Covariance Selection Quality and Approximation Algorithms

A DISSERTATION SUBMITTED TO THE GRADUATE DIVISION OF THE UNIVERSITY OF HAWAI'I AT MĀNOA IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

IN

ELECTRICAL ENGINEERING

October 2018

By

Navid Tafaghodi Khajavi

Dissertation Committee :

Prof. Anthony Kuh, Chairperson

Prof. Anders Host-Madsen

Prof. Narayana Prasad Santhanam

Prof. Zhong-Ju Zhang

Prof. Kyungim Baek

To my family

Acknowledgements

First and foremost, I would like to express the deepest appreciation to my PhD advisor Professor Anthony Kuh for all of his support, mentorship and knowledge that he provided me during my PhD. I would also like to thank Professor Anders Høst-Madsen, Professor Narayana Prasad Santhanam, Professor Aleksandar Kavčić, Professor Zhong-Ju Zhang and Professor Kyungim Baek for serving as my committee members. I specially want to thank Professor Aleksandar Kavčić for his mentorship and also his guidance and comments during the early years of my PhD, that have greatly enriched the work. I also would like to express my gratitude to Professor Peter Harremoës for providing valuable feedback and comments. I am sincerely grateful to all my mentors during the course this PhD for generously sharing their knowledge and illuminating views on various aspects of this dissertation.

I would like to take this opportunity to thank all my dear friends who have been for me a family and leave me so many unforgettable memories during my study, specially in Hawaii : Andisheh, Babak, Behrouz, Elahe, Elham, Elyas, Faranak, Hamed, Kamal, Milad, Meysam, Nasir, Pasha, Pegah, Saeed, Sepideh, Shabnam and many others. Last but not least, a special thanks to my family for their unconditional love and priceless support and encouragement throughout the duration of my studies.

This dissertation was supported in part by DOE grant OE0000394, NSF grant ECCS-1310634, the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370, and the University of Hawaii REIS project.

Abstract

This dissertation conducts a study of graphical models, discusses the quality of statistical model selection approximation, and proposes algorithms for model approximation. Graphical models are useful tools for describing the geometric structure of networks in applications that deal with high dimensional data. Learning from these high dimensional data requires large computation power which is not always available due to hardware limitation for different applications. Thus, we need to compromise between model complexity and its accuracy by using the best possible approximation algorithm that chooses a simpler, yet informative model.

The first problem we study in this work is the quality of statistical model selection. We consider the problem of quantifying the quality of approximation model. The statistical model selection often uses a distance measure such as the Kullback-Leibler (KL) divergence between the original distribution and the model distribution to quantify the quality of approximated model distribution. Although the KL divergence is minimized to obtain model approximation in many cases, there are other measures and divergences that can be used to do so. We extend the body of research by formulating the model approximation as a parametric detection problem between the original distribution and the model distribution. The proposed detection framework results in the computation of symmetric closeness measures such as receiver operating characteristic (ROC) and area under the curve (AUC). In particular, we focus on statistical model selection for Gaussian distributions, i.e. the covariance selection [1]. In the case of covariance selection, closeness measures such as KL divergence, reverse KL divergence, ROC, and AUC depend on the eigenvalues of the correlation approximation matrix (CAM). We find expressions for the KL divergence, the log-likelihood ratio, and the AUC as a function of the CAM. We present a simple integral to compute the AUC. In addition, easily computable upper and lower bounds are also found for the AUC to assess the quality of an approximated model. Through some examples and simulation for real and synthetic data, we investigate the quality of the covariance selection for both tree-structured models and non-tree structured models.

The second problem we target in this work is to formulate a general framework and algorithms to

perform covariance selection. We develop a multistage framework for graphical model approximation using a cascade of models such as trees. In particular, we look at the model approximation problem for Gaussian distributions as linear transformations of tree models. This is a new way to decompose the covariance matrix. Here, we propose an algorithm which incorporates the Cholesky factorization method to compute the decomposition matrix and thus can approximate a simple graphical model using a cascade of the Cholesky factorization of the tree approximation transformations. The Cholesky decomposition enables us to achieve a tree structure factor graph at each cascade stage of the algorithm which facilitates the use of the message passing algorithm since the approximated graph has fewer loops compared to the original graph. The overall graph is a cascade of factor graphs with each factor graph being a tree. This is a different perspective on the approximation model, and algorithms such as Gaussian belief propagation can be used on this overall graph. Here, we present theoretical results that guarantee the convergence of the proposed model approximation using the cascade of tree decompositions. In the simulations, we look at synthetic and real data and measure the performance of the proposed framework by comparing the KL divergences.

Student	:	Navid Tafaghodi Khajavi
Student ID#	:	2134-1231
Degree	:	Ph.D.
Field	:	Electrical Engineering
Graduation date	:	December 2018

Title : "Covariance Selection Quality and Approximation Algorithms"

We certify that we have read this dissertation and that, in our opinion, it is satisfactory in scope and quality as a dissertation for the degree of Doctor of Philosophy in Electrical Engineering.

Dissertation Committee :

Names	Signatures
Anthony Kuh, Chairperson	_____
Anders Høst-Madsen	_____
Narayana Prasad Santhanam	_____
Zhong-Ju Zhang	_____
Kyungim Baek, University Representative	_____

Contents

List of Figures

List of Tables

List of Algorithms

1

Introduction

In many applications, we deal with accuracy and model complexity. In signal processing and machine learning, it is a fundamental problem to balance performance quality (i.e. minimizing cost function) with computational complexity. A powerful tool in order to address this trade-off is the statistical model selection and the probabilistic graphical model. Model selection methods provide approximated models with the desired accuracy as needed for different applications. For Gaussian data, this problem is called covariance selection and the KullbackLeibler (KL) divergence is computed to quantify the distance between Gaussian distributions with the original covariance matrix and model covariance matrix (e.g. [3, 4]). Thus the following question is of great interest: How good is the model approximation of the covariance matrix? To answer this question, we need to pick a closeness criterion which has to be coherent and general enough to handle a wide variety of problems and also has asymptotic justification [5]. In the following chapter, we aim to address this question as well as introducing better model selection algorithms.

This chapter aims to provide an introduction to statistical model selection and tree approximation models, with a focus on Gaussian graphical models. We will also introduce definitions and notations necessary for Gaussian model selection and multivariate models used throughout this thesis.

1.1 Graphical Model

Graphical models [6–12] have become a well-known tool to model states in many different engineering applications, due to their effectiveness, flexibility, and simple representation. These models serve as effective tools for both modeling uncertainties through the use of probability theory and an effective approach to coping with complexity through the use of graph theory. Deployment of these models in different applications results in the increased ability to effectively learn and perform inference in large networks. Bayesian networks and Markov networks are the two most common types of graphical models. In this dissertation, we focus on Markov networks and undirected graphs.

Consider a graph with n vertices where vertices represent variables and edges represent dependencies between pair of vertices. Any two vertices that are connected with an edge are consecutive.

Some graph definitions:

1. *Complete graph:* In a complete graph all vertices are connected by an edge.
2. *Cycle in a graph:* A cycle is a path of edges and vertices where a vertex can be reached to from itself.
3. *Clique in a graph:* A clique is complete undirected subset of a graph. A clique that cannot be extended by including another vertex is called a maximal clique. Examples of maximal cliques are circled in figure 1.1.
4. *Chordal (decomposable) graph:* In a chordal graph any two non-consecutive vertices of any cycle of length $n > 3$ are joined by a chord (an edge).
5. *Connected graph:* There is a path (maximum length of $n - 1$) between every pair of vertices in a connected graph.
6. *Tree graph:* A connected graph where any two vertices are connected by exactly one path (a connected loop-free graph).

7. *Chain tree:* Any tree graph where all vertices have at most degree of two.

8. *Star tree:* Any tree graph where all vertices but one have degree of one.

To understand the usefulness of graphical model, we look at the following example which presents an undirected graphical model with 5 vertices.

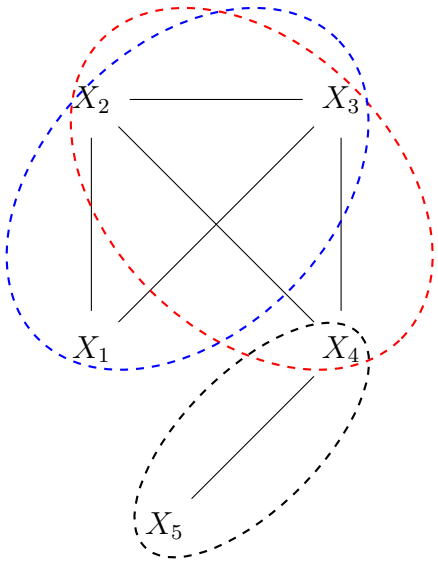

Figure 1.1: An example of graphical model with 5 vertices.

For this thesis our focus is on undirected probabilistic graphs as shown in Figure 1.1. The vertices represent random variables and the edges represent dependencies between the random variables. From this graph note that the random variables X_1, X_2, and X_3 are conditionally independent of X_5 given X_4. This is known as the Markov property. The joint probability distribution of the random variables can then be expressed in terms of products of functions of cliques of the vertices:

$$p(x_1, \ldots, x_5) = \frac{1}{Z}\phi(x_1, x_2, x_3)\phi(x_2, x_3, x_4)\phi(x_4, x_5)$$

where $\phi(.)$'s are potential functions and Z is a normalizing constant.

According to Hammersley-Clifford theorem [13], any distribution that factors by nonnegative potentials over the set of maximal cliques on an undirected graph satisfies the Markov property on the graph. Conversely, if function p is strictly positive and satisfies the Markov property on an undirected graph then it factors by nonnegative potentials over the set of maximal cliques.

As a result of Hammersley-Clifford theorem, any probability distribution function can be expressed as the equation shown above as products of functions of the cliques of the graph.

1.2 Divergences

Definition 1. KL Divergence: *The KL divergence between two multivariate continuous distributions with probability density functions (PDF) $p_{\underline{X}}(\underline{x})$ and $q_{\underline{X}}(\underline{x})$ is defined as*

$$\mathcal{D}\left(p_{\underline{X}}(\underline{x})||q_{\underline{X}}(\underline{x})\right) = \int_{\mathcal{X}} p_{\underline{X}}(\underline{x}) \log \frac{p_{\underline{X}}(\underline{x})}{q_{\underline{X}}(\underline{x})}\, d\underline{x}$$

where $\mathcal{X}$ is the feasible set. ■

KL divergence is an asymmetric divergence. Thus, changing the order of input distributions, results in a different divergence called reverse-KL divergence, $\mathcal{D}\left(q_{\underline{X}}(\underline{x})||p_{\underline{X}}(\underline{x})\right)$. One way to compute a symmetric divergence is to sum the KL divergence and its reverse KL divergence.

Definition 2. Jeffreys Divergence: *The Jeffreys divergence between two multivariate continuous distributions with PDFs $p_{\underline{X}}(\underline{x})$ and $q_{\underline{X}}(\underline{x})$ is defined as*

$$\mathcal{D}_{\mathcal{J}}\left(p_{\underline{X}}(\underline{x}),\, q_{\underline{X}}(\underline{x})\right) = \mathcal{D}\left(p_{\underline{X}}(\underline{x})||q_{\underline{X}}(\underline{x})\right) + \mathcal{D}\left(q_{\underline{X}}(\underline{x})||p_{\underline{X}}(\underline{x})\right)$$

where $\mathcal{D}(.||.)$ is the KL divergence. ■

1.3 Multivariate Gaussian models

Random vector $\underline{X} \in \mathbb{R}^n$ has multivariate Gaussian distribution $\mathcal{N}(\underline{\mu}, \mathbf{\Sigma})$ with mean vector $\underline{\mu} \in \mathbb{R}^n$ and covariance matrix $\mathbf{\Sigma} \in \mathbf{S}^n_{\succ 0}$ [1] if its has the following density function

$$p_{\underline{X}}(\underline{x}) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{\Sigma}|}} e^{-\frac{1}{2}(\underline{x}-\underline{\mu})^T \mathbf{\Sigma}^{-1}(\underline{x}-\underline{\mu})}, \quad \underline{x} \in \mathbb{R}^n.$$

1. $\mathbf{S}^n_{\succ 0}$ is the set of all n-dimensional positive-definite covariance matrices.

1.3.1 Log-likelihood

Let's assume a multivariate Gaussian statistical model. Given i.i.d. observations $\mathcal{X} : \{\underline{x}_1, \ldots, \underline{x}_m\}$ drawn from multivariate Gaussian distribution, the log-likelihood function for a Gaussian model is

$$l_{\widetilde{\mathbf{\Sigma}}}(\underline{x}_1, \ldots, \underline{x}_m) = \frac{m}{2} \log |\widetilde{\mathbf{\Sigma}}^{-1}| - \frac{m}{2} \text{tr } (\widetilde{\mathbf{\Sigma}}^{-1}\mathbf{S}) \tag{1.1}$$

where $\widetilde{\mathbf{\Sigma}}$ is the covariance matrix of the model and

$$\mathbf{S} = \frac{1}{m-1} \sum_{i=1}^{m} (\underline{x}_i - \bar{\underline{x}})(\underline{x}_i - \bar{\underline{x}})^T$$

is the sample covariance matrix and

$$\bar{\underline{x}} = \frac{1}{m} \sum_{i=1}^{m} \underline{x}_i$$

is the sample mean.

In model selection problem, it is often easier to work with the inverse $\widetilde{\mathbf{\Pi}} = \widetilde{\mathbf{\Sigma}}^{-1}$ covariance matrix which is called precision or concentration matrix.

1.3.2 KL divergence

The KL divergence between two multivariate Gaussian distribution is defined as

$$\mathcal{D}(\mathcal{N}(\underline{\mu}_1, \mathbf{\Sigma}_1)||\mathcal{N}(\underline{\mu}_2, \mathbf{\Sigma}_2)) = \frac{1}{2} \left(\text{tr}(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1) + (\underline{\mu}_2 - \underline{\mu}_1)^T \mathbf{\Sigma}_2^{-1} (\underline{\mu}_2 - \underline{\mu}_1) - n + \log \frac{|\mathbf{\Sigma}_2|}{|\mathbf{\Sigma}_1|} \right)$$

Without loss of generality in this thesis, we assume that the mean vector is $\underline{0}$. We focus on approximation of a model covariance matrix $\widetilde{\mathbf{\Sigma}}$, subject to desired graphical model conditions. Thus, we can simplify the KL divergence between the actual Gaussian distribution and its approximation as

$$\mathcal{D}(\mathcal{N}(\underline{0}, \mathbf{\Sigma})||\mathcal{N}(\underline{0}, \widetilde{\mathbf{\Sigma}})) = \frac{1}{2} \left(\text{tr}(\mathbf{\Sigma}\widetilde{\mathbf{\Sigma}}^{-1}) - n + \log |\widetilde{\mathbf{\Sigma}}\mathbf{\Sigma}^{-1}| \right) \tag{1.2}$$

where $\mathbf{\Sigma}$ is the actual covariance matrix. We can always use sample covariance matrix $\mathbf{S}$, if knowledge of the actual covariance matrix $\mathbf{\Sigma}$ is not available. In this scenario, we use $\mathcal{D}(\mathcal{N}(\underline{0}, \mathbf{S})||\mathcal{N}(\underline{0}, \widetilde{\mathbf{\Sigma}}))$ as the optimization cost function to compute model approximation covariance matrix.

1.3.3 Properties of a covariance matrix

Gaussian graphical model explores conditional independence between random variables. Let $\underline{X} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma})$ and also let's assume that random variables X_i and X_j are the i-th and j-th elements of random vector $\underline{X}$. Then, X_i and X_j are jointly Gaussian and we have

1) **Independence:** X_i and X_j are independent (uncorrelated in general without jointly Gaussian assumption) if and only if $\mathbf{\Sigma}_{ij} = 0$.
2) **Diagonals of precision matrix:** Diagonals of precision matrix $\mathbf{\Pi} = \mathbf{\Sigma}^{-1}$ are reciprocals of conditional variances given all the other random variables

$$\mathbf{\Pi}_{ii} = \frac{1}{var(X_i|\underline{X}_{\mathcal{V}\setminus i})}.$$

3) **Conditional covariances:** Conditional covariances between X_i and X_j give all the other random variables can be computed as

$$cov(X_i, X_j|\underline{X}_{\mathcal{V}\setminus\{i,j\}}) = \frac{\mathbf{\Pi}_{ij}}{\mathbf{\Pi}_{ii}\mathbf{\Pi}_{jj} - \mathbf{\Pi}_{ij}^2}.$$

4) **Conditional uncorrelatedness:** Random variables X_i and X_j are conditionally uncorrelated give all the other random variables (conditional independence for jointly Gaussian random variables) if and only if $\mathbf{\Pi}_{ij} = 0$.

1.4 Dempster covariance selection and Chow Liu algorithm

Chapter 2 discusses the model selection problem for Gaussian distributions and algorithms for finding optimal graphs when they are tree. The model selection problem is based on work by Dempster [1]. For tree structured graphs, we can find the optimal tree minimizing and information divergence, the Kullback Leibler (KL) divergence using the Chow Liu algorithm [3]. Both these papers are classical papers that have been used in signal processing, statistics, communications, and information theory.

1.5 Application

Many engineering and computer science applications require using graphs to model dependencies between nodes of the graph. These applications include a diversity of areas from social networking to biomedical applications to transportation models to energy models [14]. Sparse modeling has many applications in distributed signal processing and machine learning over graphs. One of its applications is for the electric power grid at the distribution level. The *smart grid* is a promising solution that delivers reliable energy to consumers through the power grid when there are uncertainties such as distributed renewable energy generation sources. Smart grid technologies such as smart meters and communication links are added to the distribution grid in order to obtain the high dimensional, real-time data and information and overcome uncertainties and unforeseen faults. The future grid will incorporate distributed renewable energy generation such as solar photovoltaics (PV), with these energy sources being intermittent and highly correlated. Thus, modeling is essential for signal processing and implementation of the smart grid.

1.5.1 Smart grid example

An electric distribution grid with measurements, distributed renewable energy sources, and decision making capabilities is referred to as the smart grid [15]. Smart grids often have large number of states (e.g. node voltages) making it computationally inefficient to gather the data and perform central state estimation in real-time fashion. Moreover, central state estimation requires many communication links between sensors on the distribution grid resulting in large costs. In contrast, distributed state estimators can give reasonably good estimates for large power grid systems in real-time while decreasing the number of necessary communications links. This fact causes a trade off between computational time and accuracy of estimation. The distributed state estimation method over a factor graph based on loopy Gaussian BP proposed in [16] can perform estimation in real-time fashion. To assure the convergence of loopy Gaussian BP algorithm, [16] considers simple models of the distributed renewable energy sources by approximating covariance matrices of the distributed renewable energy sources with simpler covariance matrices that have tree-like structures. However tree-like structures are poor approximators when the number of nodes is large [17–19] leading us to consider more complex graphical structures. Recently, [20] also considered a BP approach for

state estimation in power grid. The factor graph representation of a microgrid is given in figure 1.2. This figure shows the approximation of a covariance matrix with a first order Markov chain model. Here we need Model approximation in order to decrease the number of loops due to penetration of correlated renewable sources. Note that, in the middle, we have the loopy factor graph with many short loops due to complete factor graph of the covariance matrix. In the left, we remove many of those short loops by using the approximated covariance matrix factor graph.

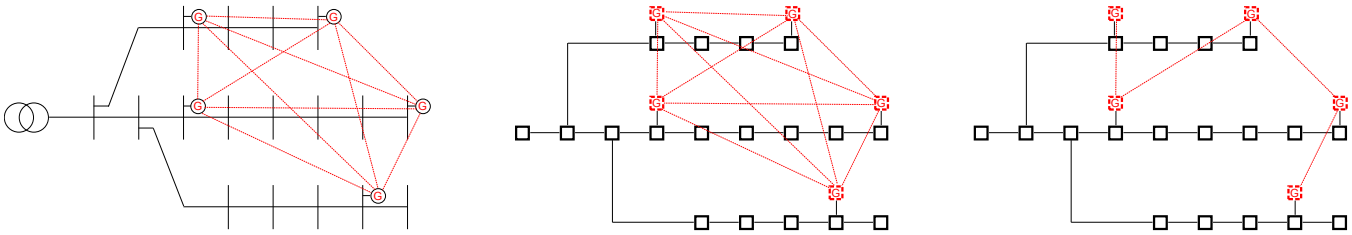

Figure 1.2: Microgrid with correlated renewable energy sources (left), its Factor Graph (Middle) and reduced Factor Graph by Chain approximation (Right).

1.6 Thesis Main Contributions

1.6.1 Quality of statistical model selection with focus on covariance selection

We formulate a parametric detection problem between the original distribution and the approximation model distribution in order to quantify the quality of any model selection algorithm. The proposed parametric detection problem is a different approach which gives us a broader view by determining whether a particular model is a good approximation or not. Additionally, this formulation leads to the calculation of the log-likelihood ratio test (LLRT) statistic, the KL divergence and the reverse KL divergence as well as the receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) where the AUC is used as the accuracy measure for the detection problem. The AUC does not depend on a specific operating point on the ROC and broadly summarizes the entire detection framework. It also effectively combines the detection probability and the false-alarm probability into one measure. The AUC determines the inherent ability of the test to distinguish (in conventional detection problem) or not to distinguish (in the model approximation

problem) between two hypotheses/models. More specifically, the detection formulation and particularly the AUC gives us additional insight about any approximation since it is a way to formalize the model approximation problem.

After formulation of the general problem, we shift focus to the Gaussian data, i.e. the covariance selection [1]. For Gaussian data, the LLRT statistic simplifies to an indefinite quadratic form. For this case, we can define a key quantity which is the correlation approximation matrix (CAM). The CAM is the product of the original correlation matrix and the inverse of the model approximation correlation matrix. For Gaussian data, this matrix contains all the information needed to compute the information divergences, the ROC curve and the area under the ROC curve, i.e. the AUC. We also show the relationship between the CAM, the AUC and the Jeffreys divergence [21], the KL divergence and the reverse KL divergence. We present an analytical expression to compute the AUC for a given CAM that can be efficiently evaluated numerically. We then show the relation between the AUC, the KL divergence, the LLRT statistics and the ROC curve. We also present analytical upper and lower bounds for the AUC which only depend on eigenvalues of the CAM.

Finally, through some examples and simulation for real and synthetic data, we explore model selection quality using the proposed detection framework for both tree structured and non-tree structured approximation models.

1.6.2 General framework and algorithms to perform covariance selection

We have proposed a new optimization method for covariance selection. Mainly, our method is a multi-stage graphical approximation using cascade of models such as trees. This cascade approximation method for Gaussian distributions is a linear transformation of tree models and a new decomposition of the covariance matrix. The purpose of this method is also to reduce the computational complexity of distributed algorithms in various applications while maintaining the desired approximation quality. To achieve this goal we approximate the Gaussian graphical model with a simpler, more tractable model.

We consider jointly Gaussian data and use cascade of tree transformation decompositions in order to perform model approximation for graphical models. The tree structure model is considered since this structure is simple and the optimal solution that minimizes the KL divergence can be

easily computed using he Chow-Liu algorithm [3]. Furthermore, the tree structure model is a loop-free model and simplifies the implementation of distributed algorithms such as Gaussian BP. The cascade tree framework enables us to approximate a complex model with multiple stages of simple tractable models such as the tree structured model. We pick trees as the model and the Cholesky decomposition to factor the tree structured covariance matrix at each stage of the cascade algorithm. Implementation of the Cholesky decomposition with the proper node ordering (permutation matrix) enables us to draw a *tree structured* factor graph for each step of the cascade tree decomposition transformation. This property can facilitates the use of Gaussian BP algorithm over the aforementioned factor graph. Note that each successive additional tree reduces the KL divergence. Also, we conjecture that the KL divergence goes to 0 as we added more and more trees. Furthermore, selecting star networks and with proper node ordering, we can get to exact representation of the covariance matrix at most with n-1 cascades.

We perform some simulations to investigate the performance of the proposed method. In our simulations, we are looking at synthetic and real data and compare the performance of the proposed framework by comparing KL divergences. We also consider the singular value decomposition (SVD) and compare its performance to the Cholesky decomposition. Our simulation results also confirm the advantages of the cascade tree framework in the sense of lowering the overall KL divergence between the original and the model distribution.

1.7 Thesis Outline

The rest of the thesis is organized as follows. In chapter 2 we introduce the detection problem framework for model selection and investigate the quality of a model approximation algorithm. We present the theoretical results related to the quality of an approximation model. Chapter 3 extends the work in chapter 2 by looking at more complex approximation models such as non-tree structured graphs and also graphical models with junction tree representation. This chapter also gives some simple tree structured examples as well as non-tree examples on synthetic and real data. In chapter 4, we represent our new covariance matrix decomposition method using a cascade of linear transformations represented by trees and its usefulness in model approximation for graphical model. We also show how the presented methodology in this chapter can be applied. Finally, We

end with some concluding remarks and future research directions in Chapter 5. We provide the proofs of theorems and lemmas in Appendix A.

2

The Covariance Selection Problem using a Detection Problem Formulation: Theoretical Analysis

This chapter investigates the problem of quantifying the quality of a general statistical model selection for graphs describing dependencies between random variables focusing on the covariance selection quality for jointly Gaussian random vectors. There is a mature body of literature on statistical model selection, but still, it is not obvious how good are these models. Statistical model selection often uses a distance measure such as the Kullback-Leibler (KL) divergence between the original distribution and the model distribution to quantify the quality of the model approximation. In this chapter, we look at this problem from a different angle. We extend the body of research by formulating the model approximation as a detection problem between the original distribution and

the model distribution. We mainly focus on the covariance selection problem and AUC bounds to quantify its quality.

2.1 Introduction

As we mentioned before, graphical models are useful tools for describing the geometric structure of networks in numerous applications such as energy, social, sensor, biological, and transportation networks [14] that deal with high dimensional data. Learning from these high dimensional data requires large computation power which is not always available [6, 8], due to hardware limitation which forces us to compromise between the accuracy and time complexity by using the best possible approximation algorithm given the required constrained graph. In other words, the main concern is to compromise between model complexity and its accuracy by choosing a simpler, yet informative and useful model. To address this concern, many approximation algorithms are proposed for model selection and imposing structure given data. A broad study of graphical models is also done in [6, 8, 10]. For the Gaussian distribution, the covariance selection problem is first presented and studied by Dempster [1]. More comprehensive covariance selection study using penalized normal likelihood has been done in [22] and its references.

The ultimate purpose of the covariance selection problem that we discuss here is to reduce the computational complexity and speed up various applications by using the power of graphical models in modeling structures. One of the special approximation models that we consider in this dissertation is the tree approximation model. Tree approximation algorithms are among the algorithms that reduce the number of computations to get quicker approximate solutions to a variety of problems. If a tree model is used, then distributed estimation algorithms such as message passing algorithm and the belief propagation algorithm [23–26] can easily be applied on factor graphs [27] and these algorithms are guaranteed to converge to the maximum likelihood solution quickly. The convergence speed depends on the longest path [28] between any two vertices in a tree graph which is at most the number of vertices minus one (chain trees).

There are algorithms in the literature that can be used to approximate the covariance matrix and the inverse covariance matrix such that the jointly Gaussian distributions associated with the covariance matrix can be represented by a more sparse graph representation while retaining desired

accuracy. Some of these algorithms are as follows: the Chow-Liu minimum spanning tree (MST) [3], the first order Markov chain approximation [16], penalized likelihood methods such as LASSO [29] and graphical LASSO [30], and sparse approximation a conditional density of latent variables (Variational inference) [31, 32]. Among these algorithms, we introduce the Chow-Liu MST algorithm for Gaussian distribution is chapter 1. The algorithm's goal is to find the optimal tree structure using a Kullback-Leibler (KL) divergence cost function. This algorithm constructs a weighted graph by computing pairwise mutual information and then utilizes one of the MST algorithms such as the Kruskal algorithm [33] or the Prim algorithm [34]. The first order Markov chain approximation method uses a regret cost function to output first order Markov chain structured graph [16] by utilizing a greedy type algorithm. Penalized likelihood methods use an L1-norm penalty term in order to sparsify the graph representation and eliminate some edges. Recently, a tree approximation in a linear, underdetermined model was proposed in [35] where the solution is based on expectation, maximization (EM) algorithm combined with the Chow Liu algorithm.

Here, we investigate the quality of the model selection, focusing on the Gaussian case, i.e. covariance selection. We ask the following important question: *"is the model approximation of the covariance matrix for the Gaussian model a good approximation?"* To answer this question, we need to pick a closeness criterion which has to be coherent and general enough to handle a wide variety of problems and also has asymptotic justification [5]. In many applications, the Kullback-Leibler (KL) divergence has been proposed as a closeness criterion between the original distribution and its model approximation distribution [1] and [3]. Besides that, other closeness measures and divergences are used for the model selection. One example is the use of the reverse KL divergence as the closeness criterion in variational methods to learn the desired approximation structure [32].

In this chapter, we bring a different perspective to the model approximation problem by formulating a general detection problem. The detection problem formulation leads to the calculation of the log-likelihood ratio test (LLRT) statistic, the KL divergence and the reverse KL divergence as well as the receiver operating characteristic (ROC) curve and the area under the ROC curve (AUC) where the AUC is used as the accuracy measure for the detection problem. The detection problem formulation is a different approach which gives us a broader view by determining whether a particular model is a good approximation or not. The AUC does not depend on a specific operating point on the ROC and broadly summarizes the entire detection framework. It also effectively combines the

detection probability and the false-alarm probability into one measure. The AUC determines the inherent ability of the test to distinguish (in conventional detection problem) or not to distinguish (in the model approximation problem) between two hypotheses/models. More specifically, the detection formulation and particularly the AUC gives us additional insight about any approximation since it is a way to formalize the model approximation problem. For Gaussian data, the LLRT statistic simplifies to an indefinite quadratic form. We define a key quantity which is the correlation approximation matrix (CAM). The CAM is the product of the original correlation matrix and the inverse of the model approximation correlation matrix. For Gaussian data, this matrix contains all the information needed to compute the information divergences, the ROC curve and the area under the ROC curve, i.e. the AUC. We also show the relationship between the CAM, the AUC and the Jeffreys divergence [21], the KL divergence and the reverse KL divergence. We present an analytical expression to compute the AUC for a given CAM that can be efficiently evaluated numerically. We then show the relation between the AUC, the KL divergence, the LLRT statistics and the ROC curve. We also present analytical upper and lower bounds for the AUC which only depend on eigenvalues of the CAM.

The rest of this chapter is organized as follows. In section 2.2 we give a general framework for the detection problem and the corresponding sufficient test statistic, the log-likelihood ratio test. The log-likelihood ratio test for Gaussian data and its distribution under both hypotheses are also presented in this section. The ROC curve and the AUC definition, as well as an analytical expression for the AUC, are given in section 2.5. Section 2.6 provides analytical lower and upper bounds for the AUC. The lower bound for the AUC uses the Chernoff bound and is a function of the CAM eigenvalues. The upper bound is obtained by finding a parametric relationship between the AUC and the KL and reverse KL divergences. Finally, section 2.7 summarizes results of this chapter.

2.2 Detection Problem Framework

In this section, we present a framework to quantify the quality of a model selection. More specifically, we formulate a detection problem to distinguish between the covariance matrix of a multivariate normal distribution and an approximation of the aforementioned covariance matrix based on the given model.

2.2.1 Model selection problem

We want to approximate a multivariate distribution by the product of lower order component distributions [36]. Let random vector $\underline{X} \in \mathbb{R}^n$, have a distribution with parameter Θ, i.e. $\underline{X} \sim f_{\underline{X}}(\underline{x})$. We want to approximate the random vector $\underline{X}$, with another random vector associated with the desired model[1]. Let the model random vector $\underline{X}_{\mathcal{M}} \in \mathbb{R}^n$ have a distribution with parameter $\Theta_{\mathcal{M}}$, associated with the desired model, i.e. $\underline{X} \sim f_{\underline{X}_{\mathcal{M}}}(\underline{x})$. Also, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathcal{M}})$ be the graph representation of the model random vector $\underline{X}_{\mathcal{M}}$ where sets $\mathcal{V}$ and $\mathcal{E}_{\mathcal{M}}$ are the set of all vertices and the set of all edges of the graph representing of $\underline{X}_{\mathcal{M}}$, respectively. Moreover, $\mathcal{E}_{\mathcal{M}} \subseteq \psi$ where ψ is the set of all edges of complete graph with vertex set $\mathcal{V}$.

Remark: Covariance selection and tree structured models for Gaussian distributions is discussed in chapter 1.

2.2.2 General detection framework

The model selection is extensively studied in the literature [1]. Minimizing the KL divergence between two distributions or the maximum likelihood criterion are proposed in many previous works in literature to quantify the quality of the model approximation. A different way to look at the problem of quantifying the quality of the model approximation is to formulate a detection problem [37]. Given the set of data, the goal of the detection problem is to distinguish between *the null hypothesis* and *the alternative hypothesis*. To set up a detection problem for the model selection, we need to define these two hypotheses as follows

- The null hypothesis, $\mathcal{H}_0$: data is generated using the known/original distribution,
- The alternative hypothesis, $\mathcal{H}_1$: data is generated using the model/approximated distribution.

Given the set up for the null hypothesis and the alternative hypothesis, we need to define a test statistic to quantify the detection problem. The likelihood ratio test (the Neyman-Pearson (NP) Lemma [38]) is the most powerful test statistic where we first define the log-likelihood ratio test

1. Examples of possible models: tree structure, sparse structure, and Markov chain.

(LLRT) as

$$l(\underline{x}) = \log \frac{f_{\underline{X}}(\underline{x}|\mathcal{H}_1)}{f_{\underline{X}}(\underline{x}|\mathcal{H}_0)} = \log \frac{f_{\underline{X}_{\mathcal{M}}}(\underline{x})}{f_{\underline{X}}(\underline{x})}$$

where $f_{\underline{X}}(\underline{x}|\mathcal{H}_0)$ is the random vector $\underline{X}$ distribution under the null hypothesis while $f_{\underline{X}}(\underline{x}|\mathcal{H}_1)$ is the random vector $\underline{X}$ distribution under the alternative hypothesis.

Let $l(\underline{X})$ be the LLRT statistic random variable. Then, we define *the false-alarm probability* and *the detection probability* by comparing the LLRT statistic under each hypothesis with a given threshold, τ, and computing the following probabilities

- The false-alarm probability, $P_0(\tau)$, under the null hypothesis, $\mathcal{H}_0$: $P_0(\tau) = \Pr(l(\underline{X}) \geqslant \tau|\mathcal{H}_0)$,
- The detection probability, $P_1(\tau)$, under the alternative hypothesis, $\mathcal{H}_1$: $P_1(\tau) = \Pr(l(\underline{X}) \geqslant \tau|\mathcal{H}_1)$.

The Neyman-Pearson Lemma [38] is the most powerful test at a given false-alarm rate (significant level). The most powerful test is defined by setting the false-alarm rate $P_0(\tau) = \bar{P}_0$ and then computing the threshold value $\tau = \tau_0$ such that $\Pr(l(\underline{X}) \geqslant \tau_0|\mathcal{H}_0) = \bar{P}_0$.

Throughout this dissertation, we may use other notation such as the KL divergence between two covariance matrices for zero-mean Gaussian distribution case or the KL divergence between two random variables in order to present the KL divergence between two distributions.

Proposition 1. *Expectation of the LLRT statistic under each hypothesis is*

- $\mathrm{E}\,(l(\underline{X})|\mathcal{H}_0) = -\mathcal{D}(f_{\underline{X}}(\underline{x}|\mathcal{H}_0)||f_{\underline{X}}(\underline{x}|\mathcal{H}_1)) = -\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}}}(\underline{x}))$,
- $\mathrm{E}\,(l(\underline{X})|\mathcal{H}_1) = \mathcal{D}(f_{\underline{X}}(\underline{x}|\mathcal{H}_1)||f_{\underline{X}}(\underline{x}|\mathcal{H}_0)) = \mathcal{D}(f_{\underline{X}_{\mathcal{M}}}(\underline{x})||f_{\underline{X}}(\underline{x}))$.

Proof. Proof is based on the KL divergence definition. ■

Remark: Relationship between the NP lemma and the KL divergence is previously stated in [39] with the similar straightforward calculation, where the LLRT statistic loses power when the wrong distribution is used instead of the true distribution for one of these hypotheses.

In a regular detection problem framework, the NP decision rule is to accept the hypothesis $\mathcal{H}_1$ if the LLRT statistic, $l(\underline{x})$, exceeds a critical value, and reject it otherwise. Furthermore, the critical value is set based on the rejection probability of the hypothesis $\mathcal{H}_0$, i.e. false-alarm probability. However, we pursue a different goal in the approximation problem scenario. Our goal is to approximate

a model distribution with PDF $f_{\underline{X}_{\mathcal{M}}}(\underline{x})$, as close as possible to the given distribution with PDF $f_{\underline{X}}(\underline{x})$. The closeness criterion is based on the modified detection problem framework where we compute the LLRT statistic and compare it with a threshold. In an ideal case where there is no approximation error, the detection probability must be equal to the false-alarm probability for the optimal detector at all possible thresholds, i.e. the receiver operating characteristic (ROC) curve [40] that represents best detectors for all threshold values should be a line of slope 1 passing through the origin.

In the next subsection, we assume that the random vector $\underline{X}$ has zero-mean Gaussian distribution. Thus, the covariance matrix of the random vector $\underline{X}$ is the parameter of interest in the model selection, i.e. covariance selection.

2.2.3 Multivariate Gaussian distribution and model selection

Let random vector $\underline{X} \in \mathbb{R}^n$, have a zero-mean jointly Gaussian distribution with covariance matrix $\mathbf{\Sigma}_{\underline{X}}$, i.e. $\underline{X} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\underline{X}})$ where the covariance matrix $\mathbf{\Sigma}_{\underline{X}}$ is positive-definite, $\mathbf{\Sigma}_{\underline{X}} \succ 0$. Here, the null hypothesis, $\mathcal{H}_0$, is the hypothesis that the parameter of interest is known and is equal to $\mathbf{\Sigma}_{\underline{X}}$ while the alternative hypothesis, $\mathcal{H}_1$, is the hypothesis that the random vector $\underline{X}$ is replaced by the model random vector $\underline{X}_{\mathcal{M}}$. In this scenario, the model random vector $\underline{X}_{\mathcal{M}}$ has a zero-mean jointly Gaussian distribution (the model approximation distribution) with covariance matrix $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}$ i.e. $\underline{X}_{\mathcal{M}} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}})$ where the covariance matrix $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}$ is also positive-definite, $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}} \succ 0$. Thus, the LLRT statistic for the jointly Gaussian random vectors ($\underline{X}$ and $\underline{X}_{\mathcal{M}}$) is simplified as

$$l(\underline{x}) = \log \frac{\mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}})}{\mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\underline{X}})} = -c + k(\underline{x}) \tag{2.1}$$

where $c = -\frac{1}{2}\log\left(|\mathbf{\Sigma}_{\underline{X}}\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}^{-1}|\right)$ is a constant and $k(\underline{x}) = \underline{x}^T\mathbf{K}\underline{x}$ where $\mathbf{K} = \frac{1}{2}(\mathbf{\Sigma}_{\underline{X}}^{-1} - \mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}^{-1})$ is an indefinite matrix with both positive and negative eigenvalues.

We define the correlation approximation matrix (CAM) associated with the covariance selection problem and dissimilarity parameters of the CAM as follows.

Definition 3. Correlation approximation matrix. *The CAM for the covariance selection*

problem is defined as $\mathbf{\Delta} \triangleq \mathbf{\Sigma}_{\underline{X}} \mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}^{-1}$ *where* $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}$ *is the model covariance matrix.* ■

Definition 4. Dissimilarity parameters for covariance selection problem. *Let* $\alpha_i \triangleq \lambda_i + \lambda_i^{-1} - 2$ *for* $i \in \{1, \ldots, n\}$ *be dissimilarity parameters of the CAM correspond to the covariance selection problem where* $\lambda_i > 0$ *for* $i \in \{1, \ldots, n\}$ *are eigenvalues of the CAM.* ■

Remark: The CAM is a positive definite matrix. Moreover, eigenvalues of the CAM contains all information necessary to compute cost functions associated with the model selection.

2.3 Dempster covariance selection

In 1972, Dempster introduced the covariance selection approach where given an empirical covariance matrix, the goal is to approximate a sparse inverse covariance matrix to estimate data. Given a covariance matrix with dense inverse, this approach can also be looked at as an approximation of a covariance matrix with sparse inverse given some model criteria (e.g. sparse graphical representation).

Recall that we define the graphical model of model $\mathcal{M}$ as $\mathcal{G}_{\mathcal{M}} = (\mathcal{V}, \mathcal{E}_{\mathcal{M}})$ where $\mathcal{V}$ is a set of all nodes and $\mathcal{E}_{\mathcal{M}} \subseteq \psi$ is a set of edges that represents the model $\mathcal{M}$.

Theorem 1. Covariance Selection [1]. *Given a multivariate Gaussian distribution with covariance matrix* $\mathbf{\Sigma}_{\underline{X}} \succ 0$, $f_{\underline{X}}(\underline{x})$, *and a model* $\mathcal{M}$, *there exists a unique approximate multivariate Gaussian distribution with covariance matrix* $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}} \succ 0$, $f_{\underline{X}_{\mathcal{M}}}(\underline{x})$, *that minimize the KL divergence,* $\mathcal{D}(f_{\underline{X}}(\underline{x}) || f_{\underline{X}_{\mathcal{M}}}(\underline{x}))$ *and satisfies the covariance selection rules, i.e. the model covariance matrix satisfies the following covariance selection rules*

- $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}(i,i) = \mathbf{\Sigma}_{\underline{X}}(i,i), \qquad \forall\ i \in \mathcal{V}$
- $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}(i,j) = \mathbf{\Sigma}_{\underline{X}}(i,j), \qquad \forall\ (i,j) \in \mathcal{E}_{\mathcal{M}}$
- $\mathbf{\Sigma}^{-1}_{\underline{X}_{\mathcal{M}}}(i,j) = 0, \qquad \forall\ (i,j) \in \mathcal{E}^c_{\mathcal{M}}$

where the set $\mathcal{E}^c_{\mathcal{M}} = \psi - \mathcal{E}_{\mathcal{M}}$ *represents the complement of the set* $\mathcal{E}_{\mathcal{M}}$ *and* ψ *is the set of all edges of a complete graph. Note that, Since* $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}$ *is symmetric,* $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}(j,i) = \mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}(i,j)$. ■

In other words, covariance selection rules state that

1) model approximation variances are the same as actual variances,

2) model approximation covariances between two connected nodes in graphical representation of the model $\mathcal{M}$ are the same as actual covariances,

3) coefficients of the model approximation precision matrix (inverse of the model approximation covariance matrix) are zero at all other positions (expressing the conditional independence).

2.4 Chow-Liu minimum spanning tree

Tree structured graphical models are one of the simplest and important class of undirected graphs. Finding the minimum spanning tree for an edge-weighted undirected graph is one of the well-known and well-studied problems in the literature [33, 34, 41]. For jointly Gaussian distributions, finding the optimal tree structured model reduces to finding the minimum spanning tree for an edge-weighted undirected graph [3].

The maximum order of the lower order distributions in tree approximation problem is two, i.e. no more than pairs of variables. Let $\underline{X}_{\mathcal{T}} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\underline{X}_{\mathcal{T}}})$ have the graph representation $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ where $\mathcal{E}_{\mathcal{T}} \subseteq \psi$ is a set of edges that represents a tree structure. Let $\underline{X}_r \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\underline{X}_r})$ have the graph representation $\mathcal{G}_r = (\mathcal{V}, \mathcal{E}_r)$ where $\mathcal{E}_r \subseteq \mathcal{E}_{\mathcal{T}}$ is the set of all edges in the graph of $\underline{X}_r$. The joint PDF for elements of the random vector $\underline{X}_r$ can be represented by joint PDFs of two variables and marginal PDFs in the following convenient form

$$f_{\underline{X}_r}(\underline{x}_r) = \prod_{(u,v)\in\mathcal{E}_r} \frac{f_{\underline{X}^u,\underline{X}^v}(\underline{x}^u, \underline{x}^v)}{f_{\underline{X}^u}(\underline{x}^u) f_{\underline{X}^v}(\underline{x}^v)} \prod_{u\in\mathcal{V}} f_{\underline{X}^u}(\underline{x}^u), \tag{2.2}$$

where $\underline{X}^u$ is the u-th element of random vector $\underline{X}$. Using equation (4.1) we can then easily construct a tree using iterative algorithms (such as the Chow-Liu algorithm [3] combined with the Kruskal [33] algorithm or the Prim [34] algorithm) by adding edges one at a time [42]. Consider the sequence of random vectors $\underline{X}_r$ with $0 \leqslant r \leqslant |\mathcal{E}_{\mathcal{T}}|$, where $\underline{X}_r$ is recursively generated by augmenting a new edge, $(i,j) \in \mathcal{E}_r$, to the graph representation of $\underline{X}_{r-1}$. For the special case of Gaussian distributions, $\mathbf{\Sigma}_{\underline{X}_r}$ has the following recursive formulation [42]

$$\mathbf{\Sigma}^{-1}_{\underline{X}_r} = \mathbf{\Sigma}^{-1}_{\underline{X}_{r-1}} + \mathbf{\Sigma}^{\dagger}_{i,j} - \mathbf{\Sigma}^{\dagger}_{i} - \mathbf{\Sigma}^{\dagger}_{j}\,, \quad \forall\, 0 \leqslant r \leqslant |\mathcal{E}_{\mathcal{T}}|$$

where $\mathbf{\Sigma}^{\dagger}_{i,j} = [\underline{e}_i \ \underline{e}_j]\mathbf{\Sigma}^{-1}_{i,j}[\underline{e}_i \ \underline{e}_j]^T$ and $\mathbf{\Sigma}^{\dagger}_{i} = \underline{e}_i\mathbf{\Sigma}^{-1}_{i}\underline{e}_i^T$ where $\underline{e}_i$ is a unitary vector with 1 at the i-th place and $\mathbf{\Sigma}_{i,j}$ and $\mathbf{\Sigma}_i$ are the 2-by-2 and 1-by-1 principle sub-matrices of $\mathbf{\Sigma}_{\underline{X}}$, with initial step $\mathbf{\Sigma}_{\underline{X}_0}$=$diag(\mathbf{\Sigma}_{\underline{X}})$ where $diag(\mathbf{\Sigma}_{\underline{X}})$ represents a diagonal matrix with diagonal elements of $\mathbf{\Sigma}_{\underline{X}}$.

Remark: For all $0 \leqslant r \leqslant |\mathcal{E}_{\mathcal{T}}|$, we have

1. $\text{tr}(\mathbf{\Sigma}_{\underline{X}_r}) = \text{tr}(\mathbf{\Sigma}_{\underline{X}})$
2. $\text{tr}(\mathbf{\Sigma}_{\underline{X}}\mathbf{\Sigma}^{-1}_{\underline{X}_r}) = n.$
3. $\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_r}(\underline{x})) = -\frac{1}{2}\log(|\mathbf{\Sigma}_{\underline{X}}\mathbf{\Sigma}^{-1}_{\underline{X}_r}|)$
4. $|\mathbf{\Sigma}_{\underline{X}}| \leqslant \ldots \leqslant |\mathbf{\Sigma}_{\underline{X}_r}| \leqslant \ldots \leqslant |\mathbf{\Sigma}_{\underline{X}_0}| = |diag(\mathbf{\Sigma}_{\underline{X}})|$
5. $H(\underline{X}) \leqslant \ldots \leqslant H(\underline{X}_r) \leqslant \ldots \leqslant H(\underline{X}_0)$

where $H(\underline{X})$ is differential entropy.

Tree approximation models are interesting to study since there are algorithms such as Chow-Liu [3] combined with the Kruskal [33] or the Prim's [34] that efficiently compute the model covariance matrix from the graph covariance matrix.

Remark: The CAM is defined as $\mathbf{\Delta} \triangleq \mathbf{\Sigma}_{\underline{X}}\mathbf{\Sigma}^{-1}_{\underline{X}_{\mathcal{M}}}$. Thus, the constant c can be written as $c = -\frac{1}{2}\log(|\mathbf{\Delta}|)$. Then, for any given covariance matrix and its model covariance matrix that satisfies conditions in theorem 1, the summation of diagonal coefficients of the CAM is equal to n, i.e. the result in theorem 1 implies that tr $(\mathbf{\Delta}) = n$. Using this result and the definition of the KL divergence for jointly Gaussian distributions, we have

$$\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}}}(\underline{x})) = c + \frac{1}{2}tr(\mathbf{\Delta}) - \frac{n}{2}$$

which results in $c = \mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}}}(\underline{x}))$.

2.4.1 Covariance selection example

Here we choose tree approximation model as an example. Figure 2.1 indicates two graphs: (a) the complete graph and (b) its tree approximation model where edges in the graph represent non-zero coefficients in the inverse of the covariance matrix [1].

The correlation coefficient between each pair of adjacent nodes has been written on each edge. The correlation coefficient between each pair of nonadjacent nodes is the multiplication of all correlations on the unique path that connects those nodes. The correlation matrix for each graph is

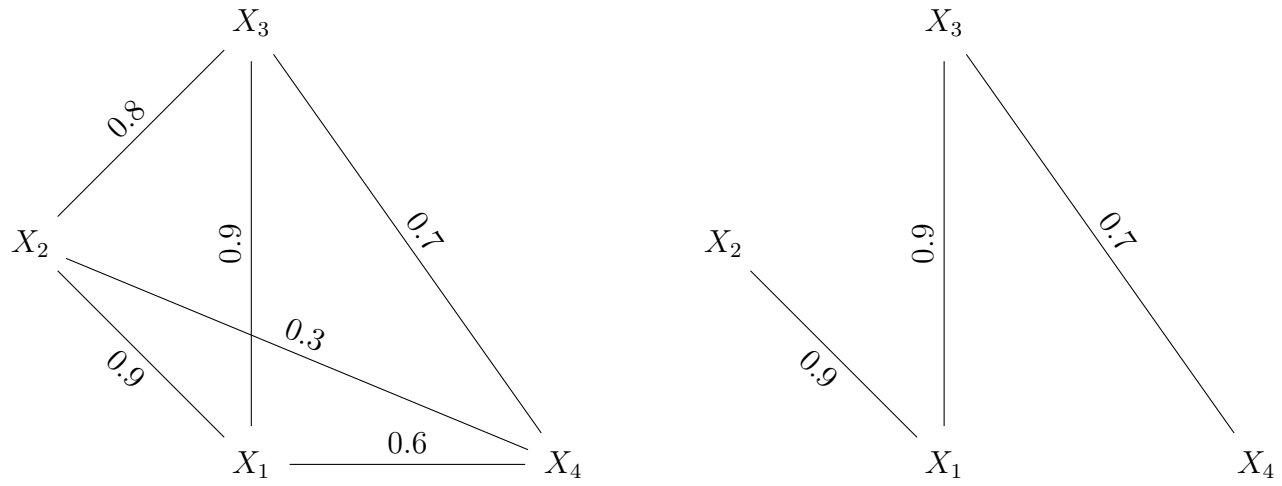

Figure 2.1: (a) The complete graph; (b) The tree approximation of the complete graph.

$$\mathbf{\Sigma}_{\underline{X}} = \begin{bmatrix} 1 & 0.9 & 0.9 & 0.6 \\ 0.9 & 1 & 0.8 & 0.3 \\ 0.9 & 0.8 & 1 & 0.7 \\ 0.6 & 0.3 & 0.7 & 1 \end{bmatrix}$$

and

$$\mathbf{\Sigma}_{\underline{X}_{\mathcal{T}}} = \begin{bmatrix} 1 & 0.9 & 0.9 & 0.63 \\ 0.9 & 1 & 0.81 & 0.567 \\ 0.9 & 0.81 & 1 & 0.7 \\ 0.63 & 0.567 & 0.7 & 1 \end{bmatrix}.$$

The CAM for the above example is

$$\mathbf{\Delta} = \begin{bmatrix} 1 & 0 & 0.0412 & -0.0588 \\ 0.0474 & 1 & 0.3042 & -0.5098 \\ 0.0474 & -0.0526 & 1 & 0 \\ 0.9789 & -1.2632 & 0.1421 & 1 \end{bmatrix}.$$

The CAM contains all information about the tree approximation [2]. Here we assume cases that Gaussian random variables have finite, nonzero variances. The value of the KL divergence for this example is $-0.5\log(|\mathbf{\Delta}|) = 0.6218$.

Remark: Without loss of generality, throughout this dissertation we work with normalized correlation matrices, i.e. the diagonal elements of the correlation matrices are normalized to be equal to one.

2.4.2 Distribution of the LLRT statistic

The random vector $\underline{X}$ has Gaussian distribution under both hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. Thus under both hypotheses, the real random variable, $k(\underline{X}) = \underline{X}^T\mathbf{K}\underline{X}$ has a generalized chi-squared distribution, i.e. the random variable, $k(\underline{X})$, is equal to a weighted sum of chi-squared random variables with both positive and negative weights under both hypotheses. Let us define $\underline{W} = \mathbf{\Sigma}_{\underline{X}}^{-\frac{1}{2}}\underline{X}$ under $\mathcal{H}_0$ and $\underline{Z} = \mathbf{\Sigma}_{\underline{X}_\mathcal{M}}^{-\frac{1}{2}}\underline{X}$ under $\mathcal{H}_1$, where $\mathbf{\Sigma}_{\underline{X}}^{\frac{1}{2}}$ and $\mathbf{\Sigma}_{\underline{X}_\mathcal{M}}^{\frac{1}{2}}$ are the square root of covariance matrices $\mathbf{\Sigma}_{\underline{X}}$ and $\mathbf{\Sigma}_{\underline{X}_\mathcal{M}}$, respectively. Then the random vectors $\underline{W} \sim \mathcal{N}(\underline{0}, \mathbf{I})$ and $\underline{Z} \sim \mathcal{N}(\underline{0}, \mathbf{I})$ are zero-mean Gaussian distributions with the same covariance matrices, $\mathbf{I}$, where $\mathbf{I}$ is the identity matrix of dimension n. Note that, the CAM is a positive definite matrix with $\lambda_i > 0$ where $1 \leqslant i \leqslant n$. Thus, the random variable $k(\underline{X})$, under both hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$ can be written as:

$$K_0 \triangleq k(\underline{X})|\mathcal{H}_0 = \frac{1}{2}\sum_{i=1}^{n}(1-\lambda_i)W_i^2$$

and

$$K_1 \triangleq k(\underline{X})|\mathcal{H}_1 = \frac{1}{2}\sum_{i=1}^{n}(\lambda_i^{-1}-1)Z_i^2$$

2. Dissimilarity parameters α_i's and eigenvalues of CAM contains all information about the tree approximation.

respectively, where random variables W_i and Z_i, are the i-th element of random vectors $\underline{W}$ and $\underline{Z}$, respectively. Moreover, random variables W_i^2 and Z_i^2, follow the first order central chi-squared distribution. Note that, similarly random variable $l(\underline{X}) \triangleq -c + k(\underline{X})$ is defined under each hypothesis as

$$L_0 \triangleq l(\underline{X})|\mathcal{H}_0 = -c + K_0$$

and

$$L_1 \triangleq l(\underline{X})|\mathcal{H}_1 = -c + K_1.$$

Remark: As a simple consequence of the covariance selection theorem, the summation of weights for the generalized chi-squared random variable, the expectation of $k(\underline{X})$, is zero under the hypothesis $\mathcal{H}_0$, i.e. $\mathrm{E}(K_0) = \frac{1}{2}\sum_{i=1}^{n}(1-\lambda_i) = 0$ [1], and this summation is positive under the hypothesis $\mathcal{H}_1$, i.e. $\mathrm{E}(K_1) = \frac{1}{2}\sum_{i=1}^{n}(\lambda_i^{-1} - 1) \geqslant 0$.

2.5 The ROC Curve and the AUC Computation

2.5.1 The receiver operating characteristic curve

The receiver operating characteristic (ROC) curve is the parametric curve where the detection probability is plotted versus the false-alarm probability for all thresholds, i.e. each point on the ROC curve represents a pair of $(P_0(\boldsymbol{\tau}), P_1(\boldsymbol{\tau}))$ for a given threshold $\boldsymbol{\tau}$. Set $z = P_0(\boldsymbol{\tau})$ and $\eta = P_1(\boldsymbol{\tau})$, the ROC curve is $\eta = h(z)$. If $P_0(\boldsymbol{\tau})$ has an inverse function, then the ROC curve is $h(z) = P_1(P_0^{-1}(z))$. In general, the ROC curve, $h(z)$, has the following properties [40]

- $h(z)$ is concave and increasing,
- $h'(z)$ is positive and decreasing,
- $\int_0^1 h'(z)\,dz \leqslant 1$.

Note that, for the ROC curve, the slope of the tangent line at a given threshold, $h'(z)$, gives the likelihood ratio for the value of the test [40].

Remark: For the ROC curve for our Gaussian random vectors we have $h'(z)$ is positive, continuous and decreasing in interval $[0, 1]$ with right continuity at 0 and left continuity at 1. Moreover,

$$\int_0^1 h'(z)\,dz = 1$$

since $h(0) = 0$ and $h(1) = 1$.

Definition 5. *Let $f_{L_0}(l)$ and $f_{L_1}(l)$ be the probability density function of the random variables L_0 and L_1, respectively.* ■

Lemma 1. *Given the ROC curve, $h(z)$, we can compute following KL divergences*

$$\mathcal{D}\left(f_{L_1}(l)||f_{L_0}(l)\right) = -\int_0^1 \log\left(h'(z)\right)\,dz.$$

and

$$\begin{aligned}\mathcal{D}\left(f_{L_0}(l)||f_{L_1}(l)\right) &= -\int_0^1 h'(z)\,\log\left(h'(z)\right)\,dz \\ &\overset{(*)}{=} -\int_0^1 \log\left(\frac{d\,h^{-1}(\eta)}{d\,\eta}\right)\,d\eta\end{aligned}$$

where (*) holds if the ROC curve, $\eta = h(z)$, has an inverse function.

Proof. These results are from the Radon-Nikodým theorem [43]. Simple, alternative calculus based proofs are given in appendix A.1. ■

2.5.2 Area under the curve

As discussed previously, we examine the ROC with a goal that the model approximation results in the ROC being a line of slope 1 passing through the origin. This is in contrast to the conventional detection problem where we want to distinguish between the two hypotheses and ideally, have a ROC that is a unit step function. The area under the curve (AUC) is defined as the integral of the ROC curve (figure 2.2) and is a measure of accuracy in decision problems.

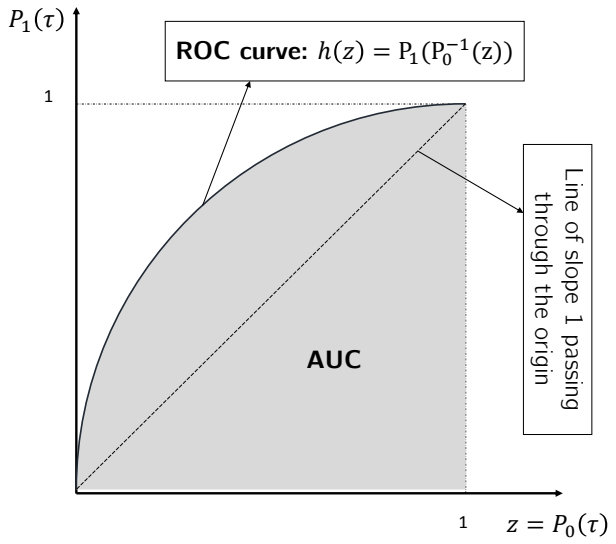

Figure 2.2: The ROC curve and the area under the ROC curve. Each point on the ROC curve indicates a detector with given detection and false-alarm probabilities.

Definition 6. *The area under the ROC cure (AUC) is defined as*

$$AUC = \int_0^1 h(z)\, d\, z = \int_0^1 P_1(\tau)\, dP_0(\tau) \tag{2.3}$$

where τ *is the detection problem threshold.* ■

Remark: The AUC is a measure of accuracy for the detection problem and $1/2 \leqslant \mathrm{AUC} \leqslant 1$. Note that, in conventional decision problems, the AUC is desired to be as close as possible to 1 while in approximation problem presented here we want the AUC to be close to $1/2$.

Theorem 2. Statistical property of AUC [44]. *Assume* L_1 *and* L_0 *are independent random variables then the AUC for the LLRT statistic is*

$$AUC = \Pr\left(L_1 > L_0\right).$$

■

Corollary 1. *From theorem 2, when PDFs for the LLRT statistic under both hypotheses exist, we can compute the AUC as*

$$AUC = \int_0^\infty \left(f_{L_1} \star f_{L_0}\right)(l)\, dl \tag{2.4}$$

where $\left(f_{L_1} \star f_{L_0}\right)(l) \triangleq \int_{-\infty}^{\infty} f_{L_1}(\tau)\, f_{L_0}(l+\tau)\, dl$ *is the cross-correlation between* $f_{L_1}(l)$ *and* $f_{L_0}(l)$.

Proof. A proof based on the definition of the AUC (2.3), is given in [45]. ■

Let us define the difference LLRT statistic random variable as $L_\Delta \triangleq L_1 - L_0$. Then, we get

$$\begin{aligned} AUC &= \Pr\left(L_\Delta > 0\right) \\ &= 1 - F_{L_\Delta}(0) \end{aligned}$$

where $F_{L_\Delta}(l)$ is the cumulative distribution function (CDF) for random variable L_Δ. Note that we define the difference LLRT statistic random variable to simplify the notation and easily show that the AUC only depends on this difference.

The two conditional random variables L_0 and L_1 are independent [3] as stated above. Thus, the cross-correlation between the corresponding two distributions is the distribution of the difference LLRT statistic, L_Δ. We can write the random variable L_Δ as

$$\begin{aligned} L_\Delta &= -c + K_1 - (-c + K_0) \\ &= K_1 - K_0. \end{aligned}$$

Replacing the definition for K_0 and K_1, we have

$$L_\Delta = \frac{1}{2}\sum_{i=1}^{n}(\lambda_i^{-1} - 1)Z_i^2 - \frac{1}{2}\sum_{i=1}^{n}(1 - \lambda_i)W_i^2.$$

We can rewrite the difference LLRT statistic, L_Δ, in an indefinite quadratic form as

$$L_\Delta = \frac{1}{2}\underline{V}^T(\mathbf{\Lambda} - \mathbf{I})\underline{V}$$

where

3. By the definition of the detection problem.

$$\underline{V} = \begin{bmatrix} \underline{W} \\ \\ \underline{Z} \end{bmatrix}$$

and

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & & & & & \\ & \ddots & & & \mathbf{0} & \\ & & \lambda_n & & & \\ & & & \lambda_1^{-1} & & \\ & \mathbf{0} & & & \ddots & \\ & & & & & \lambda_n^{-1} \end{bmatrix}.$$

2.5.3 Analytical expression for AUC

To compute the CDF of random variable L_Δ, we need to evaluate a multi-dimensional integral of jointly Gaussian distributions [46] or we need to approximate this CDF [47]. More efficiently, as discussed in [48] for the real-valued case, the CDF of the random variable L_Δ can be expressed as a single-dimensional integral of a complex function [4] in the following form

$$F_{L_\Delta}(l) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{e^{\frac{l}{2}(j\omega+\beta)}}{j\omega+\beta} \frac{1}{\sqrt{|\mathbf{I} + \frac{1}{2}(\mathbf{\Lambda}-\mathbf{I})(j\omega+\beta)|}}\, d\omega$$

where $\beta > 0$ is chosen such that matrix $\mathbf{I} + \frac{\beta}{2}(\mathbf{\Lambda} - \mathbf{I})$, is positive definite and simplifies the evaluation of the multivariate Gaussian integral [48].

Special case: When $\mathbf{\Lambda} = \mathbf{I}$, i.e. the given covariance obeys the model structure, then

$$AUC = 1 - F_{L_\Delta}(0) = 1 - \frac{1}{2\pi} \int_{-\infty}^{\infty} \frac{1}{j\omega+\beta} \; = \frac{1}{2}$$

for $\beta > 0$ and is also independent of the value of the parameter β.

4. This is the transform to the frequency domain for an arbitrary β.

Picking an appropriate value for the parameter β [5], the AUC can be numerically computed by evaluating the following one dimension complex integral

$$AUC = 1 - \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{1}{j\omega+\beta} \frac{1}{\sqrt{|\mathbf{I} + \frac{1}{2}(\mathbf{\Lambda}-\mathbf{I})(j\omega+\beta)|}}\, d\omega.$$

Furthermore, since $\mathbf{\Lambda} \succ 0$, choosing $\beta = 2$ and changing variable as $\nu = \omega/2$, we have

$$AUC = 1 - \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{1}{j\nu+1} \frac{1}{\sqrt{|\mathbf{\Lambda} + j\nu(\mathbf{\Lambda}-\mathbf{I})|}}\, d\nu. \tag{2.5}$$

Moreover, $|\mathbf{\Lambda}+j\nu(\mathbf{\Lambda}-\mathbf{I})| = \prod_{i=1}^{p}(1+\alpha_i\nu^2 - j\alpha_i\nu)$. This equation shows that the AUC only depends on α_i's.

Remark: Since the AUC integral in (2.5) cannot be evaluated in closed form, it cannot be used directly in obtaining model selection algorithms. Numerical evaluation of the AUC using the one-dimensional complex integral (2.5) is very efficient and fast compared to the numerical evaluation of a multi-dimensional integral of jointly Gaussian CDF.

2.6 Analytical Bounds for the AUC

Section 2.5 derived an analytical expression for the AUC based on zero-mean Gaussian distributions. In this section, we find analytical lower and upper bounds for the AUC. These bounds will give us insight into the behavior of the AUC.

2.6.1 Generalized Asymmetric Laplace distribution

In this subsection, we present the probability density function and moment generating function for the difference LLRT statistic random variable, L_Δ. We will use this result in computing the AUC bound.

The difference LLRT statistic random variable, L_Δ, follows the generalized asymmetric Laplace

5. The parameter β is picked such that $\mathbf{I} + \frac{\beta}{2}(\mathbf{\Lambda}-\mathbf{I}) \succ 0$ and $\beta = 2$ always satisfies this condition since $\mathbf{\Lambda} \succ 0$.

(GAL) distribution [6] [49]. For a given i where $i \in \{1, \ldots, n\}$, we define random variable L_{Δ_i} as

$$L_{\Delta_i} = \frac{\lambda_i - 1}{2} W_i^2 - \frac{1 - \lambda_i^{-1}}{2} Z_i^2. \tag{2.6}$$

Then, difference LLRT statistic random variable, L_Δ, can be written as

$$L_\Delta = \sum_{i=1}^{n} L_{\Delta_i}$$

where L_{Δ_i}'s are independent and have GAL distributions at position 0 with mean $\alpha_i/2$ and PDF [49]

$$f_{L_{\Delta_i}}(l) = \frac{e^{\frac{l}{2}}}{\pi\sqrt{\alpha_i}} K_0\left(\sqrt{\alpha_i^{-1} + \frac{1}{4}}\,|l|\right), \qquad l \neq 0 \tag{2.7}$$

where $K_0(-)$ is the modified Bessel function of second kind [50]. The moment generating function (MGF) for this distribution is

$$M_{L_{\Delta_i}}(t) = \frac{1}{\sqrt{1 - \alpha_i t - \alpha_i t^2}}$$

for all t's that satisfies $1 - \alpha_i t - \alpha_i t^2 > 0$. From (2.6), the MGF derivation for the GAL distribution is straightforward and is the multiplication of two MGFs for the chi-squared distribution.

The distribution of the difference LLRT statistic random variable, L_Δ, is

$$f_{L_\Delta}(l) = \underset{i=1}{\overset{n}{\circledast}} f_{L_{\Delta_i}}(l)$$

where $\circledast_{i=1}^{n}$ is the notation we use for convolution of n functions together. Note that, although the distribution of random variables L_{Δ_i}'s in (2.7) has discontinuity at $l = 0$, the distribution of random variable L_Δ is continuous if there are at least two distribution with non-zero parameters, α_i's, in the aforementioned convolution. Moreover, the MGF for $f_{L_\Delta}(l)$ can be computed by multiplying MGFs for L_{Δ_i} as

$$M_{L_\Delta}(t) = \prod_{i=1}^{n} M_{L_{\Delta_i}}(t) \tag{2.8}$$

for all t's in the intersection of all domains of $M_{L_{\Delta_i}}(t)$. The smallest of such intersections is $-1 < t < 0$.

6. Also known as the variance-gamma distribution or the Bessel function distribution.

2.6.2 Lower bound for the AUC (Chernoff bound application)

Given the MGF for the difference LLRT statistic distribution (2.8), we can apply the Chernoff bound [51] to find a lower bound for the AUC or upper bound for the CDF of the difference LLRT statistic random variable, L_Δ, evaluated at zero).

Proposition 2. *Lower bound for the AUC is*

$$\Pr\left(L_\Delta > 0\right) \geqslant \max\left\{\frac{1}{2}, 1 - e^{-\frac{1}{2}\sum_{i=1}^{n}\log\left(1+\frac{\alpha_i}{4}\right)}\right\} \tag{2.9}$$

Proof. One-half is a trivial lower bound for AUC. To achieve a non-trivial lower bound, we apply Chernoff bound [51] as follow

$$\Pr\left(L_\Delta < 0\right) \leqslant \inf_t \; M_{L_\Delta}(t).$$

To complete the proof we need to solve the right-hand-side (RHS) optimization problem.

Step 1: First derivatives of $M_{L_\Delta}(t)$ is

$$\begin{aligned}\frac{d}{d\,t}M_{L_\Delta}(t) &= M_{L_\Delta}(t)\left(\frac{1}{2}\sum_{i=1}^{n}\frac{\lambda_i - 1}{1-(\lambda_i-1)t} + \frac{\lambda_i^{-1}-1}{1-(\lambda_i^{-1}-1)t}\right)\\ &= M_{L_\Delta}(t)(1+2t)\sum_{i=1}^{n}\frac{\alpha_i}{2(1-\alpha_i t-\alpha_i t^2)}.\end{aligned}$$

Clearly, first derivative is zero for $t = -1/2$ which is in the feasible domain of the MGF for the difference LLRT statistic. Note that, the smallest feasible domain is $-1 < t < 0$.

Step 2: Second derivatives of $M_{L_\Delta}(t)$ is

$$\begin{aligned}\frac{d^2}{d\,t^2}M_{L_\Delta}(t) = M_{L_\Delta}(t)&\left(\frac{1}{4}\sum_{i=1}^{n}\frac{\lambda_i - 1}{1-(\lambda_i-1)t} + \frac{\lambda_i^{-1}-1}{1-(\lambda_i^{-1}-1)t}\right)^2\\ &+ M_{L_\Delta}(t)\left(\frac{1}{4}\sum_{i=1}^{n}\frac{(\lambda_i - 1)^2}{(1-(\lambda_i-1)t)^2} + \frac{(\lambda_i^{-1}-1)^2}{(1-(\lambda_i^{-1}-1)t)^2}\right).\end{aligned}$$

Therefore, we conclude that the second derivative is positive and thus the optimal solution to the RHS optimization problem is at $t = -\frac{1}{2}$. Replacing that in the definition of the moment generation

function which results in the following bound

$$\Pr\left(L_\Delta \leqslant 0\right) < \prod_{i=1}^{n} \frac{2}{\sqrt{4+\alpha_i}}$$

which can be written as

$$\Pr\left(L_\Delta > 0\right) \geqslant 1 - \prod_{i=1}^{n} \frac{2}{\sqrt{4+\alpha_i}}$$

which completes the proof. ■

2.6.3 Upper Bound for the AUC

In this section, we present a parametric upper bound for the AUC, but first, we need to present the following results.

Lemma 2. Data processing inequality of the KL divergence for the LLRT statistic. *We have*

$$\mathcal{D}(f_{L_1}(l)||f_{L_0}(l)) \leqslant \mathcal{D}(f_{\underline{X}}(\underline{x}|\mathcal{H}_1)||f_{\underline{X}}(\underline{x}|\mathcal{H}_0))$$

and

$$\mathcal{D}(f_{L_0}(l)||f_{L_1}(l)) \leqslant \mathcal{D}(f_{\underline{X}}(\underline{x}|\mathcal{H}_0)||f_{\underline{X}}(\underline{x}|\mathcal{H}_1)).$$

Proof. This lemma is an special case of the data processing property for the KL divergence [52]. By picking the appropriate measurable mapping which in this case is a quadratic function (for our case of Gaussian random vectors for the LLRT) the Lemma is proved. ■

Definition 7. Possible Feasible Region. *The AUC and the KL divergence pair lie in the possible feasible region (figure 2.3) for all possible detectors (ROC curves), i.e. no detector with the AUC and the KL divergence pair lie outside the feasible region*[7].

Theorem 3. Possible feasible region for the AUC and the KL divergence. *Given the ROC curve, the parametric possible feasible region as shown in figure 2.3 can be expressed using the positive parameter $a > 0$ as*

$$\Pr\left(L_\Delta > 0\right) = \frac{1}{1-e^{-a}} - \frac{1}{a}$$

7. The definition of the feasible region here is inspired by the joint range of f-divergences [53].

and

$$\mathcal{D}_l^* \geqslant \log(a) + \frac{a}{e^a - 1} - 1 - \log(1 - e^{-a})$$

where

$$\mathcal{D}_l^* = \min\left\{\mathcal{D}(f_{L_1}(l)||f_{L_0}(l))\,,\, \mathcal{D}(f_{L_0}(l)||f_{L_1}(l))\right\}. \tag{2.10}$$

Proof. Proof is given in the appendix A.2. ■

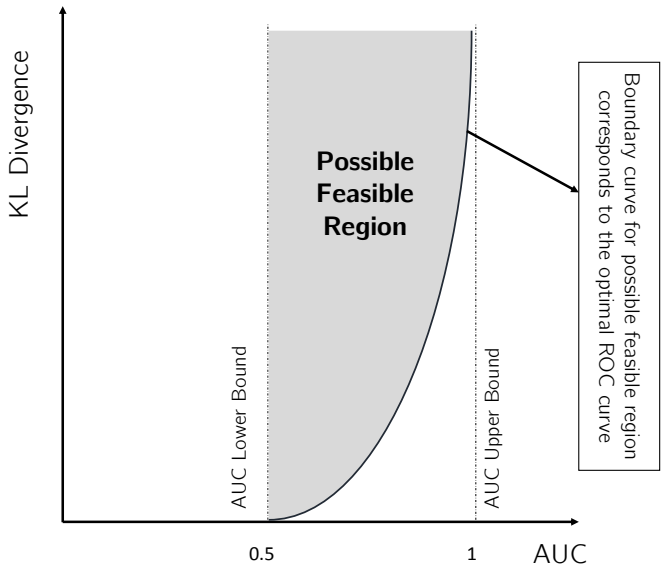

Figure 2.3: Possible feasible region for the AUC and the KL divergence pair for all possible detectors or equivalently all possible ROC curves (the KL divergence is between the LLRT statistics under different hypotheses, i.e. $\mathcal{D}(f_{L_0}(l)||f_{L_1}(l))$ or $\mathcal{D}(f_{L_1}(l)||f_{L_0}(l))$.)

Theorem 3 formulates the relationship between the AUC and the KL divergence. *The results of this theorem is generally true for any LLRT statistic.* Theorem 3 states that for any valid ROC that corresponds to a detection problem, the pair of AUC and KL divergence *must* lie in the possible feasible region (figure 2.3), i.e. outside of this region is infeasible. This possible feasible region results in the general upper bound for AUC.

Since computing the distribution of the LLRT statistics is not straightforward in most cases, proposition 3, relaxes the Theorem 3 by bounding the KL divergence between the LLRT statistics using the invariance property of KL divergence for the LLRT statistic (lemma 2).

Proposition 3. *The parametric upper bound for AUC is*

$$\Pr\left(L_\Delta > 0\right) = \frac{1}{1-e^{-a}} - \frac{1}{a}$$

and

$$\mathcal{D}^* \geqslant \log(a) + \frac{a}{e^a - 1} - 1 - \log(1 - e^{-a})$$

where $a > 0$ is a positive parameter and

$$\mathcal{D}^* = \min\left\{ \mathcal{D}(f_{\underline{X}}(\underline{x}|\mathcal{H}_1)||f_{\underline{X}}(\underline{x}|\mathcal{H}_0)) \,,\, \mathcal{D}(f_{\underline{X}}(\underline{x}|\mathcal{H}_0)||f_{\underline{X}}(\underline{x}|\mathcal{H}_1)) \right\}. \tag{2.11}$$

Proof. Proof is based on the lemma 2 and the possible feasible region presented in the theorem 3. From the lemma 2, we have

$$\mathcal{D}_l^* \leqslant \mathcal{D}^*.$$

Then, using the result in the theorem 3, we get the parametric upper bound. ■

2.6.4 Asymptotic behavior for AUC bounds

Proposition 4. Asymptotic behavior of the lower bound. *We have*

$$\Pr\left(L_\Delta > 0\right) \geqslant 1 - e^{-n\left(1-\frac{1}{n}\sum_{i=1}^{n}(1+\frac{\alpha_i}{8})^{-1}\right)}.$$

Proof. Applying the inequality

$$\frac{2x}{2+x} < \log(1+x)$$

for $x > 0$, we achieve the result. ■

Proposition 5. Asymptotic behavior of the upper bound. *The parametric upper bound for AUC has the following asymptotic behavior*

$$\Pr\left(L_\Delta > 0\right) \leqslant 1 - e^{-\mathcal{D}^* - 1}$$

where $\mathcal{D}^*$ is given in (2.11).

Proof. Proof is as follows.

$$\begin{aligned} -\log\left(1-\Pr\left(L_\Delta>0\right)\right) &= -\log\left(\frac{1}{e^a-1}+\frac{1}{a}\right) \\ &\leqslant \log\left(a\right) \\ &\leqslant \mathcal{D}^*+1. \end{aligned}$$

Applying the exponential function to both sides of the above inequality we get the upper bound. ■

Remark: The asymptotic lower bound is a function of the number of nodes, n and has an exponential decaying behavior. The asymptotic upper bound also has an exponential decaying behavior with respect to KL divergence.

Figure 2.4 shows the possible feasible region and the asymptotic behavior log-scale. As it is shown in this figure, the parametric upper bound can be approximated with a straight line especially for large values of the parameter a (the result in proposition 5). Also, figure 2.5 shows the possible feasible region and the asymptotic behavior in regular-scale.

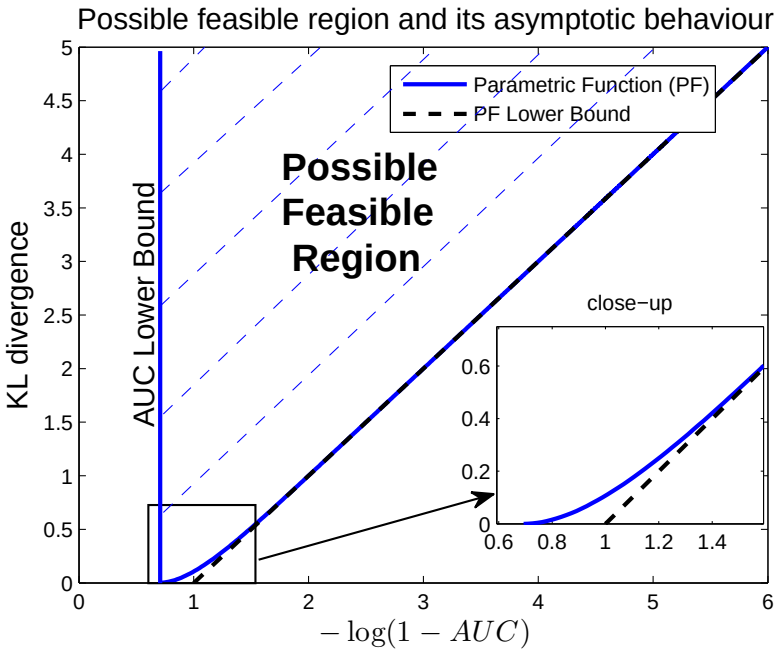

Figure 2.4: Log-scale of the possible feasible region and its asymptotic behavior (linear line) for the AUC and the KL divergence pair for all possible detectors or equivalently all possible ROC curves (the KL divergence is between the LLRT statistics under different hypotheses, i.e. $\mathcal{D}(f_{L_1}(l)||f_{L_0}(l))$ or $\mathcal{D}(f_{L_0}(l)||f_{L_1}(l))$.) Close-up part shows the non-linear behavior of the possible feasible region around one.

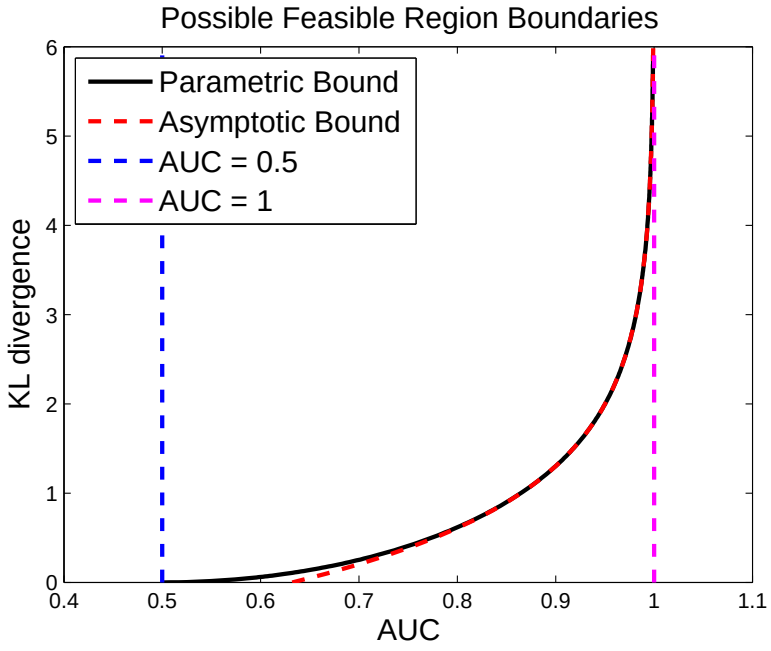

Figure 2.5: The possible feasible region boundaries and its asymptotic behavior for the AUC and the KL divergence pair for all possible detectors or equivalently all possible ROC curves (the KL divergence is between the LLRT statistics under different hypotheses, i.e. $\mathcal{D}(f_{L_0}(l)||f_{L_1}(l))$ or $\mathcal{D}(f_{L_1}(l)||f_{L_0}(l))$.).

2.7 Conclusion

In this chapter, we formulated a detection problem and investigated the quality of the model selection. More specifically, we considered Gaussian distributions and discuss the covariance selection quality of a given model. We present the correlation approximation matrix (CAM) and show its relationship with information theory divergences such as the KL divergence, the reverse KL divergence, and the Jeffreys divergence as well as the ROC curve and the area under the ROC curve, the AUC, as a measure of accuracy in the detection problem framework. This chapter also presents an analytical expression for the AUC that can be efficiently evaluated numerically. AUC analytical lower and upper bounds are also provided. We show that the AUC and the lower bound for the AUC depend on the eigenvalues of the CAM. Upper bounds for the AUC are obtained from finding a parametric relationship between the AUC and the KL/reverse KL divergences.

The detection framework presented in this chapter can be generalized for non-Gaussian models. The AUC analytical bounds obtained in this chapter can also be used in other applications that are using AUC as a relevant criterion. One example is in medicine when the AUC is used for diagnostic tests between positive instance and negative instance [54] where instead of changing the coordinates we can look at the exponent of the AUC bounds.

3

The Covariance Selection Problem using a Detection Problem Formulation: Examples

In this chapter, we consider some examples of covariance matrices for a Gaussian random vector $\underline{X}$. In the first part, we look at some covariance matrices as examples and approximate those with tree structured graphical models. In particular, we look at a Toeplitz covariance matrix as well as real solar data covariance matrices and a simulated sensor network example where sensors are place on a two dimensional grid. In the second part, we look at approximations beyond tree structure, specially graphs with Junction tree structures. In this part, we focus on Toeplitz covariance matrix example and show some theoretical results alongside simulation results on how good these models perform. Our ultimate goal in this chapter is to determine the quality of each of the above approximation scenarios.

3.1 Part I: Tree approximation model

Tree approximation models are interesting to study since there are algorithms such as Chow-Liu [3] combined with the Kruskal [33] or the Prim's [34] that efficiently compute the model covariance matrix from the graph covariance matrix.

3.1.1 Toeplitz example

Here, we assume that the covariance matrix $\mathbf{\Sigma}_{\underline{X}}$ has a Toeplitz structure with ones on the diagonal elements and the correlation coefficient $\rho > -\frac{1}{(n-1)}$ as off diagonal elements

$$\mathbf{\Sigma}_{\underline{X}} = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \dots & \rho & 1 \end{bmatrix}.$$

In this subsection, we consider a tree structured model as the graphical model approximation and examine the covariance selection problem quality. In our simulations, we compare the numerically evaluated AUC and its lower and upper bounds and discuss their asymptotic behavior as the dimension of the graphical model, n, increases.

3.1.1.1 Star approximation

$$\mathbf{\Sigma}^{star}_{\underline{X}_{\mathcal{T}}} = \begin{bmatrix} 1 & \rho & \dots & \dots & \rho \\ \rho & \ddots & \rho^2 & \dots & \rho^2 \\ \vdots & \rho^2 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \rho^2 \\ \rho & \rho^2 & \dots & \rho^2 & 1 \end{bmatrix}.$$

For this example, the KL divergence and the Jeffreys divergence can be computed in closed form as

$$\mathcal{D}(\underline{X}||\underline{X}_{star}) = \frac{1}{2}(n-1)\log(1+\rho) - \frac{1}{2}\log(1+(n-1)\rho)$$

and

$$\mathcal{D}_{\mathcal{J}}(\underline{X}\,,\,\underline{X}_{star}) = \frac{(n-1)(n-2)\rho^2}{2(1+(n-1)\rho)}$$

respectively, where

$$\mathcal{D}_{\mathcal{J}}(\underline{X}\,,\,\underline{X}_{star}) = \mathcal{D}(\underline{X}||\underline{X}_{star}) + \mathcal{D}(\underline{X}_{star}||\underline{X})$$

is the Jeffreys divergence [21]. Moreover, for large values of n we can approximate the KL divergence as

$$\mathcal{D}(\underline{X}||\underline{X}_{star}) \approx \frac{n}{2}\log(1+\rho)$$

which is linear in number of vertices in the graph. Similarly, we can also approximate the Jeffreys divergence for large values of n as follow

$$\mathcal{D}_{\mathcal{J}}(\underline{X}\,,\,\underline{X}_{star}) \approx \frac{n}{2}\rho.$$

Figure 3.1 plots the $1-$AUC v.s. the dimension of the graph, n for different correlation coefficients, $\rho = 0.1$ and $\rho = 0.9$. This figure also indicates the upper bound and the lower bound for the $1-$AUC.

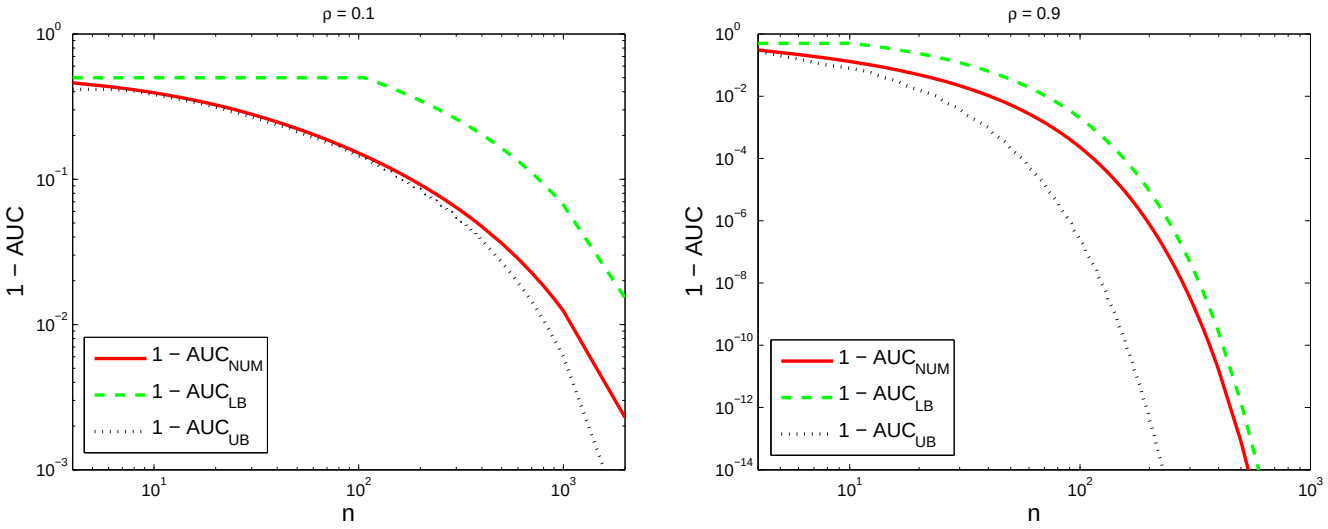

Figure 3.1: 1−AUC v.s. the dimension of the graph, n for Star approximation of the Toeplitz example with $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)**. In both figures, the numerically evaluated AUC is compared with its bounds.

3.1.1.2 Chain approximation

The chain approximated covariance matrix is as follow (nodes are connected like a first order Markov chain, 1 to n)

$$\mathbf{\Sigma}_{\underline{X}_{\mathcal{T}}}^{chain} = \begin{bmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ \rho & \ddots & \ddots & \ddots & \vdots \\ \rho^2 & \ddots & \ddots & \ddots & \rho^2 \\ \vdots & \ddots & \ddots & \ddots & \rho \\ \rho^{n-1} & \dots & \rho^2 & \rho & 1 \end{bmatrix}.$$

For this example, the KL divergence and the Jeffreys divergence can be computed in closed form as

$$\mathcal{D}(\underline{X}||\underline{X}_{chain}) = \mathcal{D}(\underline{X}||\underline{X}_{star})$$

and

$$\mathcal{D}_{\mathcal{J}}(\underline{X}\,,\,\underline{X}_{chain}) = \frac{\rho^2}{(1+(n-1)\rho)(1-\rho)} \times$$

$$\left(\frac{n(n-1)}{2} - \frac{n(1-\rho^n)}{1-\rho} + \frac{1-(n+1)\rho^n + n\rho^{n+1}}{(1-\rho)^2} \right)$$

respectively. Moreover, for large values of n we have the following approximation for the Jeffreys divergence

$$\mathcal{D}_{\mathcal{J}}(\underline{X}\,,\,\underline{X}_{chain}) \approx \frac{n}{2}\frac{\rho}{1-\rho}.$$

Remark: As we see in all of the above approximations for the KL divergences and the Jeffreys divergences for both star and chain models, for large values of n all divergences are linear in the number of graph vertices, n. Table 3.1 shows the approximated slope for large n for all of these approximations.

	$\mathcal{D}_{\mathcal{J}}(\underline{X}\,,\,\underline{X}_{\mathcal{M}})$		$\mathcal{D}(\underline{X}\|\|\underline{X}_{\mathcal{M}})$	
	Star Model	Chain Model	Star Model	Chain Model
Approx. Slope	$\frac{\rho}{2}$	$\frac{\rho}{2(1-\rho)}$	$\frac{\log(1+\rho)}{2}$	$\frac{\log(1+\rho)}{2}$

Table 3.1: Approximated slope for large n of the KL divergences and the Jeffreys divergences for both chain and star models

Figure 3.2 plots the $1-$AUC v.s. the dimension of the graph, n for different correlation coefficients, $\rho = 0.1$ and $\rho = 0.9$ as well as its upper and lower bounds.

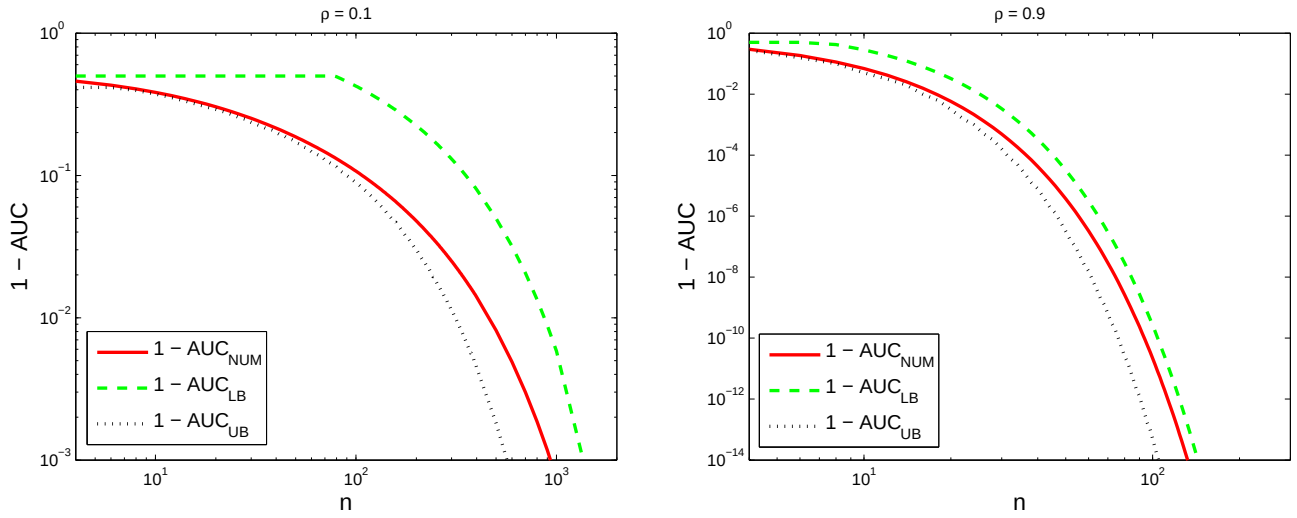

Figure 3.2: 1−AUC v.s. the dimension of the graph, n for Chain approximation of the Toeplitz example with $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)**. In both figures, the numerically evaluated AUC is compared with its bounds.

In both figure 3.1 and figure 3.2, $(1 - \text{AUC})$ and its bounds rapidly goes to 0 which means that AUC goes to one as we increase the number of nodes, n, in the graph. More precisely, bounds for $1 - \text{AUC}$ are decaying exponentially as the dimension of the graph, n, increases which is consistent with the theory obtained for analytical bounds. Furthermore, we can conclude from these figures that a smaller ρ results in a better tree approximation, i.e. covariance matrices with smaller correlation coefficients are more like tree structure model. Moreover, comparing the AUC for the star network approximation with the AUC for the chain network approximation we conclude that the star network is a much better approximation than the chain network even though that both approximation networks have the same KL divergences. We can also interpret this fact through the analytical bounds obtained in this paper. The star network is a better approximation than the chain network since the decay rate of $1 - \text{AUC}$ for the star network is less than its decay rate for the chain network.

Remark: The star approximation in the above example has lower AUC than the chain approximation. Practically, it means that the correlation coefficients between vertices that are not connected to each other in the approximated graphical structure has been approximated more accurately in the star network than in the chain network.

3.1.1.3 Divergences values on possible feasible region

Here we look at the position of divergences and AUCs pairs on the possible feasible region. We compute the KL divergences and reverse KL divergences for jointly Gaussian random vectors (2.11) and LLRT statistics random variables (2.10).

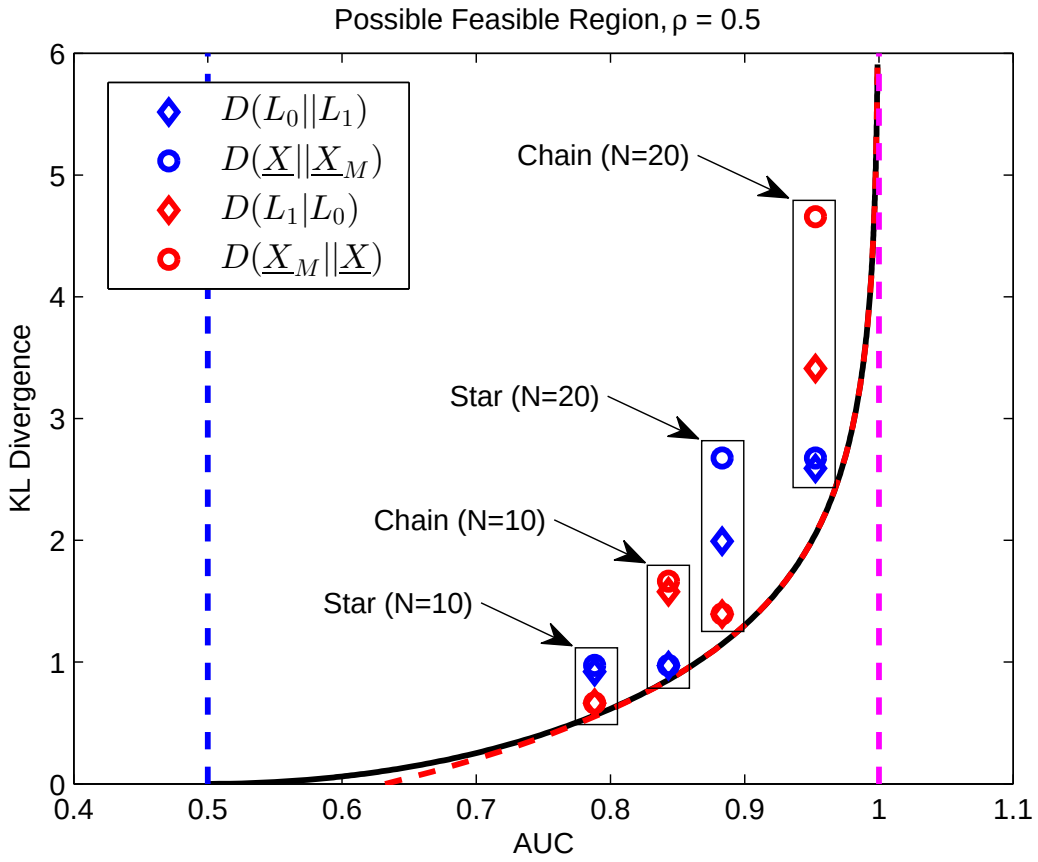

Figure 3.3: Possible feasible region for KL divergence and AUC for Toeplitz covariance matrices with both $n = 10$ and $n = 20$ and correlation $\rho = 0.5$ which shows values of KL divergence, reverse KL divergence and AUC for both star approximation and chain approximation. (KL shows the KL divergence between jointly Gaussian random vectors while KL_l show the KL divergence between LLRT statistic random variables.)

Figure 3.3 plots the possible feasible region for KL divergence and AUC. This figure illustrates divergences and AUCs pairs as points for Toeplitz covariance matrices with both $n = 10$ vertices and $n = 20$ vertices and correlation coefficient $\rho = 0.5$. This figure also compares the goodness of star approximation and chain approximation.

3.1.1.4 LLRT statistic probability density function

Figure 3.4 shows the PDFs for LLRT statistic random variable under both hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$. In this figure, we investigate the LLRT statistic random variable PDFs for Toeplitz covariance matrices with $n = 10$ vertices and $n = 20$ vertices and correlation coefficient $\rho = 0.5$. This figure also shows means of each of the plotted PDFs which are the KL divergence and the reverse KL divergence.

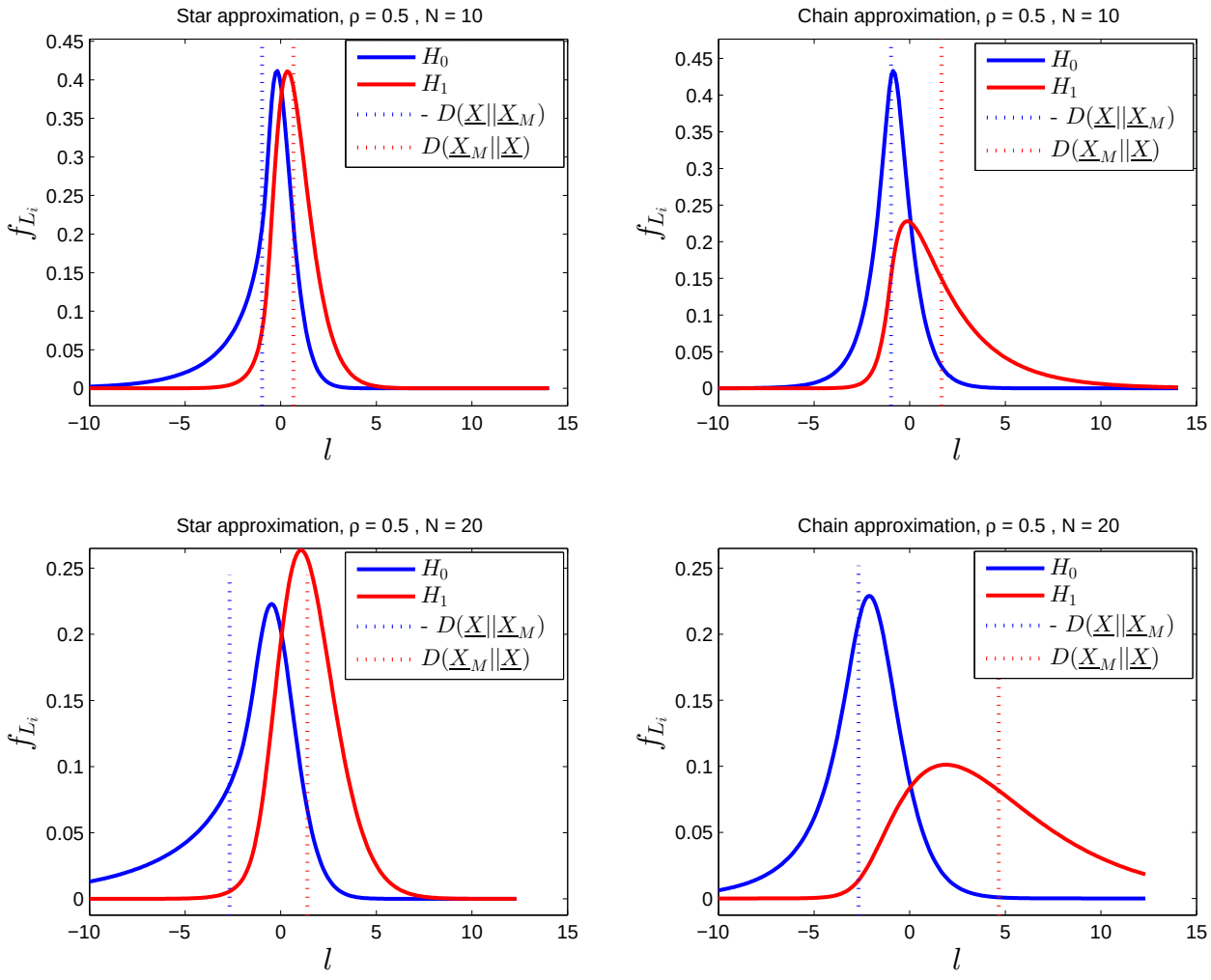

Figure 3.4: Probability distribution functions for the LLRT statistic random variable under both hypotheses $\mathcal{H}_0$ and $\mathcal{H}_1$, for Toeplitz covariance matrices with $n = 10$ vertices and $n = 20$ vertiecs and correlation coefficient $\rho = 0.5$.

Remark: The conditional pdfs of LLRT gives us complete information about the quality of the approximation model for the covariance selection problem. For a good approximation model we want these two pdfs to be as close as possible. If we look at the means of these to random variables we get the KL and reverse KL divergence. If they are both close to zero, then the model should provide a close approximation. However, this may not always be the case as two pdfs could still be different and the AUC will reflect this.

3.1.2 Real solar data example

In this subsection, we look at real solar irradiation data obtain from NREL website [55]. As discussed previously in the introduction chapter 1, part of our motivation for this research is based on looking at distributed state estimation for microgrids with penetration of distributed renewable energy sources such as roof-top solar Photo Voltaics (PV). Solar radiation data at these energy sources are highly correlated (due to received irradiation) and we are looking to approximate the distribution of these data with simpler, yet informative approximations that can also be represented by tree structures. In what follows, we will see that when the number of nodes are moderately large (19) (first example), tree approximations do not work well. However, when we have a small number of sources (6) (second example), then with the proper edge inclusion tree approximations work well.

3.1.2.1 Normalization methods

Standard normalization method: In the standard normalization method a time interval in a day and the required days of the data is selected and then by subtracting mean and dividing by deviation we normalize the data.

Zenith angle normalization method: The Zenith angle is the angle between the perpendicular line to the earth and the line to the sun where at the sunrise and the sunset it is 90 degrees. Relationship of the cosine of the Zenith angle to solar irradiation is linear in sunny days. In order to get rid of the time of day and the seasonal effects over the observed data, one can divide the received irradiation at each time with the cosine of the Zenith angle at that time. Then by using the standard normalization method, by subtracting mean of data and then divide it by its deviation, we get the normalized data [56].

3.1.2.2 Solar measurement fields definition [2]

1. **Oahu solar measurement grid sites:** We examined NREL solar data for Oahu solar measurement grid sites by looking at sensor data taking from Kaleloa, Hawaii [55]. The data consists of one second sampled data from 19 sensors where 17 sensors are at a horizontal position and 2 sensors are tilted 45 degrees toward the west. Figure 3.5 compare the data received at one of the tilted sites and the horizontal site nearby. DH1 Sensors with indexes 8 and 9 are at the same place and the 8-th sensor is in a horizontal position while the 9-th sensor is tilted 45 degrees toward the west. The same thing holds for AP6 sensor indexes 10 and 11 where sensors are at the same place and the 10-th sensor is in a horizontal position while the 11-th sensor is tilted 45 degrees toward the west [1]. We looked at different combinations of these sensors and extracted data for a year from April first, 2010 to March 31-th 2011. The data was segmented to times between 9 : 00 AM to 5 : 00 PM. We then normalized data using the standard normalization method and the zenith angle normalization with time intervals of 1 minute, 5 minutes and 10 minutes as described before. After that, we computed the unbiased estimate of the correlation matrix.

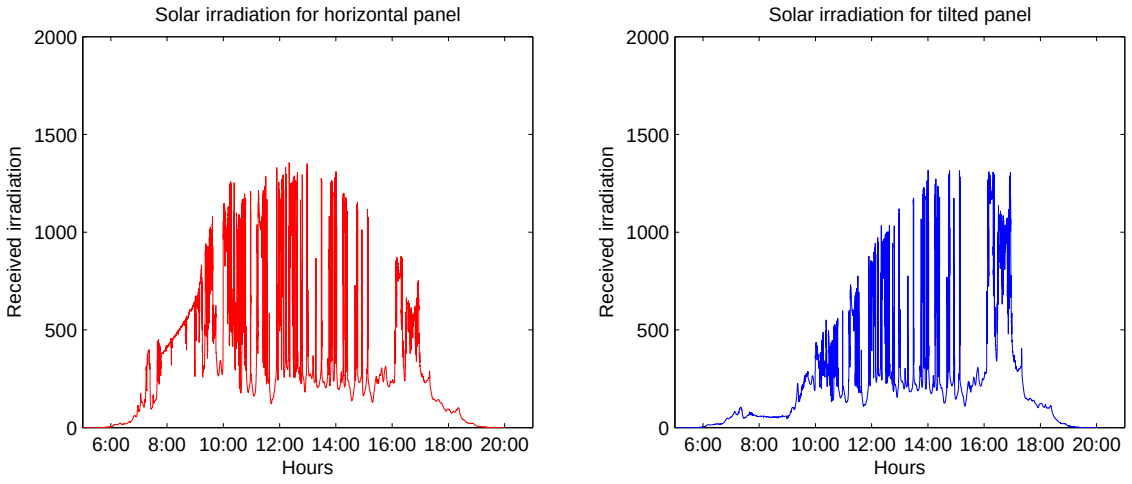

Figure 3.5: **Left:** Solar received irradiation for a panel with horizontal angle. **Right:** Solar received irradiation at the same position for a panel with angle 45 degrees tilted toward the west.

1. look at the field definition [55].

2. **Colorado sites:** We examined NREL solar data for 6 sites near the city of Denver, Colorado [55]. The sites are National Wind Technology Center (NWTC), Solar Technology Acceleration Center (STAC), Lowry Range Solar Station (LRSS), Solar Radiation Research Laboratory (SRRL), Vehicle Testing and Integration Facility (VTIF) and South Park Mountain Data (SPMD). SRRL and VTIF are fairly close to each other and the distance between them is around 400 meters while the distance between any other pair of sites is between 22Km and 92Km. The data of all sites consists of one minute sampled data except data of SPMD site which is sampled every 5 minutes. To make the data usable, we repeat the sampled data of this sensor four more times to obtain one minute sampled data. We extracted data of the year 2013. The data was segmented to times between 8 : 00 AM to 4 : 00 PM. We then normalized data using the standard normalization method and the zenith angle normalization with time intervals of 1 minute, 5 minutes and 10 minutes as described before. After that, we computed the unbiased estimate of the correlation matrix.

In what follows, we investigate the KL divergence between the empirical covariance matrices that are computed for the dataset presented previously and the optimal, Chow-Liu tree structured approximation of them. Our goal is to understand how good these tree structured approximations, model the spatial correlations between different sites in both Kaleloa, Oahu and Denver, Colorado.

As we mentioned it before, here we use the KL divergence to quantify the goodness of the tree approximation. We show the optimal value for the KL divergence using the notation $\mathcal{D}^{\circ}$ at each time of day.

Figure 3.6 is plotted for the 17 Oahu horizontal sensors (left) and the 6 Colorado sensors (right). KL divergences in this figure are plotted for three time intervals of 1 minute, 5 minutes and 10 minutes during the day while both the standard normalization and the zenith angle normalization are used. We can see that the standard normalization (subtract data by mean and then divide by deviation for that time of day) is slightly worse than the zenith angle normalization for the Colorado sensors. Note that the KL divergence for the Colorado sites is much smaller than it is for the Oahu sites. The reason is that the Oahu graph is much bigger than the Colorado sites.

Figure 3.7 discusses the seasonal effect on the graphical structure of the spatial correlations of the solar PV cells. In this figure, left plot is the Kaleloa sites and right plot is the Colorado sites.

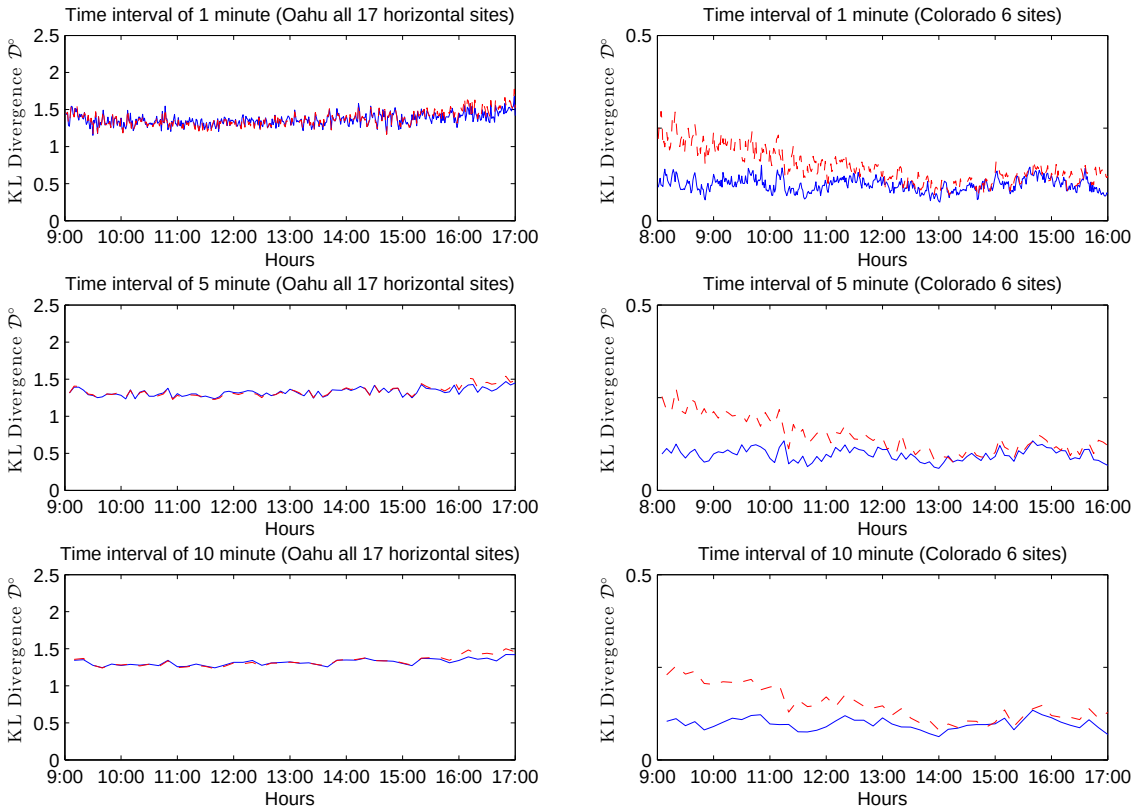

Figure 3.6: The minimum KL divergence distance comparison between all the 17 horizontal Oahu measurement grid and the 6 Colorado sites for windowing time interval 1 minute, 5 minutes and 10 minutes (Solid lines show the Zenith angle normalization while dashed lines indicate the standard normalization method.)

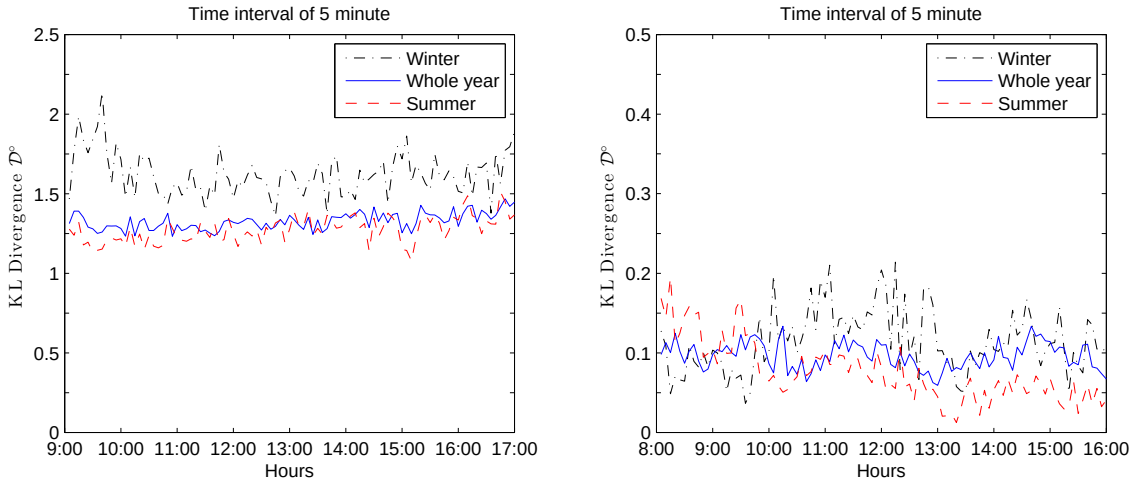

Figure 3.7: The minimum KL divergence distance comparison between seasonal data (average over summer, winter and whole year) for all the 17 horizontal Oahu measurement grid (left) and the 6 Colorado sites (right) with windowing time interval of 5 minutes

We used the Zenith angle normalization. It shows that for both sites (Oahu and Colorado) tree structure is a good model while it works slightly better in summer than winter.

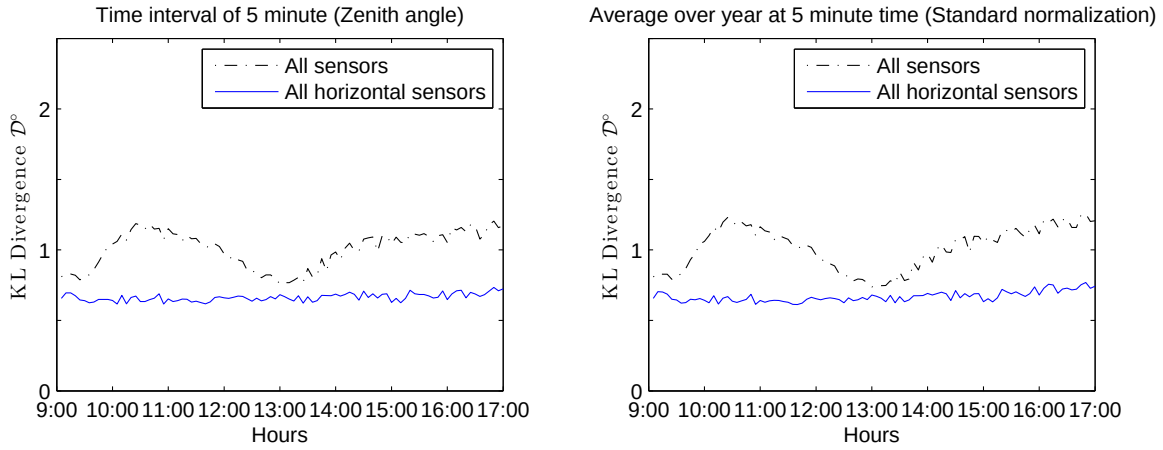

Figure 3.8: The minimum KL divergence for different times of a day by taking into account all the sensors (tilted and un-tilted) for solar irradiation data form Oahu sites.

Figure 3.8 takes into account tilted solar cells and discusses the effect on the graphical structure of the spatial correlations of the distributed PV solar sites. Those two tilted sensors are highly correlated to each others since their received irradiation pattern are similar. They also are highly correlated to the nearest geographical sensor during the morning and the afternoon. These result in the spatial correlations of all the sensors being highly correlated which makes the tree modeling slightly worse in the morning and in the afternoon than at noon.

Figure 3.9 shows the seasonal and the time of day effects for the Oahu measurement grid when we are taking all the 19 sensors into account. Tilted sensors add more strong edges to the spatial connectivity of Oahu graph representation in the morning and in the afternoon which causes slightly higher KL divergence.

Figure 3.10 depicts the normalized minimum KL divergence distance per removed edge (i.e. $\mathcal{D}_e^\circ = {}^{\mathcal{D}^\circ}\!/\!_{((p-1)(p-2)/2)}$) for all the 6 Colorado sensors, the first 6 and the last 6 Oahu sensors and the all 19 Oahu sensors. The first 6 Oahu sensors are DH3, DH4, DH5, DH10, DH11 and DH9 with indexes 201-206 while the last 6 Oahu sensors are AP5, AP4, AP7, DH6, DH7 and DH8 with indexes 212-217 [2]. We take the first 6 and the last 6 Oahu sensors to compare results together and with the result of the 6 Colorado sensors. We can conclude from this figure while tree modeling is

2. look at the field definition [55].

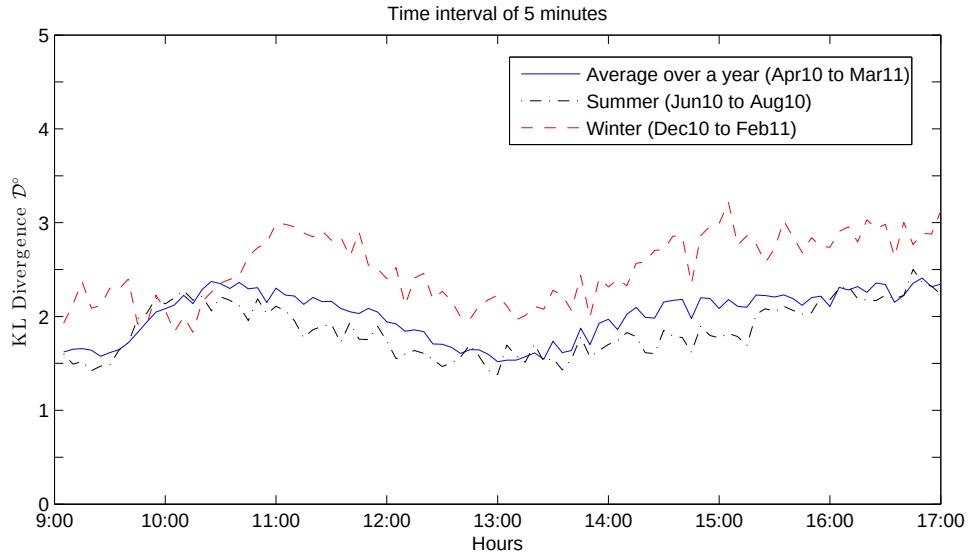

Figure 3.9: The minimum KL divergence distance by taking into account all the sensors for Oahu sites (average over summer, winter and whole year)

doing well for all scenarios, it is doing slightly better for the Oahu Sensors. Moreover, by comparing the results of this figure for the first 6, the last 6 and all the Oahu sensors , we conclude that the normalized KL divergence metric, $\mathcal{D}_e^\circ$, is almost the same for the Oahu sensors and that is less than the Colorado sensors.

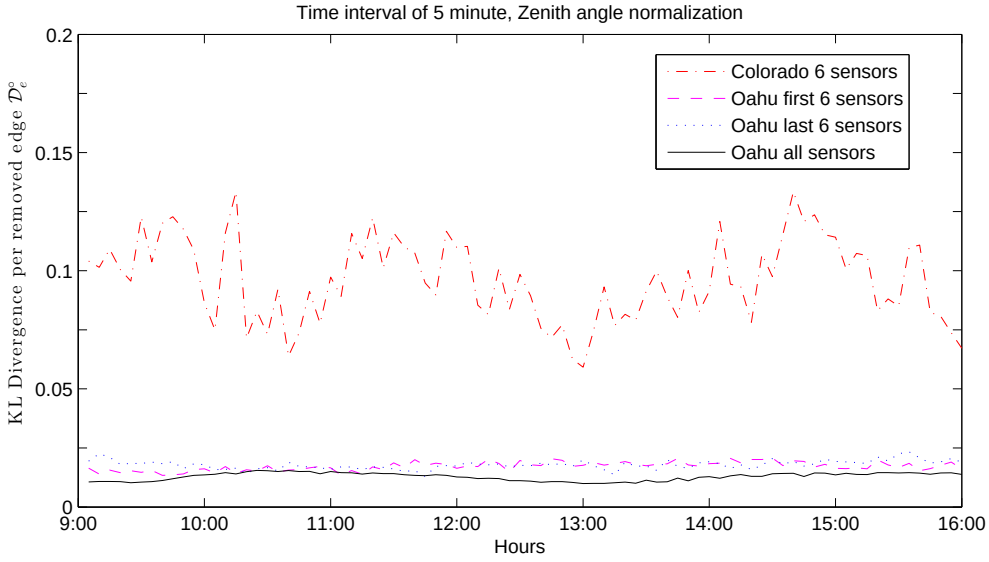

Figure 3.10: The normalized minimum KL divergence distance per removed edge for four scenarios: 1) all the 6 Colorado sensors, 2) the first 6 Oahu sensors (201-206 sensors), 3) the last 6 Oahu sensors (212-217 sensors) and 4) all the 19 Oahu sensors.

In the following Examples, a covariance matrix is calculated based on presented datasets [2]. Two datasets are used from the National Renewable Energy Laboratory (NREL) website [55]. The first dataset is the Oahu solar measurement grid which consists of 19 sensors (17 horizontal sensors and two tilted sensors) and the second one is the NREL solar data for 6 sites near Denver, Colorado. These two data sets are normalized using standard normalization method and the zenith angle normalization method [2] and then the unbiased estimate of the correlation matrix is computed [3].

3.1.2.3 The Oahu solar measurement grid dataset

From data obtained from 19 solar sensors at the island of Oahu, we computed the spatial covariance matrix during the summer season at 12:00 PM averaged over a window of 5 minutes. Then, the AUC and the KL divergence are computed for those tree structures that are generated using Markov Chain Monte-Carlo (MCMC) method. Figure 3.11 shows the distribution of those tree structures generated using MCMC method versus the KL divergence **(left)** and v.s $\log_{10}(1 - \text{AUC})$ **(right)** [4].

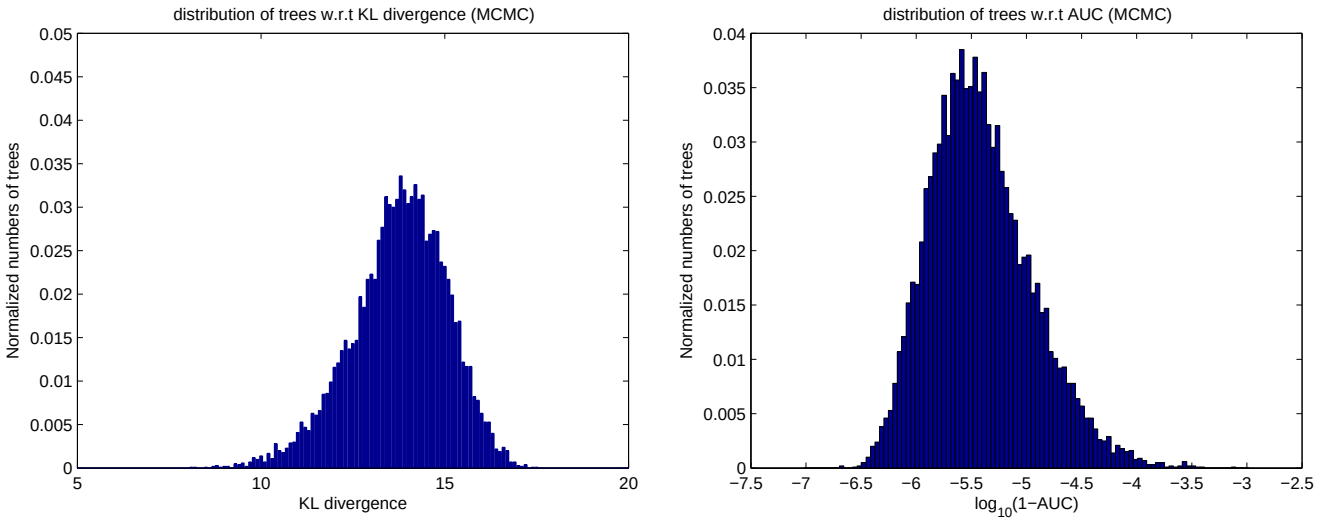

Figure 3.11: **Left:** distribution of the generated trees (Normalized histogram) using MCMC v.s. the KL divergence and **Right:** distribution of the generated trees (Normalized histogram) using MCMC v.s. $\log_{10}(1 - \text{AUC})$ for the Oahu solar measurement grid dataset in summer season at 12:00 PM.

3. See [2] for other details about the normalization methods for the solar irradiation covariance matrix.

4. In this example, since the AUC for all generated tree structures is close to one, we plots the distribution of generated trees v.s. $\log_{10}(1 - \text{AUC})$.

Looking back at figure 2.4, for the very small value of $1 - \text{AUC}$ the relationship between the KL divergence and the boundary of the possible feasible region for $-\log(1 - \text{AUC})$ is linear. This means that if the upper bound is tight then the relationship between the KL divergence and the $-\log(1 - \text{AUC})$ is almost linear. In figure 3.11, the maximum value of $1-\text{AUC}$ for this model is less than 10^{-3} which justifies why two distributions in figure 3.11 are scaled/mirrored of each other. Moreover, just by looking at the distribution of tree models in this example, it is obvious that most tree models have similar performance. Only a small portion of the tree models have better performance than the average tree models, but the difference is not that significant.

3.1.2.4 The Colorado dataset

From the solar data obtained from 6 sensors near Denver, Colorado, we computed the spatial covariance matrix during the summer season at 12:00 PM averaged over a window of 5 minutes. Then, the AUC and the KL divergence are computed for all possible tree structures. Figure 3.12 shows the distribution of all possible tree structures v.s the KL divergence **(left)** and v.s the AUC **(right)**.

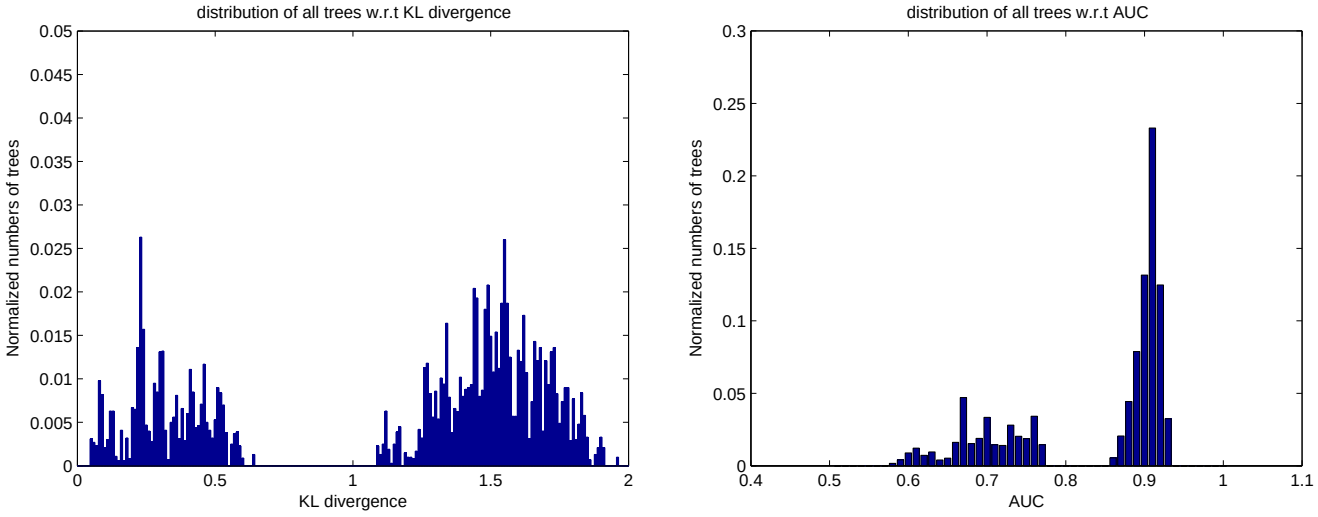

Figure 3.12: **Left:** distribution of all trees (Normalized histogram) v.s. the KL divergence and **Right:** distribution of all trees (Normalized histogram) v.s. the AUC for the Colorado dataset in summer season at 12:00 PM.

In the Colorado dataset, there are two sensors that are very close to each other compared to the distance between all other pairs of sensors. As a result, if the particular edge between these two

sensors is in the approximated tree structure we get a smaller AUC and KL divergence compared to when that particular edge is not in the tree structure. This explains why the distribution of all trees, in this case, looks like a mixture of two distributions. This result also gives us valuable insight on how to answer the following question, "How to construct informative approximation algorithms for model selection in general." This is an example where almost all trees that contain the particular edge between the two aforementioned sensors are good approximations while the rest of the tree models' give poor performance.

3.1.3 Two-dimensional sensor network example

In this example, we create a 2-dimensional (2D) sensor network using a Gaussian kernel [57] as follows

$$\Sigma_{\underline{X}}(i,j) = \left[e^{-\frac{d(i,j)^2}{2\sigma^2}} \right]$$

where $d(i,j)$ is the Euclidean distance between the i-th sensor and the j-th sensor in the 2D space. All sensors are located randomly in 2D space [5]. We set $\sigma = 1$ and generate a 2D sensor network with 20 sensors. For the 2D sensor network example, figure 3.13 shows the distribution of the generated tree structures using MCMC method v.s KL divergence **(left)** and v.s $\log_{10}(1 - \text{AUC})$ **(right)**. Again we see the mirroring effect in Fig. 3.13 as we have an almost linear relationship between the KL divergence and $-\log(1-\text{AUC})$. Note that, the covariance matrix generated has one dominant eigenvalue in most cases. Furthermore, figure 3.14 plots $1-\text{AUC}$ as well as its analytical upper bound and lower bound v.s. the dimension of the graph, n for $\sigma = 1.3$ **(left)** and $\sigma = 1.8$ **(right)**. To generate this figure, we randomly generated 1000 sensor networks and then plot the averaged AUC. As we can see in this figure, the $1-\text{AUC}$ and its bounds decay exponentially which is consistent with the theoretical results of this paper.

5. Sensors location in each dimension are drawn randomly from a Normal distribution.

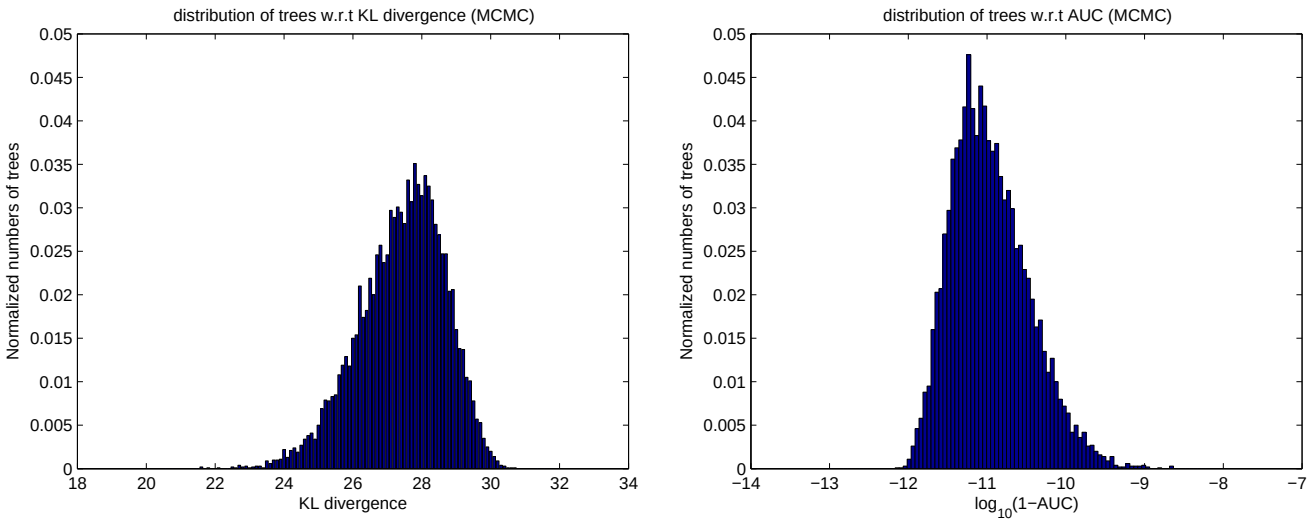

Figure 3.13: **Left:** distribution of the generated trees (Normalized histogram) using MCMC v.s. the KL divergence and **Right:** distribution of the generated trees (Normalized histogram) using MCMC v.s. $\log_{10}(1 - \text{AUC})$ for the 2D sensor network example with 20 sensors and $\sigma = 1$.

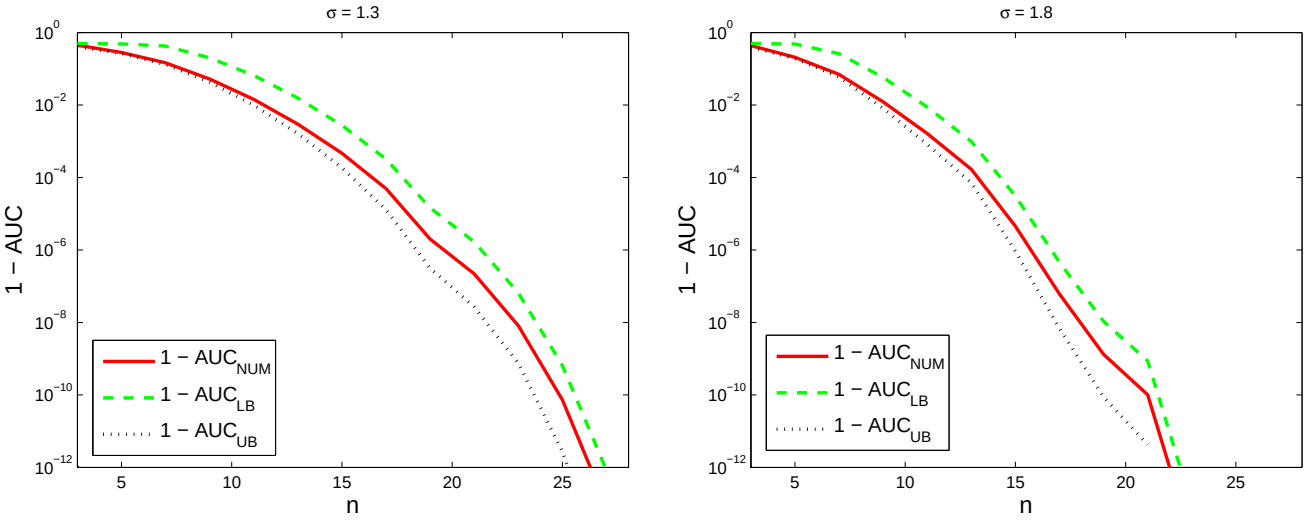

Figure 3.14: 1−AUC and its bounds v.s. the dimension of the graph, n for $\sigma = 1.3$ **(left)** and $\sigma = 1.8$ **(right)**, averaged over 1000 runs of sensor networks generated randomly.

3.2 Part II: Beyond tree approximation

In this part, we consider graphical models beyond simple tree structures.

3.2.1 Toeplitz Covariance Matrix

In this section, again we assumed that the n by n covariance matrix $\mathbf{\Sigma}_{\underline{X}}$ has a Toeplitz structure with ones on the diagonal and the correlation coefficient ρ as off-diagonal elements

$$\mathbf{\Sigma}_{\underline{X}} = \begin{bmatrix} 1 & \rho & \dots & \rho \\ \rho & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \rho \\ \rho & \dots & \rho & 1 \end{bmatrix}.$$

We aim to analyze the quality of models beyond tree structures discussed in Part I of this chapter.

Definition 8. Clique. *A maximal subset of the nodes which defines a complete subgraph is the clique subgraph.* ■

In other words, all pairs of nodes are connected in the clique subgraph.

Definition 9. Junction tree. *A junction tree is a clique tree [58] such that for each pair of cliques C_1 and C_2 in the graph, all cliques on the path between C_1 and C_2 contain their intersection, $C_1 \cap C_2$.* ■

In this example, we are interested in models which can be represented using junction trees whose vertices are cliques of the size p.[6] Going back to the model selection problem for the example, we are investigating the following two generalizations of the chain and the star networks. Note that, we can construct a junction tree for these two special models.

6. We avoid cycles by turning subsets of the nodes into supernodes.

3.2.2 pth order star network

The model covariance matrix for the pth order star network where all nodes are connected to the first p nodes which all are connected together is

$$\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}^{pth-star} = \begin{bmatrix} 1 & \rho & \dots & \dots & \dots & \dots & \rho \\ \rho & \ddots & \ddots & & & & \vdots \\ \vdots & \ddots & 1 & \rho & \dots & \dots & \rho \\ \vdots & & \rho & 1 & \rho_1 & \dots & \rho_1 \\ \vdots & & \vdots & \rho_1 & \ddots & \ddots & \vdots \\ \vdots & & \vdots & \vdots & \ddots & \ddots & \rho_1 \\ \rho & \dots & \rho & \rho_1 & \dots & \rho_1 & 1 \end{bmatrix} \Bigg\} p$$

where

$$\rho_1 = \frac{p\rho^2}{(p-1)\rho + 1}.$$

3.2.3 pth order Markov chain network

The model covariance matrix for the pth order Markov chain network is as follow

$$\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}^{pth-chain} = \begin{bmatrix} 1 & \rho & \dots & \rho & \rho_1 & \dots & \rho_{n-p-1} \\ \rho & \ddots & \ddots & & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \rho_1 \\ \rho & & \ddots & \ddots & \ddots & & \rho \\ \rho_1 & \ddots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & & \ddots & \ddots & \rho \\ \rho_{n-p-1} & \dots & \rho_1 & \rho & \dots & \rho & 1 \end{bmatrix} \Bigg\} p .$$

To satisfy Theorem 1 we have that ρ_i for $i \in \{1, \dots, n-p-1\}$ can be computed through the following recursive equation

$$\rho_i = \underline{\rho}_{i-1}^T \, \underline{v}_i \, \frac{\rho}{(p-1)\rho + 1} \tag{3.1}$$

where $\underline{v}_i = [\overbrace{1,\ldots,1}^{p},0,\ldots,0]^T$ is a vector of length n and $\underline{\rho}_i = [\rho_i,\ldots,\rho_1,\overbrace{\rho,\ldots,\rho}^{p}]^T$ where $\underline{\rho}_0 = [\overbrace{\rho,\ldots,\rho}^{p}]^T$ is the initialization step.

Lemma 3. *The KL divergence for the pth order star network and the pth order Markov chain network can be calculated as*

$$\mathcal{D}(\underline{X}||\underline{X}_{pth-chain}) = \frac{1}{2}(n-p)\log\left(\frac{p\rho+1}{(p-1)\rho+1}\right) + \frac{1}{2}\log\left(\frac{(p-1)\rho+1}{(n-1)\rho+1}\right)$$

and

$$\mathcal{D}(\underline{X}||\underline{X}_{pth-star}) = \mathcal{D}(\underline{X}||\underline{X}_{pth-chain}).$$

Proof. Note that, from [42] we have

$$|\mathbf{\Sigma}^{pth-chain}_{\underline{X}_{\mathcal{M}}}| = \frac{[(p\rho+1)(\rho-1)^p]^{(n-p)}}{[((p-1)\rho+1)(\rho-1)^{p-1}]^{(n-p-1)}}$$

and

$$|\mathbf{\Sigma}_{\underline{X}}| = ((n-1)\rho+1)(\rho-1)^{n-1}.$$

Inserting the values of these determinants into the KL divergence

$$\mathcal{D}(\underline{X}||\underline{X}_{\mathcal{M}}) = -\frac{1}{2}\log\left(|\mathbf{\Sigma}_{\underline{X}}\mathbf{\Sigma}^{-1}_{\underline{X}_{\mathcal{M}}}|\right)$$

we conclude the result for the pth order Markov chain network. To show that the KL divergence for the pth order star network is exactly equal to the KL divergence for the pth order chain network, we need to construct the corresponding junction tree for each of these networks by grouping appropriate p nodes.

Remark: Note that the KL divergence for the junction trees are equal since the mutual information between the junction nodes are exactly equal. In other words, there are $\binom{n}{p}$ nodes in the junction tree and all the possible junction trees have the same KL divergences. ■

Theorem 4. *The KL divergence for the pth order star network and the pth order Markov chain network is bounded as n goes to infinity if for a given constant number, $\kappa > 1$, the order, p, is the*

integer number in interval,

$$\mathcal{D}(\underline{X}||\underline{X}_{pth-star}) < \infty \qquad \text{as} \qquad (n \to \infty,\ {}^{n}\!/_{p} \to \kappa).$$

Proof. Let $p = \lceil {}^{n}\!/_{\kappa} \rceil$ be the smallest integer greater than or equal to ${}^{n}\!/_{\kappa}$. The KL divergence can be bounded as follow

$$\begin{aligned}
\mathcal{D}(\underline{X}||\underline{X}_{pth-star}) &= \frac{(n - \lceil {}^{n}\!/_{\kappa} \rceil)}{2} \log\left(1 + \frac{\rho}{(\lceil {}^{n}\!/_{\kappa} \rceil - 1)\rho + 1}\right) \\
&\quad + \frac{1}{2}\log\left(\frac{(\lceil {}^{n}\!/_{\kappa} \rceil - 1)\rho + 1}{(n-1)\rho + 1}\right) \\
&\overset{(a)}{\leqslant} \frac{(n - {}^{n}\!/_{\kappa})}{2} \log\left(1 + \frac{\rho}{({}^{n}\!/_{\kappa} - 1)\rho + 1}\right) \\
&\quad + \frac{1}{2}\log\left(\frac{(({}^{n}\!/_{\kappa} + 1) - 1)\rho + 1}{(n-1)\rho + 1}\right) \\
&\overset{(b)}{\leqslant} \frac{(1 - {}^{1}\!/_{\kappa})n}{2} \left(\frac{\rho}{({}^{n}\!/_{\kappa} - 1)\rho + 1}\right) \\
&\quad + \frac{1}{2}\log\left(\frac{({}^{n}\!/_{\kappa})\rho + 1}{(n-1)\rho + 1}\right)
\end{aligned}$$

Where (a) is true since for the integer order, p, we have ${}^{n}\!/_{\kappa} \leqslant p < {}^{n}\!/_{\kappa} + 1$ and (b) is true since $\log(1+z) \leqslant z$ for $z \geqslant 0$. Then, in the limit we have

$$\begin{aligned}
\lim_{n\to\infty} \mathcal{D}(\underline{X}||\underline{X}_{pth-star}) &\leqslant \frac{(1 - {}^{1}\!/_{\kappa})}{{}^{2}\!/_{\kappa}} + \frac{1}{2}\log({}^{1}\!/_{\kappa}) \\
&\leqslant \frac{\kappa - 1}{2} - \frac{\log(\kappa)}{2} < \infty
\end{aligned}$$

which complete the proof. ■

Theorem 5. *The AUC of the pth order star network and the pth order Markov chain network is is strictly less than* 1 *as* n *goes to infinity if* $p = \lceil {}^{n}\!/_{\kappa} \rceil$,

$$\Pr\left(L_{\Delta} > 0\right) \leqslant 1 - e^{-\frac{\kappa + 1 - \log(\kappa)}{2}} < 1.$$

Proof. We can conclude this result from Proposition 4 which gives an upper bound for the KL

divergence combined with the upper bound for the AUC,

$$\begin{aligned}\Pr\left(L_\Delta > 0\right) &\leqslant 1 - e^{-\lim_{n\to\infty} \mathcal{D}(\underline{X}||\underline{X}_{pth-star})-1} \\ &\leqslant 1 - e^{-\frac{\kappa+1-\log(\kappa)}{2}} \\ &< 1\end{aligned}$$

where the AUC upper bound is provided in [17]. ■

Remark: For the constant κ the gap, $e^{-\frac{\kappa+1-\log(\kappa)}{2}}$, in Theorem 5 indicates the goodness of a model approximation, i.e. the larger the gap, the better the model approximation will be.

Remark: The gap here is only calculated based on the upper bound for the KL divergence, $\lim_{n\to\infty} \mathcal{D}(\underline{X}||\underline{X}_{pth-star})$. Note that, the same calculation can be done for the reverse KL divergence, $\lim_{n\to\infty} \mathcal{D}(\underline{X}_{pth-star}||\underline{X})$, for both star and chain networks.

3.2.4 Simulation Results and Discussion

In this section, we consider the Toeplitz example presented before as the covariance matrix for a Gaussian random vector. We calculate different models such for the pth order Markov chain and the pth order star networks for various values of p. For a given order, both of the aforementioned models have the same KL divergence values as calculated in Lemma 3. Moreover, we compute AUC and compare it with its lower and upper bounds [17] for these cases.

Figure 3.15 plots (1 - AUC) in logarithmic-scale v.s. the dimension of the graph, n, in linear-scale for star approximation **(left)** and chain approximation **(right)** with different model orders, $p = 1$, $p = 3$, $p = 5$ and $p = 7$ for correlation coefficient $\rho = 0.9$. As it is indicated in this figure, (1 - AUC) decreases as the order of the model increases for both star and chain models. From this figure, we can conclude that the pth order star network performs better than the pth order Markov chain network since the exponential decay of (1-AUC) is smaller for the former model than the latter model. This can also be seen by comparing the covariance matrix $\mathbf{\Sigma}_{\underline{X}}$ and the model covariance matrix, $\mathbf{\Sigma}_{\underline{X}_{\mathcal{M}}}$ where the model covariance matrix associated with the pth order star network is more similar to the covariance matrix $\mathbf{\Sigma}_{\underline{X}}$ than the model covariance matrix associated with the pth order

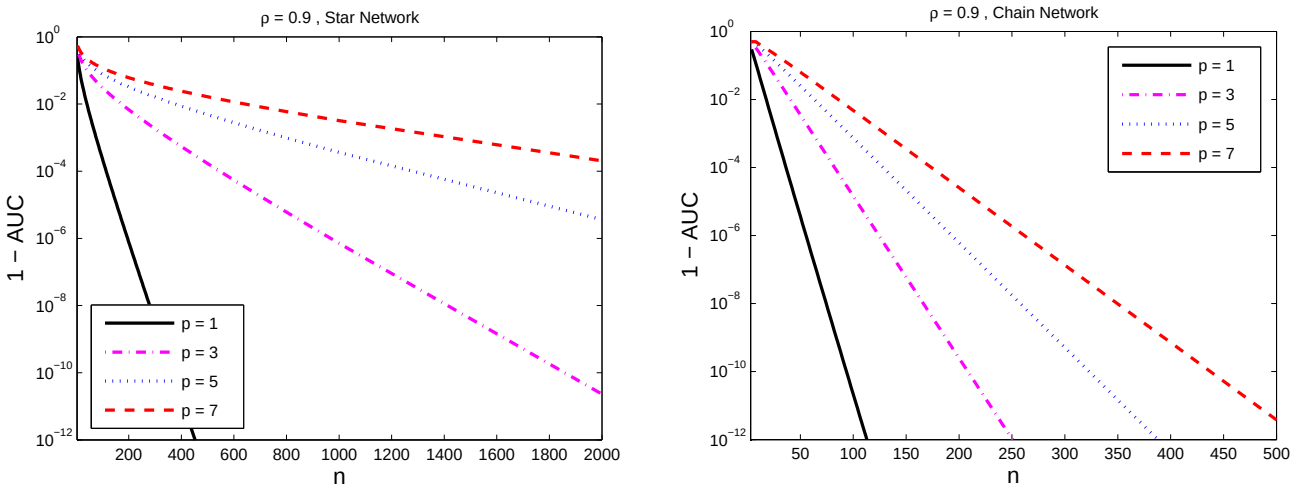

Figure 3.15: 1 - AUC (logarithmic-scale) v.s. the dimension of the graph (linear-scale), n, for star approximation **(left)** and chain approximation **(right)** with different model orders, $p = 1$, $p = 3$, $p = 5$ and $p = 7$ and correlation coefficient $\rho = 0.9$.

Markov chain network. For example, even the quality of the first order star network approximation is better than the quality of the fifth order Markov chain approximation in the simulation results provided in this figure. Figure 3.16 plots the same curves as figure 3.15 in a arithmetic scale.

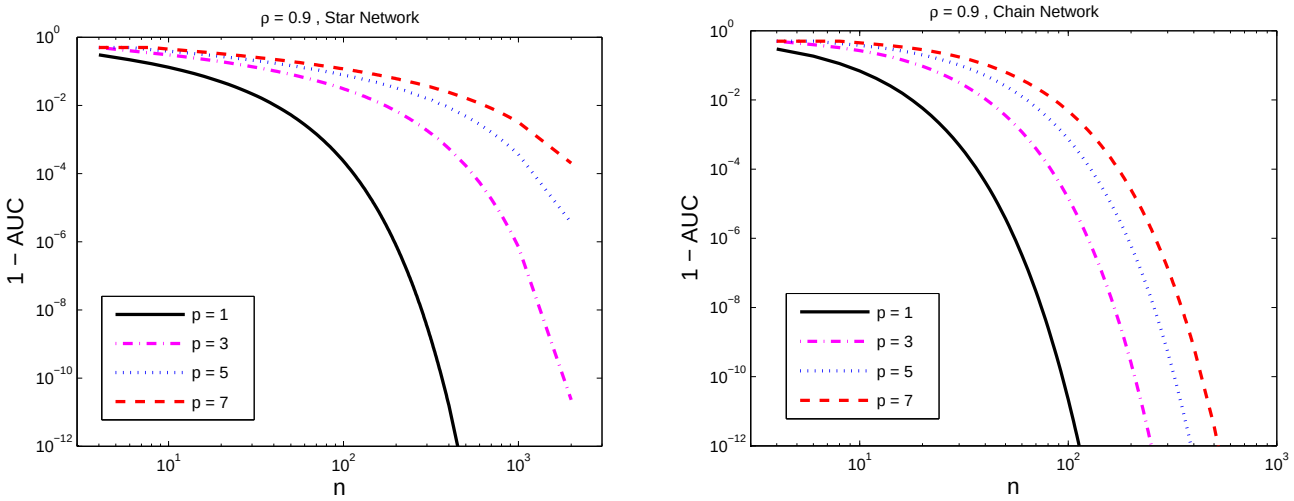

Figure 3.16: 1 - AUC v.s. the dimension of the graph, n, for star approximation **(left)** and chain approximation **(right)** with different model orders, $p = 1$, $p = 3$, $p = 5$ and $p = 7$ and correlation coefficient $\rho = 0.9$.

Figure 3.17 plots KL divergence v.s $-\log$ (1 - AUC) for the presented models. In this figure, the dimension n is set to 15, the order p is set to 1 and 3 and the correlation coefficient ρ is set to 0.9. Furthermore, the possible feasible region presented in [17] and its asymptotic behavior are

also plotted in this figure. For both models, the KL divergence and the reverse KL divergence are computed and are plotted on this figure. Note that, KL divergences for both models are equal (see Lemma 3) and are connected in this figure. As it is shown in the figure, the third order model has better performance than the first order model [7].

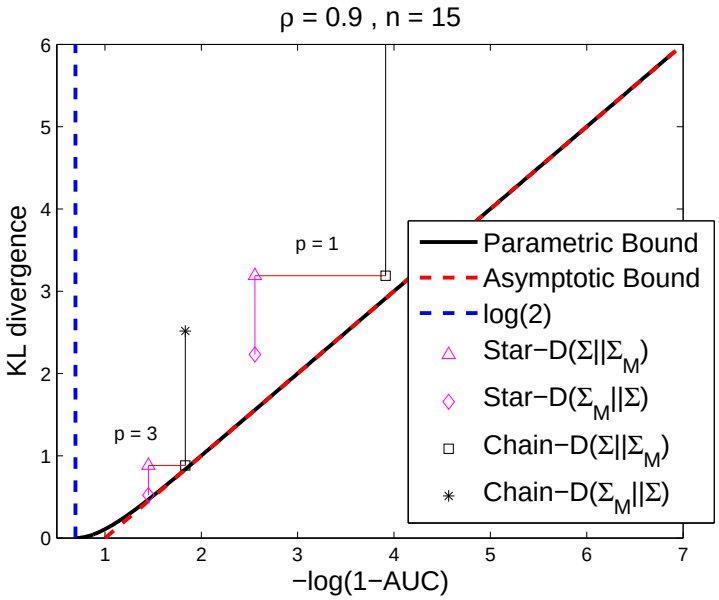

Figure 3.17: KL divergence v.s. AUC and the AUC parametric bound v.s. for graph dimension, $n = 15$ for the pth order Markov chain approximation and pth order star network for $p = 1$ and $p = 3$ with $\rho = 0.9$.

Figure 3.18 plots KL divergence v.s AUC for different graph dimensions, n, and different models. In this figure, order, p is set to one while the correlation coefficient, ρ is set to 0.1 on the left and 0.9 on the right. Moreover, the feasible region presented in [17] and its asymptotic behavior are also plotted in this figure. The AUC has computed for the first order Markov chain approximation and first order star approximation. For each model, the KL divergence and the reverse KL divergence are computed. Note that, KL divergences for both models are equal (see Lemma 3). From this figure, we can see that the value of the reverse KL divergence for star model is less than the actual KL divergence and vice versa for the chain model. Since KL divergences are equal for both models, we conclude that the star network has better quality than the chain network which we already know from the associated AUC value for each model for a fixed order, p and dimension, n. We can also observe from this figure that the approximation with a smaller ρ has better quality, i.e. has smaller associated AUC. Furthermore, setting $p = 9$, figure 3.19 shows similar results as figure 3.18. In

7. More simulation results and discussion about the points in the possible feasible region can be found in [59].

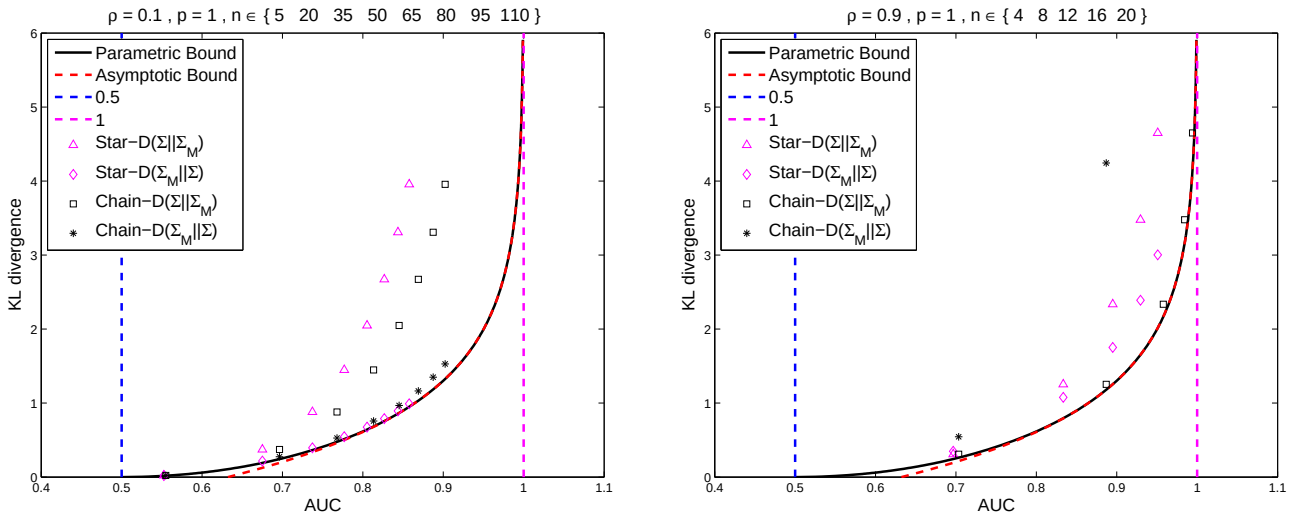

Figure 3.18: KL divergence v.s. AUC and the AUC parametric bound v.s. for different dimension of the graph, n for the pth order Markov chain approximation and pth order star network for $p = 1$ with $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)**.

these figures, the value of the KL divergence and its reverse increase as the number of nodes in the graph increases. Comparing these two figures we can observe that for a fixed graph dimension, the approximation is better for larger approximation order, p, i.e. approximation with the larger order has smaller AUC.

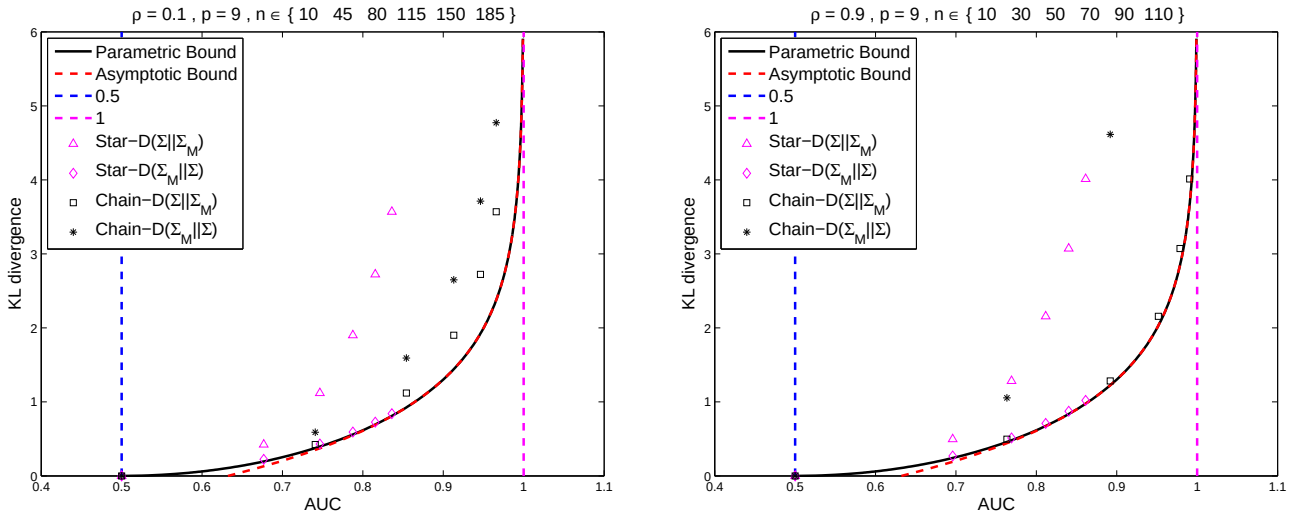

Figure 3.19: KL divergence v.s. AUC and the AUC parametric bound v.s. for different dimension of the graph, n for the pth order Markov chain approximation and pth order star network for $p = 9$ with $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)**.

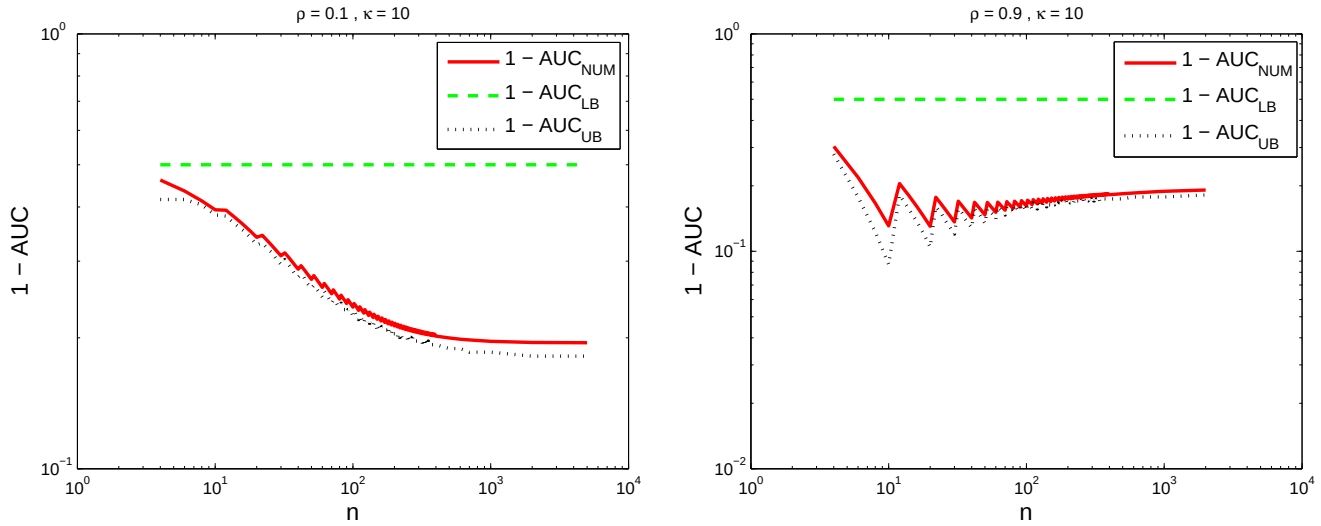

Figure 3.20: 1 - AUC and its lower and upper bounds v.s. the dimension of the graph, n for the pth order star approximation of the Toeplitz example for $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)** with the model order $p = \lceil n/\kappa \rceil$ where $\kappa = 10$.

Figure 3.20 plots 1 - AUC v.s. the dimension of the graph, n for the pth order star approximation of the Toeplitz example for $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)** while keeping the model order proportional to the number of nodes in the graphical model, n. More specifically, in this figure, we set the model order $p = \lceil n/\kappa \rceil$ where $\kappa = 10$. This figure also plots the lower bound and the upper bound for 1 - AUC.[8] From this figure, we conclude that, pth order star approximation is a good approximation model when the model order, p is proportional to the number of nodes, n, since the AUC is bounded from one as $n \to \infty$. Similarly, figure 3.21 plots 1 - AUC and its upper and lower bounds v.s. the dimension of the graph, n for the pth order Markov chain approximation of the Toeplitz example for $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)** with $p = \lceil n/\kappa \rceil$ where $\kappa = 10$. Plots in this figure are not monotonically decreasing since both the order p and the dimension n are integers and thus the ratio n/p is not exactly equal to κ for all values of p and n. Furthermore, from the figure, the pth order Markov chain approximation is a good approximation model when the model order, p is proportional to the number of nodes, n, since the AUC is bounded from one as $n \to \infty$. Comparing the plots in figure 3.20 and figure 3.21 we can clearly see that even though the AUC for both approximation models are bounded from one, the pth order star approximation model is a better model than the pth order Markov chain approximation model.

8. Bounds are presented in [17].

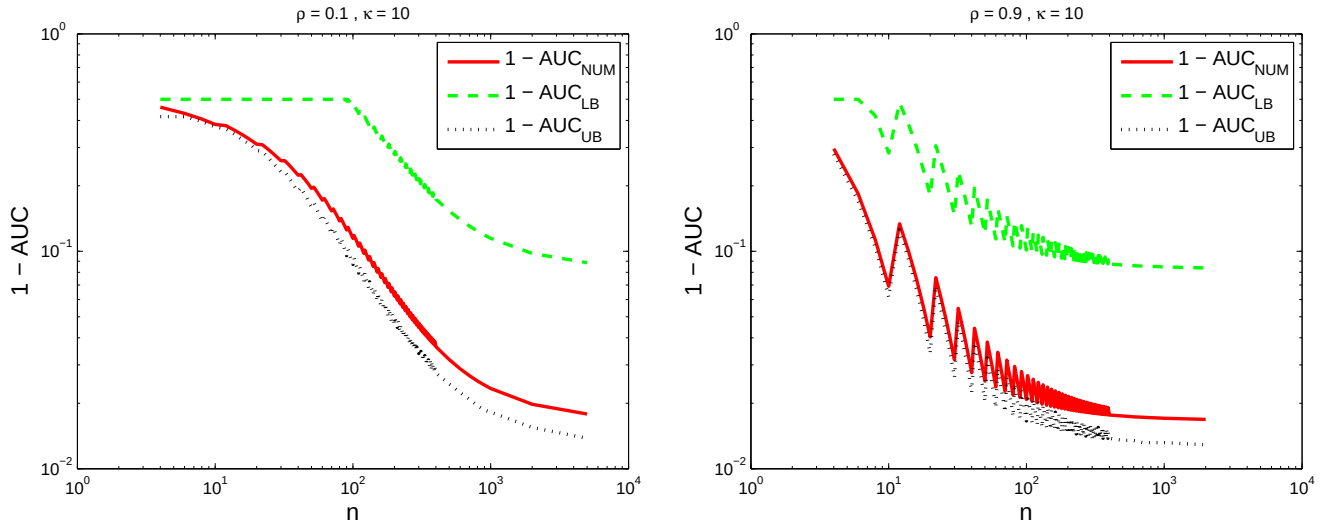

Figure 3.21: 1 - AUC and its lower and upper bounds v.s. the dimension of the graph, n for the pth order Markov chain approximation of the Toeplitz example for $\rho = 0.1$ **(left)** and $\rho = 0.9$ **(right)** with the model order $p = \lceil n/\kappa \rceil$ where $\kappa = 10$.

3.3 Conclusion

In this chapter, we discuss the quality of model approximation using the proposed detection framework and the AUC bounds. In the first part, we pick tree graphs as an example of an approximation model. We use the Chow-Liu MST algorithm to compute the maximum likelihood tree structure approximation and investigate the quality of tree model using the proposed framework. Through some examples, we show that in general, the tree approximation is not a good model as the number of nodes in the graphical model increases. The aforementioned result is also consistent with the analytical results provided in this paper that is $1-$AUC decays exponentially as the dimension of graph increases. In the second part, we look at more accurate graphical approximations that involve non-tree graphs. We discuss graphical models with junction trees such as the pth order Markov chain and the corresponding star network interpretation for a special Toeplitz covariance matrix with ones along the diagonal and correlation coefficient ρ's on the off-diagonals. These models have very short loops and have associated junction tree that connects cliques of the same size. The model covariance matrix as well as the KL divergence between the original distribution and the model distribution are computed for the presented Toeplitz covariance matrix. We also quantify the goodness of the covariance selection problem for this Toeplitz covariance matrix. For this covariance matrix, we show that if the model order, p, is proportional to the number of nodes, n, then

the model selection is asymptotically good as $n \to \infty$ since the AUC is asymptotically bounded away from one. We conduct some simulations which show that the selected model quality increases as the model order, p, increases which confirm our theoretical results.

4

Model Approximation Using Cascade of Tree Decompositions

We continue our study of graphical models and discuss statistical model approximation for large graphs. We are specifically looking at the statistical model approximation for jointly Gaussian random vectors. In this chapter, we present a general, multistage framework for graphical model approximation using a cascade of models such as trees. In particular, we look at the problem of covariance matrix approximation for Gaussian distributions as linear transformations of tree models. This is a new way to decompose the covariance matrix. Here, we propose an algorithm which incorporates the Cholesky factorization method to compute the decomposition matrix and thus can approximate a simple graphical model using a cascade of the Cholesky factorization of the tree approximation transformations. The Cholesky decomposition enables us to achieve a tree structure factor graph at each cascade stage of the algorithm which facilitates the use of the message

passing algorithm since the approximated graph has less loops compared to the original graph. The overall graph is a cascade of factor graphs with each factor graph being a tree. This is a different perspective on the approximation model, and algorithms such as Gaussian belief propagation can be used on this overall graph. Here, we present theoretical result that guarantees the convergence of the proposed model approximation using the cascade of tree decompositions. In the simulations, we look at synthetic and real data and measure the performance of the proposed framework by comparing the KL divergences.

4.1 Introduction

Learning from high dimensional data requires large computational power which is not always available. In signal processing and machine learning a fundamental problem is to balance performance quality (i.e. minimizing cost function) with computational complexity. A powerful tool in order to address this trade-off is graphical model selection. Model selection methods provide approximated models with the desired accuracy as needed for different applications. Given data, different model selection algorithms impose different structures to model data [20].

Tree approximation algorithms are among the algorithms that reduce the number of computations in order to achieve quicker approximate solutions to a variety of problems. The tree approximations are made as it is much simpler to perform inference and estimation on trees rather than graphs that have cycles or loops. An example is applying the Gaussian belief propagation (BP) algorithm [24] which will converge to the maximum likelihood solution over loop-free graphs. [1] While these algorithms approximate the correlation matrix with a more sparse graph, in many cases as the number of nodes increases in large datasets, they fail to retain the desired accuracy [17]. As a result, in many applications, we need to go beyond the tree structure approximation to achieve design accuracy that can be translated to any model approximation that can achieve a KL divergence below a certain design threshold.

Another related and mature body of work in literature is on mixture models [6] and [8], including works on mixtures of tree approximations [61], [62] and [63] and Gaussian mixture model (GMM) [64] for graphical models. While in this chapter, we are generalizing a single-tree approximation

1. The convergence of Gaussian BP with multiple loops is analyzed in literature [60].

algorithm and using a sequence of tree approximations for sparse model approximation, the aforementioned mixture of tree approximation methods considers parallel trees.

The purpose of this chapter is present a general framework to reduce the computational complexity of distributed algorithms in various applications while maintaining the desired approximation quality. To achieve this goal we approximate the associated Gaussian graphical model with a simpler, more tractable model. In this chapter, we consider jointly Gaussian data and use cascade of tree transformation decompositions in order to perform model approximation for graphical models. The tree structure model is considered since this structure is simple and the optimal solution that minimizes the KL divergence can be easily computed using the Chow-Liu algorithm [3]. Furthermore, the tree structure model is a loop-free model and simplifies the implementation of distributed algorithms such as Gaussian BP. The cascade tree framework enables us to approximate a complex model with multiple stages of simple tractable models such as the tree structured model. We pick trees as the model and the Cholesky decomposition to factor the the tree structured covariance matrix at each stage of the cascade algorithm. Implementation of the Cholesky decomposition with the proper node ordering (permutation matrix) enables us to draw a *tree structured* factor graph for each step of the cascade tree decomposition transformation. This property facilitates the use of Gaussian BP algorithm over the aforementioned factor graph. We perform some simulations to confirm the results of this chapter by looking at synthetic and real data and compare the performance of the proposed framework by comparing KL divergences. We also consider the singular value decomposition (SVD) and compare its performance to the Cholesky decomposition. Our simulation results also confirm the advantages of the cascade tree framework.

Many engineering and computer science applications require using graphs to model dependencies between nodes of the graph. These applications include a diversity of areas from social networking to biomedical applications to transportation models to energy models. For these applications graphs must be approximated by simpler structures to reduce computational complexity.

The rest of this chapter is organized as follows. In section 4.2 we provide a summary of Gaussian tree approximation. The Gaussian model approximation as a transformation is also discussed in this section. Section 4.3 presents the theory behind the proposed model approximation framework. The symmetric correlation approximation matrix (CAM) is defined and the convergence theorem is discussed in this section. Section 4.4 provides a greedy algorithm for the model approximation

using cascade of tree decompositions. The proposed algorithm is based on the symmetric CAM, the tree approximation algorithm and its Cholesky decomposition. This algorithm is suitable for message passing and Gaussian BP over factor graphs since we use the Cholesky decomposition at each of the cascade stages. In section 4.4 we also present a simple example illustrating the Cholesky algorithm transformations and the cascade of factor graphs. Section 4.5 provides some simulations over synthetic examples as well as a real solar data example from the island of Oahu obtained from an NREL website and investigates the quality of the proposed model approximation by looking at the KL divergence. Finally, Section 4.6 summarizes results of this chapter.

4.2 Gaussian tree approximation

In this section, we first review the tree approximation algorithm for Gaussian distributions. Then we explain the framework for the covariance transformation decomposition for any given model such as the tree model. The tree structure is a simple graphical model and can be computed efficiently. The loop-free structure of the tree structure also facilitates the implementation of distributed algorithms such as Gaussian BP. Later in the next section, we use the cascade of tree transformation decompositions to perform model approximation for graphical models.

4.2.1 Tree approximation for Gaussian distributions

In the tree approximation, we want to approximate a multivariate distribution by the product of lower order component distributions [36]. Let $\underline{X} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma})$ (i.e. jointly Gaussian with mean 0 and covariance matrix $\mathbf{\Sigma}$) where $\underline{X} \in \mathbb{R}^n$ have the graph representation $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where sets $\mathcal{V}$ and $\mathcal{E}$ are the set of all vertices and edges of the graph representing $\underline{X}$.[2] Let $\underline{X}_{\mathcal{T}} \sim \mathcal{N}(\underline{0}, \widetilde{\mathbf{\Sigma}})$ have the graph representation $\mathcal{G}_{\mathcal{T}} = (\mathcal{V}, \mathcal{E}_{\mathcal{T}})$ where $\mathcal{E}_{\mathcal{T}} \subseteq \mathcal{E}$ is a set of edges that represents a tree structure. The joint probability density function can be represented by joint PDFs of two variables and marginal PDFs in the following convenient form

$$f_{\underline{X}_{\mathcal{T}}}(\underline{x}) = \prod_{(u,v)\in\mathcal{E}_{\mathcal{T}}} \frac{f_{\underline{X}^u,\underline{X}^v}(\underline{x}^u, \underline{x}^v)}{f_{\underline{X}^u}(\underline{x}^u) f_{\underline{X}^v}(\underline{x}^v)} \prod_{o\in\mathcal{V}} f_{\underline{X}^o}(\underline{x}^o). \tag{4.1}$$

2. Here, we assume that all nodes are connected in the graphical structure of vector $\underline{X}$.

Definition 10. *Let $\mathcal{T}_{\boldsymbol{\Sigma}}$ denote the set of all positive definite covariance matrices with following properties:*

1) These covariance matrices have tree structured Gaussian graphical models;

2) Picking any covariance matrix in this set, $\widetilde{\boldsymbol{\Sigma}} \in \mathcal{T}_{\boldsymbol{\Sigma}}$, the Gaussian distributions $\mathcal{N}(\underline{0}, \widetilde{\boldsymbol{\Sigma}})$ and $\mathcal{N}(\underline{0}, \boldsymbol{\Sigma})$ have the same marginal distributions and joint distribution of two variables over the tree structured graph, $\mathcal{G}_{\mathcal{T}}$. ■

In the above definition $\mathcal{N}(\underline{0}, \widetilde{\boldsymbol{\Sigma}})$ obeys the product rule given in (4.1). Also, note that, the cardinality of the set $\mathcal{T}_{\boldsymbol{\Sigma}}$ is finite [65] since the number of all possible tree structured graphs with n nodes is finite.

Chow-Liu MST method [3], was initially proposed for approximating the joint distribution of discrete variables by product of lower order distributions similar to (4.1) which involves no more than a pair of variables. The proposed KL divergence is used to quantify the distance between any distribution and its tree structure approximation. The Chow-Liu MST algorithm for Gaussian distributions, minimizes the following optimization problem in order to find the optimal tree structured covariance matrix, $\boldsymbol{\Sigma}_{\mathcal{T}} \in \mathcal{T}_{\boldsymbol{\Sigma}}$

$$\boldsymbol{\Sigma}_{\mathcal{T}} = \underset{\widetilde{\boldsymbol{\Sigma}} \in \mathcal{T}_{\boldsymbol{\Sigma}}}{\arg\min}\ \mathcal{D}(f_{\underline{X}}(\underline{x}) || f_{\underline{X}_{\mathcal{T}}}(\underline{x})). \tag{4.2}$$

Here, $\mathcal{D}^{\star} \triangleq -\frac{1}{2}\log(|\boldsymbol{\Sigma}\boldsymbol{\Sigma}_{\mathcal{T}}^{-1}|)$ which minimum KL divergence that gives the distance between the given distribution and its optimal tree approximation. It is shown in [3] that the optimal solution for this problem 4.2 can be found efficiently using greedy algorithms [33], [34]. Their algorithm can be easily generalized for approximating the optimal tree structure of the joint distribution of Gaussian variables using equation (4.1) by adding edges one at a time [42]. In other words, given the knowledge of $\boldsymbol{\Sigma}$, the Chow-Liu algorithm can efficiently compute the optimal solution, i.e. $\boldsymbol{\Sigma}_{\mathcal{T}} = \text{chowliu}(\boldsymbol{\Sigma})$.

Remark: In case that the covariance matrix $\boldsymbol{\Sigma}$ is not available, we can replace it with the empirical covariance matrix obtained from data.

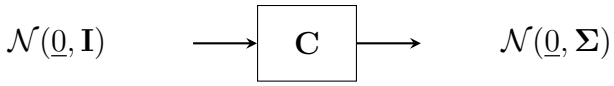

Figure 4.1a: Transformation from $\mathcal{N}(\underline{0}, \mathbf{I})$ to $\mathcal{N}(\underline{0}, \mathbf{\Sigma})$ using decomposition of the covariance matrix, $\mathbf{\Sigma}$.

Figure 4.1b: Transformation from $\mathcal{N}(\underline{0}, \mathbf{I})$ to $\mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\mathcal{M}})$ using decomposition of the model covariance matrix, $\mathbf{\Sigma}_{\mathcal{M}}$.

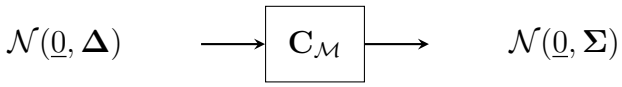

Figure 4.1c: Transformation from $\mathcal{N}(\underline{0}, \mathbf{\Delta})$ to $\mathcal{N}(\underline{0}, \mathbf{\Sigma})$ using decomposition of the model covariance matrix, $\mathbf{\Sigma}_{\mathcal{M}}$.

4.2.2 Gaussian model approximation as a transformation

Any zero-mean multivariate Gaussian distribution such as $\mathcal{N}(\underline{0}, \mathbf{\Sigma})$ where $\mathbf{\Sigma}$ is the covariance matrix, can be obtained through a linear transformation of the multivariate standard normal distribution, $\mathcal{N}(\underline{0}, \mathbf{I})$ (figure 4.1a) where $\mathbf{I}$ is the identity matrix. Moreover, the decomposition matrix $\mathbf{C}$ is defined as a square matrix that factors the covariance matrix $\mathbf{\Sigma}$, i.e. $\mathbf{\Sigma} \triangleq \mathbf{C}\mathbf{C}^T$. In this scenario, the decomposition matrix $\mathbf{C}$ is also the transformation matrix. We focus on the decomposition matrix $\mathbf{C}$ in more detail in section 4.4, some of the possible matrix decompositions that can be used to efficiently compute $\mathbf{C}$ are the Cholesky decomposition and singular value decomposition (SVD). Let's assume that the desired model covariance matrix, $\mathbf{\Sigma}_{\mathcal{M}}$ and its decomposition matrix, $\mathbf{C}_{\mathcal{M}}$, i.e. $\mathbf{\Sigma}_{\mathcal{M}} \triangleq \mathbf{C}_{\mathcal{M}}\mathbf{C}_{\mathcal{M}}^T$, are given. Then, from figure 4.1b, the model distribution $\mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\mathcal{M}})$ is the transformation of the multivariate standard normal distribution. However, to generate the Gaussian distribution with covariance matrix, $\mathcal{N}(\underline{0}, \mathbf{\Sigma})$, using the model decomposition matrix, $\mathbf{C}_{\mathcal{M}}$, the input distribution, $\mathcal{N}(\underline{0}, \mathbf{\Sigma})$, has to have a certain covariance matrix, $\mathbf{\Delta}$. This covariance matrix is called the symmetric correlation approximation matrix and is defined as $\mathbf{\Delta} = \mathbf{C}_{\mathcal{M}}^{-1}\mathbf{\Sigma}\mathbf{C}_{\mathcal{M}}^{-T}$. We will give a formal definition for the symmetric CAM in section 4.3 where we consider cascade of tree approximation decompositions for graphical model approximation.

Remark: *Invariance of Gaussian KL divergence with respect to transformation.* The KL divergence

Figure 4.2a: The i stages of the model transformation from $\underline{Z}_i \sim \mathcal{N}(\underline{0}, \mathbf{\Delta}_i)$ to $\underline{X} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma})$ using cascade tree decompositions.

Figure 4.2b: The l stages of model approximation using cascade tree transformation decomposition framework. The model approximation is generated by passing $\underline{W} \sim \mathcal{N}(\underline{0}, \mathbf{I})$ through the l steps of cascade trees and is $\underline{X}_{\mathcal{M}_l} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\mathcal{M}_l})$.

between the input Gaussian distributions in figures 4.1b and 4.1c is invariant to the transformation $\mathbf{C}_{\mathcal{M}}$, i.e. it is equal to the KL divergence between the output Gaussian distributions in figures 4.1b and 4.1c, $(\mathcal{D}(\mathbf{\Delta}||\mathbf{I}) = \mathcal{D}(\mathbf{\Sigma}||\mathbf{\Sigma}_{\mathcal{M}}))$.

Remark: In the rest of this chapter we consider the tree approximation as our model at each step of the cascade approximation.

4.3 Model Approximation Using Cascade of Tree Decompositions Principle

In this section, we focus on the cascade of trees framework for model selection using the tree decomposition transformations. We formulate the problem by considering the tree approximation as a transformation and we use multiple stages of these cascade trees to do model approximation. Let $\mathbf{\Sigma} \triangleq \mathbf{C}\mathbf{C}^T$ and $\mathbf{\Sigma}_{\mathcal{T}} \triangleq \mathbf{C}_{\mathcal{T}}\mathbf{C}_{\mathcal{T}}^T$ where $\mathbf{C}$ and $\mathbf{C}_{\mathcal{T}}$ are square transformation matrices that decompose the covariance matrices, $\mathbf{\Sigma}$ and $\mathbf{\Sigma}_{\mathcal{T}}$. Without loss of generality, in the rest of this chapter, we look at the zero-mean Gaussian distributions with normalized covariance matrix $\mathbf{\Sigma}$, i.e. covariance and correlation matrices are the same. Factoring covariances enable us to look at the problem as a transformation, as it is shown in figure 1. There are different decomposition algorithms to factor covariances such as the Cholesky decomposition and SVD. While we discuss the performance of the cascade of trees framework for model selection here, picking the decomposition algorithm will be discussed in section 4.4.

Definition 11. Symmetric correlation approximation matrix: *The symmetric correlation approximation matrix (CAM) for the tree approximation model is defined as* $\mathbf{\Delta} \triangleq \mathbf{C}_{\mathcal{T}}^{-1}\,\mathbf{\Sigma}\,\mathbf{C}_{\mathcal{T}}^{-T}$. ■

The symmetric CAM for each step of the cascade tree algorithm is also defined using the transformation matrix $\mathbf{C}_{\mathcal{T}_i}$ and the previous step symmetric CAM.

Definition 12. *The symmetric correlation approximation matrix for the i-th step of the cascade tree approximation is defined as* $\mathbf{\Delta}_i \triangleq \mathbf{C}_{\mathcal{T}_i}^{-1}\mathbf{\Delta}_{i-1}\mathbf{C}_{\mathcal{T}_i}^{-T}$ *where* $\mathbf{\Delta}_0 \triangleq \mathbf{\Sigma}$, $\mathbf{\Sigma}_{\mathcal{T}_i} = \textit{chowliu}(\mathbf{\Delta}_{i-1})$ *and* $\mathbf{\Sigma}_{\mathcal{T}_i} \triangleq \mathbf{C}_{\mathcal{T}_i}\mathbf{C}_{\mathcal{T}_i}^{T}$ *where* $\mathbf{C}_{\mathcal{T}_i}$ *is the decomposition for the i-the step covariance matrix,* $\mathbf{\Sigma}_{\mathcal{T}_i}$. ■

Figures 4.2a and 4.2b show schematic diagrams associated with the cascade tree framework. In figure 4.2a, we want to model the zero-mean multivariate Gaussian distribution, $\underline{X} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma})$, using the cascade of tree decomposition transformations. Let $\underline{X}_{\mathcal{T}_i} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\mathcal{T}_i})$ be the tree approximation distribution for the residue random vector $\underline{Z}_{i-1} \sim \mathcal{N}(\underline{0}, \mathbf{\Delta}_{i-1})$ [3] where $\mathbf{\Sigma}_{\mathcal{T}_i}$ is the tree approximation covariance matrix for $\mathbf{\Delta}_{i-1}$, i.e. $\mathbf{\Sigma}_{\mathcal{T}_i} = \text{chowliu}(\mathbf{\Delta}_{i-1})$. As shown in figures 4.2a, the i-stage cascade tree decomposition, transforms the zero-mean Gaussian random vector $\underline{Z}_i$ to the zero-mean Gaussian random vector $\underline{X}$.

Remark: For all $i \geqslant 1$, $\text{tr}\{\mathbf{\Delta}_i\} = n$. Trace of the CAM, $\mathbf{\Delta}_i$ is equal to n, since the covariance matrix $\mathbf{\Sigma}_{\mathcal{T}_i}$ at each iteration is obtained by the Chow-Liu algorithm and thus satisfies the covariance selection rules [1], i.e. $\text{tr}\{(\mathbf{\Delta}_{i-1} - \mathbf{\Sigma}_{\mathcal{T}_i})\mathbf{\Sigma}_{\mathcal{T}_i}^{-1}\} = 0$ and thus $\text{tr}\{\mathbf{\Delta}_{i-1}\mathbf{\Sigma}_{\mathcal{T}_i}^{-1}\} = n$.

In figure 4.2b we use the cascade tree decompositions to construct the approximation model. If we just use one tree, $\mathbf{C}_{\mathcal{T}_1}$ we have a tree approximation. For a cascade of l trees the approximation model is constructed using a backwards algorithm via the following cascade of linear tree approximations; $(\mathbf{C}_{\mathcal{T}_1}\mathbf{C}_{\mathcal{T}_2}\ldots\mathbf{C}_{\mathcal{T}_l})(\mathbf{C}_{\mathcal{T}_1}\mathbf{C}_{\mathcal{T}_2}\ldots\mathbf{C}_{\mathcal{T}_l})^T = \mathbf{\Sigma}_{\mathcal{M}_l}$. We also have following properties in lemma 4 and lemma 5.

Lemma 4. *Let* $\underline{W} \sim \mathcal{N}(\underline{0}, \mathbf{I})$, *then*

(a) $\mathcal{D}(f_{\underline{Z}_i}(\underline{z})||f_{\underline{W}}(\underline{w})) = \mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{X}_{\mathcal{T}_i}}(\underline{x}))$,

(b) $\mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{X}_{\mathcal{T}_i}}(\underline{x})) \leqslant \mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{W}}(\underline{w}))$,

where in (b) equality happens when $f_{\underline{Z}_{i-1}}(\underline{z}) = f_{\underline{W}}(\underline{w})$, *i.e* $\mathbf{\Delta}_{i-1} = \mathbf{I}$.

3. $\underline{Z}_0 \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma})$.

Proof. Proof of part (a) is based on the definition of KL divergence for jointly Gaussian distribution and $\text{tr}\{\mathbf{\Delta}_i\} = n$ as follow

$$\begin{aligned}\mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{X}_{\mathcal{T}_i}}(\underline{x})) &= -\frac{1}{2}\log(|\mathbf{\Delta}_{i-1}\mathbf{\Sigma}_{\mathcal{T}_i}^{-1}|) \\ &= -\frac{1}{2}\log(|\mathbf{C}_{\mathcal{T}_i}^{-1}\mathbf{\Delta}_{i-1}\mathbf{C}_{\mathcal{T}_i}^{-T}|) \\ &= -\frac{1}{2}\log(|\mathbf{\Delta}_i|) \\ &= \mathcal{D}(f_{\underline{Z}_i}(\underline{z})||f_{\underline{W}}(\underline{w})).\end{aligned}$$

Proof of part (b) follows from the KL divergence definition for Gaussian distributions and

$$\begin{aligned}\mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{W}}(\underline{w})) &= \mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{X}_{\mathcal{T}_i}}(\underline{x})) \\ &\quad + \mathcal{D}(f_{\underline{X}_{\mathcal{T}_i}}(\underline{x})||f_{\underline{W}}(\underline{w})) \\ &\geqslant \mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{X}_{\mathcal{T}_i}}(\underline{x})).\end{aligned}$$

Equality only happens if $|\mathbf{\Sigma}_{\mathcal{T}_i}| = 1$. Since $\text{tr}\{\mathbf{\Delta}_{i-1}\} = n$, then the covariance selection rule [1] dictates $\text{tr}\{\mathbf{\Sigma}_{\mathcal{T}_i}\} = n$, and thus the equality only happens if $\mathbf{\Delta}_{i-1} = \mathbf{I}$. ■

Lemma 4 states that the distribution of the i-th step residue random vector $\underline{Z}_i$ converges to the normal Gaussian random vector $\underline{W}$. Thus, in the cascade tree model approximation algorithm, we fix the number of cascade stages, l, and input the normal Gaussian random vector $\underline{W}$ to the cascade trees with l stages to do model approximation. The l-th step model covariance matrix approximation is

$$\mathbf{\Sigma}_{\mathcal{M}_l} = \mathbf{C}_{\mathcal{M}_l}\mathbf{C}_{\mathcal{M}_l}^T$$

where $\mathbf{C}_{\mathcal{M}_l} = \mathbf{C}_{\mathcal{T}_1}\mathbf{C}_{\mathcal{T}_2}...\mathbf{C}_{\mathcal{T}_i}...\mathbf{C}_{\mathcal{T}_l}$ is the model transformation. Note that, this is a backward construction (figure 4.2a).

Lemma 5. *KL divergence upper bound.*

$$\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}_i}}(\underline{x})) \leqslant \mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}_{i-1}}}(\underline{x}))$$

with equality only happens if $\mathbf{\Delta}_{i-1} = \mathbf{I}$.

Proof.

$$\begin{aligned}\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}_i}}(\underline{x})) &\overset{(a)}{=} \mathcal{D}(f_{\underline{Z}_i}(\underline{z})||f_{\underline{W}}(\underline{w})) \\ &\overset{(b)}{=} \mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{X}_{\mathcal{T}_i}}(\underline{x})) \\ &\overset{(c)}{\leqslant} \mathcal{D}(f_{\underline{Z}_{i-1}}(\underline{z})||f_{\underline{W}}(\underline{w})) \\ &\overset{(d)}{=} \mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}_{i-1}}}(\underline{x}))\end{aligned}$$

where (a) and (d) are because of the invariance of KL divergence between Gaussian distributions to the transformation; (b) and (c) follow from lemma 4. Equality in (c) holds if $f_{\underline{Z}_{i-1}}(\underline{z}) = f_{\underline{W}}(\underline{w})$. ■

Theorem 6. The Cascade tree decomposition transformation. *As the number of cascade trees, i, increases, the KL divergence between the distribution of $\underline{X}$ and the model distribution decreases, i.e. $\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}_i}}(\underline{x}))$ converges to a finite value.*

Proof. Proof follows directly from lemma 5 and positivity of KL divergence. ■

Conjecture 1. *The KL divergence converges to 0.*

Remark: Theorem 6 states that the KL divergence between the model approximation and the original distribution decreases as we add more stages of the cascade trees (Lemma 5) and we conjecture it will go to zero as the number of cascade trees goes to infinity. Note that, it is exactly equal to zero if at some iteration of the cascade tree $\mathbf{\Delta}_{i-1} = \mathbf{I}$.

4.4 Cascade Trees Algorithm

In order to use the cascade tree transformation decomposition framework presented in section 4.3, we need to pick a factorization scheme. Here we pick the Cholesky factorization. The main reason is that this scheme can preserve the sparsity pattern of the covariance matrix and thus is suitable to run message passing algorithms over factor graphs. Also, using the Cholesky decomposition, without loss of generality the diagonal coefficients of the symmetric CAM at all cascade stages are

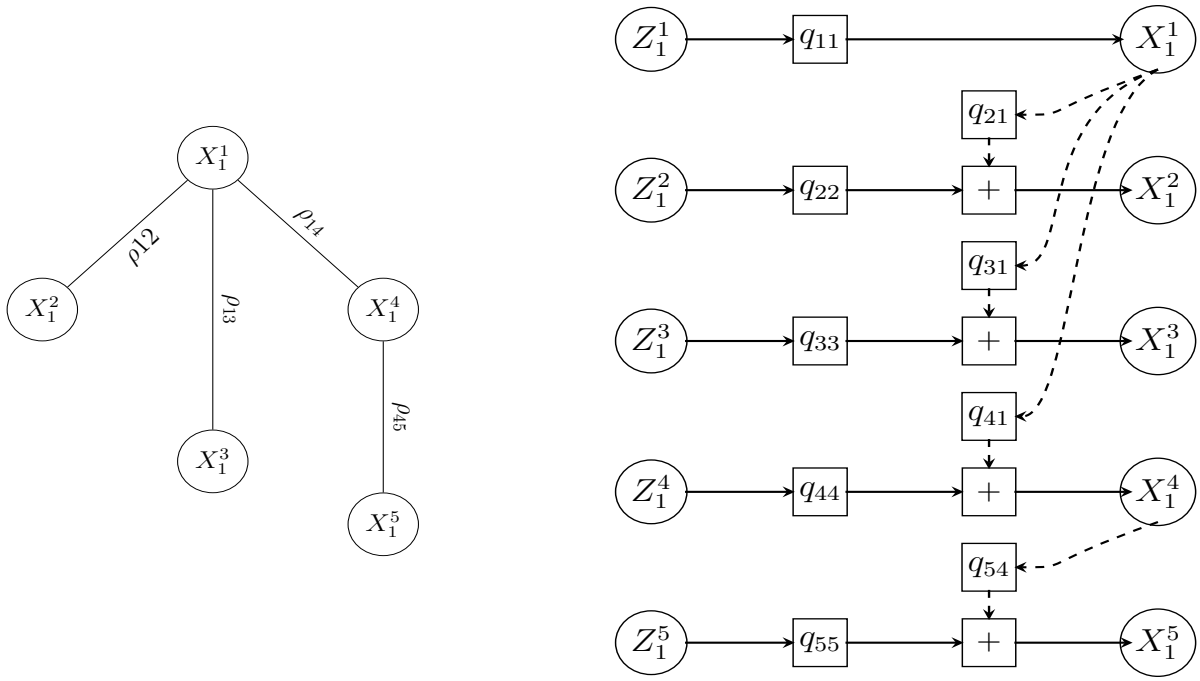

Figure 4.3: **Left:** Tree representation of the random vector $\underline{X}$ (ρ_{ij}'s are the correlation coefficients). **Right:** Factor graph representation with 5 nodes where $\mathbf{Q} = \mathbf{L}^{-1}$ and q_{ij}'s are the coefficients of the matrix $\mathbf{Q}$.

equal to one. Figure 4.3 shows a sample tree structured graph and its factor graph representation using the coefficients of the inverse of the Cholesky decomposition matrix, $\mathbf{Q}$.

4.4.1 Greedy Model Approximation Algorithm

Here we present a greedy algorithm based on the cascade trees principle. This algorithm consists of two general steps [66]:

- Finding the optimal Chow Liu tree,
- Performing the Cholesky decomposition such that it preserves the tree graph structure.

Given the symmetric CAM at each iteration of the greedy algorithm, we can efficiently find the optimal tree structure covariance matrix.

Theorem 7. *There exists a permutation matrix such that the inverse of the Cholesky decomposition preserves the sparsity pattern (position of zeros) of the inverse of the tree approximation covariance matrix [67].*

Proof. This lemma is a simplified version of the result presented in [67] where the Cholesky de-

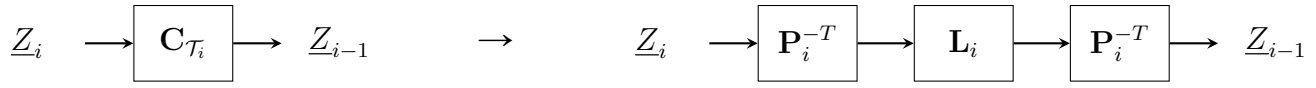

Figure 4.4a: **Left:** The i-th stage of the model transformation from $\underline{Z}_i$ to $\underline{Z}_{i-1}$. **Right:** The i-th stage of the model transformation from $\underline{Z}_i$ to $\underline{Z}_{i-1}$ using proper permutation matrix and the Cholesky decompositions.

Figure 4.4b: The l stages of cascade tree model approximation using the Cholesky decomposition with proper order (permutation) to keep the sparsity pattern in the inverse of The Cholesky decomposition. The model approximation is generated by passing $\underline{W} \sim \mathcal{N}(\underline{0}, \mathbf{I})$ through the l steps cascade trees and is $\underline{X}_{\mathcal{M}_l} \sim \mathcal{N}(\underline{0}, \mathbf{\Sigma}_{\mathcal{M}_l})$.

composition preserves the pattern of zeros corresponding to a co-chordal or homogeneous graph associated with a specific type of vertex ordering (permutation matrix). ■

Theorem 7 guarantees the existence of a loop-free factor graph based on the Cholesky decomposition coefficients of the inverse decomposition.

Figure 4.4a shows the schematic of the i-th stage of the model transformation based on the proper permutation of the Cholesky decomposition. To compute matrices $\mathbf{P}_i$ and $\mathbf{L}_i$ and reconstruct $\underline{Z}_{i-1}$ from $\underline{Z}_i$, we need to first compute the i-th stage tree approximation covariance matrix, $\mathbf{\Sigma}_{\mathcal{T}_i} = \text{chowliu}(\mathbf{\Delta}_{i-1})$. Next, we use the result of theorem 7 to find the proper re-order of the nodes. To do that, we look at the graph structure of the tree covariance matrix, $\mathbf{\Sigma}_{\mathcal{T}_i}$, and pick the nodes such that the graph associated with the subset of the picked nodes is always connected [4]. After that, we compute the Cholesky decomposition as $L_i = \text{Cholesky}(\mathbf{P}_i \mathbf{\Sigma}_{\mathcal{T}_i} \mathbf{P}_i^T)$ which has a sparse inverse. Next, we permute L_i to get the tree approximation transformation matrix, $\mathbf{C}_{\mathcal{T}_i}$. This process is shown in figure 4.4b. The model approximation covariance matrix after the l-th iteration is given as follow

$$\mathbf{\Sigma}_{\mathcal{M}_l} = \mathbf{C}_{\mathcal{M}_l} \mathbf{C}_{\mathcal{M}_l}^T$$

where $\mathbf{C}_{\mathcal{M}_l} = \mathbf{P}_1^{-1}\mathbf{L}_1\mathbf{P}_1^{-T}\mathbf{P}_2^{-1}\mathbf{L}_2\mathbf{P}_2^{-T}\dots\mathbf{P}_l^{-1}\mathbf{L}_l\mathbf{P}_l^{-T}$.

Using the Cholesky decomposition with the proper permutation matrix at each iteration enables us

4. If at any step of the algorithm, the tree graph structure associated with $\mathbf{\Sigma}_{\mathcal{T}_i}$ becomes disconnected, we will seek the same procedure for each of the disjoint segments of the graph.

to draw loop-free factor graph at each iteration of the cascade trees' framework. The factor graph representation is useful in order to run message passing algorithm and loopy Gaussian BP over the overall loopy factor graph. The greedy algorithm based on the Cholesky decomposition presented in figure 4.4b is as follow:

Algorithm 4.1: Greedy Model Approximation Algorithm using Cascade Trees' Framework and the Cholesky Decomposition

- Initialization Step [$i = 0$]:
  - $\mathbf{\Delta}_0 = \mathbf{\Sigma}$
- Continue updating [i-th Step]:
  - $i \leftarrow i + 1$
  - $\mathbf{\Sigma}_{\mathcal{T}_i} = \text{chowliu}(\mathbf{\Delta}_{i-1})$
  - Given $\mathbf{\Sigma}_{\mathcal{T}_i}$, compute the proper node ordering and construct the permutation matrix, $\mathbf{P}_i$.
  - $L_i = \text{Cholesky}(\mathbf{P}_i \mathbf{\Sigma}_{\mathcal{T}_i} \mathbf{P}_i^T)$
  - $\mathbf{C}_{\mathcal{T}_i} = \mathbf{P}_i^{-1} \mathbf{L}_i \mathbf{P}_i^{-T}$
  - $\mathbf{\Delta}_i = \mathbf{C}_{\mathcal{T}_i}^{-1} \mathbf{\Delta}_{i-1} \mathbf{C}_{\mathcal{T}_i}^{-T}$
- Stopping criterion: $i \leqslant l$
- Output $\mathbf{L}_i$'s and $\mathbf{P}_i$'s as well as the approximated model covariance matrix $\mathbf{\Sigma}_{\mathcal{M}_l} = \mathbf{C}_{\mathcal{M}_l} \mathbf{C}_{\mathcal{M}_l}^T$ where $\mathbf{C}_{\mathcal{M}_l} = \mathbf{C}_{\mathcal{T}_1} \dots \mathbf{C}_{\mathcal{T}_l}$ for some i satisfying the stopping criterion.

Remark: We can stop the algorithm sooner than we reach the maximum numbers of cascade tress if for some $i < l$ the KL divergence goal is satisfied, i.e. $\mathcal{D}(f_{\underline{Z}_i}(\underline{z}) || f_{\underline{W}}(\underline{w})) \leqslant \delta$ where δ is the maximum KL divergence between the original distribution and the approximated model distribution. [5]

Remark: In any step of the algorithm, if the graph correspond to the $\mathbf{\Delta}_i$ become disconnected, we will do tree approximation for each of the connected subgraphs.

Theorem 8. *There exists a cascade tree approximation algorithm to generate the model approximation $\mathcal{M}_i'$ such that after at most $n-1$ iteration, the model approximation error (KL divergence)*

5. Parallel computing algorithms can be useful for implementation of tree algorithms [68–71].

is exactly equal to zero, i.e. $\mathbf{\Sigma} = \mathbf{\Sigma}_{\mathcal{M}_{n-1}}$. *In other words,*

$$\mathcal{D}(f_{\underline{X}}(\underline{x})||f_{\underline{X}_{\mathcal{M}'_{n-1}}}(\underline{x})) = 0.$$

Proof. We proof by construction. We use the star approximation (graph with the star node having n-1 edges and all other nodes connected just to the star node), at each iteration of the cascade tree approximation algorithm. Moreover, we use the Cholesky decomposition to keep the sparsity pattern in the inverse of the Cholesky decomposition. Formal proof is given in appendix A.3. ■

Theorem 9. Diagonal Coefficients of the Symmetric CAM. *Diagonal coefficients of the symmetric CAM at each step of the the greedy model approximation algorithm using the Cholesky factorization, are equal to one.*

Proof. Proof is given in appendix A.4. ■

Theorem 9 shows that if we pick the Cholesky factorization and we follow the greedy model approximation algorithm presented here, then diagonal coefficients of the approximated matrix is always the same as diagonal coefficients of the covariance matrix, $\mathbf{\Sigma}$, i.e. the proposed algorithm preserves variances.

4.4.2 Complexity of the cascade tree algorithm

The Chow-Liu algorithm has the running time complexity of $\mathcal{O}(n^2 \log(n))$ while the Cholesky decomposition complexity is $\mathcal{O}(n^3)$. Thus, the overall complexity of the algorithm is $\mathcal{O}(l\, n^3)$ since we run the algorithm for at most l cascade stages. Moreover, the maximum number of edges in the resulting factor graph is ln.

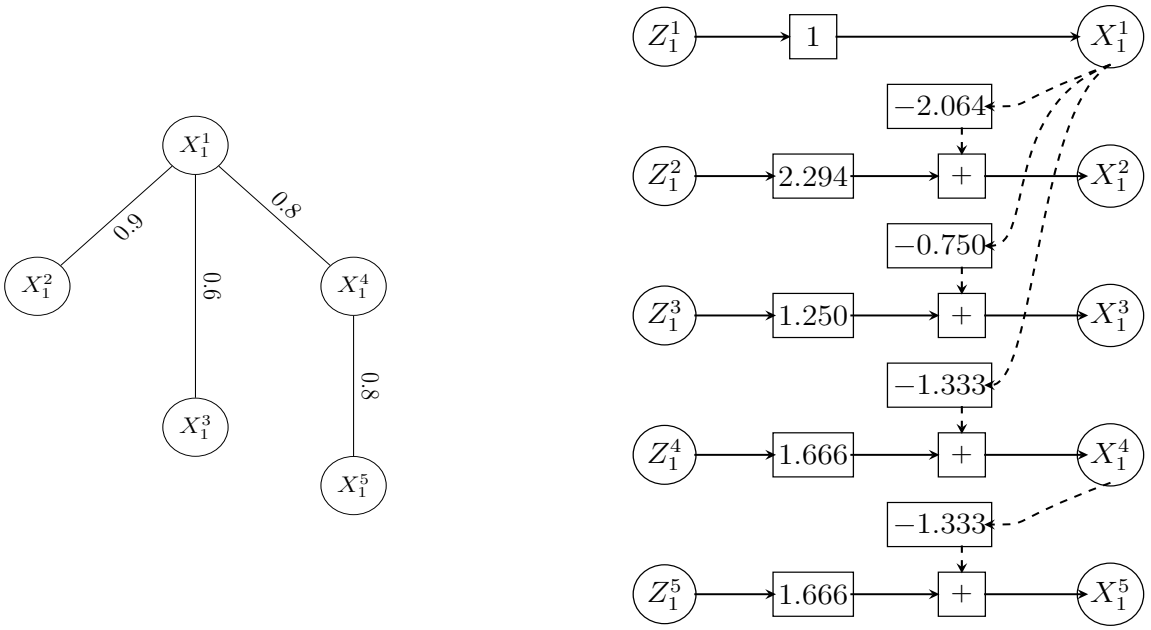

Figure 4.5: First stage cascade tree representation for the 5 nodes example and its Factor graph representation.

4.4.3 Example with 5 nodes

In this example, we start with a zero mean Gaussian distribution with covariance matrix $\mathbf{\Sigma}$ (or $\mathbf{\Delta}_0$) for random vector $\underline{X}$ as follows

$$\mathbf{\Sigma} = \begin{bmatrix} 1 & 0.9 & 0.6 & 0.8 & 0.7 \\ 0.9 & 1 & 0.5 & 0.6 & 0.6 \\ 0.6 & 0.5 & 1 & 0.4 & 0.1 \\ 0.8 & 0.6 & 0.4 & 1 & 0.8 \\ 0.7 & 0.6 & 0.1 & 0.8 & 1 \end{bmatrix}.$$

We want to approximate the random vector $\underline{X}$ with 2 stages of cascade trees as $\underline{X}_{\mathcal{M}}$. First step of the tree approximation covariance matrix $\mathbf{T}_1$ is

$$\mathbf{T}_1 = \begin{bmatrix} 1 & 0.9 & 0.6 & 0.8 & 0.64 \\ 0.9 & 1 & 0.54 & 0.72 & 0.576 \\ 0.6 & 0.54 & 1 & 0.48 & 0.374 \\ 0.8 & 0.72 & 0.48 & 1 & 0.8 \\ 0.64 & 0.576 & 0.374 & 0.8 & 1 \end{bmatrix}$$

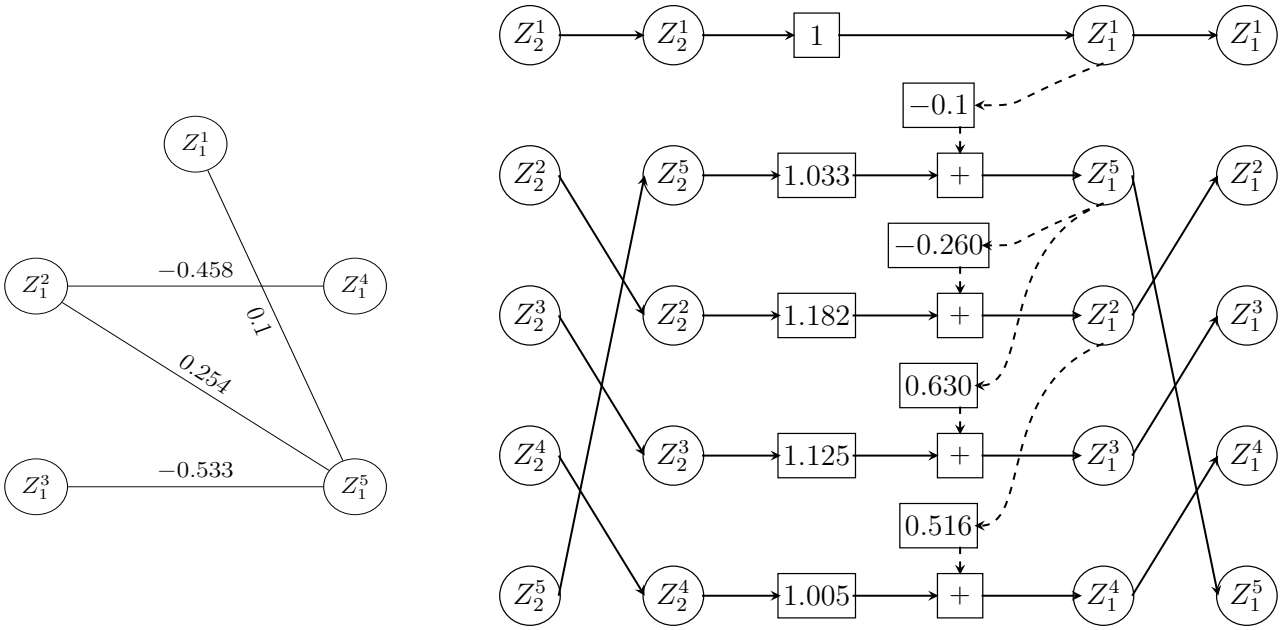

Figure 4.6: Second stage cascade tree representation for the 5 nodes example and its Factor graph representation.

while its Cholesky decomposition inverse, $\mathbf{Q}_1$ is

$$\mathbf{Q}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -2.064 & 2.294 & 0 & 0 & 0 \\ -0.75 & 0 & 1.25 & 0 & 0 \\ -1.333 & 0 & 0 & 1.666 & 0 \\ 0 & 0 & 0 & -1.333 & 1.666 \end{bmatrix},$$

and the permutation matrix, $\mathbf{P}_1$ is identity. To proceed to the second stage, we first compute the symmetric CAM, $\mathbf{\Delta}_1$ as

$$\mathbf{\Delta}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0.1 \\ 0 & 1 & -0.1147 & -0.458 & 0.252 \\ 0 & -0.114 & 1 & -0.166 & 0.374 \\ 0 & -0.458 & -0.458 & 1 & -0.133 \\ 0.1 & 0.252 & 0.252 & -0.133 & 1 \end{bmatrix}.$$

The CAM matrix, $\mathbf{\Delta}_1$, is the covariance matrix of the residue random vector, $\underline{Z}_1$, and $\underline{Z}_1 = \mathbf{Q}_1\underline{X}$ or equivalently $\underline{X} = \mathbf{C}_{\mathcal{T}_1}\underline{Z}_1$ where $\mathbf{C}_{\mathcal{T}_1} = \mathbf{Q}_1^{-1}$. Also, the KL divergence for the first step is

$\mathcal{D}(\underline{X}||\underline{X}_{\mathcal{M}_1}) = 0.375$. Then, the second step of the tree approximation covariance matrix $\mathbf{T}_2$ is

$$\mathbf{T}_2 = \begin{bmatrix} 1 & 0.025 & -0.053 & -0.011 & 0.1 \\ 0.025 & 1 & -0.134 & -0.458 & 0.252 \\ -0.053 & -0.134 & 1 & 0.061 & -0.533 \\ -0.011 & -0.458 & 0.061 & 1 & -0.115 \\ 0.1 & 0.252 & -0.533 & -0.115 & 1 \end{bmatrix}$$

while its Cholesky decomposition inverse, $\mathbf{Q}_2$, which is computed using

$$\mathbf{Q}_2 = \mathbf{P}_2^T \,\text{Cholesky}(\mathbf{P}_2\mathbf{T}_2\mathbf{P}_2^T)^{-1}\mathbf{P}_2$$

is as follow

$$\mathbf{Q}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1.033 & 0 & 0 & -0.260 \\ 0 & 0 & 1.182 & 0 & 0.630 \\ 0 & 0.516 & 0 & 1.125 & 0 \\ -0.1 & 0 & 0 & 0 & 1.005 \end{bmatrix},$$

where the permutation matrix, $\mathbf{P}_2$, is

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Furthermore, the residue random vectors, $\underline{Z}_1$ and $\underline{Z}_2$, have the following relationship $\underline{Z}_2 = \mathbf{Q}_2\underline{Z}_1$ or equivalently $\underline{Z}_1 = \mathbf{C}_{\mathcal{T}_2}\underline{Z}_2$ where $\mathbf{C}_{\mathcal{T}_2} = \mathbf{Q}_2^{-1}$. To approximate the model random vector $\underline{X}_{\mathcal{M}_2}$ using two stage of the cascade tree, we replace the second residue random vector, $\underline{Z}_2$, with the random vector $\underline{W}$. Also, the KL divergence for the second step is $\mathcal{D}(\underline{X}||\underline{X}_{\mathcal{M}_2}) = 0.051$. Figure 4.5 shows the Chow-Liu tree and the factor graph representation for of it and figure 4.6 shows the second stage of the algorithm. Since the permutation matrix is not identity in the second step of the algorithm, we need to change the ordering as it is shown in figure 4.6. For this example a cascade of two trees

produces a linear transformation that approximates the Gaussian vector $\underline{X}$ closely.

4.5 Simulation Results and Discussion

In this section, we consider some examples of covariance matrices for a Gaussian random vector $\underline{X}$. We present some simulation results on both synthetically generated covariance matrices and the covariance matrix generated from the island of Oahu real solar dataset. We also present the simulation results on the performance of the cascade tree decomposition transformation framework using different factorization methods. We take a special look at the performance of the presented algorithm in section 4.4 which is based on the Cholesky decomposition and the proper permutation to keep the sparsity pattern in the inverse of the Cholesky factorization. We also look at the performance of the singular value decomposition and the Cholesky factorization without the proper permutation (does not keep the sparsity pattern). Looking at other covariance matrices factorization methods gives some insight on how good is the performance of the greedy algorithm which is presented in section 4.4.

Remark: In all simulation results we only consider 16 digits precision after the floating point.

4.5.1 Synthetic data

We randomly generate synthetic covariance matrix with 250 nodes such that its graphical structure has about half of all possible edges and then we normalize it to have ones along the diagonal.

Figure 4.7 plots the gray scaled, sparsity pattern for the inverse of a randomly generated, synthetic covariance matrix and various approximations of it. The top left plot shows the inverse of the original normalized covariance matrix. The graph associated with this inverse covariance matrix has around $\frac{n^2}{2}$ number of edges. The bottom left plot indicates the first stage of the cascade tree approximation (the optimal Chow-Liu solution) or the approximated model after the first stage of the greedy algorithm. The top middle plot shows the inverse of the second approximated model while the plot on the bottom middle indicates the inverse of the second stage tree approximation. The plot on the top right indicates the sparsity pattern of the inverse of third approximated model,

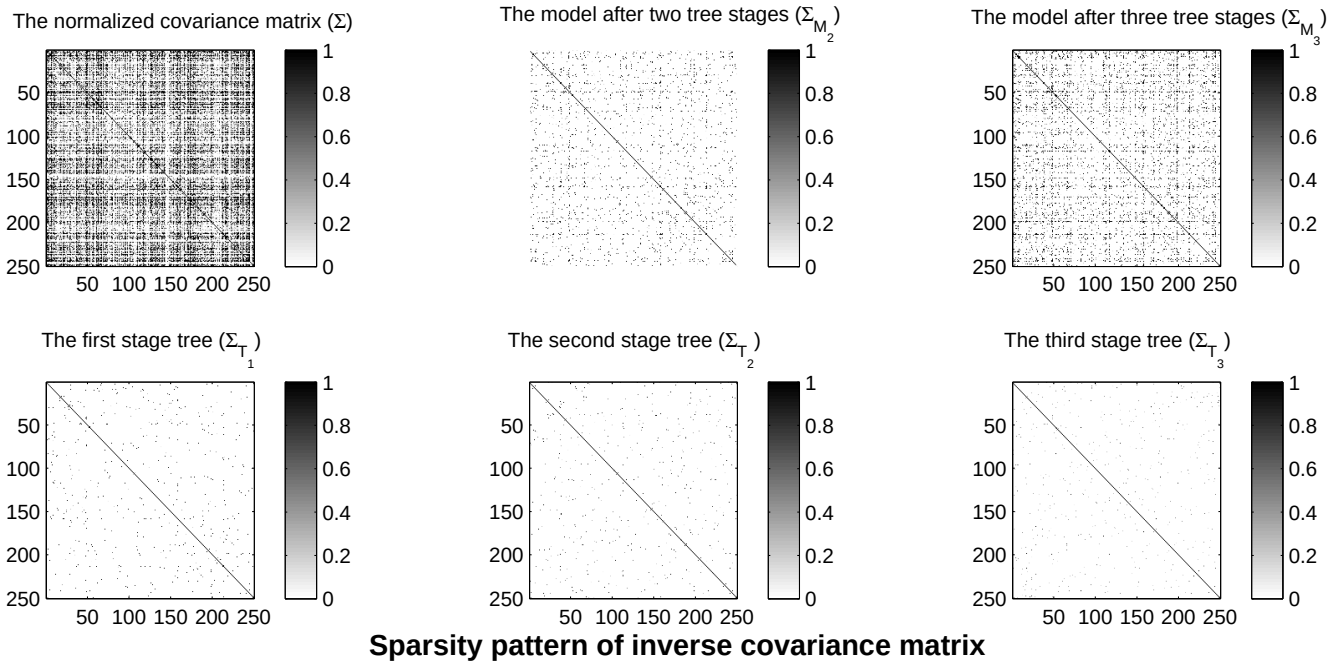

Figure 4.7: Gray scaled, sparsity pattern for the inverse of a randomly generated, synthetic covariance matrix. **Top left:** Inverse of the original normalized covariance matrix, **Bottom left:** Inverse of the first stage tree approximation and first model. **Top middle:** Inverse of the second approximated model, **Bottom middle:** Inverse of the second stage tree approximation. **Top right:** Inverse of the third approximated model, **Bottom right:** Inverse of the third stage tree approximation.

while the bottom right plot shows the third stage tree approximation. In Figure 4.7 the Chow-Liu tree approximation shown in the bottom left is a poor approximation of the top left plot by comparing the two gray-scale plots. The top middle plot is a better approximation of the top left plot and the top right plot provides the best approximation to the top left plot. The top right plot consists of the cascade of three trees having a maximum of 3×249 edges as compared to the top left plot which represents a graph with more than 30000 edges.

Figure 4.8 plots the log-scaled KL divergence between the random vector $\underline{X}$ and the approximation model vector $\underline{X}_{\mathcal{M}}$ after the i-th step of the cascade trees approximation with respect to the number of cascade trees transformation that are used in the approximation, i. This figure plots the result of the cascade trees decomposition algorithm for the performance of three different tree structures, the optimal Chow liu tree, the Star tree without permutation, and the optimal star tree with permutation, as we add more cascade steps. The left plot compares the performance of the cascade trees approximation for different choices of tree structures after 10 steps of the cascade

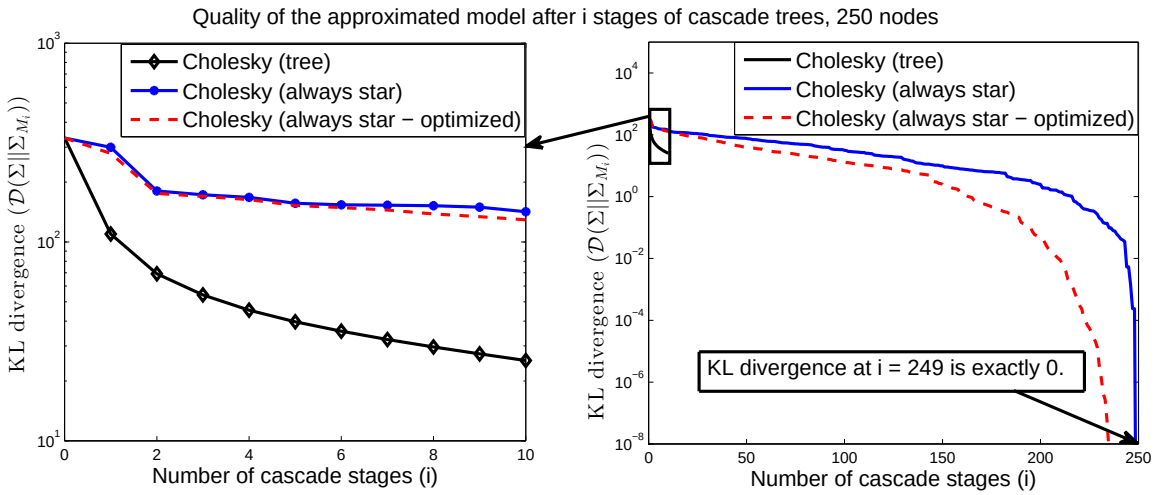

Figure 4.8: KL divergence between the distribution of the random vector $\underline{X}$ and the model distribution after the i-th step of the cascade approximation v.s. the index of the cascade trees, i, for a graph with 250 nodes. Chow-Liu algorithm is used at each iteration of the cascade approximation. **Right:** Comparing the performance of three different tree structures, the optimal Chow liu tree, the Star tree without permutation, and the optimal star tree with permutation, as we add more cascade steps. **left:** Zoomed into 10 cascade trees decompositions.

trees decompositions, while the right plot runs the cascade trees algorithm for 249 steps. Looking only at the KL divergence we can easily see that using the greedy algorithm presented in section 4.4 clearly has a better performance when we only have small number of cascade stages. On the other hand, running the cascade tree framework using the star tree approximation at each stage for 249 stages, the KL divergence goes to zero. Note that, figure 4.9 plots the KL divergence in linear scale. If we compare the Chow-Liu tree to a cascade of two trees/ three trees the KL divergence decreases by respectively 35%/ 50% (figure 4.9).

Remark: In the i-th iteration of the always star approximation we picked node i as the star node to do the approximation without any optimization, i.e. identity permutation matrix.

Figure 4.10 plots the KL divergence between the random vector $\underline{X}$ and the approximation model vector $\underline{X}_{\mathcal{M}}$ after the i-th step of the cascade trees approximation with respect to the number of cascade trees transformation that are used in the approximation, i, for a graph of 100 nodes. This figure plots the result of the cascade trees framework with different decompositions such as the Cholesky $\mathbf{L}\mathbf{L}^T$ (keep the sparsity), the Cholesky $\mathbf{U}\mathbf{U}^T$ (does not keep the sparsity) and the SVD. From figure 4.10 we see that three of the decomposition transformations perform similarly with the star decomposition transformation performing the worse.

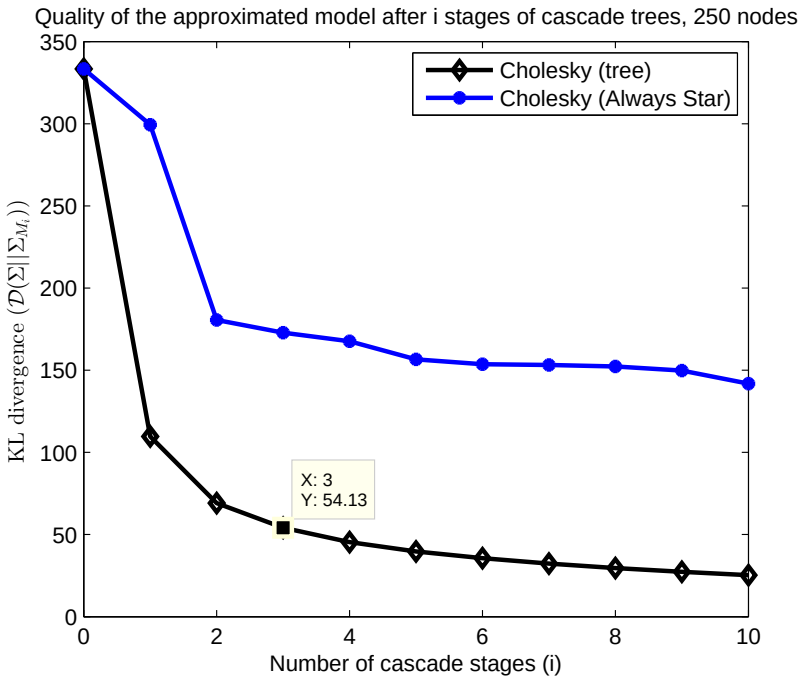

Figure 4.9: KL divergence between the distribution of the random vector $\underline{X}$ and the model distribution after the i-th step of the cascade approximation v.s. the index of the cascade trees, i, for a graph with 250 nodes.

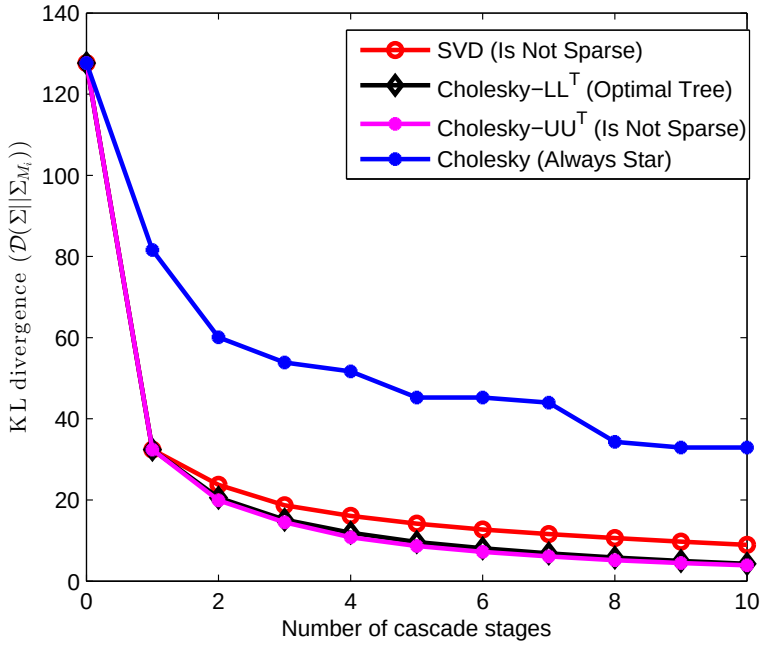

Figure 4.10: KL divergence between the distribution of the random vector $\underline{X}$ and the model distribution after the i-th step of the cascade approximation v.s. the index of the cascade trees, i for a graph with 100 nodes using different decompositions. Chow-Liu algorithm is used at each iteration of the cascade approximation.

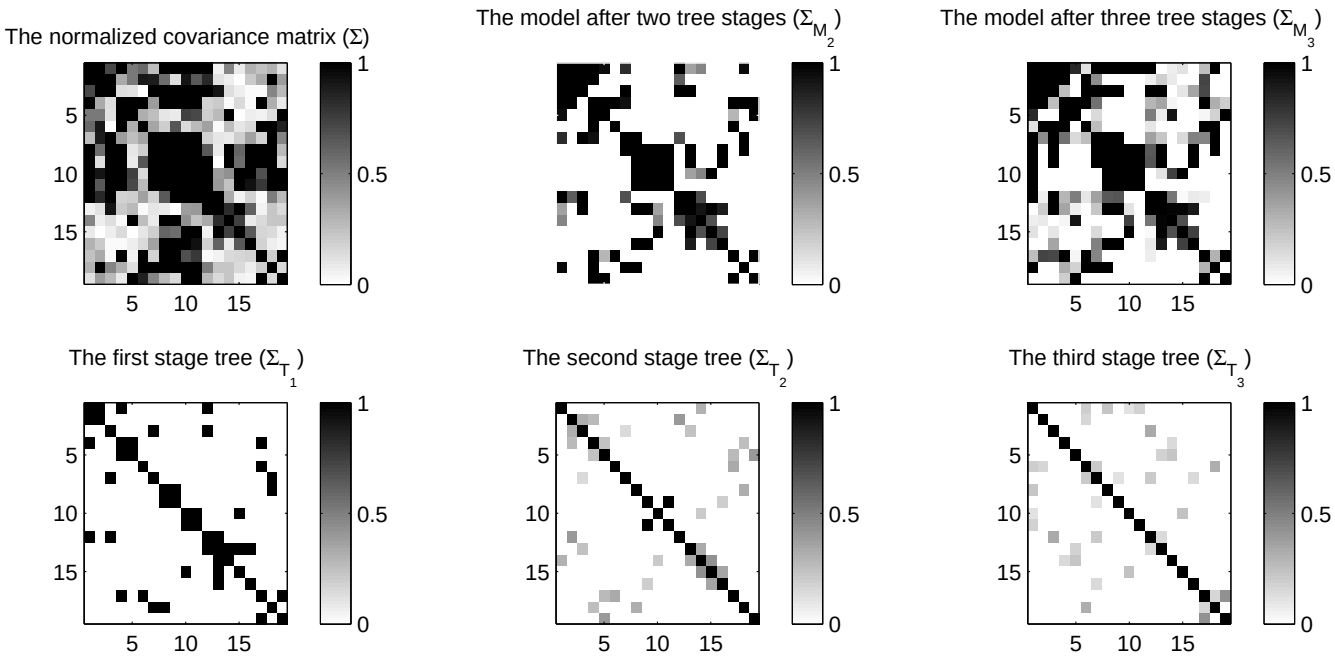

Figure 4.11: Gray scaled, sparsity pattern for the inverse of the covariance matrix generated using the Oahu solar measurement grid dataset. **Top left:** Original normalized covariance matrix, **Bottom left:** first stage tree approximation and first model. **Top middle:** second approximated model, **Bottom middle:** second stage tree approximation. **Top right:** third approximated model, **Bottom right:** third stage tree approximation.

4.5.2 The Oahu solar measurement grid dataset

In this Example, the covariance matrix is calculated based on datasets presented in [2]. The Oahu solar measurement grid dataset is obtained from the National Renewable Energy Laboratory (NREL) website [55]. This dataset consists of 19 sensors (17 horizontal sensors and two tilted sensors). For this dataset we normalized using standard normalization method and the zenith angle normalization method [2] [6]. From the data obtained from these 19 solar sensors at the island of Oahu, we computed the spatial covariance matrix during the summer season at 12:00 PM averaged over a window of 5 minutes.

6. See [2] for more detailed description of dataset and other details about the normalization methods for the solar irradiation covariance matrix.

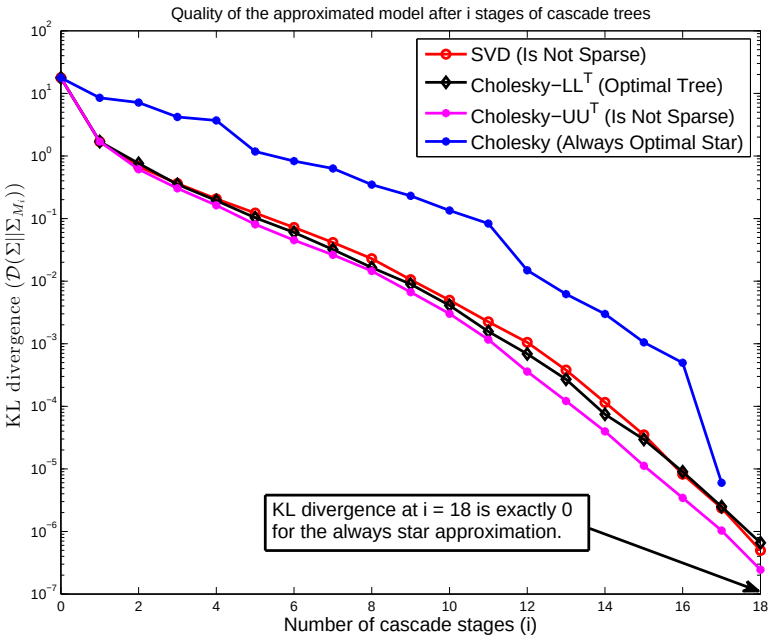

Figure 4.12: KL divergence between the distribution of the random vector $\underline{X}$ and the model distribution after the i-th step of the cascade approximation v.s. the index of the cascade trees, i, for the island of Oahu solar data using different decompositions.

Figure 4.11 plots the grayscaled, sparsity pattern for the inverse of the Oahu solar measurement grid covariance matrix and various approximations of it. The top left plot shows the inverse of the original normalized covariance matrix while the bottom left plot indicates the first stage of the cascade tree approximation or the optimal Chow-Liu approximated model. The top middle plot shows the inverse of the second approximated model while the plot on the bottom middle indicates the inverse of the second stage tree approximation. The plot on the top right indicates the sparsity pattern of the inverse of the third approximated model, while the bottom right plot shows the third stage tree approximation. Note that the sparsity pattern of the third approximation is close to original correlation matrix.

Figure 4.12 plots the log-scaled KL divergence between the distribution of the random vector $\underline{X}$ and the distribution of the model distribution after the i-th step of the cascade trees approximation with respect to the number of cascade trees transformation that are used in the approximation, i. This figure compares the performance of the proposed cascade trees approximation with different decomposition choices with the optimal star tree approximation. Looking only at the KL divergence we can easily see that using the greedy algorithm presented in section 4.4 clearly has a better per-

formance when we only have a small number of cascade stages compared to the star tree structure. On the other hand, running the cascade tree framework using the star tree approximation at each stage for 18 stages, the KL divergence goes the zero. This figure also plots the result of the cascade trees framework with different decompositions such as the Cholesky $\mathbf{L}\mathbf{L}^T$ (keep the sparsity), the Cholesky $\mathbf{U}\mathbf{U}^T$ (does not keep the sparsity) and the SVD. From figure 4.12 we see that three of the decomposition transformations perform similarly with the star decomposition transformation performing the worse. If we compare the Chow-Liu tree to a cascade of two trees/ three trees the KL divergence decreases by respectively more than 50%/ 80%. More precisely, the KL divergence after one, two and three cascade stages is equal to 1.695, 0.7615 and 0.3507. By using the Chow-Liu algorithm to produce trees and then using the Cholesky factorization in general, this algorithm performs well as the KL divergence decreases relatively quickly. However, by using the star network systematically on all nodes except one we can guarantee that the cascade algorithm converges to the model after $n-1$ steps.

Remark: This greedy algorithm is a new way to decompose covariance matrices. We want to use the Chow-Liu algorithm since it results in the KL divergence initially decaying faster. However, if we use a star network, the KL divergence goes to zero after at most $n-1$ steps.

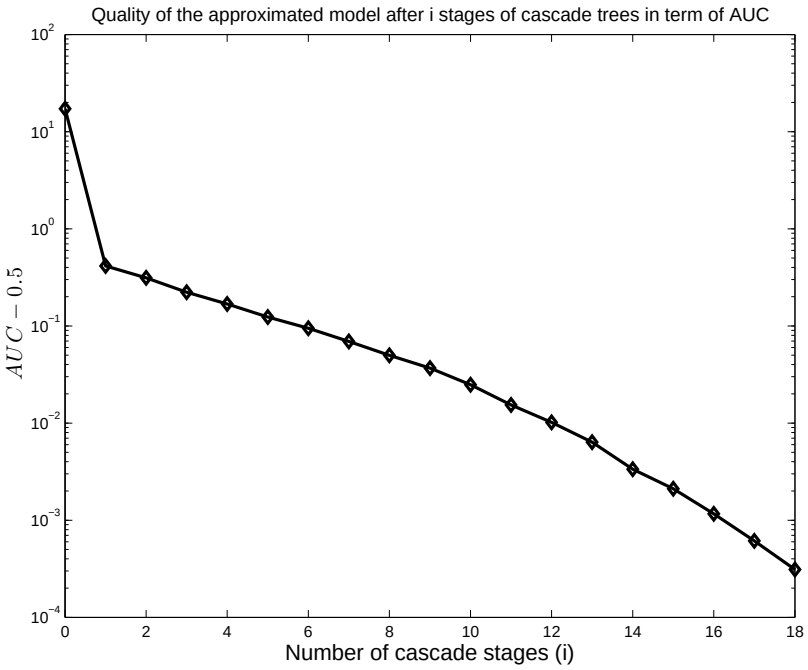

Figure 4.13: (AUC -0.5) between the distribution of the random vector $\underline{X}$ and the model distribution after the i-th step of the cascade approximation v.s. the index of the cascade trees, i, for the island of Oahu solar data using different decompositions.

Figure 4.13 plots the log-scaled (AUC -0.5) between the distribution of the random vector $\underline{X}$ and the distribution of the model distribution after the i-th step of the cascade trees approximation with respect to the number of cascade trees transformation that are used in the approximation, i. This figure plots the result of the cascade trees framework using the Cholesky decompositions $\mathbf{L}\mathbf{L}^T$ which keep the sparsity. As it is shown in this figure, (AUC -0.5) is also decreasing to 0 similar to KL divergence.

4.6 Conclusion

In this chapter, we look at the graphical model as a transformation and introduce a general framework to do model approximation for graphical models. This new framework, which we call the cascade trees framework, approximates a complex, hard to compute model with a cascade of simpler, more efficient tree models, that can be easily computed. To compute the optimal tree approximation at each stage of the cascade trees framework, we used the Chow-Liu algorithm. In

the computation of cascade tree framework we look at the best possible decomposition methods and we defined an important quantity, the symmetric CAM. The symmetric CAM allows us to use this cascade tree framework by creating a residual correlation matrix at each step. The residual correlation matrix can be viewed as the remaining part of the original correlation matrix not approximated by previous iterations. Here we used a backward construction method. For the proposed cascade trees algorithm, the algorithm we picked uses the Cholesky lower diagonal decomposition of the covariance matrix. This choice of decomposition is favorable since it preserves the sparsity pattern of the inverse covariance matrix. We present results that guarantee the convergence of the proposed model approximation using the cascade of tree decompositions. We confirm those results using the examples provided in the simulation section where we look at synthetic and real data and compare the performance of the proposed framework by comparing KL divergences.

In future research, a more generalized case than cascade of tree models can be considered where instead of tree approximation at each iteration we look at simple, easy to compute non-tree models, such as ring models. Here we focused on a backward cascade model. We are currently also considering forward cascade models. We are also looking more deeply at the convergence of these cascade tree algorithms.

5

Conclusions and Future Directions

5.1 Conclusion

In this concluding chapter, we first review the contributions of this dissertation to the following two related problems in graphical modeling: the quality of the statistical graphical models, and a systematic way to construct tractable algorithms for model approximation. Then, we identify a number of directions for further research.

In the first problem, we studied the quality of the statistical graphical models by quantifying its quality by setting up a parametric detection problem while in the second problem, we targeted a systematic way to construct tractable algorithms for model approximation by formulating a general framework where we developed a multistage framework for graphical model approximation using a cascade of models such as trees.

In chapter 2, we discussed the theory behind our proposed method of quantifying the quality of the approximation model. Instead of using common distance measure such as the KL divergence, we extended the body of research by formulating the model approximation as a parametric detection problem between the original distribution and the model distribution. The proposed detection framework resulted in the computation of symmetric closeness measures such as ROC and AUC. In the second part of chapter 2, we focused on Gaussian distributions and the covariance selection quality. We showed that closeness measures such as KL divergence, reverse KL divergence, ROC, and AUC are all depended on the eigenvalues of the CAM. Besides that, we presented theoretical expressions for the KL divergence, the log-likelihood ratio, and the AUC as a function of the CAM. We also presented a simple, computationally efficient, and complex-valued integral to calculate the AUC. In addition, easily computable upper and lower bounds are also found for the AUC to assess the quality of an approximated model.

In chapter 3, we looked at the quality of different model approximation methods through some examples and simulations for real and synthetic data. In the first part of this chapter, we investigated the quality of tree structured models while in the second part, we switched to more complex, non-tree structured models. In the first part, we used the Chow-Liu MST algorithm to compute the maximum likelihood tree structured approximation. Then, the quality of this tree algorithm was investigated using the proposed framework in chapter 2. We saw through some examples that in general, the tree approximation model is not a good model as the number of vertices in the graphical model increases which is the case in high-dimensional problems. One such example is modeling the electrical distribution grid using smart grid sensor measurements and distributed renewable energy sources. The aforementioned result is also consistent with the analytical results provided for Toeplitz example which shows that $1-\text{AUC}$ decays exponentially as the number of graph vertices increases. In the second part, we looked at non-tree graphical models where we discuss graphical models with junction tree structures such as the p-th order Markov chain and the corresponding star network interpretation for a special Toeplitz covariance matrix with ones along the diagonal and correlation coefficient ρ's on the off-diagonals. These models have very short loops and have associated junction tree that connects cliques of the same size. We computed the model covariance matrix and the KL divergence and also quantified the goodness of the approximated covariance matrix. In this example, we showed that if the model order, p, is proportional to the

number of vertices, n, then the approximated model is asymptotically good as $n \to \infty$ since the AUC is asymptotically bounded away from one. We confirmed this theoretical result by performing some simulations which showed that the selected model quality increases as the model order, p, increases. We also provided necessary criteria on the model order, p, in order to bound the AUC away from one.

In chapter 4, we formulated a general framework and algorithms for model approximation. We developed a multistage framework for graphical model approximation using a cascade of models such as trees. In particular, we looked at the model approximation problem for Gaussian distributions as linear transformations of tree models which is a new way to decompose the covariance matrix. The proposed algorithm in this chapter incorporates the Cholesky factorization method to compute the decomposition matrix and thus can approximate a simple graphical model using a cascade of the Cholesky factorization of the tree approximation transformations. The implementation of the Cholesky decomposition enables us to achieve a tree structure factor graph at each cascade stage of the algorithm which facilitates the use of the message passing algorithm since the approximated graph at each stage has fewer loops compared to the original graph. The overall graph is a cascade of factor graphs with each factor graph being a tree. This is a different perspective on the approximation model, and algorithms such as Gaussian belief propagation can be used on this overall graph. In this chapter, we also presented theoretical results that guarantee the convergence of the proposed model approximation using the cascade of tree decompositions. This is a backward construction method. In the simulations, we looked at synthetic and real data and measured the performance of the proposed framework by comparing the KL divergences.

5.2 Future Possible Research Directions

Graphical models is an effective approach to cope with modeling complexity using graph theory. In future research by looking at the first problem presented in this dissertation, we can look at other applications of the theoretical foundation to other problems in literature. For instance, the parametric detection framework presented in chapter 2 can be generalized for non-Gaussian models. The AUC analytical bounds obtained in this chapter can also be used in other applications that are using AUC as a relevant criterion. One example is in medicine when the AUC is used for diagnostic

tests between positive instance and negative instance [54] where instead of changing the coordinates we can look at the exponent of the AUC bounds.

Regarding the second problem addressed in this dissertation, in future research, a more generalized case than a cascade of tree models can be considered where instead of tree approximation at each iteration we look at simple, easy to compute non-tree models, such as ring models. Here we focused on a backward cascade model. We can also consider forward cascade models. We can also look more deeply at the convergence of these cascade tree algorithms. Moreover, we can look at the models, decompositions at each iteration of cascade approximation that makes the overall approximation to converge in finite steps (similar to the star model).

Results found in this dissertation can add to modeling distributed energy resources in the power grid. One can use presented model approximation techniques in order to decrease the number of loops due to penetration of correlated renewable sources.

A

Appendices

A.1 Proof of Lemma 1

The calculus based proof for the special case of continuous PDFs is as follow. We can apply the Leibniz integral rule [72] and compute the derivative of CDFs $P_0(l)$ and $P_1(l)$ as

$$f_{L_0}(l) = -\frac{dP_0(l)}{dl}$$

and

$$f_{L_1}(l) = -\frac{dP_1(l)}{dl}$$

since $f_{L_0}(l)$ and $f_{L_1}(l)$ are continuous functions.[1] We have

$$\begin{aligned}\mathcal{D}\left(f_{L_0}(l)||f_{L_1}(l)\right) &= \int_{-\infty}^{+\infty} \log \frac{f_{L_0}(l)}{f_{L_1}(l)}\, f_{L_0}(l)\, dl \\ &\overset{(a)}{=} -\int_0^1 \log \frac{dP_1}{dP_0}\, dP_0 \\ &\overset{(b)}{=} -\int_0^1 \log h'(z)\, dz\end{aligned}$$

where equality (a) is true since we can replace PDfs $f_{L_0}(l)$ and $f_{L_1}(l)$ using the derivative of their CDFs. Equality (b) is just a change of variable, $z = P_0(l)$, in order to write the integral in terms of the derivative of the ROC curve. Proof for the second part of this lemma is similar to the proof of the first part. ■

A.2 Proof of Theorem 3

Looking back at properties of the ROC curve, $h(z)$, where $z \in [0, 1]$, the ROC curve have to satisfy the following conditions

- **C1:** $\int_0^1 h'(z)dz = 1$
- **C2:** $h'(z) \geqslant 0$
- **C3:** $h'(z)$ is decreasing

where $h'(z)$ is the derivative of the ROC curve, $h(z)$. Also for a given ROC curve, $h(z)$, we can compute the AUC as

$$\Pr\left(L_\Delta > 0\right) = \int_0^1 h(z)dz.$$

Then, using integration by parts, we can show that

$$1 - \Pr\left(L_\Delta > 0\right) = \int_0^1 z\, h'(z)dz.$$

To compute the possible feasible region stated in the theorem 3, we need to optimize both of following KL divergences, $\mathcal{D}\left(f_{L_1}(l)||f_{L_0}(l)\right)$ and $\mathcal{D}\left(f_{L_0}(l)||f_{L_1}(l)\right)$, with respect to the derivative of

1. Both $f_{L_0}(l)$ and $f_{L_1}(l)$ are PDFs in generalized Chi-squared distributions class. This means that each of these PDFs are convolution of weighted Chi-squared distributions. Weighted Chi-squared distribution is continuous in its domain thus, convolution of these distributions is continuous in its domain.

the ROC curve given a fixed AUC, $\Pr(L_\Delta > 0)$, while conditions, C1, C2 and C3 hold. To solve this optimization, we can use the method of Lagrange multiplier.

First step: Here we minimize $\mathcal{D}(f_{L_1}(l)||f_{L_0}(l))$ with respect to the derivative of the ROC curve given the constraints. Optimization problem is as follow

$$\begin{aligned} \underset{h'(z)}{\arg\min} \quad & -\int_0^1 \log h'(z)dz \\ \text{s. t.} \quad & \int_0^1 z\, h'(z)dz = 1 - \Pr(L_\Delta > 0) \\ & \text{C1, C2 \& C3.} \end{aligned} \tag{A.1}$$

To solve this optimization problem, we first write the Lagrangian. We need two coefficients a and b corresponding to conditions in optimization problem (A.1). Then, we can write the Lagrange multiplier as a function of the derivative of the ROC curve, z, a and b as follow

$$\begin{aligned} L(h'(z), z, a, b) = & -\int_0^1 \log h'(z)dz \\ & + a\left(\int_0^1 zh'(z)dz - (1 - \Pr(L_\Delta > 0))\right) \\ & + b\left(\int_0^1 h'(z)dz - 1\right). \end{aligned}$$

Note that, the Lagrangian, $L(h'(z), z, a, b)$ is a convex functional [73] of $h'(z)$. Thus, we can compute its minimum by taking its derivative with respect to $h'(z)$. Doing so, and applying the Euler-Lagrange equation [73] we get

$$\begin{aligned} \frac{\delta L(h'(z), z, a, b)}{\delta h'(z)} &= \frac{\partial L}{\partial h'} - \frac{d}{dz}\frac{\partial L}{\partial h''} \\ &= \int_0^1 \left(az + b - \frac{1}{h'(z)}\right) dz. \end{aligned}$$

Set $\frac{\partial L(h'(z), z, a, b)}{\partial h'(z)} = 0$ we get

$$h'(z) = \frac{1}{az + b}$$

for all $z \in [0, 1]$. From C3, since $h'(z)$ is decreasing, we can conclude that $a > 0$. Moreover, from C1,

at optimum we have $\int_0^1 h'(z)dz = 1$ and thus, we can compute one of the coefficients as $b = \frac{a}{e^a - 1}$.

Computing the AUC integral and the KL divergence using the ROC curve we get the following parametric boundary for the possible feasible region

$$\Pr\left(L_\Delta > 0\right) = \frac{1}{1 - e^{-a}} - \frac{1}{a} \tag{A.2}$$

and

$$\mathcal{D} = \log(a) + \frac{a}{e^a - 1} - 1 - \log(1 - e^{-a}) \tag{A.3}$$

where $\mathcal{D} = \mathcal{D}\left(f_{L_1}(l)||f_{L_0}(l)\right)$.

Second step: Here we minimize $\mathcal{D}\left(f_{L_0}(l)||f_{L_1}(l)\right)$. The Lagrange multiplier for this step is similar to the first step but it is more straight forward if we define $g(\eta) = h^{-1}(\eta)$. Note that using integration by parts, we can show that AUC is

$$\Pr\left(L_\Delta > 0\right) = \int_0^1 \eta\, g'(\eta) d\eta.$$

Now, we can write the Lagrangian for the optimization problem with respect to $g'(\eta)$. The Lagrangian is convex with respect to $g'(\eta)$, thus taking the derivative and set it equal to zero as follow

$$\frac{\delta L(g'(\eta), \eta, a, b)}{\delta g'(\eta)} = 0$$

we can compute the parametric boundary for the possible feasible region. The parametric boundary in this case is the same as solution in (A.2) and (A.3) with $\mathcal{D} = \mathcal{D}\left(f_{L_0}(l)||f_{L_1}(l)\right)$. Thus, combining these two steps, for the optimal boundary we have

$$\mathcal{D}_l^* = \min\{\mathcal{D}\left(f_{L_1}(l)||f_{L_0}(l)\right)\ ,\ \mathcal{D}\left(f_{L_0}(l)||f_{L_1}(l)\right)\}.$$

■

A.3 Proof of Theorem 8

The proof is by construction. At each step we use a star approximation graph (a tree with one node connected to all other nodes). The Cholesky factorization is used to ensure that the sparsity pattern occurs in the inverse Cholesky matrix.

The first star approximation structure is constructed such that all the other nodes are connected to the first node. Then, the Cholesky decomposition is computed where the inverse Cholesky decomposition preserves the star structure. This particular construction causes node 1 to become disconnected from the rest of the graph (construction of $\boldsymbol{\Delta}_1$).

A formal, algebraic proof for this claim is given as follow. We generally partition the covariance matrix $\boldsymbol{\Sigma}$ as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_2 \end{bmatrix}.$$

According to the covariance selection rules from Dempster theorem [1], the model covariance matrix has the same coefficients as the covariance matrix, $\boldsymbol{\Sigma}$, at non-zero places in the inverse. For the sake of notation simplicity, let $\widetilde{\boldsymbol{\Sigma}}$ denotes the first stage model covariance matrix. For the proof of this part, we want the model inverse covariance matrix to have zeros at the block position of $\boldsymbol{\Sigma}_2$. Thus, the model covariance matrix, $\widetilde{\boldsymbol{\Sigma}}$, has the following partitioned covariance matrix

$$\widetilde{\boldsymbol{\Sigma}} = \begin{bmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \widetilde{\boldsymbol{\Sigma}}_2 \end{bmatrix}.$$

Using matrix inversing lemma [74] we have

$$\widetilde{\boldsymbol{\Sigma}}^{-1} = \begin{bmatrix} \boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\widetilde{\boldsymbol{\Sigma}}_{2|1}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1} & -\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\widetilde{\boldsymbol{\Sigma}}_{2|1}^{-1} \\ -\widetilde{\boldsymbol{\Sigma}}_{2|1}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1} & \widetilde{\boldsymbol{\Sigma}}_{2|1}^{-1} \end{bmatrix},$$

Where Schur compliment,

$$\widetilde{\boldsymbol{\Sigma}}_{2|1} = \widetilde{\boldsymbol{\Sigma}}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}$$

is the conditional covariance and is diagonal.

The inverse of model covariance matrix can be factor as

$$\widetilde{\mathbf{\Sigma}}^{-1} = \mathbf{Q}^T\mathbf{Q}$$

where $\mathbf{Q}$ is the inverse of the lower tridiagonal Cholesky decomposition of $\widetilde{\mathbf{\Sigma}}$ and can be partitioned as

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{0} \\ \mathbf{Q}_{21} & \mathbf{Q}_2 \end{bmatrix},$$

where $\mathbf{Q}_1$ and $\mathbf{Q}_2$ can be computed from the following

$$\mathbf{\Sigma}_1^{-1} = \mathbf{Q}_1^T\mathbf{Q}_1,$$

$$\widetilde{\mathbf{\Sigma}}_{2|1}^{-1} = \mathbf{Q}_2^T\mathbf{Q}_2,$$

and $\mathbf{Q}_{21}$ as follow

$$\mathbf{Q}_{21} = -\mathbf{Q}_2\mathbf{\Sigma}_{21}\mathbf{\Sigma}_1^{-1}.$$

Computing $\mathbf{Q\Sigma Q}^T$, we have

$$(\mathbf{Q\Sigma Q}^T)_1 = \mathbf{Q}_1\mathbf{\Sigma}_1\mathbf{Q}_1^T = \mathbf{I}$$

and

$$\begin{aligned}(\mathbf{Q\Sigma Q}^T)_{12} &= \mathbf{Q}_1\mathbf{\Sigma}_{12}\mathbf{Q}_2^T + \mathbf{Q}_1\mathbf{\Sigma}_1\mathbf{Q}_{12} \\ &= \mathbf{Q}_1(\mathbf{\Sigma}_{12} - \mathbf{\Sigma}_1\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_{12})\mathbf{Q}_2^T \\ &= \mathbf{Q}_1(\mathbf{0})\mathbf{Q}_2^T = \mathbf{0},\end{aligned}$$

and

$$
\begin{aligned}
(\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T)_2 &= \mathbf{Q}_{21}\boldsymbol{\Sigma}_1\mathbf{Q}_{12} + \mathbf{Q}_2\boldsymbol{\Sigma}_{21}\mathbf{Q}_{12} \\
&+ \mathbf{Q}_{21}\boldsymbol{\Sigma}_{12}\mathbf{Q}_2 + \mathbf{Q}_2\boldsymbol{\Sigma}_2\mathbf{Q}_2^T \\
&= \mathbf{Q}_2\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_1\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\mathbf{Q}_2^T \\
&- \mathbf{Q}_2\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\mathbf{Q}_2^T \\
&- \mathbf{Q}_2\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\mathbf{Q}_2 + \mathbf{Q}_2\boldsymbol{\Sigma}_2\mathbf{Q}_2^T \\
&= -\mathbf{Q}_2\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12}\mathbf{Q}_2 + \mathbf{Q}_2\boldsymbol{\Sigma}_2\mathbf{Q}_2^T \\
&\overset{(a)}{=} \mathbf{Q}_2(\boldsymbol{\Sigma}_2 - \widetilde{\boldsymbol{\Sigma}}_2 + \widetilde{\boldsymbol{\Sigma}}_2 - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_{12})\mathbf{Q}_2^T \\
&= \mathbf{I} + \mathbf{Q}_2(\boldsymbol{\Sigma}_2 - \widetilde{\boldsymbol{\Sigma}}_2)\mathbf{Q}_2^T,
\end{aligned}
$$

where (a) is true since we add and subtract $\widetilde{\boldsymbol{\Sigma}}_2$.

All diagonal coefficients of $\mathbf{Q}_2(\boldsymbol{\Sigma}_2 - \widetilde{\boldsymbol{\Sigma}}_2)\mathbf{Q}_2^T$ are zeros, since according to the covariance selection theorem [1], diagonal coefficients of $\boldsymbol{\Sigma}_2$ and $\widetilde{\boldsymbol{\Sigma}}_2$ are equal, and $\mathbf{Q}_2$ is diagonal.

Remark: Note that all diagonal coefficients of $(\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T)_2$ are ones since all diagonal coefficients of $\mathbf{Q}_2(\boldsymbol{\Sigma}_2 - \widetilde{\boldsymbol{\Sigma}}_2)\mathbf{Q}_2^T$ are zeros.

Overall, the $\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T$ (symmetric CAM) has the following structure

$$
\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T = \left[\begin{array}{c|cccc}
\mathbf{I} & \multicolumn{4}{c}{\mathbf{0}} \\
\hline
 & 1 & \boxdot & \cdots & \boxdot \\
\mathbf{0} & \boxdot & 1 & \ddots & \vdots \\
 & \vdots & \ddots & \ddots & \boxdot \\
 & \boxdot & \cdots & \boxdot & 1
\end{array}\right].
$$

For the star tree approximation we have

$$
\boldsymbol{\Delta}_1 = \mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^T = \left[\begin{array}{c|ccc}
1 & 0 & \cdots & 0 \\
\hline
0 & 1 & \boxdot & \boxdot \\
\vdots & \boxdot & \ddots & \boxdot \\
0 & \boxdot & \boxdot & 1
\end{array}\right].
$$

Repeating this construction process recursively, at the i-th iteration, node $i+1$ to node n are all connected to the i-th node in the i-th star approximation structure and thus, the i-th node become disconnected from the rest of the graph. Repeating this procedure for $n-1$ iterations, we get $\mathbf{\Delta}_{n-1} = \mathbf{I}$ which translates to zero model approximation error.

Note that, we can also optimize the choice of the star tree by minimizing the KL divergence by exhaustively searching over all the n possible star structures at each step of the cascade tree approximation algorithm.

A.4 Proof of Theorem 9

Our goal is to show that all diagonal coefficients of the symmetric CAM are equal to one. For simplicity and without losing generality of the proof we can assume that the permutation matrix is identity, i.e. we start with an appropriate ordering that satisfies theorem 7 conditions. Let $\mathbf{Q}$ be the inverse Cholesky factorization of the tree model covariance matrix. Let us also assume that the tree model is connected. With these assumptions, matrix $\mathbf{Q}$ has one non-zero coefficients in the first row (on the diagonal) and exactly two non-zero coefficients in each other row (one on the diagonal and one on its left), i.e. $\forall\ j < i \leqslant n, \quad q_{ji} \neq 0 \quad \text{and} \quad q_{ii} \neq 0$. This is true since $\mathbf{Q}$ is a lower triangular matrix that preserves tree structure.

We can right the symmetric CAM as follow

$$\begin{aligned}\mathbf{Q}\mathbf{\Sigma}\mathbf{Q}^T &= \mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}} + \widetilde{\mathbf{\Sigma}})\mathbf{Q}^T \\ &= \mathbf{I} + \mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})\mathbf{Q}^T.\end{aligned}$$

Note that, the difference $\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}$ has zeros at positions ji and ij where $q_{ji} \neq 0$. To prove this lemma, we need to show that $(\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})\mathbf{Q}^T)_{ii} = 0$ or equivalently we need to show that ith row of $\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})$ times the transpose of the ith row ($i > 1$) of $\mathbf{Q}$ is 0. There are only two non-zero coefficients in the ith row of $\mathbf{Q}$, at positions ii and ji. Thus we only need to compute $(\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}))_{ii}$ and $(\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}))_{ji}$. It is easy to see that $(\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}))_{ii} = 0$. For $(\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}}))_{ji}$, we have

$$[0 \ldots 0\ q_{ji}\ 0 \ldots 0\ q_{ii}\ 0 \ldots 0][\boxdot \ldots \boxdot\ 0\ \boxdot \ldots \boxdot\ 0\ \boxdot \ldots \boxdot]^T = 0.$$

This equality holds since $(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})_{ji} = 0$. Thus, $\forall i \leqslant n,\ \ (\mathbf{Q}(\mathbf{\Sigma} - \widetilde{\mathbf{\Sigma}})\mathbf{Q}^T)_{ii} = 0$ which results in this lemma.

Bibliography

[1] A. P. Dempster, "Covariance selection," *Biometrics*, vol. 28, no. 1, pp. 157–175, March 1972.

[2] N. Tafaghodi Khajavi, A. Kuh, and N. P. Santhanam, "Spatial correlations for solar PV generation and its tree approximation analysis," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Dec 2014, pp. 1–5.

[3] C. K. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *IEEE Transactions on Information Theory*, pp. 462–467, 1968.

[4] A. Corduneanu and C. M. Bishop, "Variational bayesian model selection for mixture distributions," in *Artificial intelligence and Statistics*, vol. 2001. Morgan Kaufmann Waltham, MA, 2001, pp. 27–34.

[5] J. B. Kadane and N. A. Lazar, "Methods and criteria for model selection," *Journal of the American statistical Association*, vol. 99, no. 465, pp. 279–290, 2004.

[6] M. I. Jordan, *Learning in graphical models*. Springer Science & Business Media, 1998, vol. 89.

[7] M. J. Wainwright, M. I. Jordan *et al.*, "Graphical models, exponential families, and variational inference," *Foundations and Trends® in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[8] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[9] D. Heckerman, "A tutorial on learning with bayesian networks," in *Learning in graphical models*. Springer, 1998, pp. 301–354.

[10] S. L. Lauritzen, *Graphical models*. Clarendon Press, 1996.

[11] J. Dahl, L. Vandenberghe, and V. Roychowdhury, "Covariance selection for nonchordal graphs via chordal embedding," *Optimization Methods & Software*, vol. 23, no. 4, pp. 501–520, 2008.

[12] I. Guyon, A. Saffari, G. Dror, and G. Cawley, "Model selection: Beyond the bayesian/frequentist divide," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 61–87, 2010.

[13] O. Frank and D. Strauss, "Markov graphs," *Journal of the american Statistical association*, vol. 81, no. 395, pp. 832–842, 1986.

[14] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, 2013.

[15] J. B. Ekanayake, N. Jenkins, K. Liyanage, J. Wu, and A. Yokoyama, *Smart grid: technology and applications*. John Wiley & Sons, 2012.

[16] N. T. Khajavi and A. Kuh, "First order Markov chain approximation of microgrid renewable generators covariance matrix," in *Proc. IEEE International Symposium on Information Theory, Istanbul, Turkey (ISIT' 13)*, July 2013, pp. 1207–1211.

[17] ——, "The quality of the covariance selection through detection problem and AUC bounds," *arXiv preprint arXiv:1605.05776*, 2016.

[18] ——, "Quality of the covariance selection through detection problem and AUC bounds," *APSIPA transactions on signal and information processing*, 2019.

[19] ——, "The goodness of covariance selection problem from AUC bounds," in *Proc. 54th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 2016.

[20] J. H. Dauwels, *On graphical models for communications and machine learning: Algorithms, bounds, and analog implementation*. ETH Zurich, 2006, vol. 17.

[21] H. Jeffreys, "An invariant form for the prior probability in estimation problems," *Proceedings*

of the Royal Society of London. Series A. Mathematical and Physical Sciences, vol. 186, no. 1007, pp. 453–461, 1946.

[22] J. Z. Huang, N. Liu, M. Pourahmadi, and L. Liu, "Covariance matrix selection and estimation via penalised normal likelihood," *Biometrika*, vol. 93, no. 1, pp. 85–98, 2006.

[23] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 498–519, 2001.

[24] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proceedings of the IEEE*, vol. 95, no. 6, pp. 1295–1322, June 2007.

[25] D. Bickson, "Gaussian belief propagation: Theory and aplication," *arXiv preprint arXiv:0811.2518*, 2008.

[26] O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson, and D. Dolev, "Gaussian belief propagation solver for systems of linear equations," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on.* IEEE, 2008, pp. 1863–1867.

[27] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[28] R. Bulterman, F. Van der Sommen, G. Zwaan, T. Verhoeff, A. Van Gasteren, and W. Feijen, "On computing a longest path in a tree," *Information Processing Letters*, vol. 81, no. 2, pp. 93–96, 2002.

[29] N. Meinshausen and P. Buhlmann, "Model selection through sparse maximum likelihood estimation," *Annals of Statistics*, pp. 1436–1464, 2006.

[30] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[31] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American Statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[32] D. J. MacKay, *Information theory, inference, and learning algorithms.* Citeseer, 2003, vol. 7.

[33] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical society*, vol. 7, no. 1, pp. 48–50, 1956.

[34] R. C. Prim, "Shortest connection networks and some generalizations," *Bell system technical journal*, vol. 36, no. 6, pp. 1389–1401, 1957.

[35] N. T. Khajavi, "Latent tree approximation in linear model," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5940–5944.

[36] P. M. L. II, "Approximating probability distributions to reduce storage requirements," *Information and control*, vol. 2, no. 3, pp. 214–225, 1959.

[37] E. L. Lehmann and J. P. Romano, *Testing statistical hypotheses*. springer, 2006.

[38] J. Neyman and E. Pearson, "On the use and interpretation of certain test criteria for purposes of statistical inference," *Biometrika*, vol. 20, 1928.

[39] S. Eguchi and J. Copas, "Interpreting Kullback-Leibler divergence with the Neyman-Pearson lemma," *Journal of Multivariate Analysis*, vol. 97, no. 9, pp. 2034–2040, 2006.

[40] L. L. Scharf, *Statistical signal processing*. Addison-Wesley Reading, MA, 1991, vol. 98.

[41] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *Annals of the History of Computing*, vol. 7, no. 1, pp. 43–57, 1985.

[42] A. Kavcic and J. M. F. Moura, "Matrices with banded inverses: Inversion algorithms and factorization of Gauss-Markov processes," *IEEE Transactions on Information Theory*, vol. 46, pp. 1495–1509, July 2000.

[43] A. N. Shiryaev, "Probability, volume 95 of graduate texts in mathematics," 1996.

[44] J. A. Hanley and B. J. McNeil, "The meaning and use of the area under a receiver operating characteristic (roc) curve." *Radiology*, vol. 143, no. 1, pp. 29–36, 1982.

[45] N. T. Khajavi and A. Kuh, "The quality of tree approximation from AUC bounds," *Information Theory and Applications Workshop*, 2016.

[46] S. B. Provost and E. M. Rudiuk, "The exact distribution of indefinite quadratic forms in

noncentral normal vectors," *Annals of the Institute of Statistical Mathematics*, vol. 48, no. 2, pp. 381–394, 1996.

[47] H.-T. Ha and S. B. Provost, "An accurate approximation to the distribution of a linear combination of non-central Chi-square random variables," *REVSTAT–Statistical Journal*, vol. 11, no. 3, pp. 231–254, 2013.

[48] T. Y. Al-Naffouri and B. Hassibi, "On the distribution of indefinite quadratic forms in Gaussian random variables," in *Information Theory, 2009. ISIT 2009. IEEE International Symposium on.* IEEE, 2009, pp. 1744–1748.

[49] S. Kotz, T. Kozubowski, and K. Podgorski, *The Laplace distribution and generalizations: a revisit with applications to communications, economics, engineering, and finance.* Springer Science & Business Media, 2012.

[50] M. Abramowitz and A. I. Stegun, "Handbook of mathematical functions," *Applied mathematics series*, vol. 55, p. 62, 1966.

[51] T. M. Cover and J. A. Thomas, *Elements of information theory.* John Wiley & Sons, 2012.

[52] S. Kullback, *Information theory and statistics.* Courier Corporation, 1968.

[53] P. Harremoës and I. Vajda, "On pairs of f-divergences and their joint range," *arXiv preprint arXiv:1007.0097*, 2010.

[54] N. P. Johnson, "Advantages to transforming the receiver operating characteristic (roc) curve into likelihood ratio co-ordinates," *Statistics in medicine*, vol. 23, no. 14, pp. 2257–2266, 2004.

[55] N. S. irradiance website. [Online]. Available: http://www.nrel.gov/midc/.

[56] S. A. Fatemi and A. Kuh, "Solar radiation forecasting using Zenith angle," *Global SIP conference*, 2013.

[57] C. E. Rasmussen and C. K. I. Williams, "Gaussian processes for machine learning," *the MIT Press*, 2006.

[58] J. R. Blair and B. Peyton, "An introduction to chordal graphs and clique trees," in *Graph theory and sparse matrix computation.* Springer, 1993, pp. 1–29.

[59] N. T. Khajavi and A. Kuh, "The covariance selection quality for graphs with junction trees through AUC bounds," in *Proc. of the Asia-Pacific Signal and Information Processing Association (APSIPA ASC)*, Dec 2016, pp. 1–5.

[60] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in gaussian graphical models of arbitrary topology," in *Advances in neural information processing systems*, 2000, pp. 673–679.

[61] A. Anandkumar, D. J. Hsu, F. Huang, and S. M. Kakade, "Learning mixtures of tree graphical models," in *Advances in Neural Information Processing Systems*, 2012, pp. 1052–1060.

[62] R. Santana, A. Ochoa-Rodriguez, and M. R. Soto, "The mixture of trees factorized distribution algorithm," in *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 2001, pp. 543–550.

[63] M. Meila and M. I. Jordan, "Learning with mixtures of trees," *Journal of Machine Learning Research*, vol. 1, no. Oct, pp. 1–48, 2000.

[64] S. Dasgupta, "Learning mixtures of Gaussians," in *Foundations of computer science, 1999. 40th annual symposium on*. IEEE, 1999, pp. 634–644.

[65] C. Greene and G. A. Iba, "Cayley's formula for multidimensional trees," *Discrete Mathematics*, vol. 13, no. 1, pp. 1–11, 1975.

[66] N. Tafaghodi Khajavi and A. Kuh, "Covariance matrix decomposition using cascade of linear tree transformations." IEEE, 2019.

[67] K. Khare and B. Rajaratnam, "Sparse matrix decompositions and graph characterizations," *Linear Algebra and its Applications*, vol. 437, no. 3, pp. 932–947, 2012.

[68] L. Ghalami and D. Grosu, "Scheduling parallel identical machines to minimize makespan: A parallel approximation algorithm," *Journal of Parallel and Distributed Computing*, 2018.

[69] ——, "A parallel approximation algorithm for scheduling parallel identical machines," in *Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2017 IEEE International*. IEEE, 2017, pp. 442–451.

[70] E. Dalkiran and L. Ghalami, "On linear programming relaxations for solving polynomial programming problems," *Computers & Operations Research*, 2018.

[71] Y. Li, L. Ghalami, L. Schwiebert, and D. Grosu, "A gpu parallel approximation algorithm for scheduling parallel identical machines to minimize makespan," in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018, pp. 619–628.

[72] H. Flanders, "Differentiation under the integral sign," *The American Mathematical Monthly*, vol. 80, no. 6, pp. 615–627, 1973.

[73] D. G. Luenberger, *Optimization by vector space methods*. John Wiley & Sons, 1997.

[74] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*. http://www2.imm.dtu.dk/pubdb/p.php?3274: Technical University of Denmark, October 2008.