Large-Deviation Analysis and Applications Of Learning Tree-Structured Graphical Models

by

Vincent Yan Fu Tan

B.A., Electrical and Information Sciences,
University of Cambridge, 2005

M.Eng., Electrical and Information Sciences,
University of Cambridge, 2005

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in Electrical Engineering and Computer Science
at the Massachusetts Institute of Technology

February 2011

Author: _____

Department of Electrical Engineering and Computer Science
December 13, 2010

Certified by: _____

Professor Alan S. Willsky
Edwin Sibley Webster Professor of Electrical Engineering
Thesis Supervisor

Accepted by: _____

Professor Terry P. Orlando
Chair, Department Committee on Graduate Students

Large-Deviation Analysis and Applications Of Learning Tree-Structured Graphical Models

by Vincent Yan Fu Tan

Submitted to the Department of Electrical Engineering and Computer Science
on December 13, 2010, in partial fulfilment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

The design and analysis of complexity-reduced representations for multivariate data is important in many scientific and engineering domains. This thesis explores such representations from two different perspectives: deriving and analyzing performance measures for learning tree-structured graphical models and salient feature subset selection for discrimination.

Graphical models have proven to be a flexible class of probabilistic models for approximating high-dimensional data. Learning the structure of such models from data is an important generic task. It is known that if the data are drawn from tree-structured distributions, then the algorithm of Chow and Liu (1968) provides an efficient algorithm for finding the tree that maximizes the likelihood of the data. We leverage this algorithm and the theory of large deviations to derive the error exponent of structure learning for discrete and Gaussian graphical models. We determine the extremal tree structures for learning, that is, the structures that lead to the highest and lowest exponents. We prove that the star minimizes the exponent and the chain maximizes the exponent, which means that among all unlabeled trees, the star and the chain are the worst and best for learning respectively. The analysis is also extended to learning forest-structured graphical models by augmenting the Chow-Liu algorithm with a thresholding procedure. We prove scaling laws on the number of samples and the number variables for structure learning to remain consistent in high-dimensions.

The next part of the thesis is concerned with discrimination. We design computationally efficient tree-based algorithms to learn pairs of distributions that are specifically adapted to the task of discrimination and show that they perform well on various datasets vis-à-vis existing tree-based algorithms. We define the notion of a salient set for discrimination using information-theoretic quantities and derive scaling laws on the number of samples so that the salient set can be recovered asymptotically.

Thesis Supervisor: Professor Alan S. Willsky
Title: Edwin Sibley Webster Professor of Electrical Engineering and Computer Science

Acknowledgments

This has been a long and, at times, arduous journey. It would have been much longer or even impossible without the help of many individuals.

I would like to thank my advisor Prof. Alan Willsky for giving me complete freedom to pursue my research interests. Alan has taught me to think deeply about which problems are worth solving and how to present ideas concisely and cogently. I thank him for drawing on his wealth of knowledge to provide me with high-level ideas during our weekly grouplet meetings and also his promptness in giving me comments on journal papers and thesis chapters. If I have the privilege to be a professor, I will strive to emulate Alan.

I would like express my sincere gratitude to my thesis committee Dr. John Fisher, Prof. Lizhong Zheng and Prof. Mike Collins. I thank John for the many technical discussions during and outside grouplet meetings. John has also been tireless in helping me with the revisions of our papers. The cute idea of approximating information-theoretic optimization problems by assuming distributions are close (leading to Chapters 3 and 4) is due in large part to my conversations with Lizhong. I thank Mike for being on my RQE committee and lending his insights on AdaBoost and Real-AdaBoost.

I am fortunate to have had the chance to be a teaching assistant with Profs. Munther Dahleh and Greg Wornell. I thank both professors for showing me two (very) different styles of teaching and for reinforcing my confidence in teaching. I especially want to thank Greg for teaching me how to design unambiguous exam problems.

There are many other professors who have shown a keen interest in my intellectual development including Prof. Vivek Goyal and Prof. Sanjoy Mitter. Vivek has been a terrific graduate counsellor, providing me with excellent advice throughout my graduate studies. In particular, he guided me through the process of writing my first journal paper and for that, I am extremely thankful. Sanjoy has always been willing to listen my ideas and for me to tap on his vast knowledge. I would also like to thank Prof. Lang Tong (Cornell) who was very helpful in refining the ideas and the writing in Chapter 3. I am grateful to professors from the Math department for teaching me Math.

The Stochastic Systems Group (SSG) has been my academic home during my PhD. I am particularly indebted to three individuals, Anima Anandkumar, Sujay Sanghavi and Pat Kriedl. Anima has been my main collaborator over the past two years. The collaboration has been both fruitful and enjoyable. I hope we continue working together in future! Sujay, who introduced me to learning tree models, has clearly had a profound influence on this thesis. I am grateful for Sujay's help during the first two years of my doctoral studies. Pat has always been present to provide me with sincere words of encouragement when the going got tough. I would like to thank my office mates, Myung

Jin Choi, Venkat Chandrasekaran, Ying Liu and Michael Chen for stimulating discussions and for putting up with my idiosyncrasies. I also want to thank other members of SSG including Ayres Fan, Dmitry Malioutov, Jason Johnson, Mike Siracusa, Emily Fox, Kush Varshney, Justin Dauwels, Jason Chang, Matt Johnson, James Saunderson and Oliver Kosut for far too many things to mention here.

My PhD experience has been greatly enhanced my interactions with many other students from EECS including Mukul Agrawal, Megumi Ando, Shashi Borade, Lihfeng Cheow, Rose Faghih, Shirin Farrahi, Peter Jones, Sertac Karaman, Trina Kok, Baris Nakiboglu, Alex Olshevsky, Mitra Osqui, Marco Pavone, Bill Richoux, Lav Varshney, Dennis Wei, Daniel Weller and Jessica Wu. Special mention goes to Hye Won Chung, Mina Karzand, John Sun and Da Wang for many discussions during our study group on network information theory in the summer and fall of 2010. Sincere thanks also goes to the students who took 6.241 in the Fall of 2008 and the students who took 6.437 in the Spring of 2010 for giving me good feedback on my teaching. I am thankful for the chance to co-chair the LIDS students conference in 2009. I thank my fellow committee members for ensuring that it was a success.

I would like to thank the LIDS staff including Jennifer Donovan, Rachel Cohen, Lynne Dell and Brian Jones.

I would like to acknowledge my Master's thesis supervisor Dr. Cédric Févotte, who encouraged me to do a PhD.

My graduate career has undoubtedly been enriched by two internships at Microsoft Research and a visit to HP Labs. I would like to thank Chris Bishop, John Winn, Markus Svénsen and John Guiver in Cambridge, UK and David Heckerman, Jonathan Carlson and Jennifer Listgarten in Los Angeles for making me feel part of their research groups and for many fruitful interactions. I thank Adnan Custovic, Angela Simpson and Iain Buchan from the University of Manchester for hosting me and for our collaboration on the asthma project. I thank Majid Fozunbal and Mitch Trott for hosting me at HP Labs in Jan 2010.

I am thankful for financial support by the Public Service Commission and the Agency for Science, Technology and Research (A*STAR), Singapore. I am also thankful for the friendship of the Singaporean community at MIT, especially Henry and Shireen.

I am extremely grateful to my parents for their support in all my endeavors. Lastly, and most importantly, words simply cannot express my love and gratitude to my wife Huili without whom this thesis would certainly have been impossible to complete. Her unwavering support for me in my pursuit of my graduate studies has kept me going in the toughest of times. I hope to be equally supportive in future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for This Thesis

IMAGINE the following scenario: There are 1000 children participating in a longitudinal study of childhood asthma. These children are recruited prenatally and followed up prospectively at various fixed ages, allowing clinician scientists to analyze the complex interplay between environmental, genetic and physiological factors on a child's susceptibility to asthma. Because a single blood sample provides a multitude of genetic information, in the form of single-nucleotide polymorphisms[1] (or SNPs), there are close to 10^6 variables in the dataset. How can the clinician scientist make useful inferences about the structure of the data given the overwhelming number of variables? By *structure* we mean the relationships between the variables, the nature of the salient features and also the existence of the latent variables that may be inferred from the data. This example is modeled on the *Manchester Asthma and Allergy Study*[2] [55, 181].

Motivated by the above example, we observe that learning the structure, interdependencies and salient features of a large collection of random variables from a dataset is an important and generic task in many scientific and engineering domains. See [87, 127, 153, 209, 215] and references therein for many more examples. This task is extremely challenging when the dimensionality of the data is large compared to the number of samples. Furthermore, structure learning and dimensionality reduction of high-dimensional distributions is also complicated as it is imperative to find the right balance between data fidelity and overfitting the data to the model. One typically simplifies this difficult structure learning problem by making two assumptions: Firstly, that there are very few interdependencies between the variables, so for example, only a few genes result in a particular positive physiological measurement. Secondly, one assumes that the number of salient features is small relative to the total number of variables, so for instance, asthma is influenced by only ten primary genetic factors within the large dataset. This thesis focuses on exploring these two aspects of model order reduction from an information-theoretic [47] perspective. In particular, we leverage on the use of the theory of *large deviations* [59, 62], which is the study of the probabilities of sequences

[1] A single-nucleotide polymorphism is a DNA sequence variation in which a nucleotide in the genome differs between members of a species.

[2] See http://www.maas.org.uk for more details.

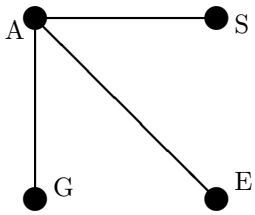

Figure 1.1. A graphical model based on the asthma example. In this case, we have identified the four salient variables to be asthma (A), an allergen-specific skin prick test (S), a particular gene (G) and a particular environmental factor (E). These variables are interlinked via a sparse undirected graph with three edges.

of events that decay exponentially fast.

This thesis analyzes the above-mentioned modeling problems under the unified formalism of *probabilistic graphical models* [69, 117, 127, 209], which provide a robust framework for capturing the statistical dependencies among a large collection of random variables. Indeed, graphical models derive their power from their ability to provide a diagrammatic representation of the multivariate distribution in the form of an undirected[3] graph $G = (V, E)$. The sparsity of the underlying graph structure allows for the design of computationally efficient algorithms for the purpose of performing statistical inference. Referring to the asthma example again, we can model the variables as the nodes in a graph V and their dependencies via the edges of the graph E (See Fig. 1.1). Learning the subset of salient variables to include in the model as well as the edges provides the clinicians deeper insight into the statistical features of the large dataset. Graphical models have found many applications in many areas of science and engineering including bioinformatics [4, 83], image processing [131, 217], iterative decoding [121], multiscale modeling [39, 40], computer vision [184], natural language processing [109] and combinatorial optimization [98, 169].

It is known that the set of trees, i.e., distributions which are Markov on acyclic graphs, is a tractable class of undirected graphical models. When the underlying structure is a tree, then it is known that statistical inference can be performed efficiently and accurately using the belief propagation or sum-product algorithm [122, 153]. Indeed, the *learning* task is also greatly simplified by appealing to the decomposition of a tree-structured graphical model into node and pairwise marginals. Given an arbitrary distribution P, the algorithm proposed by Chow and Liu [42] in their seminal paper shows how to efficiently search for the tree-structured graphical model $\widehat{Q}$ that minimizes the Kullback-Leibler divergence $D(P \,||\, Q)$ via a max-weight spanning tree [45] (MWST) procedure. When samples are available and one seeks to find the maximum-likelihood fit of the samples to a tree model, the same paper shows how to convert the problem to a MWST optimization where the edge weights are the empirical mutual information quantities, which can be easily estimated from the data.

[3]For the most part of the thesis, we focus on undirected graphs but directed graphical models (or Bayesian networks) also form an active area of research. See [56, 143, 153] for thorough expositions on this subject.

In the first part of this thesis, we ask (and answer) the following question: Given that data are independently drawn from P, a tree-structured graphical model, how well can the procedure of Chow and Liu estimate the model? Posed in a slightly different manner, how many samples (or in the asthma example, children involved in the study) does one need in order to drive the probability of error in structure learning below some pre-specified level $\delta > 0$? The results, which are proven using elements from the theory of large deviations, shed light on the nature of various classes of tree-structured distributions, and in particular, the ease or difficulty of learning them in terms of the *sample complexity*. While there has been significant research interest in learning graphical models from data (as will be thoroughly reviewed in Section 2.5.1), we derive a single figure-of-merit known as the *error exponent* which completely and precisely characterizes the ease of learning various tree models.

Of course, if there are many redundant (or non-salient) variables in the dataset, even modeling the variables via sparse graphs such as trees may lead to severe overfitting. A natural pre-processing step is to judiciously remove these variables prior to graphical modeling, i.e., to do *dimensionality reduction*. While there are many established methods to perform dimensionality reduction such as principal component analysis (PCA) [154], Isomap [196], local linear embedding [165] as well as work on combining this task with decision-making [204], we ask two fundamental questions in the second part of this thesis: How many samples are necessary and sufficient for asymptotically extracting the so-called salient feature subset for the purpose of hypothesis testing? How can one extract such salient features in a computationally efficient fashion? The successful search for such a subset of variables as a pre-processing step drastically reduces the complexity of the ensuing lower-dimensional model or classifier. It is an added advantage if the search of the salient feature set can be done efficiently. We show that this is indeed the case assuming the true distributions belong to the class of tree-structured graphical models. This is one other task in which trees have proven to afford significant computational savings.

■ 1.2 Overview of Thesis Contributions

This section provides a glimpse of the main technical contributions in the subsequent chapters. Chapter 2 provides the mathematical preliminaries. The rest of the thesis is divided coarsely into two main themes.

- Modeling: Chapters 3 to 5 deal with the analysis of modeling high-dimensional data with tree- or forest-structured distributions.
- Saliency: Chapters 6 and 7 deal with the problem of learning lower-dimensional or salient representations of data for the purpose of binary hypothesis testing.

Chapter 8 concludes the thesis, mentions some on-going research and suggests promising directions for further research.

■ 1.2.1 Chapter 2: Background

Chapter 2 provides the necessary background on five related topics:

- Fundamentals of information theory
- The method of types, large deviations and asymptotics
- Classification and boosting
- Graphical models
- Learning graphical models.

The first two topics set the stage for the use of information-thereotic ideas throughout the thesis. The properties of the entropy of a random variable, the KL-divergence (or relative entropy) between two probability measures and mutual information are reviewed. We state Fano's inequality [73] for the purpose of proving converses in the sequel. We also provide a flavor of *Euclidean information theory* [26], which states that if two probability measures P and Q are close, then the KL-divergence $D(P\,||\,Q)$ can be approximated by a weighted Euclidean norm. This approximation comes in handy when we seek to develop qualitative insights into the nature of the error exponents. Next, we describe the method of types [49], a powerful combinatorial technique to study the large-sample behavior of *types* or *empirical distributions*. It has proven to be useful for proving coding theorems and also in the study of large deviations. We state an important result, Sanov's theorem [171], which is used extensively to prove large deviation bounds in the rest of the thesis. The binary hypothesis testing section derives the optimal decision rule under the Neyman-Pearson and Bayesian settings – the likelihood ratio test. It also provides the form of the error exponents under these two settings.

In the next topic on classification and boosting, we set the stage for Chapter 6 by describing a prototypical problem in machine learning – supervised classification [21, 66, 96]. In supervised classification, the learner is provided with i.i.d. training samples of pairs of feature vectors and labels from some unknown joint distribution. Using this dataset, the learner would like to build a decision rule to discriminate unlabeled test samples. One way to do this, as we describe, is to judiciously combine weak-classifiers in a popular technique known as boosting [79]. Boosting has a variety of appealing theoretical properties which we describe in this section.

The final two topics are on graphical models [127, 209], which are also known as Markov random fields. Graphical models provide parsimonious representations for multivariate probability distributions. The subject brings together graph theory [213] and probability theory [19] to provide a diagrammatic representation of complex probability distributions over many variables. The structure of the graph allows for the design and analysis of efficient graph-based algorithms for statistical inference [153]. We review some basic definitions in graph theory and key ideas in graphical modeling.

An important line of analysis in the study of graphical models is the *learning* the structure of the graph from i.i.d. samples [107]. This problem has received a lot of attention of late and the relevant work is reviewed in the last topic. We also state and describe results from Chow and Liu [42] who provided an efficient implementation of the search for the maximum likelihood tree structure. Their work shows that the optimization reduces to a max-weight spanning tree problem with the empirical mutual information quantities as the edge weights. This algorithm is analyzed in detail in Chapters 3 and 4.

■ 1.2.2 Chapter 3: Large Deviations for Learning Discrete Tree Models

Chapter 3, which forms the basis for Chapters 4 and 5, is devoted to the study of error exponents for learning tree-structured models where each random variable can only take on a finite number of values. Consistency for structure learning for trees was established by Chow and Wagner [43]. However, while consistency is an important qualitative property, the error exponent provides a careful and precise quantitative measure of the ease of learning the structure of the graphical model. The problem is posed formally as follows: There is an unknown discrete tree-structured graphical model P Markov on $T_P = (V, E_P)$, where $|V| = d$ is the number of nodes (variables) in the graph. The learner is given access to i.i.d. samples $\mathbf{x}^n := \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and using the Chow-Liu algorithm, he can reconstruct the set of edges of the maximum-likelihood tree E_{ML}. We analytically evaluate the *error exponent* for structure learning

$$K_P := \lim_{n\to\infty} -\frac{1}{n} \log P^n(E_{\mathrm{ML}} \neq E_P), \tag{1.1}$$

by using techniques from large-deviations theory and the method of types [47, 59, 62]. We show that for non-degenerate tree models, the error probability decays exponentially fast as depicted in Fig. 1.2. The exponent K_P provides a quantitative measure of the ease of learning the model. If it is small, then the learner requires a large number of samples to learn the model and vice versa.

Our main contribution in this chapter is the evaluation of K_P in (1.1) by considering the large-deviation rates of the so-called *crossover events*, where a non-edge in E_{ML} replaces a true edge in E_P. In addition, we analyze the so-called *very-noisy* learning regime in which the algorithm is likely to make errors because the true mutual information on a non-edge is close to the mutual information of an edge. Because the error exponent is characterized exactly by a non-convex (and hence intractale) optimization problem, we then use Euclidean information theory [26] to approximate it. It is shown that the approximate error exponent can be interpreted as a *signal-to-noise* ratio for learning. This provides clear intuition as to what types of parameterizations of the models lead to difficult learning problems, i.e., problems in which one needs many samples in order to ensure that the error probability falls below a pre-specified level $\delta > 0$. Our results also extend to learning the (not necessarily unique) tree-projection of a non-tree distribution.

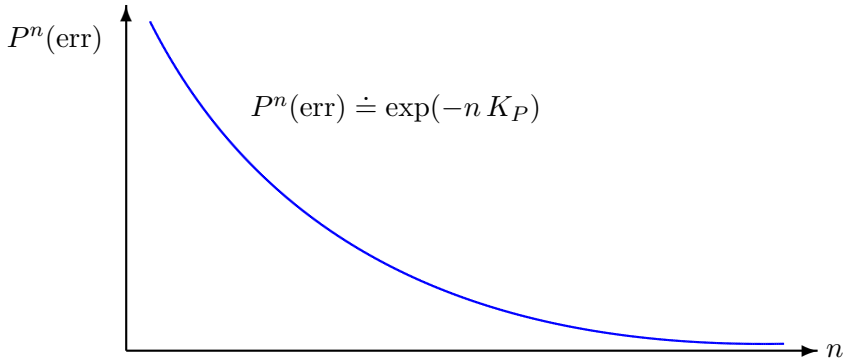

Figure 1.2. Illustration of the typical behavior of the probability of error for learning the structure of a tree model. The error probability decays with rate K_P, which in the case of (connected) trees, is strictly positive.

■ 1.2.3 Chapter 4: Large Deviations for Learning Gaussian Tree Models

Chapter 4 builds on the ideas on the previous chapter but analyzes Gaussian graphical models [69, 167], a widely-studied class of models for modeling continuous data. Many of the results from the previous chapter carry through to the Gaussian case albeit with slightly different proofs. We also leverage on the added structure of multivariate Gaussians and in particular the Markov property on trees (see Lemma 2.26), which states that the correlation coefficient of any non-neighbor pair of nodes is equal to the product of the correlation coefficients along its unique path.

Assuming that the models considered are in the very-noisy learning regime, we are able to prove what the author regards as perhaps the *most interesting* result in this thesis: That star graphs[4] are the most difficult to learn while Markov chains are the easiest. More formally, if we keep the parameterization (in terms of the correlation coefficients) of the models fixed, then the star minimizes a very-noisy approximation of the error exponent in (1.1) while the chain minimizes the same quantity. This *universal* result does not depend on the choice of the parameterization, i.e., the correlation coefficients. Furthermore, we are able to drastically reduce the computational complexity to find the exponent. It turns out that only $O(d)$ computations are required, compared to $O(d^{d-2})$ using a brute-force search. Even though the problem setups are similar, the proof techniques used in this chapter are very different from the ones employed in Chapter 3.

■ 1.2.4 Chapter 5: Learning High-Dimensional Forests

This chapter focuses on the analysis for learning forest-structured graphical models, i.e., models that are Markov on undirected, acyclic (but not necessarily connected)

[4] We will define these graphs formally in Section 2.4.1 but for the moment, *stars* are trees where all but one node has degree one. A *Markov chain* is a tree where all nodes have degree less than or equal to two.

graphs. For this class of models, there exists a subset of variables that are statistically independent of one another. In this case, the canonical Chow-Liu algorithm [42] being an ML implementation will, in general, favor richer models and thus overestimate the number of edges in the true model. The work in this chapter is motivated by high-dimensional modeling where the number of samples is small relative to the number of variables and we would like to learn very sparse models to avoid overfitting [132]. We model the paucity of data by considering learning a *sequence* of forest-structured models of increasing number of nodes d (with corresponding increasing sample size n).

There are two main contributions in this chapter: Firstly, we derive a sufficient condition on the *scaling law* on n and d and also the true number of edges k such that the probability of error of structure recovery tends to one when all these three model parameters scale. Interestingly, we show that even if d and k grow faster than any polynomial in n, structure recovery is possible in high-dimensions. Our proof relies on controlling the overestimation and underestimation errors (in k) and draws on ideas from the area of study known as *Markov order estimation* [77]. Secondly, we study the decay of the *risk* of the estimated model relative to the true model. Our results improve on recent work by Liu et al. [132] and Gupta et al. [89].

■ 1.2.5 Chapter 6: Learning Graphical Models for Hypothesis Testing

This chapters departs from the modeling framework discussed in the previous three chapters and instead considers learning *pairs* of distributions to be used in a likelihood ratio test for the specific purpose of *hypothesis testing* (or classification). The generative techniques (such as in [2, 128, 136, 211]) used to approximate high-dimensional distributions are typically not readily adaptable to discrimination or classification. This is because the purpose of modeling is to faithfully capture the entire behavior of a distribution, whereas in discrimination, the aim is to discover the salient differences in a pair of multivariate distributions. However, if the approximating distributions are trees, then we show that discriminatively learning such models is both computationally efficient and results in improved classification accuracy vis-à-vis existing tree-based algorithms such as Tree Augmented Naïve Bayes [84]. This chapter is thus devoted to the development of a new classifier that leverages on the modeling ability of tree-structured graphical models. As noted in the recent thesis by K. R. Varshney [203], "no one classifier is always superior", so the design and performance analysis of new classifiers is useful [216].

There are three contributions in this chapter. Firstly, given a set of samples, we develop computationally efficient tree-based algorithms to learn pairs of tree-structured models to be used in an approximate likelihood ratio test. We do so by maximizing a quantity known as the *tree-approximate J-divergence*, which in the discrete case reduces to a quantity proportional to the empirical log-likelihood ratio. We show that this maximization can be done efficiently, giving a pair of tree-structured distributions that are learned specifically for the purpose of discrimination. Secondly, we apply ideas from boosting [79] to learn thicker models, i.e., models that contain more edges than

trees do. This provides a systematic procedure to do pairwise feature selection [90] because the edges selected can be interpreted as the most salient ones for the purpose of discriminating between the two distributions. Finally, we validate the classification accuracy of the algorithms by performing extensive experiments on real and synthetic data.

■ 1.2.6 Chapter 7: Conditions for Salient Subset Recovery

This final technical chapter builds on the theme of salient feature selection from the previous chapter. In particular, we would like to discover what the fundamental information-theoretic limits are for the recovering the salient feature set given a set of samples. As in Chapter 5, this chapter also deals with the high-dimensional scenario in which the number of variables scale with the number of samples. The goal is to find scaling laws so that the recovery of the salient set, which is defined in terms of the error exponents of hypothesis testing, is asymptotically achievable. Conversely, we would also like to determine when it is impossible to recover the salient set, i.e., when the number of samples is insufficient for performing such a task.

There are three main contributions in this chapter. Firstly, we prove an achievability result: We show that there exists constants C, C' such that if the number of samples n satisfies

$$n > \max\left\{Ck\log\left(\frac{d-k}{k}\right), \exp(C'k)\right\} \tag{1.2}$$

where k and d are the number of salient features and the total number of features respectively, the error probability in recovering the salient set can be made arbitrarily small as the model parameters scale. One way to interpret this result is to regard k as a constant. In this case (1.2) says that the number of samples depends linearly on k and logarithmically on the "ambient dimensionality" d. Secondly, we prove a converse result. We show that under appropriate conditions, if

$$n < C''\log\left(\frac{d}{k}\right), \tag{1.3}$$

recovery of the salient set is no longer possible. This result follows from an application of Fano's inequality. Thirdly, we show that if the probability models are trees, then there exists an efficient tree-based algorithm based on a dynamic programming procedure to recover the salient set in time $O(dk^2)$.

■ 1.2.7 Chapter 8: Conclusions

This concluding chapter summarizes the thesis and suggests many possible directions for future work based on the material presented in this thesis. For example, one possible direction is to consider the learning of tree models in an online fashion. This delves into the realm of *online learning* [160, 222]. The work in this thesis is focused on the so-called *batch learning* scenario where all the data is available for learning. However,

in many real-time systems the data arrives sequentially. What can be said about the theoretical properties of the models learned at each time step? Are there any efficient algorithms to update the models sequentially?

Another possible direction is the learning of tree models with hidden (or latent) variables where observations are only available from a subset of variables. These problems have many applications from phylogenetics [71] to computer vision [152] to network tomography inference [149, 150]. We have begun our foray into this subject by performing algorithmic studies in Choi et al. [41], but we have yet to perform unified and detailed error exponent analysis. We believe that as in Chapter 4 such a line of analysis will lead to interesting insights as to which classes of latent tree models are easy to learn. These promising research directions (and more) are described in greater detail in Chapter 8.

■ 1.3 Bibliographical Notes

Portions of the material presented in this thesis have been presented at various conferences and submitted or published in various journals.

- The material in Chapter 3 was presented at the 2009 IEEE International Symposium on Information Theory [187] and to be published in the IEEE Transactions on Information Theory [188].
- The material in Chapter 4 was presented at the 2009 Annual Allerton Conference on Communication, Control, and Computing [189] and published in the IEEE Transactions on Signal Processing [193] in May 2010.
- The material in Chapter 5 was presented at the 2010 Annual Allerton Conference on Communication, Control, and Computing [191] and has been submitted to the Journal of Machine Learning Research [192] in May 2010.
- The material in Chapter 6 was presented at the 2007 IEEE Statistical Signal Processing Workshop [170] and the 2008 IEEE International Conference on Acoustics, Speech, and Signal Processing [186] and appeared in the IEEE Transactions on Signal Processing [195] in Nov 2010.
- The material in Chapter 7 was presented at the 2010 IEEE International Symposium on Information Theory [194].
- A small subset of the material in Chapter 8 was presented at the 2010 Annual Allerton Conference on Communication, Control, and Computing [41] and the NIPS Workshop on Robust Statistical Learning [9].

Chapter 2

Background

THIS background chapter describes several topics in applied mathematics, probability, and statistics that form the theoretical foundation for the thesis. Five topics are covered: (i) information theory, (ii) its application to statistics and the method of types, (iii) supervised classification and boosting (iv) graphical models and (v) learning graphical models.

The dawn of information theory was due to a landmark paper by C. E. Shannon [178]. Standard references in information theory include Cover and Thomas [47], Yeung [218] and Csiszár and Körner [50]. In the late 1970s, Csiszár and Körner [53] developed a powerful technique known as the method of types to analyze the properties of *types*, also known as *empirical distributions*. This led to the use of particularly convenient combinatorial techniques for proving Shannon's original coding theorems. The method of types is also used in the study of large deviations [59, 62, 64, 202], an area in probability theory that is concerned with the asymptotic behavior of remote tails of sequences of probability distributions. Supervised classification [21, 66, 96] is a prototypical problem in machine learning and boosting [79] is a commonly used algorithm to combine so-called weak classifiers to form a stronger classifier with better accuracy.

Graphical models, which provide parsimonious representations for multivariate probability distributions, grew out of the artificial intelligence community [153] and form a popular area of research in the machine learning [81], statistics [214], signal and image processing [215, 217] and information theory [122] communities. More recent expositions on graphical models (also known as Bayesian networks or Markov random fields) can be found in Lauritzen [127], Wainwright and Jordan [209], Koller and Friedman [117] and Bishop [21]. There has also been a surge of interest in learning such models from i.i.d. data samples [107] starting with the seminal work by Heckerman [97]. An efficient maximum-likelihood implementation for learning tree-structured graphical models was proposed by Chow and Liu [42]. The treatment of these topics in this chapter is limited to the scope required for the remainder of this thesis and is thus by no means comprehensive.

We assume that the reader is familiar with the basic notions in probability theory at the level of Bertsekas and Tsitsiklis [19] and analysis at the level of Rudin [166].

2.1 Information Theory

Information theory [47, 50, 178] is a branch of applied mathematics and electrical engineering involving the quantification of information. More precisely, information theory quantifies the fundamental limits of the compression and transmission of data. Since its inception in the late 1940s, it has broadened to find applications in many other areas, including statistical inference [53], probability theory and large deviations [49], computer science (Kolmogorov complexity) [119] and portfolio theory (Kelly gambling) in economics [114].

This section is devoted to the definition of various information-theoretic quantities, such as entropy, KL-divergence or relative entropy and mutual information (a special case of relative entropy). We state various properties of these information quantities such as the chain rule and the data-processing inequality. We introduce an important class of distributions known as exponential families by appealing to the maximum entropy principle. Standard converse tools such as Fano's inequality will also be stated. These properties and proof techniques will be useful in subsequent chapters. For example, the learning of graphical models from data in Chapters 3, 4 and 5 involves various information-theoretic quantities. The proof of the necessary conditions for salient subset recovery in Chapter 7 uses Fano's inequality. The exposition here is based largely on Cover and Thomas [47].

2.1.1 Notation

The following conventions will be adopted throughout this thesis. The set of natural numbers, integers, irrational numbers and real numbers will be denoted as $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ and $\mathbb{R}$ respectively. Random variables will be in upper case, e.g., X. Scalar variables, such as a particular value that a random variable takes on, are in lowercase, e.g., x. Vectors have bold typeface, e.g., $\mathbf{x}$, while scalars do not. Matrices have uppercase bold typeface, e.g., $\mathbf{X}$. The transpose of $\mathbf{X}$ is denoted as $\mathbf{X}^T$.

Standard asymptotic order notation [45] such as $O(\cdot)$, $\Omega(\cdot)$, $\Theta(\cdot)$, $o(\cdot)$ and $\omega(\cdot)$ will be used throughout. We say that $f(n) = O(g(n))$ if there exists $K > 0$ and $N \in \mathbb{N}$ such that $f(n) \leq Kg(n)$ for all $n > N$. We say that $f(n) = \Omega(g(n))$ if there exists $K > 0$ and $N \in \mathbb{N}$ such that $f(n) \geq Kg(n)$ for all $n > N$. We say that $f(n) = \Theta(g(n))$ if $f = O(g(n))$ and $f(n) = \Omega(g(n))$. In addition, $f(n) = o(g(n))$ if $f(n) = O(g(n))$ and $f(n) \neq \Theta(g(n))$. Finally, $f(n) = \omega(g(n))$ if $f(n) = \Omega(g(n))$ and $f(n) \neq \Theta(g(n))$.

Let X be a random variable with alphabet $\mathcal{X}$. For simplicity, our exposition in this chapter is (mostly) for the case when the alphabet $\mathcal{X}$ is finite, but extensions to the countably infinite case (and uncountable case) are usually straightforward. Let $P(x)$ be a probability mass function (pmf), i.e., $P(x) = \Pr(X = x)$ for all $x \in \mathcal{X}$. We will often denote the pmf by P instead of the more cumbersome P_X with the knowledge that $P(x)$ refers to the pmf associated to random variable X. Thus, the argument specifies the random variable. As mentioned, the majority of the definitions in this chapter also apply to continuous random variables with an associated probability density function

(pdf) $p(x)$. We use the term *distribution* to mean either the pmf (density with respect to the counting measure) or the pdf (density with respect to the Lebesgue measure). We omit the analogous information-theoretic quantities for continuous random variables for brevity. Important differences between the discrete case and continuous case will be highlighted.

The expectation operator is denoted as $\mathbb{E}[\cdot]$. When we want to make the expectation with respect to (wrt) a distribution P explicit, we will instead write $\mathbb{E}_P[\cdot]$. The variance of X and covariance between X and Y are denoted as $\mathrm{Var}(X)$ and $\mathrm{Cov}(X, Y)$ respectively. The *probability simplex* over the alphabet $\mathcal{X}$ is denoted as

$$\mathcal{P}(\mathcal{X}) := \left\{ P \in \mathbb{R}^{|\mathcal{X}|} : P(x) \geq 0, \sum_{a \in \mathcal{X}} P(a) \right\}. \tag{2.1}$$

Thus, all distributions over $\mathcal{X}$ belong to $\mathcal{P}(\mathcal{X})$. We say that $\mathcal{X}$ is the *support* of P.

Throughout the thesis, log will mean logarithm to the base e. Thus, the information-theoretic quantities stated will have units in nats.

■ 2.1.2 Entropy and Conditional Entropy

Definition 2.1. *The* entropy $H(X)$ *of a discrete random variable with pmf* P *is defined as*

$$H(X) := -\sum_{a \in \mathcal{X}} P(a) \log P(a). \tag{2.2}$$

More frequently than not, we denote the entropy as $H(P)$, i.e., we make the dependence of the entropy on the pmf explicit. The entropy can also be written as an expectation:

$$H(X) = \mathbb{E}_P \log \frac{1}{P(X)}. \tag{2.3}$$

The entropy is a measure of the randomness or uncertainty of a random variable X. It has many important operational interpretations. For instance, the entropy $H(P)$ of a random variable X with distribution P is a lower bound on the average length of the shortest description of the random variable. The asymptotic equipartition property (AEP) also states that most sample n-sequences of an ergodic process have probability about $\exp(-nH(X))$ and there are about $\exp(nH(X))$ such *typical sequences*. We will frequently exploit the latter fact in the subsequent development for learning graphical models. We now state an important property of entropy for discrete random variables.

Lemma 2.1. $0 \leq H(P) \leq \log |\mathcal{X}|$.

The continuous analog of (2.2) is known as the differential entropy (see Chapter 8 in [47]). Note that unlike the entropy defined for pmfs (or discrete distributions), the differential entropy can be negative.

The joint entropy and conditional entropies can be defined in exactly the same way.

Definition 2.2. *The* joint entropy $H(X,Y)$ *of discrete random variables* X, Y *with joint pmf* $P_{X,Y}$ *is defined as*

$$H(X,Y) := -\sum_{(a,b)\in\mathcal{X}\times\mathcal{Y}} P_{X,Y}(a,b)\log P_{X,Y}(a,b). \tag{2.4}$$

As mentioned previously, we will usually use the notation $H(P_{X,Y})$ in place of $H(X,Y)$ to make the dependence on the joint distribution $P_{X,Y}$ explicit. There is nothing new in the definition of the joint entropy. One can easily see that (2.4) reduces to (2.2) by considering (X,Y) to be a random variable defined on the alphabet $\mathcal{X}\times\mathcal{Y}$. The definition of the conditional entropy is more subtle.

Definition 2.3. *The* conditional entropy $H(X|Y)$ *of discrete random variables* X, Y *with joint pmf* $P_{X,Y}$ *is defined as*

$$H(X|Y) := -\sum_{(a,b)\in\mathcal{X}\times\mathcal{Y}} P_{X,Y}(a,b)\log P_{X|Y}(a|b). \tag{2.5}$$

Note that the expectation is taken over the joint distribution $P_{X,Y}$. In other words,

$$H(X|Y) = \mathbb{E}_{P_{X,Y}} \log \frac{1}{P(X|Y)} \tag{2.6}$$

and we abuse notation to say that $H(P_{X|Y}) = H(X|Y)$ with the understanding that the conditional entropy also depends on P_X. From the above definitions, it is also clear that

$$\begin{aligned} H(X|Y) &= \sum_{b\in\mathcal{Y}} P_Y(b) H(X|Y=b) \\ &= -\sum_{b\in\mathcal{Y}} P_Y(b) \sum_{a\in\mathcal{X}} P_{X|Y}(a|b)\log P_{X|Y}(a|b). \end{aligned} \tag{2.7}$$

We now state a few useful and simple properties of the various entropy functional introduced in this section.

Lemma 2.2. (Chain rule for entropy) $H(X,Y) = H(X) + H(Y|X)$.

Lemma 2.3. (Conditioning reduces entropy) $H(X|Y) \le H(X)$.

■ 2.1.3 Maximum Entropy and Exponential Families

In this section, we introduce an important class of distributions, known as exponential families, by appealing to the *maximum entropy principle* [104]. The principle of maximum entropy states that, subject to a set of constraints, the probability distribution which best represents the current state of knowledge is the one with largest entropy. In essence, this is the most random distribution, and reflects the maximum uncertainty about the quantities of interest.

Let $\mathbf{t} = (t_1, \ldots, t_K) : \mathcal{X} \to \mathbb{R}^K$ be a vector-valued statistic of X. That is $\mathbf{t}$ is a (measurable) function of X. Let $\mathbf{c} = (c_1, \ldots, c_K) \in \mathbb{R}^K$ be a constant vector. Then the *maximum entropy distribution* P_{ME} is given by

$$P_{\text{ME}} := \underset{P \in \mathcal{P}(\mathcal{X})}{\operatorname{argmax}} \; H(P) \quad \text{subject to} \quad \mathbb{E}_P t_k(X) = c_k, \quad k = 1, \ldots, K. \tag{2.8}$$

The constraint set $\{P \in \mathcal{P}(\mathcal{X}) : \mathbb{E}_P \mathbf{t}(X) = \mathbf{c}\}$ is called a *linear family*.

Lemma 2.4. (Maximum Entropy Distribution) *The maximum entropy distribution, if it exists, has the form*

$$P_{\text{ME}}(x) = \frac{\exp\left(\sum_{k=1}^K \theta_k t_k(x)\right)}{\sum_{a \in \mathcal{X}} \exp\left(\sum_{k=1}^K \theta_k t_k(a)\right)}, \tag{2.9}$$

where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K) \in \mathbb{R}^K$ *is a constant vector chosen to satisfy the constraints in* (2.8).

Lemma 2.4 can be proven using Lagrange multipliers and can be found for example in [110]. Expressed differently the maximum entropy distribution, parameterized by $\boldsymbol{\theta}$ can be written as

$$P_{\text{ME}}(x) = P(x; \boldsymbol{\theta}) = \exp\left(\boldsymbol{\theta}^T \mathbf{t}(x) - \Phi(\boldsymbol{\theta})\right) \tag{2.10}$$

where the *log-partition function* (also called *cumulant generating function*)

$$\Phi(\boldsymbol{\theta}) := \log \left[\sum_{a \in \mathcal{X}} \exp\left(\boldsymbol{\theta}^T \mathbf{t}(a)\right) \right]. \tag{2.11}$$

The family of distributions

$$\mathtt{E}(\mathbf{t}) := \left\{ P(x; \boldsymbol{\theta}) \propto \exp\left(\sum_{k=1}^K \theta_k t_k(x) \right), x \in \mathcal{X} \right\} \tag{2.12}$$

in which each element is parameterized by $\boldsymbol{\theta}$ as in (2.10) is called a K-parameter (linear) exponential family with *natural statistic* $\mathbf{t}(\cdot)$. See Bernando and Smith [17] or Barndorff-Nielsen [13] for more details on exponential families and their properties. The parameter $\boldsymbol{\theta}$ is known as the *natural parameter* (or *exponential parameter*) of the family. Lemma 2.4 says that maximum entropy distributions are members of exponential families. The log-partition function, a central quantity in statistical physics, has many appealing properties including those stated in the following lemma.

Lemma 2.5. (Properties of Log-Partition Function) *The derivatives of* $\Phi(\boldsymbol{\theta})$ *satisfy*

$$\frac{\partial \Phi}{\partial \theta_k} = \mathbb{E}_P t_k(X), \qquad \frac{\partial^2 \Phi}{\partial \theta_j \partial \theta_k} = \operatorname{Cov}_P(t_j(X), t_k(X)). \tag{2.13}$$

Definition 2.4. *The* Fisher information matrix in X about $\boldsymbol{\theta}$, *denoted as* $\mathbf{F}_{\boldsymbol{\theta}}$, *is the Hessian matrix of the log-partition function* $\partial^2\Phi/\partial\boldsymbol{\theta}^2$. *Furthermore,*[1]

$$\mathbf{F}_{\boldsymbol{\theta}} = \mathbb{E}_P\left[\left(\frac{\partial}{\partial\boldsymbol{\theta}}\log P(X;\boldsymbol{\theta})\right)\left(\frac{\partial}{\partial\boldsymbol{\theta}}\log P(X;\boldsymbol{\theta})\right)^T\right] \tag{2.14}$$

$$= \mathbb{E}_P\left[-\frac{\partial^2}{\partial\boldsymbol{\theta}^2}\log P(X;\boldsymbol{\theta})\right]. \tag{2.15}$$

The Fisher information matrix, a fundamental quantity in estimation theory [199], can be interpreted as a measure of curvature: it measures, on average, how "peaky" $\log P(x;\boldsymbol{\theta})$ is as a function of $\boldsymbol{\theta}$. The Cramer-Rao lower bound [199] states that the (inverse of the) Fisher information matrix serves as a lower bound on the variance of any unbiased estimator of $\boldsymbol{\theta}$. Intuitively, the "larger" the Fisher information, the better one can do at estimating $\boldsymbol{\theta}$ from data.

■ 2.1.4 Relative Entropy and Mutual Information

The relative entropy[2] is a measure of the "distance" between two distributions. In statistics, it arises as an expected logarithm of the likelihood ratio and thus, the error exponent for the asymptotics of binary hypothesis testing as we discuss in the next section. In source coding, the relative entropy $D(P\,||\,Q)$ can also be interpreted as a measure of the inefficiency (in terms of code length) of assuming that the distribution is Q when in fact, the true distribution is P.

Definition 2.5. *The* relative entropy *or* KL-divergence *between two distributions* $P(x)$ *and* $Q(x)$ *is defined as*

$$D(P\,||\,Q) = \sum_{a\in\mathcal{X}} P(a)\log\frac{P(a)}{Q(a)}. \tag{2.16}$$

The convention we use[3] is the following: $0\log(0/0) = 0$, $0\log(0/q) = 0$ and $p\log(p/0) = \infty$. Thus, $D(P\,||\,Q)$ is finite if and only if (iff) P is absolutely continuous wrt Q. In other words, if there exists $a \in \mathcal{X}$ such that $P(a) > 0$ and $Q(a) = 0$, then the KL-divergence $D(P\,||\,Q) = \infty$. For the majority of this thesis, we will assume that the distributions are positive everywhere wrt the alphabet $\mathcal{X}$, i.e., $P(a) > 0$ for all $a \in \mathcal{X}$. The extension of the definition of KL-divergence in (2.16) to continuous random variables is straightforward and we refer the reader to [47, Chapter 8].

The relative entropy is not symmetric (and hence is not a metric) but satisfies the following property, known as the *information inequality*.

[1] We assume in (2.14) and (2.15) that the derivatives of the log-likelihood $\log P(x;\boldsymbol{\theta})$ exist.

[2] The relative entropy is also called KL-divergence (for Kullback-Leibler divergence [125]), information divergence and minimum discrimination information [33].

[3] The convention can be justified by continuity arguments.

Lemma 2.6. (Information inequality) *The relative entropy is non-negative, i.e,*

$$D(P \,||\, Q) \geq 0. \tag{2.17}$$

Equality in (2.17) *holds iff* $P = Q$.

The convexity of relative entropy plays a crucial role in the subsequent development.

Lemma 2.7. (Convexity of relative entropy) *The relative entropy is jointly convex in* (P, Q)*, i.e, for every* $\lambda \in [0, 1]$ *and two pairs of distributions* (P, Q) *and* (P', Q')*,*

$$D(\lambda P + (1-\lambda)P' \,||\, \lambda Q + (1-\lambda)Q') \leq \lambda D(P \,||\, Q) + (1-\lambda) D(P' \,||\, Q'). \tag{2.18}$$

The relative entropy also satisfies a version of the chain rule.

Lemma 2.8. (Chain rule for relative entropy) *For two joint distribution* $P_{X,Y}$ *and* $Q_{X,Y}$*, the relative entropy satisfies*

$$D(P_{X,Y} \,||\, Q_{X,Y}) = D(P_{X|Y} \,||\, Q_{X|Y}) + D(P_Y \,||\, Q_Y) \tag{2.19}$$

where the conditional KL-divergence is defined as

$$D(P_{X|Y} \,||\, Q_{X|Y}) := \mathbb{E}_{P_{X,Y}} \log \frac{P_{X|Y}(X|Y)}{Q_{X|Y}(X|Y)}. \tag{2.20}$$

Note that the expectation is over the joint distribution $P_{X,Y}$. We also state a result (to be used in Chapter 5) that relates the relative entropy to the ℓ_1 norm of the difference between the distributions (also known as the *total variation distance*).

Lemma 2.9. (Pinsker's Inequality [76]) *Let*

$$\|P - Q\|_1 := \sum_{a \in \mathcal{X}} |P(a) - Q(a)| \tag{2.21}$$

be the ℓ_1 *norm of the difference between* P *and* Q*. Then,*

$$D(P \,||\, Q) \geq \frac{1}{2 \log 2} \|P - Q\|_1^2. \tag{2.22}$$

We will also frequently make use of approximations of the relative entropy functional. One such well-known connection of the KL-divergence is to the Fisher information matrix $\mathbf{F}_{\boldsymbol{\theta}}$, introduced in (2.14). Specifically, if $P(\cdot; \boldsymbol{\theta})$ and $P(\cdot; \boldsymbol{\theta}')$ are members of the same exponential family $\mathtt{E}(\mathbf{t})$ with (vector-valued) natural parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ respectively, then

$$D(P(\cdot; \boldsymbol{\theta}) \,||\, P(\cdot; \boldsymbol{\theta}')) = \frac{1}{2} (\boldsymbol{\theta} - \boldsymbol{\theta}')^T \mathbf{F}_{\boldsymbol{\theta}} (\boldsymbol{\theta} - \boldsymbol{\theta}') + o(\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2). \tag{2.23}$$

By regarding pmfs as vectors in $\mathbb{R}^{|\mathcal{X}|}$, Borade and Zheng [26] derived another useful approximation of $D(P\,||\,Q)$:

$$D(P\,||\,Q) = \frac{1}{2}\|P-Q\|_P^2 + o(\|P-Q\|^2). \tag{2.24}$$

where the weighted Euclidean norm in (2.24) is defined as $\|\mathbf{y}\|_{\mathbf{w}}^2 := \sum_i y_i^2/w_i$. In fact, the subscript P in (2.24) can be changed to any distribution in the vicinity of P and Q, i.e.,

$$D(P\,||\,Q) = \frac{1}{2}\|P-Q\|_{P_0}^2 + o(\|P-Q\|^2), \tag{2.25}$$

if $P_0 \approx P \approx Q$. This approximation is valid in the sense that the difference between $D(P\,||\,Q)$ and the quadratic approximation is small as compared to the magnitude of the true KL-divergence. *Euclidean information theory* [26] uses the approximations in (2.24) and (2.25) to simplify a variety of difficult problems in information theory. For example, it provides a single-letter characterization of the noisy broadcast channel problem in the so-called very-noisy scenario. We will use (2.24) and (2.25) as well as variants thereof to simplify expressions in the very-noisy (learning) regime.

We now introduce the notion of the mutual information between two random variables X and Y. The mutual information is a measure of the amount of information X has about Y.

Definition 2.6. *Consider two random variables X an Y with joint pmf $P_{X,Y} \in \mathcal{P}(\mathcal{X}\times\mathcal{Y})$. The* mutual information $I(X;Y)$ *is the relative entropy between $P_{X,Y}$ and the product distribution $P_X P_Y$, i.e.,*

$$I(X;Y) := D(P_{X,Y}\,||\,P_X P_Y) = \sum_{(a,b)\in\mathcal{X}\times\mathcal{Y}} P_{X,Y}(a,b)\log\frac{P_{X,Y}(a,b)}{P_X(a)P_Y(b)}. \tag{2.26}$$

From the definition and Lemma 2.6, we see that mutual information is also non-negative and equals to zero iff $P_{X,Y} = P_X P_Y$, implying that X and Y are independent. As with entropy, we will frequently use an alternative notation for mutual information. The notation $I(P_{X,Y}) = I(X;Y)$ makes the dependence on the joint distribution explicit. The mutual information can be written in terms of the entropy and conditional entropy in the following way:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{2.27}$$

That is, $I(X;Y)$ denotes the *reduction* in the uncertainty of X given Y. The conditional mutual information $I(X;Y|Z)$ is defined analogously, i.e.,

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z). \tag{2.28}$$

There is also a corresponding chain rule for mutual information, which is used to prove the data processing inequality and Fano's inequality.

Lemma 2.10. (Chain rule for mutual information) *The mutual information of X and (Y,Z) can be decomposed as*

$$I(X;Y,Z) = I(X;Y) + I(X;Z|Y). \tag{2.29}$$

■ 2.1.5 Data Processing Inequalities

Data processing inequalities are a family of inequalities that roughly state that information necessarily decreases (more precisely, does not increase) if one processes a random variable. We state two versions of this inequality in this section. To state the mutual information form of the inequality, we need the notion of a (discrete-time) Markov chain.

Definition 2.7. *Random variables X, Y and Z with joint pmf $P_{X,Y,Z}$ are said to form a* Markov chain *in that order (denoted $X - Y - Z$) if X and Z are independent given Y. That is, the joint pmf can be written as*

$$P_{X,Y,Z}(x, y, z) = P_X(x)P_{Y|X}(y|x)P_{Z|Y}(z|y). \tag{2.30}$$

The notion of graphical models, which is introduced in the later part of this chapter, is simply a generalization of Markov chains to arbitrary undirected graphs. The data processing inequality is stated as follows.

Theorem 2.11. (Data processing inequality: Mutual information form) *If $X - Y - Z$, then*

$$I(X; Y) \geq I(X; Z) \tag{2.31}$$

with equality iff $I(X; Y|Z) = 0$, i.e., X and Y are conditionally independent given Z.

An alternative form of the data processing inequality can be stated as follows: Let $\mathcal{X}, \mathcal{Y}$ be two (finite) sets. If $P_X, Q_X \in \mathcal{P}(\mathcal{X})$ are two distributions and $W_{Y|X}$ is a conditional distribution, then we can define another pair of distributions

$$P_Y(y) := \sum_{x \in \mathcal{X}} W_{Y|X}(y|x)P_X(x), \qquad Q_Y(y) := \sum_{x \in \mathcal{X}} W_{Y|X}(y|x)Q_X(x). \tag{2.32}$$

Note that $P_Y, Q_Y \in \mathcal{P}(\mathcal{Y})$. Then, we have the following KL-divergence form of the data processing inequality.

Theorem 2.12. (Data processing inequality: KL-divergence form) *Assume the setup above. Then*

$$D(P_X \,||\, Q_X) \geq D(P_Y \,||\, Q_Y). \tag{2.33}$$

Equality holds iff $W_{Y|X}(y|x)$ is deterministic channel $g(x)$ and for every $b \in \mathcal{Y}$, the ratio $P_X(a)/Q_X(a) = k_b$ is a constant for all $a \in g^{-1}(b) := \{a \in \mathcal{X} : g(a) = b\}$.

That is, processing (via a noisy channel $W_{Y|X}$) cannot increase discriminability. In fact, one can show that the data processing inequality holds for all so-called Ali-Silvey distances [14] of which KL-divergence is a special case. A special case of Theorem 2.12 will prove to be more useful in the sequel. Let $P_{X,Y}, Q_{X,Y} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$.

Corollary 2.13. (Data processing inequality: KL-divergence form) *Assuming the setup above, we have*

$$D(P_{X,Y} \,||\, Q_{X,Y}) \geq D(P_X \,||\, Q_X) \tag{2.34}$$

with equality iff

$$P_{X,Y} = P_X \cdot W_{Y|X}, \qquad Q_{X,Y} = Q_X \cdot W_{Y|X}, \tag{2.35}$$

i.e., the conditional distributions are identical.

■ 2.1.6 Fano's Inequality

Suppose that X and Y are two correlated random variables. We would like to guess the value of X given the random variable Y. It can easily be seen that if $H(X|Y) = 0$, i.e., the random variable $X = g(Y)$ where $g(\cdot)$ is some deterministic function, then we can estimate X with zero error probability. A natural question to ask is how well can one do in estimating X if $H(X|Y) > 0$. Fano's inequality provides a lower bound on the error probability in terms of the conditional entropy. We now describe the setup precisely.

There are two random variables X, Y with joint distribution $P_{X,Y}$. Suppose we wish to estimate the unknown random variable X with pmf P_X. We observe Y which is related to X via the conditional distribution $P_{Y|X}$. From Y, we form an estimate $\widehat{X}$ which takes values in the same alphabet $\mathcal{X}$. A key observation is that $X - Y - \widehat{X}$ is a Markov chain and so we can appeal to the data processing inequality in Theorem 2.11. The *probability of error* $p_{\text{err}} := \Pr(\widehat{X} \neq X)$. Then the form of Fano's inequality that we need is stated below.

Theorem 2.14. (Fano's inequality) *For any $\widehat{X}$ such that $X - Y - \hat{X}$, with p_{err} defined above, we have*

$$p_{\text{err}} \geq \frac{H(X|Y) - 1}{\log |\mathcal{X}|}. \tag{2.36}$$

Typically lower bounds of error probabilities are harder to prove and so Fano's inequality is often the only tool in proving converses in information theory. We employ Fano's inequality on several occasions in the sequel.

■ 2.2 The Method of Types and Asymptotics

The method of types, pioneered by Csiszár and Körner [49, 50], is a powerful technique for analyzing the asymptotic behavior of *types*, or *empirical distributions*. By using simple combinatorial tools, the method of types simplifies many proofs in the study of large deviations, which is a theory for the quantification of probabilities of rare events. In this section, we state some standard results from the method of types and use these results in the context of hypothesis testing and maximum likelihood estimation.

The results in this section are used to evaluate the error exponents associated to learning graphical models from data in Chapters 3, 4 and 5. The notion of a salient

set, defined in Chapter 7, is motivated by the Chernoff-Stein lemma which is stated in this section. The exposition here is based on Cover and Thomas [47], Van Trees [199] and Van der Vaart [63].

■ 2.2.1 The Method of Types

In this section, the following notation will be adopted. Let $X_1, \ldots, X_n$ be a sequence of i.i.d. random variables drawn from some distribution $P = P_X \in \mathcal{P}(\mathcal{X})$, where $\mathcal{X}$ is a finite set. We use the notation X_1^n to denote the sequence of variables. Usually this will be abbreviated as X^n. We also use the notation $x^n = (x_1, \ldots, x_n)$ to denote a sequence of n symbols (realizations) of the random variables X^n.

Definition 2.8. *The* type *or* empirical distribution *of a sequence x^n is the relative proportion of occurrences of each symbol in $\mathcal{X}$, i.e.,*

$$\widehat{P}(a; x^n) = \frac{1}{n} \sum_{k=1}^{n} \mathbb{I}\{x_k = a\} \tag{2.37}$$

where[4] *$\mathbb{I}\{x_k = a\}$ is equal to 1 if $x_k = a$ and 0 otherwise.*

The type is clearly a pmf, i.e., $\sum_{a \in \mathcal{X}} \widehat{P}(a; x^n) = 1$. In addition, we frequently abbreviate the notation for the type to be $\widehat{P}(a) = \widehat{P}(a; x^n)$. That is, the dependence on the sequence is suppressed. The type serves as an estimate of the distribution P. Indeed, for any n, we have

$$\mathbb{E}_P[\widehat{P}(\cdot; X^n)] = P(\cdot), \quad \forall\, a \in \mathcal{X}. \tag{2.38}$$

Definition 2.9. *The* set of types with denominator n, *$\mathcal{P}_n(\mathcal{X}) \subset \mathcal{P}(\mathcal{X})$ is the set of all possible types for sequences of length n generated from an alphabet $\mathcal{X}$.*

Lemma 2.15. (Cardinality of types) *The cardinality of the set of types with denominator n is*

$$|\mathcal{P}_n(\mathcal{X})| = \binom{n + |\mathcal{X}| - 1}{|\mathcal{X}| - 1}. \tag{2.39}$$

Furthermore, a convenient upper bound of $|\mathcal{P}_n(\mathcal{X})|$ is $(n+1)^{|\mathcal{X}|}$.

This result states that there are only *polynomially* many types.

Lemma 2.16. (Denseness of types [59]) *For any pmf $Q \in \mathcal{P}(\mathcal{X})$,*

$$\min_{P \in \mathcal{P}_n(\mathcal{X})} \|P - Q\|_1 \leq \frac{|\mathcal{X}|}{n}, \tag{2.40}$$

where recall that $\|P - Q\|_1$ is the ℓ_1 norm between the probability vectors P and Q defined in (2.21).

[4] The notation $\mathbb{I}\{\text{statement}\}$ denotes the indicator function. It is equal to 1 (resp. 0) if the statement is true (resp. false).

Lemma 2.16 states that for sufficiently large n, the sets $\mathcal{P}_n(\mathcal{X})$ approximate uniformly and arbitrarily well (in the sense of the ℓ_1 norm) any probability measure in $\mathcal{P}(\mathcal{X})$. Thus the union of the sets of n-types $\mathcal{P}_n(\mathcal{X})$ is dense in the simplex $\mathcal{P}(\mathcal{X})$, i.e., $\mathrm{cl}(\cup_{n\in\mathbb{N}}\mathcal{P}_n(\mathcal{X})) = \mathcal{P}(\mathcal{X})$.

Definition 2.10. *The type class of a distribution $Q \in \mathcal{P}(\mathcal{X})$ is defined as*

$$\mathsf{T}(Q) := \{x^n \in \mathcal{X}^n : \widehat{P}(\cdot; x^n) = Q\}. \tag{2.41}$$

That is, the type class of Q consists of all length-n sequences so that the empirical distribution of each sequence equals Q. Note that the type class $\mathsf{T}(Q)$ depends clearly on n but that dependence is suppressed for notational convenience. Also, if $Q \in \mathcal{P}(\mathcal{X})$ is not a type with denominator n, then the type class $\mathsf{T}(Q)$ may be empty. If for instance, $Q(a) \notin \mathbb{Q}$ for some $a \in \mathcal{X}$, i.e., some coordinate of Q is irrational, then $\mathsf{T}(Q) = \emptyset$. We now state some well-known results on the probability and the size of a type class.

Lemma 2.17. (Size of type class $\mathsf{T}(Q)$) *For any type $Q \in \mathcal{P}_n(\mathcal{X})$, the size of the type class satisfies*

$$\frac{1}{(n+1)^{|\mathcal{X}|}} \exp(nH(Q)) \leq |\mathsf{T}(Q)| \leq \exp(nH(Q)). \tag{2.42}$$

This says that the size of any type class is *exponential* in n with the exponent given roughly by $H(Q)$, the entropy of the pmf Q. Note that, in contrast, the number of types is only polynomial in n (cf. Lemma 2.15). For any set $\mathcal{A} \subset \mathcal{X}^n$, the P^n-probability of $\mathcal{A}$ is simply defined as

$$P^n(\mathcal{A}) := \sum_{x^n \in \mathcal{A}} P^n(x^n). \tag{2.43}$$

Lemma 2.18. (Probability of type class $\mathsf{T}(Q)$) *For any type $Q \in \mathcal{P}_n(\mathcal{X})$ and any distribution P, the probability of the type class $\mathsf{T}(Q)$ under the product measure P^n is bounded by*

$$\frac{1}{(n+1)^{|\mathcal{X}|}} \exp(-nD(Q\,||\,P)) \leq P^n(\mathsf{T}(Q)) \leq \exp(-nD(Q\,||\,P)). \tag{2.44}$$

This says that the probability of the type class of Q is exponentially unlikely under $P \neq Q$. More precisely, the exponent is given by the KL-divergence $D(Q\,||\,P)$. Thus, the farther apart the distributions, in terms of divergence, the more unlikely a sequence drawn from P "looks like" Q.

At this point, it is useful to introduce asymptotic notation so that the subsequent results appear cleaner but no less precise. We say that two positive sequence $\{a_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$ are *equal to first order in the exponent* if

$$\lim_{n\to\infty} \frac{1}{n} \log \frac{a_n}{b_n} = 0. \tag{2.45}$$

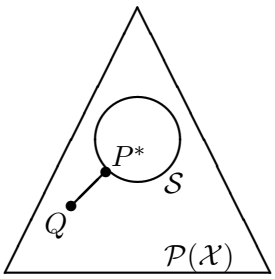

Figure 2.1. Illustration of Sanov's theorem. Note that P^* is the I-projection of Q onto the set $\mathcal{S}$.

In this case, we write $a_n \doteq b_n$. If instead

$$\limsup_{n\to\infty} \frac{1}{n} \log \frac{a_n}{b_n} \leq 0, \tag{2.46}$$

then we write $a_n \dot{\leq} b_n$. The notation $\dot{\geq}$ is defined analogously.

Using this notation, the results in (2.42) and (2.44) can be re-expressed as

$$|\mathsf{T}(Q)| \doteq \exp(nH(Q)) \tag{2.47}$$

$$P^n(\mathsf{T}(Q)) \doteq \exp(-nD(Q \,||\, P)). \tag{2.48}$$

We will also frequently use the so-called "largest-exponent-wins" principle [62]. This says that if $a_n \doteq \exp(nA)$ and $b_n \doteq \exp(nB)$ for constants $A, B \in \mathbb{R}$, then the sequence with the larger exponent dominates, i.e.,

$$a_n + b_n \doteq \exp(n \max\{A, B\}). \tag{2.49}$$

By induction, this simple relation be extended to a finite number of sequences. The "largest-exponent-wins" principle will be useful in upper bounding error probabilities in the sequel.

The relations in (2.47) and (2.48) allow us to quantify precisely the asymptotic behavior of long sequences.

■ 2.2.2 Large Deviations and Sanov's Theorem

The study of large deviations is concerned with the quantification of the probabilities of rare events. In this section, we state a useful theorem that states precisely the likelihood that a long i.i.d. sequence X^n drawn from Q has a type $\widehat{P}(\cdot; X^n)$ that belongs to a set $\mathcal{S} \subset \mathcal{P}(\mathcal{X})$ whose closure does not contain the generating distribution Q, i.e., $Q \notin \mathrm{cl}(\mathcal{S})$.

If samples X^n are drawn i.i.d. from a probability distribution Q, the probability of a type class decays exponentially fast as seen in (2.48). Since there are at most a polynomial number of type classes, the exponential that corresponds to the type "closest" (in the KL-divergence sense) to Q dominates the sum. See Fig. 2.1. The following theorem, attributed to Sanov [171] formalizes this reasoning.

Theorem 2.19. (Sanov's Theorem) *Let $\mathcal{S} \subset \mathcal{P}(\mathcal{X})$ be a measurable set in the probability simplex over $\mathcal{X}$. Let X^n be a sequence of i.i.d. random variables drawn from some distribution $Q \in \mathcal{P}(\mathcal{X})$. Then for every $n \in \mathbb{N}$,*

$$Q^n(\widehat{P}(\cdot; X^n) \in \mathcal{S}) \leq (n+1)^{|\mathcal{X}|} \exp(-nD(P^* \,||\, Q)) \tag{2.50}$$

where

$$P^* := \operatorname*{argmin}_{P \in \mathcal{S}} D(P \,||\, Q). \tag{2.51}$$

Furthermore, if $\mathcal{S}$ is the closure of its interior[5]*, then the following lower bound holds:*

$$\liminf_{n\to\infty} \frac{1}{n} \log Q^n(\widehat{P}(\cdot; X^n) \in \mathcal{S}) \geq -D(P^* \,||\, Q). \tag{2.52}$$

Recall that $\widehat{P}(\cdot; X^n)$ denotes the type of the sequence of random variables X^n, i.e., it is also a sequence of random variables. Note that (2.50) holds for all n. If we take the normalized logarithm and the lim sup in n, then the following also holds

$$\limsup_{n\to\infty} \frac{1}{n} \log Q^n(\widehat{P}(\cdot; X^n) \in \mathcal{S}) \leq -D(P^* \,||\, Q). \tag{2.53}$$

Furthermore, if $\mathcal{S} = \mathrm{cl}(\mathrm{int}(\mathcal{S}))$, then using (2.52) and (2.53), we see that the limit exists and

$$\lim_{n\to\infty} \frac{1}{n} \log Q^n(\widehat{P}(\cdot; X^n) \in \mathcal{S}) = -D(P^* \,||\, Q). \tag{2.54}$$

Using the asymptotic notation described in the previous section, (2.54) can be written as

$$Q^n(\widehat{P}(\cdot; X^n) \in \mathcal{S}) \doteq \exp(-nD(P^* \,||\, Q)). \tag{2.55}$$

That is, if the set $\mathcal{S}$, roughly speaking, does not have any isolated points, that the Q^n-probability of a type belonging to $\mathcal{S}$ is exponentially small and the (Chernoff) exponent is characterized by P^* in (2.51). The optimizing distribution P^* in (2.51) is known as the *I-projection* [51] of the distribution Q onto the set $\mathcal{S}$. Note that P^* does not necessarily have to be a type.

The exponent in (2.55)

$$J_{\mathcal{S}}(Q) := \min_{P \in \mathcal{S}} \; D(P \,||\, Q) \tag{2.56}$$

is also known as the *rate function* in the theory of large deviations. It depends on both the set $\mathcal{S}$ and the true generating distribution Q. Usually, the set $\mathcal{S}$ in Theorem 2.19 is defined as the preimage of some continuously differentiable function $f : \mathcal{P}(\mathcal{X}) \to \mathbb{R}$. In this case, Sanov's theorem says that for every (measurable) $\mathcal{A} \subset \mathbb{R}$,

$$Q^n(f(\widehat{P}(\cdot; X^n)) \in \mathcal{A}) \doteq \exp\left(-n \inf_{P \in \mathcal{P}(\mathcal{X})} \{D(P \,||\, Q) : f(P) \in \mathcal{A}\}\right). \tag{2.57}$$

[5] A set $\mathcal{S} = \mathrm{cl}(\mathrm{int}(\mathcal{S}))$ that satisfies such a topological property is said to be a *regular-closed* set. The interior is with respect to the topology in the probability simplex manifold and not the ambient space $\mathbb{R}^{|\mathcal{X}|}$. Thus, the interior should be referred to as the *relative interior* [28].

As usual, the infimum of an empty set is defined to be $+\infty$. The relation in (2.57) is a special case of the so-called *contraction principle* [59, Theorem 4.2.1] in the theory of large deviations.

■ 2.2.3 Asymptotics of Hypothesis Testing

In this section, we are concerned with the binary hypothesis problem [157, 199] and in particular, the performance of the optimum test when the number of observations n is large. The setup is as follows. Let X^n be a sequence of i.i.d. random variables drawn from $Q(x)$. We consider the two simple[6] hypotheses:

$$H_0 : Q = P_0, \qquad H_1 : Q = P_1. \tag{2.58}$$

To avoid trivialities, it is always assumed that $P_0 \neq P_1$. Traditionally, H_0 and H_1 are known as the *null* and *alternative* hypotheses respectively. Based on a realization of X^n, we would like to decide whether $Q = P_0$ or $Q = P_1$. Thus, we would like to design a *decision rule*, i.e., a function $\widehat{H} : \mathcal{X}^n \to \{H_0, H_1\}$. Corresponding to this decision rule is an *acceptance region*

$$A_n := \{x^n \in \mathcal{X}^n : \widehat{H}(x^n) = H_0\}. \tag{2.59}$$

That is, if we use the rule prescribed by the function $\widehat{H}(\cdot)$, then $x^n \in A_n$ denotes a decision in favor of H_0. The performance of the decision rule $\widehat{H}(\cdot)$ is measured by two error probabilities: The probability of false alarm is defined as

$$\alpha_n := P_0^n(A_n^c) \tag{2.60}$$

where $A_n^c := \mathcal{X}^n \setminus A_n$ denotes the *rejection region*. The probability of mis-detection is

$$\beta_n := P_1^n(A_n). \tag{2.61}$$

In statistics, α_n and β_n are known as the type-I and type-II error probabilities respectively. We use the terms false-alarm probability and type-I error probability interchangeably. The terms mis-detection probability and type-II error probability will also be used interchangeably. We now state the Neyman-Pearson lemma [146] which provides the optimum test for the hypothesis problem in (2.58).

Lemma 2.20. (Neyman-Pearson) *Assume the setup as above. For any $\eta > 0$, define the acceptance region*

$$A_n^*(\eta) := \left\{ x^n \in \mathcal{X}^n : l(x^n) := \frac{P_0^n(x^n)}{P_1^n(x^n)} > \eta \right\}. \tag{2.62}$$

Let $\alpha_n^* := P_0^n(\mathcal{X}^n \setminus A_n^*(\eta))$ and $\beta_n^* := P_1^n(A_n^*(\eta))$ be the type-I and type-II error probabilities respectively. Let B_n be any other acceptance region with type-I and type-II error probabilities α_n and β_n respectively. If $\alpha_n \leq \alpha_n^*$, then $\beta_n \geq \beta_n^*$.

[6] The hypotheses in (2.58) are called *simple* because each hypothesis only involves a single distribution (P_0 and P_1). The study of *composite hypothesis testing* is vast and we refer the reader to the book by Lehman [129] for details.

Thus, the *likelihood ratio test* (LRT) in (2.62) is the optimum test in the Neyman-Pearson sense. The LRT minimizes the type-II error over all acceptance regions with the same (or lower) type-I error probability.

Despite its obvious appeal, Lemma 2.20 lacks symmetry. In the Bayesian setup, one seeks to find acceptance regions to minimize the overall probability of error:

$$p_{\text{err}}^{(n)} := \Pr(H_0)\alpha_n + \Pr(H_1)\beta_n. \tag{2.63}$$

Here, $\Pr(H_0)$ and $\Pr(H_1)$ are the prior probabilities of hypothesis H_0 and H_1 respectively. The following lemma states the decision region that minimizes $p_{\text{err}}^{(n)}$.

Lemma 2.21. *Given a priori probabilities* $\Pr(H_0)$ *and* $\Pr(H_1)$*, probability distributions* P_0 *and* P_1 *as well as the data* x^n*, the minimum probability of acceptance region takes the form:*

$$A_n^* := \left\{ x^n \in \mathcal{X}^n : l(x^n) := \frac{P_0^n(x^n)}{P_1^n(x^n)} > \frac{\Pr(H_1)}{\Pr(H_0)} \right\} \tag{2.64}$$

i.e., the decision is in favor of H_0 *if the likelihood ratio* $l(x^n)$ *exceeds the threshold* $\eta := \Pr(H_1)/\Pr(H_0)$*.*

Hence, similar to the Neyman-Pearson setup, the LRT is again the optimum test in Bayesian hypothesis testing.

We now examine the asymptotics of hypothesis testing, i.e., we study the asymptotic decay of the error probabilities α_n and β_n as n becomes large. We first consider the Neyman-Pearson setup where the type-I error is kept fixed below some constant (size) $\epsilon > 0$. The following lemma is attributed to Chernoff and Stein [38].

Lemma 2.22. (Chernoff-Stein) *Let* $A_n \subset \mathcal{X}^n$ *be an acceptance region in favor of* H_0 *for the hypothesis test in* (2.58)*. Let the corresponding type-I and type-II error probabilities be defined as in* (2.60) *and* (2.61) *respectively. Then for any* $\epsilon > 0$*, define*

$$\beta_n^\epsilon := \inf_{A_n \subset \mathcal{X}^n} \{\beta_n : \alpha_n < \epsilon\} \tag{2.65}$$

to be the optimum type-II error probability over all acceptance regions such that the type-I error probability is below a fixed size $\epsilon > 0$*. Then*

$$\beta_n^\epsilon \doteq \exp(-nD(P_0 \,||\, P_1)). \tag{2.66}$$

This result says that the exponent governing the decay of the type-II error probability is $D(P_0 \,||\, P_1)$. Thus, if the hypotheses are well-separated, $D(P_0 \,||\, P_1)$ is large and the rate of decay of the mis-detection probability will also be large.

The Chernoff-Stein lemma lacks symmetry because the type-I error is kept below ϵ and the type-II error decays exponentially fast in n, i.e., $\beta_n \doteq \exp(-nD(P_0 \,||\, P_1))$. Often times, we may want to minimize the overall error probability, i.e., we would like ensure that *both* mis-detection and false alarm error probabilities tend to zero

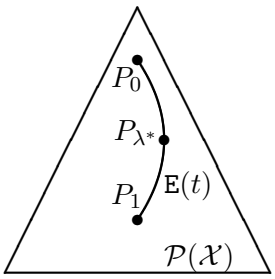

Figure 2.2. The distribution P_{λ^*} satisfies (2.68). It is equidistant from P_0 and P_1. The exponential family joining P_0 and P_1 has natural statistic $t(x) = \log P_1(x)/P_0(x)$.

exponentially fast. We now state the corresponding asymptotic result for the Bayesian case where the overall probability of error (2.63) is to be minimized. Define

$$D^*(P_0, P_1) := \lim_{n\to\infty} \inf_{A_n \subset \mathcal{X}^n} -\frac{1}{n} \log p_{\text{err}}^{(n)}. \tag{2.67}$$

Lemma 2.23. (Chernoff) *The exponent of the probability of error is D^* where*

$$D^* = D^*(P_0, P_1) = D(P_{\lambda^*} \,||\, P_0) = D(P_{\lambda^*} \,||\, P_1), \tag{2.68}$$

and where

$$P_\lambda(x) = \frac{P_0^\lambda(x) P_1^{1-\lambda}(x)}{\sum_{a\in\mathcal{X}} P_0^\lambda(a) P_1^{1-\lambda}(a)} \tag{2.69}$$

and λ^* is chosen so that the equality in (2.68) *holds.*

The exponent in (2.68), known as the *Chernoff information*, can also be easily shown to be

$$D^*(P_0, P_1) = -\min_{\lambda\in[0,1]} \log \left(\sum_{a\in\mathcal{X}} P_0^\lambda(a) P_1^{1-\lambda}(a) \right). \tag{2.70}$$

Thus, the exponent governing the rate of decay of the overall error probability is D^* in (2.70). The distribution P_{λ^*} is the distribution along the one-parameter exponential family (with the log-likelihood ratio $\log P_1(x)/P_0(x)$ as the natural statistic) connecting P_0 and P_1 such that the equality in (2.68) holds. See Fig. 2.2.

Finally, we mention that there have been many other works on asymptotics of hypothesis testing for general (non i.i.d.) sources, e.g., Han and Kobayashi [94], Han [92] and Iriyama [103]. Also see the book by Han [93] which provides a unified perspective. There are also many results for the hypothesis testing in the minimax setting. See for example Levitan and Merhav [130] and Feder and Merhav [75]. Finally, the asymptotic optimality of the generalized LRT and the Hoeffding test for composite hypothesis testing are discussed in Zeitouni et al. [221] and Hoeffding [99].

■ 2.2.4 Asymptotics of Parameter Estimation

In this section, we study the asymptotics of parameter estimation, specifically the behavior of the maximum likelihood (ML) estimate as the number of samples becomes large. The results here are used in Chapter 5 where we study the rate of convergence of the risk of an estimated forest-structured distribution to the forest projection. The material presented here can be found in greater depth in the book by Van der Vaart [63].

Let X be a random variable that is distributed according to $P_X := P(x; \boldsymbol{\theta})$, i.e., the distribution is parameterized by a continuous vector parameter $\boldsymbol{\theta} \in \Theta \subset \mathbb{R}^m$. Given a sequence of i.i.d. random variables X^n drawn from an arbitrary member of the family $P(x; \boldsymbol{\theta})$, the ML estimate is defined as the parameter $\boldsymbol{\theta}$ that maximizes the likelihood of the data, i.e.,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) := \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} \; P^n(X^n; \boldsymbol{\theta}). \tag{2.71}$$

We can rewrite (2.71) as the following:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) = \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} \; L_n(\boldsymbol{\theta}; X^n). \tag{2.72}$$

where the *normalized log-likelihood* $L_n(\boldsymbol{\theta}; X^n)$, viewed as a function of $\boldsymbol{\theta}$ is defined as

$$L_n(\boldsymbol{\theta}; X^n) := \frac{1}{n} \sum_{k=1}^{n} \log P(X_k; \boldsymbol{\theta}). \tag{2.73}$$

Since $L_n(\boldsymbol{\theta}; X^n)$ is a function of the sequence of random variables X^n drawn from $P(x; \boldsymbol{\theta})$, it is also a sequence of random variables. In addition, it is easy to rewrite the likelihood in the following way:

$$L_n(\boldsymbol{\theta}; X^n) = \sum_{a \in \mathcal{X}} \widehat{P}(a; X^n) \log P(a; \boldsymbol{\theta}) \tag{2.74}$$

$$= \mathbb{E}_{\widehat{P}(\cdot; X^n)} \log P(\cdot; \boldsymbol{\theta}). \tag{2.75}$$

Note that in (2.74), we overload notation to mean that $\widehat{P}(\cdot; X^n)$ is the type given X^n and $P(\cdot; \boldsymbol{\theta})$ is a distribution parameterized by $\boldsymbol{\theta}$. By noticing that $L_n(\boldsymbol{\theta}; X^n) + H(\widehat{P}(\cdot; X^n)) = -D(\widehat{P}(\cdot; X^n) \,||\, P(\cdot; \boldsymbol{\theta}))$, the ML estimator can be written as a *reverse I-projection* [51]:

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) := \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta} \; D(\widehat{P}(\cdot; X^n) \,||\, P(\cdot; \boldsymbol{\theta})). \tag{2.76}$$

In contrast to the usual I-projection [51] in (2.51), the minimization is over the second argument in the relative entropy function. Thus, the search for the maximum likelihood estimator can also be viewed as a minimization of the KL-divergence over an appropriate set. This observation will be useful in variety of scenarios in the rest of the thesis starting with the development of the Chow-Liu algorithm in Section 2.5.2.

It is known that under certain (fairly weak) conditions [63], the maximum likelihood estimator is *consistent*, i.e., for every $\epsilon > 0$,

$$\lim_{n\to\infty} P^n(\|\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) - \boldsymbol{\theta}\| > \epsilon) = 0. \tag{2.77}$$

Thus, the ML estimate $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n)$ converges in probability to the true parameter $\boldsymbol{\theta}$, i.e., $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) \xrightarrow{\mathrm{p}} \boldsymbol{\theta}$. In fact it can also be shown that the ML estimator converges almost surely to the true parameter $\boldsymbol{\theta}$ under some stronger regularity conditions.

The following theorem quantifies the asymptotic behavior of the ML estimator. In order to state the theorem, we introduce the notation $\xrightarrow{\text{a.s.}}$ and $\xrightarrow{\mathrm{d}}$ to denote almost sure convergence and convergence in distribution respectively [19].

Theorem 2.24. (Asymptotics of the Maximum Likelihood Estimator) *Let $P(x;\boldsymbol{\theta})$ be a family of distributions parameterized by $\boldsymbol{\theta}$ and let X be a random variable distributed according to a particular member in the family $P(x;\boldsymbol{\theta}_0)$. Let $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n)$, defined in (2.73), be an estimate of $\boldsymbol{\theta}$ based on n i.i.d. samples X^n that corresponds to a local maximum of the likelihood function. Then under some mild conditions (continuous, bounded derivatives of $P(x;\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$; bounded expectations of the derivatives), we have*

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) \xrightarrow{\text{a.s.}} \boldsymbol{\theta}_0, \tag{2.78}$$

$$\sqrt{n}(\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) - \boldsymbol{\theta}_0) \xrightarrow{\mathrm{d}} \mathcal{N}(\mathbf{0}, \mathbf{F}_{\boldsymbol{\theta}_0}^{-1}), \tag{2.79}$$

where $\mathcal{N}(\mathbf{0}, \mathbf{F}_{\boldsymbol{\theta}_0}^{-1})$ is a Gaussian probability density function with zero mean and covariance matrix $\mathbf{F}_{\boldsymbol{\theta}_0}^{-1}$.

The proof of this result involves the strong law of large numbers and the central limit theorem. See [63] for the details. The matrix $\mathbf{F}_{\boldsymbol{\theta}_0}$, defined in (2.14), is the Fisher information matrix in X about $\boldsymbol{\theta}_0$. From (2.79), we see that the larger the Fisher information, the better we are able to estimate the value of the true parameter $\boldsymbol{\theta}_0$ from a given set of observations. Indeed, from (2.79), we see that if $\mathbf{F}_{\boldsymbol{\theta}_0}$ is "large", then the number of samples required to drive the variance of the ML estimator to below a prespecified level is smaller than if $\mathbf{F}_{\boldsymbol{\theta}_0}$ is "small". From (2.79), we also observe that the ML estimator is *asymptotically normal*. The variance of the estimator asymptotically satisfies the so-called Cramer-Rao lower bound (CRLB) [113]. Therefore, ML estimation is *asymptotically efficient*[7] under the usual regularity conditions.

We now find it convenient to define some stochastic order notation in order to simplify the results in the sequel. We say that a sequence of random variables $Y_n = O_p(g_n)$ (for some deterministic positive sequence $\{g_n\}$) if for every $\epsilon > 0$, there exists a $B = B_\epsilon > 0$ such that $\limsup_{n\to\infty} \Pr(|Y_n| > Bg_n) < \epsilon$. Thus, $\Pr(|Y_n| > Bg_n) \geq \epsilon$

[7] An estimator that achieves the CRLB is said to be *efficient*. See Kay [113] or Van Trees [199].

holds for only finitely many n. Equipped with this notation, the relation in (2.79) can be rewritten alternatively as

$$\|\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n) - \boldsymbol{\theta}_0\|_q = O_p\left(\frac{1}{\sqrt{n}}\right), \tag{2.80}$$

where $\|\cdot\|_q$ is any ℓ_q norm. That is, the difference between ML estimator $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}(X^n)$ and the true parameter $\boldsymbol{\theta}_0$ decays at a rate of $1/\sqrt{n}$.

■ 2.3 Supervised Classification and Boosting

Supervised classification is a prototypical problem in machine learning. We will mainly focus on binary classification in this and subsequent sections (e.g., Chapter 6). The setup and objective of the binary classification problem are largely similar to binary hypothesis testing discussed in Section 2.2.3. Both problems seek to classify objects into two groups. The main difference between classification and hypothesis testing is that in the former, the underlying distributions (denoted as P_0 and P_1 in Section 2.2.3) are unknown. Rather, a finite set of *samples* is provided in order for the (binary) decision to be made.

More specifically, a *training dataset* $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ is provided to a learner. Each sample $(\mathbf{x}_l, y_l)$ consists of a *measurement vector* $\mathbf{x}_l \in \Omega \subset \mathbb{R}^d$ and a binary *label*[8] $y_l \in \{-1, +1\}$. It is assumed that each sample $(\mathbf{x}_l, y_l)$ is drawn from some *unknown* joint distribution[9] $p(\mathbf{x}, y)$. The problem of binary classification involves finding a function, called a *classifier*, $\widehat{H}(\cdot) : \Omega \to \{-1, +1\}$ that optimizes some objective, for example, the probability of error $\Pr(\widehat{H}(\mathbf{X}) \neq Y)$. Note that it is not possible, in general, to minimize the probability of error (also called *generalization error*) $\Pr(\widehat{H}(\mathbf{X}) \neq Y)$ because we do not have access to the true distribution $p(\mathbf{x}, y)$ from which the samples were generated. In practice, the classifier $\widehat{H}$ is selected from a function class $\mathcal{H}$ to minimize a loss function $l : \{-1, +1\}^2 \to \mathbb{R}^+$ of the training data, i.e.,

$$\widehat{H}(\cdot) = \underset{H(\cdot)\in\mathcal{H}}{\operatorname{argmin}} \frac{1}{n} \sum_{j=1}^{n} l(y_l, H(\mathbf{x}_l)). \tag{2.81}$$

Some commonly used loss functions include the zero-one loss, the hinge loss and the exponential loss. See the book by Hastie et al. [96] or the recent thesis by Varshney [203] for more details. The rest of this section is devoted to a review of some common classification techniques and a detailed description of *boosting* [79], a general method to combine classifiers to improve the overall classification accuracy.

[8] The label y_l corresponds to the hypotheses H_0, H_1 in the binary hypothesis testing context in (2.58).

[9] We abuse notation and denote the joint distribution of $\mathbf{X}$ and Y as $p(\mathbf{x}, y)$ suppressing the dependence of p on both $\mathbf{X}$ and Y. Also, the marginal of p wrt $\mathbf{X}$ may be either a pmf of pdf depending on whether Ω is a continuous space or a discrete space. The marginal of p wrt Y is a pmf with support $\{-1, +1\}$.

2.3.1 Some Commonly Used Classifiers

This section reviews of some common classification techniques used in the machine learning literature as benchmarks. One of the earliest classifiers was the *perceptron* introduced by F. Rosenblatt [162]. The perceptron is a linear classifier, that is

$$\widehat{H}(\mathbf{x}) = \begin{cases} 1 & \mathbf{v}^T\mathbf{x} + b > 0 \\ -1 & \mathbf{v}^T\mathbf{x} + b \leq 0, \end{cases} \tag{2.82}$$

for some weight vector $\mathbf{v}$ and some constant b. Traditional classification techniques such as artificial neural networks [20], classification and regression trees (CART) [31], random forests [30] as well as support vector machines [46] have become popular benchmarks for more advanced classifiers such as the geometric level set (GLS) classifier [205] developed recently by Varshney and Willsky. In addition, it is also worth mentioning that the Naïve Bayes classifier [65] is a simple and commonly used probabilistic classifier based on applying Bayes' theorem with the assumption that the features (elements of the random vector $\mathbf{X}$) are conditionally independent given the label Y. Even though this assumption is strong, the Naïve Bayes classifier has been shown to perform well on real datasets [65]. A comprehensive empirical comparison of these and other classification techniques can be found in [35].

2.3.2 Boosting and AdaBoost

In this section, we review the AdaBoost algorithm introduced by Freund and Schapire in 1995 [79]. The exposition here is based on the review article by the same authors [80]. This iterative algorithm takes as inputs the training dataset $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. One of the key features in this algorithm is the maintaining of a set of weights over the training set. The weight of this distribution on training example l at iteration t is denoted as $w_t(l)$. Initially, all the weights are set equally. AdaBoost calls a weak or base learning algorithm (classifier) repeatedly in a sequence of iterations $t = 1, \ldots, T$. At each iteration t, the weights of incorrectly classified samples are increased so that the weak classifier is forced to focus on the harder examples in the training set. The goodness of a weak classifier $h_t : \Omega \to \{-1, +1\}$ is measured by its error based on the current set of weights on the training samples

$$\epsilon_t := \sum_{l:h_t(\mathbf{x}_l) \neq y_l} w_t(l). \tag{2.83}$$

The complete algorithm is summarized in the following steps:

1. Initialize $t = 1$ and $w_1(l) = 1/n$ for all $l = 1, \ldots, n$.
2. At iteration t, train weak classifier weak classifier $h_t \in \mathcal{H}$ using the distribution w_t and calculate the error ϵ_t as in (2.83).
3. Choose $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$. Note that $\alpha_t \geq 0$ because we can assume that $\epsilon_t \leq 1/2$ without loss of generality.

4. Update the weights as

$$w_{t+1}(l) = \frac{w_t(l)}{Z_t} \times \begin{cases} \exp(-\alpha_t) & h_t(\mathbf{x}_l) = y_l \\ \exp(\alpha_t) & h_t(\mathbf{x}_l) \neq y_l, \end{cases} \tag{2.84}$$

where $Z_t := \sum_{l=1}^{n} w_t(l) \exp(-\alpha_t y_l h_t(\mathbf{x}_l))$ is a normalization constant to ensure $\sum_{l=1}^{n} w_{t+1}(l) = 1$.

5. Increment $t \leftarrow t+1$ and repeat steps 2 – 4 until $t = T$.

6. The final classifier is

$$\widehat{H}(\mathbf{x}) := \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right). \tag{2.85}$$

Step 4 says that incorrectly classified points receive larger weight in subsequent iterations because $\alpha_t \geq 0$. Observe that the final classifier, also called the *ensemble classifier*, in (2.85) is a weighted majority vote of the T weak classifiers $\{h_t\}_{t=1,\ldots,T}$. The coefficients α_t are the weights assigned to the weak classifier h_t.

The AdaBoost algorithm possesses many appealing properties and we state two such properties here. Firstly, the training error (fraction of misclassified samples) after T iterations can be upper bounded as [79]

$$\frac{1}{n}\sum_{l=1}^{n} \mathbb{I}\{\widehat{H}(\mathbf{x}_l) \neq y_l\} \overset{(a)}{\leq} \frac{1}{n}\sum_{l=1}^{n} \exp(-y_l \widehat{H}(\mathbf{x}_l)) \overset{(b)}{=} \prod_{t=1}^{T} Z_t \tag{2.86}$$

$$\overset{(c)}{=} \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} \overset{(d)}{=} \exp\left(-2\sum_{t=1}^{T} \gamma_t^2\right) \tag{2.87}$$

where $\gamma_t := 1/2 - \epsilon_t \geq 0$. The inequality in (a) holds by noting that $\mathbb{I}\{\widehat{H}(\mathbf{x}_l) \neq y_l\} \leq \exp(-y_l \widehat{H}(\mathbf{x}_l))$, i.e., the exponential loss is an upper bound on the zero-one loss. Equalities (b), (c) and (d) follow from the definitions of Z_t in Step 4, ϵ_t in (2.83) and $\gamma_t = 1/2 - \epsilon_t$ respectively. The conclusion in (2.87) means that the fraction of mistakes on the training set decays exponentially fast if each weak learner is better than random, i.e., there exists a $\gamma > 0$ such that $\gamma_t \geq \gamma$.

Secondly, the generalization error of the ensemble classifier can also be bounded in terms of the number of samples n, the number of iterations T and the VC-dimension[10] [23, 201] of the space of weak learners $C^{\mathrm{VC}}(\mathcal{H})$. More precisely, Freund and Schapire [79]

[10] We say that a set of classifiers $\mathcal{H}$ *shatters* a set of points $\mathbf{x}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ if we can classify the points in $\mathbf{x}^n$ in all possible ways. More precisely, for all 2^n labeling vectors $(y_1, \ldots, y_n) \in \{-1, +1\}^n$, there exists a function $h \in \mathcal{H}$ such that $h(\mathbf{x}_l) = y_l$ for all $l = 1, \ldots, n$. The *VC-dimension* (for Vapnik–Chervonenkis dimension) is defined as the cardinality of the largest set of points that the set of functions $\mathcal{H}$ can shatter. For example the set of linear functions in $\mathbb{R}^d$ has VC-dimension $d+1$. See the book by Vapnik [201] for details.

showed using techniques from Baum and Haussler [15] that[11]

$$\Pr(\widehat{H}(\mathbf{X}) \neq Y) \leq \frac{1}{n}\sum_{l=1}^{n}\mathbb{I}\{\widehat{H}(\mathbf{x}_l) \neq y_l\} + \widetilde{O}\left(\sqrt{\frac{TC^{\mathrm{VC}}(\mathcal{H})}{n}}\right). \tag{2.88}$$

Even though the second term in (2.88) suggests that the generalization error degrades with the number of rounds of boosting, it has been observed empirically that the generalization error remains small (close to the training error) even when T is in the thousands [29, 159]. We will compare and contrast the behavior of the weak classifier developed in Chapter 6 (based on tree-structured graphical models) to these empirical observations.

There are many extensions of the basic AdaBoost algorithm. The most straightforward multiclass generalization [79], called AdaBoost.M1, is usually adequate provided the weak learners are strong enough. In addition, Schapire and Singer [174] showed how AdaBoost can be extended to an algorithm called Real-AdaBoost to handle weak classifiers that output real values, also called confidence-rated predictions, i.e., $h_t : \Omega \to \mathbb{R}$. Thus, if $h_t(\mathbf{x}_l)$ is large and positive, this indicates that the sample $\mathbf{x}_l$ is more likely to be such that $y_l = +1$ as compared to the case when $h_t(\mathbf{x}_l)$ is small and positive. The magnitude of $h_t(\mathbf{x}_l)$ thus provide a measure of the learner's *confidence* in the prediction. We will make use of this version of AdaBoost in Chapter 6 where more details will be provided.

■ 2.4 Probabilistic Graphical Models

The majority of thesis is concerned with the analysis of learning of graphical models from data. This section is devoted to a brief introduction to graphical models. The treatment is limited to the scope of the thesis and is thus by no means comprehensive. The exposition here is based on [21, 108, 209], which provide a more thorough treatment of this subject.

Probabilistic graphical models bring together graph theory [213] and probability theory [19] to provide a diagrammatic representation of multivariate probability distributions. They offer several appealing properties. Firstly, they provide a simple way to visualize the structure of a probabilistic model in terms of its *factorization* properties. Secondly, the inspection of the model can be used to deduce various properties (such as conditional independence) of the collection of the random variables. Finally, complex operations, such as inference (marginalization or finding the maximum a-posteriori configuration), can be expressed in terms of graphical manipulations in which underlying mathematical operations can be interpreted as operations on graphs.

This section is subdivided into several subsections. We first provide a set of terminology from graph theory. Following that, we provide background on undirected graphical models (also known as Markov random fields). Finally, we focus our atten-

[11] The $\widetilde{O}(\cdot)$ suppresses dependences on small log factors.

tion on two classes of graphical models: tree-structured graphical models and Gaussian graphical models.

2.4.1 Undirected Graphs

This subsection collects the relevant terminology from graph theory that is used in the rest of the thesis. Many of the definitions are standard and can be found in for instance [95, 213]. Some of the terminology, however, is unique to this thesis.

A *undirected graph* $G = (V, E)$ consists of a set $V := \{1, \ldots, d\}$ of vertices (or nodes) and a set of edges $E \subset \binom{V}{2} := \{(i, j) : i, j \in V, i \neq j\}$. The set V is known as the vertex (or node) set and E is referred to as the *edge set* (or more informally, the *structure*). In this thesis, a graph is not allowed to contain self-loops. Since the graph G is an undirected one, the directionality of an edge does not matter, i.e., (i, j) and (j, i) refer to the same edge. A *subgraph* of a graph G is a graph $(V(F), E(F))$ such that $V(F) \subset V$ and $E(F) \subset E$. A *supergraph* of a graph G is a graph of which G is a subgraph.

The *neighborhood* of a node i is the set $\text{nbd}(i) := \{j \in V : (i, j) \in E\}$. The *closed neighborhood* is the set $\{i\} \cup \text{nbd}(i)$. Two vertices i and j are said to be *adjacent* if $i \in \text{nbd}(j)$. Two edges are *adjacent* if they have a common node. The *degree* of node i is the cardinality of the set $\text{nbd}(i)$. A *path* in a graph G is a subgraph $P = (V(P), E(P))$ such that $V(P) = \{v_1, v_2, \ldots, v_k\}$ (the vertices $\{v_i\}_{i=1}^k$ are distinct) and $E(P) = \{(v_1, v_2), \ldots, (v_{k-1}, v_k)\}$. We will sometimes abuse terminology to refer to the path as simply the edge set of P, i.e., $E(P)$, so a path can also mean a collection of edges. A *cycle* in a graph G is a subgraph $C = (V(C), E(C))$ such that $V(C) = \{v_1, v_2, \ldots, v_k\}$ (with $k \geq 3$) and $E(C) = \{(v_1, v_2), \ldots, (v_k, v_1)\}$. Again, we will sometimes refer to a cycle as the set of edges in C, i.e., $E(C)$. A graph is said to be *acyclic* if it does not contain any cycles.

The *distance* $d_G(i, j)$ between two (distinct) vertices i and j in a graph G is the length of a shortest path between them. The *diameter* of a graph G, $\text{diam}(G)$ is the maximum distance between any two nodes in the graph. If it is possible to establish a path from any vertex to any other vertex of a graph, the graph is said to be *connected*; otherwise, the graph is *disconnected*. A *clique* C is a fully connected subgraph of G, i.e., if $i, j \in C$, then $(i, j) \in E$. A *maximal clique* is one which is not properly contained in some other clique.

We will also find the notion of a line graph useful in Chapter 4. Given a graph G, its *line graph* $H = \mathcal{L}(G)$ is a graph such that each vertex of H represents an edge of G and two vertices of H are adjacent if and only if their corresponding edges are adjacent in G. See Fig. 2.4.

Trees

A *tree* $T = (V, E)$ is a connected acyclic graph. A vertex of degree one is called a *leaf*. A non-leaf is known as an *internal node*. A *subtree* of the tree T is a connected subgraph of T. A *forest* is an acyclic graph. A *proper forest* is a disconnected acyclic graph. A

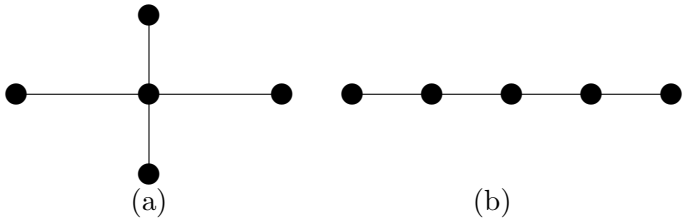

Figure 2.3. (a) A star (b) A chain

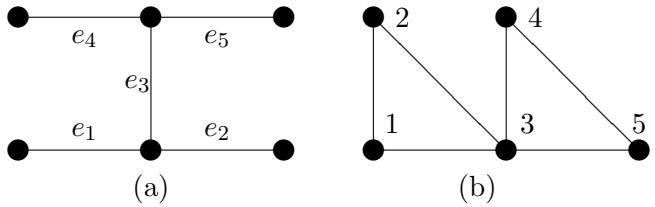

Figure 2.4. (a): A graph G. (b): The line graph $H = \mathcal{L}(G)$ that corresponds to G is the graph whose vertices are the edges of G (denoted as e_i) and there is an edge between any two vertices i and j in H if the corresponding edges in G share a node.

spanning tree is a spanning subgraph that is a tree. A *star* is a tree with one internal node and $|V| - 1$ leaves. A *chain* (or *path graph*) is a tree in which every vertex has degree one or two. See Fig. 2.3. The line graph of a path graph is another path graph with one fewer node.

We now state a collection of equivalent definitions and properties of trees which will be helpful in the sequel. A tree is an undirected graph $T = (V, E)$ that satisfies the following equivalent conditions:

- T is connected and acyclic.
- T is acyclic and a cycle is formed with the addition of one edge.
- T is connected and is not connected anymore if any edge is removed from it.
- Any two vertices in T is connected by a unique path.
- T has exactly $|V| - 1$ edges.

Let the set of trees with $d = |V|$ nodes be denoted as $\mathcal{T}^d$. The following theorem says that the total number of trees is superexponential in the number of nodes.

Theorem 2.25. (Cayley's formula [3]) *The number of trees on $d = |V|$ labeled vertices is $|\mathcal{T}^d| = d^{d-2}$.*

■ 2.4.2 Undirected Graphical Models

In order to define graphical models, we associate to each vertex $i \in V$ in a graph $G = (V, E)$ a random variable X_i that takes values in an alphabet (or set) $\mathcal{X}$. The alphabet may be finite (e.g., $\mathcal{X} = \{1, \ldots, r\}$) or infinite (e.g., $\mathcal{X} = \mathbb{R}$). Usually, we will abuse terminology and use the terms node, vertex and variable interchangeably. For instance, when we say two nodes are independent, we mean that the variables associated to those nodes are independent. For a subset $A \subset V$, the random vector X_A is defined as the collection of random variables $\{X_i : i \in A\}$ and the corresponding alphabet of X_A is simply the Cartesian product $\mathcal{X}^{|A|}$. The values that X_A takes on will be denoted as x_A. The distribution of the random vector $\mathbf{X} := (X_1, \ldots, X_d)$ is either a pmf $P(\mathbf{x})$ or pdf $p(\mathbf{x})$ depending on whether the set $\mathcal{X}$ is discrete or continuous. For simplicity, the presentation in this chapter, except for Section 2.4.4, will be for pmfs, i.e., we assume that $\mathcal{X}$ is countable.

An *undirected graphical model*[12] (or Markov random field) is a family of multivariate probability distributions where each distribution $P \in \mathcal{P}(\mathcal{X}^d)$ factorizes in accordance to an undirected graph $G = (V, E)$ where $|V| = \{1, \ldots, d\}$. The probability distribution factorizes according to the cliques in G. Let the set of maximal cliques in G be $\mathcal{C}$. We associate to each clique $C \in \mathcal{C}$ a *compatibility function* $\psi_C : \mathcal{X}^{|C|} \to [0, \infty)$. Each compatibility function is simply a local function only for elements x_C in the clique C. With this notation, an undirected graphical model is a family of distributions in which each distribution factorizes in the following specific way:

$$P(x_1, \ldots, x_d) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C). \tag{2.89}$$

The normalization constant (also called partition function)

$$Z := \sum_{(x_1, \ldots, x_d) \in \mathcal{X}^d} \prod_{C \in \mathcal{C}} \psi_C(x_C). \tag{2.90}$$

For a general undirected graphical model, the functions ψ_C do not have to be related to the marginal or conditional distributions of the set of variables in the clique C.

Example 2.1. *Define the vector* $\mathbf{x} = (x_1, \ldots, x_7)$*. An undirected graphical model on* G *as shown in Fig. 2.5 factorizes in the following way:*

$$P(\mathbf{x}) = \frac{1}{Z} \psi_{1,2,3,4}(x_1, x_2, x_3, x_4) \psi_{4,5,6,7}(x_4, x_5, x_6, x_7). \tag{2.91}$$

We now introduce the notion of *Markovianity*. A random vector $\mathbf{X} := (X_1, \ldots, X_d)$ is said to be *(locally) Markov* on a graph $G = (V, E)$ if its probability distribution $P \in \mathcal{P}(\mathcal{X}^d)$ satisfies:

$$P(x_i | x_{V \setminus \{i\}}) = P(x_i | x_{\mathrm{nbd}(i)}), \quad \forall\, i \in V, \tag{2.92}$$

[12] There is another class of graphical models known as *directed graphical models* or *Bayesian networks*. We will not require the notion of directed graphical models in this thesis. The interested reader is referred to [21], [127] and [209] for details.

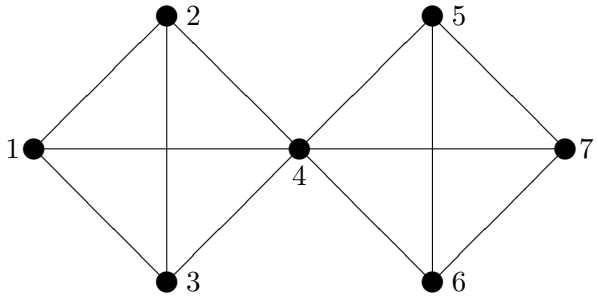

Figure 2.5. An undirected graph G. The set of maximal cliques is $\mathcal{C} = \{(1,2,3,4),(4,5,6,7)\}$. Undirected graphical models on G factorize as in (2.91).

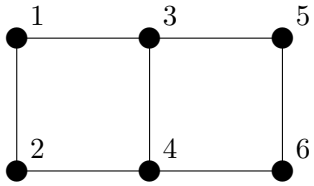

Figure 2.6. The node set $S = \{3,4\}$ separates the node sets $A = \{1,2\}$ and $B = \{5,6\}$.

where $\text{nbd}(i)$ is the set of neighbors of i in G. Eq. (2.92) is called the (local) Markov property and states that if random variable X_i is conditioned on its neighbors, then X_i is independent of the rest of the variables in the graph.

Given three non-overlapping subsets of vertices $A, B, S \subset V$, we say that S *separates* the node sets A and B if every path from a node $i \in A$ to another node $j \in B$ passes through at least one node in S. See Fig. 2.6. We say that the random vector $\mathbf{X}$ is *(globally) Markov* with respect to the graph G if, whenever node set S separates the node sets A and B, the subvectors X_A and X_B are independent conditioned on X_S i.e.,

$$P(x_A, x_B|x_S) = P(x_A|x_S)P(x_B|x_S). \tag{2.93}$$

The connection between graph structure and a joint distribution is not yet readily apparent. However, the celebrated Hammersley-Clifford theorem [91] provides a necessary and sufficient condition for the connection between a set of Markov properties (such as (2.92) and (2.93)) and a joint distribution. For strictly positive distributions, i.e, $Q(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^d$, the random vector $\mathbf{X}$ with distribution P is Markov on G (i.e., it satisfies (2.92) or (2.93)) if and only if the factorization (2.89) is satisfied for a collection of compatibility functions $\{\psi_C\}_{C\in\mathcal{C}}$.

■ 2.4.3 Tree-Structured Graphical Models

In this section, we introduce the class of tree-structured graphical models, i.e., probability distributions that factorize according to an undirected tree $T = (V, E)$. In the case of trees, the cliques are simply the nodes $i \in V$ and the edges $(i, j) \in E$. Thus,

the factorization in (2.89) specializes to the *pairwise* representation

$$P(x_1, \ldots, x_d) = \frac{1}{Z} \prod_{i \in V} \psi_i(x_i) \prod_{(i,j) \in E} \psi_{i,j}(x_i, x_j). \tag{2.94}$$

The negative logarithm of the compatibility functions $-\log \psi_i$ and $-\log \psi_{i,j}$ are called the *node potentials* and *edge potentials*, respectively. In fact, in the case of trees, a special case of the junction tree[13] theorem [48, 210] states that

$$P(x_1, \ldots, x_d) = \prod_{i \in V} P_i(x_i) \prod_{(i,j) \in E} \frac{P_{i,j}(x_i, x_j)}{P_i(x_i) P_j(x_j)}. \tag{2.95}$$

In other words, the compatibility functions can be reparameterized as $\psi_i(x_i) \propto P_i(x_i)$ and $\psi_{i,j}(x_{i,j}) \propto \frac{P_{i,j}(x_i, x_j)}{P_i(x_i) P_j(x_j)}$ where $\{P_i : i \in V\}$ and $\{P_{i,j} : (i,j) \in E\}$ are the sets of node and pairwise marginals of P respectively. The factorization in (2.95) also holds for the case when the P is Markov on some forest $F = (V, E)$. We denote the set of d-variate tree-structured distributions as $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d) \subset \mathcal{P}(\mathcal{X}^d)$, i.e.,

$$\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d) := \Bigg\{ P(\mathbf{x}) = \prod_{i \in V} P_i(x_i) \prod_{(i,j) \in E} \frac{P_{i,j}(x_i, x_j)}{P_i(x_i) P_j(x_j)} : \\ (V, E) \text{ is a tree}, |V| = d \Bigg\}. \tag{2.96}$$

The set of forest-structured distributions $\mathcal{D}(\mathcal{X}^d, \mathcal{F}^d)$ is defined analogously.

Example 2.2. *If the random vector* $\mathbf{X} = (X_1, X_2, X_3)$ *has distribution* P *and* P *is Markov on the chain* $1 - 2 - 3$*, then*

$$P(\mathbf{x}) = P_1(x_1) P_2(x_2) P_3(x_3) \frac{P_{1,2}(x_1, x_2)}{P_1(x_1) P_2(x_2)} \frac{P_{2,3}(x_2, x_3)}{P_2(x_2) P_3(x_3)} \tag{2.97}$$

$$= P_1(x_1) P_{2|1}(x_2|x_1) P_{3|2}(x_3|x_2). \tag{2.98}$$

Eq. (2.98) *is sometimes known as the* directed representation *of the chain.*

When $\mathcal{X}$ is a finite set, the number of parameters to fully describe a tree model is linear in d. Even though inference in graphical models is, in general, NP-hard [44], inference in tree models is tractable; the belief propagation or sum-product algorithm [122, 153] is exact on trees and the number of operations for computing the marginals is linear in d. In Chapters 3 and 4, we are interested to learn such models from i.i.d. data samples. The Chow-Liu algorithm [42] provides an efficient implementation for ML learning of tree-structured distributions from data. We describe the Chow-Liu algorithm in detail in Section 2.5.

[13] A *cluster tree* is a tree of clusters of variables which are linked via separators. These consists of the variables in the adjacent clusters. A cluster tree is a *junction tree* if for each pair of clusters C_γ and C_δ, all nodes in the path between C_γ and C_δ contain the intersection.

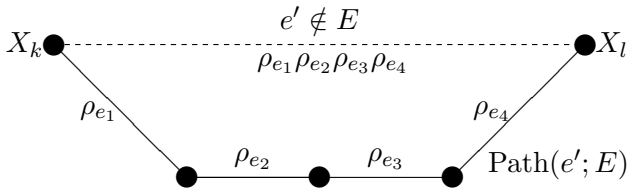

Figure 2.7. The correlation coefficient of two random variables X_k and X_l is the product of the correlation coefficients along its path $\mathrm{Path}(e'; E)$.

■ 2.4.4 Gaussian Graphical Models

In Chapter 4, we discuss the performance of the Chow-Liu algorithm for learning of tree-structured Gaussian graphical models [69, 167] from data. In this subsection, we state a few simple properties of such models.

A d-dimensional Gaussian pdf (or distribution) with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance matrix $\boldsymbol{\Sigma} \succ 0$ is given by

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}\right). \tag{2.99}$$

We use the notation $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ as a shorthand for (2.99). For Gaussian graphical models, it is known [127] that the fill-pattern of the *inverse covariance matrix* (also called precision matrix, information matrix or concentration matrix) $\boldsymbol{\Sigma}^{-1}$ encodes the structure of $p(\mathbf{x})$. More precisely, if $\mathbf{X} = (X_1, \ldots, X_d)$ is a random vector Markov on $G = (V, E)$ with distribution (pdf) $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $[\boldsymbol{\Sigma}^{-1}]_{i,j} = 0$ if and only if $(i, j) \notin E$. Thus, a sparse Gaussian graphical model, one whose graph has few edges, is such that the inverse covariance matrix is also sparse.

Let the covariance of any two random variables X_k and X_l be denoted $\mathrm{Cov}(X_k, X_l)$. Define the correlation coefficient of X_k and X_l as

$$\rho_{k,l} := \frac{\mathrm{Cov}(X_k, X_l)}{\sqrt{\mathrm{Var}(X_k)\mathrm{Var}(X_l, X_l)}}. \tag{2.100}$$

Then the following property characterizes the correlation coefficient between any two random variables in a *tree-structured Gaussian graphical model*. Given a non-edge $e' = (k, l) \notin E$, denote the unique path connecting nodes k and l in the T as $\mathrm{Path}(e'; E)$. See Fig. 2.7.

Lemma 2.26. (Markov property of Gaussian graphical models) *Let p be a Gaussian distribution Markov on a tree $T = (V, E)$ with the set of correlation coefficients $\{\rho_{k,l}\}_{k,l\in V}$. Let $e' = (k, l) \notin E$ be a non-edge in the tree T. Then the correlation coefficient $\rho_{k,l}$ is given as*

$$\rho_{k,l} = \prod_{(i,j)\in \mathrm{Path}(e';E)} \rho_{i,j}. \tag{2.101}$$

Proof. By induction, it suffices to proof the claim for the chain $X_k - X_s - X_l$, i.e., X_k and X_l are conditionally independent given X_s. Also assume without loss of generality

that $\mathbb{E}X_k = \mathbb{E}X_s = \mathbb{E}X_l = 0$. Denote the variance of X_k as $\sigma_k^2 := \mathrm{Cov}(X_k, X_k)$. Then we have

$$\begin{aligned}
\mathbb{E}[X_k X_l] &= \mathbb{E}[\mathbb{E}[X_k X_l | X_s]] && (2.102)\\
&= \mathbb{E}[\mathbb{E}[X_k|X_s] \cdot \mathbb{E}[X_k|X_s]] && (2.103)\\
&= \mathbb{E}\left[\left(\rho_{k,s}\frac{\sigma_k}{\sigma_s}X_s\right)\left(\rho_{s,l}\frac{\sigma_l}{\sigma_s}X_s\right)\right] && (2.104)\\
&= \rho_{k,s}\frac{\sigma_k}{\sigma_s}\rho_{s,l}\frac{\sigma_l}{\sigma_s}\mathbb{E}[X_s^2] && (2.105)\\
&= \rho_{k,s}\rho_{s,l}\sigma_l\sigma_k, && (2.106)
\end{aligned}$$

where (2.102) follows from iterated expectations [19], (2.103) follows from conditional independence, (2.104) follows from the fact [19] that for jointly Gaussian variables $\mathbb{E}[X_k|X_s] = \rho_{k,s}X_s\sigma_k/\sigma_s$, (2.106) follows from the definition of the variance. This completes the proof since (2.106) implies that $\rho_{k,l} = \rho_{k,s}\rho_{s,l}$. □

■ 2.5 Learning Graphical Models

This section is devoted to a review of the (vast) existing literature on methods and theoretical results on learning graphical models. It also includes a thorough description of the Chow-Liu algorithm for learning tree-structured graphical models.

■ 2.5.1 Review of Existing Work

The seminal work by Chow and Liu in [42] focused on learning tree models from data samples. The authors showed that the learning of the optimal tree distribution essentially decouples into two distinct steps: (i) a structure learning step and (ii) a parameter learning step. The structure learning step can be performed efficiently using a max-weight spanning tree (MWST) algorithm with the empirical mutual information quantities as the edge weights. The parameter learning step is a ML estimation procedure where the parameters of the learned model are equal to those of the empirical distribution. Chow and Wagner [43], in a follow-up paper, studied the consistency properties of the Chow-Liu algorithm for learning trees. They concluded that if the true distribution is Markov on a unique tree structure, then the Chow-Liu learning algorithm is consistent. This implies that as the number of samples tends to infinity, the probability that the learned structure differs from the (unique) true structure tends to zero.

Unfortunately, it is known that the exact learning of general graphical models is NP-hard [112], but there have been several works to learn approximate models. In the mid-1990s, Heckerman [97] proposed learning the structure of Bayesian networks by using the Bayesian Information Criterion [176] (BIC) to penalize more complex models and by putting priors on various structures. Meilă and Jordan [135] used the Expectation-Maximization algorithm [61] to learn mixtures of tree-structured distributions. Other authors used the maximum entropy principle or (sparsity-enforcing) ℓ_1 regularization

as approximate graphical model learning techniques. In particular, Dudik et al. [67] and Lee et al. [128] provide strong consistency guarantees on the learned distribution in terms of the log-likelihood of the samples. Johnson et al. [106] also used a similar technique known as maximum entropy relaxation (MER) to learn discrete and Gaussian graphical models. Wainwright et al. [211] proposed a regularization method for learning the graph structure based on ℓ_1 logistic regression and provided strong theoretical guarantees for learning the correct structure as the number of samples, the number of variables, and the neighborhood size grow. In a similar work, Meinshausen and Buehlmann [136] considered learning the structure of arbitrary Gaussian models using the Lasso [197]. They show that the error probability of learning the wrong structure, under some mild technical conditions on the neighborhood size, decays exponentially even when the size of the graph d grows with the number of samples n. However, the rate of decay is not provided explicitly. Zuk et al. [223] provided bounds on the limit inferior and limit superior of the error rate for learning the structure of Bayesian networks but, in contrast to our work, these bounds are not asymptotically tight. In addition, the work in Zuk et al. [223] is intimately tied to the BIC [176], whereas our analysis in Chapters 3 and 4 is for the Chow-Liu ML tree learning algorithm [42].

Recently Santhanam and Wainwright [172] and Bresler et al. [32] derived information-theoretic upper and lower bounds on the sample complexity for learning graphical models. Even more recently, Bento and Montanari [16] proved that if the graphical model possesses long range correlations (lack of correlation decay), then it is difficult to learn in terms of the sample complexity. However, Chechetka and Guestrin [37] developed good approximations for learning thin junction trees (junction trees where the sizes of the maximal cliques are small). The area of study in statistics known as *covariance selection* [57, 60] also has connections with structure learning in Gaussian graphical models. Covariance selection involves estimating the non-zero elements in the inverse covariance matrix and providing consistency guarantees of the estimate in some norm, e.g., the Frobenius norm in [164].

In two recent works that are closely related to the work presented in Chapter 5, Liu et al. [132] and Gupta et al. [89] derived consistency (and sparsistency) guarantees for learning tree and forest models. The pairwise joint distributions are modeled using kernel density estimates, where the kernels are Hölder continuous. This differs from the approach presented in Chapter 5 since it is assumed in our work that each variable can only take finitely many values, leading to stronger results on error rates for structure learning via the method of types. Furthermore, the algorithm suggested in both papers uses a subset (usually half) of the dataset to learn the full tree model and then uses the remaining subset to prune the model based on the log-likelihood on the held-out set. We suggest a more direct and consistent method based on thresholding, which uses the *entire* dataset to learn and prune the model without recourse to validation on a held-out dataset. It is well known that validation is both computationally expensive [21, pp. 33] and a potential waste of valuable data which may otherwise be employed to learn a better model. In [89], the problem of estimating forests with restricted component sizes

was considered and was proven to be NP-hard.

■ 2.5.2 The Chow-Liu algorithm

In this section, we review the classical Chow-Liu algorithm [42] for learning the ML tree distribution P_{ML} given a set of n samples $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ drawn i.i.d. from a tree-structured discrete distribution (pmf) $P \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$. Each sample $\mathbf{x}_l \in \mathcal{X}^d$ and the set $\mathcal{X}$ is finite. The extension to the Gaussian case where $\mathcal{X} = \mathbb{R}$ is considered in Chapter 4 and the development is very similar and so it is omitted here for the sake of brevity. The distribution P is assumed to be Markov on a tree $T_P = (V, E_P)$. The ML estimation problem is defined as

$$P_{\mathrm{ML}} := \operatorname*{argmax}_{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)} \log Q^n(\mathbf{x}^n). \tag{2.107}$$

Thus, P_{ML} is the tree-structured distribution that maximizes the likelihood of the samples over all possible tree-structured distributions in $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$. Let P_{ML} be Markov on the tree $T_{\mathrm{ML}} = (V, E_{\mathrm{ML}})$. Thus, the estimated edge set is E_{ML}. Note that since P_{ML} is a tree-structured distribution, from (2.95), it is completely specified by the structure E_{ML} and consistent pairwise marginals $P_{\mathrm{ML}}(x_i, x_j)$ on its edges $(i, j) \in E_{\mathrm{ML}}$.

Lemma 2.27. *The ML estimator in* (2.107) *is equivalent to the following optimization problem:*

$$P_{\mathrm{ML}} = \operatorname*{argmin}_{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)} D(\widehat{P} \,||\, Q), \tag{2.108}$$

where $\widehat{P}(\cdot) = \widehat{P}(\cdot; \mathbf{x}^n)$ *is the empirical distribution of* $\mathbf{x}^n$.

This result follows directly from (2.76). We now state the main result of the Chow-Liu tree learning algorithm [42].

Theorem 2.28. (Chow-Liu Tree Learning [42]) *The structure and parameters of the ML estimator* P_{ML} *in* (2.107) *are given by*

$$E_{\mathrm{ML}} = \operatorname*{argmax}_{E_Q : Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)} \sum_{e \in E_Q} I(\widehat{P}_e), \tag{2.109}$$

$$P_{\mathrm{ML}}(x_i, x_j) = \widehat{P}_{i,j}(x_i, x_j), \qquad \forall\, (i, j) \in E_{\mathrm{ML}}, \tag{2.110}$$

where $I(\widehat{P}_e) = I(\widehat{P}_{i,j})$ *is the mutual information of the empirical distribution* $\widehat{P}_e$.

Proof. For a fixed tree distribution $Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$, Q admits the factorization in (2.95), and we have

$$D(\widehat{P} \,||\, Q) + H(\widehat{P}) \tag{2.111}$$

$$= - \sum_{\mathbf{x} \in \mathcal{X}^d} \widehat{P}(\mathbf{x}) \log \left[\prod_{i \in V} Q_i(x_i) \prod_{(i,j) \in E_Q} \frac{Q_{i,j}(x_i, x_j)}{Q_i(x_i) Q_j(x_j)} \right], \tag{2.112}$$

$$= -\sum_{i\in V}\sum_{x_i\in\mathcal{X}} \widehat{P}_i(x_i)\log Q_i(x_i)$$
$$- \sum_{(i,j)\in E_Q}\sum_{(x_i,x_j)\in\mathcal{X}^2} \widehat{P}_{i,j}(x_i,x_j)\log\frac{Q_{i,j}(x_i,x_j)}{Q_i(x_i)Q_j(x_j)}. \tag{2.113}$$

For a fixed structure E_Q, it can be shown [42] that the above quantity is minimized when the pairwise marginals over the edges of E_Q are set to that of $\widehat{P}$, i.e., for all tree-structured distributions $Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$,

$$D(\widehat{P}\,||\,Q) + H(\widehat{P}) \tag{2.114}$$
$$\geq -\sum_{i\in V}\sum_{x_i\in\mathcal{X}} \widehat{P}_i(x_i)\log\widehat{P}_i(x_i)$$
$$- \sum_{(i,j)\in E_Q}\sum_{(x_i,x_j)\in\mathcal{X}^2} \widehat{P}_{i,j}(x_i,x_j)\log\frac{\widehat{P}_{i,j}(x_i,x_j)}{\widehat{P}_i(x_i)\widehat{P}_j(x_j)}. \tag{2.115}$$
$$= \sum_{i\in V} H(\widehat{P}_i) - \sum_{(i,j)\in E_Q} I(\widehat{P}_e). \tag{2.116}$$

The first term in (2.116) is a constant with respect to Q. Furthermore, since E_Q is the edge set of the tree distribution $Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$, the optimization for the ML tree distribution P_{ML} reduces to the MWST search for the optimal edge set as in (2.109). □

Hence, the optimal tree probability distribution P_{ML} is the reverse I-projection of $\widehat{P}$ onto the optimal tree structure given by (2.109), an MWST problem. Thus, the optimization problem in (2.108) essentially reduces to a search for the *structure* of P_{ML}. The structure of P_{ML} completely determines its distribution, since the parameters are given by the empirical distribution in (2.110). To solve (2.109), we use the samples $\mathbf{x}^n$ to compute the empirical distribution $\widehat{P}$, then use $\widehat{P}$ to compute $I(\widehat{P}_e)$, for each node pair $e \in \binom{V}{2}$. Subsequently, we use the set of empirical mutual information quantities $\{I(\widehat{P}_e) : e \in \binom{V}{2}\}$ as the edge weights for the MWST problem.[14]

Note that the search for the MWST is not the same as that for largest set of mutual information quantities as one has to take into consideration the spanning tree constraint.

We see that the Chow-Liu MWST spanning tree algorithm is an efficient way of solving the ML estimation problem, especially when the dimension d is large. This is because there are d^{d-2} possible spanning trees over d nodes (Theorem 2.25) ruling out the possibility for performing an exhaustive search for the optimal tree structure. In contrast, the MWST can be found, say using Kruskal's algorithm [45, 120] or Prim's algorithm [158], in $O(d^2 \log d)$ time.

[14] If we use the true mutual information quantities as inputs to the MWST, then the true edge set E_P is the output.

Chapter 3

Large Deviations for Learning Discrete Tree Models

■ 3.1 Introduction

IN Section 2.5.2, we saw that the implementation of the maximum likelihood (ML) estimation problem can be done efficiently via the Chow-Liu algorithm [42]. It is known that the ML estimator learns the distribution correctly asymptotically, and hence, is consistent [43].

While consistency is an important qualitative property for any estimator, the study of the rate of convergence, a precise quantitative property, is also of great practical interest. We are interested in the rate of convergence of the ML-estimator (Chow-Liu algorithm) for tree distributions as we increase the number of samples. Specifically, we study the rate of decay of the error probability or the error exponent of the ML-estimator in learning the *tree structure* of the unknown distribution. A larger exponent means that the error probability in structure learning decays more rapidly. In other words, we need relatively few samples to ensure that the error probability is below some fixed level $\delta > 0$. Such model are thus "easier" to learn. We address the following questions: Is there exponential decay of the probability of error in structure learning as the number of samples tends to infinity? If so, what is the exact error exponent, and how does it depend on the parameters of the distribution? Which edges of the true tree are most-likely to be in error; in other words, what is the nature of the most-likely error in the ML-estimator? We provide concrete and intuitive answers to the above questions, thereby providing insights into how the parameters of the distribution influence the error exponent associated with learning the structure of discrete tree distributions.

■ 3.1.1 Main Contributions

There are four main contributions in this chapter. First, using the large-deviation principle (LDP) [62] we prove that the most-likely error in ML-estimation is a tree which differs from the true tree by a single edge. Second, again using the LDP, we derive the exact error exponent for ML-estimation of tree structures. Third, we provide a succinct and intuitive closed-form approximation for the error exponent which is tight in the

very noisy learning regime, where the individual samples are not too informative about the tree structure. The approximate error exponent has a very intuitive explanation as the *signal-to-noise ratio* (SNR) for learning. Finally, our analyses and results are indeed more general: they extend to the case where the underlying distribution is not necessarily tree-structured. We show that it is also possible to use exactly the same large-deviation tools to analyze the case where the *tree-projection* (which may not be unique) is to be learned.

We analyze the *error exponent* (also called the inaccuracy rate) for the estimation of the structure of the unknown tree distribution. For the error event that the structure of the ML-estimator E_{ML} given n samples differs from the true tree structure E_P of the unknown distribution P, the error exponent is given by

$$K_P \;:=\; \lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}(\{E_{\mathrm{ML}} \neq E_P\}). \tag{3.1}$$

To the best of our knowledge, error-exponent analysis for tree-structure learning has not been considered before (See Section 2.5.1 for a brief survey of the existing literature on learning graphical models from data).

Finding the error exponent K_P in (3.1) is not straightforward since in general, one has to find the *dominant* error event with the *slowest* rate of decay among all possible error events [62, Ch. 1]. For learning the structure of trees, there are a total of $d^{d-2}-1$ possible error events,[1] where d is the dimension (number of variables or nodes) of the unknown tree distribution P. Thus, in principle, one has to consider the information projection [51] of P on all these error trees. This rules out brute-force information projection approaches for finding the error exponent in (3.1), especially for high-dimensional data.

In contrast, we establish that the search for the dominant error event for learning the structure of the tree can be limited to a polynomial-time search space (in d). Furthermore, we establish that this dominant error event of the ML-estimator is given by a tree which differs from the true tree by only a single edge. We provide a polynomial algorithm with $O(\mathrm{diam}(T_P)\, d^2)$ complexity to find the error exponent in (3.1), where $\mathrm{diam}(T_P)$ is the diameter of the tree T_P. We heavily exploit the mechanism of the ML Chow-Liu algorithm [42] for tree learning to establish these results, and specifically, the fact that the ML-estimator tree distribution depends *only* on the relative order of the empirical mutual information quantities between all the node pairs (and not their absolute values).

Although we provide a computationally-efficient way to compute the error exponent in (3.1), it is not available in closed-form. In Section 3.5, we use Euclidean information theory [25, 26] to obtain an approximate error exponent in closed-form, which can be interpreted as the signal-to-noise ratio (SNR) for tree structure learning. Numerical simulations on various discrete graphical models verify that the approximation is tight in the very noisy regime.

[1] Since the ML output E_{ML} and the true structure E_P are both spanning trees over d nodes and since there are d^{d-2} possible spanning trees [213], we have $d^{d-2}-1$ number of possible error events.

In Section 3.6, we extend our results to the case when the true distribution P is not a tree. In this case, given samples drawn independently from P, we intend to learn the *optimal tree-projection* P^* onto the set of trees. Importantly, if P is not a tree, there may be several trees that are optimal projections [43] and this requires careful consideration of the error events. We derive the error exponent even in this scenario.

■ 3.1.2 Chapter Outline

This paper is organized as follows: In Section 3.2, we state the system model and the problem statement. In Section 3.3, we derive an analytical expression for the crossover rate of two node pairs. We then relate the crossover rates to the overall error exponent in Section 3.4. We also discuss some connections of the problem we solve here with robust hypothesis testing. In Section 3.5, we leverage on ideas in Euclidean information theory to state sufficient conditions that allow approximations of the crossover rate and the error exponent. We obtain an intuitively appealing closed-form expression. By redefining the error event, we extend our results to the case when the true distribution is not a tree in Section 3.6. We compare the true and approximate crossover rates by performing numerical experiments for a given graphical model in Section 3.7. Conclusions for this chapter are provided in Section 3.8.

■ 3.2 System Model and Problem Statement

In this chapter, we consider a learning problem, where we are given a set of n i.i.d. d-dimensional samples $\mathbf{x}^n := \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from an unknown distribution $P \in \mathcal{P}(\mathcal{X}^d)$, which is Markov with respect to a tree $T_P \in \mathcal{T}^d$. Each sample or observation $\mathbf{x}_k := [x_{k,1}, \ldots, x_{k,d}]^T$ is a vector of d dimensions where each entry can only take on one of a finite number of values in the alphabet $\mathcal{X}$.

Given $\mathbf{x}^n$, the ML-estimator of the unknown distribution P is defined as

$$P_{\mathrm{ML}} := \operatorname*{argmax}_{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)} \sum_{k=1}^{n} \log Q(\mathbf{x}_k), \tag{3.2}$$

where $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d) \subset \mathcal{P}(\mathcal{X}^d)$ is defined as the set of all tree distributions on the alphabet $\mathcal{X}^d$ over d nodes.

In 1968, Chow and Liu showed that the above ML-estimate P_{ML} can be found efficiently via a MWST algorithm [42], and is described in Section 2.5.2. We denote the tree graph of the ML-estimate P_{ML} by $\widehat{T}_{\mathrm{ML}} = (V, E_{\mathrm{ML}})$ with vertex set V and edge set E_{ML}.

Given a tree distribution P, define the probability of the error event that the set of edges is *not* estimated correctly by the ML-estimator as

$$\mathcal{A}_n := \{E_{\mathrm{ML}} \neq E_P\} \tag{3.3}$$

We denote $\mathbb{P} := P^n$ as the n-fold *product probability measure* of the n samples $\mathbf{x}^n$ which are drawn i.i.d. from P. In this chapter, we are interested in studying the *rate* or

error exponent[2] K_P at which the above error probability exponentially decays with the number of samples n, given by,

$$K_P := \lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}(\mathcal{A}_n), \tag{3.4}$$

whenever the limit exists. Indeed, we will prove that the limit in (3.4) exists in the sequel. With the $\doteq$ notation, (3.4) can be written as

$$\mathbb{P}(\mathcal{A}_n) \doteq \exp(-nK_P). \tag{3.5}$$

A positive error exponent ($K_P > 0$) implies an exponential decay of error probability in ML structure learning, and we will establish necessary and sufficient conditions to ensure this.

Note that we are only interested in quantifying the probability of the error in learning the *structure* of P in (3.3). We are not concerned about the parameters that define the ML tree distribution P_{ML}. Since there are only finitely many (but a super-exponential number of) structures, this is in fact akin to an ML problem where the parameter space is discrete and finite [168]. Thus, under some mild technical conditions, we can expect exponential decay in the probability of error as mentioned in [168]. Otherwise, we can only expect convergence with rate $\mathcal{O}_p(1/\sqrt{n})$ for estimation of parameters that belong to a continuous parameter space [177]. In this work, we quantify the error exponent for learning tree structures using the ML learning procedure precisely.

■ 3.3 LDP for Empirical Mutual Information

The goal of this paper is to characterize the error exponent for ML tree learning K_P in (3.4). As a first step, we consider a simpler event, which may potentially lead to an error in ML-estimation. In this section, we derive the LDP rate for this event, and in the next section, we use the result to derive K_P, the exponent associated to the error event $\mathcal{A}_n$ defined in (3.3).

Since the ML-estimate uses the empirical mutual information quantities as the edge weights for the MWST algorithm, the relative values of the empirical mutual information quantities have an impact on the accuracy of ML-estimation. In other words, if the order of these empirical quantities is different from the true order then it can potentially lead to an error in the estimated edge set. Hence, it is crucial to study the probability of the event that the empirical mutual information quantities of any two node pairs is different from the true order.

Formally, let us consider two distinct node pairs with no common nodes $e, e' \in \binom{\mathcal{V}}{2}$ with unknown distribution $P_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$, where the notation $P_{e,e'}$ denotes the marginal of the tree-structured graphical model P on the nodes in the set $\{e, e'\}$. Similarly, P_e

[2] In the maximum-likelihood estimation literature (e.g. [11, 115]) if the limit in (3.4) exists, K_P is also typically known as the inaccuracy rate. We will be using the terms rate, error exponent and inaccuracy rate interchangeably in the sequel. All these terms refer to K_P.

is the marginal of P on edge e. Assume that the order of the true mutual information quantities follow $I(P_e) > I(P_{e'})$. A *crossover event*[3] occurs if the corresponding empirical mutual information quantities are of the reverse order, given by

$$\mathcal{C}_{e,e'} := \left\{ I(\widehat{P}_e) \leq I(\widehat{P}_{e'}) \right\}. \tag{3.6}$$

As the number of samples $n \to \infty$, the empirical quantities approach the true ones, and hence, the probability of the above event decays to zero. When the decay is exponential, we have a LDP for the above event, and we term its rate as the *crossover rate for empirical mutual information* quantities, defined as

$$J_{e,e'} := \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}\left(\mathcal{C}_{e,e'}\right), \tag{3.7}$$

assuming the limit in (3.7) exists. Indeed, we show in the proof of Theorem 3.1 that the limit exists. Intuitively (and as seen in our numerical simulations in Section 3.7), if the difference between the true mutual information quantities $I(P_e) - I(P_{e'})$ is large (*i.e.*, $I(P_e) \gg I(P_{e'})$), we expect the probability of the crossover event $\mathcal{C}_{e,e'}$ to be small. Thus, the rate of decay would be faster and hence, we expect the crossover rate $J_{e,e'}$ to be large. In the following, we see that $J_{e,e'}$ depends not only on the difference of mutual information quantities $I(P_e) - I(P_{e'})$, but also on the *distribution* $P_{e,e'}$ of the variables on node pairs e and e', since the distribution $P_{e,e'}$ influences the accuracy of estimating them.

Theorem 3.1 (Crossover Rate for Empirical MIs)**.** *The crossover rate for a pair of empirical mutual information quantities in* (3.7) *is given by*

$$J_{e,e'} \;=\; \inf_{Q\in\mathcal{P}(\mathcal{X}^4)} \left\{ D(Q \,||\, P_{e,e'}) : I(Q_{e'}) = I(Q_e) \right\}, \tag{3.8}$$

where $Q_e, Q_{e'} \in \mathcal{P}(\mathcal{X}^2)$ are marginals of Q over node pairs e and e', which do not share common nodes, i.e.,

$$Q_e(x_e) \;:=\; \sum_{x_{e'}\in\mathcal{X}^2} Q(x_e, x_{e'}), \tag{3.9a}$$

$$Q_{e'}(x_{e'}) \;:=\; \sum_{x_e\in\mathcal{X}^2} Q(x_e, x_{e'}). \tag{3.9b}$$

The infimum in (3.8) *is attained by some distribution $Q^*_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$ satisfying $I(Q^*_{e'}) = I(Q^*_e)$ and $J_{e,e'} > 0$.*

Proof. (*Sketch*) The proof hinges on Sanov's theorem [47, Ch. 11] and the contraction principle in large-deviations [62, Sec. III.5]. The existence of the minimizer follows from

[3]The event $\mathcal{C}_{e,e'}$ in (3.6) depends on the number of samples n but we suppress this dependence for convenience.

the compactness of the constraint set and Weierstrass' extreme value theorem [166, Theorem 4.16]. The rate $J_{e,e'}$ is strictly positive since we assumed, *a-priori*, that the two node pairs e and e' satisfy $I(P_e) > I(P_{e'})$. As a result, $Q^*_{e,e'} \neq P_{e,e'}$ and $D(Q^*_{e,e'} \,||\, P_{e,e'}) > 0$. See Appendix 3.A for the details. □

In the above theorem, which is analogous to Theorem 3.3 in [36], we derived the crossover rate $J_{e,e'}$ as a constrained minimization over a submanifold of distributions in $\mathcal{P}(\mathcal{X}^4)$ (See Fig. 3.5), and also proved the existence of an optimizing distribution Q^*. However, it is not easy to further simplify the rate expression in (3.8) since the optimization is non-convex.

Importantly, this means that it is not clear how the parameters of the distribution $P_{e,e'}$ affect the rate $J_{e,e'}$, hence (3.8) is not intuitive to aid in understanding the relative ease or difficulty in estimating particular tree-structured distributions. In Section 3.5, we assume that P satisfies some (so-called very noisy learning) conditions and use Euclidean information theory [25, 26] to approximate the rate in (3.8) in order to gain insights as to how the distribution parameters affect the crossover rate $J_{e,e'}$ and ultimately, the error exponent K_P for learning the tree structure.

Theorem 3.1 specifies the crossover rate $J_{e,e'}$ when the two node pairs e and e' do not have any common nodes. If e and e' share one node, then the distribution $P_{e,e'} \in \mathcal{P}(\mathcal{X}^3)$ and here, the crossover rate for empirical mutual information is

$$J_{e,e'} \;=\; \inf_{Q \in \mathcal{P}(\mathcal{X}^3)} \left\{ D(Q \,||\, P_{e,e'}) : I(Q_{e'}) = I(Q_e) \right\}. \tag{3.10}$$

In Section 3.5, we obtain an approximate closed-form expression for $J_{e,e'}$. The expression, provided in Theorem 3.7, does not depend on whether e and e' share a node.

Example: Symmetric Star Graph

It is now instructive to study a simple example to see how the overall error exponent K_P for structure learning in (3.4) depends on the set of crossover rates $\{J_{e,e'} : e, e' \in \binom{\mathcal{V}}{2}\}$. We consider a graphical model P with an associated tree $T_P = (\mathcal{V}, \mathcal{E}_P)$ which is a d-order star with a central node 1 and outer nodes $2, \ldots, d$, as shown in Fig. 3.1. The edge set is given by $\mathcal{E}_P = \{(1, i) : i = 2, \ldots, d\}$.

We assign the joint distributions $Q_a, Q_b \in \mathcal{P}(\mathcal{X}^2)$ and $Q_{a,b} \in \mathcal{P}(\mathcal{X}^4)$ to the variables in this graph in the following specific way:

1. $P_{1,i} \equiv Q_a$ for all $2 \leq i \leq d$.
2. $P_{i,j} \equiv Q_b$ for all $2 \leq i, j \leq d$, $i \neq j$.
3. $P_{1,i,j,k} \equiv Q_{a,b}$ for all $2 \leq i, j, k \leq d$, $i \neq j \neq k$.

Thus, we have identical pairwise distributions $P_{1,i} \equiv Q_a$ of the central node 1 and any other node i, and also identical pairwise distributions $P_{i,j} \equiv Q_b$ of any two distinct

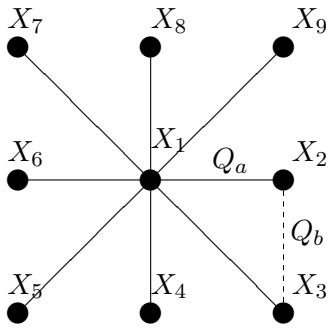

Figure 3.1. The star graph with $d = 9$. Q_a is the joint distribution on any pair of variables that form an edge e.g., x_1 and x_2. Q_b is the joint distribution on any pair of variables that do not form an edge e.g., x_2 and x_3. By symmetry, all crossover rates are equal.

outer nodes i and j. Furthermore, assume that $I(Q_a) > I(Q_b) > 0$. Note that the distribution $Q_{a,b} \in \mathcal{P}(\mathcal{X}^4)$ completely specifies the above graphical model with a star graph. Also, from the above specifications, we see that Q_a and Q_b are the marginal distributions of $Q_{a,b}$ with respect to to node pairs $(1, i)$ and (j, k) respectively *i.e.*,

$$Q_a(x_1, x_i) = \sum_{(x_j,x_k)\in\mathcal{X}^2} P_{1,i,j,k}(x_1, x_i, x_j, x_k), \tag{3.11a}$$

$$Q_b(x_j, x_k) = \sum_{(x_1,x_i)\in\mathcal{X}^2} P_{1,i,j,k}(x_1, x_i, x_j, x_k). \tag{3.11b}$$

Note that each crossover event between any non-edge e' (necessarily of length 2) and an edge e along its path results in an error in the learned structure since it leads to e' being declared an edge instead of e. Due to the symmetry, all such crossover rates between pairs e and e' are equal. By the "worst-exponent-wins" rule [62, Ch. 1], it is more likely to have a single crossover event than multiple ones. Hence, the error exponent is equal to the crossover rate between an edge and a non-neighbor pair in the symmetric star graph. We state this formally in the following proposition.

Proposition 3.2 (Error Exponent for symmetric star graph)**.** *For the symmetric graphical model with star graph and $Q_{a,b}$ as described above, the error exponent for structure learning K_P in* (3.4)*, is equal to the crossover rate between an edge and a non-neighbor node pair*

$$K_P \;=\; J_{e,e'}, \quad \text{for any} \quad e \in \mathcal{E}_P,\, e' \notin \mathcal{E}_P, \tag{3.12}$$

where from (3.8)*, the crossover rate is given by*

$$J_{e,e'} = \inf_{R_{1,2,3,4}\in\mathcal{P}(\mathcal{X}^4)} \{D(R_{1,2,3,4}||Q_{a,b}) : I(R_{1,2}) = I(R_{3,4})\}, \tag{3.13}$$

with $R_{1,2}$ and $R_{3,4}$ as the marginals of $R_{1,2,3,4}$, e.g.,

$$R_{1,2}(x_1, x_2) = \sum_{(x_3,x_4)\in\mathcal{X}^2} R_{1,2,3,4}(x_1, x_2, x_3, x_4). \tag{3.14}$$

Proof. Since there are only two distinct distributions Q_a (which corresponds to a true edge) and Q_b (which corresponds to a non-edge), there is only *one* unique rate $J_{e,e'}$, namely the expression in (3.8) with $P_{e,e'}$ replaced by $Q_{a,b}$. If the event $\mathcal{C}_{e,e'}$, in (3.6), occurs, an error definitely occurs. This corresponds to the case where *any one* edge $e \in \mathcal{E}_P$ is replaced by *any other* node pair e' not in $\mathcal{E}_P$.[4] □

Hence, we have derived the error exponent for learning a symmetric star graph through the crossover rate $J_{e,e'}$ between any node pair e which is an edge in the star graph and another node pair e' which is not an edge.

The symmetric star graph possesses symmetry in the distributions and hence it is easy to relate K_P to a sole crossover rate. In general, it is not straightforward to derive the error exponent K_P from the set of crossover rates $\{J_{e,e'}\}$ since they may not all be equal and more importantly, crossover events for different node pairs affect the learned structure E_{ML} in a complex manner. In the next section, we provide an exact expression for K_P by identifying the (sole) crossover event related to a dominant error tree. Finally, we remark that the crossover event $\mathcal{C}_{e,e'}$ is related to the notion of neighborhood selection in the graphical model learning literature [136, 211].

■ 3.4 Error Exponent for Structure Learning

The analysis in the previous section characterized the rate $J_{e,e'}$ for the crossover event $\mathcal{C}_{e,e'}$ between two empirical mutual information pairs. In this section, we connect these set of rate functions $\{J_{e,e'}\}$ to the quantity of interest, viz., the error exponent for ML-estimation of edge set K_P in (3.4).

Recall that the event $\mathcal{C}_{e,e'}$ denotes an error in estimating the order of mutual information quantities. However, such events $\mathcal{C}_{e,e'}$ need not necessarily lead to the error event $\mathcal{A}_n$ in (3.3) that the ML-estimate of the edge set E_{ML} is different from the true set E_P. This is because the ML-estimate E_{ML} is a tree and this global constraint implies that certain crossover events can be ignored. In the sequel, we will identify useful crossover events through the notion of a *dominant error tree*.

■ 3.4.1 Dominant Error Tree

We can decompose the error event for structure estimation $\mathcal{A}_n$ in (3.3) into a set of mutually-exclusive events

$$\mathbb{P}(\mathcal{A}_n) = \mathbb{P}\left(\bigcup_{T \in \mathcal{T}^d \setminus \{T_P\}} \mathcal{U}_n(T) \right) = \sum_{T \in \mathcal{T}^d \setminus \{T_P\}} \mathbb{P}\left(\mathcal{U}_n(T)\right), \tag{3.15}$$

[4] Also see theorem 3.4 and its proof for the argument that the dominant error tree differs from the true tree by a single edge.

where each $\mathcal{U}_n(T)$ denotes the event that the graph of the ML-estimate $\widehat{T}_{\mathrm{ML}}$ is a tree T different from the true tree T_P. In other words,

$$\mathcal{U}_n(T) \;:=\; \begin{cases} \left\{\widehat{T}_{\mathrm{ML}} = T\right\}, & \text{if } \; T \in \mathcal{T}^d \setminus \{T_P\}, \\ \emptyset, & \text{if } \; T = T_P. \end{cases} \tag{3.16}$$

Note that $\mathcal{U}_n(T) \cap \mathcal{U}_n(T') = \emptyset$ whenever $T \neq T'$. The large-deviation rate or the exponent for each error event $\mathcal{U}_n(T)$ is

$$\Upsilon(T) := \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}\left(\mathcal{U}_n(T)\right), \tag{3.17}$$

whenever the limit exists. Among all the error events $\mathcal{U}_n(T)$, we identify the dominant one with the slowest rate of decay.

Definition 3.1 (Dominant Error Tree)**.** *A* dominant error tree $T_P^* = (V, E_P^*)$ *is a spanning tree given by*[5]

$$T_P^* := \operatorname*{argmin}_{T \in \mathcal{T}^d \setminus \{T_P\}} \Upsilon(T). \tag{3.18}$$

Roughly speaking, a dominant error tree is the tree that is the most-likely asymptotic output of the ML-estimator in the event of an error. Hence, it belongs to the set $\mathcal{T}^d \setminus \{T_P\}$. In the following, we note that the error exponent in (3.4) is equal to the exponent of the dominant error tree.

Proposition 3.3 (Dominant Error Tree & Error Exponent)**.** *The error exponent K_P for structure learning is equal to the exponent $\Upsilon(T_P^*)$ of the dominant error tree T_P^*.*

$$K_P = \Upsilon(T_P^*). \tag{3.19}$$

Proof. From (3.17), we can write

$$\mathbb{P}\left(\mathcal{U}_n(T)\right) \doteq \exp(-n\Upsilon(T)), \quad \forall\, T \in \mathcal{T}^d \setminus \{T_P\}. \tag{3.20}$$

Now from (3.15), we have

$$\mathbb{P}(\mathcal{A}_n) \doteq \sum_{T \in \mathcal{T}^d \setminus \{T_P\}} \exp\left(-n\Upsilon(T)\right) \doteq \exp\left(-n\Upsilon(T_P^*)\right), \tag{3.21}$$

from the "worst-exponent-wins" principle [62, Ch. 1] and the definition of the dominant error tree T_P^* in (3.18). □

[5] We will use the notation argmin extensively in the sequel. It is to be understood that if there is no unique minimum (*e.g.* in (3.18)), then we arbitrarily choose one of the minimizing solutions.

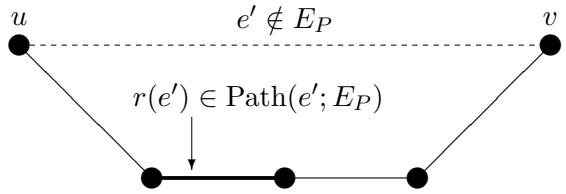

Figure 3.2. The path associated to the non-edge $e' = (u, v) \notin E_P$, denoted $\mathrm{Path}(e'; E_P) \subset E_P$, is the set of edges along the unique path linking the end points of $e' = (u, v)$. The edge $r(e') = \mathrm{argmin}_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'}$ is the dominant replacement edge associated to $e' \notin E_P$.

Thus, by identifying a dominant error tree T_P^*, we can find the error exponent $K_P = \Upsilon(T_P^*)$. To this end, we revisit the crossover events $\mathcal{C}_{e,e'}$ in (3.6), studied in the previous section. Consider a non-neighbor node pair e' with respect to E_P and the unique path of edges in E_P connecting the two nodes, which we denote as $\mathrm{Path}(e'; E_P)$. See Fig. 3.2, where we define the notion of the path given a non-edge e'. Note that e' and $\mathrm{Path}(e'; E_P)$ necessarily form a cycle; if we replace any edge $e \in E_P$ along the path of the non-neighbor node pair e', the resulting edge set $E_P \setminus \{e\} \cup \{e'\}$ is still a spanning tree. Hence, all such replacements are feasible outputs of the ML-estimation in the event of an error. As a result, all such crossover events $\mathcal{C}_{e,e'}$ need to be considered for the error event for structure learning $\mathcal{A}_n$ in (3.3). However, for the error exponent K_P, again by the "worst-exponent-wins" principle, we only need to consider the crossover event between each non-neighbor node pair e' and its dominant replacement edge $r(e') \in E_P$ defined below.

Definition 3.2 (Dominant Replacement Edge)**.** *For each non-neighbor node pair $e' \notin E_P$, its* dominant replacement edge $r(e') \in E_P$ *is defined as the edge in the unique path along E_P connecting the nodes in e' having the minimum crossover rate*

$$r(e') := \operatorname*{argmin}_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'}, \tag{3.22}$$

where the crossover rate $J_{e,e'}$ is given by (3.8)*.*

We are now ready to characterize the error exponent K_P in terms of the crossover rate between non-neighbor node pairs and their dominant replacement edges.

Theorem 3.4 (Error exponent as a single crossover event)**.** *The error exponent for ML-tree estimation in* (3.4) *is given by*

$$K_P \;=\; J_{r(e^*),e^*} \;=\; \min_{e' \notin E_P} \min_{e \in \mathrm{Path}(e'; E_P)} J_{e,e'}, \tag{3.23}$$

where $r(e^*)$ is the dominant replacement edge, defined in (3.22)*, associated to $e^* \notin E_P$ and e^* is the optimizing non-neighbor node pair*

$$e^* := \operatorname*{argmin}_{e' \notin E_P} J_{r(e'),e'}. \tag{3.24}$$

The dominant error tree $T_P^* = (V, E_P^*)$ *in* (3.18) *has edge set*

$$E_P^* = E_P \cup \{e^*\} \setminus \{r(e^*)\}. \tag{3.25}$$

In fact, we also have the following (finite-sample) upper bound on the error probability:

$$\mathbb{P}(\mathcal{A}_n) \leq \frac{(d-1)^2(d-2)}{2} \binom{n+1+|\mathcal{X}|^4}{n+1} \exp(-nK_P), \tag{3.26}$$

for all $n \in \mathbb{N}$.

Proof. (Sketch) The edge set of the dominant error tree E_P^* differs from E_P in exactly one edge (See Appendix 3.B). This is because if E_P^* were to differ from E_P in strictly more than one edge, the resulting error exponent would not be the minimum, hence contradicting Proposition 3.3. To identify the dominant error tree, we use the union bound as in (3.15) and the "worst-exponent-wins" principle [62, Ch. 1], to conclude that the rate that dominates is the minimum $J_{r(e'),e'}$ over all possible non-neighbor node pairs $e' \notin E_P$. See Appendix 3.B for the details. □

The above theorem relates the set of crossover rates $\{J_{e,e'}\}$, which we characterized in the previous section, to the overall error exponent K_P, defined in (3.4). Note that the result in (3.23) and also the existence of the limit in (3.4) means that the error probability is *tight to first order in the exponent* in the sense that $\mathbb{P}(\mathcal{A}_n) \doteq \exp(-nK_P)$. This is in contrast to the work in [223], where bounds on the upper and lower limit on the sequence $-\frac{1}{n}\log \mathbb{P}(\mathcal{A}_n)$ were established. We numerically compute the error exponent K_P for different discrete distributions in Section 3.7.

From (3.23), we see that if at least one of the crossover rates $J_{e,e'}$ in the minimization is zero, the overall error exponent K_P is zero. This observation is important for the derivation of necessary and sufficient conditions for K_P to be positive, and hence, for the error probability to decay exponentially in the number of samples n.

■ 3.4.2 Conditions for Exponential Decay

We now provide necessary and sufficient conditions that ensure that K_P is strictly positive. This is obviously of crucial importance since if $K_P > 0$, this implies exponential decay of the desired probability of error $\mathbb{P}(\mathcal{A}_n)$, where the error event $\mathcal{A}_n$ is defined in (3.3). For the purpose of stating this result, we assume that T_P, the original structure is just acyclic, *i.e.*, it may not be connected.

Theorem 3.5 (Equivalent Conditions for Exponential Decay)**.** *The following three statements are equivalent.*

(a) The probability of error $\mathbb{P}(\mathcal{A}_n)$ *decays exponentially i.e.,*

$$K_P > 0. \tag{3.27}$$

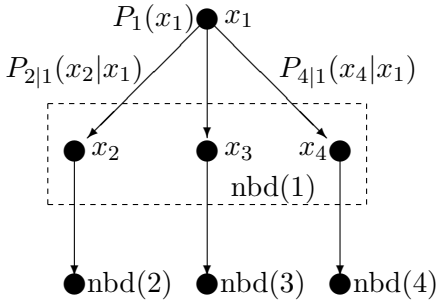

Figure 3.3. Illustration for Example 3.1.

(b) The mutual information quantities satisfy:

$$I(P_{e'}) < I(P_e), \quad \forall\, e \in \mathrm{Path}(e'; E_P),\, e' \notin E_P. \tag{3.28}$$

(c) T_P is not a proper forest.[6]

Proof. (*Sketch*) We first show that (a) $\Leftrightarrow$ (b).
($\Rightarrow$) We assume statement (a) is true *i.e.*, $K_P > 0$ and prove that statement (b) is true. Suppose, to the contrary, that $I(P_{e'}) = I(P_e)$ for some $e \in \mathrm{Path}(e'; E_P)$ and some $e' \notin E_P$. Then $J_{r(e'),e'} = 0$, where $r(e')$ is the replacement edge associated to e'. By (3.23), $K_P = 0$, which is a contradiction.
($\Leftarrow$) We now prove that statement (a) is true assuming statement (b) is true *i.e.*, $I(P_{e'}) < I(P_e)$ for all $e \in \mathrm{Path}(e'; E_P)$ and $e' \notin E_P$. By Theorem 3.1, the crossover rate $J_{r(e'),e'}$ in (3.8) is positive for all $e' \notin E_P$. From (3.23), $K_P > 0$ since there are only finitely many e', hence the minimum in (3.24) is attained at some non-zero value, *i.e.*, $K_P = \min_{e' \notin E_P} J_{r(e'),e'} > 0$.

Statement (c) is equivalent to statement (b). The proof of this claim makes use of the positivity condition that $P(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^d$ and the fact that if variables x_1, x_2 and x_3 form Markov chains $x_1 - x_2 - x_3$ and $x_1 - x_3 - x_2$, then x_1 is necessarily *jointly independent* of (x_2, x_3). Since this proof is rather lengthy, we refer the reader to Appendix 3.C for the details. □

Condition (b) states that, for each non-edge e', we need $I(P_{e'})$ to be strictly smaller than the mutual information of its dominant replacement edge $I(P_{r(e')})$. Condition (c) is a more intuitive condition for exponential decay of the probability of error $\mathbb{P}(\mathcal{A}_n)$. This is an important result since it says that for *any* non-degenerate tree distribution in which all the pairwise joint distributions are not product distributions (*i.e.*, not a proper forest), then we have exponential decay in the error probability.

In the following example, we describe a simple random process for constructing a distribution P such that all three conditions in Theorem 3.5 are satisfied with probability one (w.p. 1). See Fig. 3.3.

[6] A proper forest on d nodes is an undirected, acyclic graph that has (strictly) fewer than $d-1$ edges.

Example 3.1. *Suppose the structure of P, a spanning tree distribution with graph $T_P = (V, E_P)$, is fixed and $\mathcal{X} = \{0, 1\}$. Now, we assign the parameters of P using the following procedure. Let x_1 be the root node. Then randomly draw the parameter of the Bernoulli distribution $P_1(x_1)$ from a uniform distribution on $[0, 1]$ i.e., $P_1(x_1 = 0) = \theta_{x_1^0}$ and $\theta_{x_1^0} \sim \mathcal{U}[0, 1]$. Next let* $\mathrm{nbd}(1)$ *be the set of neighbors of x_1. Regard the set of variables $\{x_j : j \in \mathrm{nbd}(1)\}$ as the children*[7] *of x_1. For each $j \in \mathrm{nbd}(1)$, sample both $P(x_j = 0|x_1 = 0) = \theta_{x_j^0|x_1^0}$ as well as $P(x_j = 0|x_1 = 1) = \theta_{x_j^0|x_1^1}$ from independent uniform distributions on $[0, 1]$ i.e., $\theta_{x_j^0|x_1^0} \sim \mathcal{U}[0, 1]$ and $\theta_{x_j^0|x_1^1} \sim \mathcal{U}[0, 1]$. Repeat this procedure for all children of x_1. Then repeat the process for all other children. This construction results in a joint distribution $P(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^d$ w.p. 1. In this case, by continuity, all mutual informations are distinct w.p. 1, the graph is not a proper forest w.p. 1 and the rate $K_P > 0$ w.p. 1.*

This example demonstrates that $\mathbb{P}(\mathcal{A}_n)$ decays exponentially for *almost every* tree distribution. More precisely, the tree distributions in which $\mathbb{P}(\mathcal{A}_n)$ does not decay exponentially has measure zero in $\mathcal{P}(\mathcal{X}^d)$.

■ 3.4.3 Computational Complexity

Finally, we provide an upper bound on the computational complexity to compute K_P in (3.23). Our upper bound on the computational complexity depends on the *diameter* of the tree $T_P = (V, E_P)$ which is defined as

$$\mathrm{diam}(T_P) := \max_{u,v \in V} L(u, v), \tag{3.29}$$

where $L(u, v)$ is the length (number of hops) of the unique path between nodes u and v. For example, $L(u, v) = 4$ for the non-edge $e' = (u, v)$ in the subtree in Fig. 3.2.

Theorem 3.6 (Computational Complexity for K_P)**.** *The number of computations of $J_{e,e'}$ to compute K_P, denoted $N(T_P)$, satisfies*

$$N(T_P) \;\leq\; \frac{1}{2}\mathrm{diam}(T_P)(d-1)(d-2). \tag{3.30}$$

Proof. Given a non-neighbor node pair $e' \notin E_P$, we perform a maximum of $\mathrm{diam}(T_P)$ calculations to determine the dominant replacement edge $r(e')$ from (3.22). Combining this with the fact that there are a total of $|\binom{V}{2} \setminus E_P| = \binom{d}{2} - (d-1) = \frac{1}{2}(d-1)(d-2)$ node pairs not in E_P, we obtain the upper bound. □

Thus, if the diameter of the tree $\mathrm{diam}(T_P)$ is relatively low and independent of number of nodes d, the complexity is quadratic in d. For instance, for a star graph, the diameter $\mathrm{diam}(T_P) = 2$. For a balanced tree,[8] $\mathrm{diam}(T_P) = \mathcal{O}(\log d)$, hence the number of computations is $\mathcal{O}(d^2 \log d)$.

[7] Let x_1 be the root of the tree. In general, the children of a node x_k ($k \neq 1$) is the set of nodes connected to x_k that are further away from the root than x_k.

[8] A balanced tree is one where no leaf is much farther away from the root than any other leaf. The length of the longest direct path between any pair of nodes is $\mathcal{O}(\log d)$.

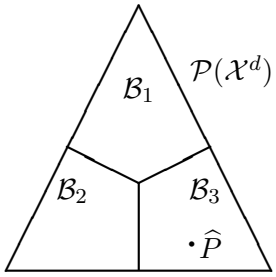

Figure 3.4. The partitions of the simplex associated to our learning problem are given by $\mathcal{B}_i$, defined in (3.31). In this example, the type $\widehat{P}$ belongs to $\mathcal{B}_3$ so the tree associated to partition $\mathcal{B}_3$ is favored.

■ 3.4.4 Relation of The Maximum-Likelihood Structure Learning Problem to Robust Hypothesis Testing

We now take a short detour and discuss the relation between the analysis of the learning problem and *robust hypothesis testing*, which was first considered by Huber and Strassen in [102]. Subsequent work was done in [151, 198, 220] albeit for differently defined uncertainty classes known as moment classes.

We hereby consider an alternative but related problem. Let $T_1, \ldots, T_M$ be the $M = d^{d-2}$ trees with d nodes. Also let $\mathcal{Q}_1, \ldots, \mathcal{Q}_M \subset \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$ be the subsets of tree-structured graphical models Markov on $T_1, \ldots, T_M$ respectively. The structure learning problem is similar to the M-ary hypothesis testing problem between the uncertainty classes of distributions $\mathcal{Q}_1, \ldots, \mathcal{Q}_M$. The uncertainty class $\mathcal{Q}_i$ denotes the set of tree-structured graphical models with different *parameters* (marginal $\{P_i : i \in V\}$ and pairwise distributions $\{P_{i,j} : (i,j) \in E_P\}$) but Markov on the same tree T_i.

In addition, we note that the probability simplex $\mathcal{P}(\mathcal{X}^d)$ can be partitioned into M subsets[9] $\mathcal{B}_1, \ldots, \mathcal{B}_M \subset \mathcal{P}(\mathcal{X}^d)$ where each $\mathcal{B}_i, i = 1, \ldots, M$ is defined as

$$\mathcal{B}_i := \bigcup_{P' \in \mathcal{Q}_i} \left\{ Q : D(P' \,||\, Q) \leq \min_{R \in \cup_{j \neq i} \mathcal{Q}_i} D(P' \,||\, R) \right\}. \tag{3.31}$$

See Fig. 3.4. According to the ML criterion in (2.108), if the type $\widehat{P}$ belongs to $\mathcal{B}_i$, then the i-th tree is favored.

In [190], the Neyman-Pearson setup of a robust binary hypothesis testing problem was considered. The null hypothesis corresponds to the true tree model P and the (composite) alternative hypothesis corresponds to the set of distributions Markov on some erroneous tree $T_Q \neq T_P$. The false-alarm probability was constrained to be smaller than $\alpha > 0$ and optimized for worst-case type-II (missed detection) error exponent using the Chernoff-Stein Lemma [47, Ch. 12]. It was established that the worst-case error exponent can be expressed in closed-form in terms of the mutual information of so-called *bottleneck edges*, *i.e.*, the edge and non-edge pair that have the smallest mutual

[9] From the definition in (3.31), we see that the relative interior of the subsets are pairwise disjoint. We discuss the scenario when P lies on the boundaries of these subsets in Section 3.6.

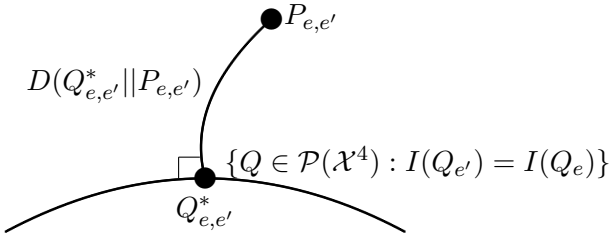

Figure 3.5. A geometric interpretation of (3.8) where $P_{e,e'}$ is projected onto the submanifold of probability distributions $\{Q \in \mathcal{P}(\mathcal{X}^4) : I(Q_{e'}) = I(Q_e)\}$.

information difference. However, in general, for the binary hypothesis testing problem, the error event *does not* decompose into a union of local events. This is in contrast to error exponent for learning the ML tree K_P, which can be computed by considering *local crossover events* $\mathcal{C}_{e,e'}$, defined in (3.6).

Note that $\{\widehat{P} \in \mathcal{B}_i\}$ corresponds to a *global event* since each $\mathcal{B}_i \subset \mathcal{P}(\mathcal{X}^d)$. The large-deviation analysis techniques we utilized to obtain the error exponent K_P in Theorem 3.4 show that such global error events can be also decomposed into a collection of local crossover events $\mathcal{C}_{e,e'}$. These local events depend only on the type *restricted* to pairs of nodes e and e' and are more intuitive for assessing (and analyzing) when and how an error can occur during the Chow-Liu learning process.

■ 3.5 Euclidean Approximations

In order to gain more insight into the error exponent, we make use of *Euclidean approximations* [26] of information-theoretic quantities to obtain an approximate but closed-form solution to (3.8), which is non-convex and hard to solve exactly. In addition, we note that the dominant error event results from an edge and a non-edge that satisfy the conditions for which the Euclidean approximation is valid, *i.e.*, the very-noisy condition given later in Definition 3.4. This justifies our approach we adopt in this section. Our use of Euclidean approximations for various information-theoretic quantities is akin to various problems considered in other contexts in information theory [1, 25, 26].

We first approximate the crossover rate $J_{e,e'}$ for any two node pairs e and e', which do not share a common node. The joint distribution on e and e', namely $P_{e,e'}$ belongs to the set $\mathcal{P}(\mathcal{X}^4)$. Intuitively, the crossover rate $J_{e,e'}$ should depend on the "separation" of the mutual information values $I(P_e)$ and $I(P_{e'})$, and also on the uncertainty of the difference between mutual information estimates $I(\widehat{P}_e)$ and $I(\widehat{P}_{e'})$. We will see that the approximate rate also depends on these mutual information quantities given by a simple expression which can be regarded as the signal-to-noise ratio (SNR) for learning.

Roughly speaking, our strategy is to "convexify" the objective and the constraints in (3.8). See Figs. 3.5 and 3.6. To do so, we recall that if P and Q are two discrete distributions with the same support $\mathcal{Y}$, and they are close entry-wise, the KL divergence

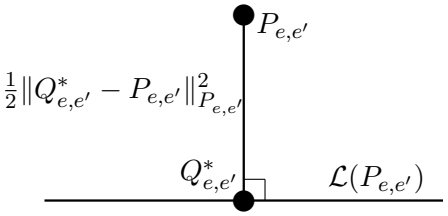

Figure 3.6. Convexifying the objective results in a least-squares problem. The objective is converted into a quadratic as in (3.39) and the linearized constraint set $\mathcal{L}(P_{e,e'})$ is given (3.40).

can be approximated [26] as

$$D(Q \,||\, P) = -\sum_{a\in\mathcal{Y}} Q(a) \log \frac{P(a)}{Q(a)}, \tag{3.32}$$

$$= -\sum_{a\in\mathcal{Y}} Q(a) \log \left[1 + \left(\frac{P(a) - Q(a)}{Q(a)}\right)\right], \tag{3.33}$$

$$= \frac{1}{2}\sum_{a\in\mathcal{Y}} \frac{(Q(a) - P(a))^2}{Q(a)} + o(\|Q - P\|^2_\infty), \tag{3.34}$$

$$= \frac{1}{2}\|Q - P\|^2_Q + o(\|Q - P\|^2_\infty), \tag{3.35}$$

where $\|y\|^2_w$ denotes the weighted squared norm of y, *i.e.*, $\|y\|^2_w := \sum_i y_i^2/w_i$. The equality in (3.34) holds because $\log(1+t) = \sum_{i=1}^{\infty}(-1)^{i+1}t^i/i$ for $t \in (-1, 1]$. The difference between the divergence and the Euclidean approximation becomes tight as $\epsilon = \|P - Q\|_\infty \to 0$. Moreover, it remains tight even if the subscript Q in (3.35) is changed to a distribution Q' in the vicinity of Q [26]. That is, the difference between $\|Q - P\|_Q$ and $\|Q - P\|_{Q'}$ is negligible compared to either term when $Q' \approx Q$. Using this fact and the assumption that P and Q are two discrete distributions that are close entry-wise,

$$D(Q \,||\, P) \;\approx\; \frac{1}{2}\|Q - P\|^2_P. \tag{3.36}$$

In fact, it is also known [26] that if $\|P - Q\|_\infty < \epsilon$ for some $\epsilon > 0$, we also have $D(P \,||\, Q) \approx D(Q \,||\, P)$.

In the following, to make our statements precise, we will use the notation $\alpha_1 \approx_\delta \alpha_2$ to denote that two real numbers α_1 and α_2 are in the δ neighborhood of each other, *i.e.*, $|\alpha_1 - \alpha_2| < \delta$.[10] We will also need the following notion of information density to state our approximation for $J_{e,e'}$.

Definition 3.3 (Information Density)**.** *Given a pairwise joint distribution $P_{i,j}$ on $\mathcal{X}^2$ with marginals P_i and P_j, the* information density *[126, 156] function, denoted by $s_{i,j}$:*

[10] In the following, we will also have continuity statements where given $\epsilon > 0$ and $\alpha_1 \approx_\epsilon \alpha_2$, implies that there exists some $\delta = \delta(\epsilon) > 0$ such that $\beta_1 \approx_\delta \beta_2$. We will be casual about specifying what the δ's are.

$\mathcal{X}^2 \to \mathbb{R}$, *is defined as*

$$s_{i,j}(x_i, x_j) := \log \frac{P_{i,j}(x_i, x_j)}{P_i(x_i)P_j(x_j)}, \quad \forall\, (x_i, x_j) \in \mathcal{X}^2. \tag{3.37}$$

Hence, for each node pair $e = (i, j)$, the information density s_e is also a random variable whose expectation is simply the mutual information between x_i and x_j, *i.e.*, $\mathbb{E}[s_e] = I(P_e)$.

Recall that we also assumed in Section 3.2 that T_P is a spanning tree, which implies that for all node pairs (i, j), $P_{i,j}$ is *not* a product distribution, *i.e.*, $P_{i,j} \neq P_i P_j$, because if it were, then T_P would be disconnected. We now define a condition for which our approximation holds.

Definition 3.4 (ϵ-Very Noisy Condition)**.** *We say that $P_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$, the joint distribution on node pairs e and e', satisfies the* ϵ-very noisy condition *if*

$$\|P_e - P_{e'}\|_\infty := \max_{(x_i,x_j)\in\mathcal{X}^2} |P_e(x_i, x_j) - P_{e'}(x_i, x_j)| < \epsilon. \tag{3.38}$$

This condition is needed because if (3.38) holds, then by continuity of the mutual information, there exists a $\delta > 0$ such that $I(P_e) \approx_\delta I(P_{e'})$, which means that the mutual information quantities are difficult to distinguish and the approximation in (3.35) is accurate.[11] Note that proximity of the mutual information values is not sufficient for the approximation to hold since we have seen from Theorem 3.1 that $J_{e,e'}$ depends not only on the mutual information quantities but on the entire joint distribution $P_{e,e'}$.

We now define the *approximate crossover rate* on disjoint node pairs e and e' as

$$\widetilde{J}_{e,e'} := \inf \left\{ \frac{1}{2} \|Q - P_{e,e'}\|^2_{P_{e,e'}} : Q \in \mathcal{L}(P_{e,e'}) \right\}, \tag{3.39}$$

where the (linearized) constraint set is

$$\begin{aligned} \mathcal{L}(P_{e,e'}) := \Big\{ & Q \in \mathcal{P}(\mathcal{X}^4) : I(P_e) + \left\langle \nabla_{P_e} I(P_e), Q - P_{e,e'} \right\rangle \\ & = I(P_{e'}) + \left\langle \nabla_{P_{e'}} I(P_{e'}), Q - P_{e,e'} \right\rangle \Big\}, \end{aligned} \tag{3.40}$$

where $\nabla_{P_e} I(P_e)$ is the gradient vector of the mutual information with respect to the joint distribution P_e. We also define the approximate error exponent as

$$\widetilde{K}_P := \min_{e' \notin E_P} \min_{e \in \mathrm{Path}(e'; E_P)} \widetilde{J}_{e,e'}. \tag{3.41}$$

We now provide the expression for the approximate crossover rate $\widetilde{J}_{e,e'}$ and also state the conditions under which the approximation is asymptotically accurate in ϵ.[12]

[11] Here and in the following, we do not specify the exact value of δ but we simply note that as $\epsilon \to 0$, the approximation in (3.36) becomes tighter.

[12] We say that a collection of approximations $\{\widetilde{\theta}(\epsilon) : \epsilon > 0\}$ of a true parameter θ is *asymptotically accurate in* ϵ (or simply asymptotically accurate) if the approximations converge to θ as $\epsilon \to 0$, *i.e.*, $\lim_{\epsilon\to 0} \widetilde{\theta}(\epsilon) = \theta$.

Theorem 3.7 (Euclidean approximation of $J_{e,e'}$)**.** *The approximate crossover rate for the empirical mutual information quantities, defined in* (3.39)*, is given by*

$$\widetilde{J}_{e,e'} \;=\; \frac{(\mathbb{E}[s_{e'} - s_e])^2}{2\,\mathrm{Var}(s_{e'} - s_e)} \;=\; \frac{(I(P_{e'}) - I(P_e))^2}{2\,\mathrm{Var}(s_{e'} - s_e)}, \tag{3.42}$$

where s_e is the information density defined in (3.37) *and the expectation and variance are both with respect to $P_{e,e'}$. Furthermore, the approximation* (3.42) *is asymptotically accurate, i.e., as $\epsilon \to 0$ (in the definition of ϵ-very noisy condition), we have that $\widetilde{J}_{e,e'} \to J_{e,e'}$.*

Proof. (*Sketch*) Eqs. (3.39) and (3.40) together define a least squares problem. Upon simiplification of the solution, we obtain (3.42). See Appendix 3.D for the details. □

We also have an additional result for the Euclidean approximation for the overall error exponent K_P. The proof is clear from the definition of $\widetilde{K}_P$ in (3.41) and the continuity of the min function.

Corollary 3.8 (Euclidean approximation of K_P)**.** *The approximate error exponent $\widetilde{K}_P$ is asymptotically accurate if all joint distributions in the set $\{P_{e,e'} : e \in \mathrm{Path}(e; E_P), e' \notin E_P\}$ satisfy the ϵ-very noisy condition.*

Hence, the expressions for the crossover rate $J_{e,e'}$ and the error exponent K_P are vastly simplified under the ϵ-very noisy condition on the joint distributions $P_{e,e'}$. The approximate crossover rate $\widetilde{J}_{e,e'}$ in (3.42) has a very intuitive meaning. It is proportional to the square of the difference between the mutual information quantities of P_e and $P_{e'}$. This corresponds exactly to our initial intuition – that if $I(P_e)$ and $I(P_{e'})$ are well separated ($I(P_e) \gg I(P_{e'})$) then the crossover rate has to be large. $\widetilde{J}_{e,e'}$ is also weighted by the precision (inverse variance) of $(s_{e'} - s_e)$. If this variance is large then we are uncertain about the estimate $I(\widehat{P}_e) - I(\widehat{P}_{e'})$, and crossovers are more likely, thereby reducing the crossover rate $\widetilde{J}_{e,e'}$.

We now comment on our assumption of $P_{e,e'}$ satisfying the ϵ-very noisy condition, under which the approximation is tight as seen in Theorem 3.7. When $P_{e,e'}$ is ϵ-very noisy, then we have $I(P_e) \approx_\delta I(P_{e'})$, which implies that the optimal solution of (3.8) $Q^*_{e,e'} \approx_{\delta'} P_{e,e'}$. When e is an edge and e' is a non-neighbor node pair, this implies that it is very hard to distinguish the relative magnitudes of the empiricals $I(\widehat{P}_e)$ and $I(\widehat{P}_{e'})$. Hence, the particular problem of learning the distribution $P_{e,e'}$ from samples is *very noisy*. Under these conditions, the approximation in (3.42) is accurate.

In summary, our approximation in (3.42) takes into account not only the absolute difference between the mutual information quantities $I(P_e)$ and $I(P_{e'})$, but also the uncertainty in learning them. The expression in (3.42) is, in fact, the SNR for the estimation of the difference between empirical mutual information quantities. This answers one of the fundamental questions we posed in the introduction, viz., that we are now able to distinguish between distributions that are "easy" to learn and those that are "difficult" by computing the set of SNR quantities $\{\widetilde{J}_{e,e'}\}$ in (3.42).

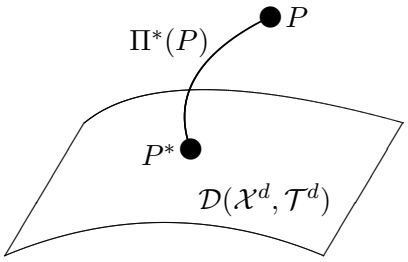

Figure 3.7. Reverse I-projection [51] of P onto the set of tree distributions $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$ given by (3.43).

■ 3.6 Extensions to Non-Tree Distributions

In all the preceding sections, we dealt exclusively with the case where the true distribution P is Markov on a tree. In this section, we extend the preceding large-deviation analysis to deal with distributions P that may not be tree-structured but in which we estimate a tree distribution from the given set of samples $\mathbf{x}^n$, using the Chow-Liu ML-estimation procedure. Since the Chow-Liu procedure outputs a tree, it is not possible to learn the structure of P correctly. Hence, it will be necessary to redefine the error event.

When P is not a tree distribution, we analyze the properties of the optimal *reverse I-projection* [51] of P onto the set of tree distributions, given by the optimization problem[13]

$$\Pi^*(P) := \min_{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)} D(P \,||\, Q). \tag{3.43}$$

$\Pi^*(P)$ is the KL-divergence of P to the closest element in $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$. See Fig. 3.7. As Chow and Wagner [43] noted, if P is not a tree, there may be several trees optimizing (3.43).[14] We denote the set of optimal projections as $\mathcal{P}^*(P)$, given by

$$\mathcal{P}^*(P) := \{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d) : D(P \,||\, Q) = \Pi^*(P)\}. \tag{3.44}$$

We now illustrate that $\mathcal{P}^*(P)$ may have more than one element with the following example.

Example 3.2. *Consider the parameterized discrete probability distribution $P \in \mathcal{P}(\{0,1\}^3)$ shown in Table 3.1 where $\xi \in (0, 1/3)$ and $\kappa \in (0, 1/2)$ are constants.*

Proposition 3.9 (Non-uniqueness of projection)**.** *For sufficiently small κ, the Chow-Liu MWST algorithm (using either Kruskal's [120] or Prim's [158] procedure) will first include the edge $(1, 2)$. Then, it will arbitrarily choose between the two remaining edges $(2, 3)$ or $(1, 3)$.*

[13] The minimum in the optimization problem in (3.43) is attained because the KL-divergence is continuous and the set of tree distributions $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$ is compact.

[14] This is a technical condition of theoretical interest in this section. In fact, it can be shown that the set of distributions such that there is more than one tree optimizing (3.43) has (Lebesgue) measure zero in $\mathcal{P}(\mathcal{X}^d)$.

x_1	x_2	x_3	Distribution $P(\mathbf{x})$
0	0	0	$(1/2-\xi)(1/2-\kappa)$
0	0	1	$(1/2+\xi)(1/2-\kappa)$
0	1	0	$(1/3+\xi)\kappa$
0	1	1	$(2/3-\xi)\kappa$
1	0	0	$(2/3-\xi)\kappa$
1	0	1	$(1/3+\xi)\kappa$
1	1	0	$(1/2-\xi)(1/2-\kappa)$
1	1	1	$(1/2+\xi)(1/2-\kappa)$

Table 3.1. Table of probability values for Example 3.2.

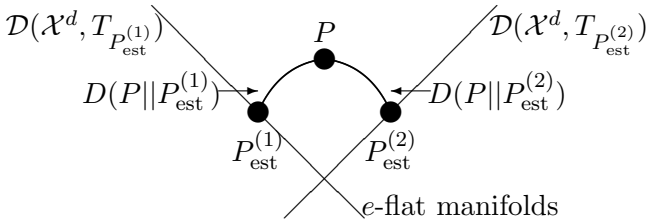

Figure 3.8. Each tree defines an e-flat submanifold [7, 8] of probability distributions. These are the two lines as shown in the figure. If the KL-divergences $D(P||P_{\text{est}}^{(1)})$ and $D(P||P_{\text{est}}^{(2)})$ are equal, then $P_{\text{est}}^{(1)}$ and $P_{\text{est}}^{(2)}$ do not have the same structure but both are optimal with respect to the optimization problem in (3.43). An example of such a distribution P is provided in Example 3.2.

The proof of this proposition is provided in Appendix 3.E where we show that $I(P_{1,2}) > I(P_{2,3}) = I(P_{1,3})$ for sufficiently small κ. Thus, the optimal tree structure P^* is not unique. This in fact corresponds to the case where P belongs to the boundary of some set $\mathcal{B}_i \subset \mathcal{P}(\mathcal{X}^d)$ defined in (3.31). See Fig. 3.8 for an information geometric interpretation.

Every tree distribution in $\mathcal{P}^*(P)$ has the maximum sum mutual information weight. More precisely, we have

$$\sum_{e\in E_Q} I(Q_e) = \max_{Q'\in\mathcal{D}(\mathcal{X}^d,\mathcal{T}^d)} \sum_{e\in E_{Q'}} I(Q'_e),\ \forall\, Q \in \mathcal{P}^*(P). \tag{3.45}$$

Given (3.45), we note that when we use a MWST algorithm to find the optimal solution to the problem in (3.43), ties will be encountered during the greedy addition of edges, as demonstrated in Example 3.2. Upon breaking the ties arbitrarily, we obtain some distribution $Q \in \mathcal{P}^*(P)$. We now provide a sequence of useful definitions that lead to definition of a new error event for which we can perform large-deviation analysis.

We denote the set of tree structures[15] corresponding to the distributions in $\mathcal{P}^*(P)$

[15] In fact, each tree defines a so-called *e-flat submanifold* [7, 8] in the set of probability distributions on $\mathcal{X}^d$ and P_{est} lies in both submanifolds. The so-called *m-geodesic* connects P to any of its optimal projection $P_{\text{est}} \in \mathcal{P}^*(P)$.

as

$$\mathcal{T}_{\mathcal{P}^*(P)} \;:=\; \{T_Q \in \mathcal{T}^d : Q \in \mathcal{P}^*(P)\}, \tag{3.46}$$

and term it as the set of *optimal tree projections*. A similar definition applies to the edge sets of optimal tree projections

$$\mathcal{E}_{\mathcal{P}^*(P)} \;:=\; \{E_Q : T_Q = (V, E_Q) \in \mathcal{T}^d, Q \in \mathcal{P}^*(P)\}. \tag{3.47}$$

Since the distribution P is unknown, our goal is to estimate the optimal tree-projection P_{est} using the empirical distribution $\widehat{P}$, where P_{est} is given by

$$P_{\text{est}} := \operatorname*{argmin}_{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)} D(\widehat{P} \,||\, Q). \tag{3.48}$$

If there are many distributions Q, we arbitrarily pick one of them. We will see that by redefining the error event, we will have still a LDP. Finding the reverse I-projection P_{est} can be solved efficiently (in time $\mathcal{O}(d^2 \log d)$) using the Chow-Liu algorithm [42] as described in Section 2.5.2.

We define $T_{P_{\text{est}}} = (V, E_{P_{\text{est}}})$ as the graph of P_{est}, which is the learned tree and redefine the new *error event* as

$$\mathcal{A}_n(\mathcal{P}^*(P)) \;:=\; \left\{E_{P_{\text{est}}} \;\notin\; \mathcal{E}_{\mathcal{P}^*(P)}\right\}. \tag{3.49}$$

Note that this new error event essentially reduces to the original error event $\mathcal{A}_n = \mathcal{A}_n(\{P\})$ in (3.3) if $\mathcal{T}_{\mathcal{P}^*(P)}$ contains only one member. So if the learned structure belongs to $\mathcal{E}_{\mathcal{P}^*(P)}$, there is no error, otherwise an error is declared. We would like to analyze the decay of the error probability of $\mathcal{A}_n(\mathcal{P}^*(P))$ as defined in (3.49), *i.e.*, find the new *error exponent*

$$K_{\mathcal{P}^*(P)} \;:=\; \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}(\mathcal{A}_n(\mathcal{P}^*(P))). \tag{3.50}$$

It turns out that the analysis of the new event $\mathcal{A}_n(\mathcal{P}^*(P))$ is very similar to the analysis performed in Section 3.4. We redefine the notion of a dominant replacement edge and the computation of the new rate $K_{\mathcal{P}^*(P)}$ then follows automatically.

Definition 3.5 (Dominant Replacement Edge)**.** *Fix an edge set $E_Q \in \mathcal{E}_{\mathcal{P}^*(P)}$. For the error event $\mathcal{A}_n(\mathcal{P}^*(P))$ defined in* (3.49)*, given a non-neighbor node pair $e' \notin E_Q$, its dominant replacement edge $r(e'; E_Q)$ with respect to E_Q, is given by*

$$r(e'; E_Q) \;:=\; \operatorname*{argmin}_{\substack{e \in \text{Path}(e'; E_Q) \\ E_Q \cup \{e'\} \setminus \{e\} \notin \mathcal{E}_{\mathcal{P}^*(P)}}} J_{e,e'}, \tag{3.51}$$

if there exists an edge $e \in \text{Path}(e'; E_Q)$ such that $E_Q \cup \{e'\} \setminus \{e\} \notin \mathcal{E}_{\mathcal{P}^*(P)}$. Otherwise $r(e'; E_Q) = \emptyset$. $J_{e,e'}$ is the crossover rate of mutual information quantities defined in (3.7)*. If $r(e'; E_Q)$ exists, the corresponding crossover rate is*

$$J_{r(e';E_Q),e'} \;=\; \min_{\substack{e \in \text{Path}(e'; E_Q) \\ E_Q \cup \{e'\} \setminus \{e\} \notin \mathcal{E}_{\mathcal{P}^*(P)}}} J_{e,e'}, \tag{3.52}$$

otherwise $J_{\emptyset,e'} = +\infty$.

In (3.51), we are basically fixing an edge set $E_Q \in \mathcal{E}_{\mathcal{P}^*(P)}$ and excluding the trees with $e \in \mathrm{Path}(e'; E_Q)$ replaced by e' if it belongs to the set of optimal tree projections $\mathcal{T}_{\mathcal{P}^*(P)}$. We further remark that in (3.51), $r(e')$ may not necessarily exist. Indeed, this occurs if every tree with $e \in \mathrm{Path}(e'; E_Q)$ replaced by e' belongs to the set of optimal tree projections. This is, however, *not* an error by the definition of the error event in (3.49) hence, we set $J_{\emptyset,e'} = +\infty$. In addition, we define the *dominant non-edge* associated to edge set $E_Q \in \mathcal{E}_{\mathcal{P}^*(P)}$ as:

$$e^*(E_Q) := \operatorname*{argmin}_{e' \notin E_Q} \min_{\substack{e \in \mathrm{Path}(e'; E_Q) \\ E_Q \cup \{e'\} \setminus \{e\} \notin \mathcal{E}_{\mathcal{P}^*(P)}}} J_{e,e'}. \tag{3.53}$$

Also, the *dominant structure* in the set of optimal tree projections is defined as

$$E_{P^*} := \operatorname*{argmin}_{E_Q \in \mathcal{E}_{\mathcal{P}^*(P)}} J_{r(e^*(E_Q); E_Q), e^*(E_Q)}, \tag{3.54}$$

where the crossover rate $J_{r(e';E_Q),e'}$ is defined in (3.52) and the dominant non-edge $e^*(E_Q)$ associated to E_Q is defined in (3.53). Equipped with these definitions, we are now ready to state the generalization of Theorem 3.4.

Theorem 3.10 (Dominant Error Tree)**.** *For the error event $\mathcal{A}_n(\mathcal{P}^*(P))$ defined in* (3.49)*, a dominant error tree (which may not be unique) has edge set given by*

$$E_{P^*} \cup \{e^*(E_{P^*})\} \setminus \{r(e^*(E_{P^*}); E_{P^*})\}, \tag{3.55}$$

where $e^*(E_{P^*})$ is the dominant non-edge associated to the dominant structure $E_{P^*} \in \mathcal{E}_{\mathcal{P}^*(P)}$ and is defined by (3.53) *and* (3.54)*. Furthermore, the error exponent $K_{\mathcal{P}^*(P)}$, defined in* (3.50) *is given as*

$$K_{\mathcal{P}^*(P)} = \min_{E_Q \in \mathcal{E}_{\mathcal{P}^*(P)}} \min_{e' \notin E_Q} \min_{\substack{e \in \mathrm{Path}(e'; E_Q) \\ E_Q \cup \{e'\} \setminus \{e\} \notin \mathcal{E}_{\mathcal{P}^*(P)}}} J_{e,e'}. \tag{3.56}$$

Proof. The proof of this theorem follows directly by identifying the dominant error tree belonging to the set $\mathcal{T}^d \setminus \mathcal{T}_{\mathcal{P}^*(P)}$. By further applying the result in Proposition 3.3 and Theorem 3.4, we obtain the result via the "worst-exponent-wins" [62, Ch. 1] principle by minimizing over all trees in the set of optimal projections $\mathcal{E}_{\mathcal{P}^*(P)}$ in (3.56). □

This theorem now allows us to analyze the more general error event $\mathcal{A}_n(\mathcal{P}^*(P))$, which includes $\mathcal{A}_n$ in (3.3) as a special case if the set of optimal tree projections $\mathcal{T}_{\mathcal{P}^*(P)}$ in (3.46) is a singleton.

■ 3.7 Numerical Experiments

In this section, we perform a series of numerical experiments with the following three objectives:

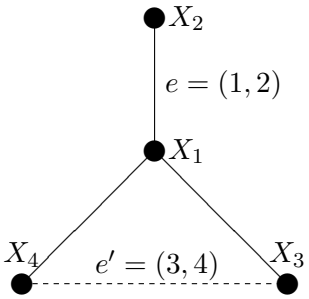

Figure 3.9. Graphical model used for our numerical experiments. The true model is a symmetric star (cf. Section 3.3) in which the mutual information quantities satisfy $I(P_{1,2}) = I(P_{1,3}) = I(P_{1,4})$ and by construction, $I(P_{e'}) < I(P_{1,2})$ for any non-edge e'. Besides, the mutual information quantities on the non-edges are equal, for example, $I(P_{2,3}) = I(P_{3,4})$.

1. In Section 3.7.1, we study the accuracy of the Euclidean approximations (Theorem 3.7). We do this by analyzing under which regimes the approximate crossover rate $\widetilde{J}_{e,e'}$ in (3.42) is close to the true crossover rate $J_{e,e'}$ in (3.8).

2. Since the LDP and error exponent analysis are asymptotic theories, in Section 3.7.2 we use simulations to study the behavior of the actual crossover rate, given a finite number of samples n. In particular, we study how fast the crossover rate, obtained from simulations, converges to the true crossover rate. To do so, we generate a number of samples from the true distribution and use the Chow-Liu algorithm to learn trees structures. Then we compare the result to the true structure and finally compute the error probability.

3. In Section 3.7.3, we address the issue of the learner not having access to the true distribution, but nonetheless wanting to compute an estimate of the crossover rate. The learner only has the samples $\mathbf{x}^n$ or equivalently, the empirical distribution $\widehat{P}$. However, in all the preceding analysis, to compute the true crossover rate $J_{e,e'}$ and the overall error exponent K_P, we used the true distribution P and solved the constrained optimization problem in (3.8). Alternatively we computed the approximation in (3.42), which is also a function of the true distribution. However, in practice, it is also useful to compute an online estimate of the crossover rate by using the empirical distribution in place of the true distribution in the constrained optimization problem in (3.8). This is an estimate of the rate that the learner can compute given the samples. We call this the *empirical rate* and formally define it in Section 3.7.3. We perform convergence analysis of the empirical rate and also numerically verify the rate of convergence to the true crossover rate.

In the following, we will be performing numerical experiments for the undirected graphical model with four nodes as shown in Fig. 3.9. We parameterize the distribution with $d = 4$ variables with a single parameter $\gamma > 0$ and let $\mathcal{X} = \{0, 1\}$, *i.e.*, all the

variables are binary. For the parameters, we set $P_1(x_1 = 0) = 1/3$ and

$$P_{i|1}(x_i = 0|x_1 = 0) = \frac{1}{2} + \gamma, \quad i = 2, 3, 4, \tag{3.57a}$$

$$P_{i|1}(x_i = 0|x_1 = 1) = \frac{1}{2} - \gamma, \quad i = 2, 3, 4. \tag{3.57b}$$

With this parameterization, we see that if γ is small, the mutual information $I(P_{1,i})$ for $i = 2, 3, 4$ is also small. In fact if $\gamma = 0$, x_1 is independent of x_i for $i = 2, 3, 4$ and as a result, $I(P_{1,i}) = 0$. Conversely, if γ is large, the mutual information $I(P_{1,i})$ increases as the dependence of the outer nodes with the central node increases. Thus, we can vary the size of the mutual information along the edges by varying γ. By symmetry, there is only one crossover rate and hence this crossover rate is also the error exponent for the error event $\mathcal{A}_n$ in (3.3). This is exactly the same as the symmetric star graph as described in Section 3.3.

■ 3.7.1 Accuracy of Euclidean Approximations

We first study the accuracy of the Euclidean approximations used to derive the result in Theorem 3.7. We denote the *true rate* as the crossover rate resulting from the non-convex optimization problem (3.8) and the *approximate rate* as the crossover rate computed using the approximation in (3.42).

We vary γ from 0 to 0.2 and plot both the true and approximate rates against the difference between the mutual informations $I(P_e) - I(P_{e'})$ in Fig. 3.10, where e denotes any edge and e' denotes any non-edge in the model. The non-convex optimization problem was performed using the Matlab function `fmincon` in the optimization toolbox. We used several different feasible starting points and chose the best optimal objective value to avoid problems with local minima. We first note from Fig. 3.10 that both rates increase as $I(P_e) - I(P_{e'})$ increases. This is in line with our intuition because if $P_{e,e'}$ is such that $I(P_e) - I(P_{e'})$ is large, the crossover rate is also large. We also observe that if $I(P_e) - I(P_{e'})$ is small, the true and approximate rates are very close. This is in line with the assumptions for Theorem 3.7. Recall that if $P_{e,e'}$ satisfies the ϵ-very noisy condition (for some small ϵ), then the mutual information quantities $I(P_e)$ and $I(P_{e'})$ are close and consequently the true and approximate crossover rates are also close. When the difference between the mutual informations increases, the true and approximate rate separate from each other.

■ 3.7.2 Comparison of True Crossover Rate to the Rate obtained from Simulations

In this section, we compare the true crossover rate in (3.8) to the rate we obtain when we learn tree structures using Chow-Liu with i.i.d. samples drawn from P, which we define as the *simulated rate*. We fixed $\gamma > 0$ in (3.57) then for each n, we estimated the probability of error using the Chow-Liu algorithm as described in Section 2.5.2. We state the procedure precisely in the following steps.

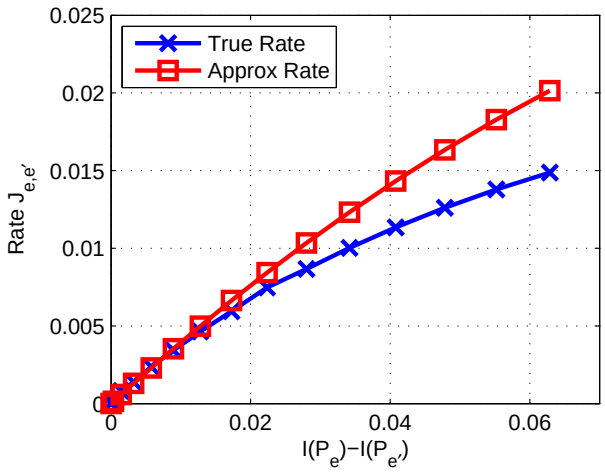

Figure 3.10. Comparison of True and Approximate Rates.

1. Fix $n \in \mathbb{N}$ and sample n i.i.d. observations $\mathbf{x}^n$ from P.
2. Compute the empirical distribution $\widehat{P}$ and the set of empirical mutual information quantities $\{I(\widehat{P}_e) : e \in \binom{\mathcal{V}}{2}\}$.
3. Learn the Chow-Liu tree $\mathcal{E}_{\mathrm{ML}}$ using a MWST algorithm with $\{I(\widehat{P}_e) : e \in \binom{\mathcal{V}}{2}\}$ as the edge weights.
4. If $\mathcal{E}_{\mathrm{ML}}$ is not equal to $\mathcal{E}_P$, then we declare an error.
5. Repeat steps 1 – 4 a total of $M \in \mathbb{N}$ times and estimate the probability of error $\mathbb{P}(\mathcal{A}_n) = \#\text{errors}/M$ and the error exponent $-(1/n)\log \mathbb{P}(\mathcal{A}_n)$, which is the simulated rate.

If the probability of error $\mathbb{P}(\mathcal{A}_n)$ is very small, then the number of runs M to estimate $\mathbb{P}(\mathcal{A}_n)$ has to be fairly large. This is often the case in error exponent analysis as the sample size needs to be substantial to estimate very small error probabilities.

In Fig. 3.11, we plot the true rate, the approximate rate and the simulated rate when $\gamma = 0.01$ (and $M = 10^7$) and $\gamma = 0.2$ (and $M = 5 \times 10^8$). Note that, in the former case, the true rate is higher than the approximate rate and in the latter case, the reverse is true. When γ is large ($\gamma = 0.2$), there are large differences in the true tree models. Thus, we expect that the error probabilities to be very small and hence M has to be large in order to estimate the error probability correctly but n does not have to be too large for the simulated rate to converge to the true rate. On the other hand, when γ is small ($\gamma = 0.01$), there are only subtle differences in the graphical models, hence we need a larger number of samples n for the simulated rate to converge to its true value, but M does not have to be large since the error probabilities are not small. The above observations are in line with our intuition.

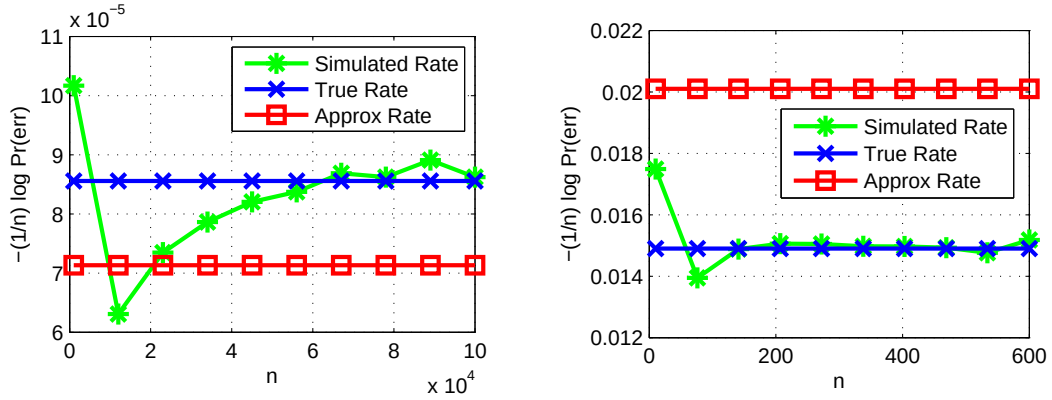

Figure 3.11. Comparison of True, Approximate and Simulated Rates with $\gamma = 0.01$ (top) and $\gamma = 0.2$ (bottom). Here the number of runs $M = 10^7$ for $\gamma = 0.01$ and $M = 5 \times 10^8$ for $\gamma = 0.2$. The probability of error is computed dividing the total number of errors by the total number of runs.

■ 3.7.3 Comparison of True Crossover Rate to Rate obtained from the Empirical Distribution

In this subsection, we compare the true rate to the *empirical rate*, which is defined as

$$\widehat{J}_{e,e'} := \inf_{Q \in \mathcal{P}(\mathcal{X}^4)} \left\{ D(Q \,||\, \widehat{P}_{e,e'}) : I(Q_{e'}) = I(Q_e) \right\}. \tag{3.58}$$

The empirical rate $\widehat{J}_{e,e'} = \widehat{J}_{e,e'}(\widehat{P}_{e,e'})$ is a function of the empirical distribution $\widehat{P}_{e,e'}$. This rate is computable by a learner, who does not have access to the true distribution P. The learner only has access to a finite number of samples $\mathbf{x}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Given $\mathbf{x}^n$, the learner can compute the empirical probability $\widehat{P}_{e,e'}$ and perform the optimization in (3.58). This is an estimate of the true crossover rate. A natural question to ask is the following: Does the empirical rate $\widehat{J}_{e,e'}$ converge to the true crossover rate $J_{e,e'}$ as $n \to \infty$? The next theorem answers this question in the affirmative.

Theorem 3.11 (Crossover Rate Consistency)**.** *The empirical crossover rate* $\widehat{J}_{e,e'}$ *in* (3.58) *converges almost surely to the true crossover rate* $J_{e,e'}$ *in* (3.8)*, i.e.,*

$$\mathbb{P}\left(\lim_{n \to \infty} \widehat{J}_{e,e'} = J_{e,e'} \right) = 1. \tag{3.59}$$

Proof. (*Sketch*) The proof of this theorem follows from the continuity of $\widehat{J}_{e,e'}$ in the empirical distribution $\widehat{P}_{e,e'}$ and the continuous mapping theorem by Mann and Wald [134]. See Appendix 3.F for the details. □

We conclude that the learning of the rate from samples is consistent. Now we perform simulations to determine how many samples are required for the empirical rate to converge to the true rate.

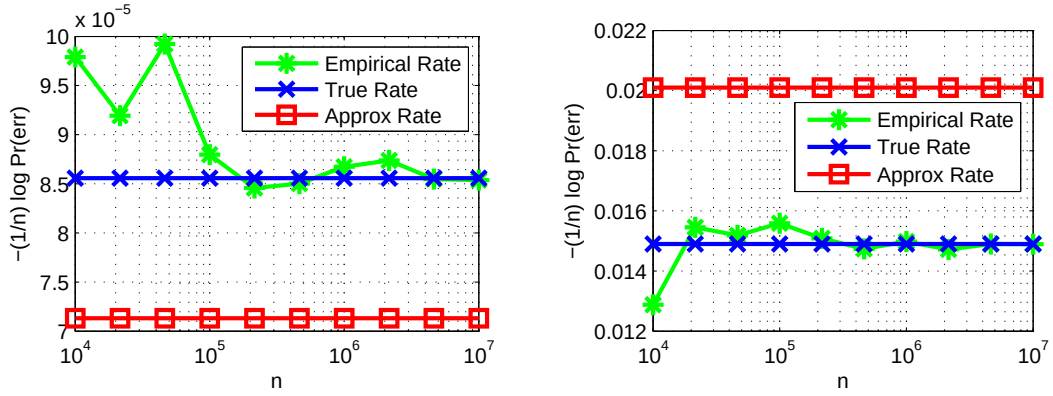

Figure 3.12. Comparison of True, Approximate and Empirical Rates with $\gamma = 0.01$ (top) and $\gamma = 0.2$ (bottom). Here n is the number of observations used to estimate the empirical distribution.

We set $\gamma = 0.01$ and $\gamma = 0.2$ in (3.57). We then drew n i.i.d. samples from P and computed the empirical distribution $\widehat{P}_{e,e'}$. Next, we solved the optimization problem in (3.58) using the `fmincon` function in Matlab, using different initializations and compared the empirical rate to the true rate. We repeated this for several values of n and the results are displayed in Fig. 3.12. We see that for $\gamma = 0.01$, approximately $n = 8 \times 10^6$ samples are required for the empirical distribution to be close enough to the true distribution so that the empirical rate converges to the true rate.

■ 3.8 Chapter Summary

In this chapter, we presented a solution to the problem of finding the error exponent for tree structure learning by extensively using tools from large-deviations theory combined with facts about tree graphs. We quantified the error exponent for learning the structure and exploited the structure of the true tree to identify the dominant tree in the set of erroneous trees. We also drew insights from the approximate crossover rate, which can be interpreted as the SNR for learning. These two main results in Theorems 3.4 and 3.7 provide the intuition as to how errors occur for learning discrete tree distributions via the Chow-Liu algorithm.

Recall that we applied the Euclidean approximation to the mutual information, which is a function of the joint distribution of pairs of edges in Section 3.5. An interesting line of further research is to consider each edge of the tree as a communication channel with an input X_i, a *channel* $P_{X_j|X_i}$ and an output X_j. The very-noisy assumption can also be equivalently applied to the channel, i.e., one assumes that $P_{X_j|X_i} \approx P_{X_j}$. This was explored in [26].

In the next chapter, we develop counterparts to the results here for the Gaussian case. Many of the results carry through but thanks to the special structure that Gaussian distributions possess, we are also able to identify which structures are easier to

learn and which are harder to learn given a fixed set of correlation coefficients.

Appendices for Chapter 3

■ 3.A Proof of Theorem 3.1

Proof. We divide the proof of this theorem into three steps. Steps 1 and 2 prove the expression in (3.8). Step 3 proves the existence of the optimizer.

Step 1: First, we note from Sanov's Theorem [47, Ch. 11] that the empirical joint distribution on edges e and e' satisfies

$$\lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}(\widehat{P}_{e,e'} \in \mathcal{F}) = \inf\{D(Q \,||\, P_{e,e'}) : Q \in \mathcal{F}\} \tag{3.60}$$

for any set $\mathcal{F} \subset \mathcal{P}(\mathcal{X}^4)$ that equals the closure of its interior, *i.e.*, $\mathcal{F} = \mathrm{cl}(\mathrm{int}(\mathcal{F}))$. We now have a LDP for the sequence of probability measures $\widehat{P}_{e,e'}$, the empirical distribution on (e, e'). Assuming that e and e' do not share a common node, $\widehat{P}_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$ is a probability distribution over four variables (the variables in the node pairs e and e'). We now define the function $h : \mathcal{P}(\mathcal{X}^4) \to \mathbb{R}$ as

$$h(Q) \;:=\; I(Q_{e'}) - I(Q_e). \tag{3.61}$$

Since $Q_e = \sum_{x_{e'}} Q$, defined in (3.9) is continuous in Q and the mutual information $I(Q_e)$ is also continuous in Q_e, we conclude that h is indeed continuous, since it is the composition of continuous functions. By applying the contraction principle [62] to the sequence of probability measures $\widehat{P}_{e,e'}$ and the continuous map h, we obtain a corresponding LDP for the new sequence of probability measures $h(\widehat{P}_{e,e'}) = I(\widehat{P}_{e'}) - I(\widehat{P}_e)$, where the rate is given by:

$$J_{e,e'} = \inf_{Q\in\mathcal{P}(\mathcal{X}^4)} \left\{D(Q \,||\, P_{e,e'}) : h(Q) \geq 0\right\}, \tag{3.62}$$

$$= \inf_{Q\in\mathcal{P}(\mathcal{X}^4)} \left\{D(Q \,||\, P_{e,e'}) : I(Q_{e'}) \geq I(Q_e)\right\}. \tag{3.63}$$

We now claim that the limit in (3.7) exists. From Sanov's theorem [47, Ch. 11], it suffices to show that the constraint set $\mathcal{F} := \{I(Q_{e'}) \geq I(Q_e)\}$ in (3.63) is a regular closed set, *i.e.*, it satisfies $\mathcal{F} = \mathrm{cl}(\mathrm{int}(\mathcal{F}))$. This is true because there are no isolated points in $\mathcal{F}$ and thus the interior is nonempty. Hence, there exists a sequence of distributions $\{Q_n\}_{n=1}^{\infty} \subset \mathrm{int}(\mathcal{F})$ such that $\lim_{n\to\infty} D(Q_n||P_{e,e'}) = D(Q^*||P_{e,e'})$, which proves the existence of the limit in (3.7).

Step 2: We now show that the optimal solution $Q^*_{e,e'}$, if it exists (as will be shown in Step 3), must satisfy $I(Q^*_e) = I(Q^*_{e'})$. Suppose, to the contrary, that $Q^*_{e,e'}$ with objective value $D(Q^*_{e,e'}||P_{e,e'})$ is such that $I(Q^*_{e'}) > I(Q^*_e)$. Then $h(Q^*_{e,e'}) > 0$, where h, as shown above, is continuous. Thus, there exists a $\delta > 0$ such that the δ-neighborhood

$$N_\delta(Q^*_{e,e'}) := \{R : \|R - Q^*_{e,e'}\|_\infty < \delta\}, \tag{3.64}$$

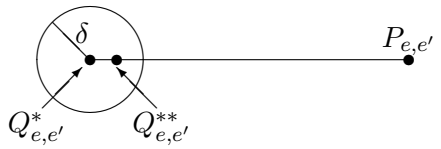

Figure 3.13. Illustration of Step 2 of the proof of Theorem 3.1.

satisfies $h(N_\delta(Q^*_{e,e'})) \subset (0,\infty)$ [166, Ch. 2]. Consider the new distribution (See Fig. 3.13)

$$Q^{**}_{e,e'} = Q^*_{e,e'} + \frac{\delta}{2}(P_{e,e'} - Q^*_{e,e'}) \tag{3.65}$$

$$= \left(1 - \frac{\delta}{2}\right) Q^*_{e,e'} + \frac{\delta}{2} P_{e,e'}. \tag{3.66}$$

Note that $Q^{**}_{e,e'}$ belongs to $N_\delta(Q^*_{e,e'})$ and hence is a feasible solution of (3.63). We now prove that $D(Q^{**}_{e,e'}||P_{e,e'}) < D(Q^*_{e,e'}||P_{e,e'})$, which contradicts the optimality of $Q^*_{e,e'}$.

$$\begin{aligned}
&D(Q^{**}_{e,e'} \,||\, P_{e,e'}) \\
&= D\left(\left(1 - \frac{\delta}{2}\right) Q^*_{e,e'} + \frac{\delta}{2} P_{e,e'} \,\middle\|\, P_{e,e'}\right), && (3.67)\\
&\leq \left(1 - \frac{\delta}{2}\right) D(Q^*_{e,e'} \,||\, P_{e,e'}) + \frac{\delta}{2} D(P_{e,e'} \,||\, P_{e,e'}), && (3.68)\\
&= \left(1 - \frac{\delta}{2}\right) D(Q^*_{e,e'} \,||\, P_{e,e'}) && (3.69)\\
&< D(Q^*_{e,e'} \,||\, P_{e,e'}), && (3.70)
\end{aligned}$$

where (3.68) is due to the convexity of the KL-divergence in the first variable [47, Ch. 2], (3.69) is because $D(P_{e,e'}||P_{e,e'}) = 0$ and (3.70) is because $\delta > 0$. Thus, we conclude that the optimal solution must satisfy $I(Q^*_e) = I(Q^*_{e'})$ and the crossover rate can be stated as (3.8).

Step 3: Now, we prove the existence of the minimizer $Q^*_{e,e'}$, which will allow us to replace the inf in (3.8) with min. First, we note that $D(Q \,||\, P_{e,e'})$ is continuous in both variables and hence continuous and the first variable Q. It remains to show that the constraint set

$$\Lambda := \{Q \in \mathcal{P}(\mathcal{X}^4) : I(Q_{e'}) = I(Q_e)\} \tag{3.71}$$

is compact, since it is clearly nonempty (the uniform distribution belongs to Λ). Then we can conclude, by Weierstrass' extreme value theorem [166, Theorem 4.16], that the minimizer $Q^* \in \Lambda$ exists. By the Heine-Borel theorem [166, Theorem 2.41], it suffices to show that Λ is bounded and closed. Clearly Λ is bounded since $\mathcal{P}(\mathcal{X}^4)$ is a bounded set. Now, $\Lambda = h^{-1}(\{0\})$ where h is defined in (3.61). Since h is continuous and $\{0\}$ is closed (in the usual topology of the real line), Λ is closed [166, Theorem 4.8]. Hence that Λ is compact. We also need to use the fact that Λ is compact in the proof of Theorem 3.11. □

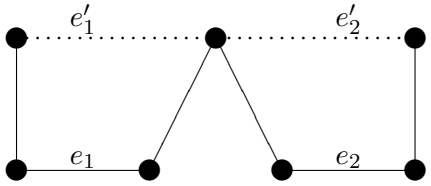

Figure 3.14. Illustration of the proof of Theorem 3.4.

■ 3.B Proof of Theorem 3.4

Proof. We first claim that E_P^*, the edge set corresponding to the dominant error tree, differs from E_P by exactly one edge.[16] To prove this claim, assume, to the contrary, that E_P^* differs from E_P by two edges. Let $E_{\mathrm{ML}} = \mathcal{E}' := E_P \setminus \{e_1, e_2\} \cup \{e'_1, e'_2\}$, where $e'_1, e'_2 \notin E_P$ are the two edges that have replaced $e_1, e_2 \in E_P$ respectively. Since $T' = (V, \mathcal{E}')$ is a tree, these edges cannot be arbitrary and specifically, $\{e_1, e_2\} \in \{\mathrm{Path}(e'_1; E_P) \cup \mathrm{Path}(e'_2; E_P)\}$ for the tree constraint to be satisfied. Recall that the rate of the event that the output of the ML algorithm is T' is given by $\Upsilon(T')$ in (3.17). Then consider the probability of the joint event (with respect to the probability measure $\mathbb{P} = P^n$).

Suppose that $e_i \in \mathrm{Path}(e'_i; E_P)$ for $i = 1, 2$ and $e_i \notin \mathrm{Path}(e'_j; E_P)$ for $i, j = 1, 2$ and $i \neq j$. See Fig. 3.14. Note that the true mutual information quantities satisfy $I(P_{e_i}) > I(P_{e'_i})$. We prove this claim by contradiction that suppose $I(P_{e'_i}) \geq I(P_{e_i})$ then, E_P does not have maximum weight because if the non-edge e'_i replaces the true edge e_i, the resulting tree[17] would have higher weight, contradicting the optimality of the true edge set E_P, which is the MWST with the true mutual information quantities as edge weights. More precisely, we can compute the exponent when T' is the output of the MWST algorithm:

$$\Upsilon(T') = \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}\left(\bigcap_{i=1,2} \{I(\widehat{P}_{e'_i}) \geq I(\widehat{P}_{e_i})\} \right), \tag{3.72}$$

$$\geq \max_{i=1,2} \lim_{n\to\infty} -\frac{1}{n} \log \mathbb{P}\left(\{I(\widehat{P}_{e'_i}) \geq I(\widehat{P}_{e_i})\} \right), \tag{3.73}$$

$$= \max\left\{ J_{e_1,e'_1}\,,\, J_{e_2,e'_2} \right\}. \tag{3.74}$$

Now $J_{e_i,e'_i} = \Upsilon(T_i)$ where $T_i := (V, E_P \setminus \{e_i\} \cup \{e'_i\})$. From Prop. 3.3, the error exponent associated to the dominant error tree, *i.e.*, $K_P = \min_{T \neq T_P} \Upsilon(T)$ and from (3.74), the dominant error tree cannot be T' and should differ from T_P by one and only one edge.

The similar conclusion holds for the two other cases (i) $e_i \in \mathrm{Path}(e'_i; E_P)$ for $i = 1, 2$, $e_2 \in \mathrm{Path}(e'_1; E_P)$ and $e_1 \notin \mathrm{Path}(e'_2; E_P)$ and (ii) $e_i \in \mathrm{Path}(e'_i; E_P)$ for $i = 1, 2$,

[16] This is somewhat analogous to the fact that the second-best MWST differs from the MWST by exactly one edge [45].

[17] The resulting graph is indeed a tree because $\{e'_i\} \cup \mathrm{Path}(e'_i; E_P)$ form a cycle so if any edge is removed, the resulting structure does not have any cycles and is connected, hence it is a tree. See Fig. 3.2.

$e_1 \in \text{Path}(e_2'; E_P)$ and $e_2 \notin \text{Path}(e_1'; E_P)$. In other words, the dominant error tree differs from the true tree by one edge.

We now use the "worst-exponent-wins principle" [62, Ch. 1], to conclude that the rate that dominates is the minimum $J_{r(e'),e'}$ over all possible $e' \notin E_P$, namely $J_{r(e^*),e^*}$ with e^* defined in (3.24). More precisely,

$$\mathbb{P}(\mathcal{A}_n) = \mathbb{P}\left(\bigcup_{e' \notin E_P} \{e' \text{ replaces any } e \in \text{Path}(e'; E_P) \text{ in } \widehat{T}_{\text{ML}}\} \right), \tag{3.75}$$

$$= \mathbb{P}\left(\bigcup_{e' \notin E_P} \bigcup_{e \in \text{Path}(e'; E_P)} \{e' \text{ replaces } e \text{ in } \widehat{T}_{\text{ML}}\} \right), \tag{3.76}$$

$$\leq \sum_{e' \notin E_P} \sum_{e \in \text{Path}(e'; E_P)} \mathbb{P}(\{e' \text{ replaces } e \text{ in } \widehat{T}_{\text{ML}}\}), \tag{3.77}$$

$$= \sum_{e' \notin E_P} \sum_{e \in \text{Path}(e'; E_P)} \mathbb{P}(\{I(\widehat{P}_{e'}) \geq I(\widehat{P}_e)\}), \tag{3.78}$$

$$\doteq \sum_{e' \notin E_P} \sum_{e \in \text{Path}(e'; E_P)} \exp(-n J_{e,e'}), \tag{3.79}$$

$$\doteq \exp\left(-n \min_{e' \notin E_P} \min_{e \in \text{Path}(e'; E_P)} J_{e,e'} \right), \tag{3.80}$$

where (3.77) is from the union bound, (3.78) and (3.79) are from the definitions of the crossover event and rate respectively (as described in Cases 1 and 2 above) and (3.80) is an application of the "worst-exponent-wins" principle [62, Ch. 1].

We conclude from (3.80) that

$$\mathbb{P}(\mathcal{A}_n) \dot{\leq} \exp(-n J_{r(e^*),e^*}), \tag{3.81}$$

from the definition of the dominant replacement edge $r(e')$ and the dominant non-edge e^*, defined in (3.22) and (3.24) respectively. The lower bound follows trivially from the fact that if $e^* \notin E_P$ replaces $r(e^*)$, then the error $\mathcal{A}_n$ occurs. Thus, $\{e^* \text{ replaces } r(e^*)\} \subset \mathcal{A}_n$ and

$$\mathbb{P}(\mathcal{A}_n) \dot{\geq} \mathbb{P}(\{e^* \text{ replaces } r(e^*) \text{ in } \widehat{T}_{\text{ML}}\}) \tag{3.82}$$

$$\doteq \exp(-n J_{r(e^*),e^*}). \tag{3.83}$$

Hence, (3.81) and (3.83) imply that $\mathbb{P}(\mathcal{A}_n) \doteq \exp(-n J_{r(e^*),e^*})$, which proves our main result in (3.23).

The finite-sample result in (3.26) comes from the upper bound in (3.80) and the following two elementary facts:

1. The exact number of n-types with alphabet $\mathcal{Y}$ is given by $\binom{n+1+|\mathcal{Y}|}{n+1}$ [50]. In particular, we have

$$\mathbb{P}(\mathcal{C}_{e,e'}) \leq \binom{n+1+|\mathcal{X}|^4}{n+1} \exp(-n J_{e,e'}), \tag{3.84}$$

for all $n \in \mathbb{N}$, since $\mathcal{C}_{e,e'}$ only involves the distribution $P_{e,e'} \in \mathcal{P}(\mathcal{X}^4)$. Note that the exponent 4 of $|\mathcal{X}|^4$ in (3.84) is an upper bound since if e and e' share a node $P_{e,e'} \in \mathcal{P}(\mathcal{X}^3)$.

2. The number of error events $\mathcal{C}_{e,e'}$ is at most $(d-1)^2(d-2)/2$ because there are $\binom{d}{2} - (d-1) = (d-1)(d-2)/2$ non-edges and for each non-edge, there are at most $d-1$ edges along its path.

This completes the proof. □

■ 3.C Proof of Theorem 3.5

Statement (a) $\Leftrightarrow$ statement (b) was proven in full after the theorem was stated. Here we provide the proof that (b) $\Leftrightarrow$ (c). Recall that statement (c) says that T_P is not a proper forest. We first begin with a preliminary lemma.

Lemma 3.12. *Suppose X, Y, Z are three random variables taking on values on finite sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ respectively. Assume that $P(x, y, z) > 0$ everywhere. Then $X - Y - Z$ and $X - Z - Y$ are Markov chains if and only if X is jointly independent of (Y, Z).*

Proof. ($\Rightarrow$) That $x - y - z$ is a Markov chain implies that

$$P(z|y, x) = P(z|y), \tag{3.85}$$

or alternatively

$$P(x, y, z) = P(x, y)\frac{P(y, z)}{P(y)}. \tag{3.86}$$

Similarly from the fact that $x - z - y$ is a Markov chain, we have

$$P(x, y, z) = P(x, z)\frac{P(y, z)}{P(z)}. \tag{3.87}$$

Equating (3.86) and (3.87), and use the positivity to cancel $P(y, z)$, we arrive at

$$P(x|y) = P(x|z). \tag{3.88}$$

It follows that $P(x|y)$ does not depend on y, so there is some constant $C(x)$ such that $P(x|y) = C(x)$ for all $y \in \mathcal{Y}$. This immediately implies that $C(x) = P(x)$ so that $P(x|y) = P(x)$. A similar argument gives that $P(x|z) = P(x)$. Furthermore, if $X - Y - Z$ is a Markov chain, so is $Z - Y - X$, therefore

$$P(x|y, z) = P(x|y) = P(x). \tag{3.89}$$

The above equation says that X is jointly independent of both Y and Z.
($\Leftarrow$) The reverse implication is clear. □

Proof. We now prove (b) $\iff$ (c) using Lemma 3.12 and the assumption that $P(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^d$.
($\Rightarrow$) If (b) is true then $I(P_{e'}) < I(P_e)$ for all $e \in \text{Path}(e'; E_P)$ and for all $e' \notin E_P$. Assume, to the contrary, that T_P is a proper forest, *i.e.*, it contains at least 2 connected components (each connected component may only have one node), say $\mathcal{G}_i = (V_i, \mathcal{E}_i)$ for $i = 1, 2$. Without loss of generality, let X_1 be in component $\mathcal{G}_1$ and X_2, X_3 belong to component $\mathcal{G}_2$. Then since $V_1 \cap V_2 = \emptyset$ and $V_1 \cup V_2 = V$, we have that X_1 jointly independent of X_2 and X_3. By Lemma 3.12, we have the following Markov chains $X_1 - X_2 - X_3$ and $X_1 - X_3 - X_2$. This implies from the Data Processing Inequality [47, Theorem 2.8.1] that $I(P_{1,2}) \geq I(P_{1,3})$ and at the same time $I(P_{1,2}) \leq I(P_{1,3})$ which means that $I(P_{1,2}) = I(P_{1,3})$. This contradicts (b) since by taking $e' = (1, 2)$, the mutual informations along the path $\text{Path}(e'; E_P)$ are no longer distinct.
($\Leftarrow$) Now assume that (c) is true, *i.e.*, T_P is not a proper forest. Suppose, to the contrary, (b) is not true, *i.e.*, there exists a $e' \notin E_P$ such that $I(P_{e'}) = I(P_{r(e')})$, where $r(e')$ is the replacement edge associated with the non-edge e'. Without loss of generality, let $e' = (1, 2)$ and $r(e') = (3, 4)$, then since T_P is not a proper forest, we have the following Markov chain $X_1 - X_3 - X_4 - X_2$. Now note that $I(P_{1,2}) = I(P_{3,4})$. In fact, because there is no loss of mutual information $I(P_{1,4}) = I(P_{3,4})$ and hence by the Data Processing Inequality we also have $X_3 - X_1 - X_4 - X_2$. By using Lemma 3.12, we have X_4 jointly independent of X_1 and X_3, hence we have a proper forest, which is a contradiction. □

■ 3.D Proof of Theorem 3.7

Proof. The proof proceeds in several steps. See Figs. 3.5 and 3.6 for intuition behind this proof.

Step 1: Let Q be such that

$$Q(x_i, x_j, x_k, x_l) = P_{e,e'}(x_i, x_j, x_k, x_l) + \epsilon_{i,j,k,l}. \tag{3.90}$$

Thus, the $\epsilon_{i,j,k,l}$'s are the deviations of Q from $P_{e,e'}$. To ensure that Q is a valid distribution we require $\sum \epsilon_{i,j,k,l} = 0$. The objective in (3.39) can now be alternatively expressed as

$$\frac{1}{2}\boldsymbol{\epsilon}^T \mathbf{K}_{e,e'} \boldsymbol{\epsilon} = \frac{1}{2} \sum_{x_i, x_j, x_k, x_l} \frac{\epsilon_{i,j,k,l}^2}{P_{e,e'}(x_i, x_j, x_k, x_l)}, \tag{3.91}$$

where $\boldsymbol{\epsilon} \in \mathbb{R}^{|\mathcal{X}|^4}$ is the vectorized version of the deviations $\epsilon_{i,j,k,l}$ and $\mathbf{K}_{e,e'}$ is a $|\mathcal{X}|^4 \times |\mathcal{X}|^4$ diagonal matrix containing the entries $1/P_{e,e'}(x_i, x_j, x_k, x_l)$ along its diagonal.

Step 2: We now perform a first-order Taylor expansion of $I(Q_e)$ in the neighborhood of $I(P_e)$.

$$I(Q_e) = I(P_e) + \boldsymbol{\epsilon}^T \nabla_{\boldsymbol{\epsilon}} I(Q_e)\Big|_{\boldsymbol{\epsilon}=0} + o(\|\boldsymbol{\epsilon}\|), \tag{3.92}$$

$$= I(P_e) + \boldsymbol{\epsilon}^T \mathbf{s}_e + o(\|\boldsymbol{\epsilon}\|), \tag{3.93}$$

where $\mathbf{s}_e$ is the length $|\mathcal{X}|^4$-vector that contains the information density values of edge e. Note that because of the assumption that P is not a proper forest, $P_{i,j} \neq P_i\, P_j$ for all (i,j), hence the linear term does not vanish.[18] The constraints can now be rewritten as

$$\boldsymbol{\epsilon}^T \mathbf{1} \;=\; 0, \qquad \boldsymbol{\epsilon}^T(\mathbf{s}_{e'} - \mathbf{s}_e) \;=\; I(P_e) - I(P_{e'}). \tag{3.94}$$

or in matrix notation as:

$$\left[\begin{array}{c} \mathbf{s}_{e'}^T - \mathbf{s}_e^T \\ \mathbf{1}^T \end{array}\right] \boldsymbol{\epsilon} = \left[\begin{array}{c} I(P_e) - I(P_{e'}) \\ 0 \end{array}\right], \tag{3.95}$$

where $\mathbf{1}$ is the length-$|\mathcal{X}|^4$ vector consisting of all ones. For convenience, we define $\mathbf{L}_{e,e'}$ to be the matrix in (3.95), *i.e.*,

$$\mathbf{L}_{e,e'} := \left[\begin{array}{c} \mathbf{s}_{e'}^T - \mathbf{s}_e^T \\ \mathbf{1}^T \end{array}\right] \in \mathbb{R}^{2\times|\mathcal{X}|^4}. \tag{3.96}$$

Step 3: The optimization problem now reduces to minimizing (3.91) subject to the constraints in (3.95). This is a standard least-squares problem. By using the Projection Theorem in Hilbert spaces, we get the solution

$$\boldsymbol{\epsilon}^* = \mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T(\mathbf{L}_{e,e'}\mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T)^{-1}\left[\begin{array}{c} I(P_e) - I(P_{e'}) \\ 0 \end{array}\right]. \tag{3.97}$$

The inverse of $\mathbf{L}_{e,e'}\mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T$ exists because we assumed T_P is not a proper forest and hence $P_{i,j} \neq P_iP_j$ for all $(i,j) \in \binom{V}{2}$. This is a sufficient condition for the matrix $\mathbf{L}_{e,e'}$ to have full row rank and thus, $\mathbf{L}_{e,e'}\mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T$ is invertible. Finally, we substitute $\boldsymbol{\epsilon}^*$ in (3.97) into (3.91) to obtain

$$\widetilde{J}_{e,e'} = \frac{1}{2}\left[(\mathbf{L}_{e,e'}\mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T)^{-1}\right]_{11}(I(P_e) - I(P_{e'}))^2, \tag{3.98}$$

where $[\mathbf{M}]_{11}$ is the (1,1) element of the matrix $\mathbf{M}$. Define ψ to be the weighting function given by

$$\psi(P_{e,e'}) := \left[(\mathbf{L}_{e,e'}\mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T)^{-1}\right]_{11}. \tag{3.99}$$

It now suffices to show that $\psi(P_{e,e'})$ is indeed the inverse variance of $s_e - s_{e'}$. We now simplify the expression for the weighting function $\psi(P_{e,e'})$ recalling how $\mathbf{L}_{e,e'}$ and $\mathbf{K}_{e,e'}$ are defined. The product of the matrices in (3.99) is

$$\mathbf{L}_{e,e'}\mathbf{K}_{e,e'}^{-1}\mathbf{L}_{e,e'}^T = \left[\begin{array}{cc} \mathbb{E}[(s_{e'} - s_e)^2] & \mathbb{E}[s_{e'} - s_e] \\ \mathbb{E}[s_{e'} - s_e] & 1 \end{array}\right], \tag{3.100}$$

[18] Indeed if P_e were a product distribution, the linear term in (3.93) vanishes and $I(Q_e)$ is approximately a quadratic in $\boldsymbol{\epsilon}$ (as shown in [26]).

where all expectations are with respect to the distribution $P_{e,e'}$. Note that the determinant of (3.100) is $\mathbb{E}[(s_{e'}-s_e)^2]-\mathbb{E}[(s_{e'}-s_e)]^2 = \mathrm{Var}(s_{e'}-s_e)$. Hence, the (1,1) element of the inverse of (3.100) is simply

$$\psi(P_{e,e'}) = \mathrm{Var}(s_{e'}-s_e)^{-1}. \tag{3.101}$$

Now, if e and e' share a node, this proof proceeds in exactly the same way. In particular, the crucial step (3.93) will also remain the same since the Taylor expansion does not change. This concludes the first part of the proof.

Step 4: We now prove the continuity statement. The idea is that all the approximations become increasingly exact as ϵ (in the definition of the ϵ-very noisy condition) tends to zero. More concretely, for every $\delta > 0$, there exists a $\epsilon_1 > 0$ such that if $P_{e,e'}$ satisfies the ϵ_1-very noisy condition, then

$$|I(P_e) - I(P_{e'})| < \delta \tag{3.102}$$

since mutual information is continuous. For every $\delta > 0$, there exists a $\epsilon_2 > 0$ such that if $P_{e,e'}$ satisfies the ϵ_2-very noisy condition, then

$$\|Q^*_{e,e'} - P_{e,e'}\|_\infty < \delta, \tag{3.103}$$

since if $P_{e,e'}$ is ϵ_2-very noisy it is close to the constraint set $\{Q : I(Q_{e'}) \geq I(Q_e)\}$ and hence close to the optimal solution $Q^*_{e,e'}$. For every $\delta > 0$, there exists a $\epsilon_3 > 0$ such that if $P_{e,e'}$ satisfies the ϵ_3-very noisy condition, then

$$\left| D(Q^*_{e,e'}||P_{e,e'}) - \frac{1}{2}\|Q^*_{e,e'} - P_{e,e'}\|^2_{P_{e,e'}} \right| < \delta, \tag{3.104}$$

which follows from the approximation of the divergence and the continuity statement in (3.103). For every $\delta > 0$, there exists a $\epsilon_4 > 0$ such that if $P_{e,e'}$ satisfies the ϵ_4-very noisy condition, then

$$\left| I(P_e) - \mathbf{s}_e^T(Q^*_{e,e'} - P_{e,e'}) \right| < \delta, \tag{3.105}$$

which follows from retaining only the first term in the Taylor expansion of the mutual information in (3.93). Finally, for every $\delta > 0$, there exists a $\epsilon_5 > 0$ such that if $P_{e,e'}$ satisfies the ϵ_5-very noisy condition, then

$$|\widetilde{J}_{e,e'} - J_{e,e'}| < \delta, \tag{3.106}$$

which follows from continuity of the objective in the constraints (3.105). Now choose $\epsilon = \min_{i=1,\ldots,5} \epsilon_i$ to conclude that for every $\delta > 0$, there exists a $\epsilon > 0$ such that if $P_{e,e'}$ satisfies the ϵ-very noisy condition, then (3.106) holds. This completes the proof. □

■ 3.E Proof of Proposition 3.9

Proof. The following facts about P in Table 3.1 can be readily verified:

1. P is positive everywhere, *i.e.*, $P(\mathbf{x}) > 0$ for all $\mathbf{x} \in \mathcal{X}^3$.
2. P is Markov on the complete graph with $d = 3$ nodes, hence P is not a tree distribution.
3. The mutual information between x_1 and x_2 as a function of κ is given by

$$I(P_{1,2}) = \log 2 + (1 - 2\kappa)\log(1 - 2\kappa) + 2\kappa \log(2\kappa).$$

   Thus $I(P_{1,2}) \to \log 2 = 0.693$ as $\kappa \to 0$.
4. For any $(\xi, \kappa) \in (0, 1/3) \times (0, 1/2)$, $I(P_{2,3}) = I(P_{1,3})$ and this pair of mutual information quantities can be made arbitrarily small as $\kappa \to 0$.

Thus, for sufficiently small $\kappa > 0$, $I(P_{1,2}) > I(P_{2,3}) = I(P_{1,3})$. We conclude that the Chow-Liu MWST algorithm will first pick the edge $(1, 2)$ and then arbitrarily choose between the two remaining edges: $(2, 3)$ or $(1, 3)$. Thus, optimal tree structure is not unique. □

■ 3.F Proof of Theorem 3.11

We first state two preliminary lemmas and prove the first one. Theorem 3.11 will then be an immediate consequence of these lemmas.

Lemma 3.13. *Let X and Y be two metric spaces and let $\mathcal{K} \subset X$ be a compact set in X. Let $f : X \times Y \to \mathbb{R}$ be a continuous real-valued function. Then the function $g : Y \to \mathbb{R}$, defined as*

$$g(y) \;:=\; \min_{x \in \mathcal{K}} f(x, y), \quad \forall\, y \in Y, \tag{3.107}$$

is continuous on Y.

Proof. Set the minimizer in (3.107) to be

$$x(y) \;:=\; \operatorname*{argmin}_{x \in \mathcal{K}} f(x, y). \tag{3.108}$$

The optimizer $x(y) \in \mathcal{K}$ exists since $f(x, y)$ is continuous on $\mathcal{K}$ for each $y \in Y$ and $\mathcal{K}$ is compact. This follows from Weierstrauss' extreme value theorem [166, Theorem 4.16]. We want to show that for $\lim_{y' \to y} g(y') = g(y)$. In other words, we need to prove that

$$\lim_{y' \to y} f(x(y'), y') \to f(x(y), y). \tag{3.109}$$

Consider the difference,

$$|f(x(y'), x') - f(x(y), y)| \leq |f(x(y), y) - f(x(y), y')|$$

$$+ |f(x(y), y') - f(x(y'), y')|. \tag{3.110}$$

The first term in (3.110) tends to zero as $y' \to y$ by the continuity of f so it remains to show that the second term, $B_{y'} := |f(x(y), y') - f(x(y'), y')| \to 0$, as $y' \to y$. Now, we can remove the absolute value since by the optimality of $x(y')$, $f(x(y), y') \geq f(x(y'), y')$. Hence,

$$B_{y'} = f(x(y), y') - f(x(y'), y'). \tag{3.111}$$

Suppose, to the contrary, there exists a sequence $\{y'_n\}_{n=1}^{\infty} \subset Y$ with $y'_n \to y$ such that

$$f(x(y), y'_n) - f(x(y'_n), y'_n) > \epsilon > 0, \quad \forall\, n \in \mathbb{N}. \tag{3.112}$$

By the compactness of $\mathcal{K}$, for the sequence $\{x(y'_n)\}_{n=1}^{\infty} \subset \mathcal{K}$, there exists a subsequence $\{x(y'_{n_k})\}_{k=1}^{\infty} \subset \mathcal{K}$ whose limit is $x^* = \lim_{k\to\infty} x(y'_{n_k})$ and $x^* \in \mathcal{K}$ [166, Theorem 3.6(a)]. By the continuity of f

$$\lim_{k\to\infty} f(x(y), y'_{n_k}) = f(x(y), y), \tag{3.113}$$

$$\lim_{k\to\infty} f(x(y'_{n_k}), y'_{n_k}) = f(x^*, y), \tag{3.114}$$

since every subsequence of a convergent sequence $\{y'_n\}$ converges to the same limit y. Now (3.112) can be written as

$$f(x(y), y'_{n_k}) - f(x(y'_{n_k}), y'_{n_k}) > \epsilon > 0, \quad \forall\, k \in \mathbb{N}. \tag{3.115}$$

We now take the limit as $k \to \infty$ of (3.115). Next, we use (3.113) and (3.114) to conclude that

$$f(x(y), y) - f(x^*, y) > \epsilon \Rightarrow f(x(y), y) > f(x^*, y) + \epsilon, \tag{3.116}$$

which contradicts the optimality of $x(y)$ in (3.108). Thus, $B_{y'} \to 0$ as $y' \to y$ and $\lim_{y'\to y} g(y') = g(y)$, which demonstrates the continuity of g on Y. □

Lemma 3.14 (The continuous mapping theorem [134])**.** *Let $(\Omega, \mathcal{B}(\Omega), \nu)$ be a probability space. Let the sequence of random variables $\{X_n\}_{n=1}^{\infty}$ on Ω converge ν-almost surely to X, i.e., $X_n \xrightarrow{a.s.} X$. Let $g : \Omega \to \mathbb{R}$ be a continuous function. Then $g(X_n)$ converges ν-almost surely to $g(X)$, i.e., $g(X_n) \xrightarrow{a.s.} g(X)$.*

Proof. Now, using Lemmas 3.13 and 3.14, we complete the proof of Theorem 3.11. First we note from (3.58) that $\widehat{J}_{e,e'} = \widehat{J}_{e,e'}(\widehat{P}_{e,e'})$, *i.e.*, $\widehat{J}_{e,e'}$ is a function of the empirical distribution on node pairs e and e'. Next, we note that $D(Q||P_{e,e'})$ is a continuous function in $(Q, P_{e,e'})$. If $\widehat{P}_{e,e'}$ is fixed, the expression (3.58) is a minimization of $D(Q||\widehat{P}_{e,e'})$, over the compact set[19] $\Lambda = \{Q \in \mathcal{P}(\mathcal{X}^4) : I(Q_{e'}) = I(Q_e)\}$, hence Lemma 3.13 applies (with the identifications $f \equiv D$ and $\Lambda \equiv \mathcal{K}$) which implies that $\widehat{J}_{e,e'}$ is continuous in the empirical distribution $\widehat{P}_{e,e'}$. Since the empirical distribution $\widehat{P}_{e,e'}$ converges almost surely to $P_{e,e'}$ [47, Sec. 11.2], $\widehat{J}_{e,e'}(\widehat{P}_{e,e'})$ also converges almost surely to $J_{e,e'}$, by Lemma 3.14. □

[19] Compactness of Λ was proven in Theorem 3.1 cf. Eq. (3.71).

Chapter 4

Large Deviations for Learning Gaussian Tree Models

4.1 Introduction

THIS chapter focuses on the error exponent analysis for learning tree-structured Gaussian graphical models given i.i.d. samples. Many of the results from the previous chapter on learning discrete tree distributions carry over, but the compact parameterization of multivariate zero-mean Gaussians (in terms of correlation coefficients) allows us to perform further analysis. In particular, the use of the Markov property for Gaussians in Lemma 2.26 allows us to identify particular classes of tree-structured Gaussian graphical models that have large error exponents (and hence can be interpreted as easier to learn) and conversely, classes of trees that have small error exponents (high sample complexity).

We answer three fundamental questions with regard to learning Gaussian tree-structured graphical models in this chapter. (i) Can we characterize the error exponent for structure learning by the ML algorithm for tree-structured Gaussian graphical models? This is *a-priori* not immediately obvious given the results in Chapter 3 because the analysis of continuous distributions (densities with respect to the Lebesgue measure) typically present more technical difficulties as compared to their discrete counterparts. (ii) How do the *structure* and *parameters* of the model influence the error exponent? (iii) What are extremal tree distributions for learning, i.e., the distributions that maximize and minimize the error exponents?

We show that the error exponent can be derived in the same way as we did in Chapter 3 for discrete tree models albeit via slightly more intricate mathematical arguments. Furthermore, we show that due to *correlation decay*, pairs of nodes which are far apart, in terms of their graph distance, are unlikely to be mistaken as edges by the ML estimator. This is not only an intuitive result, but also results in a significant reduction in the computational complexity to find the exponent – from $O(d^{d-2})$ for exhaustive search and $O(d^3)$ for discrete tree models (in Chapter 3) to $O(d)$ for Gaussians (Proposition 4.6).

We then analyze extremal tree structures for learning, given a fixed set of correlation coefficients on the edges of the tree. Our main result is the following: The *star* graph

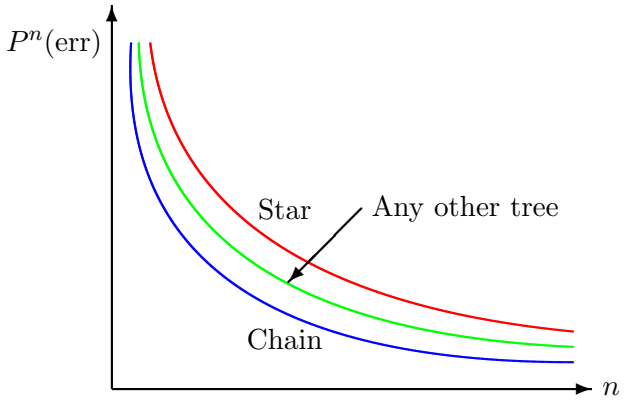

Figure 4.1. Error probability associated with the extremal structures. When n is sufficiently large, the chain minimizes the error probability and the star maximizes the error probability.

minimizes the error exponent, and if the absolute value of all the correlation coefficients of the variables along the edges is less than 0.63, then the *Markov chain* also maximizes the error exponent (Theorem 4.7). Therefore, the extremal tree structures in terms of the diameter are *also* the extremal trees for learning Gaussian tree distributions. This agrees with the intuition that the amount of correlation decay increases with the tree diameter, and that correlation decay helps the ML estimator to better distinguish the edges from the non-neighbor pairs. See Fig. 4.1 for an illustration of this result in terms of the asymptotic error probabilities for structure learning. Lastly, we analyze how changing the size of the tree influences the magnitude of the error exponent (Propositions 4.10 and 4.11).

This chapter is organized as follows: In Section 4.2, we state some additional notation that will be used in this chapter and also mention how to modify the Chow-Liu algorithm in Section 2.5.2 to learn Gaussian tree models. Sections 4.3 and 4.4 contain results on error exponents and Euclidean approximations that are analogous to those derived in Chapter 3. We mention how and why some proofs differ from their discrete counterparts. Results specific to Gaussians are presented from Section 4.5 onwards. We demonstrate in Section 4.5 how to reduce the computational complexity for calculating the exponent. In Section 4.6, we identify extremal structures that maximize and minimize the error exponent. Numerical results are presented in Section 4.7 and we conclude the discussion in Section 4.8. The proofs of all the theorems are deferred to the appendices at the end of the chapter.

■ 4.2 Problem Statement and Learning of Gaussian Tree Models

Let $\mathbf{X} = (X_1, \ldots, X_d)$ be a jointly Gaussian random vector distribution according to $p(\mathbf{x})$, a tree-structured Gaussian graphical model (see Section 2.4.4). The pdf or

distribution[1] $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Sigma})$ is Markov on a tree $T_p = (V, E_p)$. The covariance matrix is strictly positive definite, i.e., $\mathbf{\Sigma} \succ 0$. We would like to analyze the learning of the structure of p (i.e., E_p) from a set of i.i.d. samples $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ drawn from p. Each sample $\mathbf{x}_k := (x_{k,1}, \ldots, x_{k,d})^T \in \mathbb{R}^d$. We denote the set of pdfs on $\mathbb{R}^d$ by $\mathcal{P}(\mathbb{R}^d)$, the set of Gaussian pdfs on $\mathbb{R}^d$ by $\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d)$ and the set of Gaussian graphical models which factorize according to some tree in $\mathcal{T}^d$ as $\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d)$.

Here, we also mention how the Chow-Liu ML learning algorithm [42] can be adapted for estimating the structure of a Gaussian tree model p. The algorithm proceeds in very much the same way as described in Section 2.5.2 with the exception that the empirical distribution in (2.108) is now replaced by the estimate $\widehat{p}(\mathbf{x}) := \mathcal{N}(\mathbf{x}; \mathbf{0}, \widehat{\mathbf{\Sigma}})$ where

$$\widehat{\mathbf{\Sigma}} := \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k \mathbf{x}_k^T \tag{4.1}$$

is the *empirical covariance matrix*. One can then show along the lines of Section 2.5.2 that the structure learning problem reduces to the MWST problem:

$$E_{\mathrm{ML}}(\mathbf{x}^n) \;=\; \operatorname*{argmax}_{E_q : q \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d)} \sum_{e \in E_q} I(\widehat{p}_e), \tag{4.2}$$

where the edge weights are the empirical mutual information quantities given by

$$I(\widehat{p}_e) := \frac{1}{2} \log \left(\frac{1}{1 - \widehat{\rho}_e^2} \right), \tag{4.3}$$

and where the *empirical correlation coefficient* between X_i and X_j given $\mathbf{x}^n$ is given by

$$\widehat{\rho}_e = \widehat{\rho}_{i,j} := \frac{\widehat{\mathbf{\Sigma}}(i,j)}{\sqrt{\widehat{\mathbf{\Sigma}}(i,i)\widehat{\mathbf{\Sigma}}(j,j)}}. \tag{4.4}$$

Note that in (4.2), the estimated edge set $E_{\mathrm{ML}}(\mathbf{x}^n)$ is a random quantity that depends on n and, specifically, on the samples in $\mathbf{x}^n$ and we make this dependence explicit. We assume that T_p is a (connected) tree because with probability 1, the resulting optimization problem in (4.2) produces a spanning tree as all the mutual information quantities in (4.3) will be non-zero. If T_p *were* allowed to be a *proper forest*, the estimation of E_p would be inconsistent because the learned edge set will be different from the true edge set.

We now define the analysis problem formally. The definitions here are similar to those for discrete models in Section 3.2 but are included for convenience of the reader. We define the (error) event of interest

$$\mathcal{A}_n \;:=\; \{E_{\mathrm{ML}} \neq E_p\}, \tag{4.5}$$

[1] Our results also extend to the scenario where the mean of the Gaussian is unknown and has to be estimated from the samples.

where E_{ML} is the edge set of the Chow-Liu estimator in (4.2). In this chapter, we are interested to *compute* and subsequently *study* the *error exponent* K_p, or the rate at which the error probability of the event $\mathcal{A}_n$, with respect to the *true* model p, decays with the number of samples n. Similar to K_P for discrete models in (3.4), K_p for Gaussian tree models is defined as

$$K_p \;:=\; \lim_{n\to\infty} -\frac{1}{n}\log P^n(\mathcal{A}_n), \tag{4.6}$$

assuming the limit exists. We prove that the limit in (4.6) exists in the sequel. The value of K_p for different Gaussian tree-structured graphical models p provides an indication of the relative ease of estimating such models. Note that both the *parameters* and *structure* of the model influence the magnitude of K_p.

■ 4.3 Deriving the Error Exponent

This section is devoted to the derivation of K_p, defined in (4.6). The strategy is similar to that for discrete models in Sections 3.3 and 3.4 so we deliberately keep the exposition terse but highlight salient differences.

■ 4.3.1 Crossover Rates for Mutual Information Quantities

To compute K_p, we again first consider two pairs of nodes $e, e' \in \binom{V}{2}$ such that $I(p_e) > I(p_{e'})$. We now derive an LDP for the crossover event of empirical mutual information quantities $\mathcal{C}_{e,e'}$ defined in (3.6). As mentioned in Section 3.3, this is an important event for the computation of K_p because if two pairs of nodes (or node pairs) e and e' happen to crossover, this may lead to the event $\mathcal{A}_n$ occurring. Thus, for the Gaussian case we also define $J_{e,e'}$, the *crossover rate of empirical mutual information quantities*, as

$$J_{e,e'} \;:=\; \lim_{n\to\infty} -\frac{1}{n}\log P^n(\mathcal{C}_{e,e'}). \tag{4.7}$$

Note that in order to obtain a convenient characterization of $J_{e,e'}$, one cannot simply apply Sanov's theorem directly. This is because the I-projection in (2.51) has to be over all probability measures supported on $\mathbb{R}^3$ or $\mathbb{R}^4$ (an intractably large set). Thus, calculating $J_{e,e'}$ would be intractable. We also remark that, similar to discrete models, the following analysis does not depend on whether e and e' share a node. As usual, if e and e' do share a node, we say they are an *adjacent* pair of nodes. Otherwise, we say e and e' are *disjoint*. We also reserve the symbol m to denote the total number of distinct nodes in e and e'. Hence, $m = 3$ if e and e' are adjacent and $m = 4$ if e and e' are disjoint.

Theorem 4.1. (LDP for Crossover of Empirical MI) *For two node pairs $e, e' \in \binom{V}{2}$ with pdf $p_{e,e'} \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^m)$ (for $m = 3$ or $m = 4$), the crossover rate for empirical mutual information quantities is*

$$J_{e,e'} \;=\; \inf_{q\in\mathcal{P}_{\mathcal{N}}(\mathbb{R}^m)} \Big\{ D(q \,||\, p_{e,e'}) : I(q_e) = I(q_{e'}) \Big\}. \tag{4.8}$$

The crossover rate $J_{e,e'} > 0$ iff the correlation coefficients of $p_{e,e'}$ satisfy $|\rho_e| \neq |\rho_{e'}|$.

Proof. (*Sketch*) This is an application of Sanov's Theorem on arbitrary alphabets (see [64, Ch. 3] or [59, Ch. 6]), the contraction principle in large deviations theory, together with the maximum entropy principle (see Section 2.1.3). See Appendix 4.A. □

Theorem 4.1 says that in order to compute the crossover rate $J_{e,e'}$, we can restrict our attention to a problem that involves only an optimization over *Gaussian* measures, which is a finite-dimensional optimization problem. Note that the constraint in (4.8) can be written in terms of the correlation coefficients as $\rho_e^2 = \rho_{e'}^2$, where ρ_e is the correlation coefficient corresponding to the joint pdf q_e.

■ 4.3.2 Error Exponent for Structure Learning

From the discussion in Section 3.4, we see that the set of crossover rates $\{J_{e,e'}\}$ can be related to the error exponent K_p via Theorem 3.4, i.e.,

$$K_p = \min_{e' \notin E_p} \min_{e \in \mathrm{Path}(e'; E_p)} J_{e,e'}, \tag{4.9}$$

In addition, from the result in (4.9), we can derive conditions to ensure that $K_p > 0$ and hence for the error probability to decay exponentially. This result differs subtly from the corresponding one in Corollary 3.5.

Corollary 4.2. (Condition for Positive Error Exponent) *The error probability $P^n(\mathcal{A}_n)$ decays exponentially, i.e., $K_p > 0$ iff $\mathbf{\Sigma}$ has full rank and T_p is not a proper forest (as was assumed in Section 4.2).*

Proof. See Appendix 4.B for the proof. □

Note that in addition to the requirement that T_p is not a proper forest, we need the covariance matrix $\mathbf{\Sigma}$ to have full rank for the error probability to decay exponentially.

The above result provides necessary and sufficient conditions for the error exponent K_p to be positive, which implies exponential decay of the error probability in n, the number of samples. Our goal now is to analyze the influence of structure and parameters of the Gaussian pdf p on the *magnitude* of the error exponent K_p. Such an exercise requires a closed-form expression for K_p, which in turn, requires a closed-form expression for the crossover rate $J_{e,e'}$. However, the crossover rate, despite having an exact expression in (4.8), can only be found numerically, since the optimization is non-convex (due to the highly nonlinear equality constraint $I(q_e) = I(q_{e'})$). Hence, similar to Section 3.5, we provide an approximation to the crossover rate in the next section which is tight in the very noisy learning regime.

■ 4.4 Euclidean Approximations

In this section, we apply the same family of Euclidean approximation techniques to simplify the crossover rate in (4.8). We will observe that such an approximation allows

us to compare the relative ease of learning various tree structures in the subsequent sections. As in Section 3.5, we impose suitable "noisy" conditions on $p_{e,e'}$ (the joint pdf on node pairs e and e') so as to enable us to relax the non-convex optimization problem in (4.8) to a convex program.

Definition 4.1. (ϵ-Very Noisy Condition) *The joint pdf $p_{e,e'}$ on node pairs e and e' is said to satisfy the* ϵ-very noisy condition *if the correlation coefficients on e and e' satisfy $||\rho_e| - |\rho_{e'}|| < \epsilon$.*

By continuity of the mutual information in the correlation coefficient (see the function form of the mutual information in (4.3)), given any fixed ϵ and ρ_e, there exists a $\delta = \delta(\epsilon, \rho_e) > 0$ such that $|I(p_e) - I(p_{e'})| < \delta$, which means that if ϵ is small, it is difficult to distinguish which node pair e or e' has the larger mutual information given the samples $\mathbf{x}^n$. Thus, if ϵ is small, we are in the very noisy learning regime, where learning is difficult.

To perform further analysis, we require an approximation of the KL-divergence between two Gaussians. For this purpose, we recall from Verdu [206, Sec. IV-E] that we can bound the KL-divergence between two zero-mean Gaussians with covariance matrices $\mathbf{\Sigma}_{e,e'} + \mathbf{\Delta}_{e,e'}$ and $\mathbf{\Sigma}_{e,e'}$ as

$$D(\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{e,e'} + \mathbf{\Delta}_{e,e'}) \,||\, \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{e,e'})) \leq \frac{\|\mathbf{\Sigma}_{e,e'}^{-1} \mathbf{\Delta}_{e,e'}\|_F^2}{4}, \tag{4.10}$$

where $\|\mathbf{M}\|_F$ is the Frobenius norm of the matrix $\mathbf{M}$.[2] Furthermore, the inequality in (4.10) is tight when the perturbation matrix $\mathbf{\Delta}_{e,e'}$ is small. More precisely, as the ratio of the singular values $\frac{\sigma_{\max}(\mathbf{\Delta}_{e,e'})}{\sigma_{\min}(\mathbf{\Sigma}_{e,e'})}$ tends to zero, the inequality in (4.10) becomes tight. To convexify the problem, we also perform a linearization of the nonlinear constraint set in (4.8) around the unperturbed covariance matrix $\mathbf{\Sigma}_{e,e'}$. This involves taking the derivative of the mutual information with respect to the covariance matrix in the Taylor expansion. We denote this (matrix) derivative as $\nabla_{\mathbf{\Sigma}_e} I(\mathbf{\Sigma}_e)$ where $I(\mathbf{\Sigma}_e) = I(\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_e))$ is the mutual information between the two random variables of the Gaussian joint pdf $p_e = \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_e)$. We now define the *linearized constraint set* of (4.8) as the affine subspace

$$\begin{aligned} L_{\mathbf{\Delta}}(p_{e,e'}) := \{\mathbf{\Delta}_{e,e'} \in \mathbb{R}^{m \times m} : I(\mathbf{\Sigma}_e) + \langle \nabla_{\mathbf{\Sigma}_e} I(\mathbf{\Sigma}_e), \mathbf{\Delta}_e \rangle \\ = I(\mathbf{\Sigma}_{e'}) + \langle \nabla_{\mathbf{\Sigma}_{e'}} I(\mathbf{\Sigma}_{e'}), \mathbf{\Delta}_{e'} \rangle\}, \end{aligned} \tag{4.11}$$

where $\mathbf{\Delta}_e \in \mathbb{R}^{2\times 2}$ is the sub-matrix of $\mathbf{\Delta}_{e,e'} \in \mathbb{R}^{m \times m}$ ($m = 3$ or 4) that corresponds to the covariance matrix of the node pair e. We also define the *approximate crossover rate* of e and e' as the minimization of the quadratic in (4.10) over the affine subspace $L_{\mathbf{\Delta}}(p_{e,e'})$ defined in (4.11):

$$\widetilde{J}_{e,e'} := \min_{\mathbf{\Delta}_{e,e'} \in L_{\mathbf{\Delta}}(p_{e,e'})} \frac{1}{4} \|\mathbf{\Sigma}_{e,e'}^{-1} \mathbf{\Delta}_{e,e'}\|_F^2. \tag{4.12}$$

[2] Eq. (4.10) is analogous to the approximations for the discrete probability measures in (2.24) and (2.25). In contrast though, (4.10) is a bound and not an approximation.

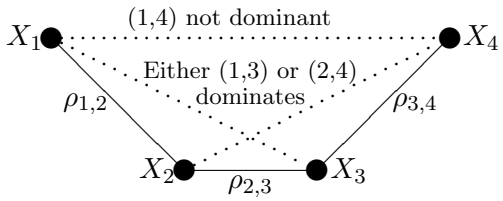

Figure 4.2. Illustration of correlation decay in a Markov chain. By Lemma 4.4(b), only the node pairs $(1,3)$ and $(2,4)$ need to be considered for computing the error exponent $\widetilde{K}_p$. By correlation decay, the node pair $(1,4)$ will not be mistaken as a true edge by the estimator because its distance, which is equal to 3, is longer than either $(1,3)$ or $(2,4)$, whose distances are equal to 2.

Eqn. (4.12) is a *convexified* version of the original optimization in (4.8).

Theorem 4.3. (Euclidean Approx. of Crossover Rate) *The approximate crossover rate for the empirical mutual information quantities, defined in* (4.12)*, is given by*

$$\widetilde{J}_{e,e'} = \frac{(\mathbb{E}[s_{e'} - s_e])^2}{2\operatorname{Var}(s_{e'} - s_e)} = \frac{(I(p_{e'}) - I(p_e))^2}{2\operatorname{Var}(s_{e'} - s_e)}. \tag{4.13}$$

In addition, the approximate error exponent corresponding to the set of crossover rates $\{\widetilde{J}_{e,e'}\}$ *is given by*

$$\widetilde{K}_p \;=\; \min_{e' \in \mathcal{E}_p} \min_{e \in \mathrm{Path}(e';\mathcal{E}_p)} \widetilde{J}_{e,e'}. \tag{4.14}$$

Proof. The proof involves solving the least squares problem in (4.12). See Appendix 4.C for the details of the calculation. □

The interpretation of (4.13) as a signal-to-noise ratio is the same as the corresponding result in Theorem 3.7. In the sequel, we limit our analysis to the very noisy regime where (4.13) and (4.14) apply.

■ 4.5 Simplification of the Error Exponent

In this section, we depart from drawing analogies with the results in Chapter 3 and develop novel results specific to Gaussian tree models. We exploit the properties of the approximate crossover rate in (4.13) to significantly reduce the complexity in finding the error exponent $\widetilde{K}_p$ to $O(d)$. As a motivating example, consider the Markov chain in Fig. 4.2. From our analysis to this point, it appears that, when computing the approximate error exponent $\widetilde{K}_p$ in (4.14), we have to consider all possible replacements between the non-edges $(1,4)$, $(1,3)$ and $(2,4)$ and the true edges along the unique paths connecting these non-edges. For example, $(1,3)$ might be mistaken as a true edge, replacing either $(1,2)$ or $(2,3)$.

We prove that, in fact, to compute $\widetilde{K}_p$ we can ignore the possibility that longest non-edge $(1,4)$ is mistaken as a true edge, thus reducing the number of computations for the approximate crossover rate $\widetilde{J}_{e,e'}$. The key to this result is the exploitation of *correlation decay*, i.e., the decrease in the absolute value of the correlation coefficient between two

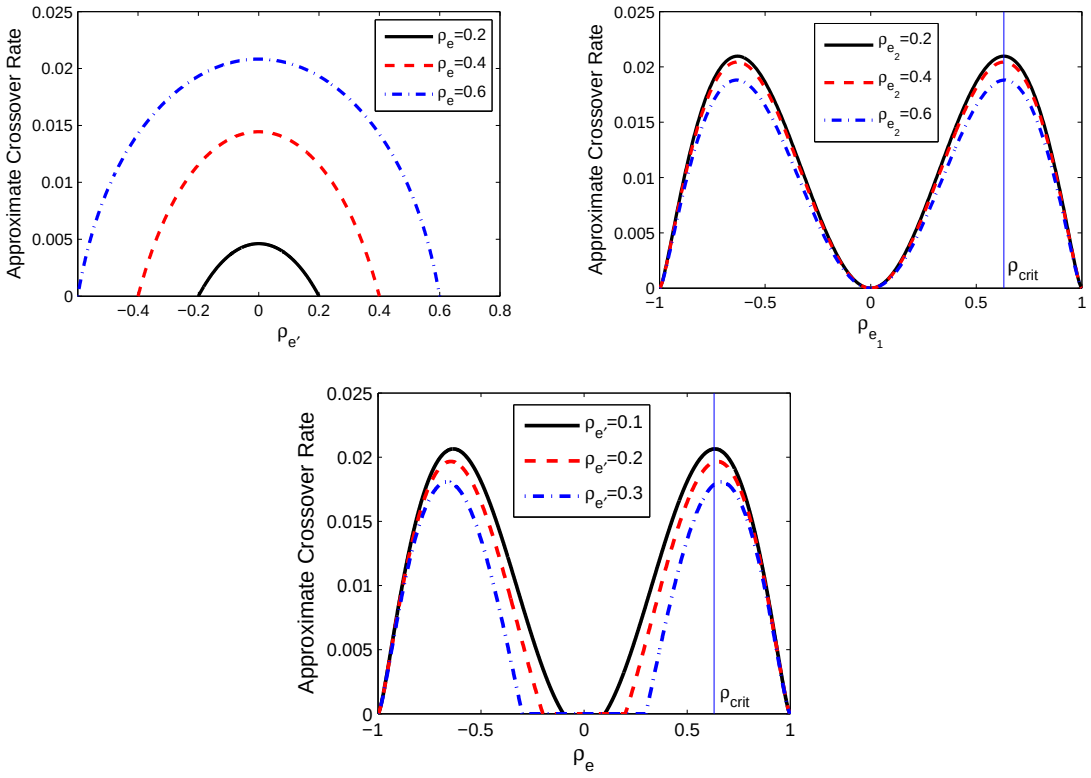

Figure 4.3. Illustration of the properties of $\widetilde{J}(\rho_e, \rho_{e'})$ in Lemma 4.4. $\widetilde{J}(\rho_e, \rho_{e'})$ is decreasing in $|\rho_{e'}|$ for fixed ρ_e (top left) and $\widetilde{J}(\rho_{e_1}, \rho_{e_1}\rho_{e_2})$ is increasing in $|\rho_{e_1}|$ for fixed ρ_{e_2} if $|\rho_{e_1}| < \rho_{\text{crit}}$ (top right). Similarly, $\widetilde{J}(\rho_e, \rho_{e'})$ is increasing in $|\rho_e|$ for fixed $\rho_{e'}$ if $|\rho_e| < \rho_{\text{crit}}$ (bottom).

nodes as the *graph distance* (the number of edges along the path between two nodes) between them increases. This follows from the Markov property (see Lemma 2.26):

$$\rho_{e'} = \prod_{e \in \text{Path}(e'; E_p)} \rho_e, \quad \forall\, e' \notin E_p. \tag{4.15}$$

For example, in Fig. 4.2, $|\rho_{1,4}| \leq \min\{|\rho_{1,3}|, |\rho_{2,4}|\}$ and because of this, the following lemma implies that $(1, 4)$ is less likely to be mistaken as a true edge than $(1, 3)$ or $(2, 4)$.

It is easy to verify that the crossover rate $\widetilde{J}_{e,e'}$ in (4.13) depends *only* on the correlation coefficients ρ_e and $\rho_{e'}$ and not the variances $\sigma_i^2 := \mathbb{E}[X_i^2]$. Thus, without loss of generality, we assume that all random variables have unit variance (which is still unknown to the learner) and to make the dependence clear, we now write $\widetilde{J}_{e,e'} = \widetilde{J}(\rho_e, \rho_{e'})$. Finally define $\rho_{\text{crit}} := 0.63055$.

Lemma 4.4. (Monotonicity of $\widetilde{J}(\rho_e, \rho_{e'})$) *$\widetilde{J}(\rho_e, \rho_{e'})$, derived in* (4.13)*, has the following properties:*

(a) $\widetilde{J}(\rho_e, \rho_{e'})$ *is an even function of both* ρ_e *and* $\rho_{e'}$.

(b) $\widetilde{J}(\rho_e, \rho_{e'})$ *is monotonically decreasing in* $|\rho_{e'}|$ *for fixed* $\rho_e \in (-1, 1)$.

(c) Assuming that $|\rho_{e_1}| < \rho_{\text{crit}}$, *then* $\widetilde{J}(\rho_{e_1}, \rho_{e_1}\rho_{e_2})$ *is monotonically increasing in* $|\rho_{e_1}|$ *for fixed* ρ_{e_2}.

(d) Assuming that $|\rho_e| < \rho_{\text{crit}}$, *then* $\widetilde{J}(\rho_e, \rho_{e'})$ *is monotonically increasing in* $|\rho_e|$ *for fixed* $\rho_{e'}$.

See Fig. 4.3 for an illustration of the properties of $\widetilde{J}(\rho_e, \rho_{e'})$.

Proof. (*Sketch*) Statement (a) follows from (4.13). We prove (b) by showing that $\partial \widetilde{J}(\rho_e, \rho_{e'})/\partial|\rho_{e'}| \leq 0$ for all $|\rho_{e'}| \leq |\rho_e|$. Statements (c) and (d) follow similarly. See Appendix 4.D for the details. □

Our intuition about correlation decay is substantiated by Lemma 4.4(b), which implies that for the example in Fig. 4.2, $\widetilde{J}(\rho_{2,3}, \rho_{1,3}) \leq \widetilde{J}(\rho_{2,3}, \rho_{1,4})$, since $|\rho_{1,4}| \leq |\rho_{1,3}|$ due to Markov property on the chain (4.15).[3] Therefore, $\widetilde{J}(\rho_{2,3}, \rho_{1,4})$ can be ignored in the minimization to find $\widetilde{K}_p$ in (4.14). Interestingly while Lemma 4.4(b) is a statement about correlation decay, Lemma 4.4(c) states that the absolute strengths of the correlation coefficients also influence the magnitude of the crossover rate.

From Lemma 4.4(b) (and the above motivating example in Fig. 4.2), finding the approximate error exponent $\widetilde{K}_p$ now reduces to finding the minimum crossover rate only over *triangles* ($(1, 2, 3)$ and $(2, 3, 4)$) in the tree as shown in Fig. 4.2, i.e., we only need to consider $\widetilde{J}(\rho_e, \rho_{e'})$ for *adjacent* edges.

Corollary 4.5 (Computation of $\widetilde{K}_p$)**.** *Under the very noisy learning regime, the approximate error exponent* $\widetilde{K}_p$ *is*

$$\widetilde{K}_p \;=\; \min_{e_i, e_j \in E_p, e_i \sim e_j} W(\rho_{e_i}, \rho_{e_j}), \tag{4.16}$$

where $e_i \sim e_j$ *means that the edges* e_i *and* e_j *are adjacent and the weights* $W(\rho_{e_1}, \rho_{e_2})$ *are defined as*

$$W(\rho_{e_1}, \rho_{e_2}) := \min\left\{\widetilde{J}(\rho_{e_1}, \rho_{e_1}\rho_{e_2}), \widetilde{J}(\rho_{e_2}, \rho_{e_1}\rho_{e_2})\right\}. \tag{4.17}$$

If the computations in (4.16) are carried out independently, the complexity is $O(d \cdot \deg_{\max})$, where $\deg_{\max}$ is the maximum degree of the nodes in the tree graph. Hence, in the worst case, the complexity is $O(d^2)$, instead of $O(d^3)$ if (4.14) is used. We can, in fact, reduce the number of computations to $O(d)$.

[3] Lemma 4.4(b) can be regarded as a "data-processing inequality" for the approximate crossover rate $\widetilde{J}(\rho_e, \rho_{e'})$.

Proposition 4.6. (Complexity in computing $\widetilde{K}_p$) *The approximate error exponent $\widetilde{K}_p$, derived in* (4.14)*, can be computed in linear time ($d-1$ operations) as*

$$\widetilde{K}_p \;=\; \min_{e\in E_p} \widetilde{J}(\rho_e, \rho_e\rho_e^*), \tag{4.18}$$

where the maximum correlation coefficient on the edges adjacent to $e \in E_p$ is defined as

$$\rho_e^* \;:=\; \max\{|\rho_{\tilde{e}}| : \tilde{e} \in E_p, \tilde{e} \sim e\}. \tag{4.19}$$

Proof. By Lemma 4.4(b) and the definition of ρ_e^*, we obtain the smallest crossover rate associated to edge e. We obtain the approximate error exponent $\widetilde{K}_p$ by minimizing over all edges $e \in E_p$ in (4.18). □

Recall that $\mathrm{diam}(T_p)$ is the diameter of T_p. The computation of K_p is reduced significantly from $O(\mathrm{diam}(T_p)d^2)$ in (3.23) and (4.9) to $O(d)$. Thus, there is a further reduction in the complexity to estimate the error exponent K_p as compared to exhaustive search which requires $O(d^{d-2})$ computations. This simplification only holds for Gaussians under the very noisy regime.

■ 4.6 Extremal Structures for Learning

In this section, we study the influence of graph structure on the approximate error exponent $\widetilde{K}_p$ using the concept of correlation decay and the properties of the crossover rate $\widetilde{J}_{e,e'}$ in Lemma 4.4. We have already discussed the connection between the error exponent and correlation decay. We also proved that non-neighbor node pairs which have shorter distances are more likely to be mistaken as edges by the ML estimator. Hence, we expect that a tree T_p which contains non-edges with shorter distances to be "harder" to learn (i.e., has a smaller error exponent $\widetilde{K}_p$) as compared to a tree which contains non-edges with longer distances. In subsequent subsections, we formalize this intuition in terms of the diameter of the tree $\mathrm{diam}(T_p)$, and show that the extremal trees, in terms of their diameter, are also extremal trees for learning. We also analyze the effect of changing the size of the tree on the error exponent.

From the Markov property in (4.15), we see that for a Gaussian tree distribution, the set of correlation coefficients fixed on the edges of the tree, along with the structure T_p, are sufficient statistics and they completely characterize p. Note that this parameterization neatly decouples the structure from the correlations. We use this fact to study the influence of changing the structure T_p while keeping the set of correlations on the edges fixed.[4] Before doing so, we state the extremal structures of trees in terms of their diameter.

[4] Although the set of correlation coefficients on the edges is fixed, the elements in this set can be arranged in different ways on the edges of the tree. We formalize this concept in (4.22).

Definition 4.2. (Extremal Trees in terms of Diameter) *Assume that* $d > 3$. *Define the* extremal trees *with* d *nodes in terms of the tree diameter* $\mathrm{diam} : \mathcal{T}^d \to \{2, \ldots, d-1\}$ *as*

$$T_{\max}(d) := \operatorname*{argmax}_{T \in \mathcal{T}^d} \mathrm{diam}(T), \tag{4.20}$$

$$T_{\min}(d) := \operatorname*{argmin}_{T \in \mathcal{T}^d} \mathrm{diam}(T). \tag{4.21}$$

Then it is clear that the two extremal structures, the chain and the star, i.e., $T_{\max}(d) = T_{\mathrm{chain}}(d)$, and $T_{\min}(d) = T_{\mathrm{star}}(d)$.

■ 4.6.1 Formulation: Extremal Structures for Learning

We now formulate the problem of finding the best and worst tree structures for learning and also the distributions associated with them. At a high level, our strategy involves two distinct steps. Firstly and primarily, we use the concept of line graphs to find the *structure* of the optimal distributions in Section 4.6.3. It turns out that the optimal structures that maximize and minimize the exponent are the Markov chain (under some conditions on the correlations) and the star respectively and these are also the extremal structures in terms of the diameter. Secondly, we optimize over the *positions* (or placement) of the correlation coefficients on the edges of the optimal structures.

Let $\boldsymbol{\rho} := (\rho_1, \rho_2, \ldots, \rho_{d-1})$ be a *fixed* vector of feasible[5] correlation coefficients, i.e., $\rho_i \in (-1, 1) \setminus \{0\}$ for all i. For a tree, it follows from (4.15) that if ρ_i's are the correlation coefficients on the edges, then $|\rho_i| < 1$ is a necessary and sufficient condition to ensure that $\boldsymbol{\Sigma} \succ 0$. Define $\boldsymbol{\Pi}_{d-1}$ to be the group of permutations of order $d-1$, hence elements in $\boldsymbol{\Pi}_{d-1}$ are permutations of a given ordered set with cardinality $d-1$. Also denote the set of tree-structured, d-variate Gaussians which have unit variances at all nodes and $\boldsymbol{\rho}$ as the correlation coefficients on the edges in some order as $\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})$. Formally,

$$\begin{aligned}\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho}) := \big\{ & p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}) \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d) : \\ & \boldsymbol{\Sigma}(i,i) = 1, \forall\, i \in V, \exists\, \boldsymbol{\pi}_p \in \boldsymbol{\Pi}_{d-1} : \boldsymbol{\sigma}_{E_p} = \boldsymbol{\pi}_p(\boldsymbol{\rho}) \big\},\end{aligned} \tag{4.22}$$

where $\boldsymbol{\sigma}_{E_p} := [\boldsymbol{\Sigma}(i,j) : (i,j) \in E_p]$ is the length-$(d-1)$ vector consisting of the covariance elements[6] on the edges (arranged in lexicographic order) and $\boldsymbol{\pi}_p(\boldsymbol{\rho})$ is the permutation of $\boldsymbol{\rho}$ according to $\boldsymbol{\pi}_p$. The tuple $(T_p, \boldsymbol{\pi}_p, \boldsymbol{\rho})$ uniquely parameterizes a Gaussian tree distribution with unit variances. Note that we can regard the permutation $\boldsymbol{\pi}_p$ as a nuisance parameter for solving the optimization for the best structure given $\boldsymbol{\rho}$. Indeed, it can happen that there are different $\boldsymbol{\pi}_p$'s such that the error exponent $\widetilde{K}_p$ is the same. For instance, in a star graph, all permutations $\boldsymbol{\pi}_p$ result in the same exponent. Despite this, we show that extremal tree *structures* are invariant to the specific choice of $\boldsymbol{\pi}_p$ and $\boldsymbol{\rho}$.

[5] We do not allow any of the correlation coefficient to be zero because otherwise, this would result in T_p being a forest.

[6] None of the elements in $\boldsymbol{\Sigma}$ are allowed to be zero because $\rho_i \neq 0$ for every $i \in V$ and the Markov property in (4.15).

For distributions in the set $\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})$, our goal is to find the best (easiest to learn) and the worst (most difficult to learn) distributions for learning. Formally, the optimization problems for the best and worst distributions for learning are given by

$$p_{\min,\boldsymbol{\rho}} := \operatorname*{argmin}_{p \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})} \widetilde{K}_p. \tag{4.23}$$

$$p_{\max,\boldsymbol{\rho}} := \operatorname*{argmax}_{p \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})} \widetilde{K}_p, \tag{4.24}$$

Thus, $p_{\max,\boldsymbol{\rho}}$ (resp. $p_{\min,\boldsymbol{\rho}}$) corresponds to the Gaussian tree model which has the largest (resp. smallest) approximate error exponent.

■ 4.6.2 Reformulation as Optimization over Line Graphs

Since the number of permutations $\boldsymbol{\pi}$ and number of spanning trees are prohibitively large, finding the optimal distributions cannot be done through a brute-force search unless d is small. Our main idea in this section is to use the notion of line graphs (See Section 2.4.1) to simplify the problems in (4.24) and (4.23). In subsequent sections, we identify the extremal tree structures before identifying the precise best and worst distributions.

Recall that the approximate error exponent $\widetilde{K}_p$ can be expressed in terms of the weights $W(\rho_{e_i}, \rho_{e_j})$ between two adjacent edges e_i, e_j as in (4.16). Therefore, we can write the extremal distribution in (4.24) as

$$p_{\max,\boldsymbol{\rho}} = \operatorname*{argmax}_{p \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})} \min_{e_i, e_j \in E_p, e_i \sim e_j} W(\rho_{e_i}, \rho_{e_j}). \tag{4.25}$$

Note that in (4.25), E_p is the edge set of a weighted graph whose edge weights are given by $\boldsymbol{\rho}$. Since the weight is between two edges, it is more convenient to consider line graphs defined in Section 2.4.1.

We now transform the intractable optimization problem in (4.25) over the set of trees to an optimization problem over all the set of line graphs:

$$p_{\max,\boldsymbol{\rho}} = \operatorname*{argmax}_{p \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})} \min_{(i,j) \in H, H = \mathcal{L}(T_p)} W(\rho_i, \rho_j), \tag{4.26}$$

and $W(\rho_i, \rho_j)$ can be considered as an edge weight between nodes i and j in a weighted line graph H. Equivalently, (4.23) can also be written as in (4.26) but with the argmax replaced by an argmin.

■ 4.6.3 Easiest and Most Difficult Structures for Learning

In order to solve (4.26), we need to characterize the set of line graphs of spanning trees $\mathcal{L}(\mathcal{T}^d) = \{\mathcal{L}(T) : T \in \mathcal{T}^d\}$. This has been studied before [95, Theorem 8.5], but the set $\mathcal{L}(\mathcal{T}^d)$ is nonetheless still very complicated. Hence, solving (4.26) directly is intractable. Instead, our strategy now is to identify the *structures* corresponding to the optimal distributions, $p_{\max,\boldsymbol{\rho}}$ and $p_{\min,\boldsymbol{\rho}}$ by exploiting the monotonicity of $\widetilde{J}(\rho_e, \rho_{e'})$ given in Lemma 4.4.

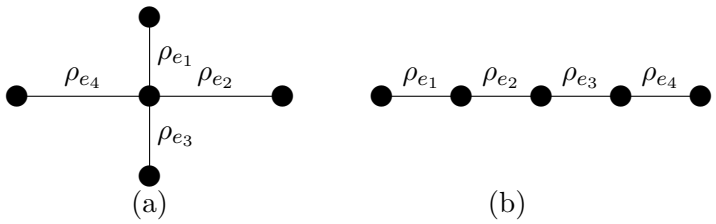

Figure 4.4. Illustration for Theorem 4.7: The star (a) and the chain (b) minimize and maximize the approximate error exponent respectively. There is more correlation decay in the chain as compared to the star because there are many more node pairs that are far apart in terms of graph distance in the chain.

Theorem 4.7. (Extremal Tree Structures) *The tree structure that minimizes the approximate error exponent* $\widetilde{K}_p$ *in* (4.23) *is given by*

$$T_{p_{\min,\boldsymbol{\rho}}} = T_{\text{star}}(d), \tag{4.27}$$

for all feasible correlation coefficient vectors $\boldsymbol{\rho}$ *with* $\rho_i \in (-1,1) \setminus \{0\}$*. In addition, if* $\rho_i \in (-\rho_{\text{crit}}, \rho_{\text{crit}}) \setminus \{0\}$ *(where* $\rho_{\text{crit}} = 0.63055$*), then the tree structure that maximizes the approximate error exponent* $\widetilde{K}_p$ *in* (4.24) *is given by*

$$T_{p_{\max,\boldsymbol{\rho}}} = T_{\text{chain}}(d). \tag{4.28}$$

Proof. See Appendix 4.E. □

See Fig. 4.4. This theorem agrees with our intuition: for the star graph, the nodes are strongly correlated (since its diameter is the smallest) while in the chain, there are many weakly correlated pairs of nodes for the same set of correlation coefficients on the edges thanks to correlation decay. Hence, it is hardest to learn the star while it is easiest to learn the chain. It is interesting to observe Theorem 4.7 implies that the extremal tree structures $T_{p_{\max,\boldsymbol{\rho}}}$ and $T_{p_{\min,\boldsymbol{\rho}}}$ are *independent of* the correlation coefficients $\boldsymbol{\rho}$ (if $|\rho_i| < \rho_{\text{crit}}$ in the case of the chain). Indeed, the experiments in Section 4.7.2 also suggest that Theorem 4.7 may likely be true for larger ranges of problems (without the constraint that $|\rho_i| < \rho_{\text{crit}}$) but this remains open. We remark that the result in Theorem 4.7 is reminiscent of the fact that graphs with high max-degree typically require more samples to learn [32, 172, 211].

The results in (4.27) and (4.28) do not yet provide the complete solution to $p_{\max,\boldsymbol{\rho}}$ and $p_{\min,\boldsymbol{\rho}}$ in (4.24) and (4.23) since there are many possible pdfs in $\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})$ corresponding to a fixed tree because we can rearrange the correlation coefficients along the edges of the tree in multiple ways. The only exception is if T_p is known to be a star then there is only one pdf in $\mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})$, and we formally state the result below.

Corollary 4.8. (Most Difficult Distribution to Learn) *The Gaussian* $p_{\min,\boldsymbol{\rho}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_{\min,\boldsymbol{\rho}})$ *defined in* (4.23)*, corresponding to the most difficult distribution to*

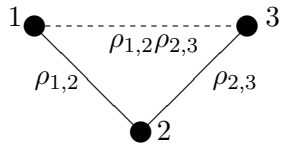

Figure 4.5. If $|\rho_{1,2}| < |\rho_{2,3}|$, then the likelihood of the non-edge $(1,3)$ replacing edge $(1,2)$ would be higher than if $|\rho_{1,2}| = |\rho_{2,3}|$. Hence, the weight $W(\rho_{1,2}, \rho_{2,3})$ is maximized when equality holds.

learn for fixed $\boldsymbol{\rho}$, has the covariance matrix whose upper triangular elements are given as $\boldsymbol{\Sigma}_{\min,\boldsymbol{\rho}}(i,j) = \rho_i$ if $i = 1, j \neq 1$ and $\boldsymbol{\Sigma}_{\min,\boldsymbol{\rho}}(i,j) = \rho_i\rho_j$ otherwise. Moreover, if $|\rho_1| \geq \ldots \geq |\rho_{d-1}|$ and $|\rho_1| < \rho_{\text{crit}} = 0.63055$, then $\widetilde{K}_p$ corresponding to the star graph can be written explicitly as a minimization over only two crossover rates:

$$\widetilde{K}_{p_{\min,\boldsymbol{\rho}}} = \min\{\widetilde{J}(\rho_1, \rho_1\rho_2), \widetilde{J}(\rho_{d-1}, \rho_{d-1}\rho_1)\}. \tag{4.29}$$

Proof. The first assertion follows directly from the Markov property (4.15) and Theorem 4.7. The next result follows from Lemma 4.4(c) which implies that $\widetilde{J}(\rho_{d-1}, \rho_{d-1}\rho_1) \leq \widetilde{J}(\rho_k, \rho_k\rho_1)$ for all $2 \leq k \leq d-1$. □

In other words, $p_{\min,\boldsymbol{\rho}}$ is a *star* Gaussian graphical model with correlation coefficients ρ_i on its edges. This result can also be explained by correlation decay. In a star graph, since the distances between non-edges are small, the estimator in (4.2) is more likely to mistake a non-edge with a true edge. It is often useful in applications to compute the minimum error exponent for a fixed vector of correlations $\boldsymbol{\rho}$ as it provides a lower bound of the decay rate of $P^n(\mathcal{A}_n)$ for any tree distribution with parameter vector $\boldsymbol{\rho}$. Interestingly, we also have a result for the easiest tree distribution to learn.

Corollary 4.9. (Easiest Distribution to Learn) *Assume that $\rho_{\text{crit}} > |\rho_1| \geq |\rho_2| \geq \ldots \geq |\rho_{d-1}|$. Then, the Gaussian $p_{\max,\boldsymbol{\rho}}(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \boldsymbol{\Sigma}_{\max,\boldsymbol{\rho}})$ defined in (4.24), corresponding to the easiest distribution to learn for fixed $\boldsymbol{\rho}$, has the covariance matrix whose upper triangular elements are*

$$\boldsymbol{\Sigma}_{\max,\boldsymbol{\rho}}(i,j) = \prod_{k=i}^{j} \rho_k, \quad \forall\, j \geq i. \tag{4.30}$$

Proof. The first assertion follows from the proof of Theorem 4.7 in Appendix 4.E and the second assertion from the Markov property in (4.15). □

In other words, in the regime where $|\rho_i| < \rho_{\text{crit}}$, $p_{\max,\boldsymbol{\rho}}$ is a *Markov chain* Gaussian graphical model with correlation coefficients arranged in a monotonic fashion on its edges. We now provide some intuition for why this is so. If a particular correlation coefficient ρ_i (such that $|\rho_i| < \rho_{\text{crit}}$) is fixed, then the edge weight $W(\rho_i, \rho_j)$, defined in (4.17), is maximized when $|\rho_j| = |\rho_i|$. Otherwise, if $|\rho_i| < |\rho_j|$, the event that the non-edge with correlation $\rho_i\rho_j$ replaces the edge with correlation ρ_i (and hence results in an error) has a higher likelihood than if equality holds. Thus, correlations ρ_i and ρ_j that are close in terms of their absolute values should be placed closer to one another

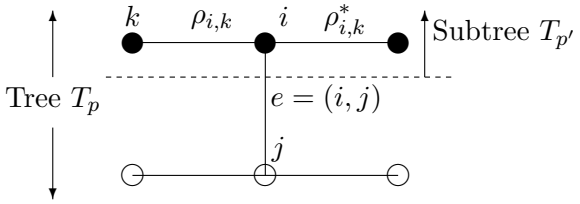

Figure 4.6. Illustration of Proposition 4.10. $T_p = (V, E_p)$ is the original tree and $e \in E_p$. $T_{p'} = (V', E_{p'})$ is a subtree. The observations for learning the structure p' correspond to the shaded nodes, the unshaded nodes correspond to unobserved variables.

(in terms of graph distance) for the approximate error exponent to be maximized. See Fig. 4.5.

■ 4.6.4 Influence of Data Dimension on Error Exponent

We now analyze the influence of *changing* the *size* of the tree on the error exponent, i.e., adding and deleting nodes and edges while satisfying the tree constraint and observing samples from the modified graphical model. This is of importance in many applications. For example, in *sequential* problems, the learner receives data at different times and would like to update the estimate of the tree structure learned. In *dimensionality reduction*, the learner is required to estimate the structure of a smaller model given high-dimensional data. Intuitively, learning only a tree with a smaller number of nodes is easier than learning the entire tree since there are fewer ways for errors to occur during the learning process. We prove this in the affirmative in Proposition 4.10.

Formally, we start with a d-variate Gaussian $p \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^d, \mathcal{T}^d; \boldsymbol{\rho})$ and consider a d'-variate pdf $p' \in \mathcal{P}_{\mathcal{N}}(\mathbb{R}^{d'}, \mathcal{T}^{d'}; \boldsymbol{\rho}')$, obtained by marginalizing p over a subset of variables and $T_{p'}$ is the tree[7] associated to the distribution p'. Hence $d' < d$ and $\boldsymbol{\rho}'$ is a subvector of $\boldsymbol{\rho}$. See Fig. 4.6. In our formulation, the only available observations are those sampled from the smaller Gaussian graphical model p'.

Proposition 4.10. (Error Exponent of Smaller Trees) *The approximate error exponent for learning p' is at least that of p, i.e., $\widetilde{K}_{p'} \geq \widetilde{K}_p$.*

Proof. Reducing the number of adjacent edges to a fixed edge $(i, k) \in E_p$ as in Fig. 4.6 (where $k \in \text{nbd}(i) \setminus \{j\}$) ensures that the maximum correlation coefficient $\rho^*_{i,k}$, defined in (4.19), does not increase. By Lemma 4.4(b) and (4.14), the approximate error exponent $\widetilde{K}_p$ does not decrease. □

Thus, lower-dimensional models are easier to learn if the set of correlation coefficients is fixed and the tree constraint remains satisfied. This is a consequence of the fact that there are fewer crossover error events that contribute to the error exponent $\widetilde{K}_p$.

[7]Note that $T_{p'}$ still needs to satisfy the tree constraint so that the variables that are marginalized out are not arbitrary (but must be variables that form the first part of a node elimination order [127]). For example, we are not allowed to marginalize out the central node of a star graph since the resulting graph would not be a tree. However, we can marginalize out any of the other nodes. In general, we can only marginalize out nodes with degree either 1 or 2.

We now consider the "dual" problem of adding a new edge to an existing tree model, which results in a larger tree. We are now provided with $(d+1)$-dimensional observations to learn the larger tree. More precisely, given a d-variate tree Gaussian pdf p, we consider a $(d+1)$-variate pdf p'' such that T_p is a subtree of $T_{p''}$. Equivalently, let $\boldsymbol{\rho} := [\rho_{e_1}, \rho_{e_2}, \ldots, \rho_{e_{d-1}}]$ be the vector of correlation coefficients on the edges of the graph of p and let $\boldsymbol{\rho}'' := [\boldsymbol{\rho}, \rho_{\text{new}}]$ be that of p''.

By comparing the error exponents $\widetilde{K}_p$ and $\widetilde{K}_{p''}$, we can address the following question: Given a new edge correlation coefficient ρ_{new}, how should one adjoin this new edge to the existing tree such that the resulting error exponent is maximized or minimized? Evidently, from Proposition 4.10, it is not possible to increase the error exponent by growing the tree but can we devise a strategy to place this new edge judiciously (resp. adversarially) so that the error exponent deteriorates as little (resp. as much) as possible?

To do so, we say edge e *contains* node v if $e = (v, i)$ and we define the nodes in the smaller tree T_p

$$v^*_{\min} := \operatorname*{argmin}_{v \in V} \max_{e \in E_p} \{|\rho_e| : e \text{ contains node } v\}. \tag{4.31}$$

$$v^*_{\max} := \operatorname*{argmax}_{v \in V} \max_{e \in E_p} \{|\rho_e| : e \text{ contains node } v\}. \tag{4.32}$$

Proposition 4.11. (Error Exponent of Larger Trees) *Assume that* $|\rho_{\text{new}}| < |\rho_e|$ *for all* $e \in E_p$. *Then,*

(a) The difference between the error exponents $\widetilde{K}_p - \widetilde{K}_{p''}$ *is minimized when* $T_{p''}$ *is obtained by adding to* T_p *a new edge with correlation coefficient* ρ_{new} *at vertex* $v^*_{\min}$ *given by* (4.31) *as a leaf.*

(b) The difference $\widetilde{K}_p - \widetilde{K}_{p''}$ *is maximized when the new edge is added to* $v^*_{\max}$ *given by* (4.32) *as a leaf.*

Proof. The node given by (4.31) is the best node to attach the new edge by Lemma 4.4(b). Statement (b) follows analogously. □

This result implies that if we receive data dimensions sequentially, we have a straightforward rule in (4.31) for identifying larger trees such that the exponent decreases as little as possible at each step.[8]

■ 4.7 Numerical Experiments

We now perform experiments with the following two objectives. Firstly, we study the accuracy of the Euclidean approximations (Theorem 4.3) to identify regimes in which the approximate crossover rate $\widetilde{J}_{e,e'}$ is close to the true crossover rate $J_{e,e'}$. Secondly,

[8] Of course, in reality, the edge might not be put according to (4.31) so we might have a smaller error exponent.

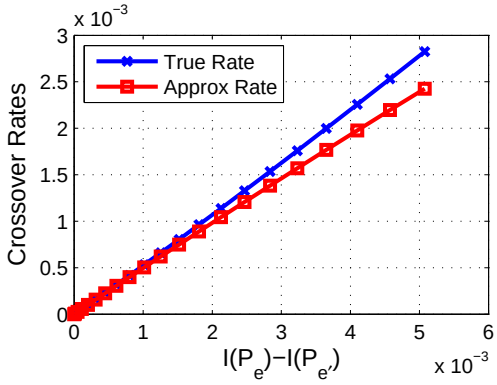

Figure 4.7. Comparison of true and approximate crossover rates in (4.8) and (4.13) respectively.

by performing simulations we study how various tree structures (e.g., chains and stars) influence the error exponents (Theorem 4.7).

■ 4.7.1 Comparison Between True and Approximate Rates

In Fig. 4.7, we plot the *true* and *approximate* crossover rates[9] (given in (4.8) and (4.12) respectively) for a 4-node symmetric star graph, whose structure is shown in Fig. 4.8. The zero-mean Gaussian graphical model has a covariance matrix $\mathbf{\Sigma}$ such that $\mathbf{\Sigma}^{-1}$ is parameterized by $\gamma \in (0, 1/\sqrt{3})$ in the following way: $\mathbf{\Sigma}^{-1}(i,i) = 1$ for all i, $\mathbf{\Sigma}^{-1}(1,j) = \mathbf{\Sigma}^{-1}(j,1) = \gamma$ for all $j = 2,3,4$ and $\mathbf{\Sigma}^{-1}(i,j) = 0$ otherwise. By increasing γ, we increase the difference of the mutual information quantities on the edges e and non-edges e'. We see from Fig. 4.7 that both rates increase as the difference $I(p_e) - I(p_{e'})$ increases. This is in line with our intuition because if $p_{e,e'}$ is such that $I(p_e) - I(p_{e'})$ is large, the crossover rate is also large. We also observe that if $I(p_e) - I(p_{e'})$ is small, the true and approximate rates are close. This is also in line with the assumptions of Theorem 4.3. When the difference between the mutual information quantities increases, the true and approximate rates separate from each other. Note, however, that the approximate rate is neither a lower nor upper bound on the true rate because we linearize the constraint set in (4.11).

■ 4.7.2 Comparison of Error Exponents Between Trees

In Fig. 4.10, we simulate error probabilities by drawing i.i.d. samples from three $d = 10$ node tree graphs – a chain, a star and a hybrid between a chain and a star as shown in Fig. 4.9. We then used the samples to learn the structure via the Chow-Liu procedure [42] by solving the MWST problem in (4.2). The $d - 1 = 9$ correlation coefficients were chosen to be equally spaced in the interval $[0.1, 0.9]$ and they were

[9]This small example has sufficient illustrative power because as we have seen, errors occur locally and only involve triangles.

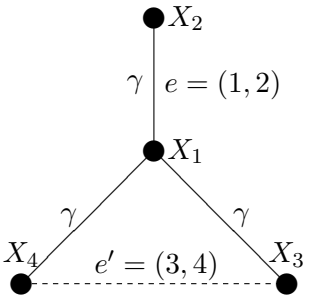

Figure 4.8. The symmetric star graphical model used for comparing the true and approximate crossover rates as described in Section 4.7.1.

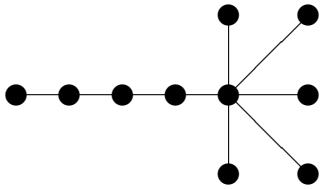

Figure 4.9. The structure of a *hybrid* tree graph with $d = 10$ nodes as described in Section 4.7.2. This is a tree with a length-$d/2$ chain and a order $d/2$ star attached to one of the leaf nodes of the chain.

randomly placed on the edges of the three tree graphs. We observe from Fig. 4.10 that for fixed n, the star and chain have the highest and lowest error probabilities $P^n(\mathcal{A}_n)$ respectively. The *simulated error exponents* given by $\{-n^{-1} \log P^n(\mathcal{A}_n)\}_{n \in \mathbb{N}}$ also converge to their true values as $n \to \infty$. The exponent associated to the star is higher than that of the chain, which is corroborated by Theorem 4.7, even though the theorem only applies in the very-noisy case (and for $|\rho_i| < 0.63055$ in the case of the chain). From this experiment, the claim also seems to be true even though the setup is not very noisy. We also observe that the error exponent of the hybrid is between that of the star and the chain.

■ 4.8 Chapter Summary

Using the theory of large deviations, we have obtained the error exponent associated with learning the structure of a Gaussian tree model. Our analysis in this chapter also answers the fundamental questions as to which set of parameters and which structures result in high and low error exponents. We conclude that Markov chains (resp. stars) are the easiest (resp. hardest) structures to learn as they maximize (resp. minimize) the error exponent. Indeed, our numerical experiments on a variety of Gaussian graphical models validate the theory presented.

The results in Chapters 3 and 4, especially those in Section 4.6.4, lead directly

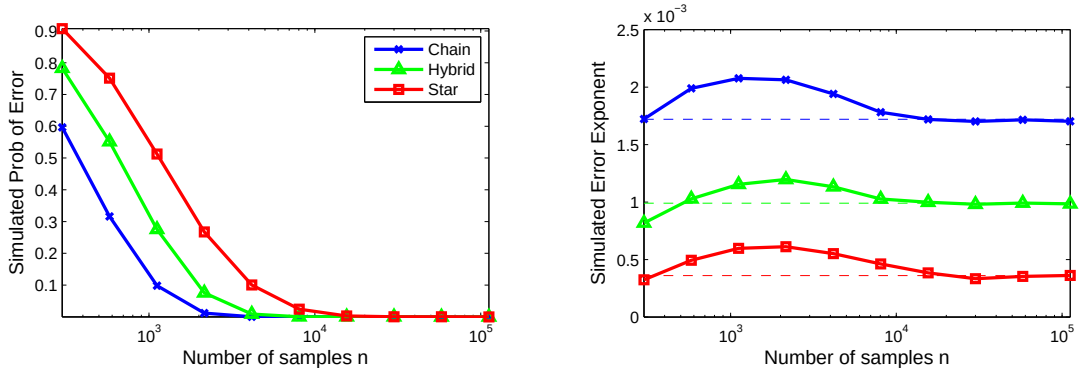

Figure 4.10. Simulated error probabilities and error exponents for chain, hybrid and star graphs with fixed $\boldsymbol{\rho}$. The dashed lines show the true error exponent K_p computed numerically using (4.8) and (4.9). Observe that the simulated error exponent converges to the true error exponent as $n \to \infty$. The legend applies to both plots.

to the natural question of how we can analyze the situation where d grows with n, i.e., the high-dimensional scenario. The analysis techniques in Chapters 3 and 4 lend useful insights for modeling data whose dimensions are much larger than the number of samples using tree-structured (and forest-structured) distributions. We explore the high-dimensional learning regime in detail in the next chapter.

Appendices for Chapter 4

■ 4.A Proof of Theorem 4.1

Proof. This proof borrows ideas from [179]. We assume $m = 4$ (i.e., disjoint edges) for simplicity. The result for $m = 3$ follows similarly. Let $V' \subset V$ be a set of $m = 4$ nodes corresponding to node pairs e and e'. Given a subset of node pairs $\mathcal{Y} \subset V' \times V'$ such that $(i, i) \in \mathcal{Y}$ for all $i \in V'$, the set of *feasible moments* [209] is defined as

$$\begin{aligned} \mathcal{M}_{\mathcal{Y}} \;:=\; \big\{ \boldsymbol{\eta}_{e,e'} \in \mathbb{R}^{|\mathcal{Y}|} \;:\; & \exists\, q(\cdot) \in \mathcal{P}(\mathbb{R}^m) \\ & \text{s.t.} \;\; \mathbb{E}_q[X_i X_j] = \eta_{i,j}, \forall\, (i,j) \in \mathcal{Y} \big\}. \end{aligned} \tag{4.33}$$

Let the set of densities with moments $\boldsymbol{\eta}_{e,e'} := \{\eta_{i,j} : (i,j) \in \mathcal{Y}\}$ be denoted as

$$\mathcal{B}_{\mathcal{Y}}(\boldsymbol{\eta}_{e,e'}) := \{q \in \mathcal{P}(\mathbb{R}^m) : \mathbb{E}_q[X_i X_j] = \eta_{i,j}, (i,j) \in \mathcal{Y}\}. \tag{4.34}$$

We now state Sanov's theorem [64] for continuous-valued distributions and also the functional form of the optimizing distribution (I-projection). A similar result was proven by Shen [179] so the proof of the lemma will be omitted.

Lemma 4.12. (Sanov's Theorem and the Contraction Principle [64, 179]) *For the event that the empirical moments of the i.i.d. observations* $\mathbf{x}^n$ *are equal to* $\boldsymbol{\eta}_{e,e'} = \{\eta_{i,j} :$

$(i,j) \in \mathcal{Y}\}$, *we have the LDP*

$$\lim_{n\to\infty} -\frac{1}{n}\log P^n\left[\bigcap_{(i,j)\in\mathcal{Y}}\left\{\frac{1}{n}\sum_{k=1}^{n} X_{k,i}X_{k,j} = \eta_{i,j}\right\}\right] = \inf_{q_{e,e'}\in\mathcal{B}_{\mathcal{Y}}(\boldsymbol{\eta})} D(q_{e,e'}\,||\,p_{e,e'}). \tag{4.35}$$

If $\boldsymbol{\eta}_{e,e'} \in \mathcal{M}_{\mathcal{Y}}$, *the optimizing pdf* $q^*_{e,e'}$ *in* (4.35) *is given by*

$$q^*_{e,e'}(\mathbf{x}) \propto p_{e,e'}(\mathbf{x}) \exp\left[\sum_{(i,j)\in\mathcal{Y}} \theta_{i,j}\, x_i x_j\right], \tag{4.36}$$

where the set of constants $\{\theta_{i,j} : (i,j) \in \mathcal{Y}\}$ *are chosen such that* $q^*_{e,e'} \in \mathcal{B}_{\mathcal{Y}}(\boldsymbol{\eta}_{e,e'})$ *given in* (4.34).

The second assertion in (4.36) is a generalization of the maximum entropy principle (see Lemma 2.4 and Chapter 12 in [47]).

From Lemma 4.12, we conclude that the optimal $q^*_{e,e'}$ in (4.35) is a Gaussian. Thus, we can restrict our search for the optimal distribution to a search over Gaussians, which are parameterized by means and covariances. The crossover event for mutual information is $\mathcal{C}_{e,e'} = \left\{\widehat{\rho}^2_{e'} \geq \widehat{\rho}^2_{e}\right\}$, since in the Gaussian case, the mutual information is a monotonic function of the square of the correlation coefficient (see Eqn. (4.3)). Thus it suffices to consider $\left\{\widehat{\rho}^2_{e'} \geq \widehat{\rho}^2_{e}\right\}$, instead of the event involving the mutual information quantities. Let $e = (i,j)$, $e' = (k,l)$ and $\boldsymbol{\eta}_{e,e'} := (\eta_e, \eta_{e'}, \eta_i, \eta_j, \eta_k, \eta_l) \in \mathcal{M}_{\mathcal{Y}} \subset \mathbb{R}^6$ be the moments of $p_{e,e'}$, where $\eta_e := \mathbb{E}[X_i X_j]$ is the covariance of X_i and X_j, and $\eta_i := \mathbb{E}[X_i^2]$ is the variance of X_i (and similarly for the other moments). Now apply the contraction principle [62, Ch. 3] to the continuous map $h : \mathcal{M}_{\mathcal{Y}} \to \mathbb{R}$, given by the difference between the square of correlation coefficients

$$h(\boldsymbol{\eta}_{e,e'}) := \frac{\eta_e^2}{\eta_i\eta_j} - \frac{\eta_{e'}^2}{\eta_k\eta_l}. \tag{4.37}$$

Following the same argument as in Theorem 3.1, the equality case dominates $\mathcal{C}_{e,e'}$, i.e., the event $\left\{\widehat{\rho}^2_{e'} = \widehat{\rho}^2_{e}\right\}$ dominates $\left\{\widehat{\rho}^2_{e'} \geq \widehat{\rho}^2_{e}\right\}$.[10] Thus, by considering the set $\{\boldsymbol{\eta}_{e,e'} : h(\boldsymbol{\eta}_{e,e'}) = 0\}$, the rate corresponding to $\mathcal{C}_{e,e'}$ can be written as

$$J_{e,e'} = \inf_{\boldsymbol{\eta}_{e,e'}\in\mathcal{M}_{\mathcal{Y}}} \left\{ g(\boldsymbol{\eta}_{e,e'}) : \frac{\eta_e^2}{\eta_i\eta_j} = \frac{\eta_{e'}^2}{\eta_k\eta_l} \right\}, \tag{4.38}$$

where the function $g : \mathcal{M}_{\mathcal{Y}} \subset \mathbb{R}^6 \to [0,\infty)$ is defined as

$$g(\boldsymbol{\eta}_{e,e'}) := \inf_{q_{e,e'}\in\mathcal{B}_{\mathcal{Y}}(\boldsymbol{\eta}_{e,e'})} D(q_{e,e'}\,||\,p_{e,e'}), \tag{4.39}$$

[10] This is also intuitively true because the most likely way the error event $\mathcal{C}_{e,e'}$ occurs is when equality holds, i.e., $\left\{\widehat{\rho}^2_{e'} = \widehat{\rho}^2_{e}\right\}$.

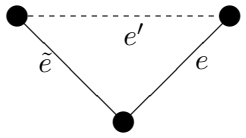

Figure 4.11. Illustration for the proof of Corollary 4.2. The correlation coefficient on the non-edge is $\rho_{e'}$ and satisfies $|\rho_{e'}| = |\rho_e|$ if $|\rho_{\tilde{e}}| = 1$.

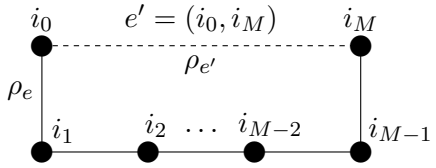

Figure 4.12. Illustration for the proof of Corollary 4.2. The unique path between i_0 and i_M is $(i_0, i_1, \ldots, i_M) = \mathrm{Path}(e'; E_p)$.

and the set $\mathcal{B}_{\mathcal{Y}}(\boldsymbol{\eta}_{e,e'})$ is defined in (4.34). Combining expressions in (4.38) and (4.39) and the fact that the optimal solution $q^*_{e,e'}$ is Gaussian yields $J_{e,e'}$ as given in the statement of the theorem (cf. Eqn. (4.8)).

The second assertion in the theorem follows from the fact that since $p_{e,e}$ satisfies $I(p_e) \neq I(p_{e'})$, we have $|\rho_e| \neq |\rho_{e'}|$ since $I(p_e)$ is a monotonic function in $|\rho_e|$. Therefore, $q^*_{e,e'} \neq p_{e,e'}$ on a set whose (Lebesgue) measure ν is strictly positive. Since $D(q^*_{e,e'}||p_{e,e'}) = 0$ if and only if $q^*_{e,e'} = p_{e,e'}$ almost everywhere-$[\nu]$, this implies that $D(q^*_{e,e'}||p_{e,e'}) > 0$ [47, Theorem 8.6.1]. □

■ 4.B Proof of Corollary 4.2

Proof. ($\Rightarrow$) Assume that $K_p > 0$. Suppose, to the contrary, that either (i) T_p is a forest or (ii) $\mathrm{rank}(\boldsymbol{\Sigma}) < d$ and T_p is not a forest. In (i), structure estimation of p will be inconsistent (as described in Section 4.2), which implies that $K_p = 0$, a contradiction. In (ii), since p is a spanning tree, there exists an edge $\tilde{e} \in E_p$ such that the correlation coefficient $\rho_{\tilde{e}} = \pm 1$ (otherwise $\boldsymbol{\Sigma}$ would be full rank). In this case, referring to Fig. 4.11 and assuming that $|\rho_e| \in (0, 1)$, the correlation on the non-edge e' satisfies $|\rho_{e'}| = |\rho_e||\rho_{\tilde{e}}| = |\rho_e|$, which implies that $I(p_e) = I(p_{e'})$. Thus, there is no unique maximizer in (4.2) with the empiricals $\widehat{p}_e$ replaced by p_e. As a result, ML for structure learning via (4.2) is inconsistent hence $K_p = 0$, a contradiction.

($\Leftarrow$) Suppose both $\boldsymbol{\Sigma} \succ 0$ and T_p *not* a proper forest, i.e., T_p is a spanning tree. Assume, to the contrary, that $K_p = 0$. Then from Chapter 3, $I(p_e) = I(p_{e'})$ for some $e' \notin E_p$ and some $e \in \mathrm{Path}(e'; E_p)$. This implies that $|\rho_e| = |\rho_{e'}|$. Let $e' = (i_0, i_M)$ be a non-edge and let the unique path from node i_0 to node i_M be $(i_0, i_1, \ldots, i_M)$ for some $M \geq 2$. See Fig. 4.12. Then, $|\rho_{e'}| = |\rho_{i_0,i_M}| = |\rho_{i_0,i_1}||\rho_{i_1,i_2}| \ldots |\rho_{i_{M-1},i_M}|$. Suppose, without loss of generality, that edge $e = (i_0, i_1)$ is such that $|\rho_{e'}| = |\rho_e|$ holds, then we can cancel $|\rho_{e'}|$ and $|\rho_{i_0,i_1}|$ on both sides to give $|\rho_{i_1,i_2}||\rho_{i_2,i_3}| \ldots |\rho_{i_{M-1},i_M}| = 1$. Cancelling $\rho_{e'}$ is legitimate because we assumed that $\rho_{e'} \neq 0$ for all $e' \in V \times V$, because

the graph of p is a *spanning* (connected) tree. Since each correlation coefficient has magnitude not exceeding 1, this means that each correlation coefficient has magnitude 1, i.e., $|\rho_{i_1,i_2}| = \ldots = |\rho_{i_{M-1},i_M}| = 1$. Since the correlation coefficients equal to ± 1, the submatrix of the covariance matrix $\mathbf{\Sigma}$ containing these correlation coefficients is not positive definite. Therefore by Sylvester's condition [101, Theorem 7.2.5], the covariance matrix $\mathbf{\Sigma} \not\succ 0$, a contradiction. Hence, $K_p > 0$. □

■ 4.C Proof of Theorem 4.3

Proof. We first assume that e and e' do not share a node. The approximation of the KL-divergence for Gaussians can be written as in (4.10). We now linearize the constraint set $L_{\mathbf{\Delta}}(p_{e,e'})$ as defined in (4.11). Given a positive definite covariance matrix $\mathbf{\Sigma}_e \in \mathbb{R}^{2\times 2}$, to simplify the notation, let $I(\mathbf{\Sigma}_e) = I(\mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Sigma}_e))$ be the mutual information of the two random variables with covariance matrix $\mathbf{\Sigma}_e$. We now perform a first-order Taylor expansion of the mutual information around $\mathbf{\Sigma}_e$. This can be expressed as

$$I(\mathbf{\Sigma}_e + \mathbf{\Delta}_e) = I(\mathbf{\Sigma}_e) + \operatorname{Tr}\left(\nabla_{\mathbf{\Sigma}_e} I(\mathbf{\Sigma}_e)^T \mathbf{\Delta}_e\right) + o(\|\mathbf{\Delta}_e\|). \tag{4.40}$$

Recall that the Taylor expansion of log-det [74] is

$$\log\det(\mathbf{A}) = \log\det(\mathbf{B}) + \langle \mathbf{A} - \mathbf{B}, \mathbf{B}^{-1} \rangle + o(\|\mathbf{A} - \mathbf{B}\|_F), \tag{4.41}$$

with the notation $\langle \mathbf{A} - \mathbf{B}, \mathbf{B}^{-1} \rangle = \operatorname{Tr}((\mathbf{A} - \mathbf{B})\mathbf{B}^{-1})$. Using this result we can conclude that the gradient of I with respect to $\mathbf{\Sigma}_e$ in the above expansion (4.40) can be simplified to give the matrix

$$\nabla_{\mathbf{\Sigma}_e} I(\mathbf{\Sigma}_e) = -\frac{1}{2} \begin{pmatrix} 0 & [\mathbf{\Sigma}_e^{-1}]_{od} \\ [\mathbf{\Sigma}_e^{-1}]_{od} & 0 \end{pmatrix}, \tag{4.42}$$

where $[\mathbf{A}]_{od}$ is the (unique) off-diagonal element of the 2×2 symmetric matrix $\mathbf{A}$. By applying the same expansion to $I(\mathbf{\Sigma}_{e'} + \mathbf{\Delta}_{e'})$, we can express the linearized constraint as

$$\langle \mathbf{M}, \mathbf{\Delta} \rangle = \operatorname{Tr}(\mathbf{M}^T \mathbf{\Delta}) = I(\mathbf{\Sigma}_e) - I(\mathbf{\Sigma}_{e'}), \tag{4.43}$$

where the symmetric matrix $\mathbf{M} = \mathbf{M}(\mathbf{\Sigma}_{e,e'})$ is defined in the following fashion: $\mathbf{M}(i,j) = \frac{1}{2}[\mathbf{\Sigma}_e^{-1}]_{od}$ if $(i,j) = e$, $\mathbf{M}(i,j) = -\frac{1}{2}[\mathbf{\Sigma}_{e'}^{-1}]_{od}$ if $(i,j) = e'$ and $\mathbf{M}(i,j) = 0$ otherwise.

Thus, the problem reduces to minimizing (over $\mathbf{\Delta}$) the approximate objective in (4.10) subject to the linearized constraints in (4.43). This is a least-squares problem. By using the matrix derivative identities

$$\nabla_{\mathbf{\Delta}} \operatorname{Tr}(\mathbf{M}\mathbf{\Delta}) = \mathbf{M}, \qquad \nabla_{\mathbf{\Delta}} \operatorname{Tr}((\mathbf{\Sigma}^{-1}\mathbf{\Delta})^2) = 2\mathbf{\Sigma}^{-1}\mathbf{\Delta}\mathbf{\Sigma}^{-1}, \tag{4.44}$$

we can solve for the optimizer $\mathbf{\Delta}^*$ yielding:

$$\mathbf{\Delta}^* = \frac{I(\mathbf{\Sigma}_e) - I(\mathbf{\Sigma}_{e'})}{(\operatorname{Tr}(\mathbf{M}\mathbf{\Sigma}))^2} \mathbf{\Sigma}\mathbf{M}\mathbf{\Sigma}. \tag{4.45}$$

Substituting the expression for $\mathbf{\Delta}^*$ into (4.10) yields

$$\widetilde{J}_{e,e'} = \frac{(I(\mathbf{\Sigma}_e) - I(\mathbf{\Sigma}_{e'}))^2}{4\,\mathrm{Tr}((\mathbf{M\Sigma})^2)} = \frac{(I(p_e) - I(p_{e'}))^2}{4\,\mathrm{Tr}((\mathbf{M\Sigma})^2)}. \tag{4.46}$$

Comparing (4.46) to our desired result (4.13), we observe that problem now reduces to showing that $\mathrm{Tr}((\mathbf{M\Sigma})^2) = \frac{1}{2}\mathrm{Var}(s_e - s_{e'})$. To this end, we note that for Gaussians, the information density is

$$s_e(X_i, X_j) = -\frac{1}{2}\log(1-\rho_e^2) - [\mathbf{\Sigma}_e^{-1}]_{od}\, X_i\, X_j. \tag{4.47}$$

Since the first term is a constant, it suffices to compute $\mathrm{Var}([\mathbf{\Sigma}_e^{-1}]_{od}X_iX_j - [\mathbf{\Sigma}_{e'}^{-1}]_{od}\, X_k\, X_l)$. Now, we define the matrices

$$\mathbf{C} := \begin{pmatrix} 0 & 1/2 \\ 1/2 & 0 \end{pmatrix},\ \mathbf{C}_1 := \begin{pmatrix} \mathbf{C} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix},\ \mathbf{C}_2 := \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{C} \end{pmatrix}, \tag{4.48}$$

and use the following identity for the normal random vector $(X_i, X_j, X_k, X_l) \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$

$$\mathrm{Cov}(aX_iX_j, bX_kX_l) = 2ab \cdot \mathrm{Tr}(\mathbf{C}_1\mathbf{\Sigma}\mathbf{C}_2\mathbf{\Sigma}), \quad \forall\, a, b \in \mathbb{R}, \tag{4.49}$$

and the definition of $\mathbf{M}$ to conclude that

$$\mathrm{Var}(s_e - s_{e'}) = 2\mathrm{Tr}((\mathbf{M\Sigma})^2) \tag{4.50}$$

This completes the proof for the case when e and e' do not share a node. The proof for the case when e and e' share a node proceeds along exactly the same lines with a slight modification of the matrix $\mathbf{M}$. □

■ 4.D Proof of Lemma 4.4

Proof. Denoting the correlation coefficient on edge e and non-edge e' as ρ_e and $\rho_{e'}$ respectively, the approximate crossover rate can be expressed as

$$\widetilde{J}(\rho_e, \rho_{e'}) = \frac{A(\rho_e^2, \rho_{e'}^2)}{B(\rho_e^2, \rho_{e'}^2)}, \tag{4.51}$$

where the numerator and the denominator are defined as

$$A(\rho_e^2, \rho_{e'}^2) := \left[\frac{1}{2}\log\left(\frac{1-\rho_{e'}^2}{1-\rho_e^2}\right)\right]^2, \tag{4.52}$$

$$B(\rho_e^2, \rho_{e'}^2) := \frac{2(\rho_{e'}^4 + \rho_{e'}^2)}{(1-\rho_{e'}^2)^2} + \frac{2(\rho_e^4 + \rho_e^2)}{(1-\rho_e^2)^2} - \frac{4\rho_{e'}^2(\rho_e^2+1)}{(1-\rho_{e'}^2)(1-\rho_e^2)}. \tag{4.53}$$

The evenness result follows from A and B because $\widetilde{J}(\rho_e, \rho_{e'})$ is, in fact a function of $(\rho_e^2, \rho_{e'}^2)$. To simplify the notation, we make the following substitutions: $x := \rho_{e'}^2$ and

$y := \rho_e^2$. Now we apply the quotient rule to (4.51). Defining $\mathcal{R} := \{(x,y) \in \mathbb{R}^2 : y \in (0,1), x \in (0,y)\}$, it suffices to show that

$$C(x,y) := B(x,y)\frac{\partial A(x,y)}{\partial x} - A(x,y)\frac{\partial B(x,y)}{\partial x} \leq 0, \tag{4.54}$$

for all $(x,y) \in \mathcal{R}$. Upon simplification, we have

$$C(x,y) = \frac{\log\left(\frac{1-x}{1-y}\right)\left[\log\left(\frac{1-x}{1-y}\right)C_1(x,y) + C_2(x,y)\right]}{2(1-y)^2(1-x)^3}, \tag{4.55}$$

where

$$C_1(x,y) := y^2x - 6xy - 1 - 2y + 3y^2 \tag{4.56}$$

and

$$C_2(x,y) := 2x^2y - 6x^2 + 2x - 2y^2x + 8xy - 2y - 2y^2. \tag{4.57}$$

Since $x < y$, the logs in $C(x,y)$ are positive, i.e., $\log\left(\frac{1-x}{1-y}\right) > 0$, so it suffices to show that

$$\log\left(\frac{1-x}{1-y}\right)C_1(x,y) + C_2(x,y) \leq 0. \tag{4.58}$$

for all $(x,y) \in \mathcal{R}$. By using the inequality $\log(1+t) \leq t$ for all $t > -1$, it again suffices to show that

$$C_3(x,y) := (y-x)C_1(x,y) + (1-y)C_2(x,y) \leq 0. \tag{4.59}$$

Now upon simplification, $C_3(x,y) = 3y^3x - 19y^2x - 3y - 2y^2 + 5y^3 - 3y^2x^2 + 14x^2y + 3x + 8xy - 6x^2$, and this polynomial is equal to zero in $\overline{\mathcal{R}}$ (the closure of $\mathcal{R}$) iff $x = y$. At all other points in $\mathcal{R}$, $C_3(x,y) < 0$. Thus, the derivative of $\widetilde{J}(\rho_e, \rho_{e'})$ with respect to $\rho_{e'}$ is indeed strictly negative on $\mathcal{R}$. Keeping ρ_e fixed, the function $\widetilde{J}(\rho_e, \rho_{e'})$ is monotonically decreasing in $\rho_{e'}^2$ and hence $|\rho_{e'}|$.

Statement (c) follows by substituting $\rho_{e'}^2 = xy$ and $\rho_e^2 = x$ in (4.51) to get $\widetilde{J}(x, xy)$ for $x, y \in (0, \rho_{\text{crit}}^2)$. Fixing y, we can again differentiate the function $g_y(x) := \widetilde{J}(x, xy)$ wrt x. We then note that $\frac{d}{dx}g_y(x)$ is positive iff $0 < x < \rho_{\text{crit}}^2$ as shown in Fig. 4.13. Statement (d) follows along the same lines. □

■ 4.E Proofs of Theorem 4.7 and Corollary 4.9

Proof. Proof of $T_{p_{\min}(\boldsymbol{\rho})} = T_{\text{star}}(d)$: Sort the correlation coefficients in decreasing order of magnitude and relabel the edges such that $|\rho_{e_1}| \geq \ldots \geq |\rho_{e_{d-1}}|$. Then, from Lemma 4.4(b), the set of crossover rates for the star graph is given by $\{\widetilde{J}(\rho_{e_1}, \rho_{e_1}\rho_{e_2})\} \cup \{\widetilde{J}(\rho_{e_i}, \rho_{e_i}\rho_{e_1}) : i = 2, \ldots, d-1\}$. For edge e_1, the correlation coefficient ρ_{e_2} is the largest correlation coefficient (and hence results in the smallest rate). For all other edges $\{e_i : i \geq 2\}$, the correlation coefficient ρ_{e_1} is the largest possible correlation coefficient (and hence results in the smallest rate). Since each member in the set of crossovers is the minimum possible, the minimum of these crossover rates is also the minimum possible among all tree graphs. □

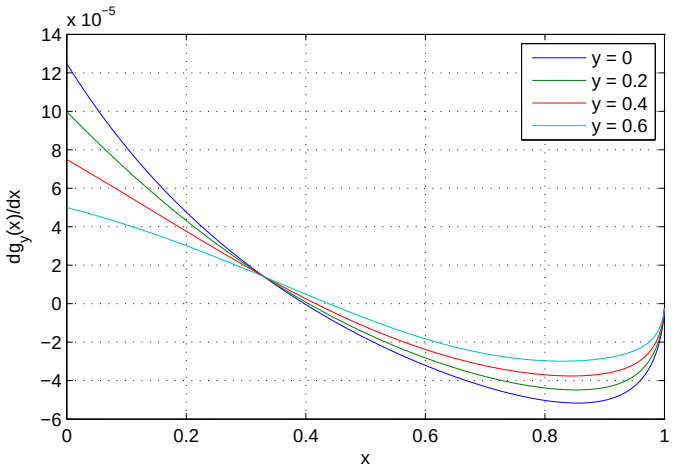

Figure 4.13. Plot of $\frac{d}{dx} g_y(x)$ for different values of y. Note that $\frac{d}{dx} g_y(x)$ is positive when $0 < x < \rho^2_{\text{crit}} = 0.3976$.

Before we prove part (b), we present some properties of the edge weights $W(\rho_i, \rho_j)$, defined in (4.17).

Lemma 4.13. (Properties of Edge Weights) *Assume that all the correlation coefficients are bounded above by ρ_{crit}, i.e., $|\rho_i| \le \rho_{\text{crit}}$. Then $W(\rho_i, \rho_j)$ satisfies the following properties:*

(a) The weights are symmetric, i.e., $W(\rho_i, \rho_j) = W(\rho_j, \rho_i)$.

(b) $W(\rho_i, \rho_j) = \widetilde{J}(\min\{|\rho_i|, |\rho_j|\}, \rho_i \rho_j)$, where $\widetilde{J}$ is the approximate crossover rate given in (4.51).

(c) If $|\rho_i| \ge |\rho_j| \ge |\rho_k|$, then

$$W(\rho_i, \rho_k) \le \min\{W(\rho_i, \rho_j), W(\rho_j, \rho_k)\}. \tag{4.60}$$

(d) If $|\rho_1| \ge \ldots \ge |\rho_{d-1}|$, then

$$W(\rho_i, \rho_j) \le W(\rho_i, \rho_{i+1}), \quad \forall\, j \ge i+1, \tag{4.61a}$$
$$W(\rho_i, \rho_j) \le W(\rho_i, \rho_{i-1}), \quad \forall\, j \le i-1. \tag{4.61b}$$

Proof. Claim (a) follows directly from the definition of $\widetilde{J}$ in (4.17). Claim (b) also follows from the definition of $\widetilde{J}$ and its monotonicity property in Lemma 4.4(d). Claim (c) follows by first using Claim (b) to establish that the RHS of (4.60) equals the minimum of $\widetilde{J}(\rho_j, \rho_j \rho_i)$ and $\widetilde{J}(\rho_k, \rho_k \rho_j)$ since $|\rho_i| \ge |\rho_j| \ge |\rho_k|$. By the same argument, the LHS of (4.60), equals $\widetilde{J}(\rho_k, \rho_k \rho_i)$. Now we have

$$\widetilde{J}(\rho_k, \rho_k \rho_i) \le \widetilde{J}(\rho_j, \rho_j \rho_i), \quad \widetilde{J}(\rho_k, \rho_k \rho_i) \le \widetilde{J}(\rho_k, \rho_k \rho_j), \tag{4.62}$$

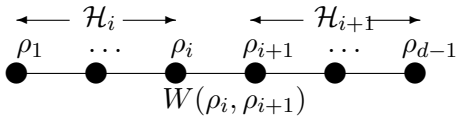

Figure 4.14. Illustration of the proof of Theorem 4.7. Let $|\rho_1| \geq \ldots \geq |\rho_{d-1}|$. The figure shows the chain H^*_{chain} (in the line graph domain) where the correlation coefficients $\{\rho_i\}$ are placed in decreasing order.

where the first and second inequalities follow from Lemmas 4.4(c) and 4.4(b) respectively. This establishes (4.60). Claim (d) follows by applying Claim (c) recursively. □

Proof. Proof of $T_{p_{\max}(\boldsymbol{\rho})} = T_{\text{chain}}(d)$: Assume, without loss of generality, that $|\rho_{e_1}| \geq \ldots \geq |\rho_{e_{d-1}}|$ and we also abbreviate ρ_{e_i} as ρ_i for all $i = 1, \ldots, d-1$. We use the idea of line graphs introduced in Section 2.4.1 and Lemma 4.13. Recall that $\mathcal{L}(\mathcal{T}^d)$ is the set of line graphs of spanning trees with d nodes. From (4.26), the line graph for the structure of the best distribution $p_{\max,\boldsymbol{\rho}}$ for learning in (4.24) is

$$H_{\max,\boldsymbol{\rho}} \;:=\; \operatorname*{argmax}_{H \in \mathcal{L}(\mathcal{T}^d)} \min_{(i,j)\in H} W(\rho_i, \rho_j). \tag{4.63}$$

We now argue that the length $d-1$ chain H^*_{chain} (in the line graph domain) with correlation coefficients $\{\rho_i\}_{i=1}^{d-1}$ arranged in decreasing order on the nodes (see Fig. 4.14) is the line graph that optimizes (4.63). Note that the edge weights of H^*_{chain} are given by $W(\rho_i, \rho_{i+1})$ for $1 \leq i \leq d-2$. Consider any other line graph $H \in \mathcal{L}(\mathcal{T}^d)$. Then we claim that

$$\min_{(i,j)\in H \setminus H^*_{\text{chain}}} W(\rho_i, \rho_j) \leq \min_{(i,j)\in H^*_{\text{chain}} \setminus H} W(\rho_i, \rho_j). \tag{4.64}$$

To prove (4.64), note that any edge $(i,j) \in H^*_{\text{chain}} \setminus H$ is *consecutive*, i.e., of the form $(i, i+1)$. Fix any such $(i, i+1)$. Define the two subchains of H^*_{chain} as $\mathcal{H}_i := \{(1,2), \ldots, (i-1, i)\}$ and $\mathcal{H}_{i+1} := \{(i+1, i+2), \ldots, (d-2, d-1)\}$ (see Fig. 4.14). Also, let $V(\mathcal{H}_i) := \{1, \ldots, i\}$ and $V(\mathcal{H}_{i+1}) := \{i+1, \ldots, d-1\}$ be the nodes in subchains $\mathcal{H}_i$ and $\mathcal{H}_{i+1}$ respectively. Because $(i, i+1) \notin H$, there is a set of edges (called cut set edges) $\mathcal{S}_i := \{(j,k) \in H : j \in V(\mathcal{H}_i), k \in V(\mathcal{H}_{i+1})\}$ to ensure that the line graph H remains connected.[11] The edge weight of each cut set edge $(j,k) \in \mathcal{S}_i$ satisfies $W(\rho_j, \rho_k) \leq W(\rho_i, \rho_{i+1})$ by (4.61) because $|j-k| \geq 2$ and $j \leq i$ and $k \geq i+1$. By considering all cut set edges $(j,k) \in \mathcal{S}_i$ for fixed i and subsequently all $(i, i+1) \in H^*_{\text{chain}} \setminus H$, we establish (4.64). It follows that

$$\min_{(i,j)\in H} W(\rho_i, \rho_j) \leq \min_{(i,j)\in H^*_{\text{chain}}} W(\rho_i, \rho_j), \tag{4.65}$$

[11] The line graph $H = \mathcal{L}(G)$ of a connected graph G is connected. In addition, any $H \in \mathcal{L}(\mathcal{T}^d)$ must be a claw-free, block graph [95, Theorem 8.5].

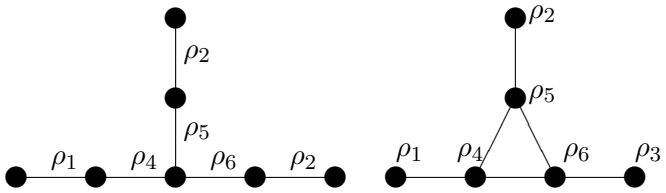

Figure 4.15. A 7-node tree T and its line graph $H = \mathcal{L}(T)$ are shown in the left and right figures respectively. In this case $H \setminus H^*_{\text{chain}} = \{(1,4),(2,5),(4,6),(3,6)\}$ and $H^*_{\text{chain}} \setminus H = \{(1,2),(2,3),(3,4)\}$. Eqn. (4.64) holds because from (4.61), $W(\rho_1,\rho_4) \le W(\rho_1,\rho_2)$, $W(\rho_2,\rho_5) \le W(\rho_2,\rho_3)$ etc. and also if $a_i \le b_i$ for $i \in \mathcal{I}$ (for finite $\mathcal{I}$), then $\min_{i\in\mathcal{I}} a_i \le \min_{i\in\mathcal{I}} b_i$.

because the other edges in H and H^*_{chain} in (4.64) are common. See Fig. 4.15 for an example to illustrate (4.64).

Since the chain line graph H^*_{chain} achieves the maximum bottleneck edge weight, it is the optimal line graph, i.e., $H_{\max,\boldsymbol{\rho}} = H^*_{\text{chain}}$. Furthermore, since the line graph of a chain is a chain, the best structure $T_{p_{\max}(\boldsymbol{\rho})}$ is also a chain and we have established (4.28). The best distribution is given by the chain with the correlations placed in decreasing order, establishing Corollary 4.9. □

Chapter 5

Learning High-Dimensional Forest-Structured Models

5.1 Introduction

IN the previous two chapters, we discussed the learning of tree-structured graphical models of fixed dimensions. More specifically, we performed large-deviations analyses for Gaussian and discrete graphical models. This chapter is motivated in part by a key challenge in learning graphical models; that is the learning problem is often compounded by the fact that typically only a *small* number of samples n are available relative to the size of the model (dimension of data) d. This is referred to as the *high-dimensional learning regime*, which differs from classical statistics where a large number of samples are available to learn a model of fixed size (as in Chapters 3 and 4). The high-dimensional setting is characterized by the fact that *both* the number of samples n *and* the number of dimensions d grow together, i.e., d is a function of n.

This chapter discusses parameter and structure learning of acyclic graphs from i.i.d. samples but differs from the previous two chapters in two distinct ways: Firstly, we focus on the high-dimensional learning regime. Secondly, we seek to learn forest-structured graphical models (instead of tree-structured ones). We choose to learn forests because when the number of samples is small relative to the data dimension, even a tree-structured distribution may overfit the data [132]. For learning the structure of the forest, the ML Chow-Liu algorithm described in Section 2.5.2, does not produce a consistent estimate in general, since ML favors richer model classes with more parameters [133] and hence, outputs a tree. We propose a consistent algorithm called CLThres, which consists of an adaptive thresholding mechanism to prune "weak" edges from the Chow-Liu tree. We provide tight upper and lower bounds on the *overestimation* and *underestimation* errors, that is, the error probability that the output of the algorithm has more or fewer edges than the true model.

This chapter contains three main contributions. Firstly, we prove that CLThres is structurally consistent, i.e., as the number of samples grows for a fixed model size, the probability of learning the incorrect structure (set of edges), decays to zero for a fixed model size. We show that the error rate is dominated by the rate of decay of the overestimation error probability. In our proofs, we use the method of types (see

Section 2.2.1) as well as Euclidean information theory (see (2.24) and (2.25)). We provide an upper bound on the error probability by using convex duality [18, 28] to find a surprising connection between the overestimation error rate and a semidefinite program [200] and show that the overestimation error in structure learning decays faster than any polynomial in n for a fixed data dimension d. Secondly, we consider the high-dimensional scenario and provide sufficient conditions on the growth of (n, d) (and also the true number of edges k) to ensure that CLThres is structurally consistent. We prove that even if d grows faster than any polynomial in n (in fact close to exponential in n), structure estimation remains consistent. We also show that for the proposed algorithm, independent models (resp. tree models) are the "hardest" (resp. "easiest") to learn in the sense that the asymptotic error rate is the highest (resp. lowest), over all models with the same scaling of (n, d). Thus, the empty graph and connected trees are the extremal forest structures for learning. Thirdly, we prove that CLThres is risk consistent. More precisely, the risk of the estimated forest distribution P^* converges to the risk of the forest projection of the true model at a rate of

$$\mathcal{R}_n(P^*) = O_p\left(\frac{d \log d}{n^{1-\gamma}}\right), \qquad \forall\, \gamma > 0. \tag{5.1}$$

We compare and contrast this rate to very recent works such as Liu et al. [132] and Gupta et al. [89].

The work in this chapter is related to and inspired by the large body of literature in information theory on *Markov order estimation*. In these works, the authors use various regularization and model selection schemes to find the optimal order of a Markov chain [52, 77, 138, 141], hidden Markov model [86, 116] or exponential family [137]. We build on some of these ideas and proof techniques to identify the correct set of edges (and in particular the number of edges) in the forest model and also to provide strong theoretical guarantees on the rate of convergence of the estimated forest-structured distribution to the true one.

This chapter is organized as follows: We define and review some mathematical notation and state the problem formally in Section 5.2. In Section 5.3, we describe the algorithm in full detail, highlighting its most salient aspect – the thresholding step. We state our main results on error rates for structure learning in Section 5.4 for a fixed forest-structured distribution. We extend these results to the high-dimensional case when (n, d, k) scale in Section 5.5. Extensions to rates of convergence of the estimated distribution to the true one, i.e., the order of risk consistency, are discussed briefly in Section 5.6. Numerical simulations on synthetic and real data are presented in Section 5.7. Finally, we conclude the discussion in Section 5.8. The proofs of the majority of the results are provided in the appendices at the end of the chapter.

■ 5.2 Notation and Problem Formulation

We now remind the reader of some notation that is used throughout this chapter. The sets of labeled trees with d nodes and labeled forests with d nodes and k (for

$0 \leq k \leq d-1$) edges are denoted as $\mathcal{T}^d$ and $\mathcal{T}_k^d$ respectively. We also use the notation $\mathcal{F}^d := \cup_{k=0}^{d-1} \mathcal{T}_k^d$ to denote the set of labeled forests with d nodes.

Let $\mathcal{X}$ be a finite set with cardinality $r := |\mathcal{X}| \geq 2$. We denote the set of d-variate distributions supported on $\mathcal{X}^d$ and Markov on a forest with k edges as $\mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d) \subset \mathcal{P}(\mathcal{X}^d)$. Similarly, $\mathcal{D}(\mathcal{X}^d, \mathcal{F}^d)$ is the set of forest-structured distributions. Let $P \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d)$ be a discrete forest-structured distribution Markov on $T_P = (V, E_P) \in \mathcal{T}_k^d$ (for some $k = 0, \ldots, d-1$). In this chapter, we always assume that graphs are *minimal representations* for the corresponding graphical model, i.e., if P is Markov on T_P, then T_P has the smallest number of edges for the requisite conditional independence relations in (2.92) and (2.93) to hold.

The *minimum mutual information* in the forest-structured distribution, denoted as

$$I_{\min} := \min_{(i,j) \in E_P} I(P_{i,j}) \tag{5.2}$$

will turn out to be a fundamental quantity in the subsequent analysis. Note from our minimality assumption on the graphical model P that $I_{\min} > 0$ since all edges in the forest have positive mutual information (none of the edges are degenerate). When we consider the scenario where d grows with n in Section 5.5, we assume that $I_{\min}$ is *uniformly* bounded away from zero.

We now state the basic learning problem formally. We are given a set of i.i.d. samples, denoted as $\mathbf{x}^n := \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Each sample $\mathbf{x}_l = (x_{l,1}, \ldots, x_{l,d}) \in \mathcal{X}^d$ is drawn independently from $P \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d)$ a forest-structured distribution. From these samples, and the prior knowledge that the undirected graph is acyclic (but not necessarily connected), estimate the true set of edges E_P as well as the true distribution P consistently. In Section 5.5, we extend the basic learning problem to the scenario where we seek to learn a *sequence* of forest-structured distributions, in order to study how d and k may scale with n while still maintaining consistency

■ 5.3 The Forest Learning Algorithm: CLThres

We now describe our algorithm for estimating the edge set E_P and the distribution P. This algorithm is a modification of the celebrated Chow-Liu algorithm for maximum-likelihood (ML) learning of tree-structured distributions [42]. We call our algorithm CLThres which stands for *Chow-Liu with Thresholding*.

The inputs to the algorithm are the set of samples $\mathbf{x}^n$ and a *regularization* sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$ (to be specified precisely later) that typically decays to zero, i.e., $\lim_{n \to \infty} \varepsilon_n = 0$. The outputs are the estimated edge set, denoted $\widehat{E}_{\widehat{k}_n}$, and the estimated distribution, denoted P^*.

1. Given $\mathbf{x}^n$, calculate the set of empirical mutual information quantities $I(\widehat{P}_{i,j})$ for $1 \leq i, j \leq d$.

2. Run a max-weight spanning tree (MWST) algorithm [120, 158] to obtain an estimate of the edge set:

$$\widehat{E}_{d-1} := \operatorname*{argmax}_{E:T=(V,E)\in\mathcal{T}^d} \sum_{(i,j)\in E} I(\widehat{P}_{i,j}). \tag{5.3}$$

Let the estimated edge set be $\widehat{E}_{d-1} := \{\widehat{e}_1, \ldots, \widehat{e}_{d-1}\}$ where the edges $\widehat{e}_i$ are sorted according to decreasing empirical mutual information values. We index the edge set by $d-1$ to emphasize that it has $d-1$ edges and hence is connected. We denote the sorted empirical mutual information quantities as $I(\widehat{P}_{\widehat{e}_1}) \geq \ldots \geq I(\widehat{P}_{\widehat{e}_{d-1}})$. These first three steps constitute the Chow-Liu algorithm [42].

3. Estimate the true number of edges using the *thresholding estimator*:

$$\widehat{k}_n := \operatorname*{argmin}_{1\leq j\leq d-1} \left\{ I(\widehat{P}_{\widehat{e}_j}) : I(\widehat{P}_{\widehat{e}_j}) \geq \varepsilon_n, I(\widehat{P}_{\widehat{e}_{j+1}}) \leq \varepsilon_n \right\}. \tag{5.4}$$

If there exists an empirical mutual information $I(\widehat{P}_{\widehat{e}_j})$ such that $I(\widehat{P}_{\widehat{e}_j}) = \varepsilon_n$, break the tie arbitrarily.[1]

4. Prune the tree by retaining only the top $\widehat{k}_n$ edges, i.e., define the *estimated edge set* of the forest to be

$$\widehat{E}_{\widehat{k}_n} := \{\widehat{e}_1, \ldots, \widehat{e}_{\widehat{k}_n}\}, \tag{5.5}$$

where $\{\widehat{e}_i : 1 \leq i \leq d-1\}$ is the ordered edge set defined in Step 2. Define the estimated tree to be $\widehat{T}_{\widehat{k}_n} := (V, \widehat{E}_{\widehat{k}_n})$.

5. Finally, define the estimated distribution P^* to be the *reverse I-projection* [51] of the joint type $\widehat{P}$ onto $\widehat{T}_{\widehat{k}_n}$, i.e.,

$$P^*(\mathbf{x}) := \operatorname*{argmin}_{Q\in\mathcal{D}(\mathcal{X}^d, \widehat{T}_{\widehat{k}_n})} D(\widehat{P} \,||\, Q). \tag{5.6}$$

It can easily be shown that the projection can be expressed in terms of the marginal and pairwise joint types:

$$P^*(\mathbf{x}) = \prod_{i\in V} \widehat{P}_i(x_i) \prod_{(i,j)\in\widehat{E}_{\widehat{k}_n}} \frac{\widehat{P}_{i,j}(x_i, x_j)}{\widehat{P}_i(x_i)\widehat{P}_j(x_j)}. \tag{5.7}$$

Intuitively, CLThres first constructs a connected tree $(V, \widehat{E}_{d-1})$ via Chow-Liu (in Steps 1 – 2) before pruning the weak edges (with small mutual information) to obtain the

[1] Here we allow a bit of imprecision by noting that the non-strict inequalities in (5.4) simplify the subsequent analyses because the constraint sets that appear in optimization problems will be closed, hence compact, insuring the existence of optimizers.

final structure $\widehat{E}_{\widehat{k}_n}$. The estimated distribution P^* is simply the ML estimate of the parameters subject to the constraint that P^* is Markov on the learned tree $\widehat{T}_{\widehat{k}_n}$.

Note that if Step 3 is omitted and $\widehat{k}_n$ is defined to be $d-1$, then CLThres simply reduces to the Chow-Liu ML algorithm (described in Section 2.5.2). The Chow-Liu algorithm, which outputs a tree, is guaranteed to fail (not be structurally consistent) if the number of edges in the true model $k < d-1$, which is the problem of interest in this chapter. Thus, Step 3, a model selection step, is essential in estimating the true number of edges k. This step is a generalization of the test for independence of discrete memoryless sources (DMS) discussed by Merhav in [137]. In our work, we exploit the fact that the empirical mutual information $I(\widehat{P}_{\widehat{e}_j})$ corresponding to a pair of independent variables $\widehat{e}_j$ will be very small when n is large, thus a thresholding procedure using the (appropriately chosen) regularization sequence $\{\varepsilon_n\}$ will remove these edges. In fact, the subsequent analysis allows us to conclude that Step 3, in a formal sense, *dominates* the error probability in structure learning. CLThres is also computationally efficient as shown by the following result.

Proposition 5.1. (Complexity of CLThres) CLThres *runs in time* $O((n+\log d)d^2)$.

Proof. The computation of the empirical mutual information values in Step 1 requires $O(nd^2)$ operations. The MWST algorithm in Step 2 requires at most $O(d^2 \log d)$ operations [158]. Steps 3 and 4 simply require the sorting of the empirical mutual information quantities on the learned tree which only requires $O(\log d)$ computations. □

■ 5.4 Structural Consistency For Fixed Model Size

In this section, we keep d and k fixed and consider a probability model P, which is assumed to be Markov on a forest in $\mathcal{T}_k^d$. This is to gain better insight into the problem before we analyze the high-dimensional scenario in Section 5.5 where d and k scale[2] with the sample size n. More precisely, we are interested in quantifying the rate at which the probability of the error event of structure learning[3]

$$\mathcal{A}_n := \left\{ \widehat{E}_{\widehat{k}_n} \neq E_P \right\} \tag{5.8}$$

decays to zero as n tends to infinity. Recall that $\widehat{E}_{\widehat{k}_n}$, with cardinality $\widehat{k}_n$, is the learned edge set by using CLThres.

Before stating the main result of this section in Theorem 5.3, we first state an auxiliary result that essentially says that if one is provided with oracle knowledge of $I_{\min}$, the minimum mutual information in the forest, then the problem is greatly simplified.

[2]In that case P must also scale, i.e., we learn a *family* of models as d and k scale.

[3]This event is analogous to the events denoted by $\mathcal{A}_n$ defined in (3.3) and (4.5) in previous chapters. However, in this chapter, we are interested in edge sets E_P that correspond to forests (and not trees).

Proposition 5.2. (Error Rate with knowledge of $I_{\min}$) *Assume that $I_{\min}$ is known in* CLThres. *Then by letting the regularization sequence be $\varepsilon_n = I_{\min}/2$ for all n, we have*

$$\lim_{n\to\infty} \frac{1}{n} \log P^n(\mathcal{A}_n) < 0, \tag{5.9}$$

i.e., the error probability decays exponentially fast.

Thus, the primary difficulty lies in estimating $I_{\min}$ or alternatively, the number of edges k. Note that if k is known, a simple modification to the Chow-Liu procedure by imposing the constraint that the final structure contains k edges will also yield exponential decay as in (5.9). However, in the realistic case where both $I_{\min}$ and k are unknown, we show in the rest of this section that we can design the regularization sequence ε_n in such a way that the rate of decay of $P^n(\mathcal{A}_n)$ decays almost exponentially fast.

■ 5.4.1 Error Rate for Forest Structure Learning

We now state one of the main results in this chapter. We emphasize that the following result is stated for a fixed forest-structured distribution $P \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d)$ so d and k are also fixed natural numbers.[4]

Theorem 5.3. (Error Rate for Structure Learning) *Assume that the regularization sequence $\{\varepsilon_n\}_{n\in\mathbb{N}}$ satisfies the following two conditions:*

$$\lim_{n\to\infty} \varepsilon_n = 0, \qquad \lim_{n\to\infty} \frac{n\varepsilon_n}{\log n} = \infty. \tag{5.10}$$

Then, if the true model $T_P = (V, E_P)$ is a proper forest ($k < d-1$), there exists a constant $C_P \in (1, \infty)$ such that

$$-C_P \le \liminf_{n\to\infty} \frac{1}{n\varepsilon_n} \log P^n(\mathcal{A}_n) \tag{5.11}$$

$$\le \limsup_{n\to\infty} \frac{1}{n\varepsilon_n} \log P^n(\mathcal{A}_n) \le -1. \tag{5.12}$$

Finally, if the true model $T_P = (V, E_P)$ is a tree ($k = d-1$), then

$$\lim_{n\to\infty} \frac{1}{n} \log P^n(\mathcal{A}_n) < 0, \tag{5.13}$$

i.e., the error probability decays exponentially fast.

[4] As in all error analyses, we have a true model P, which of course has some number k of edges. Our algorithm, of course, does not know this value.

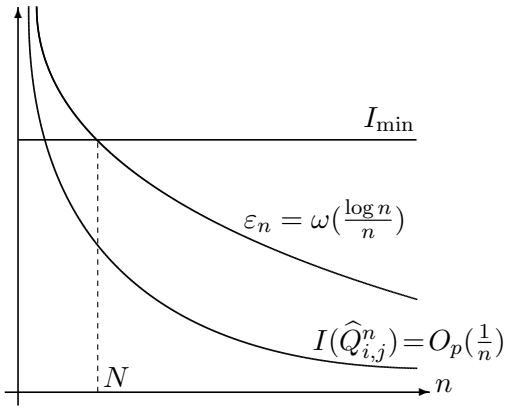

Figure 5.1. Graphical interpretation of the condition on ε_n. As $n \to \infty$, the regularization sequence ε_n will be smaller than $I_{\min}$ and larger than $I(\widehat{Q}^n_{i,j})$ with high probability.

■ 5.4.2 Interpretation of Result

From (5.12), the rate of decay of the error probability for proper forests is subexponential but nonetheless can be made faster than any polynomial for an appropriate choice of ε_n. The reason for the subexponential rate is because of our lack of knowledge of $I_{\min}$, the minimum mutual information in the true forest T_P or alternatively a lack of knowledge of k. For trees, the rate is exponential ($\doteq \exp(-nF)$ for some positive constant F). Learning proper forests is thus, strictly "harder" than learning trees. The condition on ε_n in (5.10) is needed for the following intuitive reasons:

1. Firstly, (5.10) ensures that for all sufficiently large n, we have $\varepsilon_n < I_{\min}$. Thus, the true edges will be correctly identified by CLThres implying that with high probability, there will not be underestimation as $n \to \infty$.

2. Secondly, for two independent random variables X_i and X_j with distribution $Q_{i,j} = Q_i Q_j$, the sequence[5] $\sigma(I(\widehat{Q}^n_{i,j})) = \Theta(1/n)$, where $\widehat{Q}^n_{i,j}$ is the joint empirical distribution of n i.i.d. samples drawn from $Q_{i,j}$. Since the regularization sequence $\varepsilon_n = \omega(\log n/n)$ has a slower rate of decay than $\sigma(I(\widehat{Q}^n_{i,j}))$, we have that $\varepsilon_n > I(\widehat{Q}^n_{i,j})$ with high probability as $n \to \infty$. Thus, with high probability there will not be overestimation as $n \to \infty$.

See Figure 5.1 for an illustration of this intuition. The formal proof follows from a method of types argument and we provide an outline in Section 5.4.3. A convenient choice of ε_n that satisfies (5.10) is

$$\varepsilon_n := n^{-\beta}, \qquad \forall\, \beta \in (0,1). \tag{5.14}$$

[5]The notation $\sigma(Z) = \mathrm{Var}(Z)^{1/2}$ denotes the standard deviation of the random variable Z. The fact that the standard deviation of the empirical MI $\sigma(I(\widehat{Q}^n_{i,j}))$ decays as $1/n$ can be verified by Taylor expanding $I(\widehat{Q}^n_{i,j})$ around $Q_{i,j} = Q_i Q_j$ and using the fact that ML estimates converge to the true values at a rate of $n^{-1/2}$ (see Section 2.2.4).

Note further that the upper bound in (5.12) is also independent of P since it is equal to -1 for all P. Thus, (5.12) is a *universal* result for all forest distributions $P \in \mathcal{D}(\mathcal{X}^d, \mathcal{F}^d)$. The intuition for this universality is because in the large-n regime, the typical way an error occurs is due to overestimation. The overestimation error results from testing whether pairs of random variables are independent and our asymptotic bound for the error probability of this test does not depend on the true distribution P.

The lower bound C_P in (5.11), defined in the proof in Appendix 5.B, means that we cannot hope to do much better using CLThres if the original structure (edge set) is a proper forest. Together, (5.11) and (5.12) imply that the rate of decay of the error probability for structure learning is tight to within a constant factor in the exponent. We believe that the error rates given in Theorem 5.3 cannot, in general, be improved without knowledge of $I_{\min}$. We state a converse (a necessary lower bound on sample complexity) in Theorem 5.7 by treating the unknown forest graph as a uniform random variable over all possible forests of fixed size.

■ 5.4.3 Proof Idea

The method of proof for Theorem 5.3 involves using the Gallager-Fano bounding technique [73, pp. 24] and the union bound to decompose the overall error probability $P^n(\mathcal{A}_n)$ into three distinct terms: (i) the rate of decay of the error probability for learning the top k edges (in terms of the mutual information quantities) correctly – known as the *Chow-Liu error*, (ii) the rate of decay of the *overestimation error* $\{\widehat{k}_n > k\}$ and (iii) the rate of decay of the *underestimation error* $\{\widehat{k}_n < k\}$. Each of these terms is upper bounded using a method of types (cf. Section 2.2.1) argument. It turns out, as is the case with the literature on Markov order estimation (e.g., [77]), that bounding the overestimation error poses the greatest challenge. Indeed, we show that the underestimation and Chow-Liu errors have exponential decay in n. However, the overestimation error has subexponential decay ($\approx \exp(-n\varepsilon_n)$).

The main technique used to analyze the overestimation error relies on Euclidean information theory described in Section 2.1.4 and used extensively in Chapters 3 and 4. Using this approximation and Lagrangian duality [18], we reduce a non-convex I-projection [51] problem involving information-theoretic quantities (such as divergence) to a relatively simple *semidefinite program* [200] which admits a closed-form solution. Furthermore, the Euclidean approximations become *exact* as $n \to \infty$ (i.e., $\varepsilon_n \to 0$), which is the asymptotic regime of interest. The full details of the proof can be found Appendix 5.B.

■ 5.4.4 Error Rate for Learning the Forest Projection

In our discussion thus far, P has been assumed to be Markov on a forest. In this subsection, we consider the situation when the underlying unknown distribution P is not forest-structured but we wish to learn its best forest approximation. To this end,

we define the projection of P onto the set of forests (or *forest projection*) to be

$$\widetilde{P} := \operatorname*{argmin}_{Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{F}^d)} D(P \,||\, Q). \tag{5.15}$$

If there are multiple optimizing distribution, choose a projection $\widetilde{P}$ that is minimal, i.e., its graph $T_{\widetilde{P}} = (V, E_{\widetilde{P}})$ has the *fewest number of edges* such that (5.15) holds. If we redefine the event $\mathcal{A}_n$ in (5.8) to be $\widetilde{\mathcal{A}}_n := \{\widehat{E}_{\widehat{k}_n} \neq E_{\widetilde{P}}\}$, we have the following analogue of Theorem 5.3.

Corollary 5.4. (Error Rate for Learning Forest Projection) *Let P be an arbitrary distribution and the event $\widetilde{\mathcal{A}}_n$ be defined in* (5.15)*. Then the conclusions in* (5.11) – (5.13) *in Theorem 5.3 hold if the regularization sequence $\{\varepsilon_n\}_{n \in \mathbb{N}}$ satisfies* (5.10)*.*

■ 5.5 High-Dimensional Structural Consistency

In the previous section, we considered learning a fixed forest-structured distribution P (and hence fixed d and k) and derived bounds on the error rate for structure learning. However, for most problems of practical interest such as the asthma example presented in Chapter 1, the number of data samples is small compared to the data dimension d. In this section, we prove sufficient conditions on the scaling of (n, d, k) for structure learning to remain consistent. We will see that even if d and k are much larger than n, under some reasonable regularity conditions, structure learning remains consistent.

■ 5.5.1 Structure Scaling Law

To pose the learning problem formally, we consider a *sequence* of structure learning problems indexed by the number of data points n. For the particular problem indexed by n, we have a dataset $\mathbf{x}^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$ of size n where each sample $\mathbf{x}_l \in \mathcal{X}^d$ is drawn independently from an unknown d-variate forest-structured distribution $P^{(d)} \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d)$, which has d nodes and k edges. This *high-dimensional* setup allows us to model and subsequently analyze how d and k can scale with n while maintaining consistency. We will sometimes make the dependence of d and k on n explicit, i.e., $d = d_n$ and $k = k_n$.

In order to be able to learn the structure of the models we assume that

$$\text{(A1)} \quad I_{\inf} := \inf_{d \in \mathbb{N}} \min_{(i,j) \in E_{P^{(d)}}} I(P_{i,j}^{(d)}) > 0, \tag{5.16}$$

$$\text{(A2)} \quad \kappa := \inf_{d \in \mathbb{N}} \min_{x_i, x_j \in \mathcal{X}} P_{i,j}^{(d)}(x_i, x_j) > 0. \tag{5.17}$$

That is, assumptions (A1) and (A2) insure that there exists *uniform* lower bounds on the minimum mutual information and the minimum entry in the pairwise probabilities in the forest models as the size of the graph grows. These are typical regularity assumptions for the high-dimensional setting. See Wainwright et al. [211] and Meinshausen and

Buehlmann [136] for example. We again emphasize that the proposed learning algorithm CLThres has knowledge of neither $I_{\inf}$ nor κ. Equipped with (A1) and (A2) and assuming the asymptotic behavior of ε_n in (5.10), we claim the following theorem for CLThres.

Theorem 5.5. (Structure Scaling Law) *There exists two finite, positive constants* $C_1 = C_1(I_{\inf}, \kappa)$ *and* $C_2 = C_2(I_{\inf}, \kappa)$ *such that if*

$$n > \max\left\{(2\log(d-k))^{1+\zeta},\, C_1 \log d,\, C_2 \log k\right\}, \tag{5.18}$$

for any $\zeta > 0$*, then the error probability of incorrectly learning the sequence of edge sets* $\{E_{P^{(d)}}\}_{d\in\mathbb{N}}$ *tends to zero as* $(n, d, k) \to \infty$*. When the sequence of forests are trees,* $n > \max\{C_1, C_2\} \log d$ *suffices for high-dimensional structure recovery.*

This result is proved in Appendix 5.D. From (5.18), if the model parameters (n, d, k) all grow with n but $d = o(\exp(n/C_1))$, $k = o(\exp(n/C_2))$ and $d - k = o(\exp(n^{1-\beta}/2))$ (for all $\beta > 0$), consistent structure recovery is possible in high dimensions. In other words, the number of nodes d and the number of edges k can grow faster than any polynomial in the sample size n. The difference $d - k$ can grow subexponentially in n. In Liu et al. [132], the bivariate densities are modeled by functions from a Hölder class with exponent α and it was mentioned (in Theorem 4.3) that the number of variables can grow like $o(\exp(n^{\alpha/(1+\alpha)}))$ for structural consistency. Our result is somewhat stronger but we model the pairwise joint distributions as (simpler) probability mass functions (the alphabet $\mathcal{X}$ is a finite set).

■ 5.5.2 Extremal Forest Structures

In this subsection, we study the extremal structures for learning, that is, the structures that, roughly speaking, lead to the largest and smallest error probabilities for structure learning. Define the sequence

$$h_n(P) := \frac{1}{n\varepsilon_n} \log P^n(\mathcal{A}_n), \quad \forall\, n \in \mathbb{N}. \tag{5.19}$$

Note that h_n is a function of both the number of variables $d = d_n$ and the number of edges $k = k_n$ in the models $P^{(d)}$ since it is a sequence indexed by n. In the next result, we assume (n, d, k) satisfies the scaling law in (5.18) and answer the following question: For CLThres, how does h_n in (5.19) depend on the number of edges k_n for a given d_n? Let $P_1^{(d)}$ and $P_2^{(d)}$ be two sequences of forest-structured distributions with a common number of nodes d_n and number of edges $k_n(P_1^{(d)})$ and $k_n(P_2^{(d)})$ respectively.

Corollary 5.6. (Extremal Forests) *As* $n \to \infty$*,* $h_n(P_1^{(d)}) \le h_n(P_2^{(d)})$ *whenever* $k_n(P_1^{(d)}) \ge k_n(P_2^{(d)})$ *implying that* h_n *is maximized when* $P^{(d)}$ *are product distributions (i.e.,* $k_n = 0$*) and minimized when* $P^{(d)}$ *are tree-structured distributions (i.e.,* $k_n = d_n - 1$*). Furthermore, if* $k_n(P_1^{(d)}) = k_n(P_2^{(d)})$*, then* $h_n(P_1^{(d)}) = h_n(P_2^{(d)})$*.*

This result is proved in Appendix 5.E. The intuition for this result is the following: We recall from the discussion after Theorem 5.3 that the overestimation error dominates the probability of error for structure learning. Thus, the performance of CLThres degrades with the number of missing edges. If there are very few edges (i.e., k_n is very small relative to d_n), the CLThres estimator is more likely to overestimate the number of edges as compared to if there are many edges (i.e., k_n/d_n is close to 1). We conclude that a distribution which is Markov on an *empty graph* (all variables are independent) is the *hardest* to learn (in the sense of Corollary 5.6 above). Conversely, *trees* are the *easiest* structures to learn.

■ 5.5.3 Lower Bounds on Sample Complexity

Thus far, our results are for a specific algorithm CLThres for learning the structure of Markov forest distributions. At this juncture, it is natural to ask whether the scaling laws in Theorem 5.5 are the best possible over all algorithms (estimators). To answer this question, we limit ourselves to the scenario where the true graph T_P is a uniformly distributed chance variable[6] with probability measure $\mathbb{P}$. Assume two different scenarios:

(a) T_P is drawn from the uniform distribution on $\mathcal{T}_k^d$, i.e., $\mathbb{P}(T_P = t) = 1/|\mathcal{T}_k^d|$ for all forests $t \in \mathcal{T}_k^d$. Recall that $\mathcal{T}_k^d$ is the set of labeled forests with d nodes and k edges.

(b) T_P is drawn from the uniform distribution on $\mathcal{F}^d$, i.e., $\mathbb{P}(T_P = t) = 1/|\mathcal{F}^d|$ for all forests $t \in \mathcal{F}^d$. Recall that $\mathcal{F}^d$ is the set of labeled forests with d nodes.

This following result is inspired by Theorem 1 in Bresler et al. [32]. Note that an *estimator* or *algorithm* $\widehat{T}^d$ is simply a map from the set of samples $(\mathcal{X}^d)^n$ to a set of graphs (either $\mathcal{T}_k^d$ or $\mathcal{F}^d$). We emphasize that the following result is stated with the assumption that we are *taking expectations* over the random choice of the true graph T_P.

Theorem 5.7. (Lower Bounds on Sample Complexity) *Let $\varrho < 1$ and $r := |\mathcal{X}|$. In case (a) above, if*

$$n < \varrho \, \frac{(k-1)\log d}{d \log r}, \tag{5.20}$$

then $\mathbb{P}(\widehat{T}^d \neq T_P) \to 1$ for any estimator $\widehat{T}^d : (\mathcal{X}^d)^n \to \mathcal{T}_k^d$. Alternatively, in case (b), if

$$n < \varrho \, \frac{\log d}{\log r}, \tag{5.21}$$

then $\mathbb{P}(\widehat{T}^d \neq T_P) \to 1$ for any estimator $\widehat{T}^d : (\mathcal{X}^d)^n \to \mathcal{F}^d$.

[6]The term *chance variable*, attributed to [85], describes random quantities $Y : \Omega \to W$ that take on values in arbitrary alphabets W. In contrast, a random variable X maps the sample space Ω to the reals $\mathbb{R}$.

This result, proved in Appendix 5.F is a *strong converse* and states that $n = \Omega(\frac{k}{d}\log d)$ is *necessary* for any estimator with oracle knowledge of k to succeed. Thus, we need at least logarithmically many samples in d if the fraction k/d is kept constant as the graph size grows even if k *is known precisely* and does not have to be estimated. Interestingly, (5.20) says that if k is large, then we need more samples. This is because there are fewer forests with a small number of edges as compared to forests with a large number of edges. In contrast, the performance of CLThres (which does not have knowledge of k) degrades when k is small because it is more sensitive to the overestimation error. Moreover, if the estimator does not know k, then (5.21) says that $n = \Omega(\log d)$ is *necessary* for successful recovery. We conclude that the set of scaling requirements prescribed in Theorem 5.5 is almost optimal. In fact, if the true structure T_P is a tree, then Theorem 5.7 for CLThres says that the (achievability) scaling laws in Theorem 5.5 are indeed optimal (up to constant factors in the O and Ω-notation) since $n > (2\log(d-k))^{1+\zeta}$ in (5.18) is trivially satisfied. Note that if T_P is a tree, then the Chow-Liu ML procedure or CLThres results in the sample complexity $n = O(\log d)$ (see Theorem 5.5).

■ 5.6 Risk Consistency

In this section, we develop results for risk consistency to study how fast the parameters of the estimated distribution converge to their true values. For this purpose, we define the *risk* of the estimated distribution P^* (with respect to the true probability model P) as

$$\mathcal{R}_n(P^*) := D(P \,||\, P^*) - D(P \,||\, \widetilde{P}), \tag{5.22}$$

where $\widetilde{P}$ is the forest projection of P defined in (5.15). Note that the original probability model P does not need to be a forest-structured distribution in the definition of the risk. Indeed, if P is Markov on a forest, (5.22) reduces to $\mathcal{R}_n(P^*) = D(P \,||\, P^*)$ since the second term is zero. We quantify the rate of decay of the risk when the number of samples n tends to infinity. For $\delta > 0$, we define the event

$$\mathcal{C}_{n,\delta} := \left\{ \mathbf{x}^n \in (\mathcal{X}^d)^n : \frac{\mathcal{R}_n(P^*)}{d} > \delta \right\}. \tag{5.23}$$

That is, $\mathcal{C}_{n,\delta}$ is the event that the *average risk* $\mathcal{R}_n(P^*)/d$ exceeds some constant δ. We say that the estimator P^* (or an algorithm producing P^*) is *δ-risk consistent* if the probability of $\mathcal{C}_{n,\delta}$ tends to zero as $n \to \infty$. Intuitively, achieving δ-risk consistency is easier than achieving structural consistency since the learned model P^* can be close to the true forest-projection $\widetilde{P}$ in the KL-divergence sense even if their structures differ.

We say that a reconstruction algorithm has *risk consistency of order* (or *rate*) g_n if $\mathcal{R}_n(P^*) = O_p(g_n)$. The definition of the order of risk consistency involves the true model P. Intuitively, we expect that as $n \to \infty$, the estimated distribution P^* converges to the projection $\widetilde{P}$ so $\mathcal{R}_n(P^*) \to 0$ in probability.

■ 5.6.1 Error Exponent for Risk Consistency

In this subsection, we consider a fixed distribution P and state consistency results in terms of the event $\mathcal{C}_{n,\delta}$. Consequently, the model size d and the number of edges k are fixed. This lends insight into deriving results for the order of the risk consistency and provides intuition for the high-dimensional scenario in Section 5.6.2.

Theorem 5.8. (Error Exponent for δ-Risk Consistency) *For* CLThres*, there exists a constant $\delta_0 > 0$ such that for all $0 < \delta < \delta_0$,*

$$\limsup_{n\to\infty} \frac{1}{n} \log P^n(\mathcal{C}_{n,\delta}) \leq -\delta. \tag{5.24}$$

The corresponding lower bound is

$$\liminf_{n\to\infty} \frac{1}{n} \log P^n(\mathcal{C}_{n,\delta}) \geq -\delta\, d. \tag{5.25}$$

The theorem, proved in Appendix 5.G, states that if δ is sufficiently small, the decay rate of the probability of $\mathcal{C}_{n,\delta}$ is exponential, hence clearly CLThres is δ-risk consistent. Furthermore, the bounds on the error exponent associated to the event $\mathcal{C}_{n,\delta}$ are *independent* of the parameters of P and only depend on δ and the dimensionality d. Intuitively, (5.24) is true because if we want the risk of P^* to be at most δd, then each of the empirical pairwise marginals $\widehat{P}_{i,j}$ should be δ-close to the true pairwise marginal $\widetilde{P}_{i,j}$. Note also that for $\mathcal{C}_{n,\delta}$ to occur with high probability, the edge set does not need to be estimated correctly so there is no dependence on k.

■ 5.6.2 The High-Dimensional Setting

We again consider the high-dimensional setting where the tuple of parameters (n, d_n, k_n) tend to infinity and we have a sequence of learning problems indexed by the number of data points n. We again assume that (5.16) and (5.17) hold and derive sufficient conditions under which the probability of the event $\mathcal{C}_{n,\delta}$ tends to zero for a sequence of d-variate distributions $\{P^{(d)} \in \mathcal{P}(\mathcal{X}^d)\}_{d\in\mathbb{N}}$. The proof of Theorem 5.8 leads immediately to the following corollary.

Corollary 5.9. (δ-Risk Consistency Scaling Law) *Let $\delta > 0$ be a sufficiently small constant and $a \in (0, \delta)$. If the number of variables in the sequence of models $\{P^{(d)}\}_{d\in\mathbb{N}}$ satisfies $d_n = o\left(\exp(an)\right)$, then* CLThres *is δ-risk consistent for $\{P^{(d)}\}_{d\in\mathbb{N}}$.*

Interestingly, this sufficient condition on how number of variables d should scale with n for consistency is very similar to Theorem 5.5. In particular, if d is polynomial in n, then CLThres is both structurally consistent as well as δ-risk consistent. We now study the order of the risk consistency of CLThres as the model size d grows.

Theorem 5.10. (Order of Risk Consistency) *Fix $\gamma > 0$. The risk of the sequence of estimated distributions $\{(P^{(d)})^*\}_{d\in\mathbb{N}}$ with respect to the probability models $\{P^{(d)}\}_{d\in\mathbb{N}}$*

satisfies

$$\mathcal{R}_n((P^{(d)})^*) = O_p\left(\frac{d\log d}{n^{1-\gamma}}\right). \tag{5.26}$$

That is, the risk consistency for CLThres *is of order* $(d\log d)/n^{1-\gamma}$.

This result, proved in Appendix 5.I, implies that if $d = o(n^{1-2\gamma})$ then CLThres is risk consistent, i.e., $\mathcal{R}_n((P^{(d)})^*) \to 0$ in probability. Note that this result is not the same as the conclusion of Corollary 5.9 which refers to the probability that the average risk is greater than a fixed constant δ. Also, note that the order of convergence given in (5.26) does not depend on the true number of edges k. This is a consequence of the result in (5.24) where the upper bound on the exponent associated to the event $\mathcal{C}_{n,\delta}$ is independent of the parameters of P.

The order of the risk, or equivalently the rate of convergence of the estimated distribution to the forest projection, is almost linear in the number of variables d and inversely proportional to n. We provide three intuitive reasons to explain why this is plausible:

1. The dimension of the vector of sufficient statistics in a tree-structured graphical model is of the order $O(d)$ (see Section 2.4.3).

2. The ML estimator of the natural parameters of an exponential family converges to its true value at the rate of $O_p(n^{-1/2})$ (see Section 4.2.2 in Serfling [177] or Section 2.2.4).

3. Locally, the KL-divergence behaves like the square of a weighted Euclidean norm of the natural parameters (see Eq. (2.24)).

We now compare Theorem 5.10 to the corresponding results in Liu et al. [132] and Gupta et al. [89]. In these recent papers, it was shown that by modeling the bivariate densities $\widehat{P}_{i,j}$ as functions from a Hölder class with exponent $\alpha > 0$ and using a reconstruction algorithm based on validation on a held-out dataset, the risk decays at a rate[7] of $\widetilde{O}_p(dn^{-\alpha/(1+2\alpha)})$, which is slower than the order of risk consistency in (5.26). This is due to the need to compute the bivariate densities via kernel density estimation. Furthermore, we model the pairwise joint distributions as discrete probability mass functions and not continuous probability density functions, hence there is no dependence on Hölder exponents.

■ 5.7 Numerical Results

In this section, we perform numerical simulations on synthetic and real datasets to study the effect of a finite number of samples on the probability of the event $\mathcal{A}_n$ defined in (5.8). Recall that this is the error event associated to an incorrect learned structure.

[7]The $\widetilde{O}_p(\cdot)$ notation suppresses the dependence on factors involving logarithms.

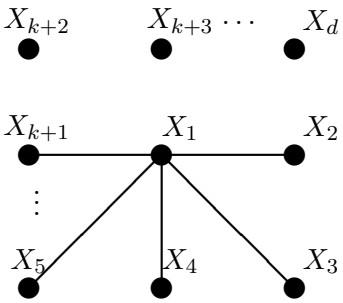

Figure 5.2. The forest-structured distribution Markov on d nodes and k edges. Variables $X_{k+1}, \ldots, X_d$ are not connected to the main star graph.

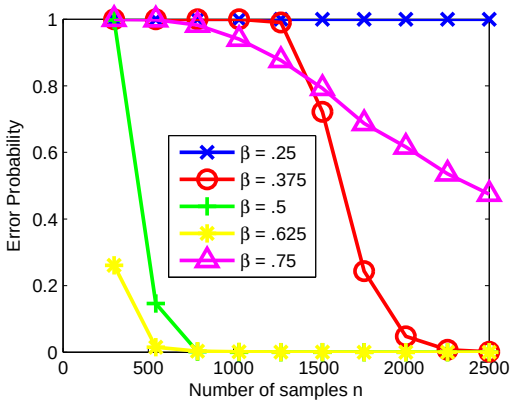

Figure 5.3. The error probability of structure learning for $\beta \in (0, 1)$.

■ 5.7.1 Synthetic Datasets

In order to compare our estimate to the ground truth graph, we learn the structure of distributions that are Markov on the forest shown in Figure 5.2. Thus, a subgraph (nodes $1, \ldots, k+1$) is a (connected) star while nodes $k+2, \ldots, d-1$ are not connected to the star. Each random variable X_j takes on values from a binary alphabet $\mathcal{X} = \{0, 1\}$. Furthermore, $P_j(x_j) = 0.5$ for $x_j = 0, 1$ and all $j \in V$. The conditional distributions are governed by the "binary symmetric channel":

$$P_{j|1}(x_j|x_1) = \begin{cases} 0.7 & x_j = x_1 \\ 0.3 & x_j \neq x_1 \end{cases} \tag{5.27}$$

for $j = 2, \ldots, k+1$. We further assume that the regularization sequence is given by $\varepsilon_n := n^{-\beta}$ for some $\beta \in (0, 1)$. Recall that this sequence satisfies the conditions in (5.10). We vary β in our experiments to observe its effect on the overestimation and underestimation errors.

In Figure 5.3, we show the simulated error probability as a function of the sample size n for a $d = 101$ node graph (as in Figure 5.2) with $k = 50$ edges. The error probability is

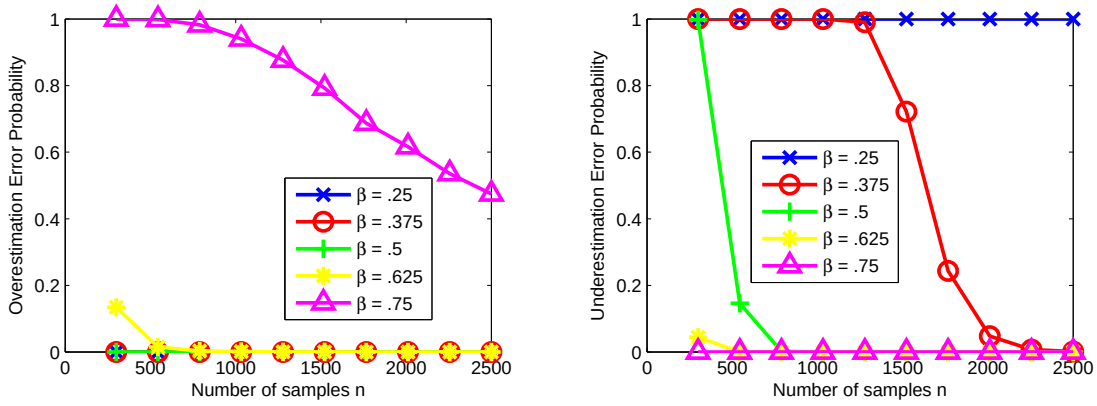

Figure 5.4. The overestimation and underestimation errors for $\beta \in (0, 1)$.

estimated based on 30,000 independent runs of CLThres (over different datasets $\mathbf{x}^n$). We observe that the error probability is minimized when $\beta \approx 0.625$. Figure 5.4 shows the simulated overestimation and underestimation errors for this experiment. We see that as $\beta \to 0$, the overestimation (resp. underestimation) error is likely to be small (resp. large) because the regularization sequence ε_n is large. When the number of samples is relatively small as in this experiment, both types of errors contribute significantly to the overall error probability. When $\beta \approx 0.625$, we have the best tradeoff between overestimation and underestimation for this particular experimental setting.

Even though we mentioned that β in (5.14) should be chosen to be close to zero so that the error probability of structure learning decays as rapidly as possible, this example demonstrates that when given a finite number of samples, β should be chosen to balance the overestimation and underestimation errors. This does not violate Theorem 5.3 since Theorem 5.3 is an asymptotic result and refers to the typical way an error occurs in the limit as $n \to \infty$. Indeed, when the number of samples is very large, it is shown that the overestimation error dominates the overall probability of error and so one should choose β to be close to zero. The question of how best to select optimal β when given only a finite number of samples appears to be a challenging one. We use cross-validation as a proxy to select this parameter for the real-world datasets in the next section.

In Figure 5.5, we fix the value of β at 0.625 and plot the KL-divergence $D(P \,||\, P^*)$ as a function of the number of samples. This is done for a forest-structured distribution P whose graph is shown in Figure 5.2 and with $d = 21$ nodes and $k = 10$ edges. The mean, minimum and maximum KL-divergences are computed based on 50 independent runs of CLThres. We see that $\log D(P \,||\, P^*)$ decays linearly. Furthermore, the slope of the mean curve is approximately -1, which is in agreement with (5.26). This experiment shows that if we want to reduce the KL-divergence between the estimated and true models by a constant factor $A > 0$, we need to increase the number of samples by roughly the same factor A.

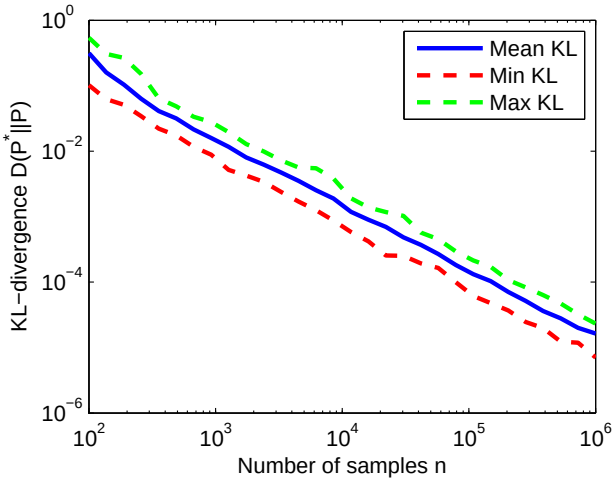

Figure 5.5. Mean, minimum and maximum (across 50 different runs) of the KL-divergence between the estimated model P^* and the true model P for a $d = 21$ node graph with $k = 10$ edges.

■ 5.7.2 Real datasets

We now demonstrate how well forests-structured distributions can model two real datasets[8] which are obtained from the UCI Machine Learning Repository [144]. The first dataset we used is known as the SPECT Heart dataset, which describes diagnosing of cardiac Single Proton Emission Computed Tomography (SPECT) images on normal and abnormal patients. The dataset contains $d = 22$ binary variables and $n = 80$ training samples. There are also 183 test samples. We learned a forest-structured distributions using the 80 training samples for different $\beta \in (0, 1)$ and subsequently computed the log-likelihood of both the training and test samples. The results are displayed in Figure 5.6. We observe that, as expected, the log-likelihood of the training samples increases monotonically with β. This is because there are more edges in the model when β is large improving the modeling ability. However, we observe that there is overfitting when β is large as evidenced by the decrease in the log-likelihood of the 183 test samples. The optimal value of β in terms of the log-likelihood for this dataset is ≈ 0.25, but surprisingly an approximation with an empty graph[9] also yields a high log-likelihood score on the test samples. This implies that according to the available data, the variables are nearly independent. The forest graph for $\beta = 0.25$ is shown in Figure 5.7(a) and is very sparse.

The second dataset we used is the Statlog Heart dataset containing physiological measurements of subjects with and without heart disease. There are 270 subjects and

[8]These datasets are typically employed for binary classification but we use them for modeling purposes.

[9]When $\beta = 0$ we have an empty graph because all empirical mutual information quantities in this experiment are smaller than 1.

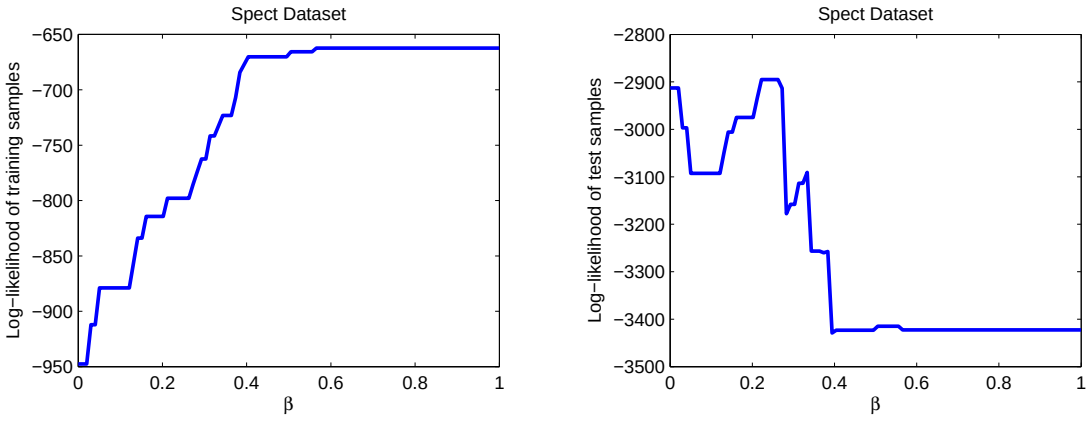

Figure 5.6. Log-likelihood scores on the SPECT dataset

$d = 13$ discrete and continuous attributes, such as gender and resting blood pressure. We quantized the continuous attributes into two bins. Those measurements that are above the mean are encoded as 1 and those below the mean as 0. Since the raw dataset is not partitioned into training and test sets, we learned forest-structured models based on a randomly chosen set of $n = 230$ training samples and then computed the log-likelihood of these training and 40 remaining test samples. We then chose an additional 49 randomly partitioned training and test sets and performed the same learning task and computation of log-likelihood scores. The mean of the log-likelihood scores over these 50 runs is shown in Figure 5.8. We observe that the log-likelihood on the test set is maximized at $\beta \approx 0.53$ and the tree approximation ($\beta \approx 1$) also yields a high likelihood score. The forest learned when $\beta = 0.53$ is shown in Figure 5.7(b). Observe that two nodes (ECG and Cholesterol) are disconnected from the main graph because their mutual information values with other variables are below the threshold. In contrast, HeartDisease, the label for this dataset, has the highest degree, i.e., it influences and is influenced by many other covariates. The strengths of the interactions between HeartDisease and its neighbors are also strong as evidenced by the bold edges.

From these experiments, we observe that some datasets can be modeled well as proper forests with very few edges while others are better modeled as distributions that are almost tree-structured (see Figure 5.7). Also, we need to choose β carefully to balance between data fidelity and overfitting. In contrast, our asymptotic result in Theorem 5.3 says that ε_n should be chosen according to (5.10) so that we have structural consistency. When the number of data points n is large, β in (5.14) should be chosen to be small to ensure that the learned edge set is equal to the true one (assuming the underlying model is a forest) with high probability as the overestimation error dominates.

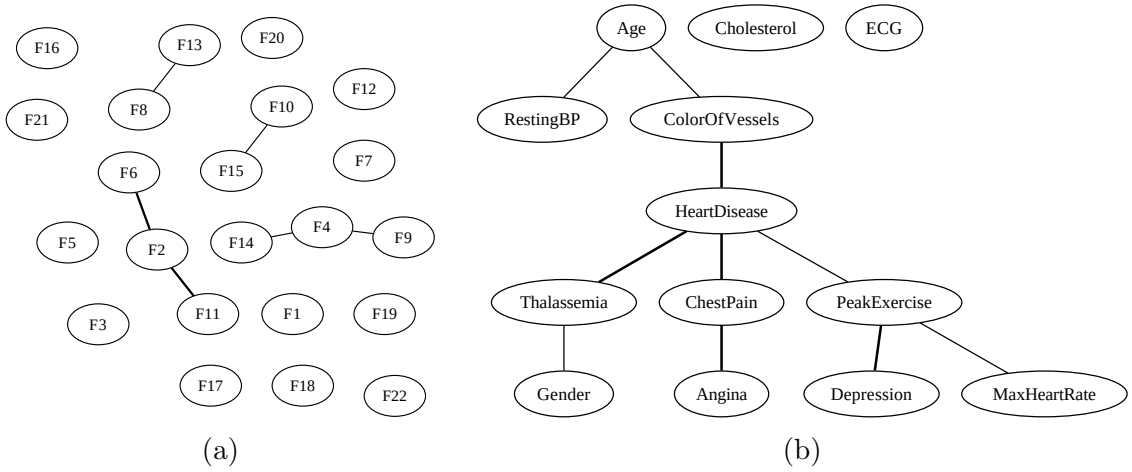

Figure 5.7. Learned forest graph of the (a) SPECT dataset for $\beta = 0.25$ and (b) HEART dataset for $\beta = 0.53$. Bold edges denote higher mutual information values. The features names are not provided for the SPECT dataset.

■ 5.8 Chapter Summary

In this chapter, we proposed an efficient algorithm CLThres for learning the parameters and the structure of forest-structured graphical models. We showed that the asymptotic error rates associated to structure learning are nearly optimal. We also provided the rate at which the error probability of structure learning tends to zero and the order of the risk consistency. There are many open problems that could possibly leverage on the proof techniques employed here. For example, we can analyze the learning of *locally tree-like graphical models* [58, 140] such as Ising models [139] on Erdös-Rényi random graphs [24] using similar thresholding-like techniques on empirical correlation coefficients. We discuss this line of research, which is currently ongoing, in greater detail in Chapter 8.

Appendices for Chapter 5

■ 5.A Proof of Proposition 5.2

Proof. (*Sketch*) The proof of this result hinges on the fact that both the overestimation and underestimation errors decay to zero exponentially fast when the threshold is chosen to be $I_{\min}/2$. This threshold is able to differentiate between true edges (with MI larger than $I_{\min}$) from non-edges (with MI smaller than $I_{\min}$) with high probability for n sufficiently large. The error for learning the top k edges of the forest also decays exponentially fast (see Chapter 3). Thus, (5.9) holds. The full details of the proof follow in a straightforward manner from Appendix 5.B. □

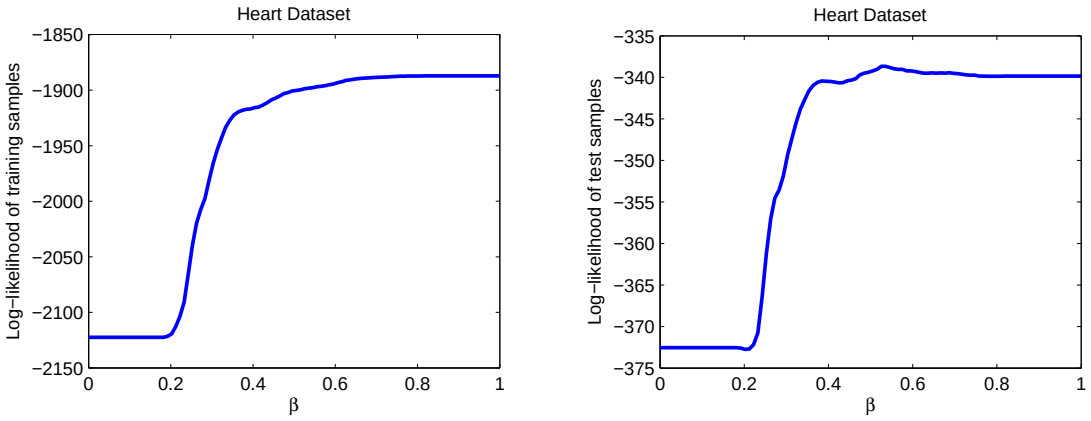

Figure 5.8. Log-likelihood scores on the HEART dataset

■ 5.B Proof of Theorem 5.3

Define the event $\mathcal{B}_n := \{\widehat{E}_k \neq E_P\}$, where $\widehat{E}_k = \{\widehat{e}_1, \ldots, \widehat{e}_k\}$ is the set of top k edges (see Step 2 of CLThres for notation). This is the Chow-Liu error as mentioned in Section 5.4.3. Note that in $\mathcal{B}_n^c$, the estimated edge set depends on k, the true model order, which is *a-priori* unknown to the learner. Further define the constant

$$K_P := \lim_{n\to\infty} -\frac{1}{n} \log P^n(\mathcal{B}_n). \tag{5.28}$$

In other words, K_P is the error exponent for learning the forest structure incorrectly assuming the true model order k is known and Chow-Liu terminates after the addition of exactly k edges in the MWST procedure [120, 158]. The existence of the limit in (5.28) and the positivity of K_P follow from the main results in Chapter 3.

We first state a result which relies on the Gallager-Fano bound [73, pp. 24]. The proof will be provided at the end of this appendix.

Lemma 5.11. (Reduction to Model Order Estimation) *For every $\eta \in (0, K_P)$, there exists a $N \in \mathbb{N}$ sufficiently large such that for every $n > N$, the error probability $P^n(\mathcal{A}_n)$ satisfies*

$$(1-\eta)P^n(\widehat{k}_n \neq k|\mathcal{B}_n^c) \leq P^n(\mathcal{A}_n) \tag{5.29}$$

$$\leq P^n(\widehat{k}_n \neq k|\mathcal{B}_n^c) + 2\exp(-n(K_P - \eta)). \tag{5.30}$$

Proof. (*of Theorem 5.3*) We prove (i) the upper bound in (5.12) (ii) the lower bound in (5.11) and (iii) the exponential rate of decay in the case of trees (5.13).

Proof of upper bound in Theorem 5.3

We now bound the error probability $P^n(\widehat{k}_n \neq k|\mathcal{B}_n^c)$ in (5.30). Using the union bound,

$$P^n(\widehat{k}_n \neq k|\mathcal{B}_n^c) \leq P^n(\widehat{k}_n > k|\mathcal{B}_n^c) + P^n(\widehat{k}_n < k|\mathcal{B}_n^c). \tag{5.31}$$

The first and second terms are known as the *overestimation* and *underestimation* errors respectively. We show that the underestimation error decays exponentially fast. The overestimation error decays only subexponentially fast and so its rate of decay dominates the overall rate of decay of the error probability for structure learning.

Underestimation Error

We now bound these terms staring with the underestimation error. By the union bound,

$$P^n(\widehat{k}_n < k|\mathcal{B}_n^c) \leq (k-1) \max_{1\leq j\leq k-1} P^n(\widehat{k}_n = j|\mathcal{B}_n^c) \tag{5.32}$$

$$= (k-1)P^n(\widehat{k}_n = k-1|\mathcal{B}_n^c), \tag{5.33}$$

where (5.33) follows because $P^n(\widehat{k}_n = j|\mathcal{B}_n^c)$ is maximized when $j = k-1$. By the rule for choosing $\widehat{k}_n$ in (5.4),

$$P^n(\widehat{k}_n = k-1|\mathcal{B}_n^c) = P^n(\exists\, e \in E_P \text{ s.t. } I(\widehat{P}_e) \leq \varepsilon_n), \tag{5.34}$$

$$\leq k \max_{e\in E_P} P^n(I(\widehat{P}_e) \leq \varepsilon_n), \tag{5.35}$$

where (5.35) follows from the union bound. Now, note that if $e \in E_P$, then $I(P_e) > \varepsilon_n$ for n sufficiently large (since $\varepsilon_n \to 0$). Thus, by Sanov's theorem

$$P^n(I(\widehat{P}_e) \leq \varepsilon_n) \leq (n+1)^{r^2} \exp\left(-n \min_{Q\in\mathcal{P}(\mathcal{X}^2)} \{D(Q\,||\,P_e) : I(Q) \leq \varepsilon_n\}\right). \tag{5.36}$$

Define the good rate function [59] in (5.36) to be $L : \mathcal{P}(\mathcal{X}^2) \times [0,\infty) \to [0,\infty)$, which is given by

$$L(P_e; a) := \min_{Q\in\mathcal{P}(\mathcal{X}^2)} \{D(Q\,||\,P_e) : I(Q) \leq a\}. \tag{5.37}$$

Clearly, $L(P_e; a)$ is continuous in a. Furthermore it is monotonically decreasing in a for fixed P_e. Thus, to every $\eta \in (0, L(\varepsilon_n; 0))$, there exists a $N \in \mathbb{N}$ such that for all $n > N$ we have $L(P_e; \varepsilon_n) > L(P_e; 0) - \eta$. As such, we can further upper bound the error probability in (5.36) as

$$P^n(I(\widehat{P}_e) \leq \varepsilon_n) \leq (n+1)^{r^2} \exp\left(-n(L(P_e; 0) - \eta)\right). \tag{5.38}$$

By using the fact that $I_{\min} > 0$, the exponent $L(P_e; 0) > 0$ and thus, we can put the pieces in (5.33), (5.35) and (5.38) together to show that the underestimation error is upper bounded as

$$P^n(\widehat{k}_n < k|\mathcal{B}_n^c) \leq k(k-1)(n+1)^{r^2} \exp\left(-n \min_{e\in E_P} (L(P_e; 0) - \eta)\right). \tag{5.39}$$

Hence, if k is constant, the underestimation error $P^n(\widehat{k}_n < k|\mathcal{B}_n^c)$ decays to zero exponentially fast as $n \to \infty$, i.e,

$$\limsup_{n\to\infty} \frac{1}{n} \log P^n(\widehat{k}_n < k|\mathcal{B}_n^c) \leq -\min_{e\in E_P} (L(P_e; 0) - \eta). \tag{5.40}$$

Now take the limit as $\eta \to 0$ to conclude that:

$$\limsup_{n\to\infty} \frac{1}{n} \log P^n(\widehat{k}_n < k | \mathcal{B}_n^c) \leq -L_P. \tag{5.41}$$

The exponent $L_P := \min_{e\in E_P} L(P_e; 0)$ is positive because we assumed that the model is minimal and so $I_{\min} > 0$, which ensures the positivity of the rate function $L(P_e; 0)$ for each true edge $e \in E_P$.

Overestimation Error

Bounding the overestimation error is harder. It follows by first applying the union bound:

$$P^n(\widehat{k}_n > k | \mathcal{B}_n^c) \leq (d-k-1) \max_{k+1\leq j\leq d-1} P^n(\widehat{k}_n = j | \mathcal{B}_n^c) \tag{5.42}$$

$$= (d-k-1) P^n(\widehat{k}_n = k+1 | \mathcal{B}_n^c), \tag{5.43}$$

where (5.43) follows because $P^n(\widehat{k}_n = j | \mathcal{B}_n^c)$ is maximized when $j = k+1$ in (5.42). Apply the union bound again, we have

$$P^n(\widehat{k}_n = k+1 | \mathcal{B}_n^c) \leq (d-k-1) \max_{e\in V\times V: I(P_e)=0} P^n(I(\widehat{P}_e) \geq \varepsilon_n). \tag{5.44}$$

From (5.44), it suffices to bound $P^n(I(\widehat{P}_e) \geq \varepsilon_n)$ for any pair of independent random variables (X_i, X_j) and $e = (i, j)$. We proceed by applying the upper bound in Sanov's theorem which yields

$$P^n(I(\widehat{P}_e) \geq \varepsilon_n) \leq (n+1)^{r^2} \exp\left(-n \min_{Q\in\mathcal{P}(\mathcal{X}^2)} \{ D(Q \,||\, P_e) : I(Q) \geq \varepsilon_n \} \right), \tag{5.45}$$

for all $n \in \mathbb{N}$. Our task now is to lower bound the good rate function in (5.45), which we denote as $M : \mathcal{P}(\mathcal{X}^2) \times [0, \infty) \to [0, \infty)$:

$$M(P_e; b) := \min_{Q\in\mathcal{P}(\mathcal{X}^2)} \{ D(Q \,||\, P_e) : I(Q) \geq b \}. \tag{5.46}$$

Note that $M(P_e; b)$ is monotonically increasing and continuous in b for fixed P_e. Because the sequence $\{\varepsilon_n\}_{n\in\mathbb{N}}$ tends to zero, when n is sufficiently large, ε_n is arbitrarily small and we are in the so-called *very-noisy regime* [26], where the optimizer to (5.46), denoted as Q_n^*, is very close to P_e. See Figure 5.9. Thus, when n is large, the KL-divergence and mutual information can be approximated as

$$D(Q_n^* \,||\, P_e) = \frac{1}{2} \mathbf{v}^T \mathbf{\Pi}_e \mathbf{v} + o(\|\mathbf{v}\|^2), \tag{5.47}$$

$$I(Q_n^*) = \frac{1}{2} \mathbf{v}^T \mathbf{H}_e \mathbf{v} + o(\|\mathbf{v}\|^2), \tag{5.48}$$

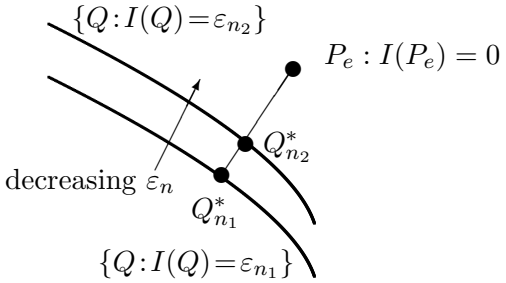

Figure 5.9. As $\varepsilon_n \to 0$, the projection of P_e onto the constraint set $\{Q : I(Q) \geq \varepsilon_n\}$, denoted Q^*_n (the optimizer in (5.46)), approaches P_e. The approximations in (5.47) and (5.48) become increasingly accurate as ε_n tends to zero. In the figure, $n_2 > n_1$ and $\varepsilon_{n_1} > \varepsilon_{n_2}$ and the curves are the (sub-)manifold of distributions such that the mutual information is constant, i.e., the mutual information level sets.

where[10] $\mathbf{v} := \mathrm{vec}(Q^*_n) - \mathrm{vec}(P_e) \in \mathbb{R}^{r^2}$. The $r^2 \times r^2$ matrices $\mathbf{\Pi}_e$ and $\mathbf{H}_e$ are defined as

$$\mathbf{\Pi}_e := \mathrm{diag}(1/\mathrm{vec}(P_e)), \tag{5.49}$$

$$\mathbf{H}_e := \nabla^2_{\mathrm{vec}(Q)} I(\mathrm{vec}(Q))\big|_{Q=P_e}. \tag{5.50}$$

In other words, $\mathbf{\Pi}_e$ is the diagonal matrix that contains the reciprocal of the elements of $\mathrm{vec}(P_e)$ on its diagonal. $\mathbf{H}_e$ is the Hessian[11] of $I(\mathrm{vec}(Q))$, viewed as a function of $\mathrm{vec}(Q)$ and evaluated at P_e. As such, the exponent for overestimation in (5.46) can be approximated by a *quadratically constrained quadratic program* (QCQP), where $\mathbf{z} := \mathrm{vec}(Q) - \mathrm{vec}(P_e)$:

$$\widetilde{M}(P_e; \varepsilon_n) = \min_{\mathbf{z} \in \mathbb{R}^{r^2}} \frac{1}{2} \mathbf{z}^T \mathbf{\Pi}_e \mathbf{z}, \tag{5.51}$$

$$\text{subject to} \quad \frac{1}{2} \mathbf{z}^T \mathbf{H}_e \mathbf{z} \geq \varepsilon_n, \quad \mathbf{z}^T \mathbf{1} = 0. \tag{5.52}$$

We now argue that the approximate rate function $\widetilde{M}$ in (5.52), can be lower bounded by a quantity that is proportional to ε_n. To show this, we resort to Lagrangian duality [18, Ch. 5]. It can easily be shown that the *Lagrangian dual* corresponding to the primal in (5.52) is

$$g(P_e; \varepsilon_n) := \varepsilon_n \max_{\mu \geq 0} \{\mu : \mathbf{\Pi}_e \succeq \mu \mathbf{H}_e\}. \tag{5.53}$$

We see from (5.53) that $g(P_e; \varepsilon_n)$ is proportional to ε_n. By weak duality [18, Proposition 5.1.3], any dual feasible solution provides a lower bound to the primal, i.e.,

$$g(P_e; \varepsilon_n) \leq \widetilde{M}(P_e; \varepsilon_n). \tag{5.54}$$

[10]The operator $\mathrm{vec}(\mathbf{C})$ vectorizes a matrix in a column oriented way. Thus, if $\mathbf{C} \in \mathbb{R}^{l \times l}$, $\mathrm{vec}(\mathbf{C})$ is a length-l^2 vector with the columns of $\mathbf{C}$ stacked one on top of another ($\mathbf{C}(:)$ in Matlab).

[11]The first two terms in the Taylor expansion of the mutual information $I(\mathrm{vec}(Q^*_n))$ in (5.48) vanish because (i) $I(P_e) = 0$ and (ii) $(\mathrm{vec}(Q^*_n) - \mathrm{vec}(P_e))^T \nabla_{\mathrm{vec}(Q)} I(\mathrm{vec}(P_e)) = 0$. Indeed, if we expand $I(\mathrm{vec}(Q))$ around a product distribution, the constant and linear terms vanish [26]. Note that $\mathbf{H}_e$ in (5.50) is an indefinite matrix because $I(\mathrm{vec}(Q))$ is not convex.

Note that strong duality (equality in (5.54)) does not hold in general due in part to the non-convex constraint set in (5.52). Interestingly, our manipulations lead lower bounding $\widetilde{M}$ by (5.53), which is a (convex) semidefinite program [200].

Now observe that the approximations in (5.47) and (5.48) are accurate in the limit of large n because the optimizing distribution Q_n^* becomes increasingly close to P_e. By continuity of the optimization problems in (perturbations of) the objective and the constraints, $\widetilde{M}(P_e;\varepsilon_n)$ and $M(P_e;\varepsilon_n)$ are close when n is large, i.e.,

$$\lim_{n\to\infty} \left|\widetilde{M}(P_e;\varepsilon_n) - M(P_e;\varepsilon_n)\right| = 0. \tag{5.55}$$

By applying the continuity statement above to (5.45), for every $\eta > 0$, there exists a $N \in \mathbb{N}$ such that

$$P^n(I(\widehat{P}_e) \geq \varepsilon_n) \leq (n+1)^{r^2} \exp\left(-n(\widetilde{M}(P_e;\varepsilon_n) - \eta)\right), \tag{5.56}$$

for all $n > N$. Define the constant

$$c_P := \min_{e\in V\times V\,:\, I(P_e)=0} \max_{\mu\geq 0} \{\mu : \mathbf{\Pi}_e \succeq \mu\, \mathbf{H}_e\}. \tag{5.57}$$

By (5.53), (5.54) and the definition of c_P in (5.57),

$$P^n(I(\widehat{P}_e) \geq \varepsilon_n) \leq (n+1)^{r^2} \exp\left(-n\varepsilon_n(c_P - \eta)\right). \tag{5.58}$$

Putting (5.43), (5.44) and (5.58) together, we see that the overestimation error is upper bounded as

$$P^n(\widehat{k}_n > k|\mathcal{B}_n^c) \leq (d-k-1)^2(n+1)^{r^2} \exp\left(-n\varepsilon_n(c_P - \eta)\right). \tag{5.59}$$

Thus, by taking the normalized logarithm (normalized by $n\varepsilon_n$), we have

$$\frac{1}{n\varepsilon_n} \log P^n(\widehat{k}_n > k|\mathcal{B}_n^c) \leq \frac{2}{n\varepsilon_n}\log(d-k-1) + \frac{r^2\log(n+1)}{n\varepsilon_n} - (c_P - \eta). \tag{5.60}$$

Taking the lim sup in n and keeping in mind that $d, k = O(1)$ and $n\varepsilon_n/\log n \to \infty$, we conclude that

$$\limsup_{n\to\infty} \frac{1}{n\varepsilon_n} \log P^n(\widehat{k}_n > k|\mathcal{B}_n^c) \leq -c_P + \eta. \tag{5.61}$$

If we now allow η in (5.61) to tend to 0, we see that it remains to prove that $c_P = 1$ for all P. For this purpose, it suffices to show that the optimal solution to the optimization problem in (5.53), denoted μ^*, is equal to one for all $\mathbf{\Pi}_e$ and $\mathbf{H}_e$. Note that μ^* can be expressed in terms of eigenvalues:

$$\mu^* = \left(\max\left\{\mathrm{eig}(\mathbf{\Pi}_e^{-1/2}\mathbf{H}_e\mathbf{\Pi}_e^{-1/2})\right\}\right)^{-1}, \tag{5.62}$$

where eig$(\cdot)$ denotes the set of real eigenvalues of a symmetric matrix. By using the definitions of $\mathbf{\Pi}_e$ and $\mathbf{H}_e$ in (5.49) and (5.50) respectively, we can verify that the matrix $\mathbf{I} - \mathbf{\Pi}_e^{-1/2}\mathbf{H}_e\mathbf{\Pi}_e^{-1/2}$ is positive semidefinite with an eigenvalue at zero. This proves that the largest eigenvalue of $\mathbf{\Pi}_e^{-1/2}\mathbf{H}_e\mathbf{\Pi}_e^{-1/2}$ is one and hence from (5.62), $\mu^* = 1$. The proof of the upper bound in (5.12) is completed by combining the estimates in (5.30), (5.41) and (5.61).

Proof of lower bound in Theorem 5.3

The key idea is to bound the overestimation error using a modification of the lower bound in Sanov's theorem in (2.52). To prove the lower bound in (5.11), assume that $k < d-1$ and note that the error probability $P^n(\widehat{k}_n \neq k | \mathcal{B}_n^c)$ can be lower bounded by $P^n(I(\widehat{P}_e) \geq \varepsilon_n)$ for any node pair e such that $I(P_e) = 0$. We seek to lower bound the latter probability by appealing to (2.44). Now choose a sequence of n-types $Q^{(n)} \in \text{int}(\{Q \in \mathcal{P}_n(\mathcal{X}^2) : I(Q) \geq \varepsilon_n\})$ such that

$$\lim_{n\to\infty} \left| M(P_e; \varepsilon_n) - D(Q^{(n)} \,||\, P_e) \right| = 0. \tag{5.63}$$

This is possible because the set of types is dense in the probability simplex (see Lemma 2.16). Thus,

$$\begin{aligned} P^n(I(\widehat{P}_e) \geq \varepsilon_n) &= \sum_{Q \in \mathcal{P}_n(\mathcal{X}^2) : I(Q) \geq \varepsilon_n} P^n(\mathsf{T}_n(Q)) && (5.64)\\ &\geq P^n(\mathsf{T}_n(Q^{(n)})) && (5.65)\\ &\geq (n+1)^{-r^2} \exp(-nD(Q^{(n)} \,||\, P_e)), && (5.66) \end{aligned}$$

where (5.66) follows from the lower bound in (2.44). By applying (5.55), and using the fact that if $|a_n - b_n| \to 0$ and $|b_n - c_n| \to 0$ then, $|a_n - c_n| \to 0$, we also have

$$\lim_{n\to\infty} \left| \widetilde{M}(P_e; \varepsilon_n) - D(Q^{(n)} \,||\, P_e) \right| = 0. \tag{5.67}$$

Hence, continuing the chain in (5.66), for any $\eta > 0$, there exists a $N \in \mathbb{N}$ such that for all $n > N$,

$$P^n(I(\widehat{P}_e) \geq \varepsilon_n) \geq (n+1)^{-r^2} \exp(-n(\widetilde{M}(P_e; \varepsilon_n) + \eta)). \tag{5.68}$$

Note that an upper bound for $\widetilde{M}(P_e; \varepsilon_n)$ in (5.52) is simply given by the objective evaluated at any feasible point. In fact, by manipulating (5.52), we see that the upper bound is also proportional to ε_n, i.e.,

$$\widetilde{M}(P_e; \varepsilon_n) \leq C_{P_e} \varepsilon_n, \tag{5.69}$$

where $C_{P_e} \in (0, \infty)$ is some constant[12] that depends on the matrices $\mathbf{\Pi}_e$ and $\mathbf{H}_e$. Define $C_P := \max_{e \in V \times V : I(P_e) = 0} C_{P_e}$. Continuing the lower bound in (5.68), we obtain

$$P^n(I(\widehat{P}_e) \geq \varepsilon_n) \geq (n+1)^{-r^2} \exp(-n\varepsilon_n(C_P + \eta)), \tag{5.70}$$

for n sufficiently large. Now take the normalized logarithm and the lim inf to conclude that

$$\liminf_{n \to \infty} \frac{1}{n\varepsilon_n} \log P^n(\widehat{k}_n \neq k | \mathcal{B}_n^c) \geq -(C_P + \eta). \tag{5.71}$$

Substituting (5.71) into the lower bound in (5.29) and taking $\eta \to 0$ completes proof of the lower bound in Theorem 5.3.

Proof of the exponential decay rate for trees in Theorem 5.3

For the claim in (5.13), note that for n sufficiently large,

$$P^n(\mathcal{A}_n) \geq \max\{(1-\eta) P^n(\widehat{k}_n \neq k_n | \mathcal{B}_n^c), P^n(\mathcal{B}_n)\}, \tag{5.72}$$

from Lemma 5.11 and the fact that $\mathcal{B}_n \subseteq \mathcal{A}_n$. If $k = d - 1$, the overestimation error probability is identically zero. Furthermore, from (5.41) and a corresponding lower bound which we omit, the underestimation error event satisfies $P^n(\widehat{k}_n < k | \mathcal{B}_n^c) \doteq \exp(-nL_P)$. Combining this fact with the definition of the error exponent K_P in (5.28) and the result in (5.72) establishes (5.13). □

Proof. (*of Lemma 5.11*) We note that $P^n(\mathcal{A}_n | \widehat{k}_n \neq k) = 1$ and thus,

$$P^n(\mathcal{A}_n) \leq P^n(\widehat{k}_n \neq k) + P^n(\mathcal{A}_n | \widehat{k}_n = k). \tag{5.73}$$

By using the definition of K_P in (5.28), the second term in (5.73) is precisely $P^n(\mathcal{B}_n)$ therefore,

$$P^n(\mathcal{A}_n) \leq P^n(\widehat{k}_n \neq k) + \exp(-n(K_P - \eta)), \tag{5.74}$$

for all $n > N_1$. We further bound $P^n(\widehat{k}_n \neq k)$ by conditioning on the event $\mathcal{B}_n^c$. Thus, for $\eta > 0$,

$$P^n(\widehat{k}_n \neq k) \leq P^n(\widehat{k}_n \neq k | \mathcal{B}_n^c) + P^n(\mathcal{B}_n) \tag{5.75}$$

$$\leq P^n(\widehat{k}_n \neq k | \mathcal{B}_n^c) + \exp(-n(K_P - \eta)), \tag{5.76}$$

for all $n > N_2$. The upper bound result follows by combining (5.74) and (5.76). The lower bound follows by the chain

$$P^n(\mathcal{A}_n) \geq P^n(\widehat{k}_n \neq k) \geq P^n(\{\widehat{k}_n \neq k\} \cap \mathcal{B}_n^c) \tag{5.77}$$

$$= P^n(\widehat{k}_n \neq k | \mathcal{B}_n^c) P^n(\mathcal{B}_n^c) \geq (1-\eta) P^n(\widehat{k}_n \neq k | \mathcal{B}_n^c), \tag{5.78}$$

which holds for all $n > N_3$ since $P^n(\mathcal{B}_n^c) \to 1$. Now the claims in (5.29) and (5.30) follow by taking $N := \max\{N_1, N_2, N_3\}$. □

[12] We can easily remove the constraint $\mathbf{z}^T \mathbf{1}$ in (5.52) by a simple change of variables to only consider those vectors in the subspace orthogonal to the all ones vector so we ignore it here for simplicity. To obtain C_{P_e}, suppose the matrix $\mathbf{W}_e$ diagonalizes $\mathbf{H}_e$, i.e., $\mathbf{H}_e = \mathbf{W}_e^T \mathbf{D}_e \mathbf{W}_e$, then one can, for example, choose $C_{P_e} = \min_{i : [\mathbf{D}_e]_{i,i} > 0} [\mathbf{W}_e^T \mathbf{\Pi}_e \mathbf{W}_e]_{i,i}$.

■ 5.C Proof of Corollary 5.4

Proof. This claim follows from the fact that three errors (i) Chow-Liu error (ii) underestimation error and (iii) overestimation error behave in exactly the same way as in Theorem 5.3. In particular, the Chow-Liu error, i.e., the error for the learning the top k edges in the forest projection model $\widetilde{P}$ decays with error exponent K_P. The underestimation error behaves as in (5.41) and the overestimation error as in (5.61). □

■ 5.D Proof of Theorem 5.5

Proof. Given assumptions (A1) and (A2), we claim that the underestimation exponent $L_{P^{(d)}}$, defined in (5.41), is uniformly bounded away from zero, i.e.,

$$L := \inf_{d\in\mathbb{N}} L_{P^{(d)}} = \inf_{d\in\mathbb{N}} \min_{e\in E_{P^{(d)}}} L(P_e^{(d)}; 0) \tag{5.79}$$

is positive. Before providing a formal proof, we provide a plausible argument to show that this claim is true. Recall the definition of $L(P_e; 0)$ in (5.37). Assuming that the joint $P_e = P_{i,j}$ is close to a product distribution or equivalently if its mutual information $I(P_e)$ is small (which is the worst-case scenario),

$$L(P_e; 0) \approx \min_{Q\in\mathcal{P}(\mathcal{X}^2)} \{D(P_e \,||\, Q) : I(Q) = 0\} \tag{5.80}$$

$$= D(P_e \,||\, P_i\, P_j) = I(P_e) \geq I_{\text{inf}} > 0, \tag{5.81}$$

where in (5.80), the arguments in the KL-divergence have been swapped. This is because when $Q \approx P_e$ entry-wise, $D(Q \,||\, P_e) \approx D(P_e \,||\, Q)$ in the sense that their difference is small compared to their absolute values [26]. In (5.81), we used the fact that the reverse I-projection of P_e onto the set of product distributions is P_iP_j. Since I_{inf} is constant, this proves the claim, i.e., $L > 0$.

More formally, let $B_{\kappa'} := \{Q_{i,j} \in \mathcal{P}(\mathcal{X}^2) : Q_{i,j}(x_i, x_j) \geq \kappa', \forall\, x_i, x_j \in \mathcal{X}\}$ be the set of joint distributions whose entries are bounded away from zero by $\kappa' > 0$. Now, consider a pair of joint distributions $P_e^{(d)}, \widetilde{P}_e^{(d)} \in B_{\kappa'}$ whose minimum values are uniformly bounded away from zero as assumed in (A2). Then there exists a constant (independent of d) $U \in (0, \infty)$ such that for all d,

$$|I(P_e^{(d)}) - I(\widetilde{P}_e^{(d)})| \leq U \|\text{vec}(P_e^{(d)}) - \text{vec}(\widetilde{P}_e^{(d)})\|_1, \tag{5.82}$$

where $\|\cdot\|_1$ is the vector ℓ_1 norm. In fact,

$$U := \max_{Q\in B_{\kappa'}} \|\nabla I(\text{vec}(Q))\|_\infty \tag{5.83}$$

is the Lipschitz constant of $I(\cdot)$ which is uniformly bounded because the joints $P_e^{(d)}$ and $\widetilde{P}_e^{(d)}$ are assumed to be uniformly bounded away from zero.

Suppose, to the contrary, $L = 0$. Then by the definition of the infimum in (5.79), for every $\epsilon > 0$, there exists a $d \in \mathbb{N}$ and a corresponding $e \in E_{P^{(d)}}$ such that if Q^* is the optimizer in (5.37),

$$\begin{aligned}\epsilon > D(Q^* \,||\, P_e^{(d)}) &\overset{(a)}{\geq} \frac{\|\mathrm{vec}(P_e^{(d)}) - \mathrm{vec}(Q^*)\|_1^2}{2\log 2} \\ &\overset{(b)}{\geq} \frac{|I(P_e^{(d)}) - I(Q^*)|^2}{(2\log 2)U^2} \overset{(c)}{\geq} \frac{I_{\inf}^2}{(2\log 2)U^2},\end{aligned} \tag{5.84}$$

where (a) follows from Pinsker's inequality (see (2.22) or [47, Lemma 11.6.1]), (b) is an application of (5.82) and the fact that if $P_e^{(d)} \in B_\kappa$ is uniformly bounded from zero (as assumed in (5.17)) so is the associated optimizer Q^* (i.e., in $B_{\kappa'}$ for some possibly different uniform $\kappa' > 0$). Statement (c) follows from the definition of $I_{\inf}$ and the fact that Q^* is a product distribution, i.e., $I(Q^*) = 0$. Since ϵ can be chosen to be arbitrarily small and the rightmost quantity in (5.84) is finite, we arrive at a contradiction. Thus L in (5.79) is positive. Finally, we observe from (5.39) that if $n > (2/L)\log k$ the underestimation error tends to zero. Take $C_2 = 2/L$ in (5.18).

Similarly, given the same assumptions, the error exponent for structure learning $K_{P^{(d)}}$, defined in (5.28), is also uniformly bounded away from zero, i.e.,

$$K := \inf_{d\in\mathbb{N}} K_{P^{(d)}} > 0. \tag{5.85}$$

Thus, according to the proof of Theorem 3.4 in Chapter 3 (see (3.75) – (3.80)) if $n > (3/K)\log d$, the error probability associated to estimating the top k edges (event $\mathcal{B}_n$) decays to zero. Take $C_1 = 3/K$ in (5.18).

Finally, from (5.59), if $n\varepsilon_n > 2\log(d-k)$, then the overestimation error tends to zero. Since from (5.10), ε_n can take the form $n^{-\beta}$ for $\beta > 0$, this is equivalent to $n^{1-\beta} > 2\log(d-k)$, which is the same as the first condition in (5.18), namely $n > (2\log(d-k))^{1+\zeta}$. By (5.30) and (5.31), these three probabilities constitute the overall error probability when learning the sequence of forest structures $\{E_{P^{(d)}}\}_{d\in\mathbb{N}}$. Thus the conditions in (5.18) suffice for high-dimensional consistency. □

■ 5.E Proof of Corollary 5.6

Proof. First note that $k_n \in \{0, \ldots, d_n - 1\}$. From (5.60), we see that for n sufficiently large, the sequence $h_n(P) := (n\varepsilon_n)^{-1}\log P^n(\mathcal{A}_n)$ is upper bounded by

$$-1 + \frac{2}{n\varepsilon_n}\log(d_n - k_n - 1) + \frac{r^2\log(n+1)}{n\varepsilon_n}. \tag{5.86}$$

The last term in (5.86) tends to zero by (5.10). Thus $h_n(P) = O((n\varepsilon_n)^{-1}\log(d_n - k_n - 1))$. Clearly, this sequence is maximized (resp. minimized) when $k_n = 0$ (resp. $k_n = d_n - 1$). Eq. (5.86) also shows that the sequence h_n is monotonically decreasing in k_n. □

■ 5.F Proof of Theorem 5.7

Proof. We first focus on part (a). Part (b) follows in a relatively straightforward manner. Define

$$\widehat{T}_{\mathrm{MAP}}(\mathbf{x}^n) := \operatorname*{argmax}_{t \in \mathcal{T}_k^d} \ \mathbb{P}(T_P = t | \mathbf{x}^n) \tag{5.87}$$

to be the maximum a-posteriori (MAP) decoding rule. By the optimality of the MAP rule, this bounds the error probability of any estimator. Let $\mathcal{W} := \widehat{T}_{\mathrm{MAP}}((\mathcal{X}^d)^n)$ be the range of the function $\widehat{T}_{\mathrm{MAP}}$. Note that $\mathcal{W} \cup \mathcal{W}^c = \mathcal{T}_k^d$. Then, we have

$$\begin{aligned}
\mathbb{P}(\widehat{T} \neq T_P) &= \sum_{t \in \mathcal{T}_k^d} \mathbb{P}(\widehat{T} \neq T_P | T_P = t) \mathbb{P}(T_P = t) && (5.88)\\
&\geq \sum_{t \in \mathcal{W}^c} \mathbb{P}(\widehat{T} \neq T_P | T_P = t) \mathbb{P}(T_P = t) && (5.89)\\
&= \sum_{t \in \mathcal{W}^c} \mathbb{P}(T_P = t) = 1 - \sum_{t \in \mathcal{W}} \mathbb{P}(T_P = t) && (5.90)\\
&= 1 - \sum_{t \in \mathcal{W}} |\mathcal{T}_k^d|^{-1} && (5.91)\\
&\geq 1 - r^{nd} |\mathcal{T}_k^d|^{-1}, && (5.92)
\end{aligned}$$

where in (5.90), we used the fact that $\mathbb{P}(\widehat{T} \neq T_P | T_P = t) = 1$ if $t \in \mathcal{W}^c$, in (5.91), the fact that $\mathbb{P}(T_P = t) = 1/|\mathcal{T}_k^d|$. In (5.92), we used the observation $|\mathcal{W}| \leq (|\mathcal{X}^d|)^n = r^{nd}$ since the function $\widehat{T}_{\mathrm{MAP}} : (\mathcal{X}^d)^n \to \mathcal{W}$ is surjective. Now, the number of labeled forests with k edges and d nodes is [3, pp. 204] $|\mathcal{T}_k^d| \geq (d-k)d^{k-1} \geq d^{k-1}$. Applying this lower bound to (5.92), we obtain

$$\mathbb{P}(\widehat{T} \neq T_P) \geq 1 - \exp\left(nd \log r - (k-1) \log d\right) > 1 - \exp\left((\varrho - 1)(k-1) \log d\right), \tag{5.93}$$

where the second inequality follows by choice of n in (5.20). The estimate in (5.93) converges to 1 as $(k, d) \to \infty$ since $\varrho < 1$. The same reasoning applies to part (b) but we instead use the following estimates of the cardinality of the set of forests [3, Ch. 30]:

$$(d-2) \log d \leq \log |\mathcal{F}^d| \leq (d-1) \log(d+1). \tag{5.94}$$

Note that we have lower bounded $|\mathcal{F}^d|$ by the number trees with d nodes which is d^{d-2} by Cayley's formula [3, Ch. 30]. The upper bound[13] follows by a simple combinatorial argument which is omitted. Using the lower bound in (5.94), we have

$$\mathbb{P}(\widehat{T} \neq T_P) \geq 1 - \exp(nd \log r) \exp(-(d-2) \log d) > 1 - d^2 \exp((\varrho - 1) d \log d), \tag{5.95}$$

with the choice of n in (5.21). The estimate in (5.95) converges to 1, completing the proof. □

[13]The purpose of the upper bound is to show that our estimates of $|\mathcal{F}^d|$ in (5.94) are reasonably tight.

■ 5.G Proof of Theorem 5.8

Proof. We assume that P is Markov on a forest since the extension to non-forest-structured P is a straightforward generalization. We start with some useful definitions. Recall from Appendix 5.B that $\mathcal{B}_n := \{\widehat{E}_k \neq E_P\}$ is the event that the top k edges (in terms of mutual information) in the edge set $\widehat{E}_{d-1}$ are not equal to the edges in E_P. Also define $\widetilde{\mathcal{C}}_{n,\delta} := \{D(P^* \,||\, P) > \delta d\}$ to be the event that the divergence between the learned model and the true (forest) one is greater than δd. We will see that $\widetilde{\mathcal{C}}_{n,\delta}$ is closely related to the event of interest $\mathcal{C}_{n,\delta}$ defined in (5.23). Let $\mathcal{U}_n := \{\widehat{k}_n < k\}$ be the underestimation event. Our proof relies on the following result, which is similar to Lemma 5.11, hence its proof is omitted.

Lemma 5.12. *For every $\eta > 0$, there exists a $N \in \mathbb{N}$ such that for all $n > N$, the following bounds on $P^n(\widetilde{\mathcal{C}}_{n,\delta})$ hold:*

$$(1-\eta)P^n(\widetilde{\mathcal{C}}_{n,\delta}|\mathcal{B}_n^c,\mathcal{U}_n^c) \leq P^n(\widetilde{\mathcal{C}}_{n,\delta}) \tag{5.96}$$

$$\leq P^n(\widetilde{\mathcal{C}}_{n,\delta}|\mathcal{B}_n^c,\mathcal{U}_n^c) + \exp(-n(\min\{K_P, L_P\} - \eta)). \tag{5.97}$$

Note that the exponential term in (5.97) comes from an application of the union bound and the "largest-exponent-wins" principle in large-deviations theory. From (5.96) and (5.97) we see that it is possible to bound the probability of $\widetilde{\mathcal{C}}_{n,\delta}$ by providing upper and lower bounds for $P^n(\widetilde{\mathcal{C}}_{n,\delta}|\mathcal{B}_n^c,\mathcal{U}_n^c)$. In particular, we show that the upper bound equals $\exp(-n\delta)$ to first order in the exponent. This will lead directly to (5.24). To proceed, we rely on the following lemma, which is a generalization of a well-known result [47, Ch. 11]. We defer the proof to the end of the section.

Lemma 5.13. (Empirical Divergence Bounds) *Let X, Y be two random variables whose joint distribution is $P_{X,Y} \in \mathcal{P}(\mathcal{X}^2)$ and $|\mathcal{X}| = r$. Let $(x^n, y^n) = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be n independent and identically distributed observations drawn from $P_{X,Y}$. Then, for every n,*

$$P_{X,Y}^n(D(\widehat{P}_{X|Y} \,||\, P_{X|Y}) > \delta) \leq (n+1)^{r^2} \exp(-n\delta), \tag{5.98}$$

where $\widehat{P}_{X|Y} = \widehat{P}_{X,Y}/\widehat{P}_Y$ is the conditional type of (x^n, y^n). Furthermore,

$$\liminf_{n\to\infty} \frac{1}{n} \log P_{X,Y}^n(D(\widehat{P}_{X|Y} \,||\, P_{X|Y}) > \delta) \geq -\delta. \tag{5.99}$$

It is worth noting that the bounds in (5.98) and (5.99) are independent of the distribution $P_{X,Y}$ (cf. discussion after Theorem 5.8). We now proceed with the proof of Theorem 5.8. To do so, we consider the directed representation of a tree distribution Q [127]:

$$Q(\mathbf{x}) = \prod_{i \in V} Q_{i|\pi(i)}(x_i|x_{\pi(i)}), \tag{5.100}$$

where $\pi(i)$ is the parent of i in the edge set of Q (assuming a fixed root). Using (5.100) and conditioned on the fact that the top k edges of the graph of P^* are the same as those

Figure 5.10. In $\widehat{E}_{\widehat{k}_n}$ (left), nodes 1 and 5 are the roots, which are in blue. The parents are defined as $\pi(i;\widehat{E}_{\widehat{k}_n}) = i-1$ for $i = 2,3,4,6$ and $\pi(i;\widehat{E}_{\widehat{k}_n}) = \emptyset$ for $i = 1,5$. In E_P (right), the parents are defined as $\pi(i;E_P) = i-1$ for $i = 2,3,4$ but $\pi(i;E_P) = \emptyset$ for $i = 1,5,6$ since $(5,6),(\emptyset,1),(\emptyset,5) \notin E_P$.

in E_P (event $\mathcal{B}_n^c$) and underestimation does not occur (event $\mathcal{U}_n^c$), the KL-divergence between P^* (which is a function of the samples $\mathbf{x}^n$ and hence of n) and P can be expressed as a sum over d terms:

$$D(P^* \,||\, P) = \sum_{i\in V} D(\widehat{P}_{i|\pi(i;\widehat{E}_{\widehat{k}_n})} \,||\, P_{i|\pi(i;E_P)}), \tag{5.101}$$

where the parent of node i in $\widehat{E}_{\widehat{k}_n}$, denoted $\pi(i;\widehat{E}_{\widehat{k}_n})$, is defined by arbitrarily choosing a root in each component tree of the forest $\widehat{T}_{\widehat{k}_n} = (V, \widehat{E}_{\widehat{k}_n})$. The parents of the chosen roots are empty sets. The parent of node i in E_P are "matched" to those in $\widehat{E}_{\widehat{k}_n}$, i.e., defined as $\pi(i;E_P) := \pi(i;\widehat{E}_{\widehat{k}_n})$ if $(i,\pi(i;\widehat{E}_{\widehat{k}_n})) \in E_P$ and $\pi(i;E_P) := \emptyset$ otherwise. See Figure 5.10 for an example. Note that this can be done because $\widehat{E}_{\widehat{k}_n} \supseteq E_P$ by conditioning on the events $\mathcal{B}_n^c$ and $\mathcal{U}_n^c = \{\widehat{k}_n \geq k\}$. Then, the error probability in (5.97) can be upper bounded as

$$\begin{aligned}
P^n(\widetilde{\mathcal{C}}_{n,\delta}|\mathcal{B}_n^c,\mathcal{U}_n^c) &= P^n\left(\sum_{i\in V} D(\widehat{P}_{i|\pi(i;\widehat{E}_{\widehat{k}_n})}||P_{i|\pi(i;E_P)}) > \delta d \Big| \mathcal{B}_n^c,\mathcal{U}_n^c\right) && (5.102)\\
&= P^n\left(\frac{1}{d}\sum_{i\in V} D(\widehat{P}_{i|\pi(i;\widehat{E}_{\widehat{k}_n})}||P_{i|\pi(i;E_P)}) > \delta \Big| \mathcal{B}_n^c,\mathcal{U}_n^c\right) && (5.103)\\
&\leq P^n\left(\max_{i\in V}\left\{D(\widehat{P}_{i|\pi(i;\widehat{E}_{\widehat{k}_n})}||P_{i|\pi(i;E_P)})\right\} > \delta \Big| \mathcal{B}_n^c,\mathcal{U}_n^c\right) && (5.104)\\
&\leq \sum_{i\in V} P^n\left(D(\widehat{P}_{i|\pi(i;\widehat{E}_{\widehat{k}_n})}||P_{i|\pi(i;E_P)}) > \delta \Big| \mathcal{B}_n^c,\mathcal{U}_n^c\right) && (5.105)\\
&\leq \sum_{i\in V}(n+1)^{r^2}\exp\left(-n\delta\right) = d(n+1)^{r^2}\exp\left(-n\delta\right), && (5.106)
\end{aligned}$$

where Eq. (5.102) follows from the decomposition in (5.101). Eq. (5.104) follows from the fact that if the arithmetic mean of d positive numbers exceeds δ, then the maximum exceeds δ. Eq. (5.105) follows from the union bound. Eq. (5.106), which holds for all $n \in \mathbb{N}$, follows from the upper bound in (5.98). Combining (5.97) and (5.106) shows that if $\delta < \min\{K_P, L_P\}$,

$$\limsup_{n\to\infty} \frac{1}{n} \log P^n(\widetilde{\mathcal{C}}_{n,\delta}) \leq -\delta. \tag{5.107}$$

Now recall that $\widetilde{\mathcal{C}}_{n,\delta} = \{D(P^* \,||\, P) > \delta d\}$. In order to complete the proof of (5.24), we need to swap the arguments in the KL-divergence to bound the probability of the event $\mathcal{C}_{n,\delta} = \{D(P \,||\, P^*) > \delta d\}$ defined in (5.23). To this end, note that for every $\epsilon > 0$ and n sufficiently large, $|D(P^* \,||\, P) - D(P \,||\, P^*)| < \epsilon$ with high probability. More precisely, the probability of the event $\{|D(P^* \,||\, P) - D(P \,||\, P^*)| \geq \epsilon\}$ decays exponentially with some exponential rate $M_P > 0$. Hence,

$$\limsup_{n\to\infty} \frac{1}{n} \log P^n(D(P \,||\, P^*) > \delta d) \leq -\delta, \tag{5.108}$$

if $\delta < \min\{K_P, L_P, M_P\}$. If P is not Markov on a forest, (5.108) holds with the forest projection $\widetilde{P}$ in place of P, i.e.,

$$\limsup_{n\to\infty} \frac{1}{n} \log P^n(D(\widetilde{P} \,||\, P^*) > \delta d) \leq -\delta. \tag{5.109}$$

The Pythagorean relationship [10, 180] states that

$$D(P \,||\, P^*) = D(P \,||\, \widetilde{P}) + D(\widetilde{P} \,||\, P^*) \tag{5.110}$$

which means that the risk is $\mathcal{R}_n(P^*) = D(\widetilde{P} \,||\, P^*)$. Combining this fact with (5.109) implies the assertion of (5.24) by choosing $\delta_0 := \min\{K_P, L_P, M_P\}$.

Now we exploit the lower bound in Lemma 5.13 to prove the lower bound in Theorem 5.8. The error probability in (5.97) can now be lower bounded as

$$P^n(\widetilde{\mathcal{C}}_{n,\delta} | \mathcal{B}_n^c, \mathcal{U}_n^c) \geq \max_{i\in V} \; P^n\left(D(\widehat{P}_{i|\pi(i;\widehat{E}_{\widehat{k}_n})} \,||\, P_{i|\pi(i;E_P)}) > \delta d \Big| \mathcal{B}_n^c, \mathcal{U}_n^c\right) \tag{5.111}$$

$$\geq \exp(-n(\delta d + \eta)), \tag{5.112}$$

where (5.111) follows from the decomposition in (5.102) and (5.112) holds for every η for sufficiently large n by (5.99). Using the same argument that allows us to swap the arguments of the KL-divergence as in the proof of the upper bound completes the proof of (5.25). □

Proof. (*of Lemma 5.13*) Define the *δ-conditional-typical set with respect to $P_{X,Y} \in \mathcal{P}(\mathcal{X}^2)$* as

$$\mathcal{S}^{\delta}_{P_{X,Y}} := \{(x^n, y^n) \in (\mathcal{X}^2)^n : D(\widehat{P}_{X|Y} \,||\, P_{X|Y}) \leq \delta\}, \tag{5.113}$$

where $\widehat{P}_{X|Y}$ is the conditional type of (x^n, y^n). We now estimate the $P^n_{X,Y}$-probability of the δ-conditional-atypical set, i.e., $P^n_{X,Y}((\mathcal{S}^{\delta}_{P_{X,Y}})^c)$

$$= \sum_{(x^n,y^n)\in\mathcal{X}^2 : D(\hat{P}_{X|Y}||P_{X|Y})>\delta} P^n_{X,Y}((x^n, y^n)) \tag{5.114}$$

$$= \sum_{Q_{X,Y}\in\mathcal{P}_n(\mathcal{X}^2) : D(Q_{X|Y}||P_{X|Y})>\delta} P^n_{X,Y}(\mathsf{T}_n(Q_{X,Y})) \tag{5.115}$$

$$\leq \sum_{Q_{X,Y}\in\mathcal{P}_n(\mathcal{X}^2):D(Q_{X|Y}||P_{X|Y})>\delta} \exp(-nD(Q_{X,Y}\,||\,P_{X,Y})) \tag{5.116}$$

$$\leq \sum_{Q_{X,Y}\in\mathcal{P}_n(\mathcal{X}^2):D(Q_{X|Y}||P_{X|Y})>\delta} \exp(-nD(Q_{X|Y}\,||\,P_{X|Y})) \tag{5.117}$$

$$\leq \sum_{Q_{X,Y}\in\mathcal{P}_n(\mathcal{X}^2):D(Q_{X|Y}||P_{X|Y})>\delta} \exp(-n\delta) \tag{5.118}$$

$$\leq (n+1)^{r^2}\exp(-n\delta), \tag{5.119}$$

where (5.114) and (5.115) are the same because summing over sequences is equivalent to summing over the corresponding type classes. Eq. (5.116) follows from the method of types result in Lemma 2.18. Eq. (5.117) follows from the KL-divergence version of the chain rule, namely, $D(Q_{X,Y}\,||\,P_{X,Y}) = D(Q_{X|Y}\,||\,P_{X|Y}) + D(Q_Y\,||\,P_Y)$ and non-negativity of the KL-divergence $D(Q_Y\,||\,P_Y)$. Eq. (5.118) follows from the fact that $D(Q_{X|Y}\,||\,P_{X|Y}) > \delta$ for $Q_{X,Y} \in (\mathcal{S}^{\delta}_{P_{X,Y}})^c$. Finally, (5.119) follows the fact that the number of types with denominator n and alphabet $\mathcal{X}^2$ is upper bounded by $(n+1)^{r^2}$. This concludes the proof of (5.98).

We now prove the lower bound in (5.99). To this end, construct a sequence of n-types $\{Q^{(n)}_{X,Y} \in \mathcal{P}_n(\mathcal{X}^2)\}_{n\in\mathbb{N}}$ such that $Q^{(n)}_Y = P_Y$ and $D(Q^{(n)}_{X|Y}||P_{X|Y}) \to \delta$. Such a sequence exists by the denseness of types in the probability simplex (see Lemma 2.16). Now we lower bound (5.115):

$$P^n_{X,Y}((\mathcal{S}^{\delta}_{P_{X,Y}})^c) \geq P^n_{X,Y}(\mathsf{T}_n(Q^{(n)}_{X,Y})) \geq (n+1)^{-r^2}\exp(-nD(Q^{(n)}_{X,Y}\,||\,P_{X,Y})). \tag{5.120}$$

Taking the normalized logarithm and $\liminf$ in n on both sides of (5.120) yields

$$\liminf_{n\to\infty}\frac{1}{n}\log P^n_{X,Y}((\mathcal{S}^{\delta}_{P_{X,Y}})^c) \geq \liminf_{n\to\infty}\left\{-D(Q^{(n)}_{X|Y}||P_{X|Y}) - D(Q^{(n)}_Y||P_Y)\right\} = -\delta. \tag{5.121}$$

This concludes the proof of Lemma 5.13. □

■ 5.H Proof of Corollary 5.9

Proof. If the dimension $d = o(\exp(n\delta))$, then the upper bound in (5.106) is asymptotically majorized by $\mathrm{poly}(n)o(\exp(na))\exp(-n\delta) = o(\exp(n\delta))\exp(-n\delta)$, which can be made arbitrarily small for n sufficiently large. Thus the probability tends to zero as $n \to \infty$. □

■ 5.I Proof of Theorem 5.10

Proof. In this proof, we drop the superscript (d) for all distributions P for notational simplicity but note that $d = d_n$. We first claim that $D(P^*\,||\,\widetilde{P}) = O_p(d\log d/n^{1-\gamma})$.

Note from (5.97) and (5.106) that by taking $\delta = (t \log d)/n^{1-\gamma}$ (for any $t > 0$),

$$P^n \left(\frac{n^{1-\gamma}}{d \log d} D(P^* \,||\, \widetilde{P}) > t \right) \leq d(n+1)^{r^2} \exp(-tn^\gamma \log d) + \exp(-\Theta(n)) \tag{5.122}$$

$$= o_n(1). \tag{5.123}$$

Therefore, the scaled sequence of random variables $\frac{n^{1-\gamma}}{d \log d} D(P^* \,||\, \widetilde{P})$ is stochastically bounded [177] which proves the claim.[14]

Now, we claim that $D(\widetilde{P} \,||\, P^*) = O_p(d \log d / n^{1-\gamma})$. A simple calculation using Pinsker's Inequality and Lemma 6.3 in [54] yields

$$D(\widehat{P}_{X,Y} \,||\, P_{X,Y}) \leq \frac{c}{\kappa} D(P_{X,Y} \,||\, \widehat{P}_{X,Y}), \tag{5.124}$$

where $\kappa := \min_{x,y} P_{X,Y}(x,y)$ and $c = 2 \log 2$. Eq. (5.124) quantifies the variation of the KL-divergence in terms of κ and its flipped version and its proof can be found in [185]. Using this fact, we can use (5.98) to show that for all n sufficiently large,

$$P^n_{X,Y}(D(P_{X|Y} \,||\, \widehat{P}_{X|Y}) > \delta) \leq (n+1)^{r^2} \exp(-n\delta\kappa/c), \tag{5.125}$$

i.e., if the arguments in the KL-divergence in (5.98) are swapped, then the exponent is reduced by a factor proportional to κ. Using this fact and the assumption in (5.17) (uniformity of the minimum entry in the pairwise joint $\kappa > 0$), we can replicate the proof of the result in (5.106) with $\delta\kappa/c$ in place of δ giving

$$P^n(D(P \,||\, P^*) > \delta) \leq d(n+1)^{r^2} \exp\left(-n\delta\kappa/c\right). \tag{5.126}$$

We then arrive at a similar result to (5.123) by taking $\delta = (t \log d)/n^{1-\gamma}$. We conclude that $D(\widetilde{P} \,||\, P^*) = O_p(d \log d / n^{1-\gamma})$. This completes the proof of the claim.

Eq. (5.26) then follows from the definition of the risk in (5.22) and from the Pythagorean theorem in (5.110). This implies the assertion of Theorem 5.10. □

[14] In fact, we have in fact proven the stronger assertion that $D(P^* \,||\, \widetilde{P}) = o_p(d \log d / n^{1-\gamma})$ since the right-hand-side of (5.123) converges to zero.

Chapter 6

Learning Graphical Models for Hypothesis Testing

6.1 Introduction

THIS chapter departs from the analysis of data modeling using tree- or forest-structured distributions. Instead we propose techniques to exploit the modeling ability of such sparse graphical models for binary classification (see Section 2.3) by discriminatively learning such models from labeled training data. The generative techniques to learn such models (such as in [2, 128, 136, 211]) are not straightforward to adapt for the purpose of binary classification (or binary hypothesis testing). As an example, for two distributions p and q that are "close" to each other; separately modeling each by a sparse graphical model would likely "blur" the differences between the two. This is because the goal of modeling is to faithfully capture the entire behavior of a single distribution, and *not* to emphasize its most salient *differences* from another probability distribution. Our motivation is to retain the generalization power of sparse graphical models, while also developing a procedure that automatically identifies and emphasizes features that help to best discriminate between two distributions.

We leverage the modeling flexibility of sparse graphical models for the task of classification: given labeled training data from two unknown distributions, we first describe how to build a pair of tree-structured graphical models to better *discriminate* between the two distributions. In addition, we also utilize ideas from boosting [173] to learn a richer (or larger) set of features[1] using the previously mentioned tree-learning algorithm as the weak classifier. This allows us to learn thicker graphical models.

There are three main contributions in this chapter: Firstly, it is known that decreasing functions of the J-divergence (a symmetric form of the KL-divergence) provide upper and lower bounds to the error probability [14, 100, 111]. Motivated by these bounds, we develop efficient algorithms to maximize a tree-based approximation to the J-divergence. We show that it is straightforward to adapt the generative tree-learning

[1] In this chapter, we use the generic term *features* to denote the marginal as well as pairwise relations between random variables, i.e., the marginals $p_i(x_i), q_i(x_i)$ and the pairwise joints $p_{i,j}(x_i, x_j), q_{i,j}(x_i, x_j)$.

procedure of Chow and Liu (described in Section 2.5.2) to a *discriminative*[2] objective related to the J-divergence over tree models. Secondly, we propose a boosting-based procedure (see Section 2.3.2) to learn a richer set of features, thus improving the modeling ability of the learned distributions $\widehat{p}$ and $\widehat{q}$. Finally, we demonstrate empirically that this family of algorithms lead to accurate classification on a wide range of synthetic and real-world datasets.

A basic form of learning of graphical models for classification is the so-called Naïve Bayes model, which corresponds to the graphs of the distributions p and q having no edges, a restrictive assumption. A comprehensive study of discriminative vs generative Naïve Bayes was done in Ng et al. [148]. Friedman et al. [84] and Wang and Wong [212] suggested an improvement to Naïve Bayes using a generative model known as TAN, a specific form of a graphical model geared towards classification. However, the p and q models learned in these papers share the same structure and hence are more restrictive than the proposed discriminative algorithm, which learns trees with possibly distinct structures for each hypothesis.

More recently, Grossman and Domingos [88] improved on TAN by proposing an algorithm for choosing the structures by greedily maximizing the conditional log-likelihood (CLL) with a minimum description length (MDL) penalty while setting parameters by maximum-likelihood and obtained good classification results on benchmark datasets. However, estimating the model parameters via maximum-likelihood is complicated because the learned structures are loopy. Su and Zhang [183] suggested representing variable independencies by conditional probability tables (CPT) instead of the structures of graphical models. Boosting has been used in Rosset and Segal [163] for density estimation and learning Bayesian networks, but the objective was on modeling and not on classification. In Jing et al. [105], the authors suggested boosting the parameters of TANs. Our procedure uses boosting to optimize for both the structures and the parameters of the pair of discriminative tree models, thus enabling the learning of thicker structures.

The rest of the chapter is organized as follows: In Section 6.2, we present some mathematical preliminaries specific to this chapter. In Section 6.3, we present a discriminative tree learning algorithm specifically tailored for the purpose of classification. This is followed by the presentation of a novel adaptation of Real-AdaBoost [82, 175] to learn a larger set of features in Section 6.4. In Section 6.5, we present numerical experiments to validate the learning method presented in Sections 6.3 and 6.4 and also demonstrate how the method can be naturally extended to multi-class classification problems. We conclude in Section 6.6 by discussing the merits of the techniques presented.

[2]In this chapter, we adopt the term "discriminative" to denote the use of both the positively and negatively labeled training samples to learn the model $\widehat{p}$, the approximate model for the positively labeled samples (and similarly for $\widehat{q}$). This is different from "generative" learning in which only the positively labeled samples are used to estimate $\widehat{p}$ (and similarly for $\widehat{q}$).

6.2 Preliminaries and Notation

6.2.1 Binary Classification

We have already introduced the setup for binary classification in Section 2.3. Here, we remind the reader of the notation for completeness. We are given a labeled training set $\mathcal{S} := \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, where each training pair $(\mathbf{x}_l, y_l) \in \mathcal{X}^d \times \{+1, -1\}$. Here, $\mathcal{X}$ may be a finite set (e.g., $\mathcal{X} = \{0, \cdots, r-1\}$) or an infinite set (e.g., $\mathcal{X} = \mathbb{R}$). Each y_l, which can only take on one of two values, represents the *class label* of that sample. Each training pair (x_l, y_l) is drawn independently from some unknown joint distribution $P_{\mathbf{X},Y}$. In this chapter, we adopt the following simplifying notation: $p(\mathbf{x}) := P_{\mathbf{X}|Y}(\mathbf{x}|y = 1)$ and $q(\mathbf{x}) := P_{\mathbf{X}|Y}(\mathbf{x}|y = -1)$ are the class-conditional distributions. Also, we assume the a-priori probabilities for the label are uniform, i.e., $P_Y(y = +1) = P_Y(y = -1) = 1/2$. This is not a restrictive assumption and we make it to lighten the notation.

Given $\mathcal{S}$, we wish to train a model so as to classify, i.e., to assign a label of $+1$ or -1 to a new sample $\mathbf{x}$. This sample is drawn according to the unknown distribution $P_{\mathbf{X}}$, but its label is unavailable. If we do have access to the true distributions p and q, the optimal test under both the Neyman-Pearson and Bayesian settings (Lemmas 2.20 and 2.21) is known to be the log-likelihood ratio test given by

$$\log \varphi(\mathbf{x}) \underset{\hat{y}=-1}{\overset{\hat{y}=+1}{\gtrless}} \eta, \tag{6.1}$$

where the *likelihood ratio* $\varphi : \mathcal{X}^d \to \mathbb{R}^+$ is the ratio of the class-conditional distributions p and q, i.e.,

$$\varphi(\mathbf{x}) := \frac{p(\mathbf{x})}{q(\mathbf{x})}. \tag{6.2}$$

In (6.1), $\eta \in \mathbb{R}$ is the *threshold* of the test. In the absence of fully specified p and q, we will instead develop efficient algorithms for constructing approximations $\widehat{p}$ and $\widehat{q}$ from the set of samples $\mathcal{S}$ such that the following statistic (for approximating $\varphi(\mathbf{x})$) is as discriminative as possible.

$$\log \widehat{\varphi}(\mathbf{x}) \underset{\hat{y}=-1}{\overset{\hat{y}=+1}{\gtrless}} \eta, \tag{6.3}$$

where $\widehat{\varphi} : \mathcal{X}^d \to \mathbb{R}^+$ is an approximation of the likelihood ratio, defined as

$$\widehat{\varphi}(\mathbf{x}) := \frac{\widehat{p}(\mathbf{x})}{\widehat{q}(\mathbf{x})}. \tag{6.4}$$

In (6.4), $\widehat{p}$ and $\widehat{q}$ are multivariate distributions (or graphical models) estimated *jointly* from both the positively and negatively labeled samples in the training set $\mathcal{S}$. We use the empirical distribution formed from samples, $\widetilde{p}$ and $\widetilde{q}$, to estimate $\widehat{p}$ and $\widehat{q}$, which are then used in the approximate likelihood ratio test in (6.4).

■ 6.2.2 The J-divergence

The *J-divergence* between two probability distributions p and q is defined as [124]

$$J(p,q) := D(p\,||\,q) + D(q\,||\,p) \tag{6.5}$$

and is a fundamental measure of the separability of distributions. It has the property that $J = 0$ if and only if $p = q$ almost everywhere. In contrast to KL-divergence, J is symmetric in its arguments. However, it is still not a metric as it does not satisfy the triangle inequality. Nevertheless, the following useful upper and lower bounds on the probability of error $\Pr(\text{err})$ [14, 100, 111] can be obtained from the J-divergence between two distributions.

$$\frac{1}{2}\min(P_{-1}, P_1)\exp(-J) \leq \Pr(\text{err}) \leq \sqrt{P_{-1}\,P_1}\left(\frac{J}{4}\right)^{-1/4}, \tag{6.6}$$

where $P_j := P_Y(y = j)$ are the prior probabilities. Thus, maximizing J minimizes both upper and lower bounds on the $\Pr(\text{err})$. Motivated by the fact that increasing the J-divergence decreases the upper and lower bounds in (6.6), we find $\widehat{\varphi}(x)$ in (6.4) by choosing graphical models $\widehat{p}$ and $\widehat{q}$ which maximize an approximation to the J-divergence.

■ 6.3 Discriminative Learning of Trees and Forests

In this section, we propose efficient discriminative algorithms for learning two tree models by optimizing a surrogate statistic for J-divergence. We show that this is equivalent to optimizing the empirical log-likelihood ratio of the training samples. We then discuss how to optimize the objective by using MWST-based algorithms. Before doing so, we define the following constraint on the parameters of the learned models $\widehat{p}$ and $\widehat{q}$, which are assumed to be Markov on trees (or forests) with edge sets $E_{\widehat{p}}$ and $E_{\widehat{q}}$ respectively.

Definition 6.1. *(Marginal Consistency) The approximating distributions $\widehat{p}$ and $\widehat{q}$ are said to be* marginally consistent *with respect to the distributions p and q if their pairwise marginals on their respective edge sets $E_{\widehat{p}}$ and $E_{\widehat{q}}$ are equal, i.e., for the model $\widehat{p}$, we have*

$$\widehat{p}_{i,j}\,(x_i, x_j) = p_{i,j}\,(x_i, x_j)\,, \quad \forall\,(i,j) \in E_{\widehat{p}}. \tag{6.7}$$

It follows from (6.7) *that $\widehat{p}_i(x_i) = p_i(x_i)$ for all nodes $i \in V$.*

We will subsequently see that if $\widehat{p}$ and $\widehat{q}$ are marginally consistent, this yields tractable optimizations for the search for the optimal structures of $\widehat{p}$ and $\widehat{q}$. Now, one naïve choice of $\widehat{p}$ and $\widehat{q}$ to approximate the log-likelihood ratio in (6.2) is to construct generative tree or forest models of p and q from the samples, i.e., learn[3] $\widehat{p} \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$

[3]Recall that the notation $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$ denotes the set of distributions in $\mathcal{P}(\mathcal{X}^d)$ which are Markov on some d-node tree in $\mathcal{T}^d$. See Section 2.4.3.

from the positively labeled samples and $\widehat{q}$ from the negatively labeled samples using the Chow-Liu method detailed in Section 2.5.2. The set of generative models under consideration can be from the set of trees $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$ or the set of k-edge forests $\mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d)$. Kruskal's MWST algorithm [120] can be employed in either case. If we do have access to the true distributions, then this process is simply fitting lower-order tree (or forest) approximations to p and q. However, the true distributions p and q are usually not available. Motivated by Hoeffding and Wolfowitz [100] (who provide guarantees when optimizing the likelihood ratio test), and keeping in mind the final objective which is classification, we design $\widehat{p}$ and $\widehat{q}$ in a *discriminative* fashion to obtain $\widehat{\varphi}(\mathbf{x})$, defined in (6.4).

■ 6.3.1 The Tree-Approximate J-divergence

We now formally define the approximation to the J-divergence originally given in (6.5).

Definition 6.2. *(Tree-approximate J-divergence) The* tree-approximate J-divergence $\widehat{J}(\widehat{p}, \widehat{q}; p, q)$ *of two tree-structured distributions $\widehat{p}$ and $\widehat{q}$ with respect to two arbitrary distributions p and q is defined as:*

$$\widehat{J}(\widehat{p}, \widehat{q}; p, q) := \int_{\Omega} (p(\mathbf{x}) - q(\mathbf{x})) \log \left[\frac{\widehat{p}(\mathbf{x})}{\widehat{q}(\mathbf{x})} \right] \, d\mathbf{x}, \tag{6.8}$$

for distributions that are mutually absolutely continuous[4] *and*

$$\widehat{J}(\widehat{p}, \widehat{q}; p, q) := \sum_{\mathbf{x} \in \mathcal{X}^d} (p(\mathbf{x}) - q(\mathbf{x})) \log \left[\frac{\widehat{p}(\mathbf{x})}{\widehat{q}(\mathbf{x})} \right], \tag{6.9}$$

for discrete distributions.

Observe that the difference between J and $\widehat{J}$ is the replacement of the true distributions p and q by the approximate distributions $\widehat{p}$ and $\widehat{q}$ in the logarithm. As we see in Proposition 6.4, maximizing the tree-approximate J-divergence over $\widehat{p}$ and $\widehat{q}$ is equivalent to maximizing the *empirical log-likelihood ratio* if the random variables are discrete. Note however, that the objectives in (6.8) and (6.9) do not necessarily share the properties of the true J-divergence in (6.6). The relationship between (6.8) (and (6.9)) and the J-divergence requires further theoretical analysis but this is beyond the scope of the chapter. We demonstrate empirically that the maximization of the tree-approximate J-divergence results in good discriminative performance in Section 6.5.

There are several other reasons for maximizing the tree-approximate J-divergence. Firstly, trees have proven to be a rich class of distributions for modeling high-dimensional data (see [10] and examples in Section 5.7). Secondly, as is demonstrated in the sequel,

[4]Two distributions p and q (for $p \neq q$) are *mutually absolutely continuous* if the corresponding *measures* ν_p and ν_q are absolutely continuous with respect to each other (or are *equivalent* measures) [123, Ch. 7]. The integral in (6.8) is understood to be over the domain in which the measures are equivalent $\Omega \subset \mathcal{X}^n$.

we are able to develop efficient learning algorithms for finding $\widehat{p}$ and $\widehat{q}$. We now state a useful property of the tree-approximate J-divergence assuming $\widehat{p}$ and $\widehat{q}$ are trees.

Proposition 6.1. *(Decomposition of tree-approximate J-divergence) Assume that (i) the pairwise marginals $p_{i,j}$ and $q_{i,j}$ in (6.8) are mutually absolutely continuous and (ii) $\widehat{p}$ and $\widehat{q}$ are tree distributions with edge sets $E_{\widehat{p}}$ and $E_{\widehat{q}}$ respectively and are also marginally consistent with p and q. Then the tree-approximate J-divergence can be expressed as a sum of marginal J divergences and weights:*

$$\widehat{J}(\widehat{p},\widehat{q};p,q)=\sum_{i\in V}J(p_i,q_i)+\sum_{(i,j)\in E_{\widehat{p}}\cup E_{\widehat{q}}}w_{ij}. \tag{6.10}$$

The multi-valued edge weights w_{ij} are given by

$$w_{ij}:=\begin{cases} I_p\left(X_i;X_j\right)-I_q\left(X_i;X_j\right) & \\ \quad +D\left(q_{i,j}||p_{i,j}\right)-D\left(q_iq_j||p_ip_j\right) & (i,j)\in E_{\widehat{p}}\setminus E_{\widehat{q}} \\ I_q\left(X_i;X_j\right)-I_p\left(X_i;X_j\right) & \\ \quad +D\left(p_{i,j}||q_{i,j}\right)-D\left(p_ip_j||q_iq_j\right) & (i,j)\in E_{\widehat{q}}\setminus E_{\widehat{p}} \\ J(p_{i,j},q_{i,j})-J\left(p_ip_j,q_iq_j\right) & (i,j)\in E_{\widehat{p}}\cap E_{\widehat{q}} \end{cases} \tag{6.11}$$

where $I_p\left(X_i;X_j\right)$ and $I_q\left(X_i;X_j\right)$ denote the mutual information between variables X_i and X_j under the p and q probability models respectively.

Proof. Since $\widehat{p}$ is a tree-structured distribution, it admits the factorization as in (2.95) with the node and pairwise marginals given by p (by marginal consistency). The distribution $\widehat{q}$ has a similar factorization. These factorizations can be substituted into (6.8) or (6.9) and the KL-divergences can then be expanded. Finally, by using the identities

$$\sum_{\mathbf{x}\in\mathcal{X}^d}p\left(\mathbf{x}\right)\log\left[\frac{p_i(x_i)}{q_i(x_i)}\right]=D(p_i||q_i), \tag{6.12}$$

$$\sum_{\mathbf{x}\in\mathcal{X}^d}p\left(\mathbf{x}\right)\log\left[\frac{p_{i,j}(x_i,x_j)}{q_{i,j}(x_i,x_j)}\right]=D(p_{i,j}||q_{i,j}), \tag{6.13}$$

and marginal consistency of $\widehat{p}$ and $\widehat{q}$, we can group terms together and obtain the result in the proposition. □

Denote the empirical distributions of the positive and negatively labeled samples as $\widetilde{p}$ and $\widetilde{q}$ respectively. Given the definition of $\widehat{J}$ in (6.8), the optimization problem for $\widehat{p}$ and $\widehat{q}$ is formally formulated as:

$$(\widehat{p},\widehat{q})=\underset{\widehat{p}\in\mathcal{D}(\mathcal{X}^d,\mathcal{T}^d(\widetilde{p})),\,\widehat{q}\in\mathcal{D}(\mathcal{X}^d,\mathcal{T}^d(\widetilde{q}))}{\operatorname{argmax}}\widehat{J}(\widehat{p},\widehat{q};\widetilde{p},\widetilde{q}), \tag{6.14}$$

where $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d(\widetilde{p})) \subset \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d)$ is the set of tree-structured distributions which are marginally consistent with $\widetilde{p}$ (see Definition 6.1). We will see that this optimization reduces to two tractable MWST problems. Furthermore, as in the Chow-Liu solution to the generative problem (Section 2.5.2), only marginal and pairwise statistics need to be computed from the training set in order to estimate the information quantities in (6.11). In the sequel, we describe how to estimate these statistics and also how to devise efficient MWST algorithms to optimize (6.14) over the set of spanning trees.

■ 6.3.2 Learning Spanning Trees

In this section, we describe an efficient algorithm for learning two trees that optimize the tree-approximate J-divergence defined in (6.8). We assume that we have no access to the true distributions p and q. However, if the distributions are discrete, we can compute the empirical distributions $\widetilde{p}$ and $\widetilde{q}$ from the positively labeled and negatively labeled samples respectively. If the distributions are continuous and belong to a parametric family such as Gaussians, we can estimate the statistics such as means and covariances from the samples using maximum-likelihood fitting. However, it turns out that for the purpose of optimizing (6.14), we only require the marginal and pairwise empirical statistics, i.e., the quantities $\widetilde{p}_i(x_i)$, $\widetilde{q}_i(x_i)$, $\widetilde{p}_{i,j}(x_i, x_j)$, and $\widetilde{q}_{i,j}(x_i, x_j)$. Estimating these pairwise quantities from the samples is substantially cheaper than computing the full empirical distribution or all the joint statistics. To optimize (6.14), we note that this objective can be rewritten as two independent optimization problems.

Proposition 6.2. (Decoupling of objective into two MWSTs) *The optimization in* (6.14) *decouples into:*

$$\widehat{p} = \operatorname*{argmin}_{p \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d(\widetilde{p}))} D(\widetilde{p} \,||\, p) - D(\widetilde{q} \,||\, p), \tag{6.15a}$$

$$\widehat{q} = \operatorname*{argmin}_{q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d(\widetilde{q}))} D(\widetilde{q} || \, q) - D(\widetilde{p} \,||\, q). \tag{6.15b}$$

Proof. The equivalence of (6.14) and (6.15) can be shown by using the definition of the tree-approximate J-divergence and noting that $\sum_{\mathbf{x}} \widetilde{p}(\mathbf{x}) \log \widehat{p}(\mathbf{x}) + H(\widetilde{p}) = -D(\widetilde{p}||\widehat{p})$. □

We have the following intuitive interpretation: the problem in (6.15a) is, in a precise sense, finding the distribution $\widehat{p}$ that is simultaneously "close to" the empirical distribution $\widetilde{p}$ and "far from" $\widetilde{q}$, while the reverse is true for $\widehat{q}$. See Fig. 6.1 for an illustration of the proposition. Note that all distances are measured using the KL-divergence. Each one of these problems can be solved by a MWST procedure with the appropriate edge weights given in the following proposition.

Proposition 6.3. (Edge Weights for Discriminative Trees) *Assume that $\widehat{p}$ and $\widehat{q}$ are marginally consistent with $\widetilde{p}$ and $\widetilde{q}$ respectively as defined in* (6.7)*. Then, for the selection of the edge set of $\widehat{p}$ in* (6.15a)*, we can apply a MWST procedure with the* weights

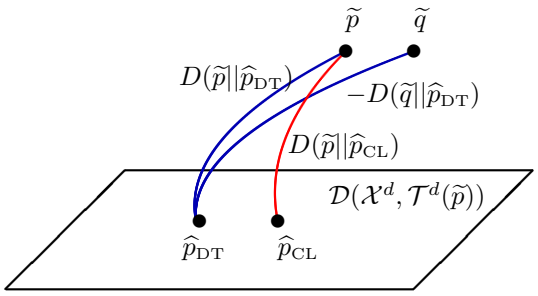

Figure 6.1. Illustration of Proposition 6.2 which shows the geometry of the distributions, where $\mathcal{D}(\mathcal{X}^d, \mathcal{T}^d(\widetilde{p}))$ is the subset of tree distributions that are marginally consistent with $\widetilde{p}$, the empirical distribution of the positively labeled samples. $\widehat{p}_{\mathrm{CL}}$, the generatively-learned distribution (via Chow-Liu), is the projection of $\widehat{p}$ onto the set of trees with the same marginals as $\widetilde{p}$ as given by the Chow-Liu optimization problem. $\widehat{p}_{\mathrm{DT}}$, the discriminatively-learned distribution, is the solution of (6.15a) which is "further" (in the KL-divergence sense) from $\widetilde{q}$ (because of the $-D(\widetilde{q}||\widehat{p})$ term).

on each pair of nodes $(i,j) \in \binom{V}{2}$ *are given by*

$$\psi_{i,j}^{(+)} := \mathbb{E}_{\widetilde{p}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right] - \mathbb{E}_{\widetilde{q}_{i,j}}\left[\log \frac{\widetilde{p}_{i,j}}{\widetilde{p}_i \widetilde{p}_j}\right]. \tag{6.16}$$

Proof. The proof can be found in Appendix 6.A. □

From (6.16), we observe that *only* the marginal and pairwise statistics are needed in order to compute the edge weights. Subsequently, the MWST is used to obtain $E_{\widehat{p}}$. Then, given this optimal tree structure, the model $\widehat{p}$ is the projection of $\widetilde{p}$ onto $E_{\widehat{p}}$. A similar procedure yields $\widehat{q}$, with edge weights $\psi_{i,j}^{(-)}$ given by an expression similar to (6.16), but with $\widetilde{p}$ and $\widetilde{q}$ interchanged.

Given: Training set $\mathcal{S}$.

1: Using the samples in $\mathcal{S}$, estimate the pairwise statistics $\widetilde{p}_{i,j}(x_i, x_j)$ and $\widetilde{q}_{i,j}(x_i, x_j)$ for all edges (i,j) using, for example, maximum-likelihood estimation.

2: Compute edge weights $\{\psi_{i,j}^{(+)}\}$ and $\{\psi_{i,j}^{(-)}\}$, using (6.16), for all edges (i,j).

3: Given the edge weights, find the optimal tree structures using a MWST algorithm such as Kruskal's [120], i.e.,

$$E_{\widehat{p}} = \mathrm{MWST}(\{\psi_{i,j}^{(+)}\}), \qquad E_{\widehat{q}} = \mathrm{MWST}(\{\psi_{i,j}^{(-)}\}). \tag{6.17}$$

4: Set $\widehat{p}$ to be the projection of $\widetilde{p}$ onto $E_{\widehat{p}}$ and $\widehat{q}$ to be the projection of $\widetilde{q}$ onto $E_{\widehat{q}}$.

5: **return** Approximate distributions $\widehat{p}(\mathbf{x})$ and $\widehat{q}(\mathbf{x})$ to be used in a likelihood ratio test $h(\mathbf{x}) = \mathrm{sgn}[\log(\widehat{p}(\mathbf{x})/\widehat{q}(\mathbf{x}))]$ to assign a label to a test sample $\mathbf{x}$.

Algorithm 1. Discriminative Trees (DT)

This discriminative tree (DT) learning procedure produces at most $d-1$ edges (pairwise features) in each tree model $\widehat{p}$ and $\widehat{q}$ (some of the edge weights $\psi_{i,j}^{(+)}$ in (6.16) may turn out to be negative so the algorithm may terminate early). The tree models $\widehat{p}$ and $\widehat{q}$ will then be used to construct $\widehat{\varphi}$, which is used in the likelihood ratio test (6.3). Algorithm 1 summarizes our method for discriminatively learning tree models. Section 6.5.2 compares the classification performance of this method with other tree-based methods such as Chow-Liu as well as TAN [84, 212]. Finally, we remark that the proposed procedure has exactly the same complexity as learning a TAN network.

■ 6.3.3 Connection to the Log-Likelihood Ratio

We now state a simple and intuitively-appealing result that relates the optimization of the tree-approximate J-divergence to the likelihood ratio test in (6.1).

Proposition 6.4. (Empirical Log-Likelihood Ratio) *For discrete distributions, optimizing the tree-approximate J-divergence in* (6.14) *is equivalent to maximizing the* empirical log-likelihood ratio *of the training samples, i.e.,*

$$(\widehat{p},\widehat{q}) = \underset{\widehat{p}\in\mathcal{D}(\mathcal{X}^d,\mathcal{T}^d(\widetilde{p})),\ \widehat{q}\in\mathcal{D}(\mathcal{X}^d,\mathcal{T}^d(\widetilde{q}))}{\operatorname{argmax}} \sum_{l=1}^{n} y_l \log\left[\frac{\widehat{p}(\mathbf{x}_l)}{\widehat{q}(\mathbf{x}_l)}\right]. \tag{6.18}$$

Proof. Partition the training set $\mathcal{S}$ into positively labeled samples $\mathcal{S}^+ := \{\mathbf{x}_l : y_l = +1\}$ and negatively labeled samples $\mathcal{S}^- := \{\mathbf{x}_l : y_l = -1\}$ and split the sum in (6.18) corresponding to these two parts accordingly. Then the sums (over the sets $\mathcal{S}^+$ and $\mathcal{S}^-$) are equal to (6.15a) and (6.15b) respectively. Finally use Proposition 6.2 to conclude that the empirical log-likelihood ratio is equivalent to the tree-approximate J-divergence defined in (6.14). □

This equivalent objective function has a very intuitive meaning. Once $\widehat{p}$ and $\widehat{q}$ have been learned, we would like $\log\widehat{\varphi}(\mathbf{x}_l) := \log[\widehat{p}(\mathbf{x}_l)/\widehat{q}(\mathbf{x}_l)]$ to be positive (and as large as possible) for all samples with label $y_l = +1$, and negative (with large magnitude) for those with label $y_l = -1$. The objective function in (6.18) precisely achieves this purpose.

It is important to note that (6.14) involves maximizing the tree-approximate J-divergence. This does not mean that we are directly minimizing the probability of error. In fact, we would not expect convergence to the true distributions p and q when the number of samples tends to infinity if we optimize the discriminative criterion (6.15).[5] However, since we are explicitly optimizing the log-likelihood ratio in (6.18), we would expect that if one has a limited number of training samples, we will learn distributions $\widehat{p}$ and $\widehat{q}$ that are better at discrimination than generative models in the likelihood ratio

[5] However, if the true distributions are trees, minimizing the KL-divergence over the set of trees ($\min_{p\in\mathcal{D}(\mathcal{X}^d,\mathcal{T}^d)} D(\widetilde{p}\|p)$) with the empirical $\widetilde{p}$ as the input is a maximum-likelihood procedure (Section 2.5.2). It consistently recovers the structure of the true distribution p exponentially fast in n (Chapters 3 and 4).

test (6.3). This can be seen in the objective function in (6.15a) which is a *blend* of two terms. In the first term $D(\widetilde{p}\,||\,p)$, we favor a model $\widehat{p}$ that minimizes the KL-divergence to its empirical distribution $\widetilde{p}$. In the second term $D(\widetilde{p}\,||\,q)$, we favor the maximization of the empirical *type-II error exponent* $D(\widetilde{q}\,||\,p)$ for testing p against the distribution in the alternate hypothesis q (Chernoff-Stein Lemma [47, Ch. 12]).

■ 6.3.4 Learning Optimal Forests

In this subsection, we mention how the objective in (6.14), can be jointly maximized over pairs of *forest* distributions $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$. Both $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$ are Markov on forests with at most $k \leq d-1$ edges. This formulation is important since if we are given a fixed budget of only k edges per distribution, we would like to maximize the *joint* objective over *both* pairs of distributions instead of decomposing the objective into two independent problems as in (6.15). This formulation also provides us with a natural way to incorporate costs for the selection of edges.

We use that notation $\mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d(\widetilde{p}))$ to denote the set of probability distributions that are Markov on forests with *at most* k edges and have the same node and edge marginals as $\widetilde{p}$, i.e., marginally consistent with the empirical distribution $\widetilde{p}$. We now reformulate (6.14) as a joint optimization over the class of forests with at most k edges given empiricals $\widetilde{p}$ and $\widetilde{q}$:

$$(\widehat{p}^{(k)}, \widehat{q}^{(k)}) = \underset{\widehat{p} \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d(\widetilde{p})), \widehat{q} \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}_k^d(\widetilde{q}))}{\operatorname{argmax}} \widehat{J}(\widehat{p}, \widehat{q}; \widetilde{p}, \widetilde{q}). \tag{6.19}$$

For each k, the resulting distributions $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$ are optimal with respect to the tree-approximate J-divergence and the final pair of distributions $\widehat{p}^{(d-1)}$ and $\widehat{q}^{(d-1)}$ corresponds exactly to $\widehat{p}$ and $\widehat{q}$, the outputs of the DT algorithm as detailed in Algorithm 1. However, we emphasize that $\widehat{p}^{(k)}, \widehat{q}^{(k)}$ (for $k < d-1$) will, in general, be different from the outputs of the DT algorithm (with at most k edges chosen for each model) because (6.19) is a *joint* objective over forests. Furthermore, each forest has *at most* k edges but could have fewer depending on the sign of the weights in (6.11). The number of edges in each forest may also be different. We now show that the objective in (6.19) can be optimized easily with a slight modification of the basic Kruskal's MWST algorithm [120].

We note the close similarity between the discriminative objective in (6.10) and the Chow-Liu optimization for a single spanning tree in (2.109). In the former, the edge weights are given by w_{ij} in (6.11) and in the latter, the edge weights are the mutual information quantities $I(X_i; X_j)$. Note that the two objective functions are *additive*. With this observation, it is clear that we can equivalently choose to maximize the second term in (6.10), i.e.,

$$\sum_{(i,j) \in E_{\widehat{p}} \cup E_{\widehat{q}}} w_{ij}, \tag{6.20}$$

over the set of trees, where each w_{ij} is a function of the empirical pairwise statistics $\widetilde{p}_{i,j}(x_i, x_j)$ and $\widetilde{q}_{i,j}(x_i, x_j)$ (and corresponding information-theoretic measures) that can

be estimated from the training data. To maximize the sum in (6.20), we use the same MWST algorithm with edge weights given by w_{ij}. In this case, we must consider the maximum of the three possible values for w_{ij}. Whichever is the maximum (or if all three are negative) indicates one of four possible actions:

1. Place an edge between i and j for $\widehat{p}$ and *not* $\widehat{q}$ (corresponding to $(i,j) \in E_{\widehat{p}} \setminus E_{\widehat{q}}$).
2. Place an edge between i and j for $\widehat{q}$ and *not* $\widehat{p}$ (corresponding to $(i,j) \in E_{\widehat{q}} \setminus E_{\widehat{p}}$).
3. Place an edge between i and j for *both* $\widehat{p}$ and $\widehat{q}$ (corresponding to $(i,j) \in E_{\widehat{p}} \cap E_{\widehat{q}}$).
4. Do not place any edge between i and j for either model $\widehat{p}$ and $\widehat{q}$ if all three values of w_{ij} in (6.11) are negative.

Proposition 6.5. (Optimality of Kruskal for Learning Forests) *For the optimization problem in* (6.19)*, the k-step Kruskal's MWST algorithm, considering the maximum over the three possible values of* w_{ij} *in* (6.11) *and the four actions above, results in optimal forest-structured distributions* $\widehat{p}^{(k)}(x)$ *and* $\widehat{q}^{(k)}(x)$ *with edge sets* $E_{\widehat{p}^{(k)}}$ *and* $E_{\widehat{q}^{(k)}}$*.*

Proof. This follows directly from the additivity of the objective in (6.10) and the optimality of Kruskal's MWST algorithm [120] for each $k = 1, \ldots, d-1$. See [45, Section 23.1] for the details. □

The k-step Kruskal's MWST algorithm is the usual Kruskal's algorithm terminated after at most $k \leq d-1$ edges have been added. The edge sets are nested and we state this formally as a corollary of Proposition 6.5.

Corollary 6.6 (Nesting of Edge Sets)**.** *The edge sets* $E_{\widehat{p}^{(k)}}$ *obtained from the maximization (6.19) are nested, i.e.,* $E_{\widehat{p}^{(k-1)}} \subseteq E_{\widehat{p}^{(k)}}$ *for all* $k = 1, \ldots, d-1$ *and similarly for* $E_{\widehat{q}^{(k)}}$*.*

This appealing property ensures that *one* single run of Kruskal's MWST algorithm recovers *all* $d-1$ substructures $\{(\widehat{p}^{(k)}, \widehat{q}^{(k)})\}_{1 \leq k \leq d-1}$. Thus, this procedure is computationally efficient.

■ 6.3.5 Assigning Costs to the Selection of Edges

In many applications, it is common to associate the selection of more features with higher costs. We now demonstrate that it is easy to incorporate this consideration into our optimization program in (6.19).

Suppose we have a set of costs $\mathcal{C} := \{c_{ij} \geq 0 : (i,j) \in \binom{V}{2}\}$, where each element c_{ij} is the cost of selecting edge (i,j). For example, in the absence of any prior information, we may regard each of these costs c_{ij} as being equal to a constant $c \geq 0$. We would like to maximize optimize $\widehat{J}$, given in (6.19), over the two models $\widehat{p}$ and $\widehat{q}$ taking the

costs of selection of edges into consideration. From Proposition 6.1, the new objective function can now be expressed as

$$\widehat{J}_{\mathcal{C}}(\widehat{p},\widehat{q};p,q) = \sum_{i\in V} J(p_i,q_i) + \sum_{(i,j)\in E_{\widehat{p}}\cup E_{\widehat{q}}} \overline{w}_{ij} \tag{6.21}$$

where the cost-modified edge weights are defined as $\overline{w}_{ij} := w_{ij} - c_{ij}$. Thus, the costs c_{ij} appear only in the new edge weights $\overline{w}_{ij}$. We can perform the same greedy selection procedure with the new edge weights $\overline{w}_{ij}$ to obtain the "cost-adjusted" edge sets $\overline{E}_{\widehat{p}^{(k)}}$ and $\overline{E}_{\widehat{q}^{(k)}}$. Interestingly, this also gives a natural stopping criterion. Indeed, whenever all the remaining $\overline{w}_{ij}$ are negative the algorithm should terminate as the overall cost will not improve.

■ 6.4 Learning a Larger Set of Features via Boosting

We have described efficient algorithms to learn tree distributions discriminatively by maximizing the empirical log-likelihood ratio in (6.18) (or the tree-approximate J-divergence). However, learning a larger set of features (more than $d-1$ edges per model) would enable better classification in general if we are *also* able to prevent overfitting. In light of the previous section, the first natural idea for learning thicker graphical models (i.e., graphical models with more edges) is to attempt to optimize an expression like (6.14), but over a set of thicker graphical models, e.g., the set of graphical models with bounded treewidth. However, this approach is complicated because the graph selection problem was simplified for trees as it was possible to determine *a-priori* the projection of the empirical distribution onto the learned structure. Such a projection also holds for the construction of junction trees, but maximum-likelihood structure learning is known to be NP-hard [112]. For graphs that are not junction trees, computing the projection parameters *a priori* is, in general, intractable. Furthermore, the techniques proposed in [2, 128, 136, 211] used to learn such graphs are tightly coupled to the generative task of approximating $\widetilde{p}$, and even for these it is not straightforward to learn parameters given the loopy structure.

■ 6.4.1 Real-AdaBoost

A review of AdaBoost (also called Discrete-AdaBoost) was given in Section 2.3.2. *Real-AdaBoost* [82, 175] is a variant of Discrete-AdaBoost for the case when it is possible to obtain real-valued *confidences* from the weak classifiers, i.e., if $h_t : \mathcal{X}^d \to \mathbb{R}$ [with more positive $h_t(\mathbf{x})$ signifying higher bias for positively labeled samples].[6] It has been observed empirically that Real-AdaBoost often performs better than its discrete counterpart [82, 175]. We found this behavior in our experiments also as will be reported in

[6] For instance, if the weak classifier is chosen to be the logistic regression classifier, then the confidences are the probabilistic outputs $p(y|\mathbf{x})$.

Section 6.5.4. The strong classifier resulting from the Real-AdaBoost procedure is

$$H_T(\mathbf{x}) = \operatorname{sgn}\left[\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right], \tag{6.22}$$

where the set of coefficients are given by $\{\alpha_t \geq 0\}_{t=1}^{T}$.

■ 6.4.2 Learning a Larger Set of Pairwise Features via Real-AdaBoost

In the language of Real-AdaBoost, the *tree-based classifiers* or the *forests-based classifiers* presented in Sections 6.3 may be regarded as *weak classifiers* to be combined to form a stronger classifier. More specifically, each weak classifier $h_t : \mathcal{X}^d \to \mathbb{R}$ is given by the log-likelihood ratio $h_t(\mathbf{x}) = \log \widehat{\varphi}_t(\mathbf{x}) = \log[\widehat{p}_t(\mathbf{x})/\widehat{q}_t(\mathbf{x})]$, where $\widehat{p}_t$ and $\widehat{q}_t$ are the tree-structured graphical model classifiers learned at the t-th boosting iteration. Running T boosting iterations, now allows us to learn a larger set of features and to obtain a better approximation of the likelihood ratio $\widehat{\varphi}(\mathbf{x})$ in (6.4). This is because the strong ensemble classifier H_T can be written as

$$H_T(\mathbf{x}) = \operatorname{sgn}\left[\sum_{t=1}^{T} \alpha_t \log\left(\frac{\widehat{p}_t(\mathbf{x})}{\widehat{q}_t(\mathbf{x})}\right)\right], \tag{6.23a}$$

$$= \operatorname{sgn}\left[\log\left(\frac{\prod_{t=1}^{T} \widehat{p}_t(\mathbf{x})^{\alpha_t}}{\prod_{t=1}^{T} \widehat{q}_t(\mathbf{x})^{\alpha_t}}\right)\right], \tag{6.23b}$$

$$= \operatorname{sgn}\left[\log\left(\frac{\widehat{p}^*(\mathbf{x})}{\widehat{q}^*(\mathbf{x})}\right)\right]. \tag{6.23c}$$

In (6.23c), $\widehat{p}^*(\mathbf{x})$, an unnormalized distribution, is of the form

$$\widehat{p}^*(\mathbf{x}) := \prod_{t=1}^{T} \widehat{p}_t(\mathbf{x})^{\alpha_t}. \tag{6.24}$$

Define $Z_p(\alpha) = Z_p(\alpha_1, \ldots, \alpha_T) = \sum_{\mathbf{x}} \widehat{p}^*(\mathbf{x})$ to be the normalizing constant for $\widehat{p}^*$ in (6.24). Hence the distribution (or graphical model) $\widehat{p}^*(\mathbf{x})/Z_p(\alpha)$ sums to unity.

Proposition 6.7. (Markovianity of Normalized Distributions) *The normalized distribution $\widehat{p}^*(\mathbf{x})/Z_p(\alpha)$ is Markov on a graph $G = (V, E_{\widehat{p}^*})$ with edge set*

$$E_{\widehat{p}^*} = \bigcup_{t=1}^{T} E_{\widehat{p}_t}. \tag{6.25}$$

The same relation in (6.25) *holds for the normalized distribution $\widehat{q}^*(\mathbf{x})/Z_q(\alpha)$.*

Proof. (Sketch) This follows by writing each $\widehat{p}_t$ as a member of an exponential family, combining $\widehat{p}_t$'s to give $\widehat{p}^*$ as in (6.24) and finally applying the Hammersley-Clifford Theorem [91]. See Appendix 6.B for the details. □

Because we are entirely concerned with accurate classification, and the value of the ratio $\widehat{\varphi}^*(\mathbf{x}) = \widehat{p}^*(\mathbf{x})/\widehat{q}^*(\mathbf{x})$ in (6.23c), we do not need to normalize our models $\widehat{p}^*$ and $\widehat{q}^*$. By leaving the models unnormalized, we retain many appealing theoretical guarantees [173] afforded by the boosting procedure, such as the exponential decay in the training error. Furthermore, we are able to interpret the resulting *normalized* models[7] as being Markov on particular loopy graphs (whose edge sets are given in Proposition 6.7), which contain a larger set of features as compared to simple tree models.

Note that after T boosting iterations, we have a maximum of $(d-1)T$ pairwise features in each model as each boosting iteration produces at most $d-1$ pairwise features (because some weights in (6.16) could be negative). To learn these features, we now need to learn tree models to minimize the *weighted* training error, as opposed to unweighted error as in Section 6.3. This can be achieved by replacing the empirical distributions $\widetilde{p}, \widetilde{q}$ with the weighted empirical distributions $\widetilde{p}_w, \widetilde{q}_w$ and the weights are updated based on whether each sample $\mathbf{x}_l$ is classified correctly. The resulting tree models will thus be projections of the weighted empirical distributions onto the corresponding learned tree structures. The method for learning a larger set of features from component tree models is summarized in Algorithm 2. Note that Algorithm 2 is essentially a restatement of Real-Adaboost but with the weak classifiers learned using Discriminative Trees (Algorithm 1).

■ 6.5 Numerical Experiments

This section is devoted to an extensive set of numerical experiments that illustrate the classification accuracy of discriminative trees and forests as well as thicker graphical models. It is subdivided into the following subsections.

1. Firstly, in Section 6.5.1, we present an illustrate example to show that our discriminative tree/forest learning procedure as detailed in Sections 6.3.2 and 6.3.4 results in effective tree-based classifiers.

2. Secondly, in Section 6.5.2 we compare our discriminative trees procedure to other tree-based classifiers using real datasets. We also extend our ideas naturally to multi-class classification problems.

3. Finally, in Section 6.5.4, we demonstrate empirically on a range of datasets that our method to learn thicker models outperforms standard classification techniques.

[7] We emphasize that the unnormalized models $\widehat{p}^*$ and $\widehat{q}^*$ are not probability distributions and thus cannot be interpreted as *graphical models*. However, the discriminative tree models learned in Section 6.3 are indeed normalized and hence are graphical models.

Given: Training data $\mathcal{S}$. Number of boosting iterations T.

1: Initialize the weights to be uniform, i.e., set $w_0^{(l)} = 1/n$ for all $1 \leq l \leq n$.

2: **for** $t = 1 : T$ **do**

3: Find discriminative tree models $\widehat{p}_t, \widehat{q}_t$ using Algorithm 1, but with the *weighted* empirical distributions $\widetilde{p}_w, \widetilde{q}_w$.

4: The weak classifier $h_t : \mathcal{X}^d \to \mathbb{R}$ is given by $h_t(\mathbf{x}) = \log\left[\widehat{p}_t(\mathbf{x})/\widehat{q}_t(\mathbf{x})\right]$.

5: Perform a convex line search to find the optimal value of the coefficients α_t:

$$\alpha_t = \underset{\beta \geq 0}{\operatorname{argmin}} \sum_{l=1}^{n} w_t^{(l)} \exp\left[-\beta y_l h_t(\mathbf{x}_l)\right].$$

6: Update and normalize the weights:

$$w_{t+1}^{(l)} = \frac{w_t^{(l)}}{\zeta_t} \exp\left[-\alpha_t y_l h_t(\mathbf{x}_l)\right], \ \forall\, l = 1, \ldots, n,$$

where $\zeta_t := \sum_{l=1}^{n} w_t^{(l)} \exp[-\alpha_t y_l h_t(\mathbf{x}_l)]$ is the normalization constant to ensure that the weights sum to unity after the update.

7: **end for**

8: **return** Coefficients $\{\alpha_t\}_{t=1}^{T}$ and models $\{\widehat{p}_t, \widehat{q}_t\}_{t=1}^{T}$. The final classifier is given in (6.23).

Algorithm 2. Boosted Graphical Model Classifiers (BGMC)

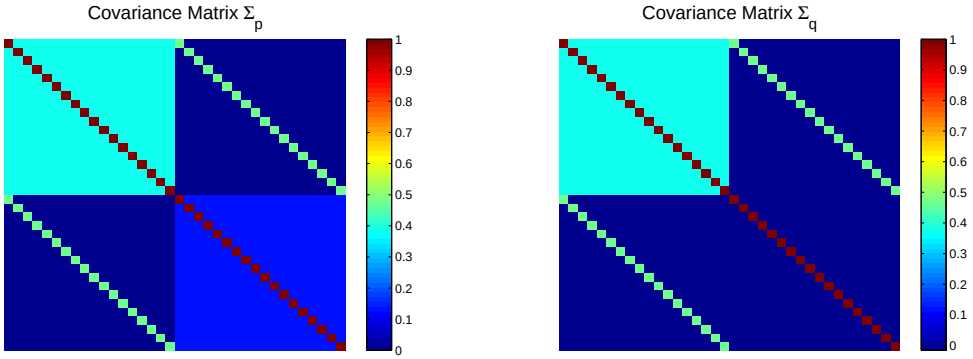

Figure 6.2. Class covariance matrices $\mathbf{\Sigma}_p$ and $\mathbf{\Sigma}_q$. The only discriminative information arises from the lower-right block.

■ 6.5.1 Discriminative Trees: An Illustrative Example

We now construct two Gaussian graphical models p and q such that the real statistics are not trees and the maximum-likelihood trees (learned from Chow-Liu) are exactly the

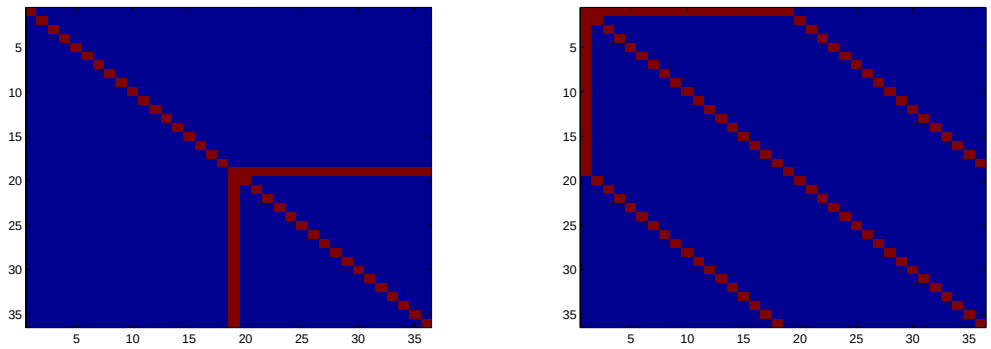

Figure 6.3. Structures of $\widehat{p}^{(k)}$ at iteration $k = d - 1$. The figures show the adjacency matrices of the graphs, where the edges selected at iteration $d - 1$ are highlighted in red. In the left plot, we show the discriminative model, which extracts the edges corresponding to the discriminative block (lower-right corner) of the class conditional covariance matrix. In the right plot, we show the generative model, which does not extract the discriminative edges.

same, but the discriminative trees procedure gives distributions that are *different*. Let p and q be the probability density functions of two zero-mean d-variate (d even) Gaussian random vectors with class-conditional covariance matrices $\mathbf{\Sigma}_p$ and $\mathbf{\Sigma}_q$ respectively, i.e., $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}; \mathbf{0}, \mathbf{\Sigma}_p) \propto \exp(-\mathbf{x}^T \mathbf{\Sigma}_p^{-1} \mathbf{x}/2)$, where

$$\mathbf{\Sigma}_p := \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{C}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathrm{A}} \end{bmatrix} + \mathbf{\Sigma}_{\mathrm{N}}, \quad \mathbf{\Sigma}_q := \begin{bmatrix} \mathbf{\Sigma}_{\mathrm{C}} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Sigma}_{\mathrm{B}} \end{bmatrix} + \mathbf{\Sigma}_{\mathrm{N}}, \tag{6.26}$$

and the noise matrix is given as

$$\mathbf{\Sigma}_{\mathrm{N}} := \begin{bmatrix} \mathbf{I}_{d/2} & \rho \mathbf{I}_{d/2} \\ \rho \mathbf{I}_{d/2} & \mathbf{I}_{d/2} \end{bmatrix}. \tag{6.27}$$

In (6.26), $\mathbf{\Sigma}_{\mathrm{C}}$, $\mathbf{\Sigma}_{\mathrm{A}}$ and $\mathbf{\Sigma}_{\mathrm{B}}$ are carefully selected $d/2 \times d/2$ positive definite matrices.

Note, from the construction, that the *only discriminative* information comes from the lower block terms in the class conditional covariance matrices as these are the only terms that differ between the two models. We set ρ to be the highest correlation coefficient of any off-diagonal element in $\mathbf{\Sigma}_p$ or $\mathbf{\Sigma}_q$. This ensures that those edges are the first $d/2$ chosen in any Chow-Liu tree. These edges connect discriminative variables to non-discriminative variables. Next we design $\mathbf{\Sigma}_{\mathrm{C}}, \mathbf{\Sigma}_{\mathrm{A}}, \mathbf{\Sigma}_{\mathrm{B}} \succ 0$ such that all of the correlation coefficient terms in the (common) upper block $\mathbf{\Sigma}_{\mathrm{C}}$ are higher than any in $\mathbf{\Sigma}_{\mathrm{A}}$ or $\mathbf{\Sigma}_{\mathrm{B}}$. This results in generative trees learned under Chow-Liu which provide no discriminative information. The additive noise term will not affect off-diagonal terms in either $\mathbf{\Sigma}_{\mathrm{A}}$ or $\mathbf{\Sigma}_{\mathrm{B}}$. The two matrices $\mathbf{\Sigma}_p$ and $\mathbf{\Sigma}_q$ are shown in Fig. 6.2.

We now apply two structure learning methods (Chow-Liu [42] and the discriminative forest-learning method in Section 6.3.4) to learn models $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$ sequentially.

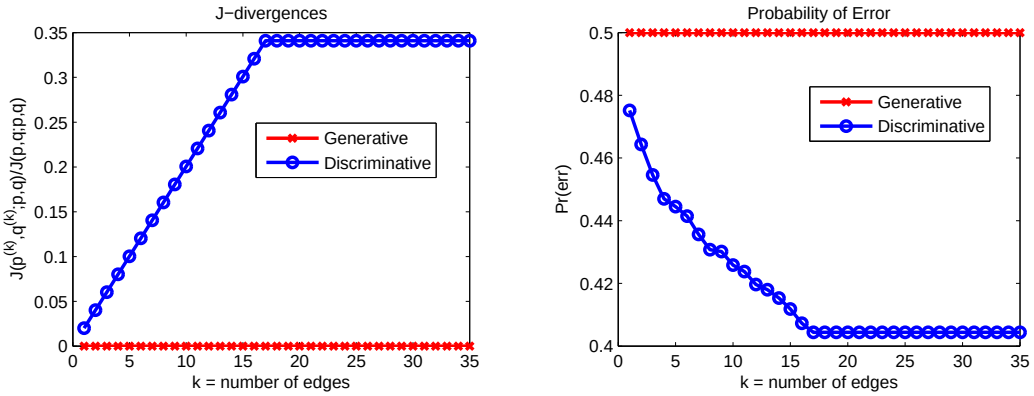

Figure 6.4. Tree-approximate J-divergence and Pr(err). Note the monotonic increase of the tree-approximate J-divergence for the discriminative model. The generative model provides no discrimination as evidenced by the zero divergence and Pr(err) $= 1/2$.

For this toy example, we assume that we have the true distributions. The learned structures are shown in Fig. 6.3. Note that, by construction, the discriminative algorithm terminates after $d/2$ steps since no more discriminative information can be gleaned without the addition of an edge that results in a loop. The generative structure is very different from the discriminative one. In fact, both the $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$ structures are exactly the same for each k. This is further validated from Fig. 6.4, where we plot the tree-approximate J-divergence between $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$ (relative to p and q) and the probability of error Pr(err) as a function of k. The Pr(err) is approximated using 10,000 test samples generated from the original distributions p and q. We see that the generative method provides no discrimination in this case, evidenced by the fact that the J-divergence is identically 0 and the Pr(err) is exactly $1/2$. As expected, the J-divergence of the discriminative models increases monotonically and the Pr(err) decreases monotonically. Thus, this example clearly illustrates the differences between the generative [42] and discriminative learning algorithms. Clearly, it is advantageous to optimize the discriminative objective (6.19) if the purpose, namely binary classification, is known *a-priori*.

■ 6.5.2 Comparison of DT to Other Tree-Based Classifiers

We now compare various tree-based graphical model classifiers, namely our proposed Discriminative Trees (DT) learning algorithm, Chow-Liu and finally TAN [84]. We perform the experiment on a quantized version of the MNIST handwritten digits dataset.[8] The results are averaged over 50 randomly partitioned training (80% of available data) and test sets (20%). The probability of error Pr(err) as a function of the number of training examples n is plotted in Fig. 6.5. We observe that in general our DT algorithm performs the best, especially in the absence of a large number of training examples.

[8]Each pixel with a non-zero value is quantized to 1.

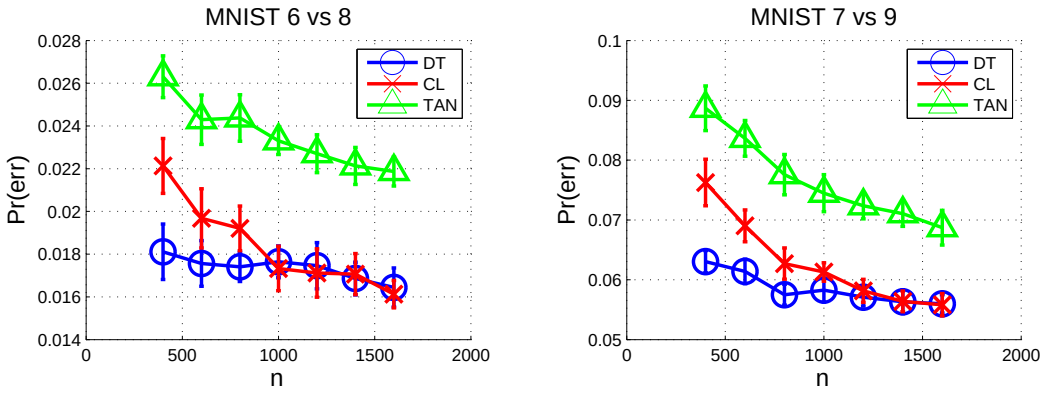

Figure 6.5. Pr(err) between DT, Chow-Liu and TAN using a pair of trees. Error bars denote 1 standard deviation from the mean. If the total number of training samples n is small, then typically DT performs much better than Chow-Liu and TAN.

This makes good intuitive sense: With a limited number of training samples, a discriminative learning method, which captures the *salient differences* between the classes, should generalize better than a generative learning method, which models the distributions of the individual classes. Also, the computational complexities of DT and TAN are exactly the same.

■ 6.5.3 Extension to Multi-class Problems

Next, we consider extending the sequential forest learning algorithm described in Section 6.3.4 to handle multi-class problems.[9] In multi-class problems, there are $M \geq 2$ classes, i.e., the class label Y described in Section 6.2.1 can take on more than 2 values. For example, we would like to determine which digit in the set $\mathcal{I} := \{0, 1, \ldots, 9\}$ a particular noisy image contains. For this experiment, we again use images from the MNIST database, which consists of $M = 10$ classes corresponding to the digits in the set $\mathcal{I}$. Since each of the $n = 60,000$ images in the database is of size 28 by 28, the dimensionality of the data is $d = 28 \times 28 = 784$. There is a separate test set containing 10,000 images, which we use to estimate the error probability. We pre-processed each image by concatenating the columns. We modeled each of the M classes by a multivariate Gaussian with mean vector $\boldsymbol{\mu}_i \in \mathbb{R}^d$ and positive definite covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{d \times d}$. To handle this multi-class classification problem, we used the well-known *one-vs-all* strategy described in Rifkin and Klautau [161] to classify the test images. We define $\widehat{p}^{(k)}_{i|j}(\mathbf{x})$ and $\widehat{p}^{(k)}_{j|i}(\mathbf{x})$ to be the learned forest distributions with at most k edges for the binary classification problem for digits i (positive class) and j (negative class)

[9] The DT algorithm can also be extended to multi-class problems in the same way.

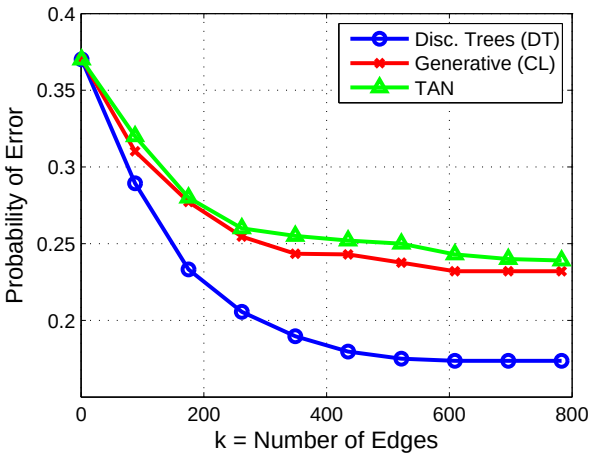

Figure 6.6. Pr(err)'s for the MNIST Digits dataset for the multi-class problem with $M = 10$ classes (hypotheses). The horizontal axis is k, the number of edges added to each model $\widehat{p}$ and $\widehat{q}$. Note that the discriminative method outperforms the generative (Chow-Liu) method and TAN.

respectively. For each k, we also define the family of functions $f_{ij}^{(k)} : \mathcal{X}^d \to \mathbb{R}$ as

$$f_{ij}^{(k)}(\mathbf{x}) := \log \left[\frac{\widehat{p}_{i|j}^{(k)}(\mathbf{x})}{\widehat{p}_{j|i}^{(k)}(\mathbf{x})} \right], \quad i, j \in \mathcal{I}. \tag{6.28}$$

Thus, $\operatorname{sgn} f_{ij}^{(k)} : \mathcal{X}^n \to \{-1, +1\}$ is the classifier (for which both forests have no more than k edges) that discriminates between digits i and j. Note that $f_{ij}^{(k)}(\mathbf{x}) = -f_{ji}^{(k)}(\mathbf{x})$. These distributions correspond to the $\widehat{p}^{(k)}$ and $\widehat{q}^{(k)}$ for the binary classification problem. The decision for the multi-class problem is then given by the composite decision function [161] $g^{(k)} : \mathcal{X}^d \to \mathcal{I}$, defined as:

$$g^{(k)}(\mathbf{x}) := \operatorname*{argmax}_{i \in \mathcal{I}} \sum_{j=0}^{M-1} f_{ij}^{(k)}(\mathbf{x}). \tag{6.29}$$

The results of the experiment are shown in Fig. 6.6. We see that the discriminative method to learn the sequence of forests results in a lower Pr(err) (estimated using the test set) than the generative method for this dataset and TAN. This experiment again highlights the advantages of our proposed discriminative learning method detailed in Section 6.3 as compared to Chow-Liu trees [42] or TAN [84].

■ 6.5.4 Comparison of BGMC to other Classifiers

In this section, we return to the binary classification problem and show empirically that our boosting procedure results in models that are better at classifying various datasets

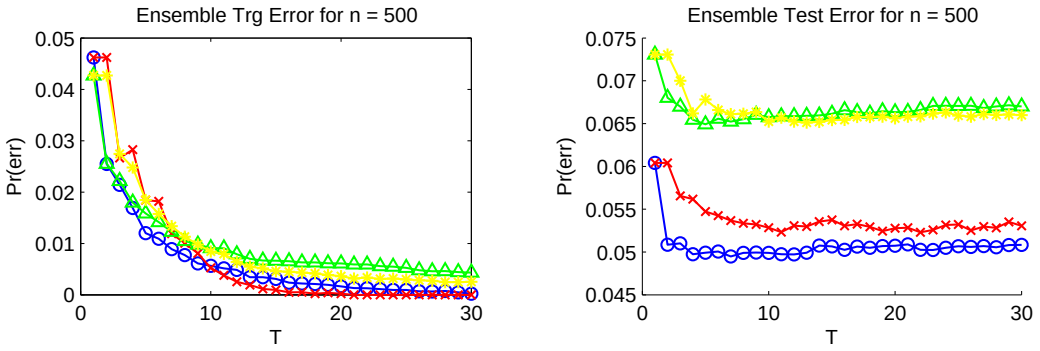

Figure 6.7. Discrimination between the digits 7 and 9 in the MNIST dataset. T is the number of boosting iterations. Yellow $\diamond$: (Chow-Liu + Discrete-AdaBoost), Green $\triangle$: (Chow-Liu + Real-AdaBoost), Red $\times$: Discriminative Trees + Discrete-AdaBoost, Blue $\circ$: Discriminative Trees + Real-AdaBoost (the proposed algorithm, BGMC). BGMC demonstrates lower training and test errors on this dataset. The training error decreases monotonically as expected. CV can be used to find the optimal number of boosting iterations to avoid overfitting. Observe from (b) that boosting (and in particular BGMC) is fairly robust to overfitting because even if T increases, the test error (also called generalization error) does not increase drastically.

as compared to boosted versions of tree-based classifiers. Henceforth, we term our method, described in Section 6.4 (and in detail in Algorithm 2) as Boosted Graphical Model Classifier (BGMC).

In Fig. 6.7, we show the evolution of the training and test errors for discriminating between the digits 7 and 9 in the MNIST dataset as a function of T, the number of boosting iterations. We set the number of training samples $n = 500$. We compare the performance of four different methods: Chow-Liu learning with either Discrete-AdaBoost or Real-AdaBoost and Discriminative Trees with either Discrete-AdaBoost or Real-AdaBoost. We observe that the test error for Discriminative Trees + Real-AdaBoost, which was the method (BGMC) proposed in Section 6.4, is the minimum. Also, after a small number of boosting iterations, the test error does not decrease any further. Cross-validation (CV) [6] may thus be used to determine the optimal number of boosting iterations. We now compare **BGMC** to a variety of other classifiers:

1. **BCL**: A boosted version of the Chow-Liu algorithm [42] where a pair of trees is learned generatively, one for each class. Note that only the positively (resp. negatively) labeled samples are used to estimate $\widehat{p}$ (resp. $\widehat{q}$). Subsequently, the trees are combined using the method detailed in Section 6.4.

2. **BTAN**: A boosted version of TAN [84]. Recall that TAN is such that two trees with the *same* structure are learned.

3. **SVM**: Support Vector Machines [201] using the quadratic kernel $K_2(\mathbf{x}^{(a)}, \mathbf{x}^{(b)}) = (1 + \langle \mathbf{x}^{(a)}, \mathbf{x}^{(b)} \rangle)^2$, with the slack parameter $C > 0$ found by CV.[10] We obtained

[10] We used 20% of the training samples to determine the best value of C.

the SVM code from [34].

For boosting, the optimal number of boosting iterations T^*, was also found by CV. For the set of experiments we performed, we found that T^* is typically small ($\approx 3-4$); hence the resulting normalized models remain sparse (Proposition 6.7).

Synthetic Dataset We generated a dataset by assuming that p and q are Markov on $d = 10 \times 10$ binary grid models with different randomly chosen parameters. We generated $n = 1200$ samples to learn boosted discriminative trees. The purpose of this experiment was to compare the number of edges added to the models and the (known) number of edges in the original grid models. The original grid models each have $2 \times 9^2 = 162$ edges and the learned models have at most $(d-1)T^* = 99 \times 3 = 297$ edges since the CV procedure results in an optimal boosting iteration count of $T^* = 3$. However, some of the edges in $\widehat{p}_1, \widehat{p}_2, \widehat{p}_3$ (and $\widehat{q}_1, \widehat{q}_2, \widehat{q}_3$) coincide and this results in $|\cup_{t=1}^{T^*} E_{\widehat{p}_t}| = 180$ (and $|\cup_{t=1}^{T^*} E_{\widehat{q}_t}| = 187$). Thus, there are 180 and 187 *distinct* edges in the $\widehat{p}^*$ and $\widehat{q}^*$ models respectively. From the top left plot in Fig. 6.8, we see that CV is effective for the purpose of finding a balance between optimizing modeling ability and preventing overfitting.

Real-World Datasets We also obtained five different datasets from the UCI Machine Learning Repository [144] as well as the previously-mentioned MNIST database. For datasets with continuous variables, the data values were quantized so that each variable only takes on a finite number of values. For datasets without separate training and test sets, we estimated the test error by averaging over 100 randomly partitioned training-test sets from the available data. The Pr(err) as a function of the number of training examples n is plotted in Fig. 6.8 for a variety of datasets. We observe that, apart from the Pendigits dataset, BGMC performs better than the other two (boosted) graphical model classifiers. Also, it compares well with SVM. In particular, for the synthetic, three MNIST, Optdigits and Chess datasets, the advantage of BGMC over the other tree-based methods is evident.

■ 6.6 Chapter Summary

In this chapter, we proposed a discriminative objective for the specific purpose of learning two tree-structured graphical models for classification. We observe that Discriminative Trees outperforms existing tree-based graphical model classifiers like TANs, especially in the absence of a large number of training examples. This is true for several reasons. First, our discriminative tree learning procedure is designed to optimize an approximation to the expectation of the log-likelihood ratio (6.18), while TAN is a generative procedure. Thus, if the intended purpose is known (e.g., in [207] the task was prediction), we can learn graphical models differently and often, more effectively for the task at hand. Secondly, we allowed the learned structures of the two models to be distinct, and each model is dependent on data with *both* positive and negative

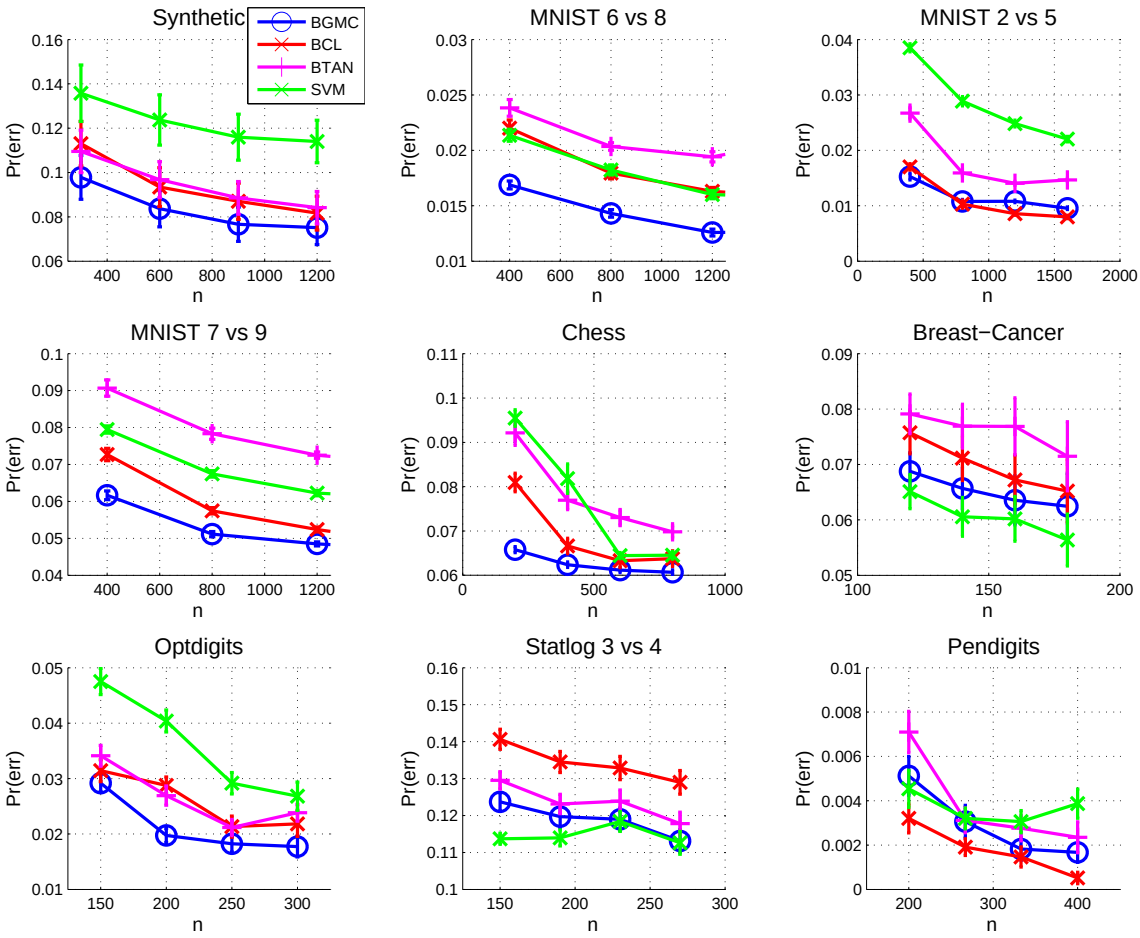

Figure 6.8. Pr(err) against n, the number of training samples, for various datasets using Boosted Graphical Model Classifiers (BGMC, blue $\circ$), Boosted Chow-Liu (BCL, red $\times$), Boosted TAN (BTAN, magenta $+$) and SVM with quadratic kernel (green $\times$). In all cases, the performance of BGMC is superior to Boosted TAN.

labels. It is worth noting that the proposed discriminative tree learning procedure does not incur any computational overhead compared to existing tree-based methods.

We showed that the discriminative tree learning procedure can be adapted to the weighted case, and is thus amenable to the use the models resulting from this procedure as weak classifiers for boosting to learn thicker models, which have better modeling ability. This is what allows us to circumvent the intractable problem of having to find the maximum-likelihood parameters of loopy graphical models.

In addition to learning two graphical models specifically for the purpose of discrimination, the proposed method also provides a principled approach to learn which pairwise features (or edges) are the most salient for classification (akin to the methods described in [90]). Our method for sequentially learning optimal forests serves precisely this purpose and also provides a natural way to incorporate costs of adding edges. Furthermore, to learn more edges than in a tree, we used boosting in a novel way to learn more complex models for the purpose of classification. Indeed, at the end of T boosting iterations, we can precisely characterize the set of edges for the normalized versions of the boosted models (Proposition 6.7). We can use these pairwise features, together with the marginal features, as inputs to *any* standard classification algorithm. Finally, our empirical results on a variety of synthetic and real datasets adequately demonstrate that the forests, trees and thicker models learned serve as good classifiers.

Appendices for Chapter 6

■ 6.A Proof of Proposition 6.3

Proof. We use $\stackrel{\mathrm{c}}{=}$ to denote equality up to a constant. Now, we can simplify the objective in the optimization problem in (6.15a), namely $D(\widetilde{p}\,||\,\widehat{p}) - D(\widetilde{q}\,||\,\widehat{p})$:

$$\stackrel{\mathrm{c}}{=} \sum_{\mathbf{x}\in\mathcal{X}^d} (\widetilde{q}(\mathbf{x}) - \widetilde{p}(\mathbf{x})) \log \left[\prod_{i\in V} \widehat{p}_i(x_i) \prod_{(i,j)\in E} \frac{\widehat{p}_{i,j}(x_i,x_j)}{\widehat{p}_i(x_i)\widehat{p}_i(x_j)} \right], \tag{6.30}$$

$$\stackrel{\mathrm{c}}{=} \sum_{\mathbf{x}\in\mathcal{X}^d} (\widetilde{q}(\mathbf{x}) - \widetilde{p}(\mathbf{x})) \sum_{(i,j)\in E} \log \left[\frac{\widetilde{p}_{i,j}(x_i,x_j)}{\widetilde{p}_i(x_i)\widetilde{p}_j(x_j)} \right], \tag{6.31}$$

$$= \sum_{(i,j)\in E} \sum_{(x_i,x_j)\mathcal{X}^2} (\widetilde{q}_{i,j}(x_i,x_j) - \widetilde{p}_{i,j}(x_i,x_j)) \log \left[\frac{\widetilde{p}_{i,j}(x_i,x_j)}{\widetilde{p}_i(x_i)\widetilde{p}_j(x_j)} \right], \tag{6.32}$$

where (6.30) follows from the fact that $\widehat{p}$ is a tree-structured distribution [and hence factorizes as (2.95)] and (6.31) follows from marginal consistency and the fact that we are optimizing only over the edge set of $\widehat{p}$ and thus the marginals can be dropped from the optimization. The final equality in (6.32), derived using (6.12) and (6.13), shows that we need to optimize over all tree structures with edge weights given by the expression in (6.16). □

■ 6.B Proof of Proposition 6.7

Proof. This result holds even when the $\widehat{p}_t$ are not trees, and the proof is straightforward. In general, a (everywhere non-zero) distribution p is Markov [127] with respect to some edge set E if and only if

$$\log p(\mathbf{x}) \stackrel{c}{=} \sum_{(i,j)\in E} \theta_{ij}\phi_{ij}(x_i, x_j) + \sum_{i\in V} \theta_i \phi_i(x_i) \tag{6.33}$$

for some constants θ and functions ϕ. This means that each tree model $\widehat{p}_t$ can be written as

$$\log \widehat{p}_t(\mathbf{x}) \stackrel{c}{=} \sum_{(i,j)\in E_{\widehat{p}_t}} \theta^t_{ij}\phi^t_{ij}(x_i, x_j) + \sum_{i\in V} \theta^t_i \phi^t_i(x_i). \tag{6.34}$$

Let $E := \bigcup_{t=1}^T E_{\widehat{p}_t}$ be the union of the edge sets after T boosting iterations. Then $\log \widehat{p}_*(\mathbf{x})$ is equal (up to constants) to

$$\sum_{t=1}^T \alpha_t \left(\sum_{(i,j)\in E_{\widehat{p}_t}} \theta^t_{ij}\phi^t_{ij}(x_i, x_j) \; + \sum_{i\in V} \theta^t_i \phi^t_i(x_i) \right), \tag{6.35}$$

$$= \sum_{(i,j)\in E} \left(\sum_{t=1}^T \alpha_t \theta^t_{ij}\phi^t_{ij}(x_i, x_j) \right) + \sum_{i\in V} \left(\sum_{t=1}^T \alpha_t \theta^t_i \phi^t_i(x_i) \right), \tag{6.36}$$

where in we interpret the right hand side of the last equality as $\theta^t_{ij} = 0$ if and only if $(i,j) \notin E_{\widehat{p}_t}$. This is seen to be of the same form as (6.33) – to see this, define the functions

$$\xi_{ij}(x_i, x_j) := \sum_{t=1}^T \alpha_t \theta^t_{ij}\phi^t_{ij}(x_i, x_j), \tag{6.37a}$$

$$\xi_i(x_i) := \sum_{t=1}^T \alpha_t \theta^t_i \phi^t_i(x_i), \tag{6.37b}$$

so that $\log \widehat{p}_*(\mathbf{x}) \stackrel{c}{=} \sum_{(i,j)\in E} \xi_{ij}(x_i, x_j) + \sum_{i\in V} \xi_i(x_i)$. By the Hammersley-Clifford Theorem [91], we have proven the desired Markov property. □

Chapter 7

High-Dimensional Salient Subset Recovery

■ 7.1 Introduction

CONSIDER the following scenario which was used as a motivating example in Chapter 1: There are 1000 children participating in a longitudinal study in childhood asthma of which 500 of them are asthmatic and the other 500 are not. 10^6 measurements of possibly relevant features (*e.g.*, genetic, environmental, physiological) are taken from each child but only a very small subset of these (say 30) is useful in predicting whether the child has asthma. The correct identification and subsequent interpretation of this *salient* subset is important to clinicians for assessing the susceptibility of other children to asthma. We expect that by focusing only on the 30 salient features, we can improve discrimination and reduce the computational cost in coming up with a decision rule. Indeed, when the salient set is small compared to the overall dimension (10^6), we also expect to be able to estimate the salient set with a small number of samples.

In this chapter, we build on the idea of salient feature extraction from the previous chapter to derive and study conditions under which we can asymptotically recover the salient feature subset for distinguishing between two probability models from i.i.d. samples. Identifying the salient set improves discrimination performance and reduces complexity. The focus in this chapter is similar to Chapter 5 and is focused on the high-dimensional regime where the number of variables d, the number of salient variables k and the number of samples n all grow. The definition of saliency is motivated by error exponents in a binary hypothesis test (cf. Section 2.2.3) and is stated in terms of relative entropies. Intuitively, we expect that if k and d do not grow too quickly with n, then consistent recovery is possible in high-dimensions.

As with the rest of the thesis, we adopt an information-theoretic perspective. More specifically, we utilize ideas from the method of types and typicality (described in Section 2.2.1) to prove achievability statements and Fano's inequality (described in Section 2.1.6) to prove converses. Furthermore, we we define the notion of *saliency* for distinguishing between two probability distributions by appealing to the Chernoff-Stein lemma in a binary hypothesis test under the Neyman-Pearson framework. We show that this definition of saliency can also be motivated by the same hypothesis testing

problem under the Bayesian framework, in which the overall error probability is to be minimized. For the asthma example, intuitively, a feature is salient if it is useful in predicting whether a child has asthma and we also expect the number of salient features to be very small. Also, conditioned on the salient features, the non-salient ones should not contribute to the distinguishability of the classes. Our mathematical model and definition of saliency in terms of the KL-divergence (or Chernoff information) captures this intuition.

There are three main contributions in this chapter. Firstly, we provide *sufficient* conditions on the scaling of the model parameters (n, d, k) so that the salient set is recoverable asymptotically. Secondly, by modeling the salient set as a uniform random variable (over all sets of size k), we derive a *necessary* condition that *any* decoder must satisfy in order to recover the salient set. Thirdly, in light of the fact that the exhaustive search decoder is computationally infeasible, we examine the case in which the underlying distributions are Markov on trees and derive efficient tree-based combinatorial optimization algorithms to search for the salient set.

The literature on feature subset selection (or variable extraction) is vast. See [90] (and references therein) for a thorough review of the field. The traditional methods include the so-called *wrapper* (assessing different subsets for their usefulness in predicting the class) and *filter* (ranking) methods. Our definition of saliency is related to the minimum-redundancy, maximum-relevancy model in [155], the notion of Markov blankets in [118] and the notion of sufficiency by Kullback [125] and is expressed using information-theoretic quantities motivated by hypothesis testing. The algorithm suggested in [147] shows that the generalization error remains small even in the presence of a large number of irrelevant features, but this chapter focuses on exact recovery of the salient set given scaling laws on (n, d, k). This work is also related to [208] and [72] on sparsity pattern recovery (or compressed sensing) but does not assume the linear observation model. Rather, samples are drawn from two arbitrary discrete multivariate probability distributions so this can also be considered as a nonparametric model selection problem.

The rest of this chapter is organized as follows: In Section 7.2, we define the notation used in this chapter and state the definition of achievability. In Section 7.3, we derive necessary and sufficient conditions for asymptotic salient subset recovery. In Section 7.4, we demonstrate that for special classes of tree-structured distributions, the recovery of the salient subset can be performed very efficiently. Conclusions are provided in Section 7.5. Most of the proofs of the statements are deferred to the appendices of this chapter.

■ 7.2 Notation, System Model and Definitions

Let $\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}$ be two sequences of distributions where $P^{(d)}, Q^{(d)} \in \mathcal{P}(\mathcal{X}^d)$, are the distributions of d-dimensional random vectors $\mathbf{x}, \mathbf{y}$ respectively. For a vector $\mathbf{x} \in \mathcal{X}^d$, $\mathbf{x}_A$ is the length-$|A|$ subvector that consists of the elements in A. Let $A^c := V_d \setminus A$.

In addition, let $V_d := \{1, \ldots, d\}$ be the *index set* and for a subset $A \subset V_d$, let $P_A^{(d)}$ be the marginal of the subset of random variables in A, i.e., the random vector $\mathbf{x}_A$. Each index $i \in V_d$, associated to marginals $(P_i^{(d)}, Q_i^{(d)})$, will be generically called a *feature*.

We assume that for each pair $(P^{(d)}, Q^{(d)})$, there exists a set of n i.i.d. samples $(\mathbf{x}^n, \mathbf{y}^n) := (\{\mathbf{x}^{(l)}\}_{l=1}^n, \{\mathbf{y}^{(l)}\}_{l=1}^n)$ drawn from $P^{(d)} \times Q^{(d)}$. Each sample $\mathbf{x}^{(l)}$ (and also $\mathbf{y}^{(l)}$) belongs to $\mathcal{X}^d$. Our goal is to distinguish between $P^{(d)}$ and $Q^{(d)}$ using the samples. Note that for each d, this setup is analogous to binary classification where one does not have access to the underlying distributions but only samples from the distribution. We suppress the dependence of $(\mathbf{x}^n, \mathbf{y}^n)$ on the dimensionality d when the lengths of the vectors are clear from the context.

■ 7.2.1 Definition of The Salient Set of Features

We now motivate the notion of saliency (and the salient set) by considering the following binary hypothesis testing problem. There are n i.i.d. d-dimensional samples $\mathbf{z}^n := \{\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(n)}\}$ drawn from either $P^{(d)}$ or $Q^{(d)}$, i.e.,

$$H_0 : \mathbf{z}^n \overset{\text{i.i.d.}}{\sim} P^{(d)}, \qquad H_1 : \mathbf{z}^n \overset{\text{i.i.d.}}{\sim} Q^{(d)}. \tag{7.1}$$

The Chernoff-Stein lemma (Lemma 2.22) says that the error exponent for (7.1) under the Neyman-Pearson formulation is $D(P^{(d)} \,||\, Q^{(d)})$. More precisely, if the probability of false alarm $P_{\text{FA}} = \Pr(\hat{H}_1 | H_0)$ is kept below α, then the probability of mis-detection $P_{\text{M}} = \Pr(\hat{H}_0 | H_1)$ tends to zero exponentially fast as $n \to \infty$ with exponent given by $D(P^{(d)} \,||\, Q^{(d)})$.

In the Bayesian formulation, we seek to minimize the overall probability of error $\Pr(\text{err}) = \Pr(H_0)P_{\text{FA}} + \Pr(H_1)P_{\text{M}}$, where $\Pr(H_0)$ and $\Pr(H_1)$ are the prior probabilities of hypotheses H_0 and H_1 respectively. It is known from Lemma 2.23 that in this case, the error exponent governing the rate of decay of $\Pr(\text{err})$ with the sample size n is the *Chernoff information* between $P^{(d)}$ and $Q^{(d)}$, i.e., $D^*(P^{(d)}, Q^{(d)})$ defined in (2.70). Similar to the KL-divergence, $D^*(P^{(d)}, Q^{(d)})$ is a measure of the separability of the distributions. It is a symmetric quantity in the distributions but is still not a metric. Given the form of the error exponents for the Neyman-Pearson and Bayesian setups, we would like to identify a size-k subset of features $S_d \subset V_d$ that "maximally distinguishes" between $P^{(d)}$ and $Q^{(d)}$. This motivates the following definitions:

Definition 7.1. (KL-divergence Salient Set) *A subset $S_d \subset V_d$ of size k is* KL-divergence salient *(or simply salient) if*

$$D(P^{(d)} \,||\, Q^{(d)}) = D(P_{S_d}^{(d)} \,||\, Q_{S_d}^{(d)}), \tag{7.2}$$

Thus, conditioned on the variables in the salient set S_d (with $|S_d| = k$ for some $1 \leq k \leq d$), the variables in the complement S_d^c do not contribute to the distinguishability (in terms of the KL-divergence) of $P^{(d)}$ and $Q^{(d)}$.

Definition 7.2. (Chernoff information Salient Set) *A subset $S_d \subset V_d$ of size k is* Chernoff information salient *if*

$$D^*(P^{(d)}, Q^{(d)}) = D^*(P^{(d)}_{S_d}, Q^{(d)}_{S_d}), \tag{7.3}$$

Thus, given the variables in S_d, the remaining variables in S_d^c do not contribute to the Chernoff information defined in (2.70). A natural question to ask is whether the two definitions above are equivalent. We claim the following lemma.

Lemma 7.1. (Equivalence of Saliency Definitions) *For a subset $S_d \subset V_d$ of size k, the following are equivalent:*

S1: S_d is KL-divergence salient.

S2: S_d is Chernoff information salient.

S3: $P^{(d)}$ and $Q^{(d)}$ admit the following decompositions into the S_d marginals and the conditional distribution of S_d^c given S_d:

$$P^{(d)} = P^{(d)}_{S_d} \cdot W_{S_d^c|S_d}, \qquad Q^{(d)} = Q^{(d)}_{S_d} \cdot W_{S_d^c|S_d}. \tag{7.4}$$

Proof. Lemma 7.1 is proved using Hölder's inequality and Jensen's inequality. See Appendix 7.A for the details. □

Observe from (7.4) that the conditionals $W_{S_d^c|S_d}$ of both models are identical. Consequently, the *likelihood ratio test* (LRT) between $P^{(d)}$ and $Q^{(d)}$ depends solely on the marginals of the salient set S_d, i.e.,

$$\frac{1}{n}\sum_{l=1}^{n} \log \frac{P^{(d)}(\mathbf{z}^{(l)})}{Q^{(d)}(\mathbf{z}^{(l)})} = \frac{1}{n}\sum_{l=1}^{n} \log \frac{P^{(d)}_{S_d}(\mathbf{z}^{(l)}_{S_d})}{Q^{(d)}_{S_d}(\mathbf{z}^{(l)}_{S_d})} \underset{\hat{H}=H_1}{\overset{\hat{H}=H_0}{\gtrless}} \gamma_n, \tag{7.5}$$

is the most powerful test of fixed size α for threshold γ_n.[1] Also, the inclusion of *any* non-salient subset of features $B \subset S_d^c$ keeps the likelihood ratio in (7.5) exactly the same, i.e.,

$$\frac{P^{(d)}_{S_d}}{Q^{(d)}_{S_d}} = \frac{P^{(d)}_{S_d \cup B}}{Q^{(d)}_{S_d \cup B}}. \tag{7.6}$$

Moreover, correctly identifying S_d from the set of samples $(\mathbf{x}^n, \mathbf{y}^n)$ results in a reduction in the number of relevant features, which is advantageous for the design of parsimonious and efficient binary classifiers.

Because of this equivalence of definitions of saliency (in terms of the Chernoff-Stein exponent and the Chernoff information), if we have successfully identified the salient set

[1] We have implicitly assumed that the distributions $P^{(d)}, Q^{(d)}$ are nowhere zero and consequently the conditional $W_{S_d^c|S_d}$ is also nowhere zero.

in (7.2), we have also found the subset that maximizes the error exponent associated to the overall probability of error Pr(err). In our results, we find that the characterization of saliency in terms of (7.2) is more convenient than its equivalent characterization in (7.3). Finally, we emphasize that the number of variables and the number of salient variables $k = |S_d|$ can grow as functions of n, i.e., $d = d(n), k = k(n)$. In the sequel, we provide necessary and sufficient conditions for the asymptotic recovery of S_d as the model parameters scale, i.e., when (n, d, k) all grow.

■ 7.2.2 Definition of Achievability

Let $\mathfrak{S}_{k,d} := \{A : A \subset V_d, |A| = k\}$ be the set of cardinality-k subsets in V_d. A *decoder* is a set-valued function ψ_n that maps the samples to a subset of size k, i.e., $\psi_n : (\mathcal{X}^d)^n \times (\mathcal{X}^d)^n \to \mathfrak{S}_{k,d}$. Note in this chapter that the decoder is given k, i.e., the cardinality of the salient set. In the following, we use the notation $\widehat{P}^{(d)}, \widehat{Q}^{(d)}$ to denote the *empirical distributions* (or types) of $\mathbf{x}^n, \mathbf{y}^n$ respectively. Note that as usual, we drop the dependence of $\widehat{P}^{(d)}$ on the samples $\mathbf{x}^n$ and also on n for brevity. All the quantities with hats depend on $(\mathbf{x}^n, \mathbf{y}^n)$.

Definition 7.3. (Exhaustive Search Decoder) *The* exhaustive search decoder *(ESD)* $\psi_n^* : (\mathcal{X}^d)^n \times (\mathcal{X}^d)^n \to \mathfrak{S}_{k,d}$ *is given as*

$$\psi_n^*(\mathbf{x}^n, \mathbf{y}^n) \in \operatorname*{argmax}_{S_d' \in \mathfrak{S}_{k,d}} D(\widehat{P}^{(d)}_{S_d'} \,||\, \widehat{Q}^{(d)}_{S_d'}). \tag{7.7}$$

where $\widehat{P}^{(d)}_{S_d'}$ *is the marginal of the empirical distribution of the variables in* S_d'*. If the* argmax *in* (7.7) *is not unique, output any set* $S_d' \in \mathfrak{S}_{k,d}$ *that maximizes the objective.*

We remark that, in practice, the ESD is computationally infeasible for large d and k since it has to compute the *empirical KL-divergence* $D(\widehat{P}^{(d)}_{S_d'} \,||\, \widehat{Q}^{(d)}_{S_d'})$ for all subsets in $\mathfrak{S}_{k,d}$. In Section 7.4, we analyze how to reduce the complexity of (7.7) for tree distributions. Nonetheless, the ESD is *consistent* for fixed d and k. That is, as $n \to \infty$, the probability that a non-salient set is selected by ψ_n^* tends to zero. We provide the exponential rate of decay in Section 7.3.2. Let $\mathbb{P}^n := (P^{(d)} \times Q^{(d)})^n$ denote the n-fold product probability measure of $P^{(d)} \times Q^{(d)}$.

Definition 7.4. (Achievability) *The sequence of model parameters* $\{(n, d, k)\}_{n \in \mathbb{N}}$ *is* achievable *for the sequence of distributions* $\{P^{(d)}, Q^{(d)} \in \mathcal{P}(\mathcal{X}^d)\}_{d \in \mathbb{N}}$ *if there exists a sequence of decoders* $\{\psi_n\}$ *such that to every* $\epsilon > 0$*, there exists a* $N_\epsilon \in \mathbb{N}$ *for which the error probability*

$$p_n(\psi_n) := \mathbb{P}^n(\psi_n(\mathbf{x}^n, \mathbf{y}^n) \neq S_d) < \epsilon, \qquad \forall\, n > N_\epsilon. \tag{7.8}$$

Thus, if $\{(n, d, k)\}_{n \in \mathbb{N}}$ *is achievable,* $\lim_n p_n(\psi_n) = 0$*. In* (7.8)*,* $(\mathbf{x}^n, \mathbf{y}^n)$ *is a set of* n *i.i.d. samples drawn from* $P^{(d)} \times Q^{(d)}$*.*

In the sequel, we provide achievability conditions for the ESD.

■ 7.3 Conditions for the High-Dimensional Recovery of Salient Subsets

In this section, we state three assumptions on the sequence of distributions $\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}$ such that under some specified scaling laws, the triple of model parameters (n, d, k) is achievable with the ESD as defined in (7.8). We provide both *positive* (achievability) and *negative* (converse) sample complexity results under these assumptions. That is, we state when (7.8) holds and also when the sequence $p_n(\psi_n)$ is uniformly bounded away from zero.

■ 7.3.1 Assumptions on the Distributions

In order to state our results, we assume that the sequence of probability distributions $\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}$ satisfy the following three conditions:

A1: (*Saliency*) For each pair of distributions $P^{(d)}, Q^{(d)}$, there exists a salient set $S_d \subset V_d$ of cardinality k such that (7.2) (or equivalently (7.3)) holds.

A2: (η-*Distinguishability*) There exists a constant $\eta > 0$, independent of (n, d, k), such that for all $d \in \mathbb{N}$ and for all non-salient subsets $S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}$, we have

$$D(P^{(d)}_{S_d} \,||\, Q^{(d)}_{S_d}) - D(P^{(d)}_{S'_d} \,||\, Q^{(d)}_{S'_d}) \geq \eta > 0. \tag{7.9}$$

A3: (L-*Boundedness of the Likelihood Ratio*) There exists a $L \in (0, \infty)$, independent of (n, d, k), such that for all $d \in \mathbb{N}$, we have

$$\log\left[\frac{P^{(d)}_{S_d}(\mathbf{x}_{S_d})}{Q^{(d)}_{S_d}(\mathbf{x}_{S_d}}\right] \in [-L, L] \tag{7.10}$$

for all length-k vectors $\mathbf{x}_{S_d} \in \mathcal{X}^k$.

Assumption A1 pertains to the existence of a salient set. Assumption A2 allows us to employ the large deviation principle [59] to quantify error probabilities. This is because all non-salient subsets $S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}$ are such that their divergences are uniformly smaller than the divergences on S_d, the salient set. Thus, for each d, the associated salient set S_d is *unique* and the error probability of selecting any non-salient set S'_d decays exponentially. A2 together with A3, a regularity condition, allows us to prove that the *exponents* of all the possible error events are *uniformly* bounded away from zero. In the next subsection, we formally define the notion of an error exponent for the recovery of salient subsets.

■ 7.3.2 Fixed Number of Variables d and Salient Variables k

In this section, we consider the situation when d and k are constant. This provides key insights for developing achievability results when (n, d, k) scale. Under this scenario,

we have a large deviations principle for the error event in (7.8). We define the *error exponent* for the ESD ψ_n^* as

$$C(P^{(d)}, Q^{(d)}) := -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}^n(\psi_n^*(\mathbf{x}^n, \mathbf{y}^n) \neq S_d). \tag{7.11}$$

Let $J_{S'_d|S_d}$ be the *error rate* at which the non-salient set $S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}$ is selected by the ESD, i.e.,

$$J_{S'_d|S_d} := -\lim_{n\to\infty} \frac{1}{n} \log \mathbb{P}^n \left(\psi_n^*(\mathbf{x}^n, \mathbf{y}^n) = S'_d\right). \tag{7.12}$$

For each $S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}$, also define the set of distributions

$$\Gamma_{S'_d|S_d} := \left\{(P, Q) \in \mathcal{P}(\mathcal{X}^{2|S_d \cup S'_d|}) : D(P_{S_d} \,||\, Q_{S_d}) = D(P_{S'_d} \,||\, Q_{S'_d})\right\}. \tag{7.13}$$

Proposition 7.2. (Error Exponent as Minimum Error Rate) *Assume that the ESD ψ_n^* is used. If d and k are constant, then the error exponent* (7.11) *is given as*

$$C(P^{(d)}, Q^{(d)}) = \min_{S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}} J_{S'_d|S_d}, \tag{7.14}$$

where the error rate $J_{S'_d|S_d}$, defined in (7.12)*, is*

$$J_{S'_d|S_d} = \min_{\nu \in \Gamma_{S'_d|S_d}} D(\nu \,||\, P^{(d)}_{S_d \cup S'_d} \times Q^{(d)}_{S_d \cup S'_d}). \tag{7.15}$$

Furthermore if A2 holds, $C(P^{(d)}, Q^{(d)}) > 0$ and hence the error probability in (7.8) *decays exponentially fast in n.*

Proof. This result is proved using Sanov's Theorem and the contraction principle in large deviations. See Appendix 7.B. □

■ 7.3.3 An Achievability Result for the High-Dimensional Case

We now consider the high-dimensional scenario when (n, d, k) all scale and we have a sequence of salient set recovery problems indexed by n for the probability models $\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}$. Thus, $d = d(n)$ and $k = k(n)$ and we are searching for how such dependencies must behave (scale) such that we have achievability in the sense of Definition 7.4. This is of interest since this regime (typically $d \gg n, k$) is most applicable to many practical problems and modern datasets such as the motivating example in the introduction. Before stating our main theorem, we define the *greatest lower bound (g.l.b.) of the error exponents* as

$$B := B(\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}) := \inf_{d\in\mathbb{N}} C(P^{(d)}, Q^{(d)}), \tag{7.16}$$

where $C(P^{(d)}, Q^{(d)})$ is given in (7.14). Clearly, $B \geq 0$ by the non-negativity of the KL-divergence. In fact, we prove that $B > 0$ under assumptions A1 – A3, i.e., the exponents in (7.14) are *uniformly* bounded away from zero. For $\epsilon > 0$, define the functions

$$g_1(k, \epsilon) := \exp\left(\frac{2k \log|\mathcal{X}|}{1-\epsilon}\right), \qquad g_2(d, k) := \frac{k}{B} \log\left(\frac{d-k}{k}\right). \tag{7.17}$$

Theorem 7.3 (Main Result: Achievability)**.** *Assume that A1 – A3 hold for the sequence of distributions* $\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}$*. If there exists an* $\epsilon > 0$ *and an* $N \in \mathbb{N}$ *such that*

$$n > \max\{g_1(k,\epsilon), g_2(d,k)\}, \quad \forall\, n > N, \tag{7.18}$$

then

$$p_n(\psi_n^*) = O(\exp(-nc)) \tag{7.19}$$

where the exponent

$$c := B - \limsup_{n\to\infty} \frac{k}{n} \log \frac{d-k}{k} > 0. \tag{7.20}$$

Proof. See Appendix 7.C for the proof. □

In other words, the sequence $\{(n,d,k)\}_{n\in\mathbb{N}}$ of parameters is achievable if (7.18) holds. Furthermore, the exhaustive search decoder in (7.7) achieves the scaling law in (7.18).

The key elements in proof include applications of large deviations bounds (*e.g.*, Sanov's theorem), asymptotic behavior of binomial coefficients and most crucially demonstrating the positivity of the g.l.b. of the error exponents B defined in (7.16). We now discuss the ramifications of Theorem 7.3.

Firstly, $n > g_1(k,\epsilon)$ means that k, the number of salient features, is only allowed to grow logarithmically in n. Secondly, $n > g_2(d,k)$ means that if k is a constant, the number of redundant features $|S_d^c| = d - k$ can grow exponentially with n, and p_n still tends to zero exponentially fast. This means that recovery of S_d is asymptotically possible even if the data dimension is extremely large (compared to n) but the number of salient ones remain a small fraction of the total number d. We state this observation formally as a corollary of Theorem 7.3.

Corollary 7.4. (Achievability for constant k) *Assume A1 – A3. Let* $k = k_0$ *be a constant and fix* $R < R_1 := B/k_0$*. Then if there exists a* $N \in \mathbb{N}$ *such that*

$$n > \frac{\log d}{R}, \qquad \forall\, n > N, \tag{7.21}$$

then the error probability obeys $p_n(\psi_n^*) = O(\exp(-nc'))$*, where the exponent is* $c' := B - k_0 R$*.*

This result means that we can recover the salient set even though the number of variables d is much larger than (exponential in) in the number of samples n as in the asthma example.

■ 7.3.4 A Converse Result for the High-Dimensional Case

In this section, we state a converse theorem (and several useful corollaries) for the high-dimensional case. Specifically, we establish a condition on the scaling of (n,d,k) so that the probability of error is uniformly bounded away from zero for *any* decoder. In

order to apply standard proof techniques (such as Fano's inequality) for converses that apply to all possible decoders ψ_n, we consider the following slightly modified problem setup where S_d is random and not fixed as was in Theorem 7.3. More precisely, let $\{\widetilde{P}^{(d)}, \widetilde{Q}^{(d)}\}_{d\in\mathbb{N}}$ be a fixed sequence of distributions, where $\widetilde{P}^{(d)}, \widetilde{Q}^{(d)} \in \mathcal{P}(\mathcal{X}^d)$. We assume that this sequence of distributions satisfies A1 – A3, namely there exists a salient set $\widetilde{S}_d \in \mathfrak{S}_{k,d}$ such that $\widetilde{P}^{(d)}, \widetilde{Q}^{(d)}$ satisfies (7.2) for all d.

Let Π be a permutation of V_d chosen uniformly at random, i.e., $\Pr(\Pi = \pi) = 1/(d!)$ for any permutation operator $\pi : V_d \to V_d$. Define the sequence of distributions $\{P^{(d)}, Q^{(d)}\}_{d\in\mathbb{N}}$ as

$$\pi \sim \Pi, \qquad P^{(d)} := \widetilde{P}^{(d)}_{\pi}, \qquad Q^{(d)} := \widetilde{Q}^{(d)}_{\pi}. \tag{7.22}$$

Put simply, we permute the indices in $\widetilde{P}^{(d)}, \widetilde{Q}^{(d)}$ (according to the realization of Π) to get $P^{(d)}, Q^{(d)}$, i.e.,

$$P^{(d)}(x_1 \dots x_d) := \widetilde{P}^{(d)}(x_{\pi(1)} \dots x_{\pi(d)}). \tag{7.23}$$

Thus, once π has been drawn, the distributions $P^{(d)}$ and $Q^{(d)}$ of the random vectors $\mathbf{x}$ and $\mathbf{y}$ are completely determined. Clearly the salient sets S_d are drawn *uniformly at random (u.a.r.)* from $\mathfrak{S}_{k,d}$ and we have the Markov chain:

$$S_d \xrightarrow{\varphi_n} (\mathbf{x}^n, \mathbf{y}^n) \xrightarrow{\psi_n} \widehat{S}_d, \tag{7.24}$$

where the length-d random vectors $(\mathbf{x}, \mathbf{y}) \sim P^{(d)} \times Q^{(d)}$ and $\widehat{S}_d$ is any estimate of S_d. Also, φ_n is the *encoder* given by the random draw of π and (7.22). ψ_n is the decoder defined in Section 7.2.2. We denote the entropy of a random vector $\mathbf{z}$ with pmf P as $H(\mathbf{z}) = H(P)$ and the conditional entropy of $\mathbf{z}_A$ given $\mathbf{z}_B$ as $H(\mathbf{z}_A|\mathbf{z}_B) = H(P_{A|B})$.

Theorem 7.5. (Converse) *Assume that the salient sets $\{S_d\}_{d\in\mathbb{N}}$ are drawn u.a.r. and encoded as in* (7.22)*. If*

$$n < \frac{\lambda k \log(\frac{d}{k})}{H(P^{(d)}) + H(Q^{(d)})}, \quad \textit{for some } \lambda \in (0,1), \tag{7.25}$$

then $p_n(\psi_n) \geq 1 - \lambda$ for any decoder ψ_n.

Proof. The converse is proven using Fano's inequality. See Appendix 7.E for the proof of this result. □

Note from (7.25) that if the non-salient set S_d^c consists of uniform random variables independent of those in S_d then $H(P^{(d)}) = O(d)$ and the bound is never satisfied. However, the converse is interesting and useful if we consider distributions with additional structure on their entropies. In particular, we assume that most of the non-salient variables are redundant (or processed) versions of the salient ones. Again appealing to the asthma example in the introduction, there could be two features in the dataset "body mass index" (in S_d) and "is obese" (in S_d^c). These two features capture the same basic information and are thus redundant, but the former may be more informative to the asthma hypothesis.

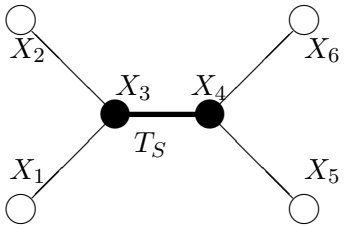

Figure 7.1. $T_S = (V(T_S), E(T_S))$ (in bold) is a subtree in T and its nodes (in black) comprise the salient set $S = \{3, 4\}$.

Corollary 7.6. (Converse with Bound on Conditional Entropy) *If there exists a* $M < \infty$ *such that*

$$\max\left\{H(P^{(d)}_{S_d^c|S_d}), H(Q^{(d)}_{S_d^c|S_d})\right\} \leq Mk \tag{7.26}$$

for all $d \in \mathbb{N}$, *and*

$$n < \frac{\lambda \log(\frac{d}{k})}{2(M + \log|\mathcal{X}|)}, \quad \textit{for some } \lambda \in (0,1), \tag{7.27}$$

then $p_n(\psi_n) \geq 1 - \lambda$ *for any decoder* ψ_n.

Corollary 7.7. (Converse for constant k) *Assume the setup in Corollary 7.6. Fix* $R > R_2 := 2(M + \log|\mathcal{X}|)$. *Then if* k *is a constant and if there exists an* $N \in \mathbb{N}$ *such that* $n < (\log d)/R$ *for all* $n > N$, *then there exist a* $\delta > 0$ *such that error probability* $p_n(\psi_n) \geq \delta$ *for all decoders* ψ_n.

We previously showed (cf. Corollary 7.4) that there is a rate of growth R_1 so that achievability holds if $R < R_1$. Corollary 7.7 says that, under the specified conditions, there is also another rate R_2 so that if $R > R_2$, recovery of S_d is no longer possible.

■ 7.4 Specialization to Tree Distributions

As mentioned previously, the ESD in (7.7) is computationally prohibitive. In this section, we assume Markov structure on the distributions and devise an efficient algorithm to reduce the computational complexity of the decoder. To do so, for each d and k, assume the following:

A4: (*Markov tree*) The distributions $P := P^{(d)}, Q := Q^{(d)}$ are undirected graphical models [127]. More specifically, P, Q are *Markov on a common tree* $T = (V(T), E(T))$, where $V(T) = \{1, \ldots, d\}$ is the *vertex set* and $E(T) \subset \binom{V}{2}$ is the *edge set*. That is, P, Q admit the factorization in (2.95) where their edge sets are common.

A5: (*Subtree*) The salient set $S := S_d$ is such that P_S, Q_S are Markov on a *common (connected) subtree* $T_S = (V(T_S), E(T_S))$ in T. See Fig. 7.1.

Note that $E_S \subset E$ has to be a connected edge set so that the marginal distribution P_S and Q_S remain Markov on a common tree $T_S = (V, E_E)$. Otherwise, additional edges may be introduced when the variables in S^c are marginalized out [127]. For example, if S is a singleton set and, in particular, is the sole non-leaf node in a star-structured graphical model, marginalizing over S yields, in general, a fully-connected graphical model. Under A4 and A5, the KL-divergence decomposes as:

$$D(P \,||\, Q) = \sum_{i \in V(T)} D_i + \sum_{(i,j) \in E(T)} W_{i,j}, \tag{7.28}$$

where $D_i := D(P_i \,||\, Q_i)$ is the KL-divergence of the marginals and the *weights* $W_{i,j} := D_{i,j} - D_i - D_j$. A similar decomposition holds for $D(P_S \,||\, Q_S)$ with $V(T_S), E(T_S)$ in (7.28) in place of $V(T), E(T)$. Let $\mathcal{T}_k(T)$ be the set of subtrees with $k < d$ vertices in T, a tree with d vertices. We now describe an efficient algorithm to learn S when T is unknown.

Firstly, using the samples $(\mathbf{x}^n, \mathbf{y}^n)$, learn a *single* Chow-Liu tree model T_{ML} using the sum of the empirical mutual information quantities $\{I(\widehat{P}_{i,j}) + I(\widehat{Q}_{i,j})\}$ as the edge weights. It is known that the Chow-Liu max-weight spanning tree algorithm is consistent and large deviations rates have also been studied (Chapter 3). Secondly, solve the following optimization:

$$T_k^* = \operatorname*{argmax}_{T_k' \in \mathcal{T}_k(T_{\mathrm{ML}})} \sum_{i \in V(T_k')} \widehat{D}_i + \sum_{(i,j) \in E(T_k')} \widehat{W}_{i,j}, \tag{7.29}$$

where $\widehat{D}_i$ and $\widehat{W}_{i,j}$ are the empirical versions of D_i and $W_{i,j}$ respectively. In (7.29), the sum of the node and edge weights over all size-k subtrees in T_{ML} is maximized. The problem in (7.29) is known as the k-CARD TREE problem [22, 78] and it runs in time $O(dk^2)$ using a dynamic programming procedure on trees. Thirdly, let the estimate of the salient set be the vertex set of T_k^*, i.e, $\psi_n(\mathbf{x}^n, \mathbf{y}^n) := V(T_k^*)$.

Proposition 7.8. (Complexity Reduction for Trees) *Assume that A4 and A5 hold. Then if k, d are constant, the algorithm described above to estimate S is consistent. Moreover, the time complexity is $O(dk^2 + nd^2|\mathcal{X}|^2)$.*

Proof. The proof can be found in Appendix 7.H. □

Hence, there are significant savings in computational complexity if the probability models P and Q are trees.

■ 7.5 Conclusion

In this chapter, we defined the notion of saliency and provided necessary and sufficient conditions for the asymptotic recovery of salient subsets in the high-dimensional regime. We also provided an computationally efficient algorithm for the search of the salient

set in the case when it is known that the true distributions are Markov on trees. We discuss possible extensions in Chapter 8.

Appendices for Chapter 7

■ 7.A Proof of Proposition 7.1

Proof. We will prove that (S3) $\Leftrightarrow$ (S1) $\Leftrightarrow$ (S2). Assuming (S1) holds, $D(P^{(d)} \,||\, Q^{(d)}) = D(P^{(d)}_{S_d} \,||\, Q^{(d)}_{S_d})$ implies that the conditional KL-divergence is identically zero, i.e,

$$D(P^{(d)}_{S_d^c|S_d} \,||\, Q^{(d)}_{S_d^c|S_d}) = 0. \tag{7.30}$$

Expanding the above expression yields the following:

$$\sum_{\mathbf{x}_{S_d}} P^{(d)}(\mathbf{x}_{S_d}) \sum_{\mathbf{x}_{S_d^c}} P^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d}) \log \frac{P^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d})}{Q^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d})} = 0. \tag{7.31}$$

From the positivity of the distributions and non-negativity of the KL-divergence, we have that

$$\sum_{\mathbf{x}_{S_d^c}} P^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d}) \log \frac{P^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d})}{Q^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d})} = 0, \tag{7.32}$$

for all $\mathbf{x}_{S_d} \in \mathcal{X}^k$. We conclude that

$$P^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d}) = Q^{(d)}_{S_d^c|S_d}(\mathbf{x}_{S_d^c}|\mathbf{x}_{S_d}), \quad \forall\, \mathbf{x}_{S_d} \in \mathcal{X}^k,\, \mathbf{x}_{S_d^c} \in \mathcal{X}^{d-k}, \tag{7.33}$$

which implies that the conditional distributions are identical. This proves (S3). The reverse implication is obvious.

Assume that S_d is KL-divergence salient (S1). Then from the above, we have (7.4). The Chernoff information is then given by

$$\begin{aligned}
D^*(P^{(d)}, Q^{(d)}) &= -\min_{t\in[0,1]} \log \left(\sum_{\mathbf{z}} (P^{(d)}(\mathbf{z}))^t (Q^{(d)}(\mathbf{z}))^{1-t} \right), \\
&= -\min_{t\in[0,1]} \log \left(\sum_{\mathbf{z}} (P^{(d)}_{S_d}(\mathbf{z}_{S_d}))^t (Q^{(d)}_{S_d}(\mathbf{z}_{S_d}))^{1-t} W_{S_d^c|S_d}(\mathbf{z}_{S_d^c}|\mathbf{z}_{S_d}) \right), \\
&= -\min_{t\in[0,1]} \log \left(\sum_{\mathbf{z}_{S_d}} (P^{(d)}_{S_d}(\mathbf{z}_{S_d}))^t (Q^{(d)}_{S_d}(\mathbf{z}_{S_d}))^{1-t} \sum_{\mathbf{z}_{S_d^c}} W_{S_d^c|S_d}(\mathbf{z}_{S_d^c}|\mathbf{z}_{S_d}) \right), \\
&= -\min_{t\in[0,1]} \log \left(\sum_{\mathbf{z}_{S_d}} (P^{(d)}_{S_d}(\mathbf{z}_{S_d}))^t (Q^{(d)}_{S_d}(\mathbf{z}_{S_d}))^{1-t} \right) = D^*(P^{(d)}_{S_d}, Q^{(d)}_{S_d}),
\end{aligned}$$

which proves that S_d is Chernoff information salient (S2). Now for the reverse implication, we claim the following lemma:

Lemma 7.9. *(*Monotonicity of Chernoff information) *For every set $A \subset V_d$, the Chernoff information satisfies*

$$D^*(P^{(d)}, Q^{(d)}) \geq D^*(P_A^{(d)}, Q_A^{(d)}), \tag{7.34}$$

with equality if and only if (7.4) *holds, i.e., the conditionals $P_{A^c|A}^{(d)}$ and $Q_{A^c|A}^{(d)}$ are identical.*

Assuming Lemma 7.9 and assuming that S_d is Chernoff information-salient, we have that $P^{(d)}$ and $Q^{(d)}$ satisfy (S3). Since (S3) $\Leftrightarrow$ (S2), this completes the proof of Lemma 7.1. It remains to prove Lemma 7.9. □

Proof. (of Lemma 7.9)
We drop the superscript (d) for notational simplicity. Then we have the following chain

$$\begin{aligned}
D^*(P,Q) &= -\min_{t\in[0,1]} \log\left(\sum_{\mathbf{z}} P(\mathbf{z})^t Q(\mathbf{z})^{1-t}\right), \\
&= -\min_{t\in[0,1]} \log\left(\sum_{\mathbf{z}} P_A(\mathbf{z}_A)^t Q_A(\mathbf{z}_A)^{1-t} P_{A^c|A}(\mathbf{z}_{A^c}|\mathbf{z}_A)^t Q_{A^c|A}(\mathbf{z}_{A^c}|\mathbf{z}_A)^{1-t}\right), \\
&= -\min_{t\in[0,1]} \log\left(\sum_{\mathbf{z}_A} P_A(\mathbf{z}_A)^t Q_A(\mathbf{z}_A)^{1-t} \sum_{\mathbf{z}_{A^c}} P_{A^c|A}(\mathbf{z}_{A^c}|\mathbf{z}_A)^t Q_{A^c|A}(\mathbf{z}_{A^c}|\mathbf{z}_A)^{1-t}\right), \\
&\geq -\min_{t\in[0,1]} \log\left(\sum_{\mathbf{z}_A} P_S(\mathbf{z}_A)^t Q_A(\mathbf{z}_A)^{1-t}\right) = D^*(P_A, Q_A),
\end{aligned} \tag{7.35}$$

where (7.35) results from Hölder's inequality: For non-negative vectors $\mathbf{v} = [v_k]$ and $\mathbf{w} = [w_k]$ that sum to 1, $\sum_k v_k^t w_k^{1-t} \leq (\sum_k v_k)^t (\sum_k w_k)^{1-t} = 1$ for every $t \in [0,1]$. The inequality in (7.34) is tight iff Hölder's inequality holds with equality. This occurs iff $\mathbf{v} = \mathbf{w}$ (since both vectors need to sum to unity). Thus, for equality to hold in (7.34), we need the conditionals $P_{A^c|A}$ and $Q_{A^c|A}$ to be identical, i.e., (7.4). This completes the proof. □

■ 7.B Proof of Proposition 7.2

Proof. Consider the following collection of events $\mathcal{E}_{S'_d} := \{\psi_n^*(\mathbf{x}^n, \mathbf{y}^n) = S'_d\}$ for all $S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}$. Alternatively,

$$\mathcal{E}_{S'_d} := \left\{ S'_d = \operatorname*{argmax}_{\tilde{S}_d \in \mathfrak{S}_{k,d}} D(\widehat{P}_{\tilde{S}_d}^{(d)} \,||\, \widehat{Q}_{\tilde{S}_d}^{(d)}) \right\}, \tag{7.36}$$

where the quantities in hats are the empirical distributions. That is $\mathcal{E}_{S'_d}$ is the event that the output of the exhaustive search decoder is the non-salient set S'_d.

We now bound the probability of each $\mathcal{E}_{S'_d}$ (wrt the probability measure $\mathbb{P}^n$). By Sanov's theorem applied to the product distribution $P^{(d)}_{S_d \cup S'_d} \times Q^{(d)}_{S_d \cup S'_d}$, we have the upper bound

$$\mathbb{P}^n(\mathcal{E}_{S'_d}) \leq (n+1)^{|\mathcal{X}|^{|S_d \cup S'_d|}} \exp(-n J_{S'_d|S_d}) \leq (n+1)^{|\mathcal{X}|^{2k}} \exp(-n J_{S'_d|S_d}), \tag{7.37}$$

where the error rate is given as the information projection:

$$J_{S'_d|S_d} = \min_{\nu \in \Gamma_{S'_d|S_d}} D(\nu \,||\, P^{(d)}_{S_d \cup S'_d} \times Q^{(d)}_{S_d \cup S'_d}). \tag{7.38}$$

Note that in the above, we have implicitly applied the contraction principle to the *continuous* function $f : \mathcal{P}(\mathcal{X}^{2|S_d \cup S'_d|}) \to \mathbb{R}$ given by the recipe

$$f\left((P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d})\right) := D(P^{(d)}_{S_d} \,||\, Q^{(d)}_{S_d}) - D(P^{(d)}_{S'_d} \,||\, Q^{(d)}_{S'_d}). \tag{7.39}$$

The constraint set $\Gamma_{S'_d|S_d}$ was defined in (7.13). Note also that the minimum in (7.38) is achieved because the objective function is continuous and the constraint set is compact. Also, the minimizer in (7.38) is achieved at the boundary of the constraint set

$$\Lambda_{S'_d|S_d} := \{\nu = (P, Q) \in \mathcal{P}(\mathcal{X}^{|S_d \cup S'_d|}) : D(P_{S_d} \,||\, Q_{S_d}) \leq D(P_{S'_d} \,||\, Q_{S'_d})\} \tag{7.40}$$

as can be readily checked, i.e., $\nu^* \in \mathrm{Bd}(\Lambda_{S'_d|S_d}) = \Gamma_{S'_d|S_d}$. This follows from the convexity of the KL-divergence objective in (7.38). Next, we complete the proof by applying the union bound and "largest-exponent-wins" principle.

$$\begin{aligned}
\mathbb{P}^n(\psi^*_n(\mathbf{x}^n, \mathbf{y}^n) \neq S_d) &= \mathbb{P}^n\left(\bigcup_{S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}} \mathcal{E}_{S'_d}\right) && (7.41)\\
&\leq \sum_{S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}} \mathbb{P}^n(\mathcal{E}_{S'_d}) && (7.42)\\
&\leq \sum_{S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}} (n+1)^{|\mathcal{X}|^{2k}} \exp(-n J_{S'_d|S_d}) && (7.43)\\
&\doteq \exp(-n C(P^{(d)}, Q^{(d)})). && (7.44)
\end{aligned}$$

Note that the constancy of k and d is crucial in (7.44). The conclusion in (7.44) means that

$$\limsup_{n \to \infty} \frac{1}{n} \log \mathbb{P}^n(\psi^*_n(\mathbf{x}^n, \mathbf{y}^n) \neq S_d) \leq -C(P^{(d)}, Q^{(d)}). \tag{7.45}$$

Together with the trivial lower bound

$$\liminf_{n \to \infty} \frac{1}{n} \log \mathbb{P}^n(\psi^*_n(\mathbf{x}^n, \mathbf{y}^n) \neq S_d) \geq -C(P^{(d)}, Q^{(d)}), \tag{7.46}$$

we conclude that the limit exists and equals the error exponent, i.e,

$$\lim_{n\to\infty} -\frac{1}{n}\log \mathbb{P}^n(\psi_n^*(\mathbf{x}^n,\mathbf{y}^n)\neq S_d) = C(P^{(d)},Q^{(d)}). \tag{7.47}$$

This completes the proof. □

■ 7.C Proof of Theorem 7.3

We first state four basic lemmas before proving Theorem 7.3.

Lemma 7.10. *For a continuously differentiable real-valued function $f : A \subset \mathbb{R}^n \to \mathbb{R}$, define the Lipschitz constant*

$$L := \sup_{\mathbf{x}\in A} \|\nabla f(\mathbf{x})\|_\infty = \sup_{\mathbf{x}\in A}\left(\max_{1\le i\le n}\left|\frac{\partial f}{\partial x_i}(x_i)\right|\right), \tag{7.48}$$

and assume $L<\infty$. Then, we have the Lipschitz *condition*

$$\forall\, \mathbf{x},\mathbf{y}\in A, \qquad |f(\mathbf{x})-f(\mathbf{y})| \le L\|\mathbf{x}-\mathbf{y}\|_1. \tag{7.49}$$

In fact this claim holds for any pair of conjugate exponents[2] $p,q\in[1,\infty]$, i.e., if the ∞ norm in (7.48) is replaced by p norm and the 1 norm in (7.49) is replaced by q norm.

Lemma 7.11. *The following bound for the binomial coefficient holds:*

$$\binom{d}{k} \le \exp\left(dH_b(\frac{k}{d})\right) \le \exp\left[k\left(\log(\frac{d}{k})+1\right)\right], \tag{7.50}$$

where H_b is the binary entropy function.

Lemma 7.12. *Let n be a positive integer and $\epsilon\in(0,1)$. Then the following relation holds*

$$\binom{n+n^{1-\epsilon}}{n} \in e^{o(n)}, \tag{7.51}$$

where the binomial coefficient defined in terms of Gamma functions, namely

$$\binom{n+n^{1-\epsilon}}{n} := \frac{\Gamma(n+n^{1-\epsilon}+1)}{\Gamma(n^{1-\epsilon}+1)\Gamma(n+1)}. \tag{7.52}$$

Lemma 7.13. *For two distributions Q_1,Q_2 with the same support Ω (a finite set), we have*

$$\frac{\partial D(Q_1\,||\,Q_2)}{\partial Q_1(a)} = 1+\log\frac{Q_1(a)}{Q_2(a)}, \qquad \frac{\partial D(Q_1\,||\,Q_2)}{\partial Q_2(a)} = -\frac{Q_1(a)}{Q_2(a)}, \qquad \forall\, a\in\Omega. \tag{7.53}$$

[2] p and q are called conjugate exponents if $1/p+1/q=1$.

We defer the proofs of the first three lemmas to after the proof of the theorem. The fourth follows by simple calculus and is thus omitted. We now prove the theorem assuming Lemmas 7.10 – 7.13.

Proof. (of Theorem 7.3)
Step 1: We first prove that the family of differentiable functions (indexed by d) $h_d : \mathcal{P}(\mathcal{X}^{2|S_d \cup S'_d|}) \to \mathbb{R}$ given by the recipe

$$h_d\left((P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d})\right) := D(P^{(d)}_{S_d} \,||\, Q^{(d)}_{S_d}) - D(P^{(d)}_{S'_d} \,||\, Q^{(d)}_{S'_d}), \tag{7.54}$$

is *equi-Lipschitz continuous in the l_1 norm*, i.e., there exists a $L' < \infty$ (independent of d), such that for all $d \in \mathbb{N}$ and for all two distinct product measures $\nu := (P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d})$ and $\tilde{\nu} := (\widetilde{P}^{(d)}_{S_d \cup S'_d}, \widetilde{Q}^{(d)}_{S_d \cup S'_d})$,

$$|h_d(\nu) - h_d(\tilde{\nu})| \leq L' \|\nu - \tilde{\nu}\|_1, \tag{7.55}$$

To prove this first claim, we first argue that ν and $\tilde{\nu}$ satisfy condition A3, i.e., the log-likelihood ratio between the distributions $P^{(d)}_{S_d \cup S'_d}$ and $Q^{(d)}_{S_d \cup S'_d}$ is uniformly bounded (by L). By using A1 and A3 (which says that the log-likelihood ratio of $P^{(d)}_{S_d}$ and $Q^{(d)}_{S_d}$ is uniformly bounded by L), we conclude that

$$\forall\, \mathbf{x}_{S_d \cup S'_d} \in \mathcal{X}^{|S_d \cup S'_d|}, \qquad \log \frac{P^{(d)}_{S_d \cup S'_d}(\mathbf{x}_{S_d \cup S'_d})}{Q^{(d)}_{S_d \cup S'_d}(\mathbf{x}_{S_d \cup S'_d})} \in [-L, L], \tag{7.56}$$

because the union of a non-salient set to the salient set S_d does not change the log-likelihood ratio (cf. the argument after Proposition 7.1). Thus, the L-boundedness condition also holds for $P^{(d)}_{S_d \cup S'_d}$ and $Q^{(d)}_{S_d \cup S'_d}$. Denote the set of such distributions (where the log-likelihood ratio is bounded by L) as $\mathcal{D}_L$. By evaluating the partial derivative of the KL-divergences in (7.54) with respect to each of its components and applying Lemma 7.13 repeatedly, we conclude that the l_∞ norm of the gradient vector of each function h_d in (7.54) is uniformly bounded, i.e., there exists a $L' < \infty$ such that

$$\sup_{(P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d}) \in \mathcal{D}_L} \left\| \nabla h_d \left((P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d}) \right) \right\|_\infty = L'. \tag{7.57}$$

In fact, we can verify directly from Lemma 7.13 that $L' = \max\{2e^L, 2L+2\} < \infty$. Now since the right-hand side of (7.57) is independent of d, we can take the supremum over all d on the left-hand side, i.e.,

$$\sup_{d \in \mathbb{N}} \left\{ \sup_{(P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d}) \in \mathcal{D}_L} \left\| \nabla h_d \left((P^{(d)}_{S_d \cup S'_d}, Q^{(d)}_{S_d \cup S'_d}) \right) \right\|_\infty \right\} = L'. \tag{7.58}$$

Finally apply Lemma 7.10 to every $d \in \mathbb{N}$ to conclude that the equi-Lipschitz continuity condition (7.55) for the family of functions $\{h_d\}_{d\in\mathbb{N}}$ in (7.54) holds with equi-Lipschitz constant L'.

Step 2: Now, most importantly, we prove that $B > 0$, where B is defined in (7.16). Assume, to the contrary, that $B = 0$ (since B cannot be negative). For a set of distributions Γ, let $D(\Gamma \,||\, \mu) := \min_{\nu\in\Gamma} D(\nu \,||\, \mu)$. By the definition of B and the infimum, there exists a $d \in \mathbb{N}$ (and a minimizing non-salient set S'_d) such that the divergence satisfies

$$D(\Gamma_{S'_d|S_d} \,||\, P^{(d)}_{S'_d\cup S_d} \times Q^{(d)}_{S'_d\cup S_d}) < \left(\frac{\eta}{2L'\sqrt{2\log 2}}\right)^2. \tag{7.59}$$

The quantity η was defined in (7.9) and represents how distinguishable the salient set S_d is from the non-salient sets $S'_d \in \mathfrak{S}_{k,d} \setminus \{S_d\}$. The quantity $L' < \infty$ is the equi-Lipschitz constant in (7.58). Let ν be the product distribution $P^{(d)}_{S'_d\cup S_d} \times Q^{(d)}_{S'_d\cup S_d}$ and ν^* be the minimizer of the optimization problem in the information projection (7.15) or equivalently (7.38), i.e.,

$$\nu^* := \operatorname{argmin} \left\{ D(\nu \,||\, P^{(d)}_{S'_d\cup S_d} \times Q^{(d)}_{S'_d\cup S_d}) : \nu \in \Gamma_{S'_d|S_d} \right\}. \tag{7.60}$$

Now referring back to (7.55) and applying Pinsker's inequality, we have the chain of inequalities

$$|h_d(\nu) - h_d(\nu^*)| \le L'\|\nu - \nu^*\|_1 \le L'\sqrt{2\log 2}\sqrt{D(\Gamma_{S'_d|S_d} \,||\, P^{(d)}_{S'_d\cup S_d} \times Q^{(d)}_{S'_d\cup S_d})} < \frac{\eta}{2}, \tag{7.61}$$

where the final inequality is because of (7.59). Notice how the finiteness and uniformity (independence from d) of L' are crucial in (7.59) and (7.61). Consequently, $h_d(\nu) \ge \eta$ (by assumption A2 on η-distinguishability) and $h_d(\nu^*) = 0$ (because $\nu^* \in \Gamma_{S'_d|S_d}$ by compactness of the constraint set $\Gamma_{S'_d|S_d}$). Thus,

$$|h_d(\nu) - h_d(\nu^*)| = h_d(\nu) - h_d(\nu^*) \ge \eta \tag{7.62}$$

and from (7.61), we conclude that $\eta < \eta/2$, which is clearly a contradiction. Hence $B > 0$.

Step 3: Now we simply put together the pieces in the proof by upper bounding the error probability p_n, defined in (7.8). Indeed, we that $\mathbb{P}^n(\psi_n(\mathbf{x}^n, \mathbf{y}^n) \neq S_d)$ is upper bounded as in the following sequence of inequalities:

$$\le \sum_{S'_d\in\mathfrak{S}_{k,d}\setminus\{S_d\}} \mathbb{P}^n(\mathcal{E}_{S'_d}), \tag{7.63}$$

$$\le \sum_{l=0}^{k-1} \binom{k}{l}\binom{d-k}{k-l} \max_{S'_d\in\mathfrak{S}_{k,d}\setminus\{S_d\}} \mathbb{P}^n(\mathcal{E}_{S'_d}), \tag{7.64}$$

$$\leq \sum_{l=0}^{k-1} \binom{k}{l}\binom{d-k}{k-l} \max_{S'_d \in \mathfrak{S}_{k,d}\setminus\{S_d\}} \binom{n+|\mathcal{X}|^{|S_d\cup S'_d|}-1}{n} \exp(-nJ_{S'_d|S_d}), \tag{7.65}$$

$$\leq \sum_{l=0}^{k-1} \binom{k}{l}\binom{d-k}{k-l} \binom{n+|\mathcal{X}|^{2k}-1}{n} \exp(-nB), \tag{7.66}$$

$$\leq \sum_{l=0}^{k-1} \exp(k)\exp\left[k\left(\log(\frac{d-k}{k})+1\right)\right] \binom{n+|\mathcal{X}|^{2k}}{n} \exp(-nB), \tag{7.67}$$

$$< k\exp\left[k\left(\log(\frac{d-k}{k})+2\right)\right] \binom{n+n^{1-\epsilon}}{n} \exp(-nB), \tag{7.68}$$

$$\leq \exp\left[k\log(\frac{d-k}{k})\right]\exp(2k+\log k)\binom{n+n^{1-\epsilon}}{n} \exp(-nB), \tag{7.69}$$

$$\leq \exp\left[k\log(\frac{d-k}{k})\right]\exp(o(n))\exp(-nB), \tag{7.70}$$

where

- (7.63) follows from the union bound and definition of the event $\mathcal{E}_{S'_d}$ given in the proof of Proposition 7.2 (cf. (7.36)).
- (7.64) follows by a simple counting argument that the number of non-salient sets S'_d that overlap with S_d in l indices is exactly $\binom{k}{l}\binom{d-k}{k-l}$. We also upper bound the probability $\mathbb{P}^n(\mathcal{E}_{S'_d})$ by the largest possible probability.
- (7.65) follows from Sanov's theorem and the fact that the number of types [49] with denominator n for a distributions with support $\mathcal{X}^{|S_d\cup S'_d|}$ is precisely $\binom{n+|\mathcal{X}|^{|S_d\cup S'_d|}-1}{n}$.
- (7.66) follows from the definition of $B>0$ in (7.16) (infimum over all error rates over all d) and the fact that $|S_d\cup S'_d|\leq 2k$ (because $|S_d|=|S'_d|=k$). Notice how the positivity of B, proved in Step 2, is crucial here.
- (7.67) follows from two applications of Lemma 7.11. In particular, we note that $\binom{k}{l}\leq \exp(kH_b(l/k))\leq \exp(k)$ (for every $l=0,1,\ldots,k-1$) and also $\binom{d-k}{k-l}$ is maximized when $l=0$. We also employ a trivial upper bound of the second binomial coefficient.
- (7.68) follows from the fact that there are only k terms in the sum and assumption that there exists a ϵ such that

$$k < \frac{(1-\epsilon)\log n}{2\log|\mathcal{X}|} \quad\Longleftrightarrow\quad \exp\left(\frac{2k\log|\mathcal{X}|}{1-\epsilon}\right) < n. \tag{7.71}$$

  This is given by the function g_1 in (7.17).
- (7.69) follows by simple rearrangement. Note that $\exp(2k+\log k)\in\exp(o(n))$ by (7.71).

- Lastly (7.70) follows from Lemma 7.12 and the absorption of all subexponential terms into $\exp(o(n))$.

Finally, from (7.70), we notice by a simple rearrangement that the exponent is given by $-n(B - o(1) - (k/n)\log((d-k)/k))$. In order to ensure that the error probability decays to zero, it is suffices to have

$$B - o(1) - \frac{k}{n}\log(\frac{d-k}{k}) > 0. \tag{7.72}$$

Condition (7.72) holds if for sufficiently large n

$$n > \frac{k}{B-\epsilon'}\log(\frac{d-k}{k}), \tag{7.73}$$

Take $\epsilon' \to 0$. We conclude from (7.71) and (7.73) that if $n > g_1(k,\epsilon) \vee g_2(d,k)$, then $\{(n,d,k)\}_{n\in\mathbb{N}}$ is achievable, where g_1 and g_2 were defined in (7.17). Now it is easy to see that the rate of decay $\limsup_{n\to\infty} n^{-1}\log p_n$ is simply given by $-c$ where c is the difference between B and the contribution from the binomial coefficient term $\binom{d-k}{k}$, i.e.,

$$c = B - \limsup_{n\to\infty}\frac{k}{n}\log(\frac{d-k}{k}), \tag{7.74}$$

which concludes the proof of Theorem 7.3. □

Now we prove the remaining lemmas.

Proof. (of Lemma 7.10)
Consider $n = 2$. The general case is easily deducible by extending the argument below inductively. Let $\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2) \in A \subset \mathbb{R}^2$ be any two points.

$$\begin{aligned}
|f(x_1,x_2) - f(y_1,y_2)| &= |f(x_1,x_2) - f(y_1,x_2) + f(y_1,x_2) - f(y_1,y_2)| && (7.75)\\
&\le |f(x_1,x_2) - f(y_1,x_2)| + |f(y_1,x_2) - f(y_1,y_2)| && (7.76)\\
&= \left|\frac{\partial f}{\partial x_1}(\xi_1)\right| |x_1 - y_1| + \left|\frac{\partial f}{\partial x_2}(\xi_2)\right| |x_2 - y_2| && (7.77)\\
&\le \sup_{\xi_1:(\xi_1,y_1)\in A}\left|\frac{\partial f}{\partial x_1}(\xi_1)\right| |x_1 - y_1| + \sup_{\xi_2:(x_2,\xi_2)\in A}\left|\frac{\partial f}{\partial x_2}(\xi_2)\right| |x_2 - y_2|\\
&\le L(|x_1 - y_1| + |x_2 - y_2|) = L\|\mathbf{x} - \mathbf{y}\|_1, && (7.78)
\end{aligned}$$

where in (7.77) we have made use of the 1-dimensional mean-value theorem [166, Ch. 5] and $\xi_j \in (x_j, y_j)$ for $j = 1, 2$ and in (7.78) we made use of the hypothesis in the lemma (cf. (7.48)). The claim thus follows. □

Proof. (of Lemma 7.11)
From [47, Ch. 11], we have the straightforward upper bound

$$\binom{d}{k} \le \exp\left(dH_b(\frac{k}{d})\right). \tag{7.79}$$

It remains to bound the binary entropy function $H_b(q)$ for $q \in [0,1]$. Note that for all $0 \le q \le 3$,

$$-(1-q)\log(1-q) \le -(1-q)(-q+\frac{q^2}{2}) = q - \frac{3}{2}q^2 + \frac{q^3}{2} \le q, \tag{7.80}$$

where we have used the fact that $\log(1-t) \ge -t + t^2/2$. Thus, we have

$$H_b(q) = -q\log q - (1-q)\log(1-q) \le -q\log q + q = q(-\log q + 1). \tag{7.81}$$

The proof is completed with the identification $q = k/d$ in (7.79). □

Proof. (of Lemma 7.12)
We make use of the following bound from [182, Corollary 2.3]:

$$\forall\, \alpha \in \mathbb{R}_+, n \in \mathbb{N}, \qquad \binom{\alpha n}{n} < \frac{1}{\sqrt{2\pi}} n^{-1/2} \frac{\alpha^{\alpha n + 1/2}}{(\alpha-1)^{(\alpha-1)n+1/2}}. \tag{7.82}$$

Note from close examination of the proof in [182] that this bound applies to the case where αn may not be an integer. In this case, the binomial coefficient is defined by the one involving Gamma functions (cf. (7.52)). Thus, taking $\alpha = 1 + n^{-\epsilon}$ in (7.82), we have

$$\binom{n+n^{1-\epsilon}}{n} = \binom{n(1+n^{-\epsilon})}{n} < \mathrm{poly}(n) \frac{(1+n^{-\epsilon})^{n(1+n^{-\epsilon})}}{(n^{-\epsilon})^{n^{1-\epsilon}}} =: \mathrm{poly}(n) M(n). \tag{7.83}$$

where $\mathrm{poly}(n) \in e^{o(n)}$ is some polynomial function in n. It suffices to prove that $M(n) \in e^{o(n)}$. Indeed,

$$\log M(n) = n(1+n^{-\epsilon})\log(1+n^{-\epsilon}) - n^{1-\epsilon}\log n^{-\epsilon} \tag{7.84}$$

$$\le n(1+n^{-\epsilon})n^{-\epsilon} + \epsilon n^{1-\epsilon}\log n \in o(n) \tag{7.85}$$

where (7.85) comes from the inequality $\log(1+t) \le t$. Thus $M(n) \in e^{o(n)}$ and this completes the proof. □

■ 7.D Proof of Corollary 7.4

Proof. Assume that $k = k_0$ is constant. The claim follows by replacing the upper bound for $\binom{d}{k_0}$ in (7.67) with the trivial upper bound d^{k_0}. If $k_0 R < B$, the corresponding exponent in (7.72) is positive. □

■ 7.E Proof of Theorem 7.5

Proof. Recall the Markov chain given in Section 7.3.4:

$$S_d \xrightarrow{\varphi_n} (\mathbf{x}^n, \mathbf{y}^n) \xrightarrow{\psi_n} \widehat{S}_d \tag{7.86}$$

Applying Fano's inequality, we have

$$\mathbb{P}^n(S_d \neq \widehat{S}_d) \geq \frac{H(S_d|\widehat{S}_d) - 1}{\log \binom{d}{k}} \tag{7.87}$$

$$= \frac{H(S_d) - I(S_d; \widehat{S}_d) - 1}{\log \binom{d}{k}} \tag{7.88}$$

$$= \frac{\log \binom{d}{k} - I(S_d; \widehat{S}_d) - 1}{\log \binom{d}{k}} \tag{7.89}$$

where (7.89) follows from the uniform distribution on S_d, which implies that $H(S_d) = \log |\mathfrak{S}_{k,d}|$. Now we upper bound the mutual information term:

$$I(S_d; \widehat{S}_d) \overset{(a)}{\leq} I(S_d; \mathbf{x}^n, \mathbf{y}^n) \overset{(b)}{\leq} H(\mathbf{x}^n, \mathbf{y}^n) \leq n(H(P^{(d)}) + H(Q^{(d)})), \tag{7.90}$$

where (a) follows from the data processing inequality and (b) follows from non-negativity of conditional entropy. Inserting (7.90) into (7.89), we have

$$\mathbb{P}^n(S_d \neq \widehat{S}_d) \geq 1 - \frac{n(H(P^{(d)}) + H(Q^{(d)}))}{\log \binom{d}{k}} - o(1) \tag{7.91}$$

$$\overset{(a)}{\geq} 1 - \frac{n(H(P^{(d)}) + H(Q^{(d)}))}{k \log \frac{d}{k}} - o(1), \tag{7.92}$$

where (a) follows from the fact that $\binom{d}{k} \geq (d/k)^k$. The claim in part (i) thus follows. Note the independence of the proof on the decoder ψ_n. □

■ 7.F Proof of Corollary 7.6

Proof. With the added assumption that the conditional entropies are bounded by a linear function in k, i.e., $\max\{H(P^{(d)}_{S^c_d|S_d}), H(Q^{(d)}_{S^c_d|S_d})\} \leq Mk$, the entropy decomposes as follows:

$$H(P^{(d)}) = H(P^{(d)}_{S_d}) + H(P^{(d)}_{S^c_d|S_d}) \overset{(a)}{\leq} \log |\mathcal{X}|^k + H(P^{(d)}_{S^c_d|S_d}) \tag{7.93}$$

$$\leq k \log |\mathcal{X}| + Mk = (\log |\mathcal{X}| + M)k, \tag{7.94}$$

where (a) is due to the fact that $P^{(d)}_{S_d} \in \mathcal{P}(\mathcal{X}^k)$ and hence $H(P^{(d)}_{S_d}) \leq \log |\mathcal{X}|^k$. Substituting this and the corresponding upper bound for $H(Q^{(d)})$ into (7.92) completes the proof of the claim in part (ii). □

■ 7.G Proof of Corollary 7.7

Proof. Take $\lambda = 1$ in (7.27). Then the claim follows by replacing d in (7.27) with Ce^{nR} (for some $C > 0$) and further noticing that the inequality is satisfied if and only if

$$R > 2(M + \log|\mathcal{X}|) + \frac{\log k}{n} = 2(M + \log|\mathcal{X}|) + o(1). \tag{7.95}$$

This completes the proof. □

■ 7.H Proof of Proposition 7.8

Proof. Recall that k and d are kept constant. We first demonstrate that the Chow-Liu algorithm for learning the common tree is consistent, as for a single tree [43, 188]. Let $\mathcal{T}^d$ be set of trees (or edge sets) with d nodes. Also recall that $\mathcal{D}(\mathcal{X}^d; \mathcal{T}^d) \subset \mathcal{P}(\mathcal{X}^d)$ is the set of distributions Markov on some tree in $\mathcal{T}^d$. The consistency claim follows from the equivalence of the following optimizations:

$$\min_{\widetilde{P},\widetilde{Q} \in \mathcal{D}(\mathcal{X}^d;\mathcal{T}^d): T_{\widetilde{P}} = T_{\widetilde{Q}}} \quad D((\widehat{P},\widehat{Q}) \,||\, (\widetilde{P},\widetilde{Q})), \tag{7.96}$$

$$\min_{\widetilde{P},\widetilde{Q} \in \mathcal{D}(\mathcal{X}^d;\mathcal{T}^d): T_{\widetilde{P}} = T_{\widetilde{Q}}} \quad D(\widehat{P} \,||\, \widetilde{P}) + D(\widehat{Q} \,||\, \widetilde{Q}), \tag{7.97}$$

$$\min_{\mathcal{E}_{\widetilde{P}}, E_{\widetilde{Q}} \in \mathcal{T}^d: E_{\widetilde{P}} = \mathcal{E}_{\widetilde{Q}}} \quad \sum_{(i,j) \in \mathcal{E}_{\widetilde{P}}} I(\widehat{P}_{i,j}) + \sum_{(i,j) \in \mathcal{E}_{\widetilde{P}}} I(\widehat{Q}_{i,j}), \tag{7.98}$$

$$\min_{E \in \mathcal{T}^d} \quad \sum_{(i,j) \in E} I(\widehat{P}_{i,j}) + \sum_{(i,j) \in E} I(\widehat{Q}_{i,j}), \tag{7.99}$$

$$\min_{E \in \mathcal{T}^d} \quad \sum_{(i,j) \in E} I(\widehat{P}_{i,j}) + I(\widehat{Q}_{i,j}), \tag{7.100}$$

where (7.98) follows from Chow-Liu and (7.99) follows from enforcing the equality constraint $E_{\widetilde{P}} = E_{\widetilde{Q}}$ into the objective. Thus, the edge weights are indeed given by the sum of the empirical mutual informations.

Furthermore, the KL-divergence is continuous in its arguments. To be more explicit, as $n \to \infty$, $\widehat{D}_i \to D_i$ and $\widehat{W}_{i,j} \to W_{i,j}$ in probability. Thus, the node and edge weights in (7.29) converge to their true values and the overall algorithm is consistent. The second claim follows from the fact that the complexity of Chow-Liu is $O(nd^2|\mathcal{X}|^2)$ and the complexity of the k-CARD TREE procedure is $O(dk^2)$ [22, 78]. □

Chapter 8

Conclusion

8.1 Summary of Main Contributions

THE overarching theme in this work is *complexity reduction*, of which two aspects were analyzed in the two parts of this thesis; learning thin graphical models and dimensionality reduction.

In the first part (Chapters 3 to 5), we analyzed structure learning of tree-structured probabilistic graphical models from an information-theoretic perspective. In particular, we took a novel approach that involves deriving a single figure-of-merit known as the *error exponent* for learning the tree structure via the Chow-Liu algorithm [42]. We concluded that in both the discrete and Gaussian settings, the error exponent can be approximated by a quantity likened to a signal-to-noise ratio. In the Gaussian case, we proved that under the very-noisy setting, stars are the most difficult for learning and chains are the easiest (if the paramterizations are kept fixed). Consistency, error rates and scaling laws for learning (more general) forest models were also studied in Chapter 5. In contrast to using heuristics such as the Akaike [5] and Bayesian [176] Information Criteria to learn forests as suggested in Edwards et al. [70], we showed that as the number of samples tends to infinity, the error probability in learning the forest structure decays (almost) exponentially fast.

The second part of the thesis analyzes dimensionality reduction for the purpose of discrimination. Specifically, in Chapter 6, we exploited the modeling capabilities of sparse graphical models to design lower-dimensional classifiers to approximate the likelihood ratio test. We showed that the learning of such models is computationally efficient and gives good classification accuracy on a variety of datasets. The suggested procedure also produces salient pairwise features for discrimination. Chapter 7 examines the issue of salient feature subset selection in greater detail and we derived scaling laws on the number of samples so that the salient set can be recovered asymptotically.

8.2 Recommendations for Future Research

The contributions in the previous chapters lead naturally to various avenues for further research. We mention some possible extensions in the following subsections.

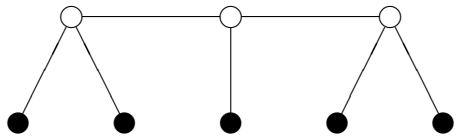

Figure 8.1. A latent tree: The shaded nodes represent the variables where there are measurements. The unshaded nodes do not provide any measurements. The structure of the latent tree is to be inferred from data.

■ 8.2.1 Optimality of Error Exponents

In Chapters 3 and 4, we analyzed the performance of the Chow-Liu ML algorithm in terms of the error exponent. However, a converse seems to be lacking. A natural question is whether this error exponent is the largest possible? In other words, does ML fitting result in the best possible rate function K_P in (3.4)? If not, are there any alternative algorithms that perform better? It is useful to derive a tight *converse* result and to study the optimality of K_P? While the strong converse in Theorem 5.7 suffices for the discussion in Chapter 5, it is conjectured to be too loose for any meaningful comparison to K_P in (3.4).

The learning problem can also be posed alternatively as a composite hypothesis test:

$$H_0 : \mathbf{x}^n \overset{\text{i.i.d.}}{\sim} P, \qquad H_1 : \mathbf{x}^n \overset{\text{i.i.d.}}{\sim} \{Q : Q \in \mathcal{D}(\mathcal{X}^d, \mathcal{T}^d \setminus \{T_P\})\}. \tag{8.1}$$

That is, under the null hypothesis, samples are drawn from P, Markov on T_P, and under the alternative, they are drawn from some other unknown distribution that is *not* Markov on T_P. The worst-case type-II error exponent for the test in (8.1) was derived in [190] but again it is unclear how this relates to the learning error exponent K_P in (3.4). Yet another way to approach optimality of error exponents is via the use of *Bahadur efficiency* [12] as was done in [27] for AR order testing. There are also a number of other converse techniques [219] that may be useful in deriving converse results to match the error exponent derived in previous chapters.

■ 8.2.2 Learning with Hidden Variables

All the learning problems in this thesis are analyzed based on a set of *fully-observed* i.i.d. samples $\mathbf{x}^n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ where each sample $\mathbf{x}_l$ is a length-d vector. Each component of this vector is in one-to-one correspondence with the nodes on a graph with d nodes. However, in most realistic and practical situations, we do not have the luxury of observing measurements from a subset of nodes known as the latent (or hidden) nodes. See Fig. 8.1. In this case, we are only provided with subvectors of length $d' < d$ and would like to reconstruct the full tree, without knowledge of the number and nature of latent nodes. This problem of reconstructing *latent trees* has received considerable attention in the phylogenetic [71, 142], computer vision [152] and network tomography [149, 150] communities. For example, in phylogenetics, (DNA or amino acid) sequences from extant species are available and one seeks to infer sequences of

extinct species as well as to reconstruct the evolutionary tree [68] taking into account substitution, insertions, deletions (indels) and possibly recombination.

By exploiting the Markov property in Lemma 2.26, we have shown [41] that the a class of latent trees known as *minimal latent trees* can be reconstructed consistently using pairwise information distances between observed nodes. These information distances generalize the notion of correlation coefficients in Gaussian models. However, while consistency is a desirable property in structure estimation, it does not provide a quantitative measure of how well one can learn the latent tree from n samples. An interesting line of research would be to extend the error exponent analysis in Chapters 3 and 4 to the latent tree case.

■ 8.2.3 Learning Loopy Random Graphical Models

The first part of this thesis focuses on learning graphical models in which the true structure is *deterministic*. It is also of theoretical interest to address the issue of learning *random* graphical models given samples drawn from the graphical model. Specifically, we assume that the underlying unknown graph $G = (V, E)$ is drawn from the ensemble of sparse Erdős-Rényi [24] random graphs $\mathcal{G}(d, \frac{c}{d})$ (where $c > 0$ is a constant).[1] Samples are then independently drawn from a graphical model Markov on the particular graph realization $G \sim \mathcal{G}(d, \frac{c}{d})$. Given the samples, what are some necessary and sufficient conditions on the sample size for asymptotic structure recovery as the number of nodes and the number of samples grow together? Are there any simple, computationally efficient algorithms for learning such classes of random graphical models from data?

This problem is motivated by the fact that many real-world networks can be modeled by random graphs [145], whose structures are usually unknown a-priori and need to be inferred. Unfortunately, exact structure estimation is, as mentioned in Section 2.5.1, NP-hard [112] unless the true graph belongs to the class of trees, in which case Chow-Liu provides an efficient implementation of maximum-likelihood estimation of the tree structure. There are also many efficient approximate algorithms (for example [32]) for learning degree-bounded graphs. However, random graphs, such as the Erdős-Rényi model do not have a bounded maximum degree.[2] The work in [211] allows for slowly growing maximum degree (with the number of nodes) but the incoherence conditions required for consistency guarantees are hard to establish for random graphs. Hence, the existing algorithms do not have provable guarantees for structural estimation of graphical models on random graphs.

In some preliminary work [9], we study homogeneous ferromagnetic Ising models, i.e, the inverse temperatures $\theta_{i,j} > 0$ are positive and constant across edges. The Ising

[1] An Erdős-Rényi [24] random graph $G \sim \mathcal{G}(d, q)$ is a graph with d nodes in which each edge is included in the graph with probability q, with the presence or absence of any two distinct edges in the graph being independent.

[2] In fact, the maximum degree grows like $O(\log d)$ for $\mathcal{G}(d, \frac{c}{d})$ [24].

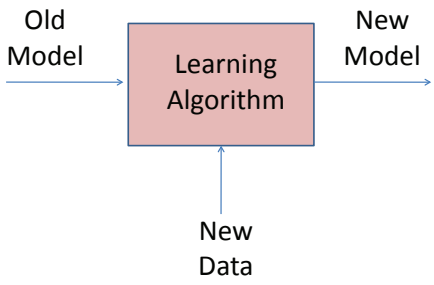

Figure 8.2. Illustration of online learning. The algorithm updates the model at each time step.

probability measure can thus be expressed as the exponential family:

$$P(\mathbf{x}\,;\,\theta) = \frac{1}{Z_\theta} \exp\left(\theta \sum_{(i,j)\in E} x_i x_j\right), \qquad \forall\, \mathbf{x} \in \{-1,+1\}^d, \tag{8.2}$$

where $Z_\theta := \sum_{\mathbf{x}} \exp(\theta \sum_{(i,j)\in E} x_i x_j)$ is the *partition function*. For this class of homogeneous ferromagnetic models, where E is a random set of edges, we showed that simple correlation thresholding results in consistently estimated models as the size of the graph d grows. It has been recently shown that such sparse random graphs are *locally tree-like* [140] and so we could possibly leverage on the Chow-Liu algorithm (or CLThres) to provide a good initial estimate of the random graph. This is currently work in progress.

■ 8.2.4 Online Learning of Graphical Models

In the preceding chapters, we provided quantitative theoretical guarantees for the estimation of tree-structured graphical models from data. In these *batch learning* problems, all the data $\mathbf{x}^n$ are present for learning. Suppose instead that the data, assumed to be drawn independently from some underlying unknown model P, are broken up into i.i.d. blocks $\mathbf{x}_1^n, \mathbf{x}_{n+1}^{2n}, \ldots$ and each of these blocks of data arrives to the learner *sequentially*, i.e., at discrete-time $t \in \mathbb{N}$ the learner receives data block $\mathbf{x}_{(t-1)n+1}^{tn}$. At time 1, the learner learns a model $P_{\mathrm{ML}}^{(1)}$ based on $\mathbf{x}_1^n$. At time 2, the learner learns an updated model $P_{\mathrm{ML}}^{(2)}$ using both the current model $P_{\mathrm{ML}}^{(1)}$ and the additional block of data $\mathbf{x}_{n+1}^{2n}$ and so on. This delves into the realm of *online learning* [160, 222]. See Fig. 8.2.

Some natural questions are in order: Firstly, how can one *update* model $P_{\mathrm{ML}}^{(t)}$ at time step $t+1$ to get a new model $P_{\mathrm{ML}}^{(t+1)}$ without having to recompute all the empirical statistics and re-run the MWST algorithm for learning trees? Secondly, how good are the models at each time step relative to one another and to the true model P? Can convergence rates and error exponents be computed given a reasonable scheme? We believe that satisfactory answers to these questions will prove to be useful in many real-time systems where batch data are not readily available.

■ 8.2.5 Estimating the Correct Number of Salient Features

In Chapter 7, we discussed the estimation of salient sets assuming that k, the cardinality of the true salient set is known. However, in many practical applications, the determination of the size of the salient set is just as important as the nature of the salient features. Thus, designing an estimation scheme to find $\widehat{k}$, an estimate of k, would be useful. In fact, some of the thresholding techniques developed in Chapter 5 used to determine the correct number of edges could be employed for this purpose.

In addition, the tree-based dynamic programming algorithm suggested in Chapter 7 has not been analyzed although it was shown to be computationally efficient and also consistent for the purpose of estimating S_d. It would be useful to study the properties and performance of this algorithm for recovering salient sets for tree models.

Bibliography

[1] E. Abbe and L. Zheng. Linear Universal Decoding for Compound Channels: an Euclidean Geometric Approach. In *International Symposium on Information Theory*, pages 1098–1102, 2008.

[2] P. Abbeel, D. Koller, and A. Y. Ng. Learning factor graphs in polynomial time and sample complexity. *Journal of Machine Learning Research*, Dec 2006.

[3] M. Aigner and G. M. Ziegler. *Proofs From THE BOOK*. Springer, 2009.

[4] E. Airoldi. Getting Started in Probabilistic Graphical Models. *PLoS Computational Biology*, 3, 2007.

[5] H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

[6] D. M. Allen. The relationship between variable selection and data agumentation and a method for prediction. *Technometrics*, 16(1):125–127, Feb 1974.

[7] S.-I. Amari. Information geometry on hierarchy of probability distributions. *IEEE Transactions on Information Theory*, 47(5):1701–1711, 2001.

[8] S.-I. Amari and H. Nagaoka. *Methods of Information Geometry*. American Mathematical Society, 2000.

[9] A. Anandkumar, V. Y. F. Tan, and A. S. Willsky. High-Dimensional Robust Structure Reconstruction of Ising Models on Random Graphs. In *NIPS Workshop on Robust Machine Learning*, 2010.

[10] F. Bach and M. I. Jordan. Beyond independent components: trees and clusters. *Journal of Machine Learning Research*, 4:1205–1233, 2003.

[11] R. R. Bahadur, S. L. Zabell, and J. C. Gupta. Large deviations, tests, and estimates. *Asymptotic Theory of Statistical Tests and Estimation*, pages 33–64, 1980.

[12] R.R. Bahadur, S.L. Zabell, and J.C. Gupta. Large deviations, tests, and estimates. *Asymptotic Theory of Statistical Tests and Estimation*, pages 33–64, 1980.

[13] O. Barndorff-Nielsen. *Information and Exponential Families in Statistical Theory*. John Wiley and Sons, Inc., New York, NY, 1978.

[14] M. Basseville. Distance measures for signal processing and pattern recognition. *Signal Processing*, 18:349–369, 1989.

[15] E. B. Baum and D. Haussler. What size net gives valid generalization? *Neural Computation*, 1(1):151–160, 1989.

[16] J. Bento and A. Montanari. Which graphical models are difficult to learn? In *Neural Information Processing Systems (NIPS)*. MIT Press, 2009.

[17] J. M. Bernando and A. F. M. Smith. *Bayesian Theory*. Wiley, 1st edition, 1994.

[18] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[19] D. P. Bertsekas and J. N. Tsitsiklis. *Introduction to Probability*. Athena Scientific, 1st, 2002.

[20] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1998.

[21] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2008.

[22] C. Blum. Revisiting dynamic programming for finding optimal subtrees in trees. *European Journal of Operations Research*, 177(1):102–115, 2007.

[23] A. Blumer, Ehrenfeucht A., Hassler D., and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery*, 36(4):929 – 965, Oct 1989.

[24] B. Bollobás. *Random Graphs*. Cambridge University Press, 2nd edition, 2001.

[25] S. Borade and L. Zheng. I-Projection and the Geometry of Error Exponents. In *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2006.

[26] S. Borade and L. Zheng. Euclidean Information Theory. In *IEEE International Zurich Seminar on Communications*, pages 14–17, 2008.

[27] S. Boucheron and E. Gassiat. Error Exponents for AR Order Testing. *IEEE Transactions on Infomation Theory*, 52(2):472–488, Feb 2006.

[28] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[29] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–849, 1998.

[30] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[31] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees: An Introduction*. Monterey, CA: Wadsworth, 1984.

[32] G. Bresler, E. Mossel, and A. Sly. Reconstruction of Markov random fields from samples: Some observations and algorithms. In *11th International workshop APPROX 2008 and 12th International workshop RANDOM*, pages 343–356., 2008.

[33] P. L. Brockett, A. Charnes, and W. W. Cooper. MDI Estimation via Unconstrained Convex Programming. *Communications and Statistics*, B-9:223–234, 1980.

[34] S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. SVM and Kernel Methods Matlab Toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France, 2005.

[35] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006.

[36] J.-R Chazottes and D. Gabrielli. Large deviations for empirical entropies of g-measures. *Nonlinearity*, 18:2545–2563, Nov 2005.

[37] A. Chechetka and C. Guestrin. Efficient Principled Learning of Thin Junction Trees. In *Advances of Neural Information Processing Systems (NIPS)*, 2007.

[38] H. Chernoff. Measure of asymptotic efficiency tests of a hypothesis based on the sum of observations. *Ann. Math. Stat.*, 23:493–507, May 1952.

[39] M. J. Choi, V. Chandrasekaran, D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Multiscale Stochastic Modeling for Tractable Inference and Data Assimilation. *Computer Methods in Applied Mechanics and Engineering*, 197:3492 – 3515, Aug 2008.

[40] M. J. Choi, V. Chandrasekaran, and A. S. Willsky. Gaussian Multiresolution Models: Exploiting Sparse Markov and Covariance Structure. *IEEE Transactions on Signal Processing*, 58:1012 – 1024, Mar 2010.

[41] M. J. Choi, V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Consistent and Efficient Reconstruction of Latent Tree Models. In *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2010.

[42] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Infomation Theory*, 14(3):462–467, May 1968.

[43] C. K. Chow and T. Wagner. Consistency of an estimate of tree-dependent probability distributions . *IEEE Transactions in Information Theory*, 19(3):369 – 371, May 1973.

[44] G. F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42:393–405, 1990.

[45] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. McGraw-Hill Science/Engineering/Math, 2nd edition, 2003.

[46] C. Cortes and V. Vapnik. Support Vector Machines. *Machine Learning*, 20(3): 273 – 297, 1995.

[47] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 2nd edition, 2006.

[48] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter. *Probabilistic networks and expert systems*. Statistics for Engineering and Information Science. Springer-Verlag, New York, 1999.

[49] I. Csiszár. The method of types. *IEEE Transactions on Information Theory*, 44 (6):2505–2523, Oct 1998.

[50] I. Csiszár and J. Korner. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Akademiai Kiado, 1997.

[51] I. Csiszár and F. Matúš. Information projections revisited. *IEEE Transactions on Information Theory*, 49(6):1474–1490, June 2003.

[52] I. Csiszár and P. Shields. The consistency of the BIC Markov order estimator. *Ann. Statist.*, 28(6):1601–1619, 2000.

[53] I. Csiszár and P. Shields. *Information Theory and Statistics: A Tutorial*. Now Publishers Inc, 2004.

[54] I. Csiszár and Z. Talata. Context tree estimation for not necessarily finite memory processes, via bic and mdl. *IEEE Transactions on Information Theory*, 52(3): 1007–16, 2006.

[55] A. Custovic, B. M. Simpson, C. S. Murray, L. Lowe, and A. Woodcock. The National Asthma Campaign Manchester Asthma and Allergy Study. *Pediatr Allergy Immunol*, 13:32–37, 2002.

[56] A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, Apr 2009.

[57] A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications*, 30 (1):56–66, Feb 2008.

[58] A. Dembo and A. Montanari. Ising Models on Locally Tree-Like Graphs. *Annals of Applied Probability*, 20(2):565–592, 2010.

[59] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications*. Springer, 2nd edition, 1998.

[60] A. Dempster. Covariance selection. *Biometrics*, 28:157–175, 1972.

[61] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39(1–38), 1977.

[62] F. Den Hollander. *Large Deviations (Fields Institute Monographs, 14)*. American Mathematical Society, Feb 2000.

[63] A. W. Van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 1998.

[64] J.-D. Deuschel and D. W. Stroock. *Large Deviations*. American Mathematical Society, Dec 2000.

[65] P. Domingos and Pazzani M. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning*, 29:103 – 137, 1997.

[66] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.

[67] M. Dudik, S. J. Phillips, and R. E. Schapire. Performance guarantees for regularized maximum entropy density estimation. In *Conference on Learning Theory (COLT)*, 2004.

[68] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge Univ. Press, 1999.

[69] D. Edwards. *Introduction to Graphical Modelling*. Springer, 2nd edition, 2000.

[70] D. Edwards, G. C. G. de Abreu, and R. Labouriau. Selecting high-dimensional mixed graphical models using minimal AIC or BIC forests. *BMC Bioinformatics*, 11(18), Jan 2010.

[71] P. L. Erdös, L. A. Székely, Steel M. A., and Warnow T. J. A few logs suffice to build (almost) all trees: Part ii. *Theoretical Computer Science*, 221:153–184, 1999.

[72] J. Fan and R. Li. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. In *Proceedings of the International Congress of Mathematicians*, 2006.

[73] R. M. Fano. *Transmission of Information*. New York: Wiley, 1961.

[74] M. Fazel, H. Hindi, and S. P. Boyd. Log-det heuristic for matrix rank minimization with applications with applications to Hankel and Euclidean distance metrics. In *American Control Conference*, 2003.

[75] M. Feder and N. Merhav. Universal composite hypothesis testing: a competitive minimax approach. *IEEE Transactions on Information Theory*, 48(8):1504 – 1517, 2002.

[76] A. A. Fedotov, P. Harremoes, and F. Topsoe. Refinements of Pinsker's inequality. *IEEE Transactions on Information Theory*, 49(6):1491 – 1498, Jun 2003.

[77] L. Finesso, C. C. Liu, and P. Narayan. The Optimal Error Exponent for Markov Order Estimation. *IEEE Transactions on Information Theory*, 42(5):1488–1497, 1996.

[78] M. Fischetti, W. Hamacher, K. Jornsten, and F. Maffioli. Weighted k-cardinality trees: Complexity and polyhedral structure. *Networks*, 24:11–21, 1994.

[79] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119– 139, 1997.

[80] Y. Freund and R. E. Schapire. A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780, Sep 1999.

[81] B. J. Frey. *Graphical Models for Machine Learning and Digital Communication*. MIT Press, Cambridge, MA, 1998.

[82] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics, Stanford University Technical Report, 1998.

[83] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799 – 805, Feb 2004.

[84] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[85] R. G. Gallager. Claude E. Shannon: A retrospective on his life, work and impact. *IEEE Transactions on Information Theory*, 47:2687–95, Nov 2001.

[86] E. Gassiat and S. Boucheron. Optimal Error Exponents in Hidden Markov Models Order Estimation. *IEEE Transactions on Infomation Theory*, 49(4):964–980, Apr 2003.

[87] D. Geiger and D. Heckerman. Learning Gaussian networks. In CA: Morgan Kaufmann San Francisco, editor, *Uncertainty in Artificial Intelligence (UAI)*, 1994.

[88] D. Grossman and P. Domingos. Learning bayesian network classifiers by maximizing conditional likelihood. In *Proc. of International Conference on Machine Learning*, 2004.

[89] A. Gupta, J. Lafferty, H. Liu, L. Wasserman, and M. Xu. Forest density estimation. In *Info. Th. and Applications (ITA) Workshop*, San Diego, CA, 2010.

[90] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[91] J. M. Hammersley and M. S. Clifford. Markov fields on finite graphs and lattices. *Unpublished*, 1970.

[92] T. S. Han. Hypothesis testing with the general source. *IEEE Transactions on Information Theory*, 46(7):2415 – 2427, 2000.

[93] T. S. Han. *Information-Spectrum Methods in Information Theory*. Springer Berlin Heidelberg, Feb 2010.

[94] T. S. Han and K. Kobayashi. The strong converse theorem for hypothesis testing. *IEEE Transactions on Information Theory*, 35(1):178 – 180, 1989.

[95] F. Harary. *Graph Theory*. Addison-Wesley, Massachusetts, 1972.

[96] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction,*. Springer Series in Statistics, 3rd edition, 2009.

[97] D. Heckerman and D. Geiger. Learning Bayesian Networks. Technical Report MSR-TR-95-02, Microsoft Research, Redmond, WA, December 1994.

[98] T. Heskes, K. Albers, and B. Kappen. Approximate Inference and Constrained Optimization. In *Uncertainty in Artificial Intelligence*, 2003.

[99] W. Hoeffding. Asymptotically Optimal Tests for Multinomial Distributions. *Ann. of Math. Stats.*, 36(2):369–401, 1965.

[100] W. Hoeffding and J. Wolfowitz. Distinguishability of sets of distributions. *Annals of Math. Statistics*, 29(3):700–718, 1958.

[101] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.

[102] P. J. Huber and V. Strassen. Minimax tests and the neyman-pearson lemma for capacities. *Annals of Statistics*, 1:251–263.

[103] K. Iriyama. Error exponents for hypothesis testing of the general source. *IEEE Transactions on Information Theory*, 51(4):1517 – 1522, 2005.

[104] E. T. Jaynes. *Model Comparison and Robustness in "Probability Theory: The Logic of Science"*, chapter 24. Cambridge University Press, 2003.

[105] Y. Jing, V. Pavlović, and J. M. Rehg. Boosted Bayesian network classifiers. *Machine Learning*, 73(2):155–184, 2008.

[106] J. Johnson, V. Chandrasekaran, and A. S. Willsky. Learning Markov Structure by Maximum Entropy Relaxation. In *Artificial Intelligence and Statistics (AISTATS)*, 2007.

[107] M. I. Jordan. *Learning in Graphical Models*. MIT Press, 1999.

[108] M. I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 19:140–155, 2004.

[109] S.-Y. Jung, Y. Park, K.-S. Choi, and Y. Kim. Markov random field based English part-of-speech tagging system. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 236–242, 1996.

[110] A. M. Kagan, Y. V. Linnik, and C. R. Rao. *Characterization Problems in Mathematical Statistics*. New York: Wiley, 1973.

[111] T. Kailath. The Divergence and Bhattacharyya Distance Measures in Signal Selection. *IEEE Transactions on Communication Technology*, 15(1):52–60, 1967.

[112] D. Karger and N. Srebro. Learning markov networks: maximum bounded tree-width graphs. In *Symposium on Discrete Algorithms*, pages 392–401, 2001.

[113] S. M. Kay. *Fundamentals Of Statistical Signal Processing*. Addison Wesley Longman, 2001.

[114] J. Kelly. A new interpretation of information rate. *Bell System Technical Journal*, 35, 1956.

[115] A. Kester and W. Kallenberg. Large deviations of estimators. *The Annals of Statistics*, pages 648–664, 1986.

[116] S. Khudanpur and P. Narayan. Order Estimation for a Special Class of Hidden Markov Sources and Binary Renewal Processes. *IEEE Transactions on Information Theory*, 48(6):1704 – 1713, 2002.

[117] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning)*. The MIT Press, 2009.

[118] D. Koller and M. Sahami. Toward optimal feature selection. Technical report, Stanford InfoLab, 1996.

[119] A. N. Kolmogorov. Logical basis for information theory and probability theory. *IEEE Transactions on Information Theory*, IT-14:662–664, 1965.

[120] J. B. Kruskal. On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. *Proceedings of the American Mathematical Society*, 7(1), Feb 1956.

[121] F. Kschischang and B. Frey. Iterative decoding of compound codes by probability propagation in graphical models. *IEEE Selected Areas of Communications*, 16(2): 219–230, Feb 1998.

[122] F. Kschischang, B. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519, Feb 2001.

[123] C. S. Kubrusly. *Measure Theory: A First Course*. Academic Press, 2006.

[124] S. Kullback. *Information Theory and Statistics*. John Wiley and Sons, New York, 1959.

[125] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[126] J. N. Laneman. On the Distribution of Mutual Information. In *Information Theory and Applications Workshop*, 2006.

[127] S. Lauritzen. *Graphical Models*. Oxford University Press, USA, 1996.

[128] S. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using L1-regularization. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.

[129] E. L. Lehmann. *Testing Statistical Hypotheses*. John Wiley & Sons, Inc., New York, NY, 1959.

[130] E. Levitan and N. Merhav. A competitive Neyman-Pearson approach to universal hypothesis testing with applications. *IEEE Transactions on Information Theory*, 48(8):2215 – 2229, 2002.

[131] S. Li. *Markov Random Field Modeling in Image Analysis*. Springer, New York, 2001.

[132] H. Liu, J. Lafferty, and L. Wasserman. Tree density estimation. *arXiv:1001.1557 [stat.ML]*, Jan 2010.

[133] D. J. C. Mackay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2002.

[134] H. B. Mann and A. Wald. On the statistical treatment of linear stochastic difference equations. *Econometrica*, 11:173–220, 1943.

[135] M. Meilă and M. I. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, Oct 2000.

[136] N. Meinshausen and P. Buehlmann. High dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

[137] N. Merhav. The Estimation of the Model Order in Exponential Families. *IEEE Transactions on Infomation Theory*, 35(5):1109–1115, 1989.

[138] N. Merhav, M. Gutman., and J. Ziv. On the estimation of the order of a Markov chain and universal data compression. *IEEE Transactions on Infomation Theory*, 35:1014–1019, 1989.

[139] M. Mézard and A. Montanari. *Information, Physics, and Computation*. Oxford University Press, 2009.

[140] A. Montanari, E. Mossel, and A. Sly. The weak limit of ising models on locally tree-like graphs. *Arxiv:0912.0719 [math.PR]*, 2009.

[141] G. Morvai and B. Weiss. Order Estimation or Markov Chains. *IEEE Transactions on Information Theory*, 51(4):1496–97, Apr 2005.

[142] E. Mossel. Phase transitions in phylogeny. *Trans. Amer. Math. Soc.*, 356:2379–2404, 2004.

[143] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, Apr 2003.

[144] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz. UCI Repository of Machine Learning Databases, University of California, Irvine, 1998.

[145] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random Graph Models of Social Networks. *Proceedings of the National Academy of Sciences USA*, 99: 2566–2572, 2002.

[146] J. Neyman and E. Pearson. On the Problem of the Most Efficient Tests of Statistical Hypotheses. *Philosophical Transactions of the Royal Society of London. Series A*, 231:289–337, 1933.

[147] A. Y. Ng. On feature selection: learning with exponentially many irrelevant features as training examples. In *Proc. 15th ICML*, pages 404–412. Morgan Kaufmann, 1998.

[148] A. Y. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and Naïve Bayes. In *Proceedings of Neural Information Processing Systems*, 2002.

[149] J. Ni and S. Tatikonda. Network Tomography Based on Additive Metrics. *Proceedings of CISS and on arXiv:0809.0158*, Aug 2008.

[150] J. Ni, H. Xie, S. Tatikonda, and R. Yang. Efficient and Dynamic Routing Topology Inference from End-to-End Measurements. *IEEE/ACM Transactions on Networking*, 18:123–135, Feb 2010.

[151] C. Pandit and S. P. Meyn. Worst-case large-deviations with application to queueing and information theory. *Stochastic Processes and Applications*, 116(5):724–756, May 2006.

[152] D. Parikh and T. H. Chen. Hierarchical Semantics of Objects (hSOs). In *International Conference on Computer Vision*, pages 1–8, 2007.

[153] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2nd edition, 1988.

[154] K. Pearson. On lines and planes of closest fit to systems of points in space. *London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series*, 2:559–572, 1901.

[155] H.C. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[156] M. S. Pinsker. *Information and Information Stability of Random Variables*. Oakland, CA: Holden-Day, 1964.

[157] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer, 2nd, 1998.

[158] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36, 1957.

[159] J. R. Quinlan. Bagging, boosting and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, 1996.

[160] S. Rakhlin. Online learning. MIT 9.520: Statistical Learning Theory Lecture notes, 2008.

[161] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, Nov 2004.

[162] F. Rosenblatt. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6):386–408, 1958.

[163] S. Rosset and E. Segal. Boosting Density Estimation. In *Proceedings of Neural Information Processing Systems*, pages 641–648, 2002.

[164] A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electron. J. Statist.*, 2:494–515, 2008.

[165] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec 2000.

[166] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, 1976.

[167] H. Rue and L. Held. *Gaussian Markov Random Fields: Theory and Applications (Monographs on Statistics and Applied Probability)*. Chapman and Hall/CRC, 2005.

[168] K. Ryu. Econometric Analysis of Mixed Parameter Models. *Journal of Economic Theory and Econometrics*, 5(113–124), 1999.

[169] S. Sanghavi, D. Malioutov, and A. S. Willsky. Belief Propagation and LP Relxation for Weighted Matching in General graphs. In *Proceedings of Neural Information Processing Systems*, 2007.

[170] S. Sanghavi, V. Y. F. Tan, and A. S. Willsky. Learning graphical models for hypothesis testing. *Proceedings of 14th IEEE Statistical Signal Processing Workshop*, pages 69–73, Aug 2007.

[171] I. Sanov. On the probability of large deviations of random variables. *Sel. Transl. Math. Statist. Probab.*, pages 213 – 244, 1961.

[172] N. Santhanam and M. J. Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. In *Proc. of IEEE Intl. Symp. on Info. Theory*, Toronto, Canada, July 2008.

[173] R. E. Schapire. A Brief Introduction to Boosting. In *Proceedings of Intl. Joint Conf. on Artificial Intelligence (IJCAI)*, 1999.

[174] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 80–91, 1998.

[175] R. E. Schapire and Y. Singer. Improved boosting using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[176] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[177] R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. Wiley-Interscience, Nov 1980.

[178] C. E. Shannon. A mathematical theory of communication. Tech, Bell System Technical Journal, Oct 1948.

[179] S. Shen. Large deviation for the empirical correlation coefficient of two Gaussian random variables. *Acta Mathematica Scientia*, 27(4):821–828, Oct 2007.

[180] G. Simon. Additivity of information in exponential family probability laws. *Amer. Statist. Assoc.*, 68(478–482), 1973.

[181] A. Simpson, V. Y. F. Tan, J. M. Winn, M. Svensén, C. M. Bishop, D. E. Heckerman, I. Buchan, and A. Custovic. Beyond Atopy: Multiple Patterns of Sensitization in Relation to Asthma in a Birth Cohort Study. *Am J Respir Crit Care Med*, 2010.

[182] P. Stănică. Good Upper and Lower Bounds on Binomial Coefficients. *Journal of Inequalities in Pure and Applied Mathematics*, 2(3), 2003.

[183] J. Su and H. Zhang. Full Bayesian Network Classifiers. In *Proceedings of International Conference on Machine Learning*, pages 897–904, 2006.

[184] R. Szeliski. Bayesian modeling of uncertainty in low-level vision. *Journal of Computer Vision*, 5(3):271–301, Dec 1990.

[185] V. Y. F. Tan. Bounding the KL-divergence by its flipped version. Available at http://web.mit.edu/vtan/www/KLDivBound.pdf, Apr 2010.

[186] V. Y. F. Tan, J. W. Fisher, and A. S. Willsky. Learning max-weight discriminative forests. In *Proceedings IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1877–1880, Mar 2008.

[187] V. Y. F. Tan, A. Anandkumar, L. Tong, and A. S. Willsky. A Large-Deviation Analysis for the Maximum Likelihood Learning of Markov Tree Structures. *submitted to IEEE Transactions on Information Theory, Arxiv 0905.0940*, May 2009.

[188] V. Y. F. Tan, A. Anandkumar, L. Tong, and A. S. Willsky. A Large-Deviation Analysis for the Maximum Likelihood Learning of Tree Structures. In *Proceedings of IEEE International Symposium on Information Theory*, pages 1140 – 1144, Seoul, Jul 2009.

[189] V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. How do the structure and parameters of tree Gaussian graphical models affect structure learning? In *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2009.

[190] V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Error Exponents for Composite Hypothesis Testing of Markov Forest Distributions. In *Proceedings of the International Symposium on Information Theory*, June 2010.

[191] V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Scaling Laws for Learning High-Dimensional Markov Forests. In *Proceedings of Allerton Conference on Communication, Control, and Computing*, 2010.

[192] V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Learning High-Dimensional Markov Forest Distributions: Analysis of Error Rates. *submitted to J. Mach. Learn. Research, on Arxiv*, May 2010.

[193] V. Y. F. Tan, A. Anandkumar, and A. S. Willsky. Learning Gaussian Tree Models: Analysis of Error Exponents and Extremal Structures. *IEEE Transactions on Signal Processing*, 58(5):2701–2714, May 2010.

[194] V. Y. F. Tan, M. J. Johnson, and A. S. Willsky. Necessary and Sufficient Conditions for High-Dimensional Salient Subset Recovery. In *Proceedings of IEEE International Symposium on Information Theory*, Austin, TX, Jun 2010.

[195] V. Y. F. Tan, S. Sanghavi, J. W. Fisher, and A. S. Willsky. Learning Graphical Models for Hypothesis Testing and Classification. *IEEE Transactions on Signal Processing*, Nov 2010.

[196] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, Dec 2000.

[197] R. Tibshirani. Regression shrinkage and selection via the Lasso. *J. Royal. Statist. Soc B.*, 58(1):267–288, 1996.

[198] J. Unnikrishnan, D. Huang, S. Meyn, A. Surana, and V. V. Veeravalli. Universal and composite hypothesis testing via mismatched divergence. *IEEE Transactions on Information Theory*. revised May 2010, on arXiv http://arxiv.org/abs/0909.2234.

[199] H. Van Trees. *Detection, Estimation and Modulation Theory: Part I*. New York, Wiley, 1968.

[200] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38: 49–95, Mar 1996.

[201] V. N. Vapnik. *The Nature of Statistical Learning Theory*. Berlin: Springer-Verlag, 1995.

[202] S. R. S. Varadhan. *Large Deviations and Applications*. Society for Industrial Mathematics, 1984.

[203] K. R. Varshney. *Frugal Hypothesis Testing and Classification*. PhD thesis, Massachusetts Institute of Technology, 2010.

[204] K. R. Varshney and A. S. Willsky. Learning Dimensionality-Reduced Classifiers for Information Fusion. In *The Twelfth International Conference on Information Fusion*, pages 1881–1888, Seattle, Washington, Jul 2009.

[205] K. R. Varshney and A. S. Willsky. Classification Using Geometric Level Sets. *Journal of Machine Learning Research*, 11:491–516, 2010.

[206] S. Verdu. Spectral efficiency in the wideband regime. *IEEE Transactions on Information Theory*, 48(6), Jun 2002.

[207] M. J. Wainwright. Estimating the "Wrong" Graphical Model: Benefits in the Computation-Limited Setting. *Journal of Machine Learning Research*, 7, 2006.

[208] M. J. Wainwright. Information-Theoretic Limits on Sparsity Recovery in the High-Dimensional and Noisy Setting. *IEEE Transactions on Infomation Theory*, pages 5728–5741, Dec 2009.

[209] M. J. Wainwright and M. I. Jordan. *Graphical Models, Exponential Families, and Variational Inference*, volume 1 of *Foundations and Trends in Machine Learning*. Now Publishers Inc, 2008.

[210] M. J. Wainwright, T. Jaakkola, and A. S. Willsky. Tree-based reparamterization analysis of sum-product and its generalizations. *IEEE Transactions on Information Theory*, 49(5):1120–1146, 2003.

[211] M. J. Wainwright, P. Ravikumar, and J. D. Lafferty. High-Dimensional Graphical Model Selection Using ℓ_1-Regularized Logistic Regression. In *Advances of Neural Information Processing Systems (NIPS)*, pages 1465–1472, 2006.

[212] C. C. Wang and A. K. C Wong. Classification of discrete data with feature space transformation. *IEEE Transactions on Automatic Control*, AC-24(3):434–437, Jun 1979.

[213] D. B. West. *Introduction to Graph Theory*. Prentice Hall, 2nd edition, 2000.

[214] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons, New York, NY, 1990.

[215] A. S. Willsky. Multiresolution Markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, Aug 2002.

[216] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8(7):13411390, 1996.

[217] J. Woods. Markov image modeling. *IEEE Transactions on Automatic Control*, 23:846–850, 1978.

[218] R. Yeung. *A First Course on Information Theory*. Springer, 2002.

[219] B. Yu. Assouad, Fano, and Le Cam. *In Festschrift for Lucien Le Cam: Research Papers in Probability and Statistics (D. Pollard, E. Torgersen and G. L. Yang, eds.), Springer, New York*, pages 423–435, 1997.

[220] O. Zeitouni and M. Gutman. On Universal Hypotheses Testing via Large Deviations. *IEEE Transactions on Information Theory*, 37(2):285–290, 1991.

[221] O. Zeitouni, J. Ziv, and N. Merhav. When is the Generalized Likelihood Ratio Test Optimal? *IEEE Transactions on Information Theory*, 38(5):1597–1602, Sep 1992.

[222] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*, pages 928–936, 2003.

[223] O. Zuk, S. Margel, and E. Domany. On the number of samples needed to learn the correct structure of a Bayesian network. In *Proc of Uncertainty in Artificial Intelligence (UAI)*, 2006.