Copyright

by

Ashish Katiyar

2021

The Dissertation Committee for Ashish Katiyar
certifies that this is the approved version of the following dissertation:

Robust Estimation of Tree Structured Probabilistic Graphical Models

Committee:

Constantine Caramanis, Supervisor

Rachel Ward

Sanjay Shakkottai

Sujay Sanghavi

Georgios (Alex) Dimakis

Robust Estimation of Tree Structured Probabilistic Graphical Models

by

Ashish Katiyar

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2021

Dedicated to Maa and Papa.

Acknowledgments

Today I am at the stage of writing my PhD dissertation. When I look back at my journey, I see that I have come a long way. Had this been a solo journey I would have crashed out long ago, but I have been very fortunate to have amazing people in my life who have supported me and kept me going. I can never thank these people enough; they are my most precious treasure. Also, right in the spirit of the whole PhD, I am writing this really close to the deadline, so I am sure this would be far from a perfect acknowledgment, but then, even the hypothetical perfect acknowledgment would fall very short of conveying my appreciation to everyone, so between friends, let's work with this.

I consider myself extremely fortunate to have Prof. Constantine Caramanis as my advisor. His advising style focused on my overall development as a researcher. I learned how to transform real-world problems into precise research problems and take a systematic approach towards problem-solving by asking the right questions. There were so many instances of us getting stuck and he was incredibly supportive and patient in such tough times. While he played a central role in honing my research skills, his contribution went far beyond just the academic setup. Throughout my PhD, I have always had the reassurance that he has my back and that, I believe, has helped me keep going this whole time. He helped me decide the next steps after my PhD. The later part of my PhD was a rather rough ride because of the challenges posed by COVID and his support during the whole time is absolutely priceless. I never imagined I would have an advisor who would care so much about my well-being.

I would like to thank Prof. Sujay Sanghavi, Prof. Sanjay Shakkottai, Prof. Alex Dimakis, and Prof. Rachel Ward for being on my committee. Their insights on the research as well as the motivation they provided helped immensely in shaping this dissertation.

I got an opportunity to amazing researchers during my PhD – Jessica, Vatsal, and Soumya. Jessica's knack for questioning everything is amazing, it is what gave rise to my first research problem. She played a crucial in solving that problem. When I collaborated with Vatsal on the second problem, I had no prior experience of working with Ising models. He contributed immensely with his vast experience and helped me by guiding me in the right direction whenever needed. He is very particular about presentation and planning which in hindsight turned out to be very helpful in my journey as a researcher. Collaborating with Soumya has been a privilege. He is an amazing researcher, I literally cannot think of any research-related conversation with him where I did not come out with a new insight.

One of the most important aspects of any PhD student's experience is the research group they are a part of and I am fortunate to be a part of an amazing one. Tianyang always had very insightful inputs in all the group meetings and I am still amazed by his multitasking skills. Jessica is the life of every room she is a part of and obviously a great researcher. It was amazing to have Kiyeon in the group, he was our TA in the convex optimization course and I learned a lot from him in my formative years. I had a lot of fun conversations with Liu on a range of topics, he is very supportive. In my initial days during the PhD, whenever I felt lost or stressed, Eirini was always there to support me. Life would have been much more difficult had I not had her support. I am very happy to have found a great friend in Jeongyeol, a great researcher, and an even better human being. I am very fortunate to be friends with Jiacheng, he is academically smart and street smart. He is always willing to go

out of his way to help in every aspect. Matthew continues to amaze me with his dedication, so smart and so hardworking. Liam always had great insights in the group meetings and is a pleasure to hang out with. Another person who has left a long-lasting impression on me is Orestis, he is one of the best presenters, I always came out learning something new whenever he was presenting. A person who is just a pleasure to be around is Isidoros, he can smile through anything that life throws. Due to the pandemic, I did not get an opportunity to interact a lot with Alexia, Georgios, Kelsey, and Sanika but I am confident that they will do great work in the future. I can't wait to see you people shine!

I absolutely loved being a part of WNCG as I got to know a great group of people. Dave and Yanni are the two people who really helped me survive the first year. From working together on homeworks and projects to cooking sessions, we stuck together through thick and thin. Dave and Jacob introduced me to the American way of life. Justin is another great person that I am fortunate to be friends. He played a very important role in my PhD by being an endless source of Colombian candy. I have had amazing times with Kartik, Yi, Nithin, Manan, Nitin, Monica, Jean, Ajil, Nihal, Ronshee, Akash, Alan, Diego, Rajesh, and Ian. I am especially grateful to the people who are my seniors, Ahmad, Murat, Shalmali, Avro for being a constant source of inspiration and a lot of board game sessions.

I cannot thank the WNCG and ECE staff enough. A huge shout out to Melanie, Melody, Karen, Jaymie, and Apipol (who was my consistent source of free Wednesday lunch post the faculty meeting). Whenever I needed them, they put their whole weight to help me in every way possible. They make everything happen, ranging from tuition and fees to TWS, faculty talks, socials, potlucks. They are the ones who have always kept the show going.

I would also like to thank my advisor during my master's, Prof. Aniruddha Datta. I really enjoyed my research experience with him. He was incredibly supportive during my PhD applications and if it weren't for him, I wouldn't be writing this dissertation.

Roommates always play a huge role in how our lives are going and I am very fortunate to have had Aseem and Manan as my roommates. Aseem played a huge role in helping me out when I was starting as a PhD student by guiding me on how to approach Professors to be my advisor. We had a great time, ate great food together, had conversations about cricket and politics as well as shared our successes and failures. Manan is probably the most relaxed roommate one can ever have. He was always down for a probability question whenever I was stuck with anything. Going through the interview process would have been extremely difficult had it not been for him. Our pizza celebrations are something I am going to miss a lot. Another person who was not a roommate but still is very close to me is Teja. He is one of the most selfless people I have ever met.

A person that deserves a paragraph of his own is Vatsal. My first impression of him was that he is a reinforcement learning expert. I am no longer sure about that part but, little did I know that he will have such an impact on my journey. My PhD journey would have been drastically different had it not been for him. There are a lot of people who help us in our journey, there are a few who make it a priority to help us out as if it was their journey. I appreciate that he helped me get both of my internships and in my job search, he cooked great food and invited me over, we had great conversations on our road trip, he tried to teach me to drive, got me gifts from every place he visited. While these things are all great, the thing I value the most is the trust that he has my back.

A special mention goes out to the group of Monica, Manan, Vatsal, and Soumya. Our

group conversations are invaluable. Monica has played a very important role in my support system. She is great at lifting spirits; she always knew the right thing to say whenever I was feeling low. Soumya is not only a genius when it comes to research, but he is also great at practical life advice. Another friend who supported me immensely during my PhD is Anshu whose support I will forever be grateful for.

My master's was a very crucial stepping stone towards the PhD and I would like to thank all my friends who helped me get through it. Kirthi, Narendra, and Ajay were great roommates who always motivated me to aim higher. I have known Narendra since my undergraduate days and he has been a great friend. I had long conversations and great times with Rajan and Pravir. Priya, Prerana, and Parul often hosted us for tea, food, darts, Jenga, and countless other things. I was fortunate to have a great bunch of friends in Sangam, Shrija, Deeksha, Nithya, and Swati. This is probably the funniest bunch of them all.

I spent 3 memorable years at DRDO in Bangalore before pursuing a master's and I was lucky to have great people around me. Most importantly, I would like to thank my mentor Sunita Ma'am for not only believing in me but guiding me at every step of the process. Alka Ma'am was a great friend and a mentor. I also had great friends as colleagues in Prashant, Gajendra, Manjeet, and Shridhar. I had a great group of friends beyond office colleagues who made this phase of my life very memorable. Rahul, Brajesh, Vishal, Kaushal, Ashish, Ajitesh, Anshu, Prateek, and Faizy were so much fun to hang out with. This part would be incomplete without mentioning Deepak and Gaurav, the people who shared my love for food and bike rides.

I have been lucky to have people beyond family who, I believe, are constants in my

life. Priya has not only been a source of constant support and motivation, but she is also the one who has exclusive access to all the aspects of my life. Smriti has always inspired me and supported me during every phase of my life. Another person who deserves a special mention is Srishti, she is one of the most talented people I know. She really cares about making the world a better place and I wish more people shared her zeal. I have known her since my high school days and she has always provided me with consistent support. She is very wise, and I am fortunate to have her nudge me in the right direction. When it comes to long-lasting friends, no one beats Apoorva, my friend since I was 4 (we have been friends for longer than I can remember). Growing up was fun with her around!

Finally, and most importantly, I would like to acknowledge my family. There is no way I can do any justice to this part. Ma and Papa have sacrificed everything to ensure that I have a good life. I feel so loved and that I belong in this world because of them. Sending me to the best school possible no matter the financial toll it took and doing it all with a smile is priceless. They have always understood me and supported me in every way possible. No matter where I get in stuck in my life, I know that they are there to help me out. Their support has helped me navigate life. This dissertation surely would not have existed if it was not for them. I am literally at a loss for words right now. I guess I will just say that their love is the purest form of love I have ever experienced and I am very fortunate to have them. I would also like to thank my Chachi who has always showered me with all her unconditional love. Although I am a single child, growing up with my cousin Ayush, I never felt like a single child. Our connection is a special one where we can talk about absolutely anything under the sun and always have each other's back. He has been a major source of support growing up. I would like to like to thank my Baba and Dadi for being the loving

grandparents that they are.

The person whom I have always looked up to is my late Nana Ji. He inspires me like no one else. He was a person who always stood up for what was right, touched a lot of lives with his wisdom. If I can ever be a fraction of the person that he was, I will consider myself successful.

Robust Estimation of Tree Structured Probabilistic Graphical Models

Ashish Katiyar, Ph.D.
The University of Texas at Austin, 2021

Supervisor: Constantine Caramanis

Undirected probabilistic graphical models or Markov Random Fields (MRFs) are a powerful tool for describing high dimensional distributions using an associated dependency graph $\mathbb{G}$, which encodes the conditional dependencies between random variables. They form the starting point for many efficient estimation and inference algorithms. Thus, learning the graphical model of a collection of random variables from their samples is a fundamental, and very well-studied problem. In this thesis, we study a natural variant of this problem - learning the graph structure when the random variables have independent unknown noise. We investigate this problem for the class of tree structured graphical models.

In the first problem, the task is to estimate tree structured Gaussian graphical models from samples which have additive independent Gaussian noise of unknown variance. The noise in different random variables breaks down the conditional independence relationship. We ask: can the original tree structure be recovered. We prove that this problem is unidentifiable, but show that this unidentifiability is limited to a small class of candidate trees. We further present additional constraints under which the problem is identifiable.

In the second problem, we consider tree structured Ising models. The random variables in Ising models have support on $\{-1, +1\}$. We consider the task of learning Ising

models when the signs of different random variables are flipped independently with possibly unequal, unknown probabilities. We prove that, surprisingly, the same limited unidentifiability results that hold for Gaussian graphical models continue to hold for Ising models.

In the final problem, we study the natural extension of these problems - what happens in the case of graphical models on discrete random variables with larger support size. We show that the setting of support size of 3 or more is richer as the tree 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. We provide an efficient algorithm to recover the tree upto the identifiability. Finally, we conclude with the sample complexity upper and lower bounds capturing the dependence of the number of samples on the underlying parameters.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Markov Random Fields - An Overview

Markov Random Fields (MRFs) provide a useful framework to model 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.

There are three perspectives for the encoded conditional independence relationships that are equivalent:

1. Global Markov Property - Suppose the graph is partitioned into three partitions A, B, C such that B separates A and C. Then, when conditioning on the nodes in B, all the nodes in A are independent of the nodes in C.

2. Local Markov Property - When conditioning on all the nodes a particular node has an edge with, that node is independent of all the remaining nodes in the graph.

3. Pairwise Markov Property - Any two nodes that do not share an edge are independent conditioned on all the remaining nodes.

Example Applications: Probabilistic graphical models have been extensively used in a wide range of applications including image processing ([18, 24, 28, 83]), bioinformatics ([12, 39]), finance ([22, 68]) etc. A special class of graphical models called Ising Models were first introduced in [32] to represent spin systems in quantum physics [10]. Recently, Ising models have also proven quite popular in biology [33], engineering [15, 64], computer vision [61], and also in the optimization and OR communities, including in finance [91], and social networks [51]. The special class of tree-structured Ising models is beneficial for applications in statistical physics over non-amenable graphs. A detailed description and further references can be found in [49].

Data driven application of graphical models can be split into two major components - (i) learning the underlying probabilistic graphical model from the data samples, (ii) performing efficient inference using the learnt graphical model. This dissertation provides novel insights into the first component of learning graphical models from data samples. The second component of efficient inference, while being interesting in its own right, is out of the scope of this dissertation.

1.2 Tree Structured Graphical Models

A special class of graphical model which has garnered a lot of interest is when the underlying graph is a tree (the graph does not contain any cycles). For tree structured graphical models, the joint distribution of all the random variables can be decomposed as a product of pairwise distributions of the random variables that share an edge. Restricting to this subclass of graphical models enables sample efficient learning as governed by the bias

variance trade-off. Furthermore, it is computationally efficient to perform exact inference for tree structured graphical models.

We next understand the implication of tree structured conditional independence on the decomposition of the probability distribution. Let $\mathbf{X} = [X_1, X_2, \ldots X_n]$ be a vector of random variables whose graphical model is a tree T. Since the graphical model is undirected, there does not exist a parent child relationship between the nodes. We arbitrarily select any node X_i as the root node and we define the parent node of any node X_j, denoted by $X_{\pi(j)}$, as the first node in the path from X_j to X_i. Without loss of generality, assume that X_1 is the root node. Then, the probability distribution of $\mathbf{X}$ can be decomposed as follows:

$$P_T(\mathbf{X}) = P_T(X_1) \prod_{j=2}^{n} P_T(X_j | X_{\pi(j)}). \tag{1.1}$$

1.2.1 Chow-Liu Algorithm

In the seminal work [17], the authors provide two key results - (i) The tree structured graphical model that best approximates a high dimensional probability distribution (has minimum KL divergence) is the maximum weight spanning tree where the weights are the mutual information between all the pairs of random variables. Furthermore, the pairwise marginals of all pairs of random variables connected by an edge match those of the high dimensional distribution. (ii) The maximum likelihood estimate of the tree structured graphical model given samples from a probability distribution is given by the maximum weight spanning tree of the empirical pairwise mutual information.

We include the proof here for completeness. Let P be any arbitrary probability distribution and P_T be the probability distribution of a tree T structured graphical model.

The KL-divergence is given as follows:

$$D_{KL}(P, P_T) = \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}) - \sum_{\mathbf{x}} P(\mathbf{x}) \log P_T(\mathbf{x}) \tag{1.2}$$

First note that $\sum_{\mathbf{x}} P(\mathbf{X} = \mathbf{x}) \log P(\mathbf{X} = \mathbf{x})$ is equal for every P_T.

Next, we show that for a given tree T, $D_{KL}(P, P_T)$ is minimized when $P_T(x_j, X_{\pi(j)} = x_{\pi(j)}) = P(X_j = x_j, X_{\pi(j)} = x_{\pi(j)})$. While this is an easy result, we could not find its proof in the literature. Suppose $\tilde{P}_T$ is a probability distribution that has the same graph T but differs on at least one pairwise marginal from P for nodes connected by an edge. Also assume that $P_T(X_j = x_j, X_{\pi(j)} = x_{\pi(j)}) = P(X_j = x_j, X_{\pi(j)} = x_{\pi(j)})$. For the ease of notation, for any probability distribution P, we denote $P(\mathbf{X} = \mathbf{x})$ by $P(\mathbf{x})$, $P(X_i = x_i)$ by $P(x_i)$, $P(X_i = x_i, X_j = x_j)$ by $P(x_i, x_j)$, and $P(X_i = x_i | X_j = x_j)$ by $P(x_i|x_j)$ Then we have that:

$$\begin{aligned}
&\sum_{\mathbf{x}} P(\mathbf{x}) \log P_T(\mathbf{x}) - \sum_{\mathbf{x}} P(\mathbf{x}) \log \tilde{P}_T(\mathbf{x}) \\
=&\sum_{\mathbf{x}} P(\mathbf{x}) \log \frac{P_T(\mathbf{x})}{\tilde{P}_T(\mathbf{x})} \\
=&\sum_{\mathbf{x}} P(\mathbf{x}) \left(\log \frac{P_T(x_1)}{\tilde{P}_T(x_1)} + \sum_{j=2}^{n} \log \frac{P_T(x_j|x_{\pi(j)})}{\tilde{P}_T(x_j|x_{\pi(j)})} \right) \\
=&\sum_{x_1} P(x_1) \log \frac{P_T(x_1)}{\tilde{P}_T(x_1)} + \sum_{j=2}^{n} \sum_{x_j, x_{\pi(j)}} P(x_j, x_{\pi(j)}) \log \frac{P_T(x_j|x_{\pi(j)})}{\tilde{P}_T(x_j|x_{\pi(j)})} \\
=&\sum_{x_1} P_T(x_1) \log \frac{P_T(x_1)}{\tilde{P}_T(x_1)} + \sum_{j=2}^{n} \sum_{x_j, x_{\pi(j)}} P_T(x_j, x_{\pi(j)}) \log \frac{P_T(x_j|x_{\pi(j)})}{\tilde{P}_T(x_j|x_{\pi(j)})} \\
=&\sum_{\mathbf{x}} P_T(\mathbf{x}) \log P_T(\mathbf{x}) - \sum_{\mathbf{x}} P_T(\mathbf{x}) \log \tilde{P}_T(\mathbf{x}) \\
=&D_{KL}(P_T, \tilde{P}_T) > 0 (\text{as} P_T \neq \tilde{P}_T)
\end{aligned}$$

Thus, $\sum_{\mathbf{x}} P(\mathbf{x}) \log P_T(\mathbf{x}) > \sum_{\mathbf{x}} P(\mathbf{x}) \log \tilde{P}_T(\mathbf{x})$. Therefore, $D_{KL}(P, P_T) < D_{KL}(P, \tilde{P}_T)$.

With this insight, let us come back to Equation (1.2).

$$\begin{aligned} D_{KL}(P, P_T) &= \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}) - \sum_{\mathbf{x}} P(\mathbf{x}) \left(\log P_T(x_1) + \sum_{j=2}^{n} \log P_T(x_j | x_{\pi(j)}) \right) \\ &= \sum_{\mathbf{x}} P(\mathbf{x}) \log P(\mathbf{x}) - \sum_{\mathbf{x}} P(\mathbf{x}) \left(\log P(x_1) + \sum_{j=2}^{n} \log \frac{P(x_j, x_{\pi(j)}) P(x_j)}{P(x_j) P(x_{\pi(j)})} + \right) \end{aligned}$$

Now, note that

$$- \sum_{\mathbf{x}} P(\mathbf{x}) \log P(x_1) = - \sum_{x_1} P(x_1) \log P(x_1) = H(X_1).$$

Similarly,

$$- \sum_{\mathbf{x}} P(\mathbf{x}) \sum_{j=2}^{n} \log P_T(x_j) = - \sum_{j=2}^{n} \sum_{x_j} P(x_j) \log P(x_j) = \sum_{j=2}^{n} H(X_j).$$

Also note that

$$\sum_{\mathbf{x}} P(x) \sum_{j=2}^{n} \log \frac{P(x_j, x_{\pi(j)})}{P(x_j) P(x_{\pi(j)})} = \sum_{j=2}^{n} \sum_{x_j, x_{\pi(j)}} P(x_j, x_{\pi(j)}) \log \frac{P(x_j, x_{\pi(j)})}{P(x_j) P(x_{\pi(j)})} = \sum_{j=2}^{n} I(X_j, X_{\pi(j)})$$

Therefore, we get

$$D_{KL}(P, P_T) = -H(\mathbf{X}) + \sum_{j=1}^{n} H(X_j) - \sum_{j=2}^{n} I(X_j, X_{\pi(j)})$$

Since, $-H(\mathbf{X}) + \sum_{j=1}^{n} H(X_j)$ is equal for all P_T, in order to minimize $D_{KL}(P, P_T)$, we need to maximize $\sum_{j=2}^{n} I(X_j, X_{\pi(j)})$. Thus, the optimal tree T is the maximum weight spanning tree with weights being the mutual information of all the pairs of random variables. This concludes the proof of the first result.

Next, we find the maximum likelihood estimate of tree structured graphical model given samples from a probability distribution. Let $\mathbf{x}^1, \mathbf{x}^2 \dots \mathbf{x}^s$ be the samples and P_T be a probability distribution with tree structured graphical model T. Then we have:

$$\begin{aligned} L_{P_T}(\mathbf{x}^1, \mathbf{x}^2 \dots \mathbf{x}^s) &= \prod_{i=1}^{s} P_T(\mathbf{x}^i) \\ &= \prod_{i=1}^{s} \left(P_T(x_1^i) \prod_{j=2}^{n} P_T(x_j^i | x_{\pi(j)}^i) \right). \end{aligned}$$

Thus the log likelihood l_{P_T} is given by:

$$\begin{aligned} l_{P_T}(\mathbf{x}^1, \mathbf{x}^2 \dots \mathbf{x}^s) &= \sum_{i=1}^{s} \left(\log P_T(x_1^i) + \sum_{j=2}^{n} \log P_T(x_j^i | x_{\pi(j)}^i) \right) \\ &= s \left(\sum_{x_1} \hat{P}_T(x_1) \log P_T(x_1) + \sum_{j=2}^{n} \sum_{x_j, x_{\pi(j)}} \hat{P}_T(x_j, x_{\pi(j)}) \log P_T(x_j^i | x_{\pi(j)}^i) \right). \end{aligned}$$

Recall the analysis as the one used to prove that the best tree structured graphical model approximation of a high dimensional distribution has pairwise marginals of the nodes connected by an edge equal to the pairwise marginals of the those nodes in the high dimensional distribution. Using the same argument we can conclude that, for a given tree T, the likelihood is maximized when the pairwise marginals of nodes connected by an edge is equal to the empirical estimate. Thus, we have that:

$$\begin{aligned} &\max_{P_T} l_{P_T}(\mathbf{x}^1, \mathbf{x}^2 \dots \mathbf{x}^s) \\ &= \max_{T} \sum_{i=1}^{s} \left(\log \hat{P}_T(x_1^i) + \sum_{j=2}^{n} \log \hat{P}_T(x_j^i | x_{\pi(j)}^i) \right) \\ &= \max_{T} \sum_{i=1}^{s} \left(\sum_{j=1}^{n} \log \hat{P}_T(x_j^i) + \sum_{j=2}^{n} \log \frac{\hat{P}_T(x_j^i, x_{\pi(j)}^i)}{\hat{P}_T(x_j^i) \hat{P}_T(x_{\pi(j)}^i)} \right). \end{aligned}$$

$$= \max_T s \left(\sum_{j=1}^{n} \sum_{x_j} \hat{P}_T(x_j) \log \hat{P}_T(x_j) + \sum_{j=2}^{n} \sum_{x_j, x_{\pi(j)}} \hat{P}_T(x_j^i, x_{\pi(j)}^i) \log \frac{\hat{P}_T(x_j^i, x_{\pi(j)}^i)}{\hat{P}_T(x_j^i)\hat{P}_T(x_{\pi(j)}^i)} \right).$$

$$= \max_T s \left(\sum_{j=1}^{n} \hat{H}(X_j) + \sum_{j=2}^{n} \hat{I}(X_j, X_{\pi(j)}) \right).$$

Since $\sum_{j=1}^{n} \hat{H}(X_j)$ is equal for every tree, the maximum likelihood tree is the maximum weight spanning tree with the weights being the empirical pairwise mutual information.

Clearly, when the underlying graphical model is tree structured, the Chow Liu algorithm correctly recovers the underlying tree. The sample complexity and error exponents of the Chow-Liu algorithm when the underlying graphical model is tree structured were presented in [8] and [71] respectively.

1.2.2 Effect of noise

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. The problem is further exacerbated by the fact that often the magnitude of the noise in unknown. For critical applications like modeling the gene interaction networks, it is even more important to ensure that the graphical model estimate is robust to the noise in the observations. Thus, it is imperative to understand the impact of noise on the graphical model estimation problem.

Noise in the random variables can break down the conditional independence relationship. For instance, if two random variables X and Z are independent conditioned on Y, we do not expect the noisy versions of these variables to satisfy the same conditional independence relationship even if the noise in the random variables is independent. We understand this with a simple example.

Example: Suppose X, Y and Z have support on $\{0, 1\}$. The data generation process is that X is a fair coin toss, if X takes the value 1, for Y we toss a biased coin whose probability of 1 is 0.99 and if X takes the value 0, for Y we toss a biased coin whose probability of 0 is 0.99. Similarly, if Y is 1, for Z we toss a biased coin whose probability of 1 is 0.99 and if Y is 0, for Z we toss a biased coin whose probability of 0 is 0.99. This is given as follows:

$$P(X = 1) = 0.5$$
$$P(Y = 1|X = 1) = 0.99, P(Y = 1|X = 0) = 0.01$$
$$P(Z = 1|Y = 1) = 0.99, P(Z = 1|Y = 0) = 0.01.$$

It is easy to see that $X \perp Z|Y$. Now let us assume that Y is noisy, that is, the bit Y gets flipped with some probability. It is easy to see that X and Z are no longer independent. Thus, noise in Y breaks down the conditional independence relationship $X \perp Z|Y$.

Therefore, noise in the random variables can introduce new edges in the graphical model, thereby obfuscating the original graph structure. This gives rise to the natural question: Can the original graph be recovered? One approach could be to apply the Chow-Liu algorithm on the noisy observations. Unfortunately, when the nodes are corrupted by noise of unequal magnitude, it can change the order of the pairwise mutual information, thereby, potentially changing the maximum weight spanning tree.

In this dissertation we study three classes of tree structured graphical models - (i) Gaussian Graphical Models, (ii) Ising Models, (iii) Discrete graphical models with support size larger than 2. We uncover novel unidentifiability phenomena for these graphical models.

1.3 Contribution and Organization

Chapter 2: Robust Estimation of Tree-Structured Gaussian Graphical Models In this chapter, we consider the task of learning the underlying tree for Gaussian graphical models when the observations from random variables have independent additive Gaussian noise with unknown variance. In the absence of noise, we can estimate the covariance matrix and it is well-known that the support of the inverse covariance matrix corresponds to the edges of the graphical model. Due to noise, instead of having access to the true covariance matrix Σ, we only have access to the noisy covariance matrix $M = \Sigma + D$, where D is an unknown positive diagonal matrix. We investigate whether is it possible to recover the conditional independence structure (graphical model) of the underlying variables. We prove that it is impossible to recover the original tree, however, it is possible to recover a small equivalence class of trees which contains the original tree. This equivalence class of trees is given by all possible permutations of the nodes within a leaf cluster (a leaf node, its parent, and its siblings form a leaf cluster). The key idea revolves around using the uncorrupted off-diagonal elements of the covariance matrix to make inferences about the graph structure. We also present some side information conditions which can make the problem identifiable.

Chapter 3: Robust Estimation of Tree-Structured Ising Models This chapter is about the robust estimation of Ising models. In this case, the noise is because of the random variables flipping their sign with unknown, possibly unequal probability. We approach this problem by estimating the probability of error for the different random variables which can lead to tree structured graphical models. Interestingly, we arrive at the exact same identifiability results as for Gaussian graphical models.

Chapter 4: Recoverability Landscape of Tree Structured Markov Random Fields under Symmetric Noise Insights from above two problems lead to a natural question: does this property of identifiability upto an equivalence class of trees in the face of independent noise hold for graphical models on generic random variables or is it a special property of Gaussian graphical models and Ising models. We show that 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. We also provide sample complexity lower bounds for the problem. Finally, we demonstrate the efficacy of our algorithm experimentally.

Chapter 2

Robust Estimation of Tree Structured Gaussian Graphical Models

2.1 Introduction

In this chapter, we study the recovery of tree structured Gaussian Graphical Models from noisy samples. For jointly Gaussian random variables, the graphical model is given by the non-zeros in the inverse of the covariance matrix, also known as the precision matrix. We ask a natural variant of this fundamental problem: suppose we observe the random variables with independent additive noise. Thus, in the infinite sample limit, rather than knowing the covariance matrix, Σ, we have access only to $M = \Sigma + D$, the sum of the covariance matrix and a diagonal matrix. In general, $(\Sigma + D)^{-1}$ does not share the sparsity structure of Σ^{-1}. In the language of probability, if two random variables X and Y are independent conditioned on Z, then we do not expect that $(X + W_1)$ and $(Y + W_2)$ are independent when conditioned on $(Z + W_3)$, even when W_1, W_2 and W_3 are independent.

Parts of this chapter are available at: Katiyar, Ashish, Jessica Hoffmann, and Constantine Caramanis."Robust estimation of tree structured Gaussian graphical models." In International Conference on Machine Learning, pp. 3292-3300. PMLR, 2019. The author formulated the problem, performed the theoretical analysis and contributed in writing the paper.

We ask: when is it possible to recover the conditional independence structure (graphical model) of the underlying variables, i.e., when can we recover the sparsity pattern of Σ^{-1}? Despite the voluminous literature on Gaussian graphical models, to the best of our knowledge, there has been no answer to this question.

Contributions of this paper. We show the following:

- A negative result of unidentifiability (Theorem 2.4.1): Even for a simple Markov chain on three nodes, the problem is unidentifiable even when an arbitrarily small amount of independent noise is added. That is, there are covariance matrices that differ only on their diagonal entries, and yet whose inverses have different sparsity patterns.
- A positive result of limited unidentifiability (Theorem 2.4.2): While unidentifiable, even for large independent noise, the ambiguity is highly limited. Specifically, we show that for tree-structured graphical models, distinguishing leaves from their immediate neighbors is impossible, but the remaining structure of the graph is identifiable (see Figure 2.1 for an illustration).
- Identifiability with Side Information:
  - (Theorem 2.4.3) We characterize an upper bound on the noise which, if given as side information, makes the problem identifiable.
  - (Theorem 2.4.4) If there is side information that in the precision matrix, for a leaf node, the diagonal entry is greater than the absolute value of the other non-zero entry, the problem is identifiable.
  - (Theorems 2.4.5, 2.4.7) Given a lower bound on the minimum eigenvalue of the true covariance matrix as side information, we characterize the upper bound on

the noise for which the problem is identifiable. We also characterize a lower bound on the noise which makes the problem unidentifiable.

2.2 Related Work

Estimating Gaussian graphical models has been a very widely explored topic. Various algorithms based on the ℓ^1 penalized log likelihood maximization have been used in, e.g., [2, 59, 23, 89, 62]. A parameter free Bayesian approach was presented in [82]. In [50] and [88], another approach was proposed which finds conditional independence relations by regression using one random variable as output and the remaining random variables as input. The output variable is conditionally independent of the input variables with regression coefficient zero.

The Chow-Liu algorithm of [17] (Section 1.2.1) is the most popular algorithm for learning tree structured graphical models. However, as discussed in Section 1.2.2, in the presence of unequal noise, it can converge to an incorrect tree.

There has been research about learning tree structured graphical models with latent variables ([16, 58, 13]). One could cast our problem as the problem of learning latent tree graphical model with the leaf nodes being the noisy random variables we observe and the latent nodes being the true underlying random variables. However, algorithms learning latent tree graphical model focus on minimal tree extensions which assume that all the latent nodes have degree greater than 2. This assumption makes these algorithms inapplicable in our setting as the leaf nodes of the original tree have degree 2 when considering graphical models containing both- the non-noisy nodes and the noisy nodes.

Robust estimation of graphical models has been extensively studies in [46, 87, 79, 37, 48, 78, 45]. However, the robustness is against outliers or missing data or Gaussian noise with known covariance or bounded noise. To the best of our knowledge, there is no work that addresses the natural setting of (unknown) additive independent Gaussian noise. This is precisely the setting that we tackle in this paper. In [90] the authors address the problem of measurement error in the directed graphical models setting. These results do not extend to the setting of undirected graphical models.

The algorithm in [34] comes closest to our setting, and in fact is complementary. In that work, the goal is to recover the graph structure in the presence of corruption in those off-diagonal terms of the covariance matrix which are not conditionally independent. Specifically, the results there do not consider (and cannot address) noise in the diagonal elements. Thus, this setting considers a perfectly complementary setting, as in this work there is noise only in the diagonal elements of the covariance matrix and not in the off diagonal elements. It would be interesting to consider if these results can be merged to obtain a general result.

2.3 Problem Statement

Let $X = [X_1, X_2 \ldots, X_n]^T$ denote a vector of jointly Gaussian random variables whose conditional independence structure is given by a tree. We call this the *true tree* T^*. We denote the covariance matrix of X by Σ^* and the precision matrix by Ω^*. That is, $X \sim \mathcal{N}(0, \Sigma^*)$. We denote the noise covariance matrix by D^*. This is a non-negative diagonal matrix. We

denote the observed noisy covariance matrix by:

$$\Sigma^o = \Sigma^* + D^*.$$

Given Σ^o as an input, recovering Σ^* exactly is never possible. Consider, for instance, independent noise added only to a leaf node. Instead, we would like to recover the underlying tree T^*. We show that in general, recovering T^* exactly is not possible. However, we show that the ambiguity is limited. We characterize this explicitly. That is, we characterize the set of possible trees T' that correspond to a covariance matrix, Σ', and a nonnegative diagonal matrix D' such that $\Sigma^o = \Sigma' + D'$.

Notation: For any matrix Σ, $(\Sigma)^T$ represents the transpose of the matrix. Σ_{ij} denotes the element at the i, j position. $\Sigma_{:,i}$ represents the i^{th} column. $\Sigma_{-i,-j}$ represents the submatrix after deleting row i and column j from Σ. $\Sigma_{-i,j}$ represents the j^{th} column without the i^{th} element. Similarly, $\Sigma_{i,-j}$ represents the i^{th} row without the j^{th} element. We use $\det(\Sigma)$ to represent the determinant of the matrix. For a random vector $X = [X_1, X_2, \ldots, X_n]^T$, X_i denotes the i^{th} component and X_{-i} denotes the subvector after removing the i^{th} component.

2.4 Identifiability Result

Let the set of all the leaf nodes of T^* be $\mathcal{L}$:

$$\mathcal{L} = \{a \mid \text{node } a \text{ is a leaf node in } T^*\}.$$

Consider all the subsets of $\mathcal{L}$ such that no two nodes in the subset share a common neighbor. Let p be the number of such subsets. Let $\mathcal{S}^q$ be the q^{th} subset. Let T^q be the tree obtained

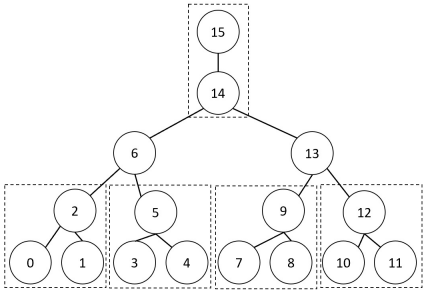


Figure 2.1: For this T^*, $\mathcal{T}_{T^*}$ is the set of all the trees obtained by permuting the nodes within each of the dotted regions. We prove that while T^* is unidentifiable, under our noise model, we can recover $\mathcal{T}_{T^*}$. In other words, the tree structure is recoverable up to permutation of leaves with their neighbors.

by exchanging the position of nodes in $\mathcal{S}^q$ with their neighbor node in T^*. Therefore, for every tree T^q, there is a corresponding set $\mathcal{S}^q$.

Definition 2.4.1. For any tree T^*, we define the equivalence class of tree $\mathcal{T}_{T^*}$ as follows:

$$\mathcal{T}_{T^*} = \{T^q \mid q \in \{1, 2, \dots p\}\}.$$

Figure 2.1 gives an example of $\mathcal{T}_{T^*}$.

2.4.1 Identifiability Results without Side Information

Theorem 2.4.1. *(Negative Result - Unidentifiability) Consider a covariance matrix Σ^* whose independence structure is given by the tree T^*. Suppose we are given a noisy covariance matrix $\Sigma^o = \Sigma^* + D^*$ where $D^*_{ii} > 0$ when i is a neighbor of a leaf node. For any tree $\tilde{T} \in \mathcal{T}_{T^*}$, it is always possible to decompose $\Sigma^o = \tilde{\Sigma} + \tilde{D}$ where the conditional independence for $\tilde{\Sigma}$ is given by the tree $\tilde{T}$ and $\tilde{D}$ is a non-negative diagonal matrix.*

Proof Outline. We give an explicit construction that demonstrates that any tree $\tilde{T} \in \mathfrak{T}_{T^*}$ is achievable. Consider any tree $\tilde{T} \in \mathfrak{T}_{T^*}$ and its corresponding leaf subset $\tilde{\mathcal{S}}$. The required decomposition of $\Sigma^o = \tilde{\Sigma} + \tilde{D}$ is given as follows:

$$\tilde{\Sigma}_{ij} = \begin{cases} \Sigma^*_{ij} - \frac{1}{\Omega^*_{ij}} & \text{if } i = j \in \tilde{\mathcal{S}} \\ \Sigma^*_{ij} + c_1^i & \text{if } i = j \in Neighbor(\tilde{\mathcal{S}}) \\ \Sigma^*_{ij} & \text{otherwise,} \end{cases} \tag{2.1}$$

where $Neighbor(\tilde{\mathcal{S}})$ is the set of neighbor nodes of all the nodes in $\tilde{\mathcal{S}}$. Also, c_1^i is chosen such that $0 < c_1^i \leq D^*_{ii}$.

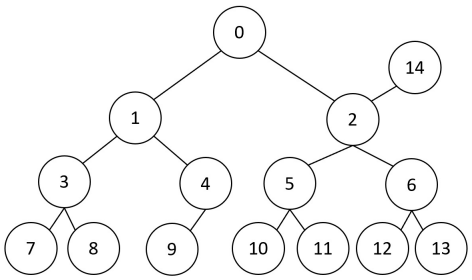
$$\tilde{D}_{ii} = \begin{cases} D^*_{ii} + \frac{1}{\Omega^*_{ii}} & \text{if } i \in \tilde{\mathcal{S}} \\ D^*_{ii} - c_1^i & \text{if } i \in Neighbor(\tilde{\mathcal{S}}) \\ D^*_{ii} & \text{otherwise.} \end{cases} \tag{2.2}$$

The full proof which includes arriving at this decomposition and showing that the conditional independence structure of $\tilde{\Sigma}$ is given by $\tilde{T}$ is in Appendix A.

Theorem 2.4.2. *(Positive Result - Limit on unidentifiability) Consider any decomposition $\Sigma^o = \Sigma' + D'$ such that the conditional independence for Σ' is given by a tree T' and D' is a non-negative diagonal matrix. Then $T' \in \mathfrak{T}_{T^*}$. Equations 2.1 and 2.2 provide a decomposition that results in this T'.*

Proof Outline. The proof of the theorem relies on showing that the off-diagonal terms of the covariance matrix suffice to specify the structure of the underlying tree up to the equivalence set $\mathfrak{T}_{T^*}$. Our proof is constructive, and hence can be considered as a proto- or conceptual- algorithm for recovering $\mathfrak{T}_{T^*}$.

The main building block of this proof is to categorize any set of 4 nodes as a *star-shape* or a *non-star-shape* (we define this below). Moreover, if it is a non star shape, we show



Node Set	Classification	i_k, pairing (If Applicable)
{0, 1, 3, 7}	Non-Star	1, {{0,1}, {3,7}}
{7, 8, 9, 10}	Non-Star	1, {{7,8}, {9,10}}
{14, 2, 10, 11}	Non-Star	2, {{14,2}, {10,11}}
{7, 9, 1, 6}	Star	N.A.
{1, 14, 10, 6}	Star	N.A.

Figure 2.2: Examples of classification of 4 nodes as star shape or non star shape. If they form a non star shape, the nodes are grouped in pairs of 2.

that it is always possible to partition the four nodes into two pairs that each lie in separate connected components of the tree.

Definition 2.4.2.

- Four nodes $\{i_1, i_2, i_3, i_4\}$ form a **non-star shape** if there exists a node i_k in the tree T^*[1] such that exactly two nodes among the four lie in the same connected component of $T^* \setminus i_k$.
- If $\{i_1, i_2, i_3, i_4\}$ do not form a non-star shape, we say they form a **star shape**.

It is easy to see that in the event that a set of 4 nodes forms a non star, there exists a grouping such that the 2 nodes in the same connected component form the first pair and the other 2 nodes form the second pair. Figure 2.2 gives examples of star shape and non star shape. This categorization is done using only the off-diagonal elements of the covariance matrix, hence this property remains invariant to diagonal perturbations, that is, every set of 4 nodes falls in the same category in any tree obtained from the decomposition of $\Sigma^o = \Sigma' + D'$ as $\Sigma'_{ij} = \Sigma^*_{ij}\ \forall\ i \neq j$. The proof of this theorem is split in 3 parts:

[1]Note that nothing prevents i_k to be one of the four nodes.

(i) Prove that it is possible to categorize any set of 4 nodes as star shape or non star shape using only off diagonal elements of the covariance matrix. Moreover, if the 4 nodes have a non star shape, we can find their grouping in two halves.

(ii) Prove that this categorization of all the possible sets of 4 nodes completely defines all the possible partitions of the original tree in 2 connected components such that the connected components have at least 2 nodes.

(iii) Prove that these partitions of a tree into connected components completely define the tree structure up to the equivalence set $\mathcal{T}_{T^*}$.

For part (i), we prove that a set of 4 nodes $\{i_1, i_2, i_3, i_4\}$ forms a non star shape such that nodes i_1 and i_2 form one pair and i_3 and i_4 form the second pair if and only if:

$$\begin{aligned} \frac{\Sigma^*_{i_1 i_3}}{\Sigma^*_{i_1 i_4}} &= \frac{\Sigma^*_{i_2 i_3}}{\Sigma^*_{i_2 i_4}}, \\ \frac{\Sigma^*_{i_2 i_1}}{\Sigma^*_{i_3 i_1}} &\neq \frac{\Sigma^*_{i_2 i_4}}{\Sigma^*_{i_3 i_4}}. \end{aligned} \tag{2.3}$$

We also prove that a set of 4 nodes $\{i_1, i_2, i_3, i_4\}$ forms a star if and only if:

$$\begin{aligned} \frac{\Sigma^*_{i_1 i_3}}{\Sigma^*_{i_1 i_4}} &= \frac{\Sigma^*_{i_2 i_3}}{\Sigma^*_{i_2 i_4}}, \\ \frac{\Sigma^*_{i_2 i_1}}{\Sigma^*_{i_3 i_1}} &= \frac{\Sigma^*_{i_2 i_4}}{\Sigma^*_{i_3 i_4}}. \end{aligned} \tag{2.4}$$

For part (ii), we first define a subtree.

Definition 2.4.3. Let $\mathcal{A}$ denote the set of all the nodes in T^*. A **subtree** $\mathcal{B}$ of a tree T^* is a set of nodes such that $\mathcal{B}$ and $\mathcal{A} \setminus \mathcal{B}$ both form connected components in T^*. The pair of subtrees $\mathcal{B}$ and $\mathcal{A} \setminus \mathcal{B}$ are called **complementary subtrees**.

We prove that if we start with a set of nodes $\{i_1, i_2, i_3, i_4\}$ that form a non star such that nodes i_1 and i_2 form a pair, we can get a partition of T^* into the smallest subtree containing i_1 and i_2 and the remaining tree. This is done using the function SMALLEST-SUBTREE$(\Sigma^o, \{i_1, i_2, i_3, i_4\})$, the details of which are provided in Appendix B.2. Upon doing this for different initializations, we get all the possible partitions of the tree such that each partition has at least 2 nodes.

For part (iii) we define equivalence clusters and edges between equivalence clusters as follows:

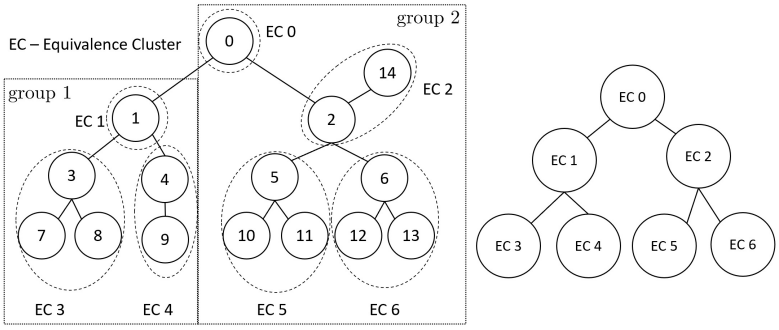
Definition 2.4.4. A set containing an internal node and all the leaf nodes connected to it forms an **equivalence cluster**. We say that there is an edge between two equivalence clusters if there is an edge between any node in one equivalence cluster and any node in the other equivalence cluster.

The subtrees obtained from part (ii) completely specify the equivalence clusters and the edges between the equivalence clusters. This gives us the set $\mathcal{T}_{T^*}$. Partitioning in part (ii) and equivalence clusters in part (iii) are illustrated in Figure 2.3. The detailed proof of each part is presented in Appendix B.

2.4.2 Identifiability Results with Side Information

Theorem 2.4.3. *(Maximum Noise Identifiability Condition) Suppose the noise is upper bounded by*

$$D^*_{aa} < \frac{1}{\Omega^*_{aa}}, \ \forall \ a \in \mathcal{L} \tag{2.5}$$



(a) (b)

Figure 2.3: (a) Suppose $\{i_1, i_2, i_3, i_4\} = \{7, 9, 5, 2\}$, part (ii) partitions the nodes in group 1 and group 2. All the equivalence clusters are also shown. (b) Edges between equivalence clusters.

and suppose that this upper bound is known as side information. In this case, the decomposition of $\Sigma^o = \Sigma' + D'$ *results in* Σ' *whose independence structure is given by* T^**.*

Proof. From Equation 2.2, for a leaf node a to exchange position with its neighbor, we need:

$$D'_{aa} \geq \frac{1}{\Omega^*_{aa}}.$$

The constraint in Equation 2.5 makes this solution infeasible. Hence any feasible solution cannot have a leaf node exchanged with its neighbor. □

Theorem 2.4.4. *(Leaf Diagonal Majorization Identifiability Condition) Suppose* Ω^* *satisfies the condition that for any leaf node* a *and its neighbor node* b *in* T^**,* $\Omega^*_{aa} > |\Omega^*_{ab}|$*. Then for any decomposition of* $\Sigma^o = \Sigma' + D'$ *which satisfies the same property, the tree structure of* Σ' *is the same as that of* Σ^**, that is,* $T' = T^*$*.*

Proof Outline. To prove this claim, we consider the decomposition of $\Sigma^o = \Sigma' + D'$ such that the conditional independence structure T' for Σ' has leaf node b and its neighbor

node a. We show that $\Omega'_{bb} < |\Omega'_{ab}|$, that is, the leaf node b in T' violates the constraint. Hence, any decomposition of Σ^o which results in an exchange of a leaf node with its neighbor is infeasible. Hence the problem becomes identifiable.

Relabeling if necessary, assume that node n is a leaf node connected to node $n-1$ in T^*. From Equation 2.1, the decomposition of $\Sigma^o = \Sigma' + D'$ to obtain a tree structure T' in which node $n-1$ is a leaf node connected to node n is given by:

$$\Sigma'_{ij} = \begin{cases} \Sigma^*_{ij} - \frac{1}{\Omega^*_{ij}} & \text{if } i = j = n \\ \Sigma^*_{ij} + c_1^i & 0 < c_1^i < D^*_{n-1n-1} \text{ if } i = j = n-1 \\ \Sigma^*_{ij} & \text{otherwise.} \end{cases}$$

We derive the expression of $\Omega' = (\Sigma')^{-1}$. We denote B^1 and B^2 as follows:

$$B^1_{ij} = \begin{cases} c_1^i & 0 < c_1^i < D^*_{n-1n-1} \text{ if } i = j = n-1 \\ 0 & \text{otherwise} \end{cases},$$

$$B^2_{ij} = \begin{cases} -\frac{1}{\Omega^*_{nn}} & \text{if } i = j = n \\ 0 & \text{otherwise} \end{cases}.$$

This gives us $\Sigma' = \Sigma^* + B^1 + B^2$. The calculation of $\Omega' = (\Sigma')^{-1}$ is presented in Appendix C. At positions $(n-1, n-1)$ and $(n-1, n)$ of Ω', we get:

$$\Omega'_{n-1n-1} = \frac{1}{c_1^{n-1}},$$
$$\Omega'_{n-1n} = \frac{\Omega^*_{nn}}{c_1^{n-1}\Omega^*_{n-1n}}.$$

By the original assumption we have $\Omega^*_{nn} > |\Omega^*_{n-1n}|$, hence $\Omega'_{n-1n-1} < |\Omega'_{n-1n}|$. Therefore any exchange of leaf node with its neighbor gives an infeasible solution.

Theorem 2.4.5. *(Minimum Eigenvalue Identifiability Condition) Suppose that a lower bound on the minimum eigenvalue* $\lambda_{\min}$ *of* Σ^* *is such that for every neighbor node* b *of a leaf node* a *in* T^*, $D^*_{bb} < \lambda_{min}$. *Then for any decomposition of* $\Sigma^o = \Sigma' + D'$ *such that the minimum eigenvalue of* Σ' *is at least* λ_{min}, *the tree structure of* Σ' *is the same as that of* Σ^*, *i.e.,* $T' = T^*$.

Corollary 2.4.6. *If the smallest eigenvalue of* Σ^* *is larger than every element of the diagonal noise matrix* D^*, and we know that this fact holds as side information, *then* T^* *is identifiable.*

Proof. Relabeling if necessary, assume that node n is a leaf node and node $n-1$ is its neighbor in T^*. We again consider the decomposition of $\Sigma^o = \Sigma' + D'$ such that the conditional independence structure T' for Σ' has leaf node $n-1$ and its neighbor node n. In order to prove this theorem we first consider an intermediate matrix Σ^I:

$$\Sigma^I = \Sigma^* + B^2.$$

Σ^I has minimum eigenvalue 0 (This is proved in the Appendix A during the proof of Theorem 2.4.1). Σ' is obtained as follows:

$$\Sigma' = \Sigma^I + B^1.$$

We denote the minimum eigenvalue of Σ' by λ'_{min} and Σ^I by λ^I_{min}. Using a standard result in matrix perturbation theory for symmetric matrices [67] we have:

$$\begin{aligned}\lambda'_{min} &\leq \lambda^I_{min} + c_1^{n-1} \\ &= c_1^{n-1} \\ &\leq D^*_{n-1n-1}.\end{aligned}$$

If $D^*_{n-1n-1} < \lambda_{min}$ then $\lambda'_{min} < \lambda_{min}$ making this decomposition infeasible. Hence any decomposition resulting in the exchange of a leaf node a with its neighbor b is infeasible if $D^*_{bb} < \lambda_{min}$. □

Theorem 2.4.5 gives a sufficient condition on the noise for identifiability if the minimum eigenvalue is lower bounded. Next, we present a sufficient condition for unidentifiability in the same setting.

Before the theorem statement, we define the following quantities for any pair of a leaf node a and its neighbor b in T^*:

$$\begin{aligned} e^{ab} &= 1 + \frac{\Omega^*_{aa}}{|\Omega^*_{ab}|}, \\ f^{ab} &= \frac{(\Omega^*_{aa})^2}{(\Omega^*_{ab})^2} + \frac{\Omega^*_{aa}}{|\Omega^*_{ab}|}, \\ g^{ab} &= \frac{\Omega^*_{aa}(\Omega^*_{aa}\Omega^*_{bb} - (\Omega^*_{ab})^2)}{(\Omega^*_{ab})^2} + \sum_{\substack{j=1 \\ j\neq a,b}}^{n} \frac{\Omega^*_{aa}|\Omega^*_{bj}|}{|\Omega^*_{ab}|}, \\ h^{ab} &= \max_{\substack{i=1\ldots n \\ i\neq a,b}} \Big(\sum_{\substack{j=1 \\ j\neq a,b}}^{n} |\Omega^*_{ij}| + \frac{\Omega^*_{aa}|\Omega^*_{bi}|}{|\Omega^*_{ab}|} \Big). \end{aligned} \tag{2.6}$$

Theorem 2.4.7. *(Minimum Eigenvalue Unidentifiability Condition) Suppose that a lower bound on the minimum eigenvalue of Σ^* is λ_{min}. If for any decomposition of $\Sigma^o = \Sigma' + D'$, the same constraint holds, the problem will be unidentifiable if, for a leaf node a and its neighbor b, the noise in node b is lower bounded as follows:*

$$D^*_{bb} \geq \begin{cases} e^{ab}\lambda_{min} & \text{if } \lambda_{min} \leq \frac{(e^{ab}-f^{ab})}{e^{ab}g^{ab}}, \\ \frac{f^{ab}}{1/\lambda_{min}-g^{ab}} & \text{if } \frac{(e^{ab}-f^{ab})}{e^{ab}g^{ab}} < \lambda_{min} < \frac{1}{g^{ab}}, \frac{1}{h^{ab}}. \end{cases}$$

If this holds, there exists a feasible Σ' with conditional independence structure T' which has node b as a leaf node and node a as its neighbor.

Proof Outline. Suppose Σ' has node b as leaf node and node a as its neighbor and the rest of the structure is the same as T^*. We provide a lower bound on the minimum eigenvalue of Σ' by upper bounding the maximum eigenvalue of Ω' using Gerschgorin's Theorem [67]. The details are provided in Appendix D.

Note that a lower bound on the noise for unidentifiability can be given only below a threshold of λ_{min}. If λ_{min} is above this threshold, we cannot draw a conclusion about identifiability using this theorem.

2.5 Examples and Illustrations

In this section we provide an example to illustrate the theorem statements.

Consider a Markov Chain (MC) on 4 nodes whose covariance matrix is given as follows:

$$\Sigma^* = \begin{bmatrix} 1.1508 & -0.1885 & 0.0548 & -0.0069 \\ -0.1885 & 0.2356 & -0.0686 & 0.0086 \\ 0.0548 & -0.0686 & 0.7472 & -0.0934 \\ -0.0069 & 0.0086 & -0.0934 & 0.1367 \end{bmatrix},$$

Then its precision matrix is:

$$\Omega^* = \begin{bmatrix} 1 & 0.8 & 0 & 0 \\ 0.8 & 5 & 0.4 & 0 \\ 0 & 0.4 & 1.5 & 1 \\ 0 & 0 & 1 & 8 \end{bmatrix}.$$

and T^* is given in Figure 2.4(a). Let the noise matrix be:

$$D^* = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}.$$

We have $\Sigma^o = \Sigma^* + D^*$.

2.5.1 Example for Theorem 2.4.1

By Theorem 2.4.1, there exists a decomposition of $\Sigma^o = \Sigma' + D'$ such that the conditional independence structure of Σ' is given by a tree T' with node 2 as a leaf node. A possible decomposition is as follows:

$$\begin{aligned} \Sigma' &= \begin{bmatrix} 0.1508 & -0.1885 & 0.0548 & -0.0069 \\ -0.1885 & 10.2356 & -0.0686 & 0.0086 \\ 0.0548 & -0.0686 & 0.7472 & -0.0934 \\ -0.0069 & 0.0086 & -0.0934 & 0.1367 \end{bmatrix}, \\ D' &= \begin{bmatrix} 1.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.1 \end{bmatrix}. \end{aligned} \tag{2.7}$$

The precision matrix Ω' is then:

$$\Omega' = \begin{bmatrix} 6.9687 & 0.1250 & -0.5 & 0 \\ 0.1250 & 0.1 & 0 & 0 \\ -0.5 & 0 & 1.5 & 1 \\ 0 & 0 & 1 & 8 \end{bmatrix}. \tag{2.8}$$

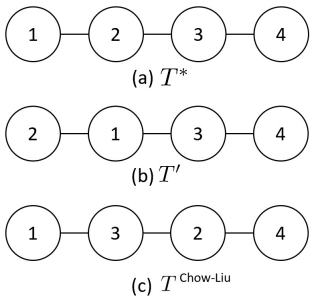


Figure 2.4: (a) T^* is a Markov Chain on 4 nodes. (b) T' is an element of $\mathcal{T}_{T^*}$, thus $\exists \Sigma', D'$ such that $\Sigma^o = \Sigma' + D'$, D' is diagonal with non-negative entries and the conditional independence structure of Σ' is given by T'. (c) Running the Chow-Liu algorithm on the Σ^o gives a tree which is not in $\mathcal{T}_{T^*}$, hence it gives an infeasible solution.

Thus, in the conditional independence structure of Σ', node 2 is a leaf node attached to node 1 as shown in Figure 2.4(b).

Chow-Liu. We now note that running the Chow-Liu algorithm on Σ^o gives a MC as shown in Figure 2.4(c). This tree does not belong to $\mathcal{T}_{T^*}$. This is an example of how the Chow-Liu algorithm can give an infeasible solution.

2.5.2 Example of Theorem 2.4.3

The noise matrix D^* satisfies the condition of Theorem 2.4.3:

$$D^*_{11} < \frac{1}{\Omega^*_{11}}, D^*_{44} < \frac{1}{\Omega^*_{44}}.$$

Hence by the theorem statement, with side information that $D'_{11} < 1$, the decomposition in Equation 2.7 is no longer feasible. Similarly a decomposition with node 3 as a leaf node is also not feasible. Hence the only feasible solutions have the same structure as T^* and the problem is identifiable.

2.5.3 Example of Theorem 2.4.4

Ω^* satisfies the condition of Theorem 2.4.4, that is, for leaf nodes 1 and 4:

$$\Omega^*_{11} > |\Omega^*_{12}|, \Omega^*_{44} > |\Omega^*_{34}|.$$

In the presence of side information that for any leaf node b connected to node a in T', $\Omega'_{bb} > |\Omega'_{ab}|$, the decomposition in Equation 2.7 becomes infeasible as $\Omega'_{22} < |\Omega'_{12}|$. Similarly, exchanging nodes 3 and 4 also results in an infeasible Σ'. Hence the problem becomes identifiable with this side information.

2.5.4 Example of Theorem 2.4.5.

A lower bound on the minimum eigenvalue of Σ^* is $\lambda_{min} = 0.6$. The noise in node 2 does not satisfy the condition of Theorem 2.4.5, that is:

$$D^*_{22} > \lambda_{min}.$$

Therefore, we cannot say anything about the feasibility of the decomposition when node 2 becomes a leaf node connected to node 1. However, the condition of Theorem 2.4.5 is satisfied by node 3, that is:

$$D^*_{33} < \lambda_{min}.$$

Therefore any decomposition which results in node 3 becoming a leaf node violates the minimum eigenvalue constraint (if Σ' were such that node 3 were a leaf node, the minimum eigenvalue of Σ' could at most be $0.0046 < \lambda_{min}$).

2.5.5 Example of Theorem 2.4.7

In order to illustrate Theorem 2.4.7, we consider leaf node 1 and its neighbor node 2. The values $e^{12}, f^{12}, g^{12}, h^{12}$ for the current example are:

$$e^{12} = 2.25, f^{12} = 2.8125, g^{12} = 7.3125, h^{12} = 9.$$

If $\lambda_{min} = 0.6$, we cannot draw a conclusion about the identifiability of the problem using Theorem 2.4.7 as $\lambda_{min} > 1/h^{12}$. If instead $\lambda_{min} = 0.1$, it satisfies $\lambda_{min} < 1/h^{12}, 1/g^{12}$. Hence we can arrive at a lower bound on the noise for unidentifiability using Theorem 2.4.5 which is given as follows:

$$D_{22}^{*} > 1.0465.$$

Chapter 3

Robust Estimation of Tree Structured Ising Models

3.1 Introduction

In this paper, we explore the problem of learning the underlying graph of tree-structured Ising models with independent, unknown, unequal error probabilities. In 2011, [14] highlighted the importance of robustness in Ising models. Recent works in [26, 27, 44] have tried to address this problem. However, they assume the side information of the error probability, which is mostly unavailable and difficult to estimate in most practical settings. In the closely related work for tree-structured Ising models, [53, 55] address this problem as they build on the Chow-Liu algorithm of [17]. In [53], they consider the simplified case where each node has an equal probability of error and [55] assumes that the error doesn't alter the order of mutual information. Both assumptions imply that asymptotically, Chow-Liu converges to the correct tree. However, these assumptions don't arise naturally and are difficult to check from access to only noisy data. To the best of our knowledge, there doesn't exist an analysis of what happens beyond this limiting assumption of order preservation of

Parts of this chapter are available at: Katiyar, Ashish, Vatsal Shah, and Constantine Caramanis. "Robust estimation of tree structured Ising models." arXiv preprint arXiv:2006.05601 (2020). The author formulated the problem, performed the theoretical analysis and contributed in writing the paper.

mutual information.

In fact, section 5.1 of [9] provides an example of the unidentifiability of the problem for a graph on 3 nodes and says that the problem is ill-defined. We reconsider this problem, and show that for the special class of tree structured Ising models, although the problem is not identifiable, nevertheless the unidentifiability is limited to an equivalence class of trees. Thus, more appropriately, one can cast the problem of learning in the presence of unknown, unequal noise as the problem of learning this equivalence class.

Key Contributions

1. We show that the problem of learning tree structured Ising models when the observations flip with independent, unknown, possibly unequal probability is unidentifiable (Theorem 3.3.8).

2. The unidentifiability is restricted to the equivalence class of trees obtained by permuting within the leaf nodes and their neighbors (Theorem 3.3.4).

While we also developed an algorithm to recover the equivalence class of trees from the noisy samples and performed the sample complexity analysis for the same, we do not include it in this chapter as the algorithm presented in Chapter 4 is also applicable in the case of Ising models and outperforms this algorithm.

3.2 Related Work

Efficient algorithms for structure learning of Ising models can be divided into three main categories based on their assumptions: i) special graph structures [1, 17, 19, 66, 8], ii) nature of interaction between variables such as correlation decay property (CDP) [6, 7, 9, 42, 60], iii) bounded degree/width [5, 20, 36, 47, 85, 76]. However, these algorithms assume access to uncorrupted samples.

In the last decade, there has been a lot of research on robust estimation of graphical models [37, 45, 46, 79, 87]. However, extending the above frameworks to the robust structure learning of Ising models remains a challenge. [26, 27, 44] have tried to solve the problem of robust estimation of general Ising models under the assumption of access to the probability of error for each node. Recently, [53, 55] proposed algorithms to estimate the underlying graph structure of tree-structured Ising models in the presence of noise under the strong assumption that the probability of error does not alter the order of mutual information order for the tree. Both these assumptions are restrictive and impractical. In this paper, we present the first algorithm that can robustly recover the underlying tree structured Ising model (upto an equivalence class) in the presence of corruption via unknown, unequal, independent noise.

3.3 Identifiability Result

Problem Setup: Let $\mathbf{X} = [X_1^*, X_2^* \dots X_n^*]$ be a vector of random variables with support on $\{-1, 1\}$. Suppose the conditional independence structure of the variables of $\mathbf{X}$ is given by a tree T^*. This implies that the distribution of $\mathbf{X}$ can be represented by an Ising model. In our model, we have observations where each X_i^* flips with probability q_i. We denote

the probability of error by the vector $\mathbf{q} = [q_1, q_2, \ldots q_n]$ and the noisy random variables by $\mathbf{X'} = [X'_1, X'_2 \ldots X'_n]$. The error in X^*_i disrupts the tree structured conditional independence and the graphical model of $\mathbf{X'}$ is a complete graph if $q_i > 0 \; \forall i \in [n]$. In fact, $\mathbf{X'}$ need not be an Ising model. Given samples of $\mathbf{X'}$, we want to find the tree structure T^*. T

Model Assumptions

Assumption 3.3.1. *(Bounded Mean) The absolute value of the mean -* $|\mathbb{E}[X_i]| \leq \mu_{max} < 1$ $\forall i \in [n]$.

Assumption 3.3.2. *(Bounded Correlation) Correlation* $\rho_{i,j}$ *of any two nodes* X_i *and* X_j *connected by an edge -* $\rho_{min} \leq |\rho_{i,j}| \leq \rho_{max}$ *where* $0 < \rho_{min} \leq \rho_{max} < 1$.

Assumption 3.3.3. *(Bounded error probability) The error probability -* $0 \leq q_i \leq q_{max} < 0.5$ $\forall i \in [n]$.

These assumptions arise naturally. *Assumption 3.3.1* ensures that no variable approaches a constant and hence gets disconnected from the tree. The lower bound in *Assumption 3.3.2* also ensures that every node is connected. The upper bound in *Assumption 3.3.2* ensures that no two nodes are duplicated. *Assumption 3.3.3* ensures the noisy node doesn't become independent of every other node due to the error.

Limited unidentifiability of the problem

In Theorem 3.3.4, we prove that it is possible to recover $\mathcal{T}_{T^*}$ (as defined in 2.4.1) from the samples of $\mathbf{X'}$. Further, we prove that given the distribution of $\mathbf{X'}$, there exists an Ising

model for each tree in $\mathcal{T}_{T^*}$ such that, for some noise vector, its noisy distribution is the same as that of $\mathbf{X}'$ in Theorem 3.3.8

Theorem 3.3.4. *Suppose $\tilde{\mathbf{X}}$ and $\mathbf{X}$ are binary valued random variables satisfying assumption 3.3.1 whose conditional independence is given by trees T' and T^* respectively satisfying assumption 3.3.2. Assume that each node in both these distributions T' and T^* is allowed to be flipped independently with probability satisfying assumption 3. Let $\mathcal{E}^*$ and $\mathcal{E}'$ represent the noisy distributions of $\mathbf{X}$ and $\tilde{\mathbf{X}}$ respectively. If $\mathcal{E}' = \mathcal{E}^*$, then $T' \in \mathcal{T}_{T^*}$.*

Proof. The proof of this theorem relies on this key observation: the probability distribution of the noisy samples completely defines the categorization of any set of 4 nodes as star/non-star shape (as defined in 2.4.2). Once we prove this key observation, the rest of the proof follows from the proof of theorem 2.4.2. Next, we see how to classify a set of 4 nodes as star/non-star using the noisy samples.

We denote the correlation between two nodes X_i and X_j in the non-noisy setting by $\rho_{i,j}$ and in the noisy setting by $\rho'_{i,j}$. Similarly the covariance is denoted by $\Sigma_{i,j}$ and $\Sigma'_{i,j}$. We utilize the correlation decay property of tree structured Ising models which is stated in Lemma 3.3.5.

Lemma 3.3.5. *(Correlation Decay) Any 2 nodes X_{i_1} and X_{i_k} have the conditional independence relation specified by a tree structured Ising Model such that the path between them is $(X_{i_1} \to X_{i_2} \to X_{i_3} \cdots \to X_{i_k})$ if and only if their correlation is given by:*

$$\rho_{i_1 i_k} = \prod_{l=2}^{k} \rho_{i_{l-1}, i_l}. \tag{3.1}$$

The proof of this lemma is provided in Appendix, B.1. We also prove that $\mathbb{E}[X_i^e] = (1-2q_i)\mathbb{E}[X_i]$ and $\Sigma'_{i,j} = (1-2q_i)(1-2q_j)\Sigma_{i,j}$ in Appendix B.2.

Categorizing a set of 4 nodes as star/non-star

We first look at a graphical model on 3 nodes X_1, X_2, X_3 whose conditional independence is given by a chain with $X_2 \perp X_3 | X_1$. By Lemma 3.3.5, we have $\Sigma_{2,3}\Sigma_{1,1} = \Sigma_{1,2}\Sigma_{1,3}$.

Suppose the sign of X_1, X_2, X_3 flip independently with probability q_1, q_2, q_3 respectively. Substituting the values of $\Sigma_{2,3}, \Sigma_{1,1}, \Sigma_{1,2}$ and $\Sigma_{1,3}$ in terms of their noisy counterparts gives us:

$$(1-2q_1)^2 = 1 - \Sigma'_{1,1} + \frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}}. \tag{3.2}$$

If we had prior knowledge about the underlying conditional independence relation, this quadratic equation, which depends only on the quantities measurable from noisy data, could be solved to estimate the probability of error of X_1.

We prove in Appendix B.3 that Equation (3.2) gives a valid solution for any configuration of 3 nodes in a tree structured Ising model. Therefore, in the absence of the knowledge that $X_2 \perp X_3 | X_1$, we can estimate a probability of error for each X_i which enforces the underlying graph structure to represent the other 2 nodes independent conditioned on X_i. Thus, irrespective of the true underlying conditional independence relation we can always find a probability of error for each node which makes any other pair of nodes conditionally independent. We use this concept to classify a tree on 4 nodes as star or non-star shaped.

We follow a notation where $\hat{q}_i^{j,k}$ denotes the estimated probability of error of X_i which enforces $X_j \perp X_k | X_i$.

Condition for star/non-star shape:

Any set of 4 nodes $\{X_1, X_2, X_3, X_4\}$ is categorized as a non-star with (X_1, X_2) forming one pair and (X_3, X_4) forming another pair if and only if:

$$\hat{q}_1^{2,3} = \hat{q}_1^{2,4} \neq \hat{q}_1^{3,4}, \hat{q}_2^{1,3} = \hat{q}_2^{1,4} \neq \hat{q}_2^{3,4},$$
$$\hat{q}_3^{2,4} = \hat{q}_3^{1,4} \neq \hat{q}_3^{1,2}, \hat{q}_4^{2,3} = \hat{q}_4^{1,3} \neq \hat{q}_4^{1,2}.$$

From Equation (3.2), this is equivalent to the condition that $\frac{\rho'_{1,3}}{\rho'_{1,4}} = \frac{\rho'_{2,3}}{\rho'_{2,4}}, \frac{\rho'_{1,2}}{\rho'_{1,4}} \neq \frac{\rho'_{3,2}}{\rho'_{3,4}}$.

Any set of 4 nodes $\{X_1, X_2, X_3, X_4\}$ is categorized as a star if and only if:

$$\hat{q}_1^{2,3} = \hat{q}_1^{2,4} = \hat{q}_1^{3,4}, \hat{q}_2^{1,3} = \hat{q}_2^{1,4} = \hat{q}_2^{3,4},$$
$$\hat{q}_3^{2,4} = \hat{q}_3^{1,4} = \hat{q}_3^{1,2}, \hat{q}_4^{2,3} = \hat{q}_4^{1,3} = \hat{q}_4^{1,2}.$$

This is equivalent to the condition that $\frac{\rho'_{1,3}}{\rho'_{1,4}} = \frac{\rho'_{2,3}}{\rho'_{2,4}} = \frac{\rho'_{1,2}}{\rho'_{1,4}}$.

In order to see how these conditions correspond to a star/non-star shape, lets consider a chain on 4 nodes as shown in Figure 3.1. Let each X_i be flipped with probability q_i. With access only to the noisy samples, we estimate the probability of error for each node in order to find the underlying tree. The key idea is that when we estimate the probability of error for a given node, it should be consistent across different conditional independence relations. For instance in the present case, the error estimates $\hat{q}_2^{1,3}$ and $\hat{q}_2^{1,4}$ of X_2 satisfy $\hat{q}_2^{1,3} = \hat{q}_2^{1,4} = q_2$. We show that $\hat{q}_2^{3,4} \neq \hat{q}_2^{1,3}$ (Lemma 3.3.6(b)). We also prove that $\hat{q}_1^{2,3} = \hat{q}_1^{2,4} \neq \hat{q}_1^{3,4}$(Lemma 3.3.6). These imply that $X_3 \not\perp X_4 | X_2$ and $X_3 \not\perp X_4 | X_1$. By symmetry, we have $X_1 \not\perp X_2 | X_3$ and $X_1 \not\perp X_2 | X_4$. These conditional independence statements imply that X_1, X_2, X_3 and X_4

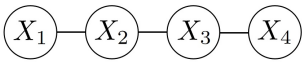


Figure 3.1: A chain structure.

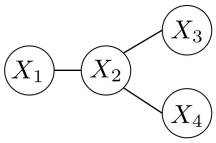


Figure 3.2: A Star structure.

form a chain with (X_1, X_2) on one side of the chain and (X_3, X_4) on the other side of the chain.

Next, we consider the case when 4 nodes form a star structured graphical model as in Figure 3.2. Under the same noisy observation setting we prove that $\hat{q}_1^{2,3} = \hat{q}_1^{2,4} = \hat{q}_1^{3,4}$, $\hat{q}_2^{1,3} = \hat{q}_2^{1,4} = \hat{q}_2^{3,4}$, $\hat{q}_3^{1,2} = \hat{q}_3^{1,4} = \hat{q}_3^{2,4}$ and $\hat{q}_4^{1,3} = \hat{q}_4^{1,2} = \hat{q}_4^{3,2}$ (Lemma 3.3.7). Thus, we can conclude that the underlying graphical model is star structured.

Lemma 3.3.6. *Let the graphical model on X_1, X_2, X_3 and X_4 form a chain as shown in Figure* 3.1*. Suppose the bits of each X_i are flipped with probability $q_i < 0.5$. Then the following holds:*

$$(a)\ \hat{q}_1^{2,3} = \hat{q}_1^{2,4},\ \ (b)\ \hat{q}_2^{1,3} \neq \hat{q}_2^{3,4}, \hat{q}_1^{2,3} \neq \hat{q}_1^{3,4}$$

Lemma 3.3.7. *Let the graphical model on X_1, X_2, X_3 and X_4 form a star as shown in Figure 3.2. Suppose the bits of each X_i are flipped with probability $q_i < 0.5$. Then the following holds:*

$$\hat{q}_1^{2,3} = \hat{q}_1^{2,4} = \hat{q}_1^{4,3}$$

The proof of these lemmas and the details of extending these results to generic trees require basic algebraic manipulations and can be found in Appendix B.4. □

Theorem 3.3.8. *Let $\mathcal{E}'$ denote the probability distribution of $\mathbf{X}'$ when the error probability of all the neighbors of leaf nodes is non-zero. For any $\tilde{T} \in \mathfrak{T}_{T^*}$, there exists a set of random*

variables $\tilde{\mathbf{X}}$ *with conditional independence given by* $\tilde{T}$ *and a corresponding error probability vector* $\tilde{\mathbf{q}}$ *such that* $\mathcal{E}' = \tilde{\mathcal{E}}$ *where* $\tilde{\mathcal{E}}$ *denotes the noisy distribution of* $\tilde{\mathbf{X}}$.

We prove this theorem by explicit calculation of $\tilde{\mathbf{q}}$. We utilize Lemma 3.3.5 to enforce the conditional independence relations in any tree $\tilde{T} \in \mathcal{T}_{T^*}$. The proof is included in Appendix B.5.

Interestingly these unidentifiability results for noisy tree structured Ising models match the ones for noisy tree structured Gaussian graphical models inspite of them being graphical models on different class of random variables.

Chapter 4

Recoverability Landscape of Tree Structured Markov Random Fields under Symmetric Noise

4.1 Introduction

In this chapter, we focus on learning the underlying tree-structured graphical model on non-noisy discrete random variables with common support size k using samples that are corrupted by a k-ary symmetric noise channel. 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 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*.

Parts of this chapter are available at: Katiyar, Ashish, Soumya Basu, Vatsal Shah, and Constantine Caramanis. "Recoverability Landscape of Tree Structured Markov Random Fields under Symmetric Noise." arXiv preprint arXiv:2102.08554 (2021). The author formulated the problem, designed the algorithm, performed experiments, and contributed in the theoretical analysis and paper writing.

Our contributions can be summarized as follows:

1. **Identifiability Characterization**: In *Theorem 4.4.2*, 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 4.4.4*), 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 4.4.4*) 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 4.5)*.

3. **Sample Complexity Analysis:** We provide novel sample complexity lower bounds and upper bounds (*Section 4.6*). Our upper bounds are shown to have orderwise tight dependence on underlying graph parameters, size of the graph, edge parameters (related to underlying conditional MF), 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:**[1] 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.

4.2 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 [7, 9, 5, 6, 42, 36, 86, 60]. 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 [17]. Chow-Liu algorithm's error exponents for Gaussian graphical models and graphical models on discrete random variables were analyzed in [71] and [69] respectively. Results in [69] were further refined in [72] under additional assumptions of homogeneity and zero external field in tree-structured Ising models. In [8] the authors approximate the distribution of generic Ising models using tree-structured Ising models. More recently, in [21], the authors provide an algorithm to learn tree-structured Ising models providing total variation distance guarantees. In [4], the authors provide finite sample guarantees for the Chow-Liu algorithm. As these algorithms assume access to non-noisy samples,

[1]The code containing the implementation of the algorithm is available at `https://github.com/ashishkatiyar13/NoisyTreeMRF`

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 [26, 44, 27] 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. In [72, 54, 56], the authors study the recovery of trees using noisy samples. 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.

In [73] the authors extend our results for Gaussian graphical models and Ising models, providing better sample complexity results and a more efficient algorithm. These results 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 [58, 13, 16], and *independently and concurrently* in [11], 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.

4.3 Problem Setup

Let $\mathbf{X} = [X_1, X_2 \dots X_n]$ be the vector of random variables with a common support set, $\mathcal{S} = \{s_1, s_2, \dots s_k\}$ 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,

$$X_i' = \begin{cases} X_i & \text{w.p. } 1 - q_i, \\ U_i & \text{w.p. } q_i, \end{cases} \tag{4.1}$$

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 $\mathbf{X}' = [X_1', X_2' \dots X_n']$. 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 proposed in [40], is defined as follows:

$$d_{i,j} = -\log \frac{|det(P_{i,j})|}{\sqrt{det(P_i)det(P_j)}}, d_{i',j'} = -\log \frac{|det(P_{i',j'})|}{\sqrt{det(P_{i'})det(P_{j'})}}. \tag{4.2}$$

We require the following assumptions that are natural and standard in this line of literature (c.f. [13, 16]).

Assumption 4.3.1. *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 4.3.2. *The distance* $d_{i,j}$ *between adjacent non-noisy random variables is bounded:* $0 < d_{min} < d_{i,j} < d_{max}$.

Assumption 4.3.3. *The probability of error is upper bounded away from 1:* $q_i \leq q_{max} < 1$.

Assumption 4.3.1 ensures that the probability mass at any support is not arbitrarily small for any random variable. The bounds on the distance in Assumption 4.3.2 ensure that no adjacent random variables are duplicates or independent. Assumption 4.3.3 ensures that the noisy observations are not independent of the underlying random variables. Our sample complexity lower bounds in Section 4.6 show that the problem becomes infeasible if these assumptions are not satisfied.

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

Definition 4.3.1. The **leaf cluster** of any leaf node is the set containing that leaf node, its parent node and all its sibling leaf nodes.

4.4 Identifiability Results

In this section, we prove that the identifiability of the underlying tree is determined by the joint PMF of leaf parent pairs. 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.

4.4.1 Potential unidentifiability is limited to leaf clusters

For any tree T^*, recall the definition of the equivalence class of trees $\mathcal{T}_{T^*}$ from 2.4.1. 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 4.4.1. *Suppose the random variables in* $\mathbf{X}$ *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 key ingredient of this proof is the use of the information distance metric $d_{i,j}$ as defined in (4.2) to categorize a set of 4 nodes as star/non-star (defined in 2.4.2). Once we have the star/non-star categorization, the proof of Theorem 2.4.2 gives us the desired result.

Remarks: (i) Lemma 4.4.1 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 [11]. (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 produces the noisy observation. This analysis along with the proof of Lemma 4.4.1 is included in Appendix C.1.

4.4.2 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:

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

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.

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:

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

Note that $P_{a',b'}$ and $P_{a,b}$ are related as follows:

$$P_{a',b'} = E_a P_{a,b} E_b. \tag{4.3}$$

It is also easy to see that:

$$P_{a'} = (1 - q_a)P_a + \tfrac{q_a}{k}I. \tag{4.4}$$

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 (4.1). The conditional independence relationship gives us:

$$P_{1,3} = P_{1,2} P_2^{-1} P_{2,3}. \tag{4.5}$$

From Equation (4.3), 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 (4.4), we have $P_{2'} = (1 - q_2)P_2 + \frac{q_2}{k} I$. By substituting these in Equation (4.5) we get the following quadratic equation with matrix coefficients in noise parameter q_2 (details in Appendix C.2):

$$\frac{q_2^2}{k^2}(O - kI) - \frac{q_2}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'} P_{1,'3'}^{-1} P_{1',2'} - P_{2'} = 0, \tag{4.6}$$

where the 0 on the RHS is a $k \times k$ matrix of all 0s. The key insight here is that, Equation (4.6) 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 the we can satisfy Equation (4.6) 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:

$$\min_{0 \le x \le q_{max}} \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'} P_{1,'3'}^{-1} P_{1',2'} - P_{2'}\|_F = 0. \tag{4.7}$$

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.

4.4.3 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 (4.7). 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 4.4.2. *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* (4.7) *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 (4.7) has a solution when the parent node is the middle node. Whenever Equation (4.7) 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.3.

4.4.4 Examples

In this section, we do not assume access to q_{max} and analyse the solution to Equation (4.7) with the constraint $0 < x < 1$. Extension to the setting of $0 < x < q_{max}$ is straightforward where we reject any solution $x > q_{max}$. We first prove that symmetric graphical models are unidentifiable. Next, we present perturbed symmetric graphical models that are unidentifiable for $k = 3$ but are identifiable for $k \geq 4$. Finally, we show that our analysis recovers the existing results for $k = 2$.

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 4.3.2 enforces $\exp(-d_{max}/(k-1)) < \alpha_{a,b} < \exp(-d_{min}/(k-1))$.

Theorem 4.4.3. *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. 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* (4.7) *has a solution.*

The proof is included in Appendix C.4. Since, Equation (4.7) has a solution for every leaf node X_2 as the candidate center node, using Theorem 4.4.2, 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 $0 < c_{a,b} < k$, the term in the (i, j) position of $\Delta_{a,b}$ is:

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

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:

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

Here $\alpha_{a,b}$ and $\delta_{a,b}$ are chosen such that Assumption 4.3.2 is satisfied. We find that perturbed symmetric graphical models are unidentifiable for $k = 3$ but become identifiable for $k \geq 4$.

Theorem 4.4.4. *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 4.3.2 are satisfied. Then, equation* (4.7) *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 4.4.2, this implies that the exact tree is identifiable when $k \geq 4$. For $k = 3$, we explicitly calculate the solution to Equation (4.7). 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 C.5.

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

Lemma 4.4.5. *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 symmetric channel with positive probability of error and we observe* $\mathbf{X}'$*. For any 3 nodes* (X_1, X_2, X_3)*, Equation* (4.7) *always has a valid solution.*

The proof of Lemma 4.4.5 is in Appendix C.6. Corollary 4.4.6 recovers the unidentifiability results for Ising models.

Corollary 4.4.6. *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* $\mathcal{T}^{sub}_{T^*} = \mathcal{T}_{T^*}$.

4.5 Algorithm

In this section, we present the algorithm to recover a tree from $\mathcal{T}^{sub}_{T^*}$ 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.

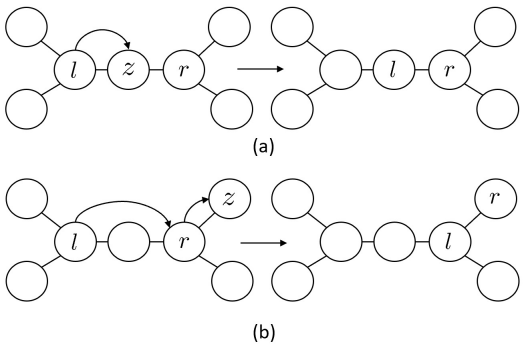


Figure 4.1: (a) If the node z lies between l and r, l becomes z, hence getting closer to r. (b) If the node r lies between l and z, both l and r shift towards the right with l becoming r and r becoming z.

Algorithm 1 Recover Tree Structure

Input: Pairwise noisy distributions, $P'_{i,j}$ $\forall i, j \in [n]$
Output: List of edges, $Edges$

procedure FindTree($P'_{i,j}$ $\forall i, j \in [n]$)
    $ActiveSet \leftarrow \{1, 2, \ldots n\}$, $Edges \leftarrow \{\}$, $Parents \leftarrow \{\}$
    while $|ActiveSet| > 2$ **do**
        $leaf, parent \leftarrow$ GetLeafParent($P'_{i,j}$, $ActiveSet$, $Edges$, $Parents$)
        $ActiveSet \leftarrow ActiveSet \setminus leaf$
        $Edges \leftarrow Edges \cup (leaf, parent)$
        $Parents \leftarrow Parents \cup parent$
    end while
    $Edges \leftarrow Edges \cup (ActiveSet[0], ActiveSet[1])$
return $Edges$
end procedure

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 (4.1). Finding the center node can be done by checking the feasibility of Equation (4.7) for different candidate center nodes.

If Equation (4.7) has a solution for more than one nodes, we use an alternative method which uses the 3 nodes in conjunction with different 4^{th} 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 4.3.1 and 4.3.3 respectively, $\eta_{max} = (1-k)\log(1-q_{max}) - 0.5k\log(kp_{min})$ (details in Appendix C.7). 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 (4.7) using empirical PMF estimates, we need the following additional assumption:

Assumption 4.5.1. *When Equation* (4.7) *does not have a solution, we have the following inequality:*

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

This assumption ensures that when Equation (4.7) 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 C.7, 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 lose 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 lose 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 (4.7) is chosen as the center node.

4.6 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 4.6.1 (**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 4.3.1, 4.3.2, 4.3.3 and 4.5.1 are satisfied. Then, the finite sample Algorithm 1 correctly recovers* $\mathfrak{T}_{T^*}^{sub}$ *with probability at least* $1-\delta$ *if the number of samples* N *satisfies*

$$N = \mathcal{O}\Bigg(\max \Bigg\{ \frac{k^2 \exp(8d_{\max})}{(1-q_{max})^{6(k-1)}(0.9p_{min}^{2.5})^{2k}(1-\exp(-2d_{min}))^2(k-1)^{2(k-1)}}, \\ \frac{k \exp(16d_{\max})}{t_0^2(1-q_{max})^{12(k-1)}(0.9p_{min}^{2.5})^{4k}(k-1)^{4(k-1)}} \Bigg\} \log\left(\frac{2nk(n-1)}{\delta}\right) \Bigg)$$

In the unidentifiable setting, since Equation (4.7) 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 $\mathfrak{T}_{T^*}$, the sample complexity has better dependence on d_{max}, q_{max} and p_{min}.

Theorem 4.6.2 (**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 4.3.1, 4.3.2,*

4.3.3 and 4.5.1 are satisfied. Then any algorithm that correctly recovers $\mathfrak{T}_{T^*}^{sub}$ with probability at least $1-\delta$ requires N samples where

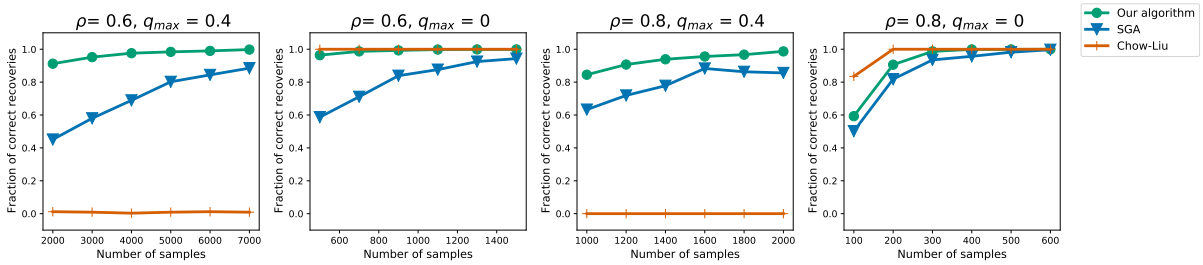
$$N = \Omega\left(\frac{\exp\left(\frac{2d_{\max}}{k-1}\right)}{(k-1)(1-q_{\max})^2\left(1-\exp\left(-\frac{d_{\min}}{k-1}\right)\right)}(1-\delta)\log(n)\right)$$

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

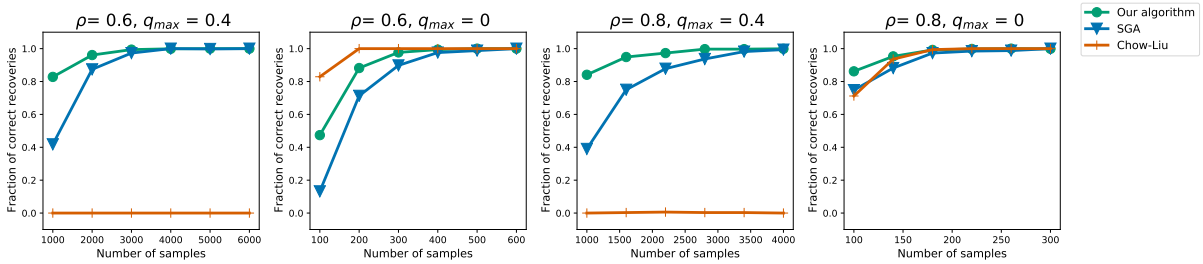
$$N = \Omega\left(\max_{d\in\{d_{\max},d_{\min}\}} \exp\left(-\tfrac{2d}{k-1}\right)\left(1-\exp\left(-\tfrac{d}{k-1}\right)\right)\frac{k(1-\delta)\log(n)}{t_0^2}\right)$$

We note that our lower bounds on sample complexity shows 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 on the lower bound for the quadratic term $Q(x)$, t_0, scales as $\Theta(\frac{1}{t_0^2})$ highlighting the significance of the term $Q(x)$ in the recovery of MRFs under unknown symmetric noise model.

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 4.4.4), 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 $\mathfrak{T}_{T^*}^{sub}$ for each graph in the family is unique. For the lower bounds in the unidentifiable scenario, we generalize the construction in [73] to $k > 2$ support size using symmetric graphical models. Our derivation for KL divergence for symmetric graphical model, and perturbed symmetric graphical models used in the lower bound proofs can be of independent interest.



(a) Chain Graph



(b) Star Graph

Figure 4.2: 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) $\rho_{max} = 0.8, q_{max} = 0.0$

4.7 Experiments

In this section, we present the experiments demonstrating the efficacy of our algorithm (The code can be found at `https://github.com/ashishkatiyar13/NoisyTreeMRF`.). We first demonstrate the performance of our algorithm for the $k = 2$ setting and demonstrate that our algorithm considerably outperforms the algorithm in [73]. Next, we showcase the performance of our algorithm for the $k = 4$ setting with the perturbed symmetric model. As discussed in Section 4.4.4, the exact tree is identifiable in this scenario.

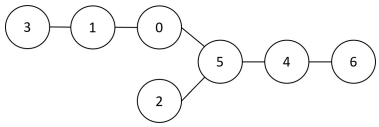


Figure 4.3: Randomly generated graph used for algorithm evaluation.

4.7.1 Support size, $k = 2$ (Unidentifiable setting):

In this part, we compare the performance of our algorithm for chain and star graphs to that of SGA proposed in [73]. We use the exact same settings as in [73] 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 = ρ. (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. (v) Number of iterations = 1000

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 4.2a and 4.2b respectively.

4.7.2 Support size, $k = 4$ (Identifiable Setting):

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 4.3).

Setting : (i) Number of nodes = 7.

(ii) Graph Shape = {Chain, Star, Random}

(iii) Distance of all the adjacent nodes = $\exp(-0.7)$.

(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.

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.

We also perform extensive experiments where we evaluate the impact of the probability of error and the distance between adjacent nodes and present the results in Appendix C.10.

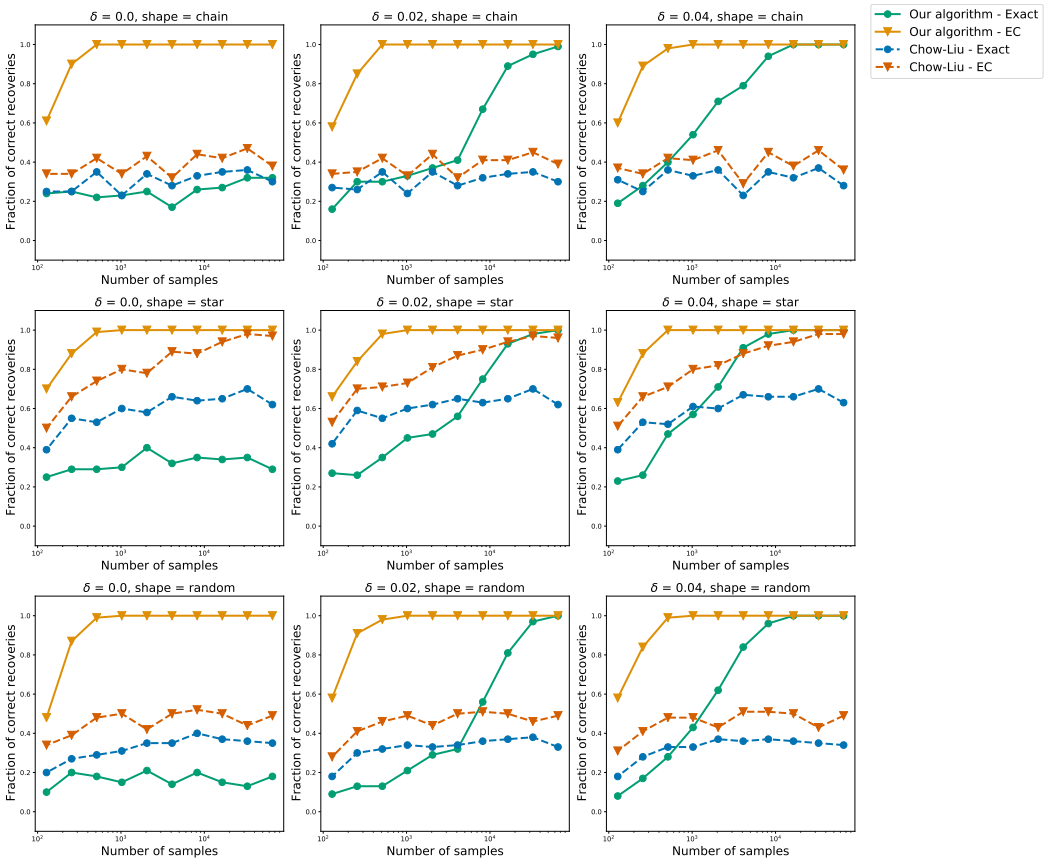


Figure 4.4: Comparing the performance of our algorithm and Chow-Liu over different values of $\delta_{i,j} \in \{0.00, 0.02, 0.04\}$ and different graph shapes - chain, star, random. Setting: $d_{min} = d_{max} = \exp(-0.7)$, $q_{max} = 0.2$, # of nodes= 7. For both algorithms, we provide results for two cases: i) when the exact underlying tree is recovered, ii) when a tree from the equivalence class is recovered.

Appendices

Appendix A

Robust Estimation of Tree Structured Gaussian Graphical Models

A.1 Proof of Theorem 1

Consider any tree $T^q \in \mathcal{T}_{T^*}$ and its corresponding set $\mathcal{S}^q$. We find the covariance matrix Σ^q with the same off diagonal elements as Σ^o whose independence structure is given by T^q. Upon obtaining Σ^q, getting the D^q matrix is immediate. To begin with, let us consider the case when $\mathcal{S}^q$ has just one node, i.e, $\mathcal{S}^q$ consists of one of the leaves of T^*.

Proposition A.1.1. *Suppose the covariance matrix Σ^* has conditional independence structure T^* with leaf node a and its neighbor b. Consider a covariance matrix Σ^q defined as follows:*

$$\Sigma^q_{ij} = \begin{cases} \Sigma^*_{ij} - \frac{1}{\Omega^*_{aa}} & \text{if } i = j = a \\ \Sigma^*_{ij} + c^i_1 & 0 < c^i_1 < D^*_{ij} \text{ if } i = j = b \\ \Sigma^*_{ij} & \text{otherwise,} \end{cases}$$

The conditional independence structure T^q of Σ^q is given by the tree obtained by exchanging positions of node a and b in T^*.

Proof. Relabeling if necessary, assume that node n is a leaf node and node $n-1$ is its

neighbor in T^*. Define B^1 and B^2 as follows:

$$B^1_{ij} = \begin{cases} c^i_1 & 0 < c^i_1 < D^*_{n-1n-1} \text{ if } i = j = n-1 \\ 0 & \text{otherwise} \end{cases},$$

$$B^2_{ij} = \begin{cases} -\frac{1}{\Omega^*_{nn}} & \text{if } i = j = n \\ 0 & \text{otherwise} \end{cases}.$$

We also define an intermediate matrix $\Sigma^I = \Sigma^* + B^2$. Therefore $\Sigma^q = \Sigma^I + B^1$. The proof of this proposition can be split in the following steps:

(i) We prove that for Σ^I column n is a multiple of column $n-1$ making it a low rank matrix.

(ii) We add B^1 to Σ^I to get Σ^q. In Σ^q column n is a multiple of column $n-1$ at all elements other than $n - 1^{st}$. This makes node $n-1$ a leaf node connected to node n as we see in Lemma A.1.2.

(iii) We prove that the independence structure of the rest of the nodes does not change. This is done by proving 2 claims:

(a) Conditional independence relations do not change when if conditioning is not on node n or node $n-1$.

(b) Any pair of nodes which were independent conditioned on $n-1$ in Σ^* are independent conditioned on n in Σ^q.

A.1.1 Proof of Part(i) - Column n of Σ^I is a multiple of column $n-1$:

The precision matrix Ω^* is of the form:

$$\Omega^* = \left[\begin{array}{ccc|c} \Omega^*_{11} & \dots & \Omega^*_{1n-1} & 0 \\ \vdots & \ddots & \vdots & \vdots \\ \Omega^*_{1n-1} & \dots & \Omega^*_{n-1n-1} & \Omega^*_{n-1n} \\ \hline 0 & \dots & \Omega^*_{n-1n} & \Omega^*_{nn} \end{array}\right]. \tag{A.1}$$

For notational convenience, in what follows, we label the blocks in (A.1) as Ω^*_x, Ω^*_y and Ω^*_z, so that:

$$\Omega^* = \left[\begin{array}{c|c} \Omega^*_x & \Omega^*_y \\ \hline (\Omega^*_y)^T & \Omega^*_z \end{array}\right].$$

As depicted in (A.1), block Ω^*_y is a $n-1$ length vector with a non zero only at position $n-1$. The covariance matrix $\Sigma^* = (\Omega^*)^{-1}$ is as follows:

$$\Sigma^* = \left[\begin{array}{ccc|c} \Sigma^*_{11} & \dots & \Sigma^*_{1n-1} & \Sigma^*_{1n} \\ \vdots & \ddots & \vdots & \vdots \\ \Sigma^*_{1n-1} & \dots & \Sigma^*_{n-1n-1} & \Sigma^*_{n-1n} \\ \hline \Sigma^*_{1n} & \dots & \Sigma^*_{n-1n} & \Sigma^*_{nn} \end{array}\right].$$

As with Ω^*, we write it in blocks as:

$$\Sigma^* = \left[\begin{array}{c|c} \Sigma^*_x & \Sigma^*_y \\ \hline (\Sigma^*_y)^T & \Sigma^*_z \end{array}\right]. \tag{A.2}$$

By the matrix inversion lemma, we have:

$$\Sigma^*_x = (\Omega^*_x)^{-1} + (\Omega^*_x)^{-1}\Omega^*_y[\Omega^*_z - (\Omega^*_y)^T(\Omega^*_x)^{-1}(\Omega^*_y)]^{-1}(\Omega^*_y)^T(\Omega^*_x)^{-1}.$$

To ease notation, we define $c_2 \triangleq [\Omega_z^* - (\Omega_y^*)^T(\Omega_x^*)^{-1}(\Omega_y^*)]^{-1}$. The $(n-1)^{st}$ column of Σ_x^* is given as follows:

$$(\Sigma_x^*)_{:,n-1} = [1 + c_2(\Omega_x^*)^{-1}_{n-1,n-1}(\Omega_{n-1n}^*)^2](\Omega_x^*)^{-1}_{:,n-1}. \tag{A.3}$$

Note that $(\Sigma_x^*)_{n-1,n-1} = \Sigma^*_{n-1n-1}$ and $(\Omega_x^*)_{n-1,n-1} = \Omega^*_{n-1n-1}$.

By the matrix inversion lemma, we also have:

$$\Sigma_y^* = -(\Omega_x^*)^{-1}\Omega_y^*[\Omega_z^* - (\Omega_y^*)^T(\Omega_x^*)^{-1}(\Omega_y^*)]^{-1}.$$

Substituting c_2 for $[\Omega_z^* - (\Omega_y^*)^T(\Omega_x^*)^{-1}(\Omega_y^*)]^{-1}$ and the value of Ω_y^* from equation (A.1) we get:

$$\Sigma_y^* = -c_2\Omega^*_{n-1n}(\Omega_x^*)^{-1}_{:,n-1}. \tag{A.4}$$

By Equations (A.3) and (A.4) we have:

$$\Sigma_y^* = \frac{-c_2\Omega^*_{n-1n}}{[1 + c_2(\Omega_x^*)^{-1}_{n-1,n-1}(\Omega_{n-1n}^*)^2]}(\Sigma_x^*)_{:,n-1}. \tag{A.5}$$

Hence, the n^{th} column of Σ^* is a multiple of the $(n-1)^{st}$ column except for the n^{th} element. Also, by the matrix inversion lemma $\Sigma^*_{nn} = \Sigma_z^* = c_2$.

Now we look at the intermediate matrix Σ^I which is given as follows:

$$\Sigma^I = \left[\begin{array}{ccc|c} \Sigma^*_{11} & \cdots & \Sigma^*_{1n-1} & \Sigma^*_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma^*_{1n-1} & \cdots & \Sigma^*_{n-1n-1} & \Sigma^*_{n-1n} \\ \hline \Sigma^*_{1n} & \cdots & \Sigma^*_{n-1n} & \Sigma^*_{nn} - \frac{1}{\Omega^*_{nn}} \end{array}\right]. \tag{A.6}$$

Now we prove that Σ^I is a rank deficient matrix and its n^{th} column is a multiple of its $(n-1)^{st}$ column. Specifically, letting $c_3 \triangleq \frac{-c_2\Omega^*_{n-1n}}{[1+c_2(\Omega_x^*)^{-1}_{n-1,n-1}(\Omega_{n-1n}^*)^2]}$, we show that $\Sigma^I_{:,n} = c_3\Sigma^I_{:,n-1}$. This is true for the first $(n-1)$ elements by Equation (A.5). Basically we need to prove the

following:

$$\Sigma^*_{nn} - \frac{1}{\Omega^*_{nn}} = c_3 \Sigma^*_{n-1n}. \tag{A.7}$$

Expanding the LHS in Equation (A.7), we get

$$\begin{aligned} \Sigma^*_{nn} - \frac{1}{\Omega^*_{nn}} &= \frac{1}{\Omega^*_{nn} - (\Omega^*_{n-1n})^2 (\Omega^*_x)^{-1}_{n-1n-1}} - \frac{1}{\Omega^*_{nn}} \\ &= \frac{c_2}{\Omega^*_{nn}} (\Omega^*_{n-1n})^2 (\Omega^*_x)^{-1}_{n-1n-1}. \end{aligned} \tag{A.8}$$

For the RHS of Equation (A.7), we substitute Σ^*_{n-1n} from Equation (A.4) and the value of c_3 to get the following:

$$\begin{aligned} c_3 \Sigma^*_{n-1n} &= \frac{c_2^2 (\Omega^*_{n-1n})^2}{[1 + c_2 (\Omega^*_x)^{-1}_{n-1,n-1} (\Omega^*_{n-1n})^2]} (\Omega^*_x)^{-1}_{n-1n-1} \\ &= \frac{c_2 (\Omega^*_{n-1n})^2}{[c_2^{-1} + (\Omega^*_x)^{-1}_{n-1,n-1} (\Omega^*_{n-1n})^2]} (\Omega^*_x)^{-1}_{n-1n-1} \\ &= \frac{c_2}{\Omega^*_{nn}} (\Omega^*_{n-1n})^2 (\Omega^*_x)^{-1}_{n-1n-1}. \end{aligned} \tag{A.9}$$

From Equations (A.8) and (A.9) we conclude that that $(\Sigma^I)_{:,n} = c_3 (\Sigma^I)_{:,n-1}$. Hence, Σ^I is a rank deficient matrix. Also note that the first $n-1$ principal sub matrices of Σ^I have positive determinant by the positive definiteness of Σ^*. Hence, $rank(\Sigma^I) = n-1$.

A.1.2 Proof of part (ii) - Node $n-1$ is a leaf node connected to node n in the independence structure of Σ^q:

Next we add B^1 to Σ^I to get Σ^q:

$$\Sigma^q = \left[\begin{array}{ccc|c} \Sigma^*_{11} & \dots & \Sigma^*_{1n-1} & \Sigma^*_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ \Sigma^*_{1n-1} & \dots & \Sigma^*_{n-1n-1} + c_1^{n-1} & \Sigma^*_{n-1n} \\ \hline \Sigma^*_{1n} & \dots & \Sigma^*_{n-1n} & \Sigma^*_{nn} - \frac{1}{\Omega^*_{nn}} \end{array}\right],$$

for any $0 < c_1^{n-1} < D^*_{n-1n-1}$. In Σ^q column $n-1$ is not multiple of column n, hence it is a symmetric positive definite matrix making it a valid covariance matrix. Also, column $n-1$ is a multiple at all indices except at index n. In order to prove that node $n-1$ is a leaf node connected to node n, we use Lemma A.1.2.

Lemma A.1.2. *If in any covariance matrix Σ, column $n-1$ is a multiple $\alpha \neq 0$ of column n except at position $n-1$, then in the independence structure of Σ, node $n-1$ is a leaf node connected to node n.*

Proof of Lemma 1: We look at the edges of node $n-1$ given by the $(n-1)^{st}$ column of $\Omega = \Sigma^{-1}$.

$$|\Omega_{n-1i}| = \frac{|det(\Sigma_{-(n-1),-i})|}{det(\Sigma)}$$

For $i \notin n, n-1$, $\Omega_{n-1i} = 0$ as the submatrix $\Sigma_{-(n-1),-i}$ is rank deficient by assumption. Note that $\Omega_{n-1n} \neq 0$, because by contradiction if that was true, Ω would be a block diagonal with node $n-1$ as one block. This would imply that Σ would be a block diagonal with node $n-1$ as one block, which cannot be the case as $\Sigma_{n-1n} = \alpha\Sigma_{nn} \neq 0$. Hence node $n-1$ is a

leaf node connected to node n. $\square$

By Lemma A.1.2, node $n-1$ is a leaf node connected to node n in T^q.

A.1.3 Proof of part (iii) - Structure of the remaining tree does not change:

In order to prove this part, we need the following lemma:

Lemma A.1.3. *For any random vector* $Y = [Y_1, Y_2, \ldots, Y_n]$, $Y \sim \mathcal{N}(0, \Sigma)$, Y_i *is independent of* Y_j *conditioned on* Y_k *if and only if*

$$\Sigma_{ij} = \frac{\Sigma_{ik}\Sigma_{jk}}{\Sigma_{kk}}.$$

Proof of Lemma A.1.3: The probability distribution of Y_{-k} conditioned on Y_k is given as follows:

$$Y_{-k} \mid Y_k \sim \mathcal{N}(\Sigma_{-k,k}\Sigma_{kk}^{-1}Y_k, \Sigma_{-k,-k} - \frac{\Sigma_{k,-k}\Sigma_{-k,k}}{\Sigma_{kk}}).$$

For Y_i to be independent of Y_j conditioned on Y_k, the i,j component of the conditional covariance matrix must be zero, giving

$$\Sigma_{ij} = \frac{\Sigma_{ik}\Sigma_{jk}}{\Sigma_{kk}}. \quad \square$$

Proof of part (iiia) - Conditional independence relations, when conditioning is not on n or $n-1$, don't change:

This is a direct consequence of Lemma A.1.3 as $\Sigma^q_{kk} = \Sigma^*_{kk}$ for $k \neq n, n-1$.

Proof of part (iiib) - Any pair of nodes which were independent conditioned on $n-1$ in Σ^* are independent conditioned on n in Σ^q:

Suppose node i and node j were independent conditioned on node $n-1$ in Σ^* and $i, j \neq n$. Then by Lemma A.1.3 we have:

$$\Sigma^*_{ij} = \frac{\Sigma^*_{n-1i}\Sigma^*_{n-1j}}{\Sigma^*_{n-1n-1}}.$$

From Equation(A.2), note that $\Sigma^*_{n-1i} = (\Sigma^*_x)_{n-1i}$ and $\Sigma^*_{n-1j} = (\Sigma^*_x)_{n-1j}$, also $\Sigma^*_{ni} = (\Sigma^*_y)_i$ and $\Sigma^*_{nj} = (\Sigma^*_y)_j$. So, by Equation (A.5), we have:

$$\Sigma^*_{ij} = \frac{\Sigma^*_{ni}\Sigma^*_{nj}}{c_3\Sigma^*_{n-1n}}.$$

Since the off diagonal terms of Σ^* and Σ^q are equal, we have:

$$\Sigma^q_{ij} = \frac{\Sigma^q_{ni}\Sigma^q_{nj}}{c_3\Sigma^q_{n-1n}}.$$

By Equation (A.7) we can substitute the denominator to obtain:

$$\Sigma^q_{ij} = \frac{\Sigma^q_{ni}\Sigma^q_{nj}}{\Sigma^q_{nn}}.$$

Therefore, by Lemma A.1.3, in the graphical structure for Σ^q, i and j are independent conditioned on n. □

Proving parts (i), (ii) and (iii) proves Proposition A.1.1, that the conditional independence structure of Σ^q is given by the tree T^q. For a leaf node a and its neighbor b in T^*, the decomposition $\Sigma^o = \Sigma^q + D^q$ which results in the exchange to nodes a and b is as follows:

$$\Sigma^q_{ij} = \begin{cases} \Sigma^*_{ij} - \frac{1}{\Omega^*_{aa}} & \text{if i = j = a} \\ \Sigma^*_{ij} + c^i_1 & 0 < c^i_1 < D^*_{ij} \text{ if } i = j = b \\ \Sigma^*_{ij} & \text{otherwise,} \end{cases}$$

$$D^q_{ii} = \begin{cases} D^*_{ii} + \frac{1}{\Omega^*_{aa}} & \text{if } i = a \\ D^*_{ii} - c^i_1 & \text{if } i = b \\ D^*_{ii} & \text{otherwise,} \end{cases}$$

□

Thus far, we have only considered the case when $\mathcal{S}_q$ has just one node. This analysis directly extends to the case when $\mathcal{S}_q$ has more than one nodes. The Σ^q and D^q matrices in that case are as follows:

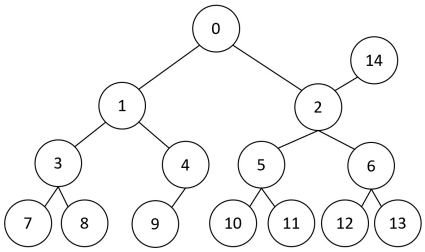
$$\Sigma^q_{ij} = \begin{cases} \Sigma^*_{ij} - \frac{1}{\Omega^*_{ij}} & \text{if } i = j \in \mathcal{S}^q \\ \Sigma^*_{ij} + c^i_1 & \text{if } i = j \in Neighbor(\mathcal{S}^q) \\ \Sigma^*_{ij} & \text{otherwise,} \end{cases}$$

$$D^q_{ii} = \begin{cases} D^*_{ii} + \frac{1}{\Omega^*_{ii}} & \text{if } i \in \mathcal{S}^q \\ D^*_{ii} - c^i_1 & \text{if } i \in Neighbor(\mathcal{S}^q) \\ D^*_{ii} & \text{otherwise,} \end{cases}$$

where $Neighbor(\mathcal{S}^q)$ is the set of neighbor nodes of all the nodes in $\mathcal{S}^q$. Also, c^i_1 is chosen such that $0 < c^i_1 < D^*_{ii}$. This completes the proof of Theorem 1. □

A.2 Proof of Theorem 2

We prove this theorem by proving that the off diagonal terms of covariance matrix are enough to determine the structure of the underlying tree up to the equivalence set $\mathcal{T}_{T^*}$. The main building block of this proof and of the algorithm presented in Section 5 is to categorize any set of 4 nodes as a star shape or a non-star shape. Moreover, if it is a non star star shape we further divide the set of 4 nodes in half forming 2 pairs of nodes.



Node Set	Classification	i_k, pairing (If Applicable)
{0, 1, 3, 7}	Non-Star	1, {{0,1}, {3,7}}
{7, 8, 9, 10}	Non-Star	1, {{7,8}, {9,10}}
{14, 2, 10, 11}	Non-Star	2, {{14,2}, {10,11}}
{7, 9, 1, 6}	Star	N.A.
{1, 14, 10, 6}	Star	N.A.

Figure A.1: Examples of classification of 4 nodes as star shape or non-star shape.

Definition A.2.1.

- Four nodes $\{i_1, i_2, i_3, i_4\}$ form a **non-star shape** if there exists a node i_k in the tree T^*[1] such that exactly two nodes among the four lie in the same connected component of $T^* \setminus i_k$.
- If $\{i_1, i_2, i_3, i_4\}$ does not form a non-star shape, we say they form a **star shape**.

It is easy to see that in the event that a set of 4 nodes forms a non star, there exists a grouping such that the 2 nodes in the same connected component form the first pair and the other 2 nodes form the second pair. Examples of star shape and non-star shape are presented in Figure A.1. This categorization is done using only the off-diagonal elements of the covariance matrix, hence this property remains invariant to diagonal perturbations, that is, every set of 4 nodes falls in the same category in any tree obtained from the decomposition of $\Sigma^o = \Sigma' + D'$ as $\Sigma'_{ij} = \Sigma^*_{ij}\ \forall i \neq j$.

The proof of this theorem is split in 3 parts:

(i) Prove that it is possible to categorize any set of 4 nodes as star shape or non-star shape using only off diagonal elements of the covariance matrix.

[1]Note that nothing prevents i_k to be one of the four nodes.

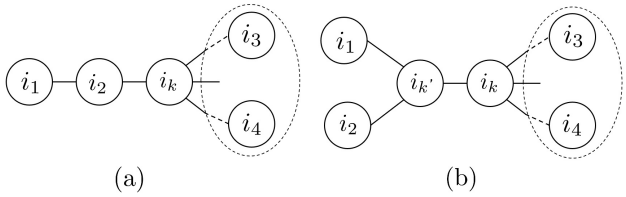


Figure A.2: Conditional independence for non-star shape

(ii) Prove that this categorization of 4 nodes completely defines all the possible partitions of the original tree in 2 connected components such that the connected components have at least 2 node.

(iii) Prove that these partitions of a tree into connected components completely define the tree structure up to the equivalence set $\mathcal{T}_{T^*}$.

A.2.1 Proof of Part (i) - Categorization of 4 nodes as star/non-star shape:

We first state the conditions using only off-diagonal elements for a set of 4 nodes to be categorized as non-star shape. Assume that a set of 4 nodes $\{i_1, i_2, i_3, i_4\}$ satisfy the definition of a non-star shape such that nodes i_1 and i_2 form one pair and i_3 and i_4 form the second pair. This is true if and only if:

$$\begin{aligned} &\frac{\Sigma^*_{i_1 i_3}}{\Sigma^*_{i_1 i_4}} = \frac{\Sigma^*_{i_2 i_3}}{\Sigma^*_{i_2 i_4}}, \\ &\frac{\Sigma^*_{i_2 i_1}}{\Sigma^*_{i_3 i_1}} \neq \frac{\Sigma^*_{i_2 i_4}}{\Sigma^*_{i_3 i_4}} \text{ and} \\ &\frac{\Sigma^*_{i_2 i_1}}{\Sigma^*_{i_4 i_1}} \neq \frac{\Sigma^*_{i_2 i_3}}{\Sigma^*_{i_3 i_4}}. \end{aligned} \tag{A.10}$$

The first equality and the second inequality imply the last inequality. When nodes $\{i_1, i_2, i_3, i_4\}$ form a non star shape, they either satisfy a conditional independence structure shown in Fig-

ure A.2(a) or A.2(b) for some nodes i_k and $i_{k'}$.

For Figure A.2(a), the following conditional independence relations hold:

$$i_1 \perp i_3, i_4 | i_2, \tag{A.11}$$

$$i_3 \not\perp i_4 | i_2. \tag{A.12}$$

Using Lemma A.1.3, we get the following conditions for the conditional independence relation in Equations (A.11) and (A.12):

$$\Sigma^*_{i_2 i_2} = \frac{\Sigma^*_{i_1 i_2}\Sigma^*_{i_3 i_2}}{\Sigma^*_{i_1 i_3}} = \frac{\Sigma^*_{i_1 i_2}\Sigma^*_{i_4 i_2}}{\Sigma^*_{i_1 i_4}} \neq \frac{\Sigma^*_{i_3 i_2}\Sigma^*_{i_4 i_2}}{\Sigma^*_{i_3 i_4}}. \tag{A.13}$$

Using Equation (A.13) we get the relations in Equation (2.3).

For Figure A.2(b), the following conditional independence relations hold:

$$i_1 \perp i_3, i_4 | i_{k'}, \tag{A.14}$$

$$i_2 \perp i_3, i_4 | i_{k'}, \tag{A.15}$$

$$i_3 \not\perp i_4 | i_{k'}. \tag{A.16}$$

Using Lemma A.1.3, we get the following conditions for the conditional independence relation in Equations (A.14), (A.15) and (A.16):

$$\Sigma^*_{i_{k'} i_{k'}} = \frac{\Sigma^*_{i_1 i_{k'}}\Sigma^*_{i_3 i_{k'}}}{\Sigma^*_{i_1 i_3}} = \frac{\Sigma^*_{i_1 i_{k'}}\Sigma^*_{i_4 i_{k'}}}{\Sigma^*_{i_1 i_4}} = \frac{\Sigma^*_{i_2 i_{k'}}\Sigma^*_{i_3 i_{k'}}}{\Sigma^*_{i_2 i_3}} = \frac{\Sigma^*_{i_2 i_{k'}}\Sigma^*_{i_4 i_{k'}}}{\Sigma^*_{i_2 i_4}} \neq \frac{\Sigma^*_{i_3 i_{k'}}\Sigma^*_{i_4 i_{k'}}}{\Sigma^*_{i_3 i_4}}. \tag{A.17}$$

Using Equation (A.17), we get the conditions in Equation (2.3). Note that for both the cases in Figure A.2, the Equation (2.3) remains the same if i_1 and i_2 exchange positions.

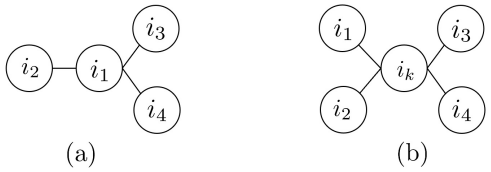


Figure A.3: Conditional independence for star shape.

Next, we state the conditions using only off-diagonal elements for a set of 4 nodes to be categorized as a star shape. Assume that a set of 4 nodes $\{i_1, i_2, i_3, i_4\}$ satisfy the definition of a star shape. This is true if and only if:

$$\begin{aligned} \frac{\Sigma^*_{i_1 i_3}}{\Sigma^*_{i_1 i_4}} &= \frac{\Sigma^*_{i_2 i_3}}{\Sigma^*_{i_2 i_4}}, \\ \frac{\Sigma^*_{i_2 i_1}}{\Sigma^*_{i_3 i_1}} &= \frac{\Sigma^*_{i_2 i_4}}{\Sigma^*_{i_3 i_4}} \text{ and} \\ \frac{\Sigma^*_{i_2 i_1}}{\Sigma^*_{i_4 i_1}} &= \frac{\Sigma^*_{i_2 i_3}}{\Sigma^*_{i_3 i_4}}. \end{aligned} \tag{A.18}$$

First 2 equalities imply the third equality. Any set of 4 nodes $\{i_1, i_2, i_3, i_4\}$ can form a star structure only if their conditional independence relation is given by Figure A.3(a) or A.3(b) for some node i_k. For Figure A.3(a), the conditional independence relations are given as:

$$i_2 \perp i_3, i_4 | i_1, \tag{A.19}$$

$$i_3 \perp i_4 | i_1. \tag{A.20}$$

Using Lemma A.1.3, we get the following for these conditional independence relations in Equations (A.19) and (A.20):

$$\Sigma^*_{i_1 i_1} = \frac{\Sigma^*_{i_1 i_2}\Sigma^*_{i_1 i_3}}{\Sigma^*_{i_2 i_3}} = \frac{\Sigma^*_{i_1 i_2}\Sigma^*_{i_1 i_4}}{\Sigma^*_{i_2 i_4}} = \frac{\Sigma^*_{i_1 i_4}\Sigma^*_{i_1 i_3}}{\Sigma^*_{i_4 i_3}}. \tag{A.21}$$

Equation (A.21) implies Equation (2.4).

For Figure A.3(b), the conditional independence relations are given as:

$$i_1 \perp i_2, i_3, i_4 | i_k, \tag{A.22}$$

$$i_2 \perp i_3, i_4 | i_k, \tag{A.23}$$

$$i_3 \perp i_4 | i_k. \tag{A.24}$$

Using Lemma A.1.3, we get the following for the conditional independence relations in Equations (A.22), (A.23) and (A.24):

$$\Sigma^*_{i_k i_k} = \frac{\Sigma^*_{i_1 i_k}\Sigma^*_{i_2 i_k}}{\Sigma^*_{i_1 i_2}} = \frac{\Sigma^*_{i_1 i_k}\Sigma^*_{i_3 i_k}}{\Sigma^*_{i_1 i_3}} = \frac{\Sigma^*_{i_1 i_k}\Sigma^*_{i_4 i_k}}{\Sigma^*_{i_1 i_4}} = \frac{\Sigma^*_{i_2 i_k}\Sigma^*_{i_3 i_k}}{\Sigma^*_{i_2 i_3}} = \frac{\Sigma^*_{i_2 i_k}\Sigma^*_{i_4 i_k}}{\Sigma^*_{i_2 i_4}} = \frac{\Sigma^*_{i_3 i_k}\Sigma^*_{i_4 i_k}}{\Sigma^*_{i_3 i_4}}. \tag{A.25}$$

Equation (A.25) implies Equation (2.4).

Hence using only the off diagonal terms, checking the conditions in Equations (2.3) and (2.4), any set of 4 nodes can be classified as a star shape or non-star shape. □

A.2.2 Proof of Part (ii) - Partitioning of the tree in 2 connected components:

We prove this by presenting an explicit algorithm to obtain a specific partition of the original tree T^*, which would also be a valid partition of T', using the categorization of any set of 4 nodes as a star shape or non-star shape. This procedure can be performed with different initializations to obtain all the possible partitions.

Let $\mathcal{A}$ denote the set of all the nodes in T^*.

Definition A.2.2. A **subtree** $\mathcal{B}$ of a tree T^* is a set of nodes such that $\mathcal{B}$ and $\mathcal{A} \setminus \mathcal{B}$ form a connected component in T^*. The pair of subtrees $\mathcal{B}$ and $\mathcal{A} \setminus \mathcal{B}$ are called complementary subtrees.

For any set of 4 nodes $\{i_1, i_2, i_3, i_4\}$ that form a non-star shape such that nodes i_1 and i_2 form a pair, we obtain the smallest subtree containing i_1 and i_2 by Algorithm 2. Basically, we fix i_1, i_2 and i_3 and scan through all the remaining nodes to form a set of 4 nodes and check if it forms a star or non-star shape. If this set of 4 nodes forms a star shape or forms a non-star shape such that the scanned node pairs with i_1 or i_2, we put it in group 1, otherwise, we put it in group 2. Once we are done scanning through all the nodes, group 1 gives the smallest subtree and group 2 gives its complementary subtree.

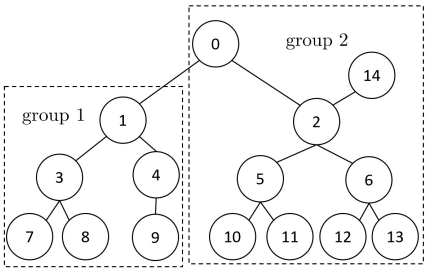


Figure A.4: Suppose $i_1 = 7$, $i_2 = 9$ and $i_3 = 5$. If j is in group 2, $\{i_1, i_2, i_3, j\}$ is categorized as a non star and j pairs with i_3. If j is in group 1, $\{i_1, i_2, i_3, j\}$ is either categorized as a star or it is categorized as a non star and j pairs with i_1 or i_2.

Algorithm 2 Partition all the nodes in complementary subtrees.

Input - Observed Covariance Matrix (Σ^o), Set of 4 nodes($\{i_1, i_2, i_3, i_4\}$)
Output - The smallest subtree containing i_1 and i_2($group1$) and the complementary subtree ($group2$).

```
1: procedure SmallestSubtree(Σ^o, {i_1, i_2, i_3, i_4})
2:     n_rows ← size(Σ^o, 1)
3:     index ← {i_1, i_2, i_3, 0}
4:     for j = 1 to n_rows do
5:         if j in group1 or group2 then
6:             continue
7:         end if
8:         index[4] = j
9:         status, pair1, pair2 ← IsStarShape(index, Σ^o)
10:        if status then                ▷ If {i_1, i_2, i_3, j} forms a star shape, add j to group1.
11:            group1.append(j)
12:        else
13:            if j pairs with index[3] then          ▷ If j pairs with i_3, add j to group2.
14:                group2.append(j)
15:            else
16:                group1.append(j)                   ▷ Otherwise add j to group1.
17:            end if
18:        end if
19:    end for
20:    return group1, group2
21: end procedure
```

Proof of Correctness of Algorithm 2

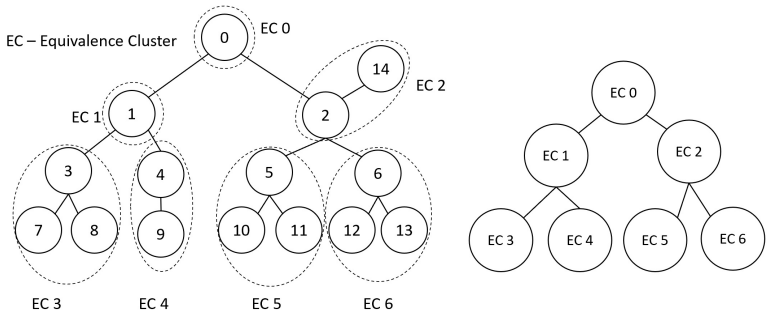
Consider the tree T^*. We denote the smallest subtree containing nodes i_1 and i_2 by $\mathcal{B}$. Let $i_{k'}$ denote the node in $\mathcal{B}$ that has an edge with the connected component formed by $\mathcal{A} \setminus \mathcal{B}$. Let i_k be the node in $\mathcal{A} \setminus \mathcal{B}$ that has an edge with a node in $\mathcal{B}$. In this case i_k is a node such that nodes i_1 and i_2 lie in the same connected component of $T^* \setminus i_k$. By the definition of non-star shape, i_3 cannot be in $\mathcal{B}$. Also, a node j can be in $\mathcal{A} \setminus \mathcal{B}$ if and only if nodes $\{i_1, i_2, i_3, j\}$ are non star and j pairs with i_3 as nodes i_1 and i_2 still lie in the same connected component of $T^* \setminus i_k$. This is illustrated in Figure A.4.
Using different i_1 and i_2, we get all the possible partitions of the tree T^*.

A.2.3 Proof of Part (iii) - Recovering the tree up to unidentifiability using tree partitions

Before going to the proof of this part, we define the terms equivalence cluster, cluster tree, cluster subtrees, complementary cluster subtrees and the root of a cluster subtree as follows:

Definition A.2.3. A set containing an internal node and all the leaf nodes connected to it forms an **equivalence cluster**. We say that there is an edge between two equivalence clusters if there is an edge between any node in one equivalence cluster and any node in the other equivalence cluster. An equivalence cluster which has an edge with at most one more equivalence cluster is called a leaf equivalence cluster.

Definition A.2.4. A tree with equivalence clusters as vertices and edges between equivalence clusters as the edges is called a **cluster tree**.



(a) (b)

Figure A.5: (a) Equivalence clusters for the given tree. (b) The cluster tree with equivalence clusters as vertices.

Example of equivalence clusters and a cluster tree are presented in Figure A.5. The cluster tree completely defines the set $\mathcal{T}_{T^*}$.

Definition A.2.5. A **cluster subtree** is a set where the equivalence clusters are plugged in for the corresponding nodes in a subtree. Complementary cluster subtrees are the subtrees obtained when this is done for a pair of complementary subtrees.

Definition A.2.6. The **root of a cluster subtree** is the equivalence cluster that has an edge with the complementary cluster subtree.

To prove this theorem we show that the partitions obtained in part (ii) completely define the cluster tree. We call the subtrees obtained from part (ii) input subtrees. Note that each input subtree has at least 2 nodes. We prove this in 2 steps:

(i) The input subtrees define the equivalence clusters.

(ii) The input subtrees define the edges between the equivalence clusters.

Algorithm to find equivalence clusters

The algorithm to find the equivalence clusters takes all the input subtrees and performs the following steps:

1. Initialize the set of discovered equivalence clusters as an empty set.
2. Identify one input subtree which does not have a subset of nodes forming another input subtree. This input subtree forms an equivalence cluster. Append it to the list of equivalence clusters.
3. Construct trimmed subtrees by removing the equivalence cluster from the input subtrees.
4. Repeat steps 2 and 3 with trimmed subtrees as input subtrees.

Proof of Correctness:

We prove the correctness of this algorithm by induction on the number of equivalence clusters.

Base Case ($k = 1$):

When there is 1 equivalence cluster, there is 1 input subtree and it is the equivalence cluster.

Inductive Step:

Assume the algorithm works for a tree with k or less equivalence clusters. We prove that the algorithm works for a tree with $k + 1$ equivalence clusters.

Relabeling if necessary, assume that $k + 1$ is a leaf equivalence cluster. Hence it forms a subtree and no subset of the equivalence cluster can form a subset of another input subtree (as the smallest input subtree which contains at least 2 of these nodes is the whole equivalence

cluster). Thus in Step 2, $k+1$ is recognized as an equivalence cluster.

By trimming in Step 3, we remove the $k+1^{st}$ equivalence cluster from all the subtrees. Hence, we are left with a tree with k equivalence clusters. By inductive assumption, the algorithm can find these k equivalence clusters. Therefore, the algorithm finds all the $k+1$ equivalence clusters.

Algorithm to find the edges between equivalence clusters

For this part we identify the root of every cluster subtree as follows:

An equivalence cluster is the root of a cluster subtree if and only if, upon its removal, the remaining elements can be written as a union of smaller cluster subtrees which are a subset of the original cluster subtree.

To prove this claim, assume that we remove an equivalence cluster other than the root. In that case the root must have an edge with the complementary cluster subtree and hence it cannot be obtained by a union of smaller cluster subtrees which are a subset of the original cluster subtrees.

The algorithm to find the edges between equivalence clusters performs the following steps:

1. Initialize the set of edges as a null set and the set of unexplored complementary cluster subtrees as the set of all the complementary cluster subtrees.
2. Select a pair of complementary cluster subtrees from the set of unexplored complementary cluster subtrees.
3. Find the root nodes of both the cluster subtrees and append an edge between the two roots in the set of edges.

4. Trim the currently selected cluster subtrees from all the cluster subtrees in the unexplored set for which the currently explored cluster subtrees are a subset(this also deletes the currently selected cluster subtrees from the unexplored set). Repeat Steps 2, 3 and 4 with the trimmed cluster subtrees till the unexplored set is empty.

Proof of Correctness:

We prove the correctness of this algorithm by induction on the number of equivalence clusters.

Base Case ($k = 2$): In this case there are 2 cluster subtrees which are complementary cluster subtrees. Both of them have 1 equivalence cluster which is also the root. Hence the algorithm finds the edge between the two cluster subtrees.

Inductive Step: Suppose the algorithm works for a tree with k or less equivalence clusters. We prove that the algorithm works for a tree with $k + 1$ equivalence clusters.

Relabeling if necessary, assume that $k + 1$ is a leaf equivalence cluster. Hence there exists a pair of complementary cluster subtrees where one cluster subtree contains the $k + 1$ equivalence cluster and the other cluster contains the first k equivalence cluster. Hence the edge of the $(k + 1)^{st}$ equivalence cluster is added to the list of edges. Once this edge is recognized, the $(k + 1)^{st}$ equivalence cluster is trimmed and the algorithm correctly finds the edges of the remaining cluster tree by the inductive assumption.

Hence the input subtrees completely define the equivalence clusters and the edges between them. This completes the proof of theorem 2. □

A.3 Proof of Theorem 4

To prove this claim, we consider the decomposition of $\Sigma^o = \Sigma' + D'$ such that the conditional independence structure T' for Σ' has leaf node b and its neighbor node a. We show that $\Omega'_{bb} < |\Omega'_{ab}|$, that is, the leaf node b in T' violates the constraint. Hence, any decomposition of Σ^o which results in an exchange of a leaf node with its neighbor is infeasible. Therefore, the problem becomes identifiable.

Relabeling if necessary, assume that node n is a leaf node connected to node $n-1$ in T^*. Recall that the decomposition of $\Sigma^o = \Sigma' + D'$ from Proposition A.1.1 to obtain a tree structure T' in which node $n-1$ is a leaf node connected to node n is given by:

$$\Sigma'_{ij} = \begin{cases} \Sigma^*_{ij} - \frac{1}{\Omega^*_{ij}} & \text{if } i = j = n \\ \Sigma^*_{ij} + c & 0 < c < D^*_{n-1n-1} \text{ if } i = j = n-1 \\ \Sigma^*_{ij} & \text{otherwise.} \end{cases}$$

We derive the expression of $\Omega' = (\Sigma')^{-1}$. We denote B_1 and B_2 as follows:

$$B_1 = \begin{cases} c & 0 < c < D^*_{n-1n-1} \text{ if } i = j = n-1 \\ 0 & \text{otherwise} \end{cases},$$

$$B_2 = \begin{cases} -\frac{1}{\Omega^*_{nn}} & \text{if } i = j = n \\ 0 & \text{otherwise} \end{cases}.$$

This gives us $\Sigma' = \Sigma^* + B_1 + B_2$. Hence Σ' is Σ^* plus a rank 2 matrix. To calculate its inverse, we first evaluate:

$$\begin{aligned} (\Sigma^* + B_1)^{-1} &= \Omega^* - \frac{1}{1 + tr(\Omega^* B_1)} \Omega^* B_1 \Omega^* \\ &= \Omega^* - \frac{c\Omega^*_{:,n-1}\Omega^*_{n-1,:}}{1 + c\Omega^*_{n-1n-1}}. \end{aligned} \tag{A.26}$$

We next evaluate Ω' as follows:

$$\Omega' = (\Sigma^* + B_1 + B_2)^{-1} = (\Sigma^* + B_1)^{-1} - \frac{1}{1 + tr((\Sigma^* + B_1)^{-1} B_2)} (\Sigma^* + B_1)^{-1} B_2 (\Sigma^* + B_1)^{-1}.$$

This expression can be simplified by substituting the value of $(\Sigma^* + B_1)^{-1}$ from Equation (A.26) to arrive at:

$$\Omega' = \Omega^* + \frac{(1 + c\Omega^*_{n-1n-1})}{c(\Omega^*_{n-1n})^2} \Omega^*_{:,n} \Omega^*_{n,:} - \frac{1}{\Omega^*_{n-1n}} (\Omega^*_{:,n-1} \Omega^*_{n,:} + \Omega^*_{:,n} \Omega^*_{n-1,:}). \tag{A.27}$$

Now we look at the terms in positions $(n-1, n-1)$ and $(n-1, n)$ of Ω'.

$$\begin{aligned}
\Omega'_{n-1n-1} &= \Omega^*_{n-1n-1} + \frac{(1 + c\Omega^*_{n-1n-1})}{c} - 2\Omega^*_{n-1n-1} \\
&= \frac{1}{c}. \\
\Omega'_{n-1n} &= \Omega^*_{n-1n} + \frac{(1 + c\Omega^*_{n-1n-1})}{c\Omega^*_{n-1n}} \Omega^*_{nn} - \Omega^*_{n-1n} - \frac{\Omega^*_{nn} \Omega^*_{n-1n-1}}{\Omega^*_{n-1n}} \\
&= \frac{\Omega^*_{nn}}{c\Omega^*_{n-1n}}.
\end{aligned}$$

By the original assumption we have $\Omega^*_{nn} > |\Omega^*_{n-1n}|$, hence $\Omega'_{n-1n-1} < |\Omega'_{n-1n}|$. Therefore the leaf node $n-1$ in T' violates the additional constraint and hence this decomposition of Σ^o is infeasible. Extending the argument, any decomposition of Σ^o which results in a tree T' in which leaf node of T^* exchanges position with its neighbor is infeasible. Hence T^* and T' have the same structure. □

A.4 Proof of Theorem 6

To prove this theorem, we consider Σ' such that the conditional independence structure has b as the leaf node and a as its neighbor. Rest of the struture is the same as T^*. We

find a lower bound on the minimum eigenvalue of Σ', λ'_{min}. If this lower bound is greater than λ_{min}, this implies that there exists a feasible decomposition which has conditional independence structure different from T^*.

In order to lower bound the minimum eigenvalue of Σ', we upper bound the maximum eigenvalue of Ω'. We do this using a corollary of Gerschgorin's Theorem. We use the result that the maximum eigenvalue of Ω' is upper bounded by the maximum of the sum of absolute values of all the row entries:

$$\frac{1}{\lambda'_{min}} \leq \max_i \Big(\sum_{j=1}^{n} |\Omega'_{ij}| \Big). \tag{A.28}$$

From the expression of Ω' stated in Equation (A.27) (by relabeling the nodes n and $n-1$ as nodes a and b respectively), we have:

$$\sum_{j=1}^{n} |\Omega'_{ij}| = \begin{cases} \frac{1}{c}\left(\frac{(\Omega^*_{aa})^2}{(\Omega^*_{ab})^2} + \frac{\Omega^*_{aa}}{\Omega^*_{ab}}\right) + \frac{\Omega^*_{aa}(\Omega^*_{aa}\Omega^*_{bb} - (\Omega^*_{ab})^2)}{(\Omega^*_{ab})^2} + \sum_{\substack{j=1 \\ j\neq a,b}}^{n} \frac{\Omega^*_{aa}|\Omega^*_{qj}|}{|\Omega^*_{ab}|} & \text{if } i = a, \\ \frac{1}{c}\left(1 + \frac{\Omega^*_{aa}}{\Omega^*_{ab}}\right) & \text{if } i = b. \\ \left(\sum_{\substack{j=1 \\ j\neq a,b}}^{n} |\Omega^*_{ij}| + \frac{\Omega^*_{aa}|\Omega^*_{qi}|}{|\Omega^*_{ab}|} \right) & \text{otherwise.} \end{cases}$$

Using the definitions in Equation 6, we can rewrite the upper bound in Equation (A.28) as follows:

$$\frac{1}{\lambda'_{min}} \leq \max\left(\frac{e^{ab}}{c}, \frac{f^{ab}}{c} + g^{ab}, h^{ab}\right).$$

Rewriting this as:

$$\frac{1}{\lambda'_{min}} \leq \begin{cases} \frac{e^{ab}}{c} & \text{if } c \leq \frac{e^{ab}-f^{ab}}{g^{ab}} \\ \frac{f^{ab}}{c} + g^{ab} & \text{if } \frac{e^{ab}-f^{ab}}{g^{ab}} < c \leq \frac{f^{ab}}{h^{ab}-g^{ab}} \\ h^{ab} & \textit{otherwise.} \end{cases}$$

First, let us concentrate on the first case. For unidentifiability, we need:

$$c \geq e^{ab}\lambda_{min}.$$

To remain in the first case, we need $c \leq \frac{e^{ab}-f^{ab}}{g^{ab}}$. Therefore, if $\lambda_{min} \leq \frac{(e^{ab}-f^{ab})}{e^{ab}g^{ab}}$ and $D^*_{bb} \geq e^{ab}\lambda_{min}$, there would exist a feasible value of c which allows node a and b to switch positions.

Next we look at the second case. If $\lambda_{min} < \frac{1}{g^{ab}}$, for unidentifiability, we need:

$$c \geq \frac{f^{ab}}{1/\lambda_{min} - g^{ab}}.$$

To remain in the second case, we need $c \leq \frac{f^{ab}}{h^{ab}-g^{ab}}$. Therefore, if $\lambda_{min} < \frac{1}{h^{ab}}$ and $D^*_{bb} \geq \frac{f^{ab}}{1/\lambda_{min}-g^{ab}}$, there would exist a feasible value of c which allows node a and b to switch positions. If $\lambda_{min} > \frac{1}{g^{ab}}$, nothing can be said about unidentifiability. To enter the third case, we need $\lambda_{min} > \frac{1}{h^{ab}}$ which would again imply that nothing could be said about identifiability.

Appendix B

Robust Estimation of Tree Structured Ising Models

B.1 Proof of Lemma 3.3.5

We prove this by induction on the number of nodes k in the path $(X_{i_1} \to X_{i_2} \to X_{i_3} \cdots \to X_{i_k})$ for any 2 nodes X_{i_1}, X_{i_k}.

Base Case $k = 3$:

The path is $(X_{i_1} \to X_{i_2} \to X_{i_3})$, therefore we have $X_{i_1} \perp X_{i_3}|X_{i_2}$. For random variables with a support size of 2, this is true if and only if they are conditionally uncorrelated, that is,

$$\mathbb{E}[X_{i_1}X_{i_3}|X_{i_2}] = \mathbb{E}[X_{i_1}|X_{i_2}]\mathbb{E}[X_{i_3}|X_{i_2}]. \tag{B.1}$$

$\mathbb{E}[X_{i_1}|X_{i_2}]$ is linear in X_{i_2} since the support size of X_{i_2} is 2 and therefore we need to need to fit only 2 points $\mathbb{E}[X_{i_1}|X_{i_2} = 1]$ and $\mathbb{E}[X_{i_1}|X_{i_2} = -1]$ to completely represent the conditional expectation. Therefore the linear least square error (LLSE) estimator of X_{i_1} given X_{i_2} is also the minimum mean squared estimator $\mathbb{E}[X_{i_1}|X_{i_2}]$. Utilizing the standard result for LLSE, we have:

$$\mathbb{E}[X_{i_1}|X_{i_2}] = \mathbb{E}[X_{i_1}] + \Sigma_{i_1,i_2}\Sigma^{-1}_{i_2,i_2}(X_{i_2} - \mathbb{E}[X_{i_2}]). \tag{B.2}$$

Similarly we have:

$$\mathbb{E}[X_{i_3}|X_{i_2}] = \mathbb{E}[X_{i_3}] + \Sigma_{i_3,i_2}\Sigma^{-1}_{i_2,i_2}(X_{i_2} - \mathbb{E}[X_{i_2}]). \tag{B.3}$$

Substituting $\mathbb{E}[X_{i_1}|X_{i_2}]$ and $\mathbb{E}[X_{i_3}|X_{i_2}]$ from Equations (B.2) and (B.3) in Equation (B.1) we get:

$$\begin{aligned}
\mathbb{E}[X_{i_1}X_{i_3}|X_{i_2}] =& \mathbb{E}[X_{i_1}]\mathbb{E}[X_{i_3}] + \mathbb{E}[X_{i_1}]\Sigma_{i_3,i_2}\Sigma^{-1}_{i_2,i_2}(X_{i_2} - \mathbb{E}[X_{i_2}]) + \\
& \mathbb{E}[X_{i_3}]\Sigma_{i_1,i_2}\Sigma^{-1}_{i_2,i_2}(X_{i_2} - \mathbb{E}[X_{i_2}]) + \\
& \Sigma_{i_1,i_2}\Sigma_{i_3,i_2}(\Sigma^{-1}_{i_2,i_2}(X_{i_2} - \mathbb{E}[X_{i_2}]))^2 \\
\mathbb{E}[X_{i_1}X_{i_3}] =& \mathbb{E}[\mathbb{E}[X_{i_1}X_{i_3}|X_{i_2}]] \\
=& \mathbb{E}[X_{i_1}]\mathbb{E}[X_{i_3}] + \Sigma_{i_1,i_2}\Sigma_{i_3,i_2}\Sigma^{-1}_{i_2,i_2}.
\end{aligned}$$

Therefore we get $\Sigma_{i_1,i_3}\Sigma_{i_2,i_2} = \Sigma_{i_1,i_2}\Sigma_{i_3,i_2}$ which implies $\rho_{i_1i_3} = \rho_{i_1i_2}\rho_{i_2i_3}$.

Inductive Case:

Let the statement be true for any path involving k nodes. For a path $(X_{i_1} \to X_{i_2} \to X_{i_3} \cdots \to X_{i_{(k+1)}})$ we have $X_{i_1} \perp X_{i_{(k+1)}}|X_{i_k}$. Therefore the same calculation as the base case holds true by replacing X_{i_2} by X_{i_k} and X_{i_3} by $X_{i_{(k+1)}}$. Therefore $\rho_{i_1i_{(k+1)}} = \rho_{i_1i_k}\rho_{i_ki_{(k+1)}}$. By the inductive assumption, $\rho_{i_1i_k} = \prod_{l=2}^{k}\rho_{i_{l-1},i_l}$, therefore, $\rho_{i_1i_{(k+1)}} = \prod_{l=2}^{k+1}\rho_{i_{l-1},i_l}$.

B.2 Proof of Covariance of noisy variables.

Lemma B.2.1. *Consider 2 Random variables X_i and X_j with support on $\{-1, 1\}$ whose covariance is denoted by $\Sigma_{i,j}$. Now consider the noisy version of these random variables X_i^e*

and X_j^e whose covariance is denoted by $\Sigma'_{i,j}$. Then we have:

$$\begin{aligned}\mathbb{E}[X_i^e] &= (1-2q_i)\mathbb{E}[X_i] \\ \Sigma'_{i,j} &= (1-2q_i)(1-2q_j)\Sigma_{i,j}\end{aligned}$$

Proof. By the noise model we have:

$$\begin{aligned}\mathbb{E}[X_i^e] &= (1-q_i)\mathbb{E}[X_i] + q_i\mathbb{E}[-X_i] \\ \mathbb{E}[X_i^e] &= (1-2q_i)\mathbb{E}[X_i].\end{aligned} \tag{B.4}$$

We also have:

$$\begin{aligned}\mathbb{E}\big[X_i^e X_j^e\big] =&(1-q_i)(1-q_j)\mathbb{E}[X_iX_j] + (1-q_j)q_i\mathbb{E}[-X_iX_j] + \\ &(1-q_i)q_j\mathbb{E}[-X_iX_j] + q_iq_j\mathbb{E}[X_iX_j] \\ =&(1-2q_i)(1-2q_j)\mathbb{E}[X_iX_j].\end{aligned} \tag{B.5}$$

Therefore,

$$\begin{aligned}\Sigma'_{i,j} &= \mathbb{E}\big[X_i^e X_j^e\big] - \mathbb{E}[X_i^e]\mathbb{E}\big[X_j^e\big] \\ &= (1-2q_i)(1-2q_j)(\mathbb{E}[X_iX_j] - \mathbb{E}[X_i]\mathbb{E}[X_j]) \\ &= (1-2q_i)(1-2q_j)\Sigma_{i,j}\end{aligned} \tag{B.6}$$

□

We can use Equation (B.4) to calculate the variance of every random variable in terms of the variance of its noisy counterpart as follows:

$$\begin{aligned}\Sigma_{i,i} =&1 - \mathbb{E}[X_i]^2 \\ =&1 - \frac{\mathbb{E}[X_i^e]^2}{(1-2q_i)^2} \\ =&1 - \frac{1-\Sigma'_{i,i}}{(1-2q_i)^2}\end{aligned} \tag{B.7}$$

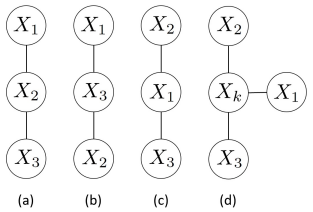


Figure B.1: Different possible configurations of any set of 3 nodes.

B.3 Proof that the Quadratic gives a valid solution

Consider the quadratic in Equation (3.2). We prove that this equation always has a valid solution $q_1 < 0.5$ for any set of 3 nodes in a tree structured graphical model.

Whenever $0 < 1 - \Sigma'_{1,1} + \frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}} < 1$, the solution is of the form $q_1 = \eta, 1 - \eta$ where $0 \leq \eta < 0.5$. From Equations (B.6) and (B.7), we have:

$$1 - \Sigma'_{1,1} + \frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}} = (1 - 2q_1^2)(1 - \Sigma_{1,1} + \frac{\Sigma_{1,2}\Sigma_{1,3}}{\Sigma_{2,3}}). \tag{B.8}$$

The different possible configurations of any 3 nodes X_1, X_2 and X_3 in any tree structured graphical model are shown in Figure B.1. For case (a) we have $\Sigma_{2,2}\Sigma_{1,3} = \Sigma_{1,2}\Sigma_{2,3}$ by Lemma 3.3.5. This gives us:

$$1 - \Sigma'_{1,1} + \frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}} = (1 - 2q_1)^2(1 - \Sigma_{1,1} + \frac{\Sigma_{1,2}^2}{\Sigma_{2,2}}).$$

Using the assumption that the absolute value of correlation is upper bounded away from 1 and lower bounded away from 0, we have $0 < \Sigma_{1,2}^2 < \Sigma_{1,1}\Sigma_{2,2}$. Also, $0 < \Sigma_{1,1} \leq 1$ and $0 < (1 - 2q_1)^2 \leq 1$. Therefore, for case (a), $0 < 1 - \Sigma'_{1,1} + \frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}} < 1$ and the quadratic equation has valid roots. By symmetry, the quadratic equation gives valid roots for case (b) too.

Case (c) is the underlying truth, therefore the quadratic equation recovers the true underlying error.

For case(d), we have $\Sigma_{k,k}\Sigma_{1,3} = \Sigma_{1,k}\Sigma_{3,k}$, $\Sigma_{k,k}\Sigma_{1,2} = \Sigma_{1,k}\Sigma_{2,k}$ and $\Sigma_{k,k}\Sigma_{2,3} = \Sigma_{2,k}\Sigma_{3,k}$. This gives us:

$$1 - \Sigma'_{1,1} + \frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}} = (1-2q_1)^2(1 - \Sigma_{1,1} + \frac{\Sigma^2_{1,k}}{\Sigma_{k,k}}). \tag{B.9}$$

The same arguments as case (a) hold true for case (d) with node 2 replaced by node k. Therefore, the quadratic has a valid solution in this case too.

B.4 Proof of Lemma 3.3.6, Lemma 3.3.7 and Star/Non-star Condition for Generic Trees

B.4.1 Proof of Lemma 3.3.6(a)

Proof. Note that $\hat{q}_1^{2,3}$ and $\hat{q}_1^{2,4}$ are given by solving an equation similar to (3.2). As the solution to the quadratic is defined completely by the covariance terms, all we need to prove is:

$$\frac{\Sigma'_{1,2}\Sigma'_{1,3}}{\Sigma'_{2,3}} = \frac{\Sigma'_{1,2}\Sigma'_{1,4}}{\Sigma'_{2,4}} \iff \frac{\Sigma'_{1,3}}{\Sigma'_{2,3}} = \frac{\Sigma'_{1,4}}{\Sigma'_{2,4}}.$$

By substituting the value of $\Sigma'_{i,j}$ from Equation B.6, we now need to prove that:

$$\frac{\Sigma_{1,3}}{\Sigma_{2,3}} = \frac{\Sigma_{1,4}}{\Sigma_{2,4}} \iff \frac{\rho_{1,3}}{\rho_{2,3}} = \frac{\rho_{1,4}}{\rho_{2,4}}.$$

Using the correlation decay property, we get that $\rho_{1,3} = \rho_{1,2}\rho_{2,3}$, $\rho_{1,4} = \rho_{1,2}\rho_{2,3}\rho_{3,4}$ and $\rho_{2,4} = \rho_{2,3}\rho_{3,4}$. Therefore LHS = RHS = $\rho_{1,2}$.

B.4.2 Proof of Lemma 3.3.6(b)

Proof. Using the same arguments as in the proof of Lemma 3.3.6(a), we can conclude that we need to prove:

$$\frac{\Sigma'_{1,3}\Sigma'_{2,4}}{\Sigma'_{2,1}\Sigma'_{3,4}} \neq 1, \frac{\Sigma'_{2,3}\Sigma'_{1,4}}{\Sigma'_{1,2}\Sigma'_{3,4}} \neq 1$$

Substituting $\Sigma'_{i,j}$ from Equation (B.6), we get:

$$\frac{\Sigma'_{1,3}\Sigma'_{2,4}}{\Sigma'_{2,1}\Sigma'_{3,4}} = \frac{\rho_{1,3}\rho_{2,4}}{\rho_{2,1}\rho_{3,4}}, \frac{\Sigma'_{2,3}\Sigma'_{1,4}}{\Sigma'_{1,2}\Sigma'_{3,4}} = \frac{\rho_{2,3}\rho_{1,4}}{\rho_{1,2}\rho_{3,4}}.$$

Using the correlation decay property, we get that:

$$\frac{\rho_{1,3}\rho_{2,4}}{\rho_{2,1}\rho_{3,4}} = \frac{\rho_{2,3}\rho_{1,4}}{\rho_{1,2}\rho_{3,4}} = \rho_{2,3}^2 \leq \rho_{max}^2 < 1 \tag{B.10}$$

B.4.3 Proof of Lemma 3.3.7

Proof. This is equivalent to proving that

$$\frac{\Sigma'_{1,3}}{\Sigma'_{2,3}} = \frac{\Sigma'_{1,4}}{\Sigma'_{2,4}}, \frac{\Sigma'_{1,2}}{\Sigma'_{2,4}} = \frac{\Sigma'_{1,3}}{\Sigma'_{3,4}}$$

which is again equivalent to:

$$\frac{\rho_{1,3}}{\rho_{2,3}} = \frac{\rho_{1,4}}{\rho_{2,4}}, \frac{\rho_{1,2}}{\rho_{2,4}} = \frac{\rho_{1,3}}{\rho_{3,4}}.$$

Using the correlation decay property it is easy to see that:

$$\frac{\rho_{1,3}}{\rho_{2,3}} = \frac{\rho_{1,4}}{\rho_{2,4}} = \rho_{1,2}, \frac{\rho_{1,2}}{\rho_{2,4}} = \frac{\rho_{1,3}}{\rho_{3,4}}.$$

B.4.4 Proof of Star/Non-star Condition for Generic Trees

We show how to utilize the result on a set of 4 nodes to classify any set of 4 nodes as star/non-star in a generic tree.

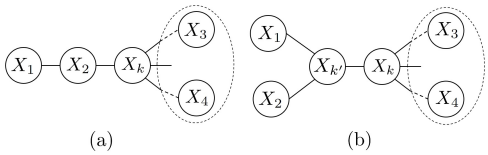


Figure B.2: Possible conditional independence relations for non-star shape if they don't form a chain

If any 4 nodes $\{X_1, X_2, X_3, X_4\}$ in a tree graphical model form a non-star shape such that (X_1, X_2) from a pair and are not arranged in a chain, there exist nodes X_k and $X_{k'}$ such that the conditional independence structure is given by either Figure B.2(a) or B.2(b). For the conditional independence in Figure B.2(a), we know that:

$$\begin{aligned}
&\hat{q}_4^{2,3} = \hat{q}_4^{k,2} \text{ By Lemma 3.3.7 on } \{X_2, X_3, X_4, X_k\}, \\
&\hat{q}_4^{1,3} = \hat{q}_4^{k,1} \text{ By Lemma 3.3.7 on } \{X_1, X_3, X_4, X_k\}, \\
&\hat{q}_4^{k,2} = \hat{q}_4^{k,1} \neq \hat{q}_4^{1,2} \text{ By Lemma 3.3.6(a) and Lemma 3.3.6(b)} \\
&\qquad \text{on } \{X_1, X_2, X_k, X_4\}.
\end{aligned} \tag{B.11}$$

This gives us $\hat{q}_4^{2,3} = \hat{q}_4^{1,3} \neq \hat{q}_4^{1,2}$. Similarly, we have $\hat{q}_3^{2,4} = \hat{q}_3^{1,4} \neq \hat{q}_3^{1,2}$.

We also know that:

$$\begin{aligned}
&\hat{q}_2^{1,3} = \hat{q}_2^{1,k} \neq \hat{q}_2^{k,3} \text{ By Lemma 3.3.6(a) and Lemma 3.3.6(b)} \\
&\quad \text{on } \{X_1, X_2, X_k, X_3\}, \\
&\hat{q}_2^{1,4} = \hat{q}_2^{1,k} \neq \hat{q}_2^{k,4} \text{ By Lemma 3.3.6(a) and Lemma 3.3.6(b)} \\
&\quad \text{on } \{X_1, X_2, X_k, X_4\}, \\
&\hat{q}_2^{k,3} = \hat{q}_2^{3,4} = \hat{q}_2^{k,4} \text{ By Lemma 3.3.7 on } \{X_2, X_3, X_4, X_k\}, \\
&\hat{q}_1^{2,3} = \hat{q}_1^{2,k} \neq \hat{q}_1^{k,3} \text{ By Lemma 3.3.6(a) and Lemma 3.3.6(b)} \\
&\quad \text{on } \{X_1, X_2, X_k, X_3\}, \\
&\hat{q}_1^{2,4} = \hat{q}_1^{2,k} \neq \hat{q}_1^{k,4} \text{ By Lemma 3.3.6(a) and Lemma 3.3.6(b)} \\
&\quad \text{on } \{X_1, X_2, X_k, X_4\}, \\
&\hat{q}_1^{k,3} = \hat{q}_1^{3,4} = \hat{q}_1^{k,4} \text{ By Lemma 3.3.7 on } \{X_1, X_3, X_4, X_k\}.
\end{aligned} \tag{B.12}$$

These equations imply $\hat{q}_2^{1,3} = \hat{q}_2^{1,4} \neq \hat{q}_2^{3,4}$ and $\hat{q}_1^{2,3} = \hat{q}_1^{2,4} \neq \hat{q}_1^{3,4}$. If the conditional independence is as shown in Figure B.2(b), we have:

$$\begin{aligned}
&\hat{q}_1^{2,3} = \hat{q}_1^{k',2} = \hat{q}_1^{2,4} = \hat{q}_1^{k',4} \text{By Lemma 3.3.7 on} \\
&\{X_1, X_2, X_3, X_{k'}\} \text{ and on } \{X_1, X_2, X_4, X_{k'}\}, \\
&\hat{q}_1^{k',4} \neq \hat{q}_1^{k,4} \text{By Lemma 3.3.6(b) on } \{X_1, X_k, X_{k'}, X_4\}, \\
&\hat{q}_1^{k,4} = \hat{q}_1^{3,4} \text{By Lemma 3.3.7 on } \{X_1, X_k, X_3, X_4\}.
\end{aligned} \tag{B.13}$$

These equations give us $\hat{q}_1^{2,3} = \hat{q}_1^{2,4} \neq \hat{q}_1^{3,4}$. Furthermore, by Equation (B.10), we have:

$$\begin{aligned}
&\frac{\rho'_{1,3}\rho'_{2,4}}{\rho'_{1,2}\rho'_{3,4}} \leq \rho_{max}^2 < 1 \\
&\frac{\rho'_{1,3}\rho'_{2,4}}{\rho'_{1,4}\rho'_{2,3}} = 1
\end{aligned} \tag{B.14}$$

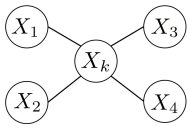


Figure B.3: Possible conditional independence relations for a star shape.

By symmetry, the remaining conditions in Equation (3.3) are also satisfied.

When the 4 nodes form a star structure in the tree, their conditional independence is given by either Figure 3.2 or there exists a node X_k such that the conditional independence is as shown in Figure B.3. Lemma 3.3.7 proves that Equation 3.3 is satisfied if the conditional independence is given by Figure 3.2. If the conditional independence is given by Figure B.3, we have:

$$\begin{aligned}
\hat{q}_1^{2,3} = \hat{q}_1^{2,k} = \hat{q}_1^{k,3} \text{ By Lemma 3.3.7 on } \{X_1, X_2, X_3, X_k\}, \\
\hat{q}_1^{4,3} = \hat{q}_1^{4,k} = \hat{q}_1^{k,3} \text{ By Lemma 3.3.7 on } \{X_1, X_3, X_4, X_k\}, \\
\hat{q}_1^{2,4} = \hat{q}_1^{2,k} = \hat{q}_1^{k,4} \text{ By Lemma 3.3.7 on } \{X_1, X_2, X_4, X_k\}.
\end{aligned} \tag{B.15}$$

This implies that $\hat{q}_1^{2,3} = \hat{q}_1^{4,3} = \hat{q}_1^{4,2}$. By symmetry, all the remaining conditions of Equation 3.3 are also satisfied.

This completes the proof that just by having access to the noisy probability distribution, it is possible to categorize any set of 4 nodes as a star/non-star shape.

B.5 Proof of Theorem 3.3.8

Given the noisy variance $\Sigma'_{i,j}$ and an estimate of the error probability vector $\hat{\mathbf{q}}$, we estimate the non-noisy covariance as:

$$\hat{\Sigma}_{i,j} = \frac{\Sigma'_{i,j}}{(1-2\hat{q}_i)(1-2\hat{q}_j)} = \frac{\Sigma_{i,j}(1-2q_i)(1-2q_j)}{(1-2\hat{q}_i)(1-2\hat{q}_j)} \forall i \neq j. \tag{B.16}$$

For the error probability vector $\hat{\mathbf{q}}$, using Equation (B.7) the non-noisy variance is estimated as:

$$\hat{\Sigma}_{i,i} = 1 - \frac{1 - \Sigma'_{i,i}}{(1 - 2\hat{q}_i)^2} \tag{B.17}$$

To check if any conditional independence relation $X_i \perp X_j | X_k$ is true, we need to verify if it satisfies the correlation decay equation $\hat{\Sigma}_{i,j}\hat{\Sigma}_{k,k} = \hat{\Sigma}_{i,k}\hat{\Sigma}_{k,j}$.

We first consider T' where only one leaf node exchanges position with its neighbor. Suppose in the original tree, node X_1 is a leaf node connected to node X_2.

Consider the error vector $\hat{\mathbf{q}}$:

$$\begin{aligned}
&\hat{q}_i = q_i \ \forall \ i \neq 1, 2, \\
&\hat{q}_1 = \frac{1}{2}\left(1 - (1 - 2q_1)\sqrt{\frac{\Sigma_{1,2}^2}{\Sigma_{2,2}} - \Sigma_{1,1} + 1}\right), \\
&\hat{q}_2 = 0.
\end{aligned} \tag{B.18}$$

To prove that this error vector results in T', we need to prove that any node $X_k \neq X_1, X_2$ which satisfies $X_1 \perp X_k | X_2$ in T^* must satisfy $X_2 \perp X_k | X_1$ in T'. We note that:

$$\begin{aligned}
&\hat{\Sigma}_{1,2} = \frac{\Sigma_{1,2}(1 - 2q_1)(1 - 2q_2)}{(1 - 2\hat{q}_1)}, \hat{\Sigma}_{1,k} = \frac{\Sigma_{1,k}(1 - 2q_1)}{(1 - 2\hat{q}_1)} \\
&\hat{\Sigma}_{2,k} = \Sigma_{2,k}(1 - 2q_2), \hat{\Sigma}_{1,1} = 1 - \frac{(1 - \Sigma_{1,1})(1 - 2q_1)^2}{(1 - 2\hat{q}_1)^2}
\end{aligned} \tag{B.19}$$

Using $\Sigma_{1,k}\Sigma_{2,2} = \Sigma_{1,2}\Sigma_{2,k}$, it is easy to check that $\hat{\Sigma}_{2,k}\hat{\Sigma}_{1,1} = \hat{\Sigma}_{1,k}\hat{\Sigma}_{1,2}$.

Furthermore, we need to prove that any pair of nodes $X_{k_1}, X_{k_2} \neq X_1, X_2$ such that $X_{k_1} \perp X_{k_2} | X_2$ in T^* satisfy $X_{k_1} \perp X_{k_2} | X_1$ in T'. Doing similar substitutions by replacing node 2 by node k_1 and node k by node k_2 gives us $\hat{\Sigma}_{1,1}\hat{\Sigma}_{k_1,k_2} = \hat{\Sigma}_{1,k_1}\hat{\Sigma}_{1,k_2}$ which proves that $X_{k_1} \perp X_{k_2} | X_1$.

The remaining conditional independences not involving X_1 and X_2 remain intact as the error probability for the remaining nodes is assigned to the original probability of error.

Now, consider a tree T' in which a set of leaf nodes $\mathcal{S}'$ exchange positions with their neighbors. For this case consider the error probability vector $\hat{\mathbf{q}}$ such that:

$$\begin{aligned}
\hat{q}_i =& \frac{1}{2}\left(1-(1-2q_i)\sqrt{\frac{\Sigma_{i,j}^2}{\Sigma_{j,j}}-\Sigma_{i,i}+1}\right), \\
& \forall\ i \in \mathcal{S}', j = Parent(i) \\
\hat{q}_j =& 0\ \forall\ i \in \mathcal{S}', j = Parent(i) \\
\hat{q}_k =& q_k, \text{ otherwise.}
\end{aligned}$$

This is obtained by performing the same procedure on each leaf node one by one.

Appendix C

Recoverability Landscape of Tree Structured Markov Random Fields under Symmetric Noise

C.1 Proof of Lemma 1

This proof relies on the classification of a set of 4 nodes as star/non-star. We use the information distance metric $d_{i,j}$ as defined in Equation (4.2) in order to achieve this

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 C.1. 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

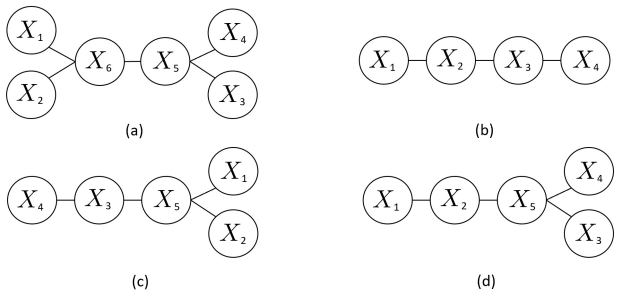


Figure C.1: Four possible configurations of (X_1, X_2, X_3, X_4) when they form a non-star such that (X_1, X_2) form a pair.

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:

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

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

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

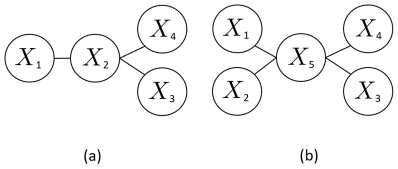


Figure C.2: Two possible configurations of (X_1, X_2, X_3, X_4) when they form a star.

Star condition: When the 4 nodes form a star, they can have either of the two configurations in Figure C.2. 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.

C.2 Obtaining Equation (4.6)

From Equation (4.3), 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 (4.4), we have $P_{2'} = (1 - q_2)P_2 + \frac{q_2}{k} I$. Substituting these in Equation (4.5), we get:

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

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

$$\begin{aligned} P_{2',3'}P^{-1}_{1',3'}P_{1',2'} =& \frac{E_2}{1-q_2}(1-q_2)\left(P_{2'} - \frac{q_2}{k} I\right)\frac{E_2}{1-q_2} \\ \left(\frac{E_2}{1-q_2}\right)^{-1} P_{2',3'}P^{-1}_{1',3'}P_{1',2'} \left(\frac{E_2}{1-q_2}\right)^{-1} =& (1-q_2)\left(P_{2'} - \frac{q_2}{k} I\right) \end{aligned} \tag{C.3}$$

Note that:

$$\begin{aligned} \frac{E_2}{1-q_2} &= I + \frac{q_2 O}{k(1-q_2)} \\ \left(\frac{E_2}{1-q_2}\right)^{-1} &= I - \frac{q_2 O}{k} \end{aligned} \tag{C.4}$$

Substituting this back in Equation (C.3)

$$\begin{aligned} &\left(I - \frac{q_2 O}{k}\right) P_{2',3'}P^{-1}_{1',3'}P_{1',2'} \left(I - \frac{q_2 O}{k}\right) = (1-q_2)\left(P_{2'} - \frac{q_2}{k} I\right) \\ &\frac{q_2^2}{k^2}(OP_{2',3'}P^{-1}_{1',3'}P_{1',2'}O - kI) - \frac{q_2}{k}(OP_{2',3'}P^{-1}_{1',3'}P_{1',2'} + P_{2',3'}P^{-1}_{1',3'}P_{1',2'}O - kP_{2'} - I) \\ &\qquad + P_{2',3'}P^{-1}_{1',3'}P_{1',2'} - P_{2'} = 0 \end{aligned} \tag{C.5}$$

To simplify this, we observe that:

$$\begin{aligned} OP_{2',3'} &= OP_{1',3'} = OP'_3 \\ P_{2',3'}O &= P_{2'}O \\ P_{1',2'}O &= P_{1',3'}O = OP'_1 \\ OP_{1',2'} &= OP_{2'} \end{aligned} \tag{C.6}$$

Substituting these back in Equation (C.5), we get:

$$\frac{q_2^2}{k^2}(O - kI) - \frac{q_2}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'}P^{-1}_{1',3'}P_{1',2'} - P_{2'} = 0 \tag{C.7}$$

C.3 Proof of Theorem 4.4.2

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 (4.7). 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. Now we see what happens when we check whether node X_2 is the middle node by solving Equation (C.1) 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:

$$P'_{2,3} = E_2P_{2,3}E_3, P'_{1,3} = E_1P_{1,3}E_3, P'_{1,2} = E_1P_{1,2}E_2.$$

Substituting these in Equations (C.1) and (4.6), we get:

$$\begin{aligned} E_2P_{2,1}P_1^{-1}P_{1,2}E_2 =& \tilde{E}_2^{1,3}\frac{1}{(1-\tilde{q}_2^{1,3})}\left(P_{2'} - \frac{\tilde{q}_2^{1,3}}{k}I\right)\tilde{E}_2^{1,3} \\ & \text{s.t.} 0 \leq \tilde{q}_2^{1,3} < 1, \end{aligned} \tag{C.8}$$

$$\begin{aligned} &\frac{(\tilde{q}_2^{1,3})^2}{k^2}(O - kI) - \frac{\tilde{q}_2^{1,3}}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) \\ &+ E_2P_{2,1}P_1^{-1}P_{1,2}E_2 - P_{2'} = 0 \text{ s.t. } 0 \leq \tilde{q}_2^{1,3} < 1. \end{aligned} \tag{C.9}$$

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 (C.9) 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:

$$\tilde{q}_1 = 0, \tilde{q}_2 = \tilde{q}_2^{1,3}, \tilde{q}_i = q_i \ \forall i \notin \{1,2\}. \tag{C.10}$$

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 (C.8) 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 (C.9) does not have a solution and constructing $\tilde{\mathbf{X}}$ when Equation (C.9) 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:

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

$$\tilde{q}_{p_i} = 0 \ \forall i \in \mathcal{L}_{\tilde{T}},$$

$$\tilde{q}_i = q_i \text{ otherwise,}$$

where X_{p_i} is the parent node of X_i. It is straightforward to see that the graphical model of $\tilde{\mathbf{X}}$ is $\tilde{T}$. □

C.4 Proof of Theorem 4.4.3

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

$$\left(\alpha_1 I + (1-\alpha_1)\frac{O}{k}\right)\left(\alpha_2 I + (1-\alpha_2)\frac{O}{k}\right) = \left(\alpha_1\alpha_2 I + (1-\alpha_1\alpha_2)\frac{O}{k}\right). \tag{C.11}$$

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:

$$P_{2',3'}P_{1',3'}^{-1}P_{1',2'} = E_2 P_{2,1}P_1^{-1}P_{1,2}E_2. \tag{C.12}$$

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 C.11, we get that:

$$P_{2',3'}P_{1',3'}^{-1}P_{1',2'} = E_2P_{2,1}P_1^{-1}P_{1,2}E_2 = \frac{1}{k}\left((1-q_2)^2\alpha_{1,2}^2 I + (1-(1-q_2)^2\alpha_{1,2}^2)\frac{O}{k}\right).$$

With these expressions, along with $P_{2'} = \frac{I}{k}$, we now look at the quadratic in Equation (4.7).

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

Thus, Equation 4.7 has a solution $x = 1-(1-q_2)\alpha_{1,2}$. □

C.5 Proof of Theorem 4.4.4

Using Equation (C.12), and recalling that $P_1 = P_{1'} = P_2 = P_{2'} = \frac{I}{k}$, we have that:

$$\frac{x^2}{k^2}(O-kI) - \frac{x}{k}(OP_{2'} + P_{2'}O - kP_{2'} - I) + P_{2',3'}P_{1,'3'}^{-1}P_{1',2'} - P_{2'} \tag{C.13}$$

$$=\frac{x^2}{k^2}(O-kI) - \frac{2x}{k^2}(O-kI) + E_2P_{2,1}P_1^{-1}P_{1,2}E_2 - \frac{I}{k} \tag{C.14}$$

$$=\left(\frac{x-1}{k}\right)^2(O-kI) - \frac{O}{k^2} + kE_2P_{2,1}P_{1,2}E_2. \tag{C.15}$$

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:

$$E_2P_{2,1}P_{1,2}E_2 = \left((1-q_2)I + q_2\frac{O}{k}\right)\left((\alpha_{a,b}-\delta_{a,b})I + (1-\alpha_{a,b})\frac{O}{k} + \Delta_{a,b}\right) \tag{C.16}$$

$$\left((\alpha_{a,b}-\delta_{a,b})I + (1-\alpha_{a,b})\frac{O}{k} + \Delta_{a,b}^T\right)\left((1-q_2)I + q_2\frac{O}{k}\right). \tag{C.17}$$

Now we have:

$$\begin{aligned}
E_2P_{2,1} =& \left((1-q_2)I + q_2\frac{O}{k}\right)\left((\alpha_{a,b}-\delta_{a,b})I + (1-\alpha_{a,b})\frac{O}{k} + \Delta_{a,b}\right) \\
=&(1-q_2)(\alpha_{a,b}-\delta_{a,b})I + (1-q_2)(1-\alpha_{a,b})\frac{O}{k} + (1-q_2)\Delta_{a,b} \\
&+ q_2(\alpha_{a,b}-\delta_{a,b})\frac{O}{k} + q_2(1-\alpha_{a,b})\frac{O}{k} + q_2\delta_{a,b}\frac{O}{k} \\
&= (1-q_2)(\alpha_{a,b}-\delta_{a,b})I + (1-(1-q_2)\alpha_{a,b})\frac{O}{k} + (1-q_2)\Delta_{a,b}
\end{aligned}$$

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:

$$E_2P_{2,1} = (\alpha'_{a,b}-\delta'_{a,b})I + (1-\alpha'_{a,b})\frac{O}{k} + \Delta'_{a,b}.$$

Noting that $P_{1,2}E_2 = (E_2P_{2,1})^T$, we get:

$$E_2P_{2,1}P_{1,2}E_2 = \frac{1}{k^2}\left(((\alpha'_{a,b}-\delta'_{a,b})^2 + (\delta'_{a,b})^2)I + \frac{O}{k}(1-(\alpha'_{a,b})^2) + (\alpha'_{a,b}-\delta'_{a,b})((\Delta'_{a,b})^T + \Delta'_{a,b})\right)$$

This gives us:

$$\begin{aligned}
Q^2(x) =& \| \left(\frac{x-1}{k}\right)^2 (O-kI) - \frac{O}{k^2} + kE_bP_{b,a}P_{a,b}E_b\|_F^2 \\
=& \| \left(\frac{x-1}{k}\right)^2 (O-kI) + ((\alpha'_{a,b}-\delta'_{a,b})^2 + \delta'^2_{a,b})\frac{I}{k} - \alpha'^2_{a,b}\frac{O}{k^2} + \frac{(\alpha'_{a,b}-\delta'_{a,b})}{k}(\Delta'^T_{a,b} + \Delta'_{a,b})\|_F^2
\end{aligned}$$

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:

$$\begin{aligned} Q^2(x) =& k\left(\frac{\gamma}{k^2} - \frac{\gamma}{k} - \frac{2e}{k}\right)^2 + 2k\left(\frac{\gamma}{k^2} + \frac{e}{k}\right)^2 + (k^2 - 3k)\frac{\gamma^2}{k^4} \\ =& \tfrac{1}{k^3}\left((k-1)\gamma + 2ke\right)^2 + \tfrac{2}{k^3}\left(\gamma + ke\right)^2 + \tfrac{k-3}{k^3}\gamma^2 \end{aligned}$$

$Q^2(x)$ 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 (4.7) 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.

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

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 (4.7) 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 C.2, a solution exists for Equation (4.7) if and only if it exists for:

$$P_{2',3'}P_{1',3'}^{-1}P_{1',2'} = \tilde{E}_2^{1,3}\frac{1}{(1-\tilde{q}_2^{1,3})}\left(P_{2'} - \frac{\tilde{q}_2^{1,3}}{k}I\right)\tilde{E}_2^{1,3} \text{ s.t. } 0 \leq \tilde{q}_2^{1,3} < 1. \tag{C.18}$$

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 (C.18) 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 $0 \leq \tilde{q}_2^{1,3} < 1$:

$$9P_{2,1}P_{1,2} = E_2^{-1}\tilde{E}_2^{1,3}\tilde{E}_2^{1,3}E_2^{-1}. \tag{C.19}$$

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 (C.11), 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 (C.11), we get:

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

We note that in Equation (C.19), 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 i^{th} column of M.

Since Equation (C.19) has a solution, we have the following properties of M:

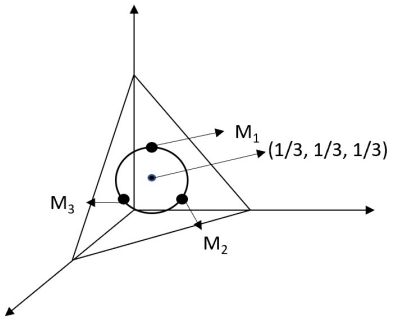


Figure C.3: Position of the three column vectors of matrix M for unidentifiability.

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

C.6 Proof of Lemma 4.4.5

We first analyze what happens to the solution of Equation (4.7) 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 4.4.5 as well as the algorithm design.

Lemma C.6.1. *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 (4.7) 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 (4.7). 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 (4.3) and substituting them in Equation (4.6) we get the following quadratic equation:

$$\frac{(\tilde{q}_2^{1,3})^2}{k^2} (O - kI) - \frac{\tilde{q}_2^{1,3}}{k} (OP_2' + P_2'O - kP_2' - I) + E_2 P_{2,4} P_4^{-1} P_{4,2} E_2 - P_2' = 0. \tag{C.20}$$

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

Next we go on to prove Lemma 4.4.5

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 (4.7) 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 (4.7) boils down to Equation (C.9) 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:

$$\frac{(\tilde{q}_2^{1,3})^2}{4} - \frac{(\tilde{q}_2^{1,3})}{2} + \frac{(P_{2,1})_{0,0}(P_{2,1})_{1,0}}{(P_{2,1})_{0,0} + (P_{2,1})_{1,0}} + \frac{(P_{2,1})_{0,1}(P_{2,1})_{1,1}}{(P_{2,1})_{0,1} + (P_{2,1})_{1,1}} = 0 \text{ s.t. } 0 \leq \tilde{q}_2^{1,3} < 1 \quad \text{(C.21)}$$

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 (C.21) 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 (4.7) has a solution when we estimate the probability of error of X_2 which enforces $X_4 \perp X_3|X_2$. Using Lemma C.6.1, we conclude that Equation (4.7) has a solution which enforces $X_1 \perp X_3|X_2$. $\square$

C.7 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 4.5.1), we can recover $\mathcal{T}^{sub}_{T^*}$. In the absence of the knowledge of t_0 , the algorithm returns one tree from $\mathcal{T}^{sub}_{T^*}$. 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 $\eta_{max} = (1-k)\log(1-q_{max}) - 0.5k\log(kp_{min})$. 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 $det(P_{i'}) < (1/k)^k$ and $det(P_i) \geq p^k_{min}$. This gives us:

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

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:

$$N(X_i) = sort(X_j : d_{i',j'} \leq t_{real},\ \text{key } = d_{i',j'}) \tag{C.22}$$

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

C.7.1 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.

C.7.1.1 QuadraticError

In this subroutine, we test if Equation (4.7) has a solution. Note that the quadratic in Equation (4.7) with matrix coefficients is equivalent to having k^2 quadratic equations. Equation (4.7) 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 3 Find the Error of the quadratic in Equation (4.7)

Input - Pairwise noisy distributions, a set of 3 nodes, test center node among the three nodes.
Output - Error of the quadratic in Equation (4.7).

procedure QUADRATICERROR($P_{i',j'}, NodeTriplet, TestCenter$)
  $A, B, C \leftarrow$ Matrix Quadratic Coefficients from Equation (4.7) for given $NodeTriplet, TestCenter$.
  $MeanRoot \leftarrow 0$
  for i_1 in $1 \dots k$ **do**
    for i_2 in $1 \dots k$ **do**
      $MeanRoot \leftarrow MeanRoot + \frac{root(A[i_1,i_2]x^2+B[i_1,i_2]x+C[i_1,i_2])}{k^2}$
    end for
  end forreturn $\|A(MeanRoot)^2 + B(MeanRoot) + C\|_F$
end procedure

C.7.1.2 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 4 Recover Center Node in the Unidentifiable setting

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

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

C.7.1.3 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.

Further robustifying FindCenter: At any point in the algorithm, suppose in the previous iterations we have recovered the edges $\{z, z_1\}, \{z, z_2\}, \ldots \{z, z_j\}$, then all the star/non-star tests involving (l, r, z, z_i) $\forall i \in \{1, 2, \ldots j\}$ are have the same star/non-star characterization and if they are non-star then z_i pairs with z in all the tests. We have the same

phenomena for the already recovered edges of l and r. Thus, when executing the algorithm with finite samples, we can robustify the FindCenter subroutine by considering all the nodes whose edge with node z has been recovered and assign them the same star/non-star classification as the majority. We do the same for nodes l and r also.

Algorithm 5 Find a leaf parent pair.

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

procedure GetLeafParent($P_{i',j'}$, *ActiveSet*, *Edges*, *Parents*)
  if $|ActiveSet \cap Parents| > 0$ **then**
    $r \leftarrow ActiveSet \cap Parents[0]$
  else
    $r \leftarrow ActiveSet[0]$
  end if
  $l \leftarrow N(r)[0] \cap ActiveSet$
  $i \leftarrow 1, visited \leftarrow \{l, r\}$
  while $i < len(N(r))$ **do**
    $z \leftarrow N(r)[i]$
    if $z \in visited$ or $z \notin ActiveSet$ **then**
      $i \leftarrow i + 1$
      continue
    end if
    $visited \leftarrow visited \cup z$
    $C \leftarrow$FindCenter$(P_{i',j'}, (l, r, z))$
    if $|C| == 1$ **then**
      $l\_r\_order = True$
    end if
    if $C == z$ **then**
      $l \leftarrow z$
    else if $C == r$ **then**
      $l \leftarrow r, r \leftarrow z, i \leftarrow 0$
    else if $|C| > 1$ **then**
      if $l\_r\_order == True$ and $r, l \in C$ **then**
        break
      end if
      $r, l \leftarrow$LeafClusterResolution$(C, Parents, ActiveSet)$
      break
    end if
  end whilereturn r, l
end procedure

C.7.1.4 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 (4.7), 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 (4.7), 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 6 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 ∩ Parents| > 0 then
3:         l ← C ∩ Parents[0] return C \ {l}[0], l
4:     end if
5:     MinError ← ∞
6:     for (X_{i_1}, X_{i_2}) ∈ C do
7:         for X_{i_3} ∈ N(X_{i_1}) ∩ N(X_{i_2}) do
8:             if X_{i_3} ∈ 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}} ≤ d_max +
   2η_max then
9:                 CandidateParent ← (X_{i_1}, X_{i_2}, X_{i_3})
10:            else CandidateParent ← (X_{i_1}, X_{i_2})
11:            end if
12:            for X_i ∈ CandidateParent do
13:                err ← QuadraticError((X_{i_1}, X_{i_2}, X_{i_3}), X_i)
14:                if err < MinError then
15:                    MinError ← err, l ← X_i
16:                end if
17:            end for
18:        end for
19:    end for
20:    r ← C \ {l}[0]
21: return r, l
22: end procedure
```

C.7.1.5 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 $\mathcal{O}(n)$)

C.7.1.6 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.

C.7.1.7 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.

C.7.2 Proof of correctness

C.7.2.1 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 4.4.1. 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,

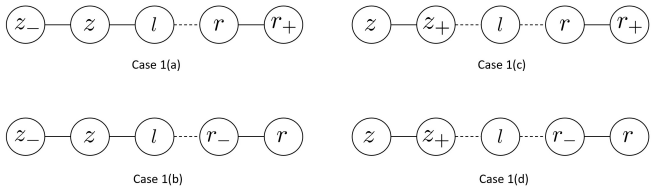


Figure C.4: All the possible when node z lies to the left of node l

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 C.4.

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.)

$$\begin{aligned} d_{r',z'_-} =& d_{r,r'} + d_{r,z} + d_{z,z'_-} \\ \leq& \eta_{max} + (d_{max} + \eta_{max}) + (d_{max} + \eta_{max}) \\ =& 2d_{max} + 3\eta_{max} \\ d_{l',z'_-} =& d_{l,l'} + d_{l,z} + d_{z,z'_-} \end{aligned}$$

$$
\begin{aligned}
&\leq \eta_{max} + (d_{max} + \eta_{max}) + (d_{max} + \eta_{max}) \\
&= 2d_{max} + 3\eta_{max} \\
d_{z',r'_+} &= d_{z',z} + d_{z,r} + d_{r,r'_+} \\
&\leq 2d_{max} + 3\eta_{max} \\
d_{l',r'_+} &= d_{l,l'} + d_{l,r} + d_{r,r'_+} \\
&\leq 2d_{max} + 3\eta_{max}
\end{aligned}
$$

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:

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

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.

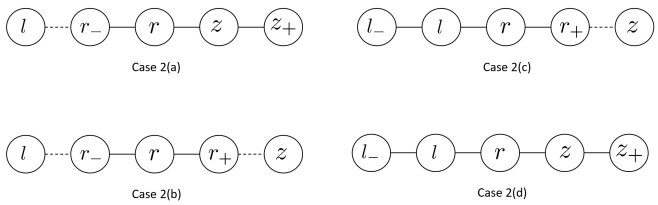


Figure C.5: All the possible when node z lies to the right of node r

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 C.5.

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).

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

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, $d_{z^{t+1},r^{t+1}} \leq d_{z'_+,z} \leq d_{max} + \eta_{max}$.

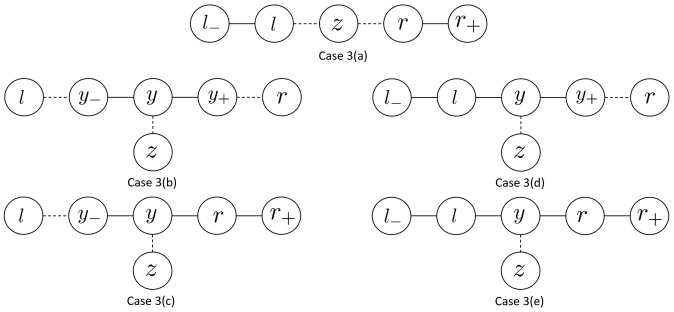


Figure C.6: All the possible when node z does not lie to the left of l or right of r

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 (4.7) 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 (4.7). 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 (4.7). For an unidentifiable leaf node, both the leaf and parent have a solution to Equation (4.7) 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.

C.7.3 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 $\kappa_{\hat{i'},j'}$

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 $t_{emp} = 0.5(4d_{max} + 3\eta_{max})$.

C.8 Sample Complexity Upper Bound

Let us define 2 events:

$$\mathcal{B}_1 = \{(E_{a'})_{i,i} < 0.1p_{min}, \forall a, i\}, \mathcal{B}_2 = \{\|E_{a',b'}\| < \epsilon \forall a, b\}$$

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 $\mathcal{B}_1$, $det(\hat{P}_{a'}), det(\hat{P}_{b'}) > (0.9p_{min})^k$, therefore we have:

$$|det(\hat{P}_{a',b'})| \geq 0.5\exp(-4d_{\max})(1-q_{max})^{3(k-1)}(kp_{min})^{1.5k}(0.9p_{min})^k$$

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

Lemma C.8.1. *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:

$$\begin{aligned}
\min \quad & |\lambda_k| && \text{(C.23)}\\
\text{s.t.} \quad & \sum_{i=1}^{k} |\lambda_i| \leq 1 && \text{(C.24)}\\
& \prod_{i=1}^{k} |\lambda_i| \geq c, && \text{(C.25)}\\
& |\lambda_1| \geq |\lambda_2| \ldots |\lambda_k|, && \text{(C.26)}
\end{aligned}$$

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 $|\tilde{\lambda_k}| = |\lambda_i^*| - \epsilon'$ 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 C.23 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 C.23 has the same solution as the following optimization problem:

$$\begin{aligned} \min \quad & |\lambda_k| \\ \text{s.t.} \quad & 0 < |\lambda_k| \leq \frac{1}{k} \end{aligned}$$

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

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 $c(k-1)^{k-1} \leq |\lambda_k^*| \leq ck^{k-1}$ □

Using Lemma C.8.1, 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:

$$\begin{aligned} \|\hat{P}_{a',b'}^{-1}\| \leq & \frac{1}{0.5 \exp(-4d_{\max})(1-q_{max})^{3(k-1)}(kp_{min})^{1.5k}(0.9p_{min})^k(k-1)^{k-1}} \\ \leq & \frac{8(k-1)}{\exp(-4d_{\max})(1-q_{max})^{3(k-1)}(kp_{min})^{2.5k}(0.9)^k} \triangleq \frac{1}{z_1} \end{aligned} \tag{C.27}$$

where the second inequality uses the fact that $(\frac{k-1}{k})^k \geq 1/4$

C.8.1 Sample Complexity for Existence of a solution to Equation 4.7

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.

$$\begin{aligned}
P_{b',c'}P_{a',c'}^{-1}P_{a',b'} &= (\hat{P}_{b',c'} + E_{b',c'})(\hat{P}_{a',c'} + E_{a',c'})^{-1}(\hat{P}_{a',b'} + E_{a',b'}) \\
&= (\hat{P}_{b',c'} + E_{b',c'}) \left(\hat{P}_{a',c'}^{-1} + \sum_{m=1}^{\infty} (-\hat{P}_{a',c'}^{-1}E_{a',c'})^m \hat{P}_{a',c'}^{-1} \right) (\hat{P}_{a',b'} + E_{a',b'}) \\
&= \hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{ab} + E_{b',c'}\hat{P}_{a',c'}^{-1}\hat{P}_{a',b'} + \hat{P}_{b',c'}\hat{P}_{a',c'}^{-1}E_{a',b'} + \hat{P}_{b',c'}\tilde{E}_{ac}\hat{P}_{a',b'} \\
&+ E_{b',c'}\hat{P}_{a',c'}^{-1}E_{a',b'} + E_{b',c'}\tilde{E}_{ac}\hat{P}_{a',b'} + \hat{P}_{b',c'}\tilde{E}_{ac}E_{a',b'} + E_{b',c'}\tilde{E}_{ac}E_{a',b'},
\end{aligned}$$

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 (C.27), $\|\hat{P}_{a',c'}^{-1}\|_2 \leq z_1^{-1}$. 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:

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

$$\leq Q(x) + (3x+1)\|E_{b'}\|_F + \|\hat{P}_{b',c'}\hat{P}^{-1}_{a',c'}\hat{P}_{a',b'} - P_{b',c'}P^{-1}_{a',c'}P_{ab}\|_F$$

$$\implies |\hat{Q}(x) - Q(x)| \leq 4\sqrt{k}\epsilon + 3\sqrt{k}\epsilon z_1^{-1} + 8\sqrt{k}\epsilon z_1^{-2} \leq 15\sqrt{k}\epsilon z_1^{-2}$$

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

$$\epsilon < \frac{t_0 z_1^2}{30\sqrt{k}} \tag{C.28}$$

C.8.2 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.

$$\frac{|det(P_{1,3}P_{2,4})|}{|det(P_{1,4}P_{2,3})|} = \frac{|det((\hat{P}_{1,3} + E_{1,3})(\hat{P}_{2,4} + E_{2,4}))|}{|det((\hat{P}_{1,4} + E_{1,4})(\hat{P}_{2,3} + E_{2,3}))|} \tag{C.29}$$

Using the analysis from [73], 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 [3] 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:

$$\epsilon < \frac{z_1(1-\alpha)}{20k}. \tag{C.30}$$

From Equations (C.28), (C.30) we choose ϵ as follows:

$$\epsilon < \min\left\{\frac{z_1(1-\alpha)}{20k}, \frac{t_0 z_1^2}{30\sqrt{k}}\right\}. \tag{C.31}$$

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:

$$P((E_{a'})_{i,i}) > 0.1p_{min}) \leq \exp(-2N(0.1p_{min})^2).$$

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

$$P(\bar{\mathcal{B}}_1) \leq kn \exp(-2N(0.1p_{min})^2).$$

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

$$N \geq \frac{50}{p_{min}^2} \log\left(\frac{2nk}{\delta}\right). \tag{C.32}$$

Next, we upper bound the probability $P(\bar{\mathcal{B}}_2)$.

The matrix Bernstein's inequality ([74]) states that for independent random matrices $S_1 \dots S_N$ 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 $v(Z) = \max\{\|\sum_{i=1}^{N} \mathbb{E}\left[S_i S_i^T\right]\|\}$. In order to apply this in our setting, define $S_i = \mathbb{1}^i_{a',b'} - P_{a',b'}$ where $\mathbb{1}^i_{a',b'}$ 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 $v(Z)$.

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

$$= \mathbb{E}\big[(\not\perp^i_{a',b'})(\not\perp^i_{a',b'})^T\big] - \mathbb{E}\big[P_{a',b'}P^T_{a',b'}\big]$$

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

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

$$N \geq \frac{2(2+\epsilon/3)}{\epsilon^2} \log\left(\frac{2nk(n-1)}{\delta}\right) \tag{C.33}$$

From Equations (C.32) and (C.33), the algorithm outputs the correct tree if:

$$N \geq \max\left\{\frac{50}{p_{min}^2} \log\left(\frac{2nk}{\delta}\right), \frac{2(2+\epsilon/3)}{\epsilon^2} \log\left(\frac{2nk(n-1)}{\delta}\right)\right\} \tag{C.34}$$

From the value of ϵ as defines in Equation (C.31), we can see that the sample complexity is dominated by the second term. Substituting the value of ϵ from Equation (C.31), we get that the sample complexity is of the following order:

$$N = \mathcal{O}\Bigg(\max\Bigg\{\frac{k^2 \exp(8d_{\max})}{(1-q_{max})^{6(k-1)}(0.9p_{min}^{2.5})^{2k}(1-\exp(-2d_{min}))^2(k-1)^{2(k-1)}}, \\ \frac{k \exp(16d_{\max})}{t_0^2(1-q_{max})^{12(k-1)}(0.9p_{min}^{2.5})^{4k}(k-1)^{4(k-1)}}\Bigg\} \log\left(\tfrac{2nk(n-1)}{\delta}\right)\Bigg)$$

C.9 Sample Complexity Lower Bound

C.9.1 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 C.9.1 (Fano's Inequality, Lemma 6.2 in Bresler et al.[8])**.** *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*

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

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 $(1 - \delta + \frac{1}{\log(M)})$.

Symmetric Graphical Models: For symmetric graphical models [16], the marginals of all the random variables are uniform on the support and the conditional distribution for two

random variables X_i, X_j such that $(X_i, X_j) \in \mathcal{E}$ is given by:

$$P_{i|j} = \alpha_{i,j} I + (1 - \alpha_{i,j}) \frac{O}{k},$$

where O is the $k \times k$ matrix of all $1's$, k is the support size, and $0 < \alpha_{i,j} < 1$. This characterization has the following property:

Lemma C.9.2. *Consider any 2 nodes X_{i_1}, X_{i_t} in a symmetric graphical model such that the path between X_{i_1} and X_{i_t} is $X_{i_1} - X_{i_2} - \cdots - X_{i_{t-1}} - X_{i_t}$. Then, the conditional PMF matrix of X_{i_1} conditioned on X_{i_t} is given as follows:*

$$P_{i_1|i_t} = \alpha_{i_1,i_t} I + (1 - \alpha_{i_1,i_t}) \frac{O}{k} = \prod_{p=1}^{t-1} \alpha_{i_p,i_{p+1}} I + \left(1 - \prod_{p=1}^{t-1} \alpha_{i_p,i_{p+1}}\right) \frac{O}{k},$$

that is, $\alpha_{i_1,i_t} = \prod_{p=1}^{t-1} \alpha_{i_p,i_{p+1}}$

We remark that when considering noisy random variables we have that:

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

For each node X_i, we define $\alpha_{i',i} = 1 - q_i$. Therefore, we get:

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

such that $\alpha_{i',i} > 0$ (as $q_i \leq q_{\max} < 1$).

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 $v'(i) = v((i+1) \mathrm{mod} k)$ for all $1 \leq i \leq k$. Then we have $v'' = \mathcal{R}^j(v)$ as $v''(i) = v((i+j) \mathrm{mod} k)$ 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 C.9.3. *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. □

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:

$$\begin{aligned} P_{b',c'}P_{a',c'}^{-1}P_{a',b'} &= E_bP_{b,c}E_c(E_aP_{a,c}E_c)^{-1}E_aP_{a,b}E_b \\ &= E_bP_{b,a}P_a^{-1}P_{a,c}E_cE_c^{-1}P_{a,c}^{-1}E_a^{-1}E_aP_{a,b}E_b \\ &= E_bP_{b,a}P_a^{-1}P_{a,b}E_b \end{aligned}$$

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

$$Q(x) = \|\frac{x^2}{k^2}(O - kI) - \frac{x}{k}(OP_b' + P_b'O - kP_b' - I) + P_{b',c'}P_{a',c'}^{-1}P_{a',b'} - P_b'\|_F, \tag{C.35}$$

$$= \|\left(\frac{x^2 - 2x + 1}{k^2}\right)O - \left(\frac{x^2 - 2x + 1}{k}\right)I - \frac{O}{k^2} + kE_bP_{b,a}P_{a,b}E_b\|_F, \tag{C.36}$$

$$= \|\left(\frac{x - 1}{k}\right)^2(O - kI) - \frac{O}{k^2} + kE_bP_{b,a}P_{a,b}E_b\|_F \tag{C.37}$$

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 (C.35) along with $E_b = I$, we get:

$$\begin{aligned} Q^2(x) =& \| \left(\frac{x-1}{k}\right)^2 (O - kI) - \frac{O}{k^2} + kE_bP_{b,a}P_{a,b}E_b\|_F^2 \\ =& \| \left(\frac{x-1}{k}\right)^2 (O - kI) + ((\alpha-\delta)^2+\delta^2)\frac{I}{k} - \alpha^2\frac{O}{k^2} + \frac{(\alpha-\delta)}{k}(\Delta^T+\Delta)\|_F^2 \end{aligned}$$

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:

$$\begin{aligned} Q^2(x) =& k\left(\frac{\gamma}{k^2} - \frac{\gamma}{k} - \frac{2e}{k}\right)^2 + 2k\left(\frac{\gamma}{k^2} + \frac{e}{k}\right)^2 + (k^2 - 3k)\frac{\gamma^2}{k^4} \\ =& \tfrac{1}{k^3}\left((k-1)\gamma + 2ke\right)^2 + \tfrac{2}{k^3}\left(\gamma + ke\right)^2 + \tfrac{k-3}{k^3}\gamma^2 \end{aligned}$$

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

$$Q^2(x) \geq \frac{2(k-3)\delta^2(\alpha-\delta)^2k^2}{k-1}. \tag{C.38}$$

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

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

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 $k-1$,

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

$$= \begin{cases} 1 = (1-\theta'-(k-1)\theta) + (\theta'-\theta) + k\theta,\ j = 0 \\ (1-\theta'-(k-1)\theta) + (\theta'-\theta)\omega^j,\ j \neq 0 \end{cases}$$

Therefore, we have following the derivations in [29]

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

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.

C.9.2 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 the symmetric class of tree graphical models.

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 [73], but uses discrete symmetric distribution instead of using Ising models.

The family of distributions is given as $(P^{(i)} : i = 0, 1, \ldots, t^2 - 1)$. 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. This is shown in Figure C.7. The edges have two different type of conditional as described below.

$$P^{(0)}_{j'|(2t+1)'} = \alpha_{min}(1 - q_{max})I + (1 - \alpha_{min}(1 - q_{max}))\frac{O}{k}, \forall j \in \{1, 2 \ldots t\},$$
$$P^{(0)}_{j'|(2t+1)'} = \alpha_{max}I + (1 - \alpha_{max})\frac{O}{k}, \forall j \in \{t + 1, t + 2 \ldots 2t\}.$$

For any $i = 1, \ldots, t^2-1$, 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 $i_b = i-\lfloor\frac{i-1}{t}\rfloor t$. As noted in [73], the pair (i_a, i_b) is unique for every $i = 1, \ldots, t^2-1$. We use another discrete symmetric distribution for all these edges: (i_a, i_b) for any $i = 1, \ldots, t^2 - 1$. Specifically, the conditional pmf of the different edges of the i-th graphical model is given below.

$$P^{(i)}_{j'|(2t+1)'} = \alpha_{min}(1 - q_{max})I + (1 - \alpha_{min}(1 - q_{max}))\frac{O}{k} \forall i \in \{1, 2, 3, \ldots t\} \setminus \{i_a\},$$
$$P^{(i)}_{j'|(2t+1)'} = \alpha_{max}I + (1 - \alpha_{max})\frac{O}{k} \forall i \in t + 1, t + 2 \ldots 2t,$$
$$P^{(i)}_{i'_a|i'_b} = \alpha_{min}(1 - q_{max})I + (1 - \alpha_{min}(1 - q_{max}))\frac{O}{k}.$$

We finally note that all the graphs $P^{(i)}$ for $i \in \{0, 1, \ldots, t^2 - 1\}$ have a different equivalence class. In particular, we see that $P^{(0)}$ admits all possible permutation of star

nodes (with node i being the root, and remaining $2t$ nodes being the leaf nodes, for all $i \in \{1, \ldots, 2t+1\}$). For $P^{(i)}$, the equivalence structure is given by two leaf clusters connected by a single edge. The nodes $\{1, \ldots, 2t+1\} \setminus \{i_a, i_b\}$ forms one leaf cluster, while $\{i_a, i_b\}$ forms the other leaf cluster. As (i_a, i_b) is unique for all $i \in \{1, \ldots, t^2 - 1\}$, all the t^2 graphs under consideration have different equivalence classes (see, Figure C.7). Also, all the leaf nodes are indistinguishable from it's parents in each of these graphs.

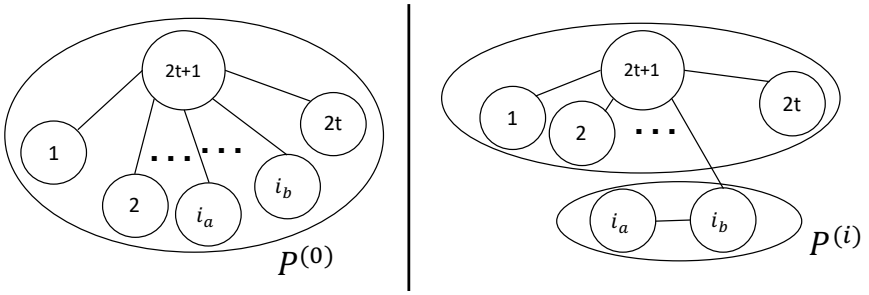


Figure C.7: The family of distributions used for providing lower bound for completely unidentifiable case. The graphical model corresponding to $P^{(0)}$ a single recoverable leaf cluster. The graphical model corresponding to $P^{(i)}$, for each $i = 1, \ldots, t^2 - 1$, has nodes $\{i_a, i_b\}$ as one recoverable leaf cluster, and the remaining nodes as another recoverable leaf cluster.

Symmetrized KL-divergence: For the symmetrized KL-divergence $J(P^{(0)}, P^{(1)})$ computation we focus our attention on $i = 1$, in which case $i_a = 1$ and $i_b = (t+1)$. The computation remains identical for other $i \geq 2$ due to symmetry.

For this purpose, we need to compute $\mathbb{E}_{\mathbf{X} \sim P^{(0)}} \log \left(\frac{P^{(0)}(\mathbf{X})}{P^{(1)}(\mathbf{X})} \right)$ and $\mathbb{E}_{\mathbf{X} \sim P^{(1)}} \log \left(\frac{P^{(1)}(\mathbf{X})}{P^{(0)}(\mathbf{X})} \right)$. Let us look at $\left(\frac{P^{(0)}(\mathbf{X})}{P^{(1)}(\mathbf{X})} \right)$. Recall that the nodes $t+1, t+2 \ldots 2t+1$ have 0 noise. We first see that the expression for $P^{(0)}(\mathbf{X})$ can be decomposed as follows due to the discrete symmetric

conditional PMF and the graph structure:

$$P^{(0)}(\mathbf{X}) = P^{(0)}(X'_{2t+1}) \prod_{i=1}^{2t} P^{(0)}(X'_i | X'_{2t+1})$$

Similarly, the decomposition for $P^{(1)}(\mathbf{X})$ is:

$$P^{(1)}(\mathbf{X}) = P^{(1)}(X'_{2t+1}) P^{(1)}(X'_1 | X'_{t+1}) \prod_{i=2}^{2t} P^{(0)}(X'_i | X'_{2t+1})$$

Furthermore, due to the property of discrete symmetric model we have $P^{(0)}(X'_{2t+1}) = P^{(1)}(X'_{2t+1}) = 1/k$. This gives us:

$$\frac{P^{(0)}(\mathbf{X})}{P^{(1)}(\mathbf{X})} = \frac{P^{(0)}(X'_1 | X'_{2t+1})}{P^{(1)}(X'_1 | X'_{t+1})}.$$

Therefore,

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

We find the symmetrized KL divergence between $P^{(0)}$ and $P^{(1)}$. We primarily need the following four conditional PMF matrices for the calculation of the symmetrized KL divergence:

$$P^{(0)}_{1'|(2t+1)'} = (1 - q_{\max})\alpha_{min} I + (1 - (1 - q_{\max})\alpha_{min}) \frac{O}{k} \tag{C.39}$$

$$P^{(0)}_{1'|(t+1)'} = (1 - q_{\max})\alpha_{min}\alpha_{max} I + (1 - (1 - q_{\max})\alpha_{min}\alpha_{max}) \frac{O}{k} \tag{C.40}$$

$$P^{(1)}_{1'|(2t+1)'} = (1 - q_{\max})\alpha_{min}\alpha_{max} I + (1 - (1 - q_{\max})\alpha_{min}\alpha_{max}) \frac{O}{k} \tag{C.41}$$

$$P^{(1)}_{1'|(t+1)'} = (1 - q_{\max})\alpha_{min} I + (1 - (1 - q_{\max})\alpha_{min}) \frac{O}{k} \tag{C.42}$$

For notational simplicity let us use $\alpha_n = (1 - q_{\max})$.

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

$$
\begin{aligned}
&= \mathbb{E}_{X'_1, X'_{2t+1} \sim P^{(0)}} \log(P^{(0)}(X'_1|X'_{2t+1}) \\
&= \sum_{(X'_1, X'_{2t+1}) \in \mathcal{S}^2} P^{(0)}(X'_1, X'_{2t+1}) \log(P^{(0)}(X'_1|X'_{2t+1}) \\
&= \frac{1}{k} \sum_{(X'_1, X'_{2t+1}) \in \mathcal{S}^2} P^{(0)}(X'_1|X'_{2t+1}) \log(P^{(0)}(X'_1|X'_{2t+1}) \\
&= \frac{1}{k} \left(\sum_{(X'_1 = X'_{2t+1})} P^{(0)}(X'_1|X'_{2t+1}) \log(P^{(0)}(X'_1|X'_{2t+1}) + \sum_{(X'_1 \neq X'_{2t+1})} P^{(0)}(X'_1|X'_{2t+1}) \log(P^{(0)}(X'_1|X'_{2t+1}) \right) \\
&= \frac{1}{k} \left(k \left(\alpha_{min}\alpha_n + \frac{1 - \alpha_{min}\alpha_n}{k} \right) \log \left(\alpha_{min}\alpha_n + \frac{1 - \alpha_{min}\alpha_n}{k} \right) \right) \\
&+ \frac{1}{k} \left((k^2 - k) \left(\frac{1 - \alpha_{min}\alpha_n}{k} \right) \log \left(\frac{1 - \alpha_{min}\alpha_n}{k} \right) \right)
\end{aligned}
$$

For the second term we have similarly,

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{X} \sim P^{(0)}} \log(P^{(1)}(X'_1|X'_{t+1}) \\
&= \frac{1}{k} \left(\sum_{(X'_1 = X'_{t+1})} P^{(0)}(X'_1|X'_{t+1}) \log(P^{(1)}(X'_1|X'_{t+1}) + \sum_{(X'_1 \neq X'_{t+1})} P^{(0)}(X'_1|X'_{t+1}) \log(P^{(1)}(X'_1|X'_{t+1}) \right) \\
&= \frac{1}{k} \left(k \left(\alpha_{min}\alpha_{max}\alpha_n + \frac{1 - \alpha_{min}\alpha_{max}\alpha_n}{k} \right) \log \left(\alpha_{min}\alpha_n + \frac{1 - \alpha_{min}\alpha_n}{k} \right) \right) \\
&+ \frac{1}{k} \left((k^2 - k) \left(\frac{1 - \alpha_{max}\alpha_{min}\alpha_n}{k} \right) \log \left(\frac{1 - \alpha_{min}\alpha_n}{k} \right) \right)
\end{aligned}
$$

Recall the p.m.f. for a tree structured graphical model with vertex set V and edge set E, and alphabet $\mathcal{X} = [K]^{|V|}$, is

$$
P(\mathbf{X}) = \prod_{i \in V} P_i(X_i) \prod_{(i,j) \in E} \frac{P_{i,j}(X_i, X_j)}{P_i(X_i), P_j(X_j)},
$$

In the symmetric setting, we get that:

$$
P(\mathbf{X}) = \tfrac{1}{k} \prod_{(i,j) \in E} P_{i,j}(X_i|X_j).
$$

Computing the symmetrized KL divergence involves calculating the following 4 terms which can be done using Equation (C.39):

$$\begin{aligned}
\mathbb{E}_{P^{(0)}}\log(P^{(0)}(X'_{2t+1}|X'_1)) &= \left(\alpha_{min}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_n)}{k}\right)\log\left(\alpha_{min}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_n)}{k}\right) \\
&\quad + \left(\tfrac{(k-1)}{k}(1-\alpha_{min}\alpha_n)\log\left(\tfrac{1-\alpha_{min}\alpha_n}{k}\right)\right) \\
\mathbb{E}_{P^{(0)}}\log(P^{(1)}(X'_{t+1}|X'_1)) &= \left(\alpha_{min}\alpha_{max}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_{max}\alpha_n)}{k}\right)\log\left(\alpha_{min}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_n)}{k}\right) \\
&\quad + \left(\tfrac{(k-1)}{k}(1-\alpha_{min}\alpha_{max}\alpha_n)\log\left(\tfrac{1-\alpha_{min}\alpha_n}{k}\right)\right) \\
\mathbb{E}_{P^{(1)}}\log(P^{(1)}(X'_{t+1}|X'_1)) &= \left(\alpha_{min}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_n)}{k}\right)\log\left(\alpha_{min}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_n)}{k}\right) \\
&\quad + \left(\tfrac{(k-1)}{k}(1-\alpha_{min}\alpha_n)\log\left(\tfrac{1-\alpha_{min}\alpha_n}{k}\right)\right) \\
\mathbb{E}_{P^{(1)}}\log(P^{(0)}(X'_{2t+1}|X'_1)) &= \left(\alpha_{min}\alpha_{max}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_{max}\alpha_n)}{k}\right)\log\left(\alpha_{min}\alpha_n + \tfrac{(1-\alpha_{min}\alpha_n)}{k}\right) \\
&\quad + \left(\tfrac{(k-1)}{k}(1-\alpha_{min}\alpha_{max}\alpha_n)\log\left(\tfrac{1-\alpha_{min}\alpha_n}{k}\right)\right)
\end{aligned}$$

This gives us:

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

Substituting these quantities from above and simplifying, we get:

$$\begin{aligned}
J(P^{(0)}, P^{(1)}) &= 2\alpha_{min}\alpha_n(1-\alpha_{max})\left(\tfrac{k-1}{k}\right)\log\left(1+\tfrac{k\alpha_{min}\alpha_n}{1-\alpha_{min}\alpha_n}\right) \\
&= 2\exp(-\tfrac{d_{\max}}{k-1})(1-q_{\max})(1-\exp(-\tfrac{d_{\min}}{k-1}))\left(\tfrac{k-1}{k}\right)\log\left(1+\tfrac{k\exp(-\frac{d_{\max}}{k-1})(1-q_{\max})}{1-\exp(-\frac{d_{\max}}{k-1})(1-q_{\max})}\right) \\
&\leq 2(k-1)\exp(-\tfrac{2d_{\max}}{k-1})(1-q_{\max})^2(1-\exp(-\tfrac{d_{\min}}{k-1}))
\end{aligned}$$

We have the maximum distance between two nodes given as $d_{\max} = -(k-1)\log(\alpha_{min})$ and $d_{\min} = -(k-1)\log(\alpha_{max})$. The noise is related as $\alpha_n = (1-q_{max})$. Substituting, these terms above provides us the second equality. Using $\log(1+x) \leq x$ gives the final inequality.

Lower Bound Proof - Part I: We are now in a position to prove the first part of Theorem 4.6.2.

By the application of Lemma C.9.1, and expressions of $J(P^{(0)}, P^{(k)})$ we obtain that for attaining a probability error of at most $\delta > 0$ we require at least N samples where

$$\begin{aligned} N &> (1 - \delta + \tfrac{1}{\log(n)}) \frac{2\log(n)}{\frac{n^2}{n^2+1} 2(k-1)\exp(-\frac{2d_{\max}}{k-1})(1-q_{\max})^2(1-\exp(-\frac{d_{\min}}{k-1}))} \\ &\geq \frac{(1-\delta)\exp(\frac{2d_{\max}}{k-1})\log(n)}{(k-1)(1-q_{\max})^2(1-\exp(-\frac{d_{\min}}{k-1}))} \end{aligned}$$

C.9.3 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 C.8), $P^{(0)}, P^{(1)}, \ldots, P^{(n)}$, 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.

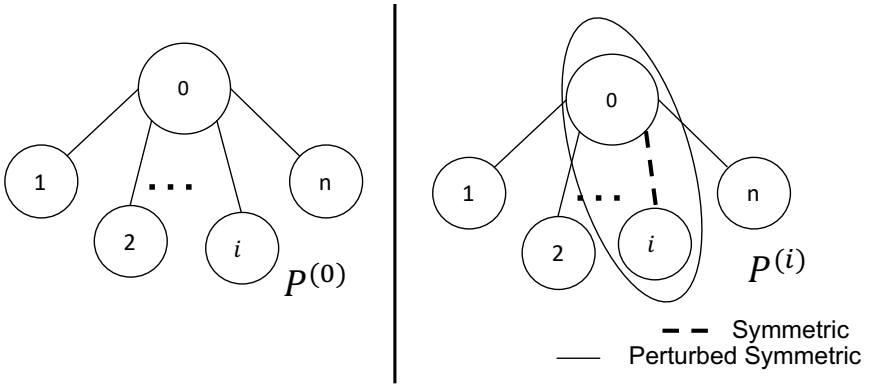


Figure C.8: The family of distributions used for providing lower bound with t_0 dependence. The graphical model corresponding to $P^{(0)}$ is completely identifiable. The graphical model corresponding to $P^{(i)}$, for each $i = 1, \ldots, n$, has edge $\{i, 0\}$ which forms a recoverable leaf cluster, and the rest are all identifiable.

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

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

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:

$$P^{(i)}_{j|0} = (\alpha - \delta)I + (1 - \alpha)\frac{O}{k} + \Delta, \forall j \in [n], j \neq i.$$
$$P^{(i)}_{i|0} = \alpha I + (1 - \alpha)\frac{O}{k}.$$

Recall from Equation (C.38), 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:

$$P^{(0)}(\mathbf{X}) = \frac{1}{k}\prod_{j=1}^{n} P^{(0)}(X_j|X_0), \tag{C.43}$$

$$P^{(i)}(\mathbf{X}) = \frac{1}{k}\prod_{j=1}^{n} P^{(i)}(X_j|X_0). \tag{C.44}$$

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 C.43 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 $k^2 - 2k$ com-

binations of x_i, x_0).

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

Evaluating the remaining terms on similar lines gives us:

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

This gives us:

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

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

$$\det(P_{i|0}^{(i)}) = \alpha^{(k-1)}\left(\left(1-\tfrac{\delta}{\alpha}\right)^k - \left(\tfrac{-\delta}{\alpha}\right)^k\right)$$
$$\det(P_{i|0}^{(i)}) \leq \alpha^{(k-1)}, \quad \det(P_{i|0}^{(i)}) \geq \alpha^{(k-1)}\tfrac{3^k-1}{4^k}$$

Since the distance is bounded by d_{min} and d_{max}, it enforces:

$$d_{max} \geq -(k-1)\log(\alpha) - \log(\tfrac{3^k-1}{4^k}) \geq -(k-1)\log(\alpha) - k\log\left(\tfrac{3}{4}\right),$$
$$d_{min} \leq -(k-1)\log(\alpha)$$
$$\alpha \geq 2\exp(-d_{\max}/(k-1)),\ \alpha \leq \exp(-d_{\min}/(k-1)).$$

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

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

Lower Bound Proof - Part II: We now derive the second part of Theorem 4.6.2, thus concluding its proof.

Plugging the above symmetrized KL bound in Lemma C.9.1 we obtain that for a probability error of at most $\delta > 0$ we require at least N samples where

$$N > (1 - \delta + \tfrac{1}{\log(n)}) \frac{\log(n)}{\frac{n}{n+1}\frac{(k-1)}{8k(k-3)\exp(-2d_{\min}/(k-1))}\left(1 + \frac{1}{1-\exp(-d_{\min}/(k-1))}\right) \times t_0^2}$$
$$\geq \frac{(1-\delta)\exp(-\frac{2d_{\min}}{k-1})(1-\exp(-\frac{d_{\min}}{k-1}))8k(k-3)\log(n)}{(k-1)(2-\exp(-\frac{d_{\min}}{k-1}))t_0^2}$$

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

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

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

C.10 Experiments

We present the performance of our algorithm for the perturbed symmetric model. *All the experiments in this section are for* $k = 4$.

C.10.1 Varying q_{max}

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

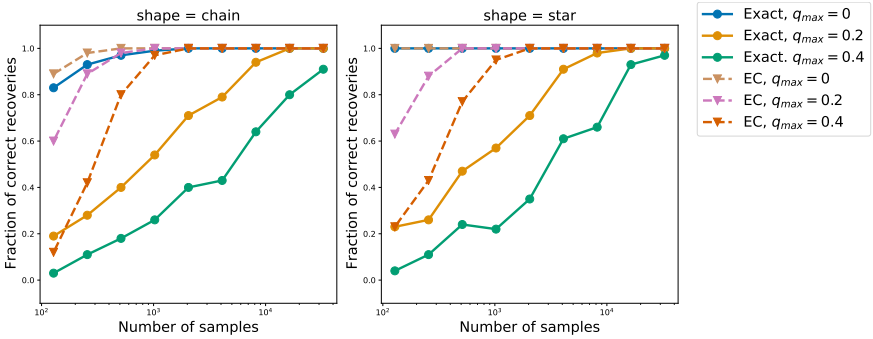


Figure C.9: Comparing the performance of our algorithm for different values of $q_{max} \in \{0, 0.2, 0.4\}$ and different graph shapes - chain, star. Setting: $d_{min} = d_{max} = \exp(-0.7)$, $\delta = 0.04$ # of nodes= 7. We provide results for two cases: i) when the exact underlying tree is recovered, ii) when a tree from the equivalence class is recovered.

Setting: (i) Number of nodes = 7.

(ii) Graph Shape = {Chain, Star}

(iii) Distance of all the adjacent nodes = $\exp(-0.7)$.

(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 C.9.

C.10.2 Varying d

Finally, we present the results for different values of d.

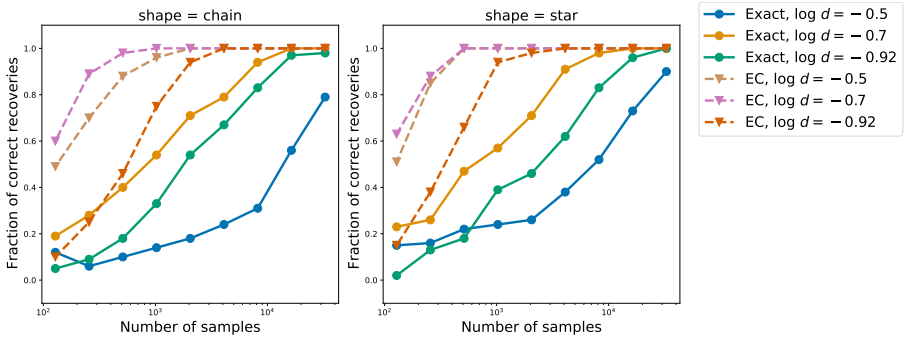


Figure C.10: Comparing the performance of our algorithm for different values of d and different graph shapes - chain, star. Setting: $q_{max} = 0.2$, $\delta = 0.02$ # of nodes= 7. We provide results for two cases: i) when the exact underlying tree is recovered, ii) when a tree from the equivalence class is recovered.

Setting: (i) Number of nodes = 7.

(ii) Graph Shape = {Chain, Star}.

(iii) Distance of all the adjacent nodes $\in \{\exp(-0.5), \exp(-0.7), \exp(-0.92)\}$.

(iv) Error probability is uniformly sampled from $[0, 0.2]$.

(v) $\delta = 0.02$

(vi) Assume access to q_{max}, d_{min} but not to d_{max}, t_0.

(vii) Number of iterations = 100

Takeaway: The algorithm performs the best for intermediate values of d. When the distance is too high or too low, the convergence is slower. Interestingly, the performance for exact recovery and equivalence class recovery show different trends - exact recovery is more difficult when the distance is large whereas the recovery of the equivalence class is more difficult when the distance is small. The results are presented in Figure C.10.

Index

Bibliography

[1] Anima Anandkumar, Daniel J Hsu, Furong Huang, and Sham M Kakade. Learning mixtures of tree graphical models. In *Advances in Neural Information Processing Systems*, pages 1052–1060, 2012.

[2] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine learning research*, 9(Mar):485–516, 2008.

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

[4] 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.

[5] 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.

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

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

[8] 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.

[9] 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.

[10] Stephen G Brush. History of the lenz-ising model. *Reviews of modern physics*, 39(4):883, 1967.

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

[12] Robert Castelo and Alberto Roverato. A robust procedure for gaussian graphical model search from microarray data with p larger than n. *Journal of Machine Learning Research*, 7(Dec):2621–2650, 2006.

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

[14] Yuxin Chen. Learning sparse ising models with missing data.

[15] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 129–136. IEEE, 2010.

[16] 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.

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

[18] George R Cross and Anil K Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):25–39, 1983.

[19] Sanjoy Dasgupta. Learning polytrees. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 134–141. Morgan Kaufmann Publishers Inc., 1999.

[20] Constantinos Daskalakis, Nishanth Dikkala, and Gautam Kamath. Testing ising models. *IEEE Transactions on Information Theory*, 2019.

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

[22] Sacha Epskamp, Lourens J Waldorp, René Mõttus, and Denny Borsboom. The gaussian graphical model in cross-sectional and time-series data. *Multivariate Behavioral Research*, 53(4):453–480, 2018.

[23] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

[24] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

[25] Sergei Konstantinovich Godunov, AG Antonov, OP Kiriljuk, and VI Kostin. *Guaranteed accuracy in numerical linear algebra*, volume 252. Springer Science & Business Media, 2013.

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

[27] 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.

[28] Martin Hassner and Jack Sklansky. The use of markov random fields as models of texture. In *Image Modeling*, pages 185–198. Elsevier, 1981.

[29] Leon Sot (https://math.stackexchange.com/users/214617/leon sot). Simple identity involving q-pochhammer symbol. Mathematics Stack Exchange. URL:https://math.stackexchange.com (version: 2017-01-03).

[30] Takeshi Inagaki. Critical ising model and financial market. *arXiv preprint cond-mat/0402511*, 2004.

[31] Ilse CF Ipsen and Rizwana Rehman. Perturbation bounds for determinants and characteristic polynomials. *SIAM Journal on Matrix Analysis and Applications*, 30(2):762–776, 2008.

[32] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

[33] Ariel Jaimovich, Gal Elidan, Hanah Margalit, and Nir Friedman. Towards an integrated protein–protein interaction network: A relational markov network approach. *Journal of Computational Biology*, 13(2):145–164, 2006.

[34] Majid Janzamin and Animashree Anandkumar. High-dimensional covariance decomposition into sparse markov and independence models. *The Journal of Machine Learning Research*, 15(1):1549–1591, 2014.

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

[36] 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.

[37] Mladen Kolar and Eric P Xing. Estimating sparse precision matrices from data with missing values. 2012.

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

[39] Nicole Krämer, Juliane Schäfer, and Anne-Laure Boulesteix. Regularized estimation of large-scale gene association networks using graphical gaussian models. *BMC bioinformatics*, 10(1):384, 2009.

[40] 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.

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

[42] 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.

[43] Binglin Li, Shuangqing Wei, Yue Wang, and Jian Yuan. Chernoff information of bottleneck gaussian trees. In *Information Theory (ISIT), 2016 IEEE International Symposium on*, pages 970–974. IEEE, 2016.

[44] 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.

[45] Han Liu, Fang Han, Ming Yuan, John Lafferty, Larry Wasserman, et al. High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

[46] Po-Ling Loh and Martin J Wainwright. High-dimensional regression with noisy and missing data: Provable guarantees with non-convexity. In *Advances in Neural Information Processing Systems*, pages 2726–2734, 2011.

[47] Andrey Y Lokhov, Marc Vuffray, Sidhant Misra, and Michael Chertkov. Optimal structure and parameter learning of ising models. *Science advances*, 4(3):e1700791, 2018.

[48] Karim Lounici. High-dimensional covariance matrix estimation with missing observations. *Bernoulli*, 20(3):1029–1058, 2014.

[49] Fabio Martinelli, Alistair Sinclair, and Dror Weitz. The ising model on trees: Boundary conditions and mixing time. In *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pages 628–639. IEEE, 2003.

[50] Nicolai Meinshausen, Peter Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34(3):1436–1462, 2006.

[51] Nicolas Guenon des Mesnards and Tauhid Zaman. Detecting influence campaigns in social networks using the ising model. *arXiv preprint arXiv:1805.10244*, 2018.

[52] Elchanan Mossel, Sébastien Roch, and Allan Sly. Robust estimation of latent tree graphical models: Inferring hidden states with inexact parameters. *IEEE transactions on information theory*, 59(7):4357–4373, 2013.

[53] Konstantinos E. Nikolakakis, Dionysios S. Kalogerias, and Anand D. Sarwate. Learning tree structures from noisy data. In *Proceedings of Machine Learning Research*,

volume 89 of *Proceedings of Machine Learning Research*, pages 1771–1782. PMLR, 2019.

[54] 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.

[55] Konstantinos E Nikolakakis, Dionysios S Kalogerias, and Anand D Sarwate. Non-parametric structure learning on hidden tree-shaped distributions. *arXiv preprint arXiv:1909.09596*, 2019.

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

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

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

[59] Garvesh Raskutti, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. Model selection in gaussian graphical models: High-dimensional consistency of ℓ_1-regularized mle. In *Advances in Neural Information Processing Systems*, pages 1329–1336, 2009.

[60] 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.

[61] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 860–867. Citeseer, 2005.

[62] Adam J Rothman, Peter J Bickel, Elizaveta Levina, Ji Zhu, et al. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

[63] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[64] Elad Schneidman, Michael J Berry II, Ronen Segev, and William Bialek. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, 440(7087):1007, 2006.

[65] Didier Sornette. Physics and financial economics (1776–2014): puzzles, ising and agent-based models. *Reports on progress in physics*, 77(6):062001, 2014.

[66] Nathan Srebro. Maximum likelihood bounded tree-width markov networks. *Artificial intelligence*, 143(1):123–138, 2003.

[67] G.W. Stewart and Ji-Guang Sun. *Matrix Perturbation Theory*. Computer science and scientific computing. Academic Press, 1990.

[68] Makram Talih and Nicolas Hengartner. Structural learning with time-varying components: tracking the cross-section of financial time series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(3):321–341, 2005.

[69] 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.

[70] Vincent YF Tan, Animashree Anandkumar, and Alan S Willsky. Learning gaussian tree models: Analysis of error exponents and extremal structures. *arXiv preprint arXiv:0909.5216*, 2009.

[71] 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.

[72] 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.

[73] 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.

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

[75] Dorina Andru Vangheli. Ising-like statistical models and stock markets real evolution. *Annals of the West University of Timisoara. Physics Series*, 49:170, 2006.

[76] Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2016.

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

[78] Jun-Kun Wang and Shou-de Lin. Robust inverse covariance estimation under noisy measurements. In *International Conference on Machine Learning*, pages 928–936, 2014.

[79] Lingxiao Wang and Quanquan Gu. Robust gaussian graphical model estimation with arbitrary corruption. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3617–3626. JMLR. org, 2017.

[80] Wolfgang Weidlich. The statistical description of polarization phenomena in society. *British Journal of Mathematical and Statistical Psychology*, 24(2):251–266, 1971.

[81] Wolfgang Weidlich. Physics and social science—the approach of synergetics. *Physics reports*, 204(1):1–163, 1991.

[82] Eleanor Wong, Suyash Awate, and P Thomas Fletcher. Adaptive sparsity in gaussian graphical models. In *International Conference on Machine Learning*, pages 311–319, 2013.

[83] John Woods. Markov image modeling. *IEEE Transactions on Automatic Control*, 23(5):846–850, 1978.

[84] Jiansheng Wu. Ising model as a model of multi-agent based financial market.

[85] Shanshan Wu, Sujay Sanghavi, and Alexandros G Dimakis. Sparse logistic regression learns all discrete pairwise graphical models. *arXiv preprint arXiv:1810.11905*, 2018.

[86] 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.

[87] Eunho Yang and Aurélie C Lozano. Robust gaussian graphical modeling with the trimmed graphical lasso. In *Advances in Neural Information Processing Systems*, pages 2602–2610, 2015.

[88] Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11(Aug):2261–2286, 2010.

[89] Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[90] Kun Zhang, Mingming Gong, Joseph Ramsey, Kayhan Batmanghelich, Peter Spirtes, and Clark Glymour. Causal discovery in the presence of measurement error: Identifiability conditions. *arXiv preprint arXiv:1706.03768*, 2017.

[91] W-X Zhou and Didier Sornette. Self-organizing ising model of financial markets. *The European Physical Journal B*, 55(2):175–181, 2007.

Vita

Ashish Katiyar received M.S. in Electrical and Computer Engineering from Texas A&M University in 2017 and B.Tech in Electrical Engineering from the Indian Institute of Technology Jodhpur in 2012. He is currently working towards Ph.D. in Electrical and Computer Engineering at the University of Texas at Austin. His research interests are broadly in theoretical machine learning with current focus on robust graphical model estimation. He has held internship positions at Facebook (2020); InterDigital (2019); IIT Kanpur (2011); University of Technology of Troyes (2010). He was a scientist at Defence Research and Development Organization, India from 2012-2015.

Permanent address: a.katiyar@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.