_{1',3'}\kappa_{2',4'}} \geq 1/\kappa_{max}^2$$

$$\frac{\sqrt{\kappa_{1',3'}\kappa_{4',2'}\kappa_{1',2'}\kappa_{4',3'}}}{\kappa_{1',4'}\kappa_{2',3'}} \geq 1/\kappa_{max}^2$$

The finite sample test is as follows:

$$\frac{\sqrt{\hat{\kappa}_{1',3'}\hat{\kappa}_{2',4'}\hat{\kappa}_{1',4'}\hat{\kappa}_{2',3'}}}{\hat{\kappa}_{1',2'}\hat{\kappa}_{3',4'}} \leq (1 + \kappa_{max}^2)/2$$

$$\frac{\sqrt{\hat{\kappa}_{1',2'}\hat{\kappa}_{3',4'}\hat{\kappa}_{1',4'}\hat{\kappa}_{2',3'}}}{\hat{\kappa}_{1',3'}\hat{\kappa}_{2',4'}} \geq 1$$

$$\frac{\sqrt{\hat{\kappa}_{1',3'}\hat{\kappa}_{4',2'}\hat{\kappa}_{1',2'}\hat{\kappa}_{4',3'}}}{\hat{\kappa}_{1',4'}\hat{\kappa}_{2',3'}} \geq 1$$

A set of 4 nodes (X_1, X_2, X_3, X_4) is classified as a star if:

$$\frac{\sqrt{\hat{\kappa}_{1',3'}\hat{\kappa}_{2',4'}\hat{\kappa}_{1',4'}\hat{\kappa}_{2',3'}}}{\hat{\kappa}_{1',2'}\hat{\kappa}_{3',4'}} \geq (1 + \kappa_{max}^2)/2$$

$$\frac{\sqrt{\hat{\kappa}_{1',2'}\hat{\kappa}_{3',4'}\hat{\kappa}_{1',4'}\hat{\kappa}_{2',3'}}}{\hat{\kappa}_{1',3'}\hat{\kappa}_{2',4'}} \geq (1 + \kappa_{max}^2)/2$$

$$\frac{\sqrt{\hat{\kappa}_{1',3'}\hat{\kappa}_{4',2'}\hat{\kappa}_{1',2'}\hat{\kappa}_{4',3'}}}{\hat{\kappa}_{1',4'}\hat{\kappa}_{2',3'}} \geq (1 + \kappa_{max}^2)/2$$

If neither of the above conditions is satisfied for any pair, the test fails and this set of 4 nodes is not classified as star/non-star.

Neighborhood Thresholding: In the finite sample setting, we allow for a slack in the threshold to ensure that, with high probability, the empirical neighborhood vector contains all the nodes from the underlying neighborhood vector. The empirical neighborhood vector is defined as follows:

$$N'(X_i) = sort(X_j : \hat{d}_{i',j'} \leq t_{emp}, \text{ key } = \hat{d}_{i',j'}),$$

where the threshold is $t_{emp} = (4d_{max} + 3\eta_{max}) + ln(0.5)$.

H Sample Complexity Upper Bound

Let us define 2 events:

$$\mathcal{B}_1 = \{(E_{a'})_{i,i} < (1 - 0.5^{1/k})p_{min}, \forall a, i\}, \mathcal{B}_2 = \{\|E_{a',b'}\| < \epsilon \forall a, b\}$$

For any X_a, X_b we only consider nodes such that:

$$\sqrt{|det(\hat{P}_{a'|b'}\hat{P}_{b'|a'})|} > 0.5 \exp(-4d_{\max})(1 - q_{max})^{3(k-1)}(kp_{min})^{1.5k}$$

$$\implies \frac{|det(\hat{P}_{a',b'})|}{\sqrt{|det(\hat{P}_{a'}\hat{P}_{b'})|}} > 0.5 \exp(-4d_{\max})(1 - q_{max})^{3(k-1)}(kp_{min})^{1.5k}.$$

In the event $\mathcal{B}_1$, $det(\hat{P}_{a'}), det(\hat{P}_{b'}) > 0.5p_{min}^k$, therefore we have:

$$|det(\hat{P}_{a',b'})| \geq 0.25 \exp(-4d_{\max})(1 - q_{max})^{3(k-1)}(kp_{min})^{1.5k}p_{min}^k$$

Next we bound the minimum absolute eigenvalue of $\hat{P}_{a',b'}$.

Lemma 4. *For any $k \times k$ matrix M such that $M_{i,j} \geq 0$, $\sum_{i,j} M_{i,j} = 1$ and $|det(M)| \geq c$ where $0 < c \leq \left(\frac{1}{k}\right)^k$, then the minimum absolute eigenvalue of M satisfies $c(k-1)^{k-1} \leq |\lambda_{min}(M)| \leq ck^{k-1}$.*

Proof. Let $\lambda_1, \lambda_2 \ldots \lambda_k$ be the eigenvalues of M such that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k|$. Standard results tell us that:

$$\sum_i |\lambda_i| \leq \sum_{i,j} M_{i,j} = 1, |det(M)| = \prod_i |\lambda_i| \geq c$$

We are interested in the solution to the following optimization problem:

$$\min \qquad |\lambda_k| \tag{30}$$

$$\text{s.t.} \qquad \sum_{i=1}^{k} |\lambda_i| \leq 1 \tag{31}$$

$$\prod_{i=1}^{k} |\lambda_i| \geq c, \tag{32}$$

$$|\lambda_1| \geq |\lambda_2| \ldots |\lambda_k|, \tag{33}$$

where $0 < c \leq (1/k)^k$. Denote the optimal solution to the above problem by $\lambda_1^*, \lambda_2^*, \ldots \lambda_k^*$.

Claim: $\sum_i |\lambda_i^*| = 1, \prod_{i=1}^{k} |\lambda_i^*| = c, |\lambda_1^*| = |\lambda_2^*| = \cdots = |\lambda_{k-1}^*|$.

In order to prove this, we prove that if these do not hold true, there exists a smaller $|\lambda_k|$.

By contradiction, let us assume that $\sum_i |\lambda_i^*| = 1 - \epsilon$ for some $0 < \epsilon < 1$. Then it is easy to see that $\exists \tilde{\lambda}_i, \epsilon' > 0$ such that $|\tilde{\lambda}_i| = |\lambda_i^*| + \frac{\epsilon}{k-1} \ \forall i \in \{1, 2 \ldots, k-1\}$ and $|\tilde{\lambda}_k| = |\lambda_k^*| - \epsilon'$ such that $\prod_{i=1}^{k} |\tilde{\lambda}_i| = c$. Therefore, $|\lambda_i^*|$ is not optimal. Thus, $\sum_i |\lambda_i^*| = 1$.

By contradiction, let us assume that $\prod_i |\lambda_i^*| = (1 + \epsilon)c$ for some $0 < \epsilon$. Consider $\tilde{\lambda}_i$ such that $\tilde{\lambda}_i = \lambda_i^*$ $\forall i \in \{1, 2 \ldots, k-1\}$ and $\tilde{\lambda}_k = \lambda_k^*/(1 + \epsilon)$. Then $\tilde{\lambda}_i$ is feasible and has smaller objective value, thus $\prod_i |\lambda_i^*| = c$.

We prove the last part by contradiction too. Let us assume by contradiction that at least one of $|\lambda_i^*|$ is not equal for $i \in \{1, 2, \ldots k-1\}$. Consider $\tilde{\lambda}_i$ such that $|\tilde{\lambda}_i| = \frac{\sum_{j=1}^{k-1} |\lambda_j^*|}{k-1}$. Then, by the AM-GM inequality, we have that:

$$\prod_{i=1}^{k-1} |\tilde{\lambda}_i| = \left(\frac{\sum_{j=1}^{k-1} |\lambda_j^*|}{k-1}\right)^{k-1} = (1 + \epsilon) \prod_{i=1}^{k-1} |\lambda_i^*|$$

for some $\epsilon > 0$. Choosing $|\tilde{\lambda}_k| = |\lambda_k^*|/(1+\epsilon)$, we get a feasible $\tilde{\lambda}_i$ with a smaller objective function. This concludes the proof of the claim.

Thus, the solution to the optimization problem 30 satisfies:

$$|\lambda_1^*| = |\lambda_2^*| = \cdots = |\lambda_{k-1}^*| = \frac{1 - \lambda_k^*}{k - 1}, \left(\frac{1 - \lambda_k^*}{k - 1}\right)^{k-1} \lambda_k^* = c.$$

Therefore, Equation 30 has the same solution as the following optimization problem:

$$\text{min} \qquad |\lambda_k|$$

$$\text{s.t.} \qquad 0 < |\lambda_k| \leq \frac{1}{k}$$

$$|\lambda_k| \left(\frac{1 - |\lambda_k|}{k-1} \right)^{k-1} = c,$$

where $0 < c \leq (1/k)^k$. The solution to the above optimization problem satisfies $|\lambda_k^*| \left(\frac{1-|\lambda_k^*|}{k-1} \right)^{k-1} = c$. The solution exists because $|\lambda_k| \left(\frac{1-|\lambda_k|}{k-1} \right)^{k-1}$ is monotonically increasing in $|\lambda_k|$ and:

$$|\lambda_k| \left(\frac{1 - |\lambda_k|}{k-1} \right)^{k-1} = 0, \text{ when } |\lambda_k| = 0,$$

$$|\lambda_k| \left(\frac{1 - |\lambda_k|}{k-1} \right)^{k-1} = \left(\frac{1}{k} \right)^k, \text{ when } |\lambda_k| = 1/k.$$

Therefore, $|\lambda_k^*|$ satisfies:

$$|\lambda_k^*| = c \left(\frac{(k-1)}{1 - |\lambda_k^*|} \right)^{k-1}$$

Since $0 < |\lambda_k^*| \leq 1/k$, we have that $c(k-1)^{k-1} \leq |\lambda_k^*| \leq ck^{k-1}$ \hfill $\square$

Using Lemma 4, the minimum absolute eigenvalue of $\hat{P}_{a',b'}$ is lower bounded by $|det(\hat{P}_{a',b'})|(k-1)^{k-1}$. Therefore, we have that:

$$\|\hat{P}_{a',b'}^{-1}\| \leq \frac{1}{0.25 \exp(-4d_{\max})(1 - q_{max})^{3(k-1)}(kp_{min})^{1.5k}p_{min}^k(k-1)^{k-1}} \triangleq \frac{1}{z_1} \tag{34}$$

H.1 Sample Complexity for Existence of a solution to Equation 7

We are interested in the error in the estimate of $Q(x)$ as defined below:

$$\hat{Q}(x) = \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(O\hat{P}_b' + \hat{P}_b'O - k\hat{P}_b' - I) + \hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{a',b'} - \hat{P}_b'\|_F$$

$$Q(x) = \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_b' + P_b'O - kP_b' - I) + P_{b',c'}P_{a',c'}^{-1}P_{a',b'} - P_b'\|_F$$

We derive the error bound for the term $P_{b',c'}P_{a',c'}^{-1}P_{ab}$ when estimated using the respective empirical estimates.

$$P_{b',c'}P_{a',c'}^{-1}P_{a',b'} = (\hat{P}_{b',c'} + E_{b',c'})(\hat{P}_{a',c'} + E_{a',c'})^{-1}(\hat{P}_{a',b'} + E_{a',b'})$$

$$= (\hat{P}_{b',c'} + E_{b',c'})\left(\hat{P}_{a',c'}^{-1} + \sum_{m=1}^{\infty}(-\hat{P}_{a',c'}^{-1}E_{a',c'})^m \hat{P}_{a',c'}^{-1} \right)(\hat{P}_{a',b'} + E_{a',b'})$$

$$= \hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{ab} + E_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{a',b'} + \hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}E_{a',b'} + \hat{P}_{b',c'}\tilde{E}_{ac}\hat{P}_{a',b'}$$

$$+ E_{b',c'}\hat{P}_{a',c'}^{-1}E_{a',b'} + E_{b',c'}\tilde{E}_{ac}\hat{P}_{a',b'} + \hat{P}_{b',c'}\tilde{E}_{ac}E_{a',b'} + E_{b',c'}\tilde{E}_{ac}E_{a',b'},$$

here we use the notation $\tilde{E}_{ac} := \sum_{m=1}^{\infty}(-\hat{P}_{a',c'}^{-1}E_{a',c'})^m \hat{P}_{a',c'}^{-1}$. Using the triangle inequality and submultiplicative property of the spectral norm, we get that:

$$\|\tilde{E}_{ac}\|_2 \leq \frac{\|\hat{P}_{a',c'}^{-1}\|_2^2 \|E_{a',c'}\|_2}{1 - \|\hat{P}_{a',c'}^{-1}\|_2 \|E_{a',c'}\|_2}$$

We choose such an ϵ in the event $\mathcal{B}_2$ that ensures that $\|P_{a',c'}^{-1}\|_2 \|E_{a',c'}\|_2 < 0.5$. This gives us:

$$\|\tilde{E}_{ac}\|_2 \leq 2\|\hat{P}_{a',c'}^{-1}\|_2^2 \|E_{a',c'}\|_2$$

In the event $\mathcal{B}_2$, $\|E_{a',b'}\|_2$, $\|E_{b',c'}\|_2\|E_{a',c'}\|_2 < \epsilon$. In the event $\mathcal{B}_1$, from Equation (34), $\|\hat{P}_{a',c'}^{-1}\|_2 \leq z_1^{-1}$. Therefore, $\|\tilde{E}_{ac}\|_2 \leq 2z_1^{-2}\epsilon$. Since $\hat{P}_{a',b'}, \hat{P}_{b',c'}$ are joint PMF matrices, we have that $\|\hat{P}_{a',b'}\|_2, \|\hat{P}_{b',c'}\|_2 < 1$. Substituting these along with triangle inequality and submultiplicative property of the spectral norm gives us the following:

$$\|\hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{a',b'} - P_{b',c'}P_{a',c'}^{-1}P_{a'b'}\|_2 \leq 3\epsilon z_1^{-1} + 8\epsilon z_1^{-2}$$

This gives us:

$$\hat{Q}(x) = \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(O\hat{P}_b' + \hat{P}_b'O - k\hat{P}_b' - I) + \hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{a',b'} - \hat{P}_b'\|_F$$
$$\leq Q(x) + (3x+1)\|E_{b'}\|_F + \|\hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{a',b'} - P_{b',c'}P_{a',c'}^{-1}P_{ab}\|_F$$
$$\implies |\hat{Q}(x) - Q(x)| \leq 4\sqrt{k}\epsilon + 3\sqrt{k}\epsilon z_1^{-1} + 8\sqrt{k}\epsilon z_1^{-2} \leq 15\sqrt{k}\epsilon z_1^{-2}$$

We need that $|\hat{Q}(x) - Q(x)| < t_0/2$. This is satisfied when:

$$\epsilon < \frac{t_0 z_1^2}{30\sqrt{k}} \tag{35}$$

H.2 Sample Complexity for Star/Non-Star test

Consider a set of 4 nodes $\{X_1, X_2, X_3, X_4\}$ such that they form a non-star such that $\{X_1, X_2\}$ form a pair.

$$\frac{|det(P_{1,3}P_{2,4})|}{|det(P_{1,4}P_{2,3})|} = \frac{|det((\hat{P}_{1,3} + E_{1,3})(\hat{P}_{2,4} + E_{2,4}))|}{|det((\hat{P}_{1,4} + E_{1,4})(\hat{P}_{2,3} + E_{2,3}))|} \tag{36}$$

Using the analysis from Tandon et al. [2021], a set of 4 nodes is correctly classified if for any pair of nodes $\{a, b\}$ that are in each other's neighborhood sets, we have that $|det(P_{a,b}) - det(\hat{P}_{a,b})| < \frac{z_1(1-\alpha)}{20}$, where $\alpha = \frac{1+\exp(-2d_{min})}{2}$. We can bound the difference in the empirical estimate of the determinant and the true determinant using the matrix perturbation result in Chapter 5 of Bhatia [2007] as follows:

$$|det(P_{a,b}) - det(\hat{P}_{a,b})| \leq k\max\{\|P_{a,b}\|, \|\hat{P}_{a,b}\|\}^{k-1}\|E_{a,b}\|_2 \leq k\|E_{a,b}\|_2$$

Under event $\mathcal{B}_2$ we have that $\|E_{a,b}\| < \epsilon$. Thus the algorithm correctly classifies nodes as star/non-star when:

$$\epsilon < \frac{z_1(1-\alpha)}{20k}. \tag{37}$$

From Equations (35), (37) we choose ϵ as follows:

$$\epsilon < \min\left\{\frac{z_1(1-\alpha)}{20k}, \frac{t_0 z_1^2}{30\sqrt{k}}\right\}. \tag{38}$$

Next, we find the number of samples needed for $\mathcal{B}_1$ and $\mathcal{B}_2$ to hold true with high probability.

$$P(\mathcal{B}_1, \mathcal{B}_2) \geq 1 - P(\bar{\mathcal{B}}_1) - P(\bar{\mathcal{B}}_2)$$

For a given a, i, by Hoeffding's inequality we have that:

$$P((E_{a'})_{i,i}) > (1 - 0.5^{1/k})p_{min}) \leq \exp(-2N((1 - 0.5^{1/k})p_{min})^2).$$

By the union bound on all the nodes and all the alphabets we get:

$$P(\bar{\mathcal{B}}_1) \leq kn\exp(-2N((1 - 0.5^{1/k})p_{min})^2).$$

In order to achieve $P(\bar{\mathcal{B}}_1) \leq \delta/2$, we have the following bound on the sample complexity:

$$N \geq \frac{1}{2p_{min}^2(1 - 0.5^{1/k})^2}\log\left(\frac{2nk}{\delta}\right). \tag{39}$$

Note that:

$$\frac{1}{(1 - 0.5^{1/k})^2} = \left(\frac{2^{1/k}}{2^{1/k} - 1}\right)^2$$

$$\leq \frac{2}{(2^{1/k} - 1)^2}$$

$$= \frac{2}{\left(\exp\left(\frac{\log 2}{k}\right) - 1\right)^2}$$

$$\leq \frac{2k^2}{(\log 2)^2}.$$

Thus, when

$$N \geq \left(\frac{k}{p_{min} \log 2}\right)^2 \log\left(\frac{2nk}{\delta}\right), \tag{40}$$

we have $P(\bar{\mathcal{B}}_1) \leq \delta/2$. Next, we upper bound the probability $P(\bar{\mathcal{B}}_1)$.

The matrix Bernstein's inequality (Tropp [2015]) states that for independent random matrices $S_1 \ldots S_N$ with dimension $d_1 \times d_2$ such that $\mathbb{E}[S_i] = 0$, $\|S_i\| < L$ $\forall i$ and $Z = \sum_{i=1}^N S_i$, then

$$P(\|Z\| > t) \leq (d_1 + d_2) \exp\left(\frac{-t^2/2}{v(Z) + Lt/3}\right)$$

where $v(Z) = \max\{\|\sum_{i=1}^N \mathbb{E}[S_i S_i^T]\|\}$. In order to apply this in our setting, define $S_i = \mathbb{1}_{a',b'}^i - P_{a',b'}$ where $\mathbb{1}_{a',b'}^i$ is the indicator matrix for sample i with a 1 in the position corresponding to the value of X_a' and X_b' in that sample.

It is easy to see that $\mathbb{E}[S_i] = 0, \|S_i\| \leq 2$. Also, in this setting, $E_{a',b'} = \frac{1}{N}Z$. Next, we bound $v(Z)$.

$$\mathbb{E}[S_i S_i^T] = \mathbb{E}[(\mathbb{1}_{a',b'}^i - P_{a',b'})(\mathbb{1}_{a',b'}^i - P_{a',b'})^T]$$

$$= \mathbb{E}[(\mathbb{1}_{a',b'}^i)(\mathbb{1}_{a',b'}^i)^T] - \mathbb{E}[P_{a',b'} P_{a',b'}^T]$$

$$\implies \|\sum_{i=1}^N \mathbb{E}[S_i S_i^T]\| \leq 2N$$

This bounds the probability of $\|E_{a',b'}\| > \epsilon$ as follows:

$$P(\|E_{a',b'}\| > \epsilon) = P(\|Z\| > n\epsilon) \leq 2k \exp\left(\frac{-N\epsilon^2}{4(1 + \epsilon/3)}\right)$$

By the union bound on all the pair of nodes, we have:

$$P(\bar{\mathcal{B}}_2) \leq kn(n - 1) \exp\left(\frac{-N\epsilon^2}{4(1 + \epsilon/3)}\right).$$

For $P(\bar{\mathcal{B}}_2) \leq \delta/2$, the lower bound on the number of samples is given by

$$N \geq \frac{2(2 + \epsilon/3)}{\epsilon^2} \log\left(\frac{2nk(n - 1)}{\delta}\right) \tag{41}$$

From Equations (40) and (41), the algorithm outputs the correct tree if:

$$N \geq \max\left\{\left(\frac{k}{p_{min} \log 2}\right)^2 \log\left(\frac{2nk}{\delta}\right), \frac{2(2 + \epsilon/3)}{\epsilon^2} \log\left(\frac{2nk(n - 1)}{\delta}\right)\right\}. \tag{42}$$

From Equation (34), we have that:

$$z_1 = \frac{(1 - q_{max})^{3(k-1)}(kp_{min})^{2.5k}}{4k \exp(4d_{max})}\left(1 - \frac{1}{k}\right)^{k-1}.$$

Note that:

$$\frac{1}{e} < \left(1 - \frac{1}{k}\right)^{k-1} < \frac{1}{2} \; \forall \, k \geq 2$$

Thus we have:

$$\epsilon = \mathcal{O}\left(\min\left\{\frac{(1 - q_{max})^{3(k-1)}(kp_{min})^{2.5k}(1 - \exp(-2d_{min}))}{k^2 \exp(4d_{max})}, \frac{t_0(1 - q_{max})^{6(k-1)}(kp_{min})^{5k}}{k^{2.5}\exp(8d_{max})}\right\}\right)$$

From the value of ϵ, we can see that the sample complexity in Equation (41) is dominated by the second term. Substituting the value of ϵ we get that the sample complexity is of the following order:

$$N = \mathcal{O}\left(\max\left\{\frac{k^4 \exp(8d_{\max})}{(1 - q_{max})^{6(k-1)}(kp_{min})^{5k}(1 - \exp(-2d_{min}))^2},\right.\right.$$
$$\left.\left.\frac{k^5 \exp(16d_{\max})}{t_0^2(1 - q_{max})^{12(k-1)}(kp_{min})^{10k}}\right\}\log\left(\frac{2nk(n-1)}{\delta}\right)\right)$$

I   Sample Complexity Lower Bound

I.1   Preliminaries

In this section, we present some definitions, and results that we will use for our lower bound proof.

Information theoretic lower bound: We now present the information theoretic lower bound for required samples in recovering a distribution.

We first define the symmetrized KL-divergence between two distributions P and Q as

$$J(P, Q) = \mathbb{E}_{\mathbf{X} \sim P} \log\left(\frac{P(\mathbf{X})}{Q(\mathbf{X})}\right) + \mathbb{E}_{\mathbf{X} \sim Q} \log\left(\frac{P(\mathbf{X})}{Q(\mathbf{X})}\right).$$

Lemma 5 (Fano's Inequality, Lemma 6.2 in Bresler et al.Bresler et al. [2020]). *For $M \geq 2$, given the $(M+1)$ distributions $\{P_0, \ldots, P_M\}$, for any estimator $\Psi : [k]^n \times N \to \{0, 1, \ldots, M\}$ that uses N i.i.d. samples $\mathbf{X}'(1:N)$, and for any $\delta > 0$ we have for*

$$N \leq (1 - \delta)\frac{\log(M)}{\frac{1}{M+1}\sum_{k=1}^{M} J(P^{(k)}, P^{(0)})}, \quad \inf_{\Psi} \max_{0 \leq k \leq M} P^{(j)}(\Psi(\mathbf{X}'(1:N)) \neq j) \geq \delta - \frac{1}{\log(M)}.$$

The above inequality provides such a characterization in the minimax sense. In particular, it says among the M distributions there exists at least one from which N (as defined in the lemma) i.i.d. samples are required to identify that distribution correctly with probability at least $(1 - \delta + \frac{1}{\log(M)})$.

Partially Mixing Graphical Model (PMGM): We introduce a special class of tree structured graphical models called the *Partially Mixing Graphical Models* (PMGM). In order to define them, we first define the matrices A as follows:

$$A = \begin{bmatrix} 0 & 0 & \ldots & 0 & 0 \\ 0 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \ldots & -1 & 1 \\ 0 & 0 & \ldots & 1 & -1 \end{bmatrix}.$$

Definition 2. *A Partially Mixing Graphical Model (PMGM) is a tree structured graphical model on discrete random variables with common support such that the marginal distribution of all the random variables is a uniform distribution and the conditional PMF matrix of two random variables X_i, X_j such that $(X_i, X_j) \in \mathcal{E}$ is given by:*

$$P_{i|j} = I + \alpha_{i,j}A, \; where \; 0 < \alpha_{i,j} < 1.$$

Intuitively, in PMGMs, if a sample of $X_i \in \{s_1, \ldots s_{k-2}\}$, then all the random variables in that sample are equal. Only for a subset of the support $\{s_{k-1}, s_k\}$, the random variables are unequal in one sample.

It is easy to see that $\det(P_{i|j}) = 1 - 2\alpha_{i,j}$. Thus, for PMGMs, the distance between two adjacent nodes X_i and X_j is $d_{i,j} = -\log(1 - 2\alpha_{i,j})$.

Circulant Matrices: Let $\mathcal{R}$ be a rotational operation of a vector $v \in \mathbb{R}^k$ which maps it to $v' = \mathcal{R}(v) \in \mathbb{R}^k$ with $v'(i) = v((i+1) \bmod k)$ for all $1 \leq i \leq k$. Then we have $v'' = \mathcal{R}^j(v)$ as $v''(i) = v((i+j) \bmod k)$ for any $j \geq 1$, and for all $1 \leq i \leq k$. Then a ciculant matrix created from vector v is given as $Cir(v) = (v; \mathcal{R}(v); \mathcal{R}^2(v); \ldots; \mathcal{R}^{(k-1)}(v))$. For any circulant matrix in $\mathbb{R}^{k \times k}$ with vector v, denoted as $Cir(v)$, the determinant is given as

$$\det(Cir(v)) = \prod_{j=0}^{k-1} \sum_{i=0}^{k-1} v_i \omega^{ji}.$$

The following lemma states that when a graphical model has the conditional PMF as circulant matrix for each edge, then if one node has uniform marginal then all other nodes have uniform marginals as well.

Lemma 6. *Consider a tree graphical model such that the conditional PMF matrix corresponding to every edge is a circulant matrix. Then, if the marginals of one of the nodes is uniformly distributed on the support, the marginals of all the remaining nodes are also uniform.*

Proof. Suppose the node with uniform marginals is X_1. Suppose node X_2 has an edge with X_1 and $P(X_2|X_1)$ is a circulant matrix. Thus we have $P(X_2, X_1) = \frac{P(X_2|X_1)}{k}$. Therefore, $P(X_2, X_1)$ is also a circulant matrix. When the joint PMF matrix is circulant, all the rows and columns the marginal distribution of both the random variables is uniform. Therefore, the marginal distribution of X_2 is also uniform. Thus the marginal distribution of all the nodes connected to X_1 is uniform. Once we know that the marginals of one hop neighbors of X_1 are uniform, we can infer the same about the two hop neighbors of X_1. This can further be extended for all the nodes in the graph. $\qquad\square$

Simplifying the Quadratic Bound: Suppose the marginals of all the random variables are uniform, that is, $P_b' = \frac{1}{k}I$ and the underlying graphical model on X_a, X_b, X_c is a chain with X_a as the center node. We want to bound the following quadratic:

$$Q(x) = \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_b' + P_b'O - kP_b' - I) + P_{b',c'}P_{a',c'}^{-1}P_{a',b'} - P_b'\|_F.$$

The conditional independence relation gives us $P_{b,c} = P_{b,a}P_a^{-1}P_{a_c}$. Recall that $E_a = (1 - q_a)I + \frac{q_a}{k}O$ and similarly we have E_b, E_c. We have the following:

$$P_{b',c'}P_{a',c'}^{-1}P_{a',b'} = E_b P_{b,c} E_c (E_a P_{a,c} E_c)^{-1} E_a P_{a,b} E_b$$
$$= E_b P_{b,a} P_a^{-1} P_{a,c} E_c E_c^{-1} P_{a,c}^{-1} E_a^{-1} E_a P_{a,b} E_b$$
$$= E_b P_{b,a} P_a^{-1} P_{a,b} E_b$$

In the circulant setting, we have that $P_a = \frac{1}{k}I$. This gives us $P_{b',c'}P_{a',c'}^{-1}P_{a',b'} = kE_b P_{b,a} P_{a,b} E_b$. Substituting these in the quadratic, we get:

$$Q(x) = \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_b' + P_b'O - kP_b' - I) + P_{b',c'}P_{a',c'}^{-1}P_{a',b'} - P_b'\|_F, \tag{43}$$

$$= \|\left(\frac{x^2 - 2x + 1}{k^2}\right)O - \left(\frac{x^2 - 2x + 1}{k}\right)I - \frac{O}{k^2} + kE_b P_{b,a} P_{a,b} E_b\|_F, \tag{44}$$

$$= \|\left(\frac{x-1}{k}\right)^2(O - kI) - \frac{O}{k^2} + kE_b P_{b,a} P_{a,b} E_b\|_F \tag{45}$$

Perturbed Symmetric Distribution: We now focus on a special case of circulant matrices which will be used in our lower bound construction later on. The conditional PMF for two nodes a and b in a perturbed symmetric distribution model takes the following form:

$$P_{b|a} = (\alpha - \delta)I + (1 - \alpha)\frac{O}{k} + \Delta$$

$$\Delta = \begin{bmatrix} 0 & \delta & 0 & \dots & 0 \\ 0 & 0 & \delta & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \delta \\ \delta & 0 & 0 & \dots & 0 \end{bmatrix}.$$

Note that this is a class that we define by perturbing the discrete symmetric model slightly.

We first consider the noiseless setting ($E_b = I$). In order to obtain the results for the noisy case, it is sufficient to replace α by $(1-q)\alpha$ and δ by $(1-q)\delta$. For our model, we have that:

$$P_{b,a} = \frac{1}{k}\left((\alpha - \delta)I + (1 - \alpha)\frac{O}{k} + \Delta\right).$$

Noting that $P_{b,a} = P_{a,b}^T$, $\Delta\Delta^T = \delta^2 I$, $\Delta O = O\Delta^T = \delta O$, we get:

$$P_{b,a}P_{a,b} = \frac{1}{k^2}\left(((\alpha - \delta)^2 + \delta^2)I + \frac{O}{k}(1 - \alpha^2) + (\alpha - \delta)(\Delta^T + \Delta)\right)$$

Lower bounding the Quadratic Bound: Substituting this in Equation (43) along with $E_b = I$, we get:

$$Q^2(x) = \left\| \left(\frac{x-1}{k}\right)^2 (O - kI) - \frac{O}{k^2} + kE_b P_{b,a} P_{a,b} E_b \right\|_F^2$$

$$= \left\| \left(\frac{x-1}{k}\right)^2 (O - kI) + ((\alpha - \delta)^2 + \delta^2)\frac{I}{k} - \alpha^2 \frac{O}{k^2} + \frac{(\alpha - \delta)}{k}(\Delta^T + \Delta) \right\|_F^2$$

Each diagonal element (total k) of the matrix is $\left(\frac{x-1}{k}\right)^2 - \frac{(x-1)^2}{k} + \frac{(\alpha-\delta)^2+\delta^2}{k} - \frac{\alpha^2}{k^2}$.
Each element at the positions of the support $(\Delta + \Delta^T)$ (total $2k$) is $\left(\frac{x-1}{k}\right)^2 - \frac{\alpha^2}{k^2} + \frac{\delta(\alpha-\delta)}{k}$.
Every remaining element (total $k^2 - 3k$) is $\left(\frac{x-1}{k}\right)^2 - \frac{\alpha^2}{k^2}$. To simplify the above equation, we define $\gamma = (1-x)^2 - \alpha^2$, $e = \delta(\alpha - \delta)$. Each diagonal element is $\frac{\gamma}{k^2} - \frac{\gamma}{k} - \frac{2e}{k}$.
Each element at the positions of the support $(\Delta + \Delta^T)$ (total $2k$) is $\frac{\gamma}{k^2} + \frac{e}{k}$.
Every remaining element (total $k^2 - 3k$) is $\frac{\gamma}{k^2}$. Thus, we get:

$$Q^2(x) = k\left(\frac{\gamma}{k^2} - \frac{\gamma}{k} - \frac{2e}{k}\right)^2 + 2k\left(\frac{\gamma}{k^2} + \frac{e}{k}\right)^2 + (k^2 - 3k)\frac{\gamma^2}{k^4}$$

$$= \frac{1}{k^3}\left((k-1)\gamma + 2ke\right)^2 + \frac{2}{k^3}\left(\gamma + ke\right)^2 + \frac{k-3}{k^3}\gamma^2$$

$Q^2(x)$ is minimized for $\gamma = -\frac{2ke}{k-1}$. Substituting this, we get:

$$Q^2(x) \geq \frac{2(k-3)\delta^2(\alpha - \delta)^2 k^2}{k-1}. \tag{46}$$

Computing the determinant of conditional PMF: Let us consider the perturbed symmetric distribution $C(v(\theta, \theta'))$ with the vector

$$v(\theta, \theta') = \left((1 - \theta' - (K-2)\theta), \theta', \underbrace{\theta, \dots, \theta}_{k-2 \text{ times}}\right).$$

For $\theta = \frac{1-\alpha}{k}$ and $\delta = (\theta' - \theta)$ we have $C(v(\theta, \theta')) = P_{b|a}$. We make this switch as this helps us computing the determinant easily.

We now derive some of the necessary results which we will apply in our lower bound graph construction. The determinant of the matrix $C(v(\theta, \theta'))$ is derived first. We have for any $j = 0$ to $k-1$,

$$
\begin{aligned}
\sum_{i=0}^{k-1} v(\theta, \theta')_i \omega^{ji} &= (1 - \theta' - (k-2)\theta) + \theta' \omega^j + \theta \sum_{i=2}^{k-1} \omega^{ji} \\
&= (1 - \theta' - (k-1)\theta) + (\theta' - \theta)\omega^j + \theta \sum_{i=0}^{k-1} \omega^{ji} \\
&= \begin{cases} 1 = (1 - \theta' - (k-1)\theta) + (\theta' - \theta) + k\theta, \ j = 0 \\ (1 - \theta' - (k-1)\theta) + (\theta' - \theta)\omega^j, \ j \neq 0 \end{cases}
\end{aligned}
$$

Therefore, we have following the derivations in Sot

$$
\begin{aligned}
\det(P_{b|a}) = \det(Cir(v(\theta, \theta'))) &= \prod_{j=1}^{k-1} \left((1 - \theta' - (k-1)\theta) - (\theta - \theta')\omega^j \right) \\
&= \frac{(1 - \theta' - (k-1)\theta)^k}{(1 - k\theta)} \prod_{j=0}^{k-1} \left(1 - \frac{(\theta - \theta')}{(1 - \theta' - (k-1)\theta)}\omega^j \right) \\
&= \frac{(1 - \theta' - (k-1)\theta)^k - (\theta - \theta')^k}{(1 - k\theta)} \\
&= (1 - k\theta)^{(k-1)} \left(\left(1 - \frac{\theta' - \theta}{1 - k\theta} \right)^k - \left(\frac{\theta - \theta'}{1 - k\theta} \right)^k \right) \\
&= \alpha^{(k-1)} \left(\left(1 - \frac{\delta}{\alpha} \right)^k - \left(\frac{-\delta}{\alpha} \right)^k \right)
\end{aligned}
$$

In the last line we substitute $\alpha = (1 - k\theta)$ and $\delta = (\theta' - \theta)$ to get back to the form common to other parts of the proof.

I.2 Lower Bound for recovering the equivalence class of trees

In this section we derive the lower bound on the sample complexity to recover the equivalence class when the underlying model has is totally unidentifiable (no leaf is distinguishable from it's parent). For this purpose, we consider PMGMs.

Family of distributions: With the above background, we are now ready to derive the lower bounds. We consider the family of probability distributions which is structurally similar to Appendix A in Tandon et al. [2021], but are PMGMs instead of Ising models. The family of distributions is given as $(P^{(i)} : i = 0, 1, \ldots, t^2)$. The graph $P^{(0)}$ consists of $n = 2t + 1$ nodes $(1, 2, \ldots, 2t + 1)$. Here, we use odd number of nodes for simplifying exposition. There are $2t$ edges where node $j = 1, \ldots, 2t$ are connected to node $(2t + 1)$. Nodes $1, 2 \ldots t$ have distance d_{max} from node $2t + 1$ and are corrupted with probability q_{max}. Nodes $t + 1, t + 2 \ldots 2t$ have distance d_{min} from node $2t + 1$ and have 0 probability of error. Node $2t + 1$ also has 0 probability of error.

For any $i = 1, \ldots, t^2$, the distribution $P^{(i)}$ is constructed from $P^{(0)}$ by disconnecting the edge $(i_a, 2t + 1)$, and adding edge $(i_b + t, 2t + 1)$ where $i_a = (1 + \lfloor \frac{i-1}{t} \rfloor)$, and $i_b = i - \lfloor \frac{i-1}{t} \rfloor t$.
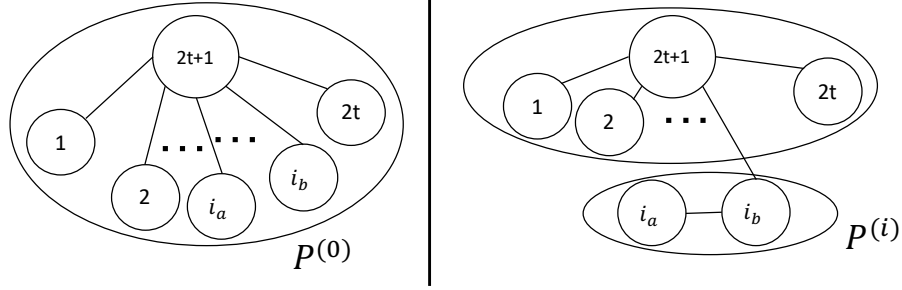
Figure 12: The family of distributions used for providing lower bound for completely unidentifiable case. The graphical model corresponding to $P^{(0)}$ a single recoverable leaf cluster. The graphical model corresponding to $P^{(i)}$, for each $i = 1, \ldots, t^2 - 1$, has nodes $\{i_a, i_b\}$ as one recoverable leaf cluster, and the remaining nodes as another recoverable leaf cluster.

As noted in Tandon et al. [2021], the pair (i_a, i_b) is unique for every $i = 1, \ldots, t^2$. The graph structures of $P^{(0)}$ and $P^{(i)}$ are illustrated in Figure 12.

We next look at the probability distribution matrices for $P^{(0)}$ and $P^{(i)}$. Without loss of generality we consider $P^{(i)}$.

To express the probability distribution, we first define w_m, w_M and S as:

$$w_M = \frac{1}{2}(1 - \exp(-d_{max})), w_m = \frac{1}{2}(1 - \exp(-d_{min})), S = kI - O$$

For $P^{(0)}$ we have:

$$P_i^{(0)} = \frac{I}{k}, i \in \{1, 2, \ldots, 2t + 1\},$$

$$P_{i|2t+1}^{(0)} = I + w_M A, i \in \{1, 2, \ldots t\},$$

$$P_{i|2t+1}^{(0)} = I + w_m A, i \in \{t + 1, t + 2, \ldots 2t\},$$

$$P_{i'|i}^{(0)} = I + \frac{q}{k}S, i \in \{1, 2, \ldots t\},$$

$$P_{i'|i}^{(0)} = I, i \in \{t + 1, t + 2, \ldots 2t + 1\}.$$

For $P^{(1)}$ we have:

$$P_i^{(1)} = \frac{I}{k}, i \in \{1, 2, \ldots, 2t + 1\},$$

$$P_{i|2t+1}^{(1)} = I + w_M A, i \in \{2, 3, \ldots t\},$$

$$P_{i|t+1}^{(1)} = I + w_M A,$$

$$P_{i|2t+1}^{(1)} = I + w_m A, i \in \{t + 1, t + 2, \ldots 2t\},$$

$$P_{i'|i}^{(1)} = I + \frac{q}{k}S, i \in \{1, 2, \ldots t\},$$

$$P_{i'|i}^{(1)} = I, i \in \{t + 1, t + 2, \ldots 2t + 1\}.$$

Using

$$(I + aA)(I + bA) = I + (a + b - 2ab)A,$$

$$(I + aA)(I + bS) = I + bS + a(1 - kb)A,$$

we get:

$$P^{(0)}_{1'|(2t+1)'} = P^{(1)}_{1'|(t+1)'} = I + \frac{q_{max}}{k}S + w_M(1 - q_{max})A, \tag{47}$$

$$P^{(0)}_{1'|(t+1)'} = P^{(1)}_{1'|(2t+1)'} = I + q_{max}S + (w_m + w_M - 2w_mw_M)(1 - q_{max})A. \tag{48}$$

Note that:

$$\mathbb{E}_{\mathbf{X}\sim P^{(0)}} \log\left(\frac{P^{(0)}(\mathbf{X})}{P^{(1)}(\mathbf{X})}\right) = \mathbb{E}_{\mathbf{X}\sim P^{(0)}} \log(P^{(0)}(X'_1|X'_{2t+1}) - \mathbb{E}_{\mathbf{X}\sim P^{(0)}} \log(P^{(1)}(X'_1|X'_{t+1}))).$$

We have that:

$$\mathbb{E}_{\mathbf{X}\sim P^{(0)}} \log(P^{(0)}(X'_1|X'_{2t+1})$$

$$=\mathbb{E}_{X'_1, X'_{2t+1}\sim P^{(0)}} \log(P^{(0)}(X'_1|X'_{2t+1})$$

$$= \sum_{(X'_1, X'_{2t+1})\in\mathcal{S}^2} P^{(0)}(X'_1, X'_{2t+1}) \log(P^{(0)}(X'_1|X'_{2t+1})$$

$$=\frac{1}{k} \sum_{(X'_1, X'_{2t+1})\in\mathcal{S}^2} P^{(0)}(X'_1|X'_{2t+1}) \log(P^{(0)}(X'_1|X'_{2t+1})$$

$$=\frac{1}{k}\left[(k-2)\left(1 - \left(1 - \frac{1}{k}\right)q_{max}\right)\log\left(1 - \left(1 - \frac{1}{k}\right)q_{max}\right)\right.$$

$$+ (k^2 - k - 2)\left(\frac{q_{max}}{k}\log\left(\frac{q_{max}}{k}\right)\right)$$

$$+ 2\left(\left((1 - q_{max})(1 - w_M) + \frac{q_{max}}{k}\right)\log\left((1 - q_{max})(1 - w_M) + \frac{q_{max}}{k}\right)\right.$$

$$\left.\left.+ \left(\frac{q_{max}}{k} + w_M(1 - q_{max})\right)\log\left(\frac{q_{max}}{k} + w_M(1 - q_{max})\right)\right)\right].$$

Similarly, we also have that :

$$\mathbb{E}_{\mathbf{X}\sim P^{(0)}} \log(P^{(1)}(X'_1|X'_{t+1})$$

$$=\frac{1}{k}\left[(k-2)\left(1 - \left(1 - \frac{1}{k}\right)q_{max}\right)\log\left(1 - \left(1 - \frac{1}{k}\right)q_{max}\right)\right.$$

$$+ (k^2 - k - 2)\left(\frac{q_{max}}{k}\log\left(\frac{q_{max}}{k}\right)\right)$$

$$+ 2\left(\left((1 - q_{max})(1 - w_M - w_m + 2w_mw_M) + \frac{q_{max}}{k}\right)\log\left((1 - q_{max})(1 - w_M) + \frac{q_{max}}{k}\right)\right.$$

$$\left.\left.+ \left(\frac{q_{max}}{k} + (w_M + w_m - 2w_mw_M)(1 - q_{max})\right)\log\left(\frac{q_{max}}{k} + w_M(1 - q_{max})\right)\right)\right].$$

Thus, we get:

$$\mathbb{E}_{\mathbf{X}\sim P^{(0)}} \log\left(\frac{P^{(0)}(\mathbf{X})}{P^{(1)}(\mathbf{X})}\right) = \frac{2}{k}w_m(1 - 2w_M)(1 - q_{max})\log\frac{(1 - q_{max})(1 - w_M) + \frac{q_{max}}{k}}{\frac{q_{max}}{k} + w_M(1 - q_{max})}$$

$$= \frac{2}{k}w_m(1 - 2w_M)(1 - q_{max})\log\left(1 + \frac{(1 - 2w_M)(1 - q_{max})}{\frac{q_{max}}{k} + w_M(1 - q_{max})}\right)$$

$$\leq \frac{2w_m(1 - 2w_M)^2(1 - q_{max})^2}{q_{max} + w_M(1 - q_{max})k}.$$

Therefore, we have:

$$J(P^{(0)}, P^{(1)}) \leq \frac{4w_m(1 - 2w_M)^2(1 - q_{max})^2}{q_{max} + w_M(1 - q_{max})k}$$

Substituting w_m and w_M, we get:

$$J(P^{(0)}, P^{(1)}) \leq \frac{4(1 - \exp(-d_{min}))\exp(-2d_{max})(1 - q_{max})^2}{2q_{max} + (1 - \exp(-d_{max}))(1 - q_{max})k}$$

Using the fact that $J(P^{(0)}, P^{(1)}) = J(P^{(0)}, P^{(i)}) \ \forall \ i \in \{2, \dots, t^2\}$, Lemma 5 gives us that for any algorithm to correctly recover $\mathcal{T}_{T^*}^{sub}$ with probability at least $1 - \delta$, the number of samples required are:

$$N = \Omega \left(\frac{2q_{max} + (1 - \exp(-d_{max}))(1 - q_{max})k}{(1 - \exp(-d_{min})) \exp(-2d_{max})(1 - q_{max})^2} (1 - \delta) \log n \right)$$

I.3 Lower bound for recovering $\mathcal{T}_{T^*}^{sub}$ when $\mathcal{T}_{T^*}^{sub} \subset \mathcal{T}_{T^*}$

In this section, we focus on the dependence of t_0 which can not be captured when the graph is completely unidentifiable. Therefore, we create graphs using perturbed symmetric distribution where the graph is partly identifiable (a subset of leaf nodes is distinguishable from it's parent).

Family of distributions: We consider graphical models with random variables whose support size is $k \geq 4$. We construct a family of $n+1$ star structured distributions on $n+1$ nodes (as shown in Figure 13), $P^{(0)}, P^{(1)}, \dots, P^{(n)}$, such that $P^{(0)}$ is completely identifiable while $P^{(i)}$ is such that leaf node i and the center node 0 is unidentifiable.

We next provide the details of the family of graphical models.
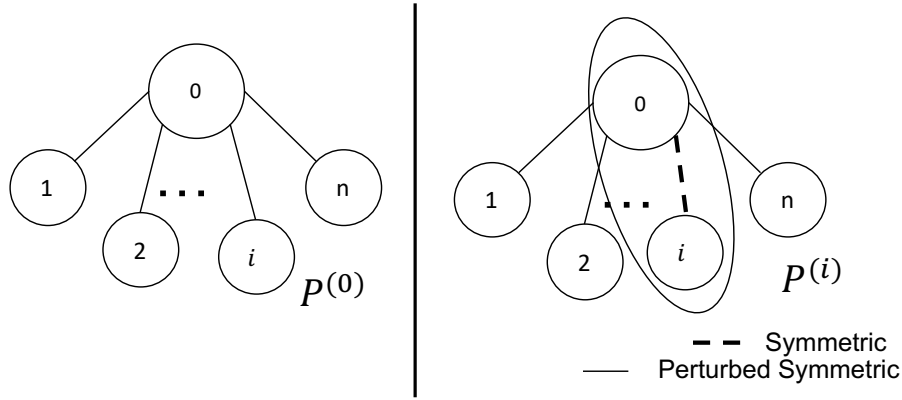

Figure 13: The family of distributions used for providing lower bound with t_0 dependence. The graphical model corresponding to $P^{(0)}$ is completely identifiable. The graphical model corresponding to $P^{(i)}$, for each $i = 1, \dots, n$, has edge $\{i, 0\}$ which forms a recoverable leaf cluster, and the rest are all identifiable.

For $P^{(0)}$, the conditional distribution matrices are as follows:

$$P_{X_j|X_0}^{(0)} = (\alpha - \delta)I + (1 - \alpha)\frac{O}{k} + \Delta, \forall j \in [n],$$

where

$$\Delta = \begin{bmatrix} 0 & \delta & 0 & \dots & 0 \\ 0 & 0 & \delta & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \delta \\ \delta & 0 & 0 & \dots & 0 \end{bmatrix}$$

For $P^{(i)}$, the conditional distribution matrices are as follows:

$$P_{X_j|X_0}^{(i)} = (\alpha - \delta)I + (1 - \alpha)\frac{O}{k} + \Delta, \forall j \in [n], j \neq i.$$

$$P_{X_i|X_0}^{(i)} = \alpha I + (1 - \alpha)\frac{O}{k}.$$

Recall from Equation (46), this conditional distribution ensures that in $P^{(0)}$, all the leaves can be identified. It also ensures that in $P^{(i)}$ all the leaves other than i can be identified. It is easy to see that $(\alpha - \delta)I + (1 - \alpha)\frac{O}{k} + \Delta = C(v(\theta, \theta'))$ for $\theta = \frac{1-\alpha}{k}, \theta' = \frac{1-\alpha}{k} + \delta$. The marginals of all the random variables in all the distributions are

uniform on the support. Given the graph structure and the uniform marginals, the joint PMF of the random variables can be decomposed as follows:

$$P^{(0)}(\mathbf{X}) = \frac{1}{k} \prod_{j=1}^{n} P^{(0)}(X_j|X_0), \tag{49}$$

$$P^{(i)}(\mathbf{X}) = \frac{1}{k} \prod_{j=1}^{n} P^{(i)}(X_j|X_0). \tag{50}$$

Recall that $P^{(0)}_{X_j|X_0}$ is the matrix form of conditional distribution whereas $P^{(0)}(X_j|X_0)$ is the scalar value of the conditional PMF for any X_j and X_0.

KL Divergence Computation   We now calculate the symmetrized KL divergence between $P^{(0)}$ and $P^{(i)}$ for $i \neq 0$ denoted by $J(P^{(0)}, P^{(i)})$.

$$J(P^{(0)}, P^{(i)}) = \mathbb{E}_{\mathbf{X} \sim P^{(i)}} \log \frac{P^{(i)}(\mathbf{X})}{P^{(0)}(\mathbf{X})} + \mathbb{E}_{\mathbf{X} \sim P^{(0)}} \log \frac{P^{(0)}(\mathbf{X})}{P^{(i)}(\mathbf{X})}$$

Substituting $P^{(0)}(\mathbf{X}), P^{(i)}(\mathbf{X})$ from equation 49 and noting that $P^{(0)}(X_j|X_0) = P^{(i)}(X_j|X_0) \ \forall j \neq i$, we get that:

$$J(P^{(0)}, P^{(i)}) = \mathbb{E}_{P^{(i)}} \log \frac{P^{(i)}(X_i|X_0)}{P^{(0)}(X_i|X_0)} + \mathbb{E}_{P^{(0)}} \log \frac{P^{(0)}(X_i|X_0)}{P^{(i)}(X_i|X_0)}$$

Therefore to compute $J(P^{(0)}, P^{(i)})$, we need $\mathbb{E}_{P^{(i)}} \log P^{(i)}(X_i|X_0)$, $\mathbb{E}_{P^{(i)}} \log P^{(0)}(X_i|X_0)$, $\mathbb{E}_{P^{(0)}} \log P^{(0)}(X_i|X_0)$ and $\mathbb{E}_{P^{(0)}} \log P^{(i)}(X_i|X_0)$. We first calculate $\mathbb{E}_{P^{(i)}} \log P^{(i)}(X_i|X_0)$. Note that $P^{(i)}(X_i = x_i|X_0 = x_0)$ takes only 2 values - $\alpha + (1 - \alpha)/k$ (whenever $x_i = x_0$, that is, for k combinations of x_i, x_0), $(1 - \alpha)/k$ (whenever $X_i \neq X_0$, that is, for $k^2 - k$ combinations of x_i, x_0).

$$\begin{aligned}
\mathbb{E}_{P^{(i)}} \log P^{(i)}(X_i|X_0) &= \sum_{x_i, x_0 \in \mathcal{S} \times \mathcal{S}} P^{(i)}(X_i = x_i, X_0 = x_0) \log P^{(i)}(X_i = x_i|X_0 = x_0) \\
&= \sum_{x_i = x_0} P^{(i)}(X_i = x_i, X_0 = x_0) \log P^{(i)}(X_i = x_i|X_0 = x_0) \\
&\quad + \sum_{x_i \neq x_0} P^{(i)}(X_i = x_i, X_0 = x_0) \log P^{(i)}(X_i = x_i|X_0 = x_0) \\
&= \sum_{x_i = x_0} P^{(i)}(X_i = x_i|X_0 = x_0) P^{(i)}(X_0 = x_0) \log P^{(i)}(X_i = x_i|X_0 = x_0) \\
&\quad + \sum_{x_i \neq x_0} P^{(i)}(X_i = x_i|X_0 = x_0) P^{(i)}(X_0 = x_0) \log P^{(i)}(X_i = x_i|X_0 = x_0) \\
&= k \left(\alpha + \frac{1 - \alpha}{k} \right) \frac{1}{k} \log \left(\alpha + \frac{1 - \alpha}{k} \right) + k(k - 1) \frac{1 - \alpha}{k} \frac{1}{k} \log \left(\frac{1 - \alpha}{k} \right) \\
&= \left(\alpha + \frac{1 - \alpha}{k} \right) \log \left(\alpha + \frac{1 - \alpha}{k} \right) + \frac{k - 1}{k}(1 - \alpha) \log \left(\frac{1 - \alpha}{k} \right).
\end{aligned}$$

We next calculate $\mathbb{E}_{P^{(i)}} \log P^{(0)}(X_i|X_0)$. $P^{(0)}(X_i|X_0)$ takes 3 different values - $\left(\alpha + \frac{1-\alpha}{k} - \delta \right)$ (for k combinations of x_i, x_0), $\frac{1-\alpha}{k} + \delta$ (for k combinations of x_i, x_0), $\frac{1-\alpha}{k}$ (for $k^2 - 2k$ combinations of x_i, x_0).

$$\begin{aligned}
\mathbb{E}_{P^{(i)}} \log P^{(0)}(X_i|X_0) &= k \left(\alpha + \frac{1 - \alpha}{k} \right) \frac{1}{k} \log \left(\alpha + \frac{1 - \alpha}{k} - \delta \right) + k \left(\frac{1 - \alpha}{k} \right) \frac{1}{k} \log \left(\frac{1 - \alpha}{k} + \delta \right) + \\
&\quad + k(k - 2) \frac{1 - \alpha}{k} \frac{1}{k} \log \left(\frac{1 - \alpha}{k} \right) \\
&= \left(\alpha + \frac{1 - \alpha}{k} \right) \log \left(\alpha + \frac{1 - \alpha}{k} - \delta \right) + \frac{1 - \alpha}{k} \log \left(\frac{1 - \alpha}{k} + \delta \right) \\
&\quad + \frac{k - 2}{k}(1 - \alpha) \log \left(\frac{1 - \alpha}{k} \right)
\end{aligned}$$

Evaluating the remaining terms on similar lines gives us:

$$\mathbb{E}_{X_i, X_0 \sim P^{(0)}} \log P^{(0)}(X_i | X_0) = \left(\alpha + \frac{1-\alpha}{k} - \delta \right) \log \left(\alpha + \frac{1-\alpha}{k} - \delta \right)$$
$$+ \left(\frac{1-\alpha}{k} + \delta \right) \log \left(\frac{1-\alpha}{k} + \delta \right) + \frac{k-2}{k}(1-\alpha) \log \left(\frac{1-\alpha}{k} \right),$$

$$\mathbb{E}_{X_i, X_0 \sim P^{(0)}} \log P^{(i)}(X_i | X_0) = \left(\alpha + \frac{1-\alpha}{k} - \delta \right) \log \left(\alpha + \frac{1-\alpha}{k} \right)$$
$$+ \left(\frac{k-1}{k}(1-\alpha) + \delta \right) \log \left(\frac{1-\alpha}{k} \right).$$

This gives us:

$$J(P^{(0)}, P^{(i)}) = \delta \left[\log \left(1 + \frac{k\delta}{1-\alpha} \right) - \log \left(1 - \frac{k\delta}{k\alpha + (1-\alpha)} \right) \right]$$
$$\leq k\delta^2 \left(\frac{1}{1-\alpha} + \frac{1}{1 + (k-1)\alpha} \right)$$
$$\leq \frac{(k-1)}{8k(k-3)\alpha^2} \left(\frac{1}{1-\alpha} + \frac{1}{1+(k-1)\alpha} \right) \times t_0^2, \quad \text{for } t_0 \leq \frac{k\sqrt{k-3}\alpha^2}{\sqrt{2(k-1)}}, k \geq 4.$$

The second last inequality holds as for $\log((1+ax)/(1-bx)) \leq (a+b)x$ for $x > 0$, $a > 0$, $b > 0$, and $b \leq a$.

We now reason about the final inequality. We have $Q^2(x) \geq \frac{2(k-3)k^2}{(k-1)} \delta^2 (\alpha - \delta)^2$ for $k \geq 4$. If we have $\delta < \alpha/4$ then we have $Q^2(x) \geq \frac{(k-3)k^2}{8(k-1)} \delta^2 \alpha^2$. But we are dealing with the situation when $Q^2(x) \geq t_0^2$. This means we must choose δ in a way such that $t_0^2 \leq \frac{(k-3)k^2}{8(k-1)} \delta^2 \alpha^2$. Let $\delta = \frac{\sqrt{(k-1)}}{k\sqrt{8(k-3)}\alpha} t_0$. This choice satisfies $\delta \leq \alpha/4$ for $t_0 \leq \frac{k\sqrt{(k-3)}\alpha^2}{\sqrt{2(k-1)}}$. Hence, replacing $\frac{\sqrt{(k-1)}}{k\sqrt{8(k-3)}\alpha} t_0$ gives the final inequality for the symmetrized KL divergence above.

As we have $\delta \leq \alpha/4$ and $k \geq 4$, we can simplify the determinant term as

$$\det(P_{X_i|X_0}) = \alpha^{(k-1)} \left(\left(1 - \frac{\delta}{\alpha} \right)^k - \left(\frac{-\delta}{\alpha} \right)^k \right)$$
$$\det(P_{X_i|X_0}) \leq \alpha^{(k-1)}, \quad \det(P_{a|b}) \geq \alpha^{(k-1)} \frac{3^k - 1}{4^k}$$
$$d_{max} \leq -(k-1)\log(\alpha) - \log\left(\frac{3^k-1}{4^k}\right) \leq -(k-1)\log(\alpha \sqrt[3]{16/5}), \quad d_{min} \geq -(k-1)\log(\alpha)$$
$$\alpha \geq \frac{1}{2} \exp(-d_{max}/(k-1)), \quad \alpha \leq \exp(-d_{min}/(k-1)).$$

If we use $\alpha = \exp(-d_{min}/(k-1))$ for our construction, the symmetrized KL divergence in terms of the distance bounds, for $k \geq 4$ and $t_0 \leq \frac{k\sqrt{k-3}\alpha^2}{\sqrt{2(k-1)}}$, is

$$J(P^{(0)}, P^{(i)}) \leq \frac{(k-1)}{8k(k-3)\alpha^2} \left(\frac{1}{1-\alpha} + \frac{1}{1+(k-1)\alpha} \right) \times t_0^2$$
$$\leq \frac{(k-1)}{8k(k-3)\exp(-2d_{min}/(k-1))} \left(1 + \frac{1}{1-\exp(-d_{min}/(k-1))} \right) \times t_0^2$$

Lower Bound Proof - Part II: We now derive the second part of Theorem 5, thus concluding its proof.

Plugging the above symmetrized KL bound in Lemma 5 we obtain that for a probability error of at most $\delta > 0$ we require at least N samples where

$$N > \left(1 - \delta + \frac{1}{\log(n)}\right) \frac{\log(n)}{\frac{n}{n+1} \frac{(k-1)}{8k(k-3)\exp(-2d_{min}/(k-1))} \left(1 + \frac{1}{1-\exp(-d_{min}/(k-1))}\right) \times t_0^2}$$
$$\geq \frac{(1-\delta)\exp(-\frac{2d_{min}}{k-1})(1 - \exp(-\frac{d_{min}}{k-1}))8k(k-3)\log(n)}{(k-1)(2 - \exp(-\frac{d_{min}}{k-1}))t_0^2}$$
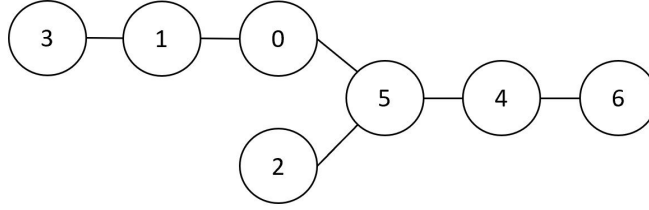
Figure 14: Randomly generated graph used for algorithm evaluation.

Therefore, we have $N = \Omega\left(\dfrac{(1-\delta)\exp(-\frac{2d_{\min}}{k-1})(1-\exp(-\frac{d_{\min}}{k-1}))k\log(n)}{t_0^2}\right)$

Instead using $\alpha = \frac{1}{2}\exp(-d_{\max}/(k-1))$ in our construction, following similar steps, we obtain

$$N = \Omega\left(\frac{(1-\delta)\exp(-\frac{2d_{\max}}{k-1})(1-\exp(-\frac{d_{\max}}{k-1}))k\log(n)}{t_0^2}\right).$$

Combining these two we obtain the final lower bound in this setting ($k \geq 4$ and $t_0 \leq \frac{\sqrt{3}}{4\sqrt{10}}k\exp(-2\frac{d_{\max}}{k-1})$) as

$$N = \Omega\left(\max_{d\in\{d_{\max},d_{\min}\}}\frac{(1-\delta)\exp(-\frac{2d}{k-1})(1-\exp(-\frac{d}{k-1}))k\log(n)}{t_0^2}\right).$$

J   Experiments

Number of iterations used for $k = 2$ in comparison with SGA : We used 1000 iterations to obtain the results for $k = 2$ presented in Figure 3.

Next we provide more results for varying graph shapes and other parameters. *All the experiments in this section are for $k = 4$.*

J.1   Varying graph shape, δ

In this part we see the impact of δ on the performance of the algorithm for different graphs. We execute the algorithm for a lot of randomly generated graphs and the algorithm converges to the correct output. We report the results for 3 different graph structures - star, chain and one of the many randomly generated graphs (Figure 14).

Setting: (i) Number of nodes = 7.
(ii) Graph Shape = {Chain, Star, Random}
(iii) Distance of all the adjacent nodes = 0.7.
(iv) Error probability is uniformly sampled from $[0, 0.2]$.
(v) $\delta \in \{0.00, 0.02, 0.04\}$
(vi) Assume access to q_{max}, d_{min} but not to d_{max}, t_0.
(vii) Number of iterations = 100

Takeaways:

1. We witness the transition from unidentifiability to identifiability. When $\delta = 0$, the exact graph cannot be recovered and hence the exact recovery fraction remains low consistently regardless of the number of samples. Higher δ has faster convergence to the correct graph.

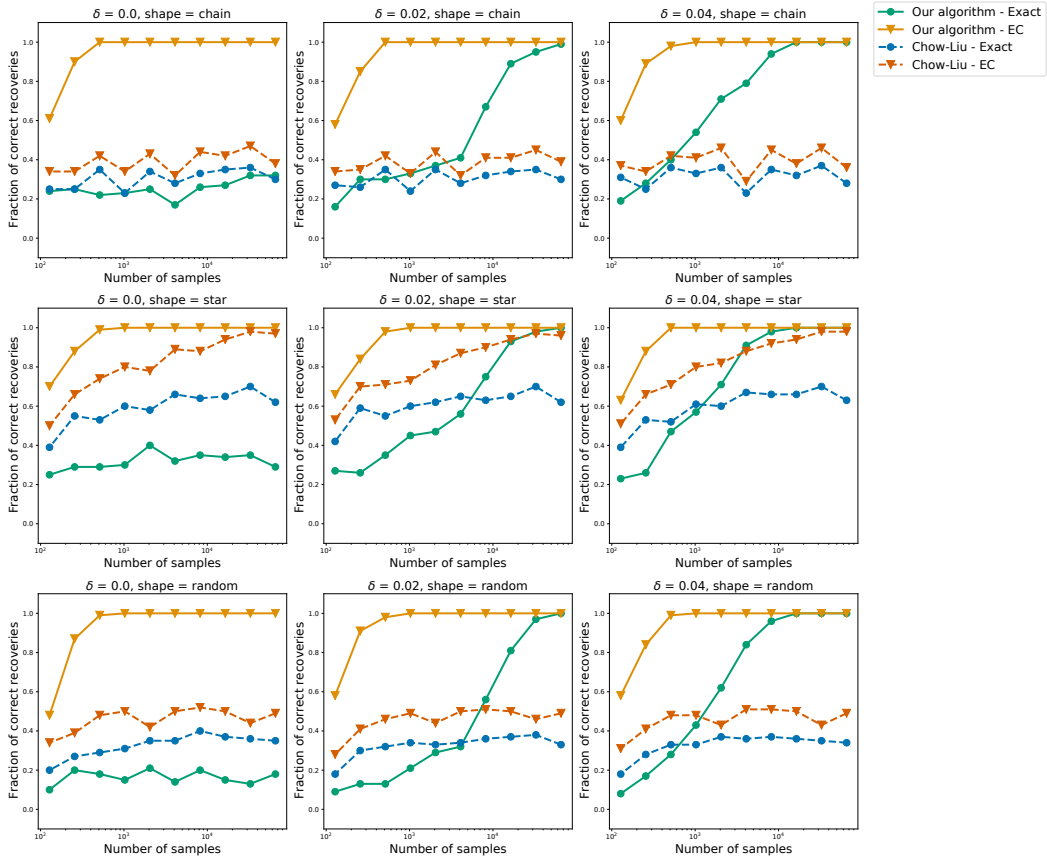2. Learning a tree from the equivalence class requires much fewer samples.

Figure 15: Comparing the performance of our algorithm and Chow-Liu over different values of $\delta_{i,j} \in \{0.00, 0.02, 0.04\}$ and different graph shapes - chain, star, random. Setting: $d_{min} = d_{max} = 0.7$, $q_{max} = 0.2$, # of nodes= 7. For both algorithms, we provide results for two cases: i) when the exact underlying tree is recovered, ii) when a tree from the equivalence class is recovered.
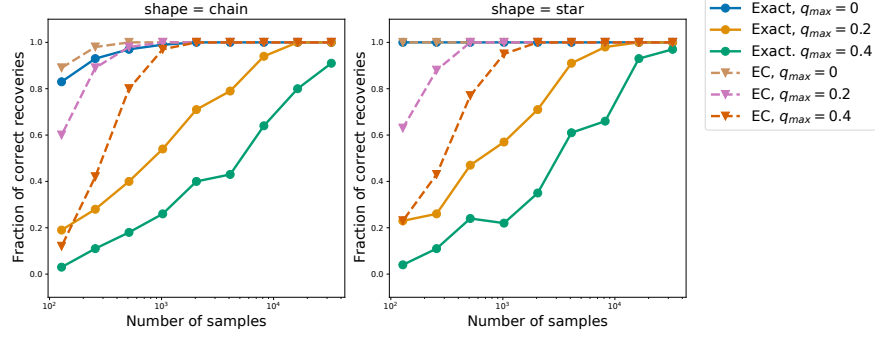
Figure 16: Comparing the performance of our algorithm for different values of $q_{max} \in \{0, 0.2, 0.4\}$ and different graph shapes - chain, star. Setting: $d_{min} = d_{max} = 0.7$, $\delta = 0.04$ # of nodes= 7. We provide results for two cases: i) when the exact underlying tree is recovered, ii) when a tree from the equivalence class is recovered.

3. For the given noise model when the probability of error is randomly selected, for a significant number of realizations in the star shape, the Chow-Liu remains in the equivalence class. However, it lags behind considerably compared to our algorithm.

4. Chow-Liu has high error for complete recovery.

J.2 Varying q_{max}

Now, we study the impact of the probability of error on the performance of the algorithm.

Setting: (i) Number of nodes = 7.
(ii) Graph Shape = {Chain, Star}
(iii) Distance of all the adjacent nodes = 0.7.
(iv) Error probability is uniformly sampled from $[0, q_max]$, where, $q_{max} \in \{0, 0.2, 0.4\}$.
(v) $\delta = 0.04$
(vi) Assume access to q_{max}, d_{min} but not to d_{max}, t_0.
(vii) Number of iterations = 100
Takeaway: The convergence is slower for higher q_{max} as demonstrated in Figure 16.

K   Future Work Insights

K.1   Extension 1:

In this section we give insights into the setting when the distribution of noise is non-uniform but known, that is:

$$X_i' = \begin{cases} X_i & \text{w.p. } 1 - q_i, \\ Q_i & \text{w.p. } q_i, \end{cases} \tag{51}$$

where Q_i are discrete i.i.d random variables random variables with support on $\mathcal{S}$ with a known distribution $\forall$ $i \in \{1, 2, \dots n\}$.

Let the PMF of Q_i be:

$$P(Q_i = s_j) = \gamma_j \ \forall j \in \{1, 2, \dots k\}$$

Define the matries M and M_D as follows:

$$M = \begin{bmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_k \\ \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_k \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_k \\ \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_k \end{bmatrix}.$$

$$M_D = diag([\gamma_1, \gamma_2, \gamma_3, \ldots, \gamma_k])$$

Define matrix E_a as follows:

$$E_a = (1 - q_a)I + q_a M$$

Then we have that:

$$P_{a',b'} = E_a P_{a,b} E_b^T$$

Following on the lines of Appendix C and noting that $\left(\frac{E_2}{1-q_2}\right)^{-1} = I - q_2 M$ we find the error estimate for X_2 when $X_1 \perp X_3 | X_2$ is given by:

$$
q_2^2 (M P_{2',3'} P_{1',3'}^{-1} P_{1',2'} M^T - M_D) - q_2(M P_{2',3'} P_{1',3'}^{-1} P_{1',2'} + P_{2',3'} P_{1',3'}^{-1} P_{1',2'} M^T - P_{2'} - M_D) \\
+ P_{2',3'} P_{1',3'}^{-1} P_{1',2'} - P_{2'} = 0
\tag{52}
$$

This gives us a test for which node is a center node among three nodes. Note that the star/non-star test would be valid as it is in this setting. Thus the techniques in this paper could be used to address the setting when the noise has a non-uniform distribution.

K.2 Extension 2:

The next potential extension is to the setting when the observation could come from different *known* distributions with a corresponding *unknown* probability of error. In this setting the observed random variable would take the form:

$$
X_i' = \begin{cases} X_i & \text{w.p. } 1 - \sum_{j=1}^{e} q_{i,j}, \\ Q_{i,j} & \text{w.p. } q_{i,j}, \ \forall j \in \{1, 2, \ldots e\}. \end{cases}
\tag{53}
$$

In this setting, we can define matrices M_j as follows:

$$
M_j = \begin{bmatrix} \gamma_{1,j} & \gamma_{2,j} & \gamma_{3,j} & \cdots & \gamma_{k,j} \\ \gamma_{1,j} & \gamma_{2,j} & \gamma_{3,j} & \cdots & \gamma_{k,j} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \gamma_{1,j} & \gamma_{2,j} & \gamma_{3,j} & \cdots & \gamma_{k,j} \\ \gamma_{1,j} & \gamma_{2,j} & \gamma_{3,j} & \cdots & \gamma_{k,j} \end{bmatrix}.
$$

Matrix E_a is defined as:

$$
E_a = (1 - \sum_{j=1}^{e} q_{i,j})I + \sum_{j=1}^{e} q_{i,j} M_j.
\tag{54}
$$

Now, the observed PMF is given by:

$$P_{a',b'} = E_a P_{a,b} E_b^T$$

Error estimation step would now require estimating each of $q_{i,j}$. Then, arguing on similar lines as this paper, one could check the consistency in the estimates of $q_{i,j}$ to get insights into the graph structure.

K.3 Extension 3:

Solving this problem for non-tree Markov Random Fields introduces new challenges as the ideas of finding center node among 3 nodes and star/non-star classifications are no longer well defined. An approach in this setting could be to find minimal separator nodes for different pairs of nodes to study the identifiability. Getting insights into whether this problem would be solvable using polynomial time algorithms would also be very interesting.