The Dissertation Committee for Abhik Kumar Das
certifies that this is the approved version of the following dissertation:

An Information Theoretic Approach to Structured High-Dimensional Problems

Committee:

Sriram Vishwanath, Supervisor

Sujay Sanghavi

Alex Dimakis

Felipe Voloch

Syed Ali Jafar

An Information Theoretic Approach to Structured High-Dimensional Problems

by

Abhik Kumar Das, B.Tech., M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2013

To Ma, Baba, and my sister Ankita.

Acknowledgments

I consider myself very fortunate to have a wonderful supervisor like Sriram Vishwanath, and I am thankful to him for his guidance, support and encouragement during my time as a graduate student at UT Austin. Sriram has been an excellent mentor and role-model; he has helped me out in numerous aspects, like help with formulating the research problems and having one-on-one discussions about ways to tackle them, his constructive criticism, his tips for improving my presentation skills and technical writing style. I am thankful to Sujay Sanghavi, with whom I have had a fruitful collaboration on some of my research problems, and am indebted to him for his valuable career-related advice. I am grateful to Syed Jafar, Alex Dimakis and Felipe Voloch for agreeing to be a part of my dissertation committee, and for their comments and suggestions related to improving different aspects of this dissertation. I am also glad to have interacted with Sanjay Shakkottai, who has been my favorite teacher during my time at UT Austin.

My graduate student life would be incomplete without the excellent interactions that I have had with fellow graduate students at UT Austin. Among the senior students in my research group, Rajiv Soundararajan, Jubin Jose, Shreeshankar Bodas, and Shweta Agrawal, were immensely helpful during my initial years as a graduate student with their valuable advice on research mentality as well as the personal front. Kumar Appaiah, Praneeth Netrapalli, Sharayu Moharir

and Siddhartha Banerjee have been great friends who have helped me out throughout my time in graduate school. I treasure the conversations I've had with fellow LINC and WNCG group members, Deepjyoti Deka, Youngchun Kim, Avhishek Chatterjee, Ankit Rawat, Ioannis Mitliagkas, Hongbo Si, Aneesh Reddy, Anish Mittal, Srinadh Bhojanapalli, Sarabjot Singh, Harpreet Dhillon, Ethan Elenberg, Subhashini Krishnasamy, Abhishek Gupta, Joyce Ho, Avik Ray and Yongseok Yoo. I also thank Janet Preuss and Karen Little for having meticulously taken care of my appointments and reimbursements through my time as a PhD student.

I have had an awesome social life at Austin which kept me in great spirits during the course of my studies at UT Austin. I thank my friends, Shatam Agrawal, Pradeep Dhananjay, Aswin Balasubramanian, Guneet Kaur, Harsh Shah, Kiran Divakar, Aditya Aravind, Tanvi Joshi and Kriti Kapoor for having put up with me – I won't never forget the great times we spent pulling all-nighters, watching movies and going on road trips. I am also thankful to my friends from my undergraduate days, Mudit Jain, Vivek Tiwari, Siddhartha Patowary, Sidhant Misra, Man Prakash Gupta and Pratap, who have been a great source of warmth and happiness, and have given me encouragement and their best wishes through all these years.

My parents and my sister Ankita have always stood by me and given their unflinching support to every aspect and endeavor of my life. I have no words to express my gratitude and love for them, it is a fact that this dissertation would not have seen the light of day without their blessings and best wishes. I am especially grateful to my brother-in-law Samarjit for giving me motivation. I am also thankful to my grandparents and maternal uncle for being my well-wishers.

An Information Theoretic Approach to Structured High-Dimensional Problems

Abhik Kumar Das, Ph.D.
The University of Texas at Austin, 2013

Supervisor: Sriram Vishwanath

A majority of the data transmitted and processed today has an inherent structured high-dimensional nature, either because of the process of encoding using high-dimensional codebooks for providing a systematic structure, or dependency of the data on a large number of agents or variables. As a result, many problem setups associated with transmission and processing of data have a structured high-dimensional aspect to them. This dissertation takes a look at two such problems, namely, communication over networks using network coding, and learning the structure of graphical representations like Markov networks using observed data, from an information-theoretic perspective. Such an approach yields intuition about good coding architectures as well as the limitations imposed by the high-dimensional framework. The dissertation studies the problem of network coding for networks having multiple transmission sessions, i.e., multiple users communicating with each other at the same time. The connection between such networks and the information-theoretic interference channel is examined, and the concept of interference alignment, derived from interference channel literature, is coupled with linear network coding to develop novel coding schemes offering good

guarantees on achievable throughput. In particular, two setups are analyzed – the first where each user requires data from only one user (multiple unicasts), and the second where each user requires data from potentially multiple users (multiple multicasts). It is demonstrated that one can achieve a rate equalling a significant fraction of the maximal rate for each transmission session, provided certain constraints on the network topology are satisfied. The dissertation also analyzes the problem of learning the structure of Markov networks from observed samples – the learning problem is interpreted as a channel coding problem and its achievability and converse aspects are examined. A rate-distortion theoretic approach is taken for the converse aspect, and information-theoretic lower bounds on the number of samples, required for any algorithm to learn the Markov graph up to a pre-specified edit distance, are derived for ensembles of discrete and Gaussian Markov networks based on degree-bounded graphs. The problem of accurately learning the structure of discrete Markov networks, based on power-law graphs generated from the configuration model, is also studied. The effect of power-law exponent value on the hardness of the learning problem is deduced from the converse aspect – it is shown that discrete Markov networks on power-law graphs with smaller exponent values require more number of samples to ensure accurate recovery of their underlying graphs for any learning algorithm. For the achievability aspect, an efficient learning algorithm is designed for accurately reconstructing the structure of Ising model based on power-law graphs from the configuration model; it is demonstrated that optimal number of samples suffices for recovering the exact graph under certain constraints on the Ising model potential values.

Table of Contents

List of Figures

Chapter 1

Introduction

The recent decade has seen an explosion in the amount of data that needs to be communicated and processed. With the advent of the internet and mobile devices, understanding and designing ways of communicating and analyzing data in an efficient manner has gained significant interest. At present, most of the data typically possess a structured high-dimensional nature. For example, data is generally preprocessed and provided a regular structure (for protection against errors and corruption) through the use of high-dimensional codebooks prior to transmission across channels or networks. Likewise, data may be modeled as being generated from the interactions among several variables or agents, thereby imparting a high-dimensional build. As such, it is important to consider this inherent high-dimensional nature of data for the purpose of processing and analysis.

The problem of communication over networks is a relevant one, as networks like cellular networks, WiFi and the internet have become a part and parcel of our daily lives. The number of users involved in these networks tend to be large; therefore, it is important to design coding schemes that allow users to simultaneously share the network resources for data transmission. One of the major challenges of this problem lies in ensuring that most users transmit data at

rates as close as possible to the capacity values supported by the network. It is known that for point-to-point channels and some instances of simple networks one can bring the achievable transmission rates for coding schemes closer to capacity values if the codeword lengths are sufficiently long, i.e., the dimensionality of coding schemes is large enough [1–3]. Also, the problem of designing coding schemes can be shown to reduce to the problem of finding efficient sphere packings, also referred to as the sphere packing problem [1]. Therefore, designing high-dimensional codebooks is analogous to finding a solution to the sphere packing problem in the high-dimensional regime. This observation motivates the treatment of the task of designing codebooks for communication over networks as a structured high-dimensional problem. The goal for the case of multiple users communicating over networks is to design multiple high-dimensional sphere packings (one packing per user) that are practical, efficient and resolvable from each other, so that the destinations for the users can recover the data meant for them.

As mentioned before, the observation of data may be modeled as the result of interdependencies among a collection of variables or agents, with the interdependencies either being absolute or probabilistic in nature. A succinct way of describing the data or the process generating it is through the use of graph-based representations, where the variables or agents become the nodes and the interdependencies become the edges. For example, if the observed data is assumed to be generated from probability distributions, Markov networks are the undirected graph structures that encode the distribution as well as the conditional independence relations among the variables [4] (see Figure 1.1). Likewise, human mu-

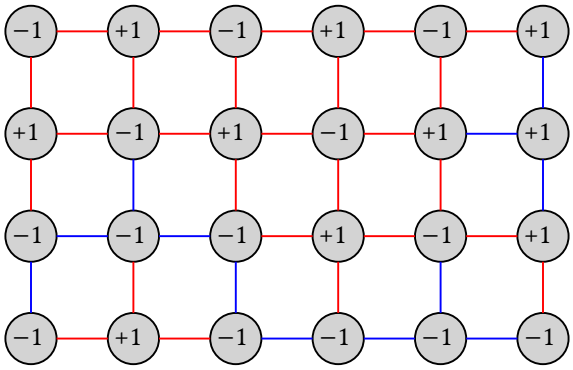


Figure 1.1: Example of Markov network: lattice Ising model

tations and diseases can be depicted by networks that show interactions among genes or protein complexes. An important problem associated with this modeling scheme is learning the underlying graph structure from observed data. In the context of Markov networks, this is also referred to as the problem of learning the structure of Markov networks or graphical model selection – it crops up in a wide variety of fields, ranging from computer vision and image processing to biology and statistical physics. Given the number of variables is large, the observed data has a high-dimensional aspect to it; therefore, graphical model selection can be interpreted as a structured high-dimensional problem. A major challenge of this problem lies in deriving the necessary and sufficient conditions on the nature and volume of observed data required for reconstructing the graph topology.

Thus, the problems of communication over networks and learning graph structure from data generated using graph-based representations, like Markov networks, are different versions of the structured high-dimensional framework. This dissertation attempts to address some of the challenges and issues in the context

of these problems using tools and techniques from information theory.

1.1 Motivation

The significance of communication over networks makes it a well-studied and established area of research. The noisy channel coding theorem, stated by Shannon, characterizes the capacity of point-to-point channels with noise having arbitrary probability distributions [1]. However, our understanding of coding schemes that can achieve maximal throughput in networks with multiple users is still limited. On the theoretical side, coding schemes that achieve network capacity are known only for networks with one unicast session (one source and one destination) [5, 6], one source multicast session (one source and multiple destinations; each destination requires data from the source), and one destination incast session (multiple sources and one destination; the destination requires data from all sources) [6, 7]. It is known that routing is sufficient to achieve the capacity in networks with single unicast session. For networks with one source multicast or one destination incast session, a more sophisticated coding scheme called network coding, that involves joint encoding of incoming data packets at every intermediate node, is required for achieving capacity. Nevertheless, the information-theoretic study of these special cases give us intuition on the architectural properties of optimal coding schemes for general instances of communication networks.

This provides the motivation for information-theoretic study of the problem of communication over general networks, which is one of the focuses of this dissertation. As mentioned before, optimal coding strategies based on network

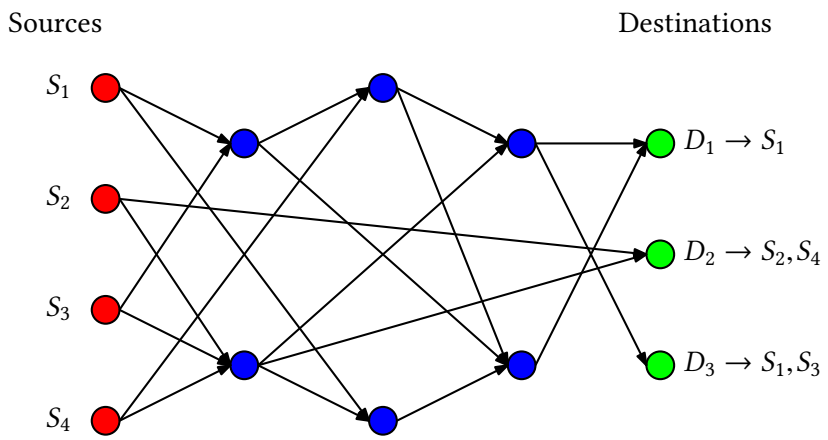


Figure 1.2: Communication network with multiple sessions.

coding have been designed for networks with either one source or one destination. We take a look at network setups with multiple sessions, i.e., multiple sources and destinations communicating with each other, and employing network coding as the coding scheme. An example of network with multiple sessions is illustrated in Figure 1.2, where destinations D_1, D_2, D_3 needs messages from sources $S_1, \{S_2, S_4\}, \{S_1, S_3\}$ respectively. In particular, we examine the performance and limitations of network coding as well as characterize the achievable rates for networks with multiple unicasts (multiple sources and destinations with each source communicating with a unique destination) or multicasts (multiple sources and destinations with each source communicating with multiple destinations). An important observation in this context is that the application of network coding in networks with multiple sessions makes it analogous, in structure, to an interference channel (or its generalized version). As such, coding strategies for interference channels, such as interference alignment, could potentially be coupled with net-

work coding to provide throughput guarantees [8–11]. The interplay of multiple sessions in a general network setup can be very complicated with arbitrary correlations among sessions, arising due to factors like presence of bottleneck links and sharing of network paths. In this dissertation, we investigate the influence of network structure and interaction of multiple sessions on the achievable rates, as well as the feasibility and limitations of (linear) network coding approach.

Markov networks provide a powerful framework for representing probability distributions in multi-dimensional space succinctly. The problem of learning Markov networks is an important task, and it involves estimating the structure of the underlying undirected graph as well as the probability distribution parameters. There are two aspects to the problem of learning Markov networks. One aspect is concerned with learning algorithms that can accurately estimate the structure and parameters of a Markov network using the observed samples generated by its probability distribution. We refer to this as the achievability aspect, since it has the same spirit as the achievability aspect of channel coding theorems. The other aspect is concerned with obtaining information-theoretic limits of the learning problem, i.e., necessary conditions on the nature and number of observed samples that characterizes the Markov network. We refer to this as the converse aspect, since it is analogous to the converse aspect of channel coding theorems. Understanding both these aspects of the learning problem is useful in general – the converse aspect provides a description of settings where recovery of the Markov network structure is impossible, regardless of the learning algorithm or cleverness of its design, while the achievability aspect focuses on designing practical learning al-

gorithms as well as algorithmic issues such as computational complexity.

This provides the motivation for studying these aspects for the problem of learning Markov graphs from an information-theoretic perspective, which is the other focus of this dissertation. There has been a decent amount of work in the context of the converse aspect of the learning problem, where bounds on sample complexity have been derived for exact recovery of specific families of Markov graphs. We take a look at this from a rate-distortion theoretic perspective – in place of exact recovery, we permit some amount of distortion in the estimate of the Markov graph structure, and examine the potential reduction in the bounds (this paradigm is analogous to rate-distortion theory in information theory [**?**]). We also place emphasis on the formulation of strong converse type results, similar to those in channel coding theorems. In other words, a typical result should state that unless the number of available samples exceeds some threshold, the probability of error in learning the Markov network structure goes to one as the problem size increases. Also, a graphical structure that often occurs in natural situations is the power-law graph, i.e., a graph whose degree sequence exhibits a power-law or Pareto distribution. The standard property of a power-law graph is as follows – given $\alpha > 1$, the number of nodes with degree k in a power-law graph with exponent α is roughly proportional to $k^{-\alpha}$. Examples of instances where power-law behavior has been observed include social networks [12], protein complex networks [13], gene networks [14] and portions of the internet [15]. As such, many Markov networks derived from natural situations or setups are typically based on power-law graphs. A direction that we explore is the connection between the power-law

exponent α and hardness of the problem of learning Markov graphs, and examination of both the converse and achievability aspects. One of the main hurdles in the problem of designing algorithms for learning power-law graph-based Markov networks is the possibility of large variation in the node degrees; for example, the minimum degree could be constant, while the maximum degree could scale with the number of nodes. We consider the family of power-law graphs generated by the deterministic configuration model [16], and make use of its structural properties for designing the learning algorithm for exact recovery of Markov network topology. In this dissertation, we investigate the relationship between the hardness of learning Markov networks and their structural properties, providing some partial answers for specific ensembles of Markov networks and graphs.

1.2 Main Contributions

We now provide an overview of the main contributions of the dissertation. We analyze the problem of (linear) network coding for setups with multiple sessions. We demonstrate the relationship between networks with multiple sessions and the information-theoretic interference channel, and show that linear network coding coupled with interference alignment techniques can achieve a rate equal or close to $\frac{1}{2}$ per source for a broad class of networks having three unicast sessions with mincuts of one [8, 9, 17]. We extend this idea to networks with multiple multicasts, focus on designing practical coding schemes and examine the impact of network topology on the complexity of the alignment scheme [18]. We show that it is possible to achieve a rate of $\frac{1}{L+d+1}$ per source under certain network con-

straints, using linear network coding coupled with interference alignment, where each destination gets data from L sources, and d depends only on the network.

We consider the problem of deriving information-theoretic limits on the number of samples and probability of error for the problem of learning the graph structure of Markov networks, where we permit distortion in terms of edit distance in the graph estimate [19]. We provide strong converse results for both finite alphabet-based and Gaussian Markov networks based on graphs coming from the ensemble of degree-bounded graphs. We also study the problem of learning the structure of discrete Markov networks, based on power-law graphs generated using the configuration model. We examine the effect of power-law exponent on the hardness of the learning problem and show that it is inherently difficult to learn Markov graphs with smaller power-law exponents, in terms of sample complexity. Furthermore, we design an efficient learning algorithm that accurately reconstructs the graph structure of power-law graph-based Ising model.

1.3 Dissertation Outline

The rest of the dissertation is organized as follows. We provide background on prior literature related to the topics of the dissertation in Chapter 2. We present a detailed description of the work completed in the context of (linear) network coding for multiple sessions and learning the underlying graph structure of Markov networks in Chapters 3 and 4 respectively. Note that the proofs of most of the lemmas and theorems mentioned in these chapters are available in the appendices. Finally, we conclude the dissertation by summarizing it in Chapter 5.

Chapter 2

Background

We present a detailed account of prior literature/background related to the topics mentioned in the dissertation. To be precise, we review the research work that has been done in the context of the problem of communication over networks utilizing network coding, and the problem of learning graph structures using observed data from graphical representations, especially Markov networks.

2.1 Network Communication

The problem of communication over networks is a established and relevant area of research. One aspect of communication networks that has been well-studied is characterizing its capacity region and designing coding schemes that achieve good throughput. There has been a good deal of progress on this front – channel coding theorems and capacity regions have been derived for special cases of network topologies such as broadcast channels [2, 20], multiple access channels [21, 22], relay channels [23, 24] and interference channels [25, 26]. It is also known that the capacity between sources and their destinations in a network is dependent on the minimum cut set (its cardinality is referred to as mincut) between them with respect to the rest of the network [1]. However, ascertaining the capacity region

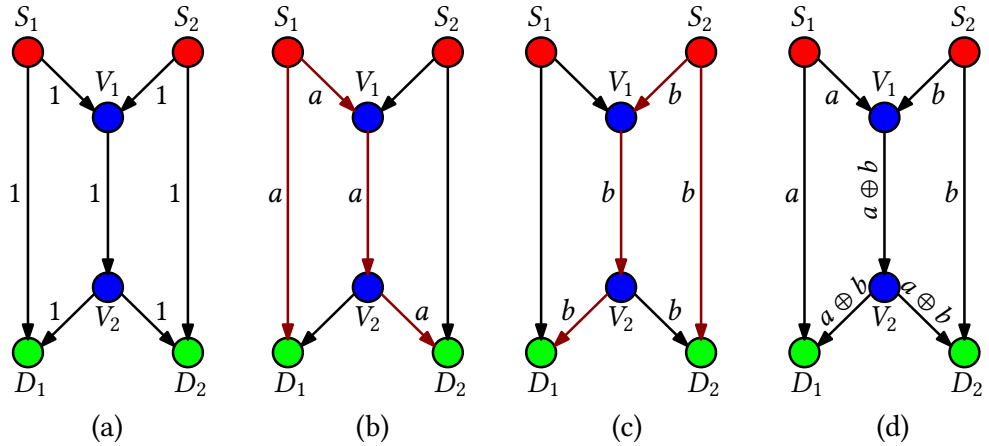


Figure 2.1: Benefits of using network coding over routing.

and achievable throughput for general networks is still an open problem.

The traditional way of transmitting data across networks has been routing, that employs directed paths for unicast sessions and directed trees for multicast sessions. When the data is routed over a unicast path, each intermediate node forwards the packets it receives to its outgoing links. In a multicast session over a tree, the intermediate nodes may duplicate packets and forward them to several outgoing links. It has been shown that routing achieves capacity for the case of a single unicast connection in the network, but proves to be sub-optimal for multicast connection(s). The concept of network coding, introduced in [6, 27], generalizes the routing approach by allowing the intermediate nodes to generate new packets by combining or jointly encoding the data packets they receive. This methodology offers several important benefits such as increase in the achievable throughput and improvement in the reliability and robustness of the network. An example showing the advantage of network coding over routing is the butterfly network,

depicted in Fig 2.1 [28]. The network has two sources, S_1 and S_2, and two destinations, D_1 and D_2. We assume that each directed edge of the network can transmit one packet per time-slot or channel use. With routing approach, the packets are transmitted over two trees – the first tree transmits the packets generated by S_1, and the second tree transmits packets generated by S_2. However, the network does not contain two edge-disjoint trees with S_1 and S_2 as the roots. Hence, multicast sessions involving S_1 and S_2 cannot be implemented through routing. For example, the trees depicted in Figures 2.1(b) and 2.1(c) share the bottleneck edge (V_1, V_2). However, this conflict is resolvable using the network coding approach, as shown in Figure 2.1(d). To demonstrate this, suppose packets a and b (in bits) are transmitted by S_1 and S_2 respectively. These packets are sent to node V_1 which generates a new packet $a \oplus b$ (bitwise-XOR) which is then sent to D_1 and D_2. This allows the each of the destinations to reconstruct both packets a and b.

Linear network coding is a special case of network coding, where packets modeled as elements of a finite field and they are encoded at the intermediate nodes of a network using arithmetic operations of the finite field to form linear combinations. As a result, each destination receives packets that are a linear combination of the packets transmitted by the sources and it recovers the desired packets by solving a system of linear equations over the finite field. In other words, the use of linear network coding provides us a linear transfer function representation of the network [29]. This technique has been shown to achieve the maximum throughput for network setups with one unicast session or multicast/incast session involving one source/destination, where the coefficients for linear combinations can be gen-

erated using a deterministic algorithm [30] or chosen uniformly at random from the finite field [31]. Despite this, linear network coding has been shown to be inadequate in characterizing the limits of inter-session linear network coding [32–35], which includes the practical cases of multiple unicast and multicast sessions.

There is evidence that linear network coding significantly outperforms routing in terms of achievable throughput for networks with multiple sessions [36]. However, there exist only approximation methods for determining the achievable rates in such settings [37], and sub-optimal heuristic methods for constructing linear network codes. For example, an approach, based on coding pairs of flows using poison-antidote butterfly structures and packing networks using these butterflies to improve the throughput, is examined in [38]. The design of sub-optimal linear codes for networks with multiple unicasts, based on linear and integer program methods, is analyzed in [39]. While [40] develops online and off-line back pressure algorithms for finding approximately throughput-optimal network codes within the class of codes restricted to XOR coding between pairs of flows, [41] describes a tiling approach to design codes using dynamic programming for networks with multiple unicasts on a triangular lattice. The feasible and infeasible connectivity levels for networks with unicasts are identified in [42], and network code assignments are provided for the feasible ones. The problem of determining the feasibility and construction of linear network codes for two interacting multicast sessions is analyzed in [43, 44] using a graph-theoretic approach. An important point to note is that most of these approaches are applicable to specific network topologies and they do not give general throughput guarantees.

The simplest, yet non-trivial, example of a network with multiple sessions operating at the same time is the interference channel. The concept of interference alignment, developed in [10], allows one to achieve the optimal degrees of freedom. The basic idea behind interference alignment is to encode the source signals into appropriately designed subspaces such that the subspace containing the desired signals and the subspaces containing the interference signals don't overlap at the destinations, thereby allowing the destinations to recover their desired signals provided the signal-to-noise ratio (SNR) is large. Interestingly, interference alignment has been found to be versatile and has been applied to a wide variety of scenarios, including compound broadcast channels [45], cellular networks [46], relay networks [47], index coding [48, 49], and distributed storage [50–52].

2.2 Learning Markov Networks

Markov networks, also known as (undirected) graphical models, provide an efficient means of compactly encoding probability distributions as undirected graphs in the high-dimensional regime. The random variables in the probability distribution get mapped to nodes of the undirected graph, while the interrelationships (or the lack of thereof) among them get mapped to its edges. As such, Markov networks are widely used for modeling and designing applications in a multitude of settings, for example, social network modeling [53, 54], image processing and computer vision [55, 56] and computational biology [57, 58]. However, with the increasing use of this framework in complex and less well-understood domains, the problem of selecting the most suitable or accurate Markov network from among

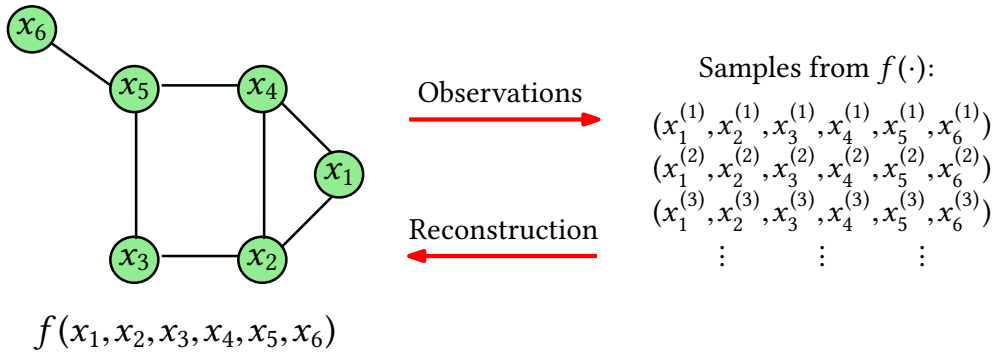


Figure 2.2: Learning Markov networks from given samples.

the exponentially large space of possible network structures has gained a great deal of importance. Thus, the problem of recovering Markov networks from observed samples generated by their probability distributions, also referred to as the graphical model selection problem, is an active area of study and research.

A pictorial view of the problem of learning Markov networks is depicted in Figure 2.2, where a Markov network based on six random variables needs to be learnt from its sample values. As mentioned earlier, this learning problem has two aspects to it – achievability and converse aspects. While the converse aspect is beneficial in the sense that it provides us lower bounds on the sample complexity related to learning, the achievability aspect deals with design of learning algorithms that are efficient and have low probability of error using as minimal samples as possible. There is a significant body of literature related to both these aspects, especially for the specialized cases of Ising model [59–63], and Gaussian Markov networks [64–66]. The graph ensembles that have been considered include degree-bounded graphs [59–61, 64, 66], graphs with limited number of edges [60]

and random graphs such as Erdös-Rényi and small-world graphs [61, 66].

A common theme in deriving information-theoretic limits on the sample complexity is to treat the graphical model selection problem as a noisy channel coding problem and make use of Fano's inequality, that generally gives weak bounds. The only known strong converse results are mentioned in [61] and [67], for the cases of exact reconstruction of Ising model based on Erdös-Rényi graphs and Gaussian Markov networks based on degree-bounded graphs respectively. [68] derives lower bounds on the sample complexity of learning the Markov graph based on two ensembles of power-law graphs, the configuration model [16] and the Chung-Lu model [69], both having power-law exponent greater than 3.

The learning algorithms designed for recovering Markov networks can broadly be classified into three classes – search-based, optimization-based, and greedy techniques. The search-based algorithms find the smallest set of nodes through exhaustive search, conditioned on which a node is independent of other nodes [59, 61, 66]. The optimization-based algorithms frame the learning problem as a convex optimization problem, but require a strong incoherence assumption [65]. The algorithms that use greedy methods discover the neighborhoods of nodes by minimizing some function of the random variables, like conditional entropy, in a greedy fashion [62, 63]. In the context of power-law graphs, [68] examines the performance of these learning algorithms and observes that the sample complexity scales poorly with the number of nodes if the variation in the degrees of nodes is large; it concludes by stating that the task of deriving efficient learning algorithms for power-law structured Markov networks is an outstanding open problem.

Chapter 3

Network Coding for Multiple Sessions

The presence of multiple sessions forms a significant fraction of traffic in most wired and wireless networks today. Therefore, coding schemes that can better utilize network resources to serve multiple connections have many potential applications. In this chapter, we consider the problem of network coding for multiple sessions over networks representable by directed acyclic graphs. In particular, we make use of the linear network coding structure for designing codebooks. As mentioned before, the use of linear network coding results in a linear transfer function representation for the network in terms of its transmission streams; these streams can "mix" with each other and generate "interference" at the destinations [70, 71], that can significantly impact the achievable rates. It is known that the throughput achieved using linear network coding between any set of sources and destinations is upper bounded by the graphical mincut between them; this is also referred to as the generalized mincut-max-flow theorem [29]. A sufficient but somewhat restrictive condition for interference-free transmission in networks employing linear network coding is derived in [70], but it is generally difficult to design coding schemes satisfying the condition for multiple sessions case.

We analyze the problem of designing codebooks from an interference align-

ment perspective and adopt a strategy that couples the concepts of linear network coding (over multiple time-slots) to that of interference alignment, through the use of precoding matrices (or vectors) – we refer to this coding scheme as precoding-based network alignment (PBNA), along the lines of [72, 73]. Note that a similar approach is adopted in the context of analyzing multiple groupcasts for achieving the optimal transmission rates associated with index coding problem [74, 75].

Main Results: We observe that a network with multiple sessions, employing linear network coding as the coding scheme, has a structure similar to that of the interference channel; this allows us to design a PBNA scheme that achieves a rate of $\frac{1}{2}$ per session under certain structural constraints for networks having three unicast sessions with mincut of one per session. We also introduce the concept of interference graph for networks having multiple multicast sessions. We use the interference graph to design precoding matrices – we show that for networks with K sources and mincuts of either zero or one for any source-destination pair, each source can achieve a rate of $\frac{1}{L+1}$ using a PBNA scheme over $(L+1)$ time-slots if the interference graph is acyclic, where every destination is interested in messages from L $(L < K)$ sources and some structural constraints are satisfied. We obtain a weaker achievability result if the interference graph has cycles – we show that a rate of $\frac{1}{L+d+1}$ per source can be achieved with the alignment scheme over $(L+d+1)$ transmissions under certain structural constraints, where d depends only on the topology of interference graph and satisfies $0 \leq d < K-L$. We proceed to develop an algorithm that gives the optimal (or smallest feasible) value of d for a given interference graph, and therefore, reasonable rates for the PBNA scheme.

3.1 System Model and Preliminaries

We consider a communication network represented by a directed acyclic graph $G = (V, E)$, where V is the set of nodes and E is the set of directed links. We assume that each link represents a noiseless channel and transmissions across different links do not interfere with each other. There are K sources $S_1, S_2, \ldots, S_K$, and M destinations $D_1, D_2, \ldots, D_M$, among the nodes in V. We have multiple multicast sessions in G, i.e., D_i is interested in messages from some subset of sources, say $\mathcal{A}_i \subset \{S_1, S_2, \ldots, S_K\}$. For the sake of simplicity, we let $|\mathcal{A}_i| = L$ for all i. Then the special case of multiple unicasts satisfies $K = M$, $L = 1$, and $\mathcal{A}_i = \{S_i\}$ for all i. We assume that the messages generated by different sources are probabilistically independent of each other and transmitted in form of symbols from $\mathbb{F}_q$ – the finite field with q elements, where q is a prime number or its power. We also restrict the capacities of links in E to one symbol (from $\mathbb{F}_q$) per channel use or time-slot.

We employ linear network coding for communication between the sources and destinations in G. In other words, every node generates and transmits linear combinations of its received packets, where the coefficients for linear combination come from $\mathbb{F}_q$. These coefficients can be interpreted as variables, say $\xi_1, \xi_2, \ldots, \xi_s$ (s is determined by G), drawing values from $\mathbb{F}_q$. Then a linear network coding scheme refers to choosing a suitable assignment for $\underline{\xi} := [\xi_1 \ \xi_2 \ \cdots \ \xi_s] \in \mathbb{F}_q^s$.

As a starting point for tackling the problem of designing coding schemes for multiple sessions, we assume that the mincut between S_j and D_i is one if $S_j \in \mathcal{A}_i$, and at most one for remaining choices of i, j – this ensures that D_i is connected to all sources in $\mathcal{A}_i$. We also assume that the mincut between all sources in $\mathcal{A}_i$ and

D_i with respect to G is one, so that D_i can receive at most one symbol per time-slot from them. If $x_i \in \mathbb{F}_p$ is the symbol transmitted by S_i, the following relation holds:

$$y_i = \sum_{j=1}^{K} m_{ij}(\underline{\xi})x_j, \quad i = 1, 2, \ldots, M, \tag{3.1}$$

where y_i is the symbol received by D_i, and $m_{ij}(\underline{\xi})$ is the transfer function between S_j and D_i. Note that y_i and $m_{ij}(\underline{\xi})$ are multivariate polynomials from the polynomial ring $\mathbb{F}_p[\underline{\xi}]$ for all i, j. The transfer functions are determined by the adjacency matrix of the line graph of G; a description about their generation and structure is given in [29]. Since D_i is only interested in messages from sources in $\mathcal{A}_i$, the presence of non-zero transfer functions $m_{ij}(\underline{\xi})$, $S_j \notin \mathcal{A}_i$, acts as "interference" to the decoding processes at the destinations. Note that $m_{ij}(\underline{\xi}) \not\equiv 0$ for $S_j \in \mathcal{A}_i$, since the mincut between each source in $\mathcal{A}_i$ and D_i is one. Also, the mincut between S_j and D_i being zero for some i, j implies that $m_{ij}(\underline{\xi}) \equiv 0$. We define $\mathcal{B}_i = \{S_j \notin \mathcal{A}_i : m_{ij}(\underline{\xi}) \not\equiv 0\}$ – the set of interfering sources for D_i. We also assume $\mathcal{B}_i \neq \emptyset$ – this ensures the presence of interference at each of the destinations.

The generalized max-flow-mincut theorem states that multicast sessions can hope to achieve maximal throughout if there exists an assignment of $\underline{\xi}$, say $\underline{\xi}_0 \in \mathbb{F}_p^s$, such that $m_{ij}(\underline{\xi}_0) = 0$ for $S_j \in \mathcal{B}_i$ and $m_{ij}(\underline{\xi}_0) \neq 0$ for $S_j \in \mathcal{A}_i$. However, there exists a broad class of networks for which such an assignment of $\underline{\xi}$ does not exist, thereby making multiple sessions with maximal data rates infeasible.

3.1.1 Applying Interference Alignment

Note that the relations in (3.1) have a form similar to that of the information-theoretic interference channel, where x_j with $S_j \in \mathcal{B}_i$, play the role of interfering signals and the transfer functions play the role of channel gains. This observation motivates the use of interference alignment techniques for designing coding schemes. However, there are two major points of difference between the two settings. The first point is that transmitted messages come from a finite field in networks, whereas they are real or complex-valued in interference channels. The second point is that the channel gains in interference channel are generated from some probability distribution, whereas the channel gains are transfer functions in the network setup, that are deterministic and influenced by its structure.

As mentioned before, we focus on the application of interference alignment schemes for designing codebooks; for simplicity, we restrict ourselves to codebooks that ensure the sources in $\mathcal{G}$ transmit at equal rates. We consider n successive time-slots and define $\underline{\xi}^{(k)}$ as the assignment of $\underline{\xi}$ for the kth time-slot, $k = 1, 2, \ldots, n$. Given a, b, n such that $a \leq b$ and $n \geq La + b$, we define $\mathbf{z}_i \in \mathbb{F}_q^{a\times 1}$ as the message vector of S_i, and consider a $n \times a$ precoding matrix $\mathbf{V}_i$ that encodes $\mathbf{z}_i$ into n symbols. Then D_i receives a $n \times 1$ vector $\mathbf{y}_i$, satisfying the following relation:

$$\mathbf{y}_i = \sum_{j=1}^{K} \mathbf{M}_{ij}\mathbf{V}_j\mathbf{z}_j, \quad i = 1, 2, \ldots, M. \tag{3.2}$$

Note that $\mathbf{M}_{ij}$ is a $n \times n$ diagonal matrix with $m_{ij}(\underline{\xi}^{(k)})$ as its (k,k)th entry. We define $\underline{\delta}$ as the vector of variables in $\underline{\xi}^{(1)}, \underline{\xi}^{(2)}, \ldots, \underline{\xi}^{(n)}$ and those used in the precoding

matrices. We also define the following vector spaces over polynomial ring $\mathbb{F}_q[\underline{\delta}]$:

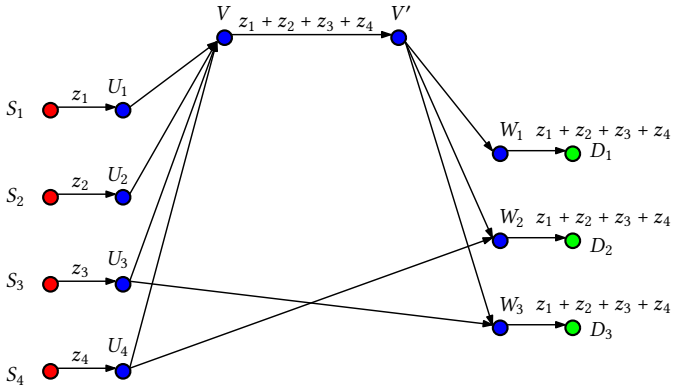
$$\mathcal{U}_i = \text{span}([\mathbf{M}_{ij}\mathbf{V}_j : \ S_j \in \mathcal{A}_i]),$$
$$\mathcal{W}_i = \text{span}([\mathbf{M}_{ij}\mathbf{V}_j : \ S_j \in \mathcal{B}_i]),$$

for $i = 1, 2, \ldots, M$, where span($\mathbf{E}$) denotes the vector space generated by the column vectors of some matrix $\mathbf{E}$. Then the alignment approach seeks to design precoding matrices that satisfy the following conditions for some assignment of $\underline{\delta}$:

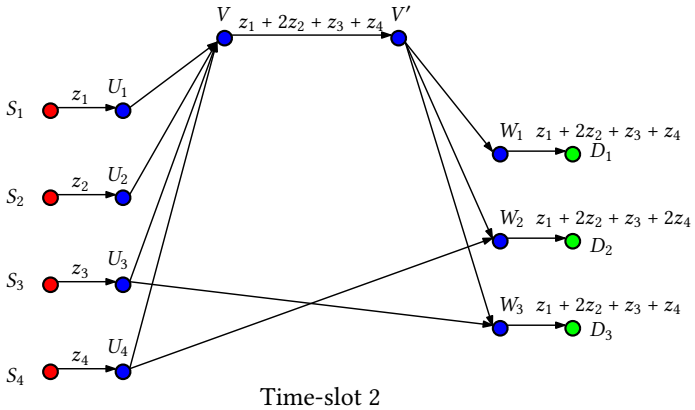
$$\dim(\mathcal{U}_i) = La, \ \dim(\mathcal{W}_i) \leq b, \ \dim(\mathcal{U}_i \cap \mathcal{W}_i) = 0, \tag{3.3}$$

for $i = 1, 2, \ldots, M$, where $\dim(\mathcal{U})$ denotes the dimension number of vector space $\mathcal{U}$. The constraint on the dimension of $\mathcal{W}_i$ maps the interference vectors to a single subspace at each destination. The constraint on the dimension of $\mathcal{U}_i \cap \mathcal{W}_i$ guarantees that the subspace spanned by the interference vectors is linearly independent of the subspace spanned by the desired vectors; this along with the constraint on the dimension of $\mathcal{U}_i$ permits error-free recovery of desired messages at the destinations. Therefore, S_i can transmit a symbols in n successive time-slots, thereby achieving a rate of $\frac{a}{n}$ – this is what we refer to as the PBNA coding scheme.

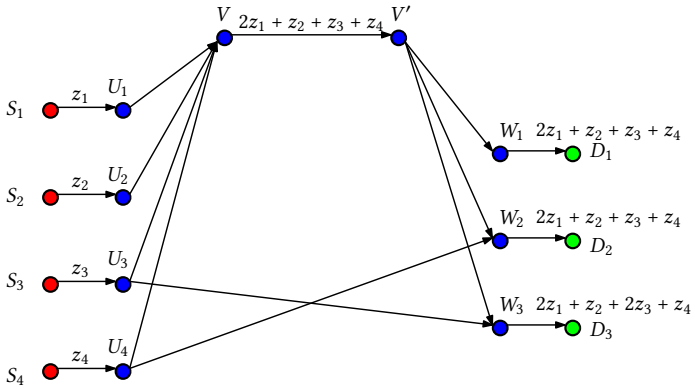
Illustrative Example: We present a network example (see Figure 3.1) that highlights the benefit of using PBNA scheme – note that this is motivated by an analogous example examined in the context of index coding for multiple groupcasts in [74]. There are $K = 4$ sources and $M = 3$ destinations in the network such that $\mathcal{A}_1 = \{S_1, S_2\}$, $\mathcal{A}_2 = \{S_1, S_3\}$, $\mathcal{A}_3 = \{S_2, S_4\}$ (i.e., $L = 2$). Also, the min-cut between each source and destination is one, so that the rate of each source is



Time-slot 1



Time-slot 2



Time-slot 3

Figure 3.1: Network example using PBNA scheme.

upper-bounded by $\frac{1}{2}$. Note that the presence of the bottleneck link (V, V') allows only one source to transmit per time-slot for the routing approach, and therefore, a rate of $\frac{1}{4}$ per source can be achieved via time-sharing. It can also be shown that the linear network coding cannot achieve a rate of $\frac{1}{2}$ per source. However, the use of PBNA scheme enables each source to transmit one message in three time-slots, thereby achieving a rate of $\frac{1}{3}$ per source – this is depicted in Figure 3.1, where linear network coding is performed over finite field $\mathbb{F}_3 = \{0, 1, 2\}$, and $z_i \in \mathbb{F}_3$ is the scalar message for S_i. Then D_i receives $\mathbf{y}_i \in \mathbb{F}_3^3$ across three time-slots, given by

$$\begin{aligned}
\mathbf{y}_1 &= [1 \;\; 1 \;\; 2]^t z_1 + [1 \;\; 2 \;\; 1]^t z_2 + [1 \;\; 1 \;\; 1]^t (z_3 + z_4), \\
\mathbf{y}_2 &= [1 \;\; 1 \;\; 2]^t z_1 + [1 \;\; 1 \;\; 1]^t z_3 + [1 \;\; 2 \;\; 1]^t (z_2 + z_4), \\
\mathbf{y}_3 &= [1 \;\; 2 \;\; 1]^t z_2 + [1 \;\; 1 \;\; 1]^t z_4 + [1 \;\; 1 \;\; 2]^t (z_1 + z_3).
\end{aligned}$$

Note that $\mathcal{U}_1 = \text{span}([1 \;\; 1 \;\; 2]^t, [1 \;\; 2 \;\; 1]^t)$ and $\mathcal{W}_1 = \text{span}([1 \;\; 1 \;\; 1]^t)$, from construction of $\mathbf{y}_1$. Thus, $\mathcal{U}_1$ and $\mathcal{W}_1$ are linearly independent vector spaces with $\dim(\mathcal{U}_1) = 2$ and $\dim(\mathcal{W}_1) = 1$ – this allows D_1 to recover the messages from sources in $\mathcal{A}_1$ (for example, by taking inner products of $\mathbf{y}_1$ with $[1 \;\; 0 \;\; 2]^t$ and $[1 \;\; 2 \;\; 0]^t$). Likewise, D_2 and D_3 can recover messages from sources in $\mathcal{A}_2$ and $\mathcal{A}_3$ respectively; this completes the PBNA scheme description achieving a sum rate of $\frac{4}{3}$ (this can be shown to be the optimal sum rate for this network example).

3.2 Network Coding for Unicast Sessions

In this section, we describe the PBNA scheme for networks with multiple unicast sessions. In particular, we focus on a special case – networks that have

three unicast sessions, i.e., $K = M = 3$, $L = 1$, $\mathcal{A}_i = \{S_i\}$ and $\mathcal{B}_i = \{S_1, S_2, S_3\} \backslash \{S_i\}$ for all i; this is the smallest non-trivial instance of the problem and therefore, its analysis can be used as stepping stone for better understanding of the performance of linear network coding schemes in the scenario of unicast connections.

3.2.1 Results for Achievable Data Rates

We first consider the case where all the transfer functions of the network with three unicast sessions are non-zero polynomials; we handle the case where some of the interference transfer functions are zero later. Note that since $L = 1$, one of the ways to achieve data rate close to $\frac{1}{2}$ per session is to design precoding matrices with a close to b and n close to $a + b$ in values (so that $\frac{a}{n}$ is close to $\frac{1}{2}$). Next, we define the following rational functions, based on the transfer functions:

$$p_1(\underline{\xi}) = \frac{m_{11}(\underline{\xi})m_{32}(\underline{\xi})}{m_{12}(\underline{\xi})m_{31}(\underline{\xi})}, \quad p_2(\underline{\xi}) = \frac{m_{22}(\underline{\xi})m_{31}(\underline{\xi})}{m_{21}(\underline{\xi})m_{32}(\underline{\xi})},$$
$$p_3(\underline{\xi}) = \frac{m_{33}(\underline{\xi})m_{21}(\underline{\xi})}{m_{23}(\underline{\xi})m_{31}(\underline{\xi})}, \quad t(x) = \frac{m_{12}(\underline{\xi})m_{23}(\underline{\xi})m_{31}(\underline{\xi})}{m_{13}(\underline{\xi})m_{32}(\underline{\xi})m_{21}(\underline{\xi})}.$$

Given a positive integer v, we also define the following set of rational functions:

$$\mathcal{S}_v = \left\{ \frac{f(t(\underline{\xi}))}{g(t(\underline{\xi}))} : f(x), g(x) \in \mathbb{F}_q[x], \gcd(f,g) = 1, \deg(f) \leq v, \deg(g) \leq v \right\}.$$

We consider two cases depending on whether $t(\underline{\xi})$ is a constant in $\mathbb{F}_p$ or not.

Case I: $t(\underline{\xi})$ is not a constant. Here, we choose $a = v$, $b = v + 1$ and $n = 2v + 1$. Thereafter, we consider the following choice of precoding matrices:

$$\mathbf{V}_1 = [\mathbf{e} \quad \mathbf{Te} \quad \mathbf{T}^2\mathbf{e} \quad \cdots \quad \mathbf{T}^{v-1}\mathbf{e}], \tag{3.4}$$

$$\mathbf{V}_2 = \mathbf{M}_{31}\mathbf{M}_{32}^{-1}[\mathbf{e} \quad \mathbf{Te} \quad \mathbf{T}^2\mathbf{e} \ \cdots \ \mathbf{T}^{v-1}\mathbf{e}], \tag{3.5}$$

$$\mathbf{V}_3 = \mathbf{M}_{21}\mathbf{M}_{23}^{-1}[\mathbf{Te} \quad \mathbf{T}^2\mathbf{e} \quad \mathbf{T}^3\mathbf{e} \ \cdots \ \mathbf{T}^{v}\mathbf{e}], \tag{3.6}$$

where $\mathbf{e}$ is the vector of $(2v+1)$ ones and $\mathbf{T}$ is the $(2v+1) \times (2v+1)$ diagonal matrix with its (k,k)th element as $t(\underline{\xi}^{(k)})$. Then the following result holds good:

Theorem 3.2.1. *If $t(\underline{\xi})$ is not a constant and $p_i(\underline{\xi}) \notin \mathcal{S}_v$, $i = 1,2,3$, for some positive integer v, then one can achieve a rate of $\frac{v}{2v+1}$ per source using PBNA scheme with large enough $\mathbb{F}_q$, for networks with three unicast sessions and mincuts of one.*

Proof. Refer to Appendix A.1 □

Note that as $v \to \infty$, the achievable rate tuple in Theorem 3.2.1 approaches $\left(\frac{1}{2}, \frac{1}{2}, \frac{1}{2}\right)$. In other words, each unicast session achieves a rate of $\frac{1}{2}$ in an asymptotic fashion, if v can be chosen arbitrarily large. Therefore, we have the following corollary:

Corollary 3.2.2. *If $t(\underline{\xi})$ is not a constant and $p_i(\underline{\xi}) \notin \mathcal{S}_v$, $i = 1,2,3$, for all positive integers v, then one can achieve a rate of $\frac{1}{2}$ per source using PBNA scheme with large enough $\mathbb{F}_q$, for networks with three unicast sessions and mincuts of one.*

One possibility where $p_i(\underline{\xi}) \notin \mathcal{S}_v$, $i = 1,2,3$, gets satisfied is when each of $m_{ii}(\underline{\xi})$, $i = 1,2,3$, possesses a variable that does not occur in $m_{ij}(\underline{\xi})$, $i \neq j$. Interestingly, it turns out that one can achieve a rate of $\frac{1}{2}$ per source for this case using a coding scheme across two time-slots, inspired by the ergodic alignment scheme [11].

Note that checking the membership of $p_i(\underline{\xi})$, $i = 1,2,3$, in S_v requires that we test polynomial equality/inequality for $3|\mathcal{S}_v| \leq 3q^{2v+2}$ instances. This can

become cumbersome if v is large. Generally, transfer functions tend to be well-structured; hence, it is possible to make the assumptions on $p_i(\underline{\xi})$'s less restrictive, that can be checked efficiently. This fact is shown in [17], where $\mathcal{S}_v$ is replaced by

$$\mathcal{S} = \left\{1,\ t(\underline{\xi}),\ 1 + t(\underline{\xi}),\ \frac{t(\underline{\xi})}{1 + t(\underline{\xi})}\right\},$$

provided finite field size q is a power of 2. The proof of this involves careful degree counting and identifying the graph-related properties of transfer functions.

Case II: $t(\underline{\xi})$ is a constant. Here, we choose $a = b = 1$ and $n = 2$. If $t(\underline{\xi}) \equiv c \in \mathbb{F}_q$, with $c \neq 0$, we consider the following choice of precoding matrices:

$$\mathbf{V}_1 = [1 \quad 1]^T, \quad \mathbf{V}_2 = \mathbf{M}_{31}\mathbf{M}_{32}^{-1}[1 \quad 1]^T, \quad \mathbf{V}_3 = \mathbf{M}_{21}\mathbf{M}_{23}^{-1}[c \quad c]^T. \tag{3.7}$$

Thus, the coding scheme is based on two time-slots and the following result holds:

Theorem 3.2.3. *If $t(\underline{\xi})$ is a constant in $\mathbb{F}_q$ and $p_i(\underline{\xi})$, $i = 1, 2, 3$, are non-constants, then one can achieve a rate of $\frac{1}{2}$ per source using PBNA scheme in two time-slots with large enough $\mathbb{F}_q$, for networks with three unicast sessions and mincuts of one.*

Proof. Refer to Appendix A.2 □

Next, we consider the case when some of the interference transfer functions are zero (note that $m_{ii}(\underline{\xi}) \not\equiv 0$, else communication between S_i and D_i would not be possible). Such a situation is desirable in the context of designing PBNA schemes as it eliminates the need of satisfying at least one of the alignment conditions. Then we can choose one of the precoding matrices freely and the PBNA scheme gets simplified – we set $a = b = 1$, $n = 2$, and the following result holds:

Theorem 3.2.4. *If some interference transfer function, $m_{ij}(\underline{\xi})$, $i \neq j$, equals zero, then one can achieve a rate of $\frac{1}{2}$ per source using PBNA scheme in two time-slots with large enough $\mathbb{F}_q$, for networks with three unicast sessions and mincuts of one.*

Proof. Refer to Appendix A.3 □

Summary: We develop a systematic mechanism for analyzing the achievable rates for networks with unicast sessions. We show that under certain conditions, a rate of half the mincut per session can be achieved in a network with three unicast sessions and mincuts of one. The primary ingredient in designing the coding scheme is the combination of the notions of interference alignment, borrowed from interference channel literature, and linear network coding.

3.3 Network Coding for Multicast Sessions

In this section, we describe the PBNA scheme for networks with multiple multicast sessions. For this, we use the notion of interference graph whose structural properties influences the achievable rates of any coding strategy.

Interference Graph: We consider an undirected bipartite graph $\mathcal{H} = (\mathcal{X}, \mathcal{Y}, \mathcal{F})$, where $\mathcal{X} = \{S_1, S_2, \ldots, S_K\}$, $\mathcal{Y} = \{\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_M\}$ are the node partitions, and $\mathcal{F}$ is the set of undirected edges such that $(S_j, \mathcal{W}_i) \in \mathcal{F}$ iff $S_j \in \mathcal{B}_i$. Thus, $\mathcal{H}$ encodes the set of sources whose signals act as interference, and therefore, need to be aligned to a single subspace at each destination - hence, we refer to it as the interference graph. The topology of $\mathcal{H}$ has a direct bearing on the achievable rates of the sources; for example, abundant low-degree nodes in $\mathcal{Y}$ and smaller values of

$|\mathcal{F}|$ could result in potentially higher achievable rates due to lesser number of interference terms (and alignment constraints) at the destinations. The interference graph of the network example in Figure 3.1 is illustrated in Figure 3.2(a).

3.3.1 Results for Achievable Data Rates

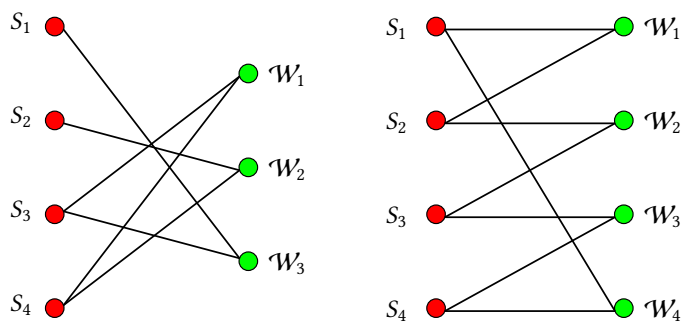
We analyze the connection between structural properties of the interference graph and achievable source rates. We first consider the case where the interference graph $\mathcal{H}$ has no cycles. Then we have the following achievability result:

Theorem 3.3.1. *If $\mathcal{H}$ has no cycles, then one can achieve a rate of $\frac{1}{L+1}$ per source in $(L+1)$ time-slots using PBNA scheme with large enough $\mathbb{F}_q$, under certain constraints that are checkable in time polynomial in $L, |\mathcal{F}|$ and transfer function degrees.*

Proof. Refer to Appendix A.4. □

The absence of cycles in the interference graph is advantageous in the sense that it enables one to choose a set of precoding matrices/vectors independently of each other and use them to construct precoding matrices/vectors for the remaining sources. Also, feasibilty of the PBNA scheme can be checked in polynomial time (in terms of network parameters) and coding is practical as it uses $(L + 1)$ time-slots. Thus, the sources can achieve a sum rate of $\frac{K}{L+1}$ and each destination can receive data from the desired sources at a rate of $\frac{L}{L+1}$ using PBNA strategy.

The presence of cycles in the interference graph can impose restrictions on the choice of precoding matrices that may the affect the ease of satisfying alignment constraints. We illustrate this using a network with $K = M = 4$ and $L = 2$ –



(a) (b)

Figure 3.2: Examples of some interference graphs.

we set $\mathcal{A}_1 = \{S_3, S_4\}$, $\mathcal{A}_2 = \{S_1, S_4\}$, $\mathcal{A}_3 = \{S_1, S_2\}$, $\mathcal{A}_4 = \{S_2, S_3\}$, and assume all the transfer functions are non-trivial. We also define the following rational function:

$$t(\underline{\xi}) \equiv \frac{m_{12}(\underline{\xi})m_{23}(\underline{\xi})m_{34}(\underline{\xi})m_{41}(\underline{\xi})}{m_{11}(\underline{\xi})m_{22}(\underline{\xi})m_{33}(\underline{\xi})m_{44}(\underline{\xi})}.$$

The interference graph for this network is depicted in Figure 3.2(b); it is easy to see that the graph $\mathcal{H}$ is a cycle; we have the following negative result for this setup:

Theorem 3.3.2. *If $\mathcal{H}$ is the cycle interference graph of the network described above (see Figure 3.2(b)) and $t(\underline{\xi})$ is a non-constant rational function (i.e., $t(\underline{\xi}) \not\equiv a$, $a \in \mathbb{F}_q$), then one cannot achieve a rate of $\frac{1}{3}$ per source in finite number of time-slots.*

Proof. Refer to Appendix A.5. □

Thus, the presence of cycles in the interference graph can result in the PBNA scheme requiring large number of time-slots for each source to achieve a rate close to $\frac{1}{L+1}$ and sum rate close to $\frac{K}{L+1}$. One way of tackling this problem is to allow the destinations to decode some of the interference messages, i.e., D_i agrees

to decode messages from some sources in $\mathcal{B}_i$ along with those from sources in $\mathcal{A}_i$. This approach reduces the number of relations in (3.3) to be satisfied, thereby effectively removing edges from the interference graph $\mathcal{H}$. For example, if D_i decodes messages from $S_j \in \mathcal{B}_i$, the alignment constraints involving $\mathbf{V}_j$ that need to be satisfied at D_i get eliminated; this is equivalent to removing $(S_j, \mathcal{W}_i) \in \mathcal{F}$ from $\mathcal{H}$. However, the tradeoff of this approach is reduction in the source rates since each destination needs to decode potentially more than L messages.

We define $\mathcal{E}_i \subseteq \mathcal{B}_i$ as the set of extra sources whose messages are decoded by D_i, so that D_i now recovers messages from sources in $\bar{\mathcal{A}}_i = \mathcal{A}_i \cup \mathcal{E}_i$, and the new interfering set of sources for D_i is $\bar{\mathcal{B}}_i = \mathcal{B}_i \setminus \mathcal{E}_i$. These updates are equivalent to the process of removing edges in $\{(S_j, \mathcal{W}_i) : S_j \in \mathcal{E}_i\}$ from $\mathcal{H}$ to get a new interference graph $\bar{\mathcal{H}} = (\mathcal{X}, \mathcal{Y}, \bar{\mathcal{F}})$, where $\bar{\mathcal{F}} = \{(S_j, \mathcal{W}_i) : S_j \in \bar{\mathcal{B}}_i\}$. Our objective is to remove these edges in such a way that cycles are eliminated from $\mathcal{H}$ and resultant $\bar{\mathcal{H}}$ is acyclic in nature – thereafter, we can use PBNA scheme to provide guarantees on achievable source rates. In particular, we have the following achievability result:

Theorem 3.3.3. *Suppose $\bar{\mathcal{H}}$, generated from $\mathcal{H}$ as described above, has no cycles, and let $d = \max_{1 \le i \le M} |\mathcal{E}_i|$. Then one can achieve a rate of $\frac{1}{L+d+1}$ per source in $(L+d+1)$ time-slots using PBNA scheme with large enough $\mathbb{F}_q$, under certain constraints that are checkable in time polynomial in L, d, $|\bar{\mathcal{F}}|$ and transfer function degrees.*

Proof. Refer to Appendix A.6 □

Note that if $\mathcal{H}$ has cycles, there can be multiple candidates for subgraph $\bar{\mathcal{H}}$ that has no cycles. Since we want to maximize the data rates for the sources,

we are interested in the smallest value that d can take – we refer to this optimal value as d^*. Therefore, we need to solve the following graph-theoretic optimization problem over $\mathcal{H}$ – what is the minimum value of d so that if we remove some set of $\min(d, |\mathcal{B}_i|)$ edges from node $\mathcal{W}_i \in \mathcal{Y}$ ($|\mathcal{B}_i|$ is the degree of node $\mathcal{W}_i$), the resulting graph $\bar{\mathcal{H}}$ has no cycles? We first assume that $\mathcal{H}$ is a connected graph. Then a modified optimization problem, that gives the same optimal value d^*, is as follows – what is the minimum value of d so that if we remove at most d edges from each node in $\mathcal{Y}$, the resulting subgraph $\mathcal{K}$ is a spanning tree of $\mathcal{H}$? We denote the optimal $\bar{\mathcal{H}}$ and $\mathcal{K}$, obtained as solutions to these optimization problems, by $\bar{\mathcal{H}}^*$ and $\mathcal{K}^*$ respectively. Note that $\bar{\mathcal{H}}^*$ can obtained from $\mathcal{K}^*$ by removing edges from $\mathcal{K}^*$, if needed, such that the difference between degrees of $\mathcal{W}_i$ in $\mathcal{H}$ and $\bar{\mathcal{H}}^*$ is $\min(d^*, |\mathcal{B}_i|)$ for all i. In other words, it suffices to obtain $\mathcal{K}^*$ from $\mathcal{H}$.

We use the concepts from matroid theory to design an algorithm that outputs $\mathcal{K}^*$ for a given $\mathcal{H}$. For this, we recall the definitions of matroid and its dual:

Definition 3.3.1. Consider a finite set E and a family of subsets of E, denoted by $\mathcal{I}$. Then $\mathcal{M} = (E, \mathcal{I})$ is a matroid if (a) $\emptyset \in \mathcal{I}$ ($\emptyset$ is null set), (b) $A \in \mathcal{I} \Rightarrow B \in \mathcal{I}$, where $B \subseteq A$, and (c) there exists $x \in A$ with $B \cup \{x\} \in \mathcal{I}$, if $A, B \in \mathcal{I}$ and $|A| > |B|$. The elements of $\mathcal{I}$ are called independent sets, and an independent set having the largest number of elements is called a basis. Also, the dual of $\mathcal{M}$ is another matroid $\bar{\mathcal{M}}$ such that for every independent set in it, there is a disjoint basis in $\mathcal{M}$.

A detailed description about properties of matroids can be found in [76].

Since $\mathcal{H}$ is connected, we consider its graphic matroid $\mathcal{M}$ – its indepen-

dent set is an edge-set of $\mathcal{H}$ that forms a tree or forest subgraph of $\mathcal{H}$ (i.e., the independent set forms a subgraph with no cycles). The bases of $\mathcal{M}$ are the edge-sets that form spanning trees of $\mathcal{H}$. Then an independent set of the dual of $\mathcal{M}$, denoted by $\bar{\mathcal{M}}$, is an edge-set of $\mathcal{H}$ whose complement form a subgraph containing some spanning tree of $\mathcal{H}$ (i.e., the independent set forms a subgraph whose complement is connected). Also, given positive integer d, we consider a partition matroid $\mathcal{M}_d$ – its independent set is an edge-set of $\mathcal{H}$ such that at most d edges are chosen from every node in $\mathcal{Y}$. The bases of $\mathcal{M}_d$ are edge-sets of $\mathcal{H}$ that form subgraphs with $\min(d, |\mathcal{B}_i|)$ edges incident on node $\mathcal{W}_i \in \mathcal{Y}$ for all i.

Note that $\bar{\mathcal{M}} \cap \mathcal{M}_d$ is a matroid – its independent set is an edge-set of $\mathcal{H}$ such that has at most d edges incident at every node in $\mathcal{Y}$ and whose complement forms a subgraph containing a spanning tree of $\mathcal{H}$. Thus, the problems of finding d^* and $\mathcal{K}^*$ reduces to finding the minimum value of d for which the complement of a basis of $\bar{\mathcal{M}} \cap \mathcal{M}_d$ forms a spanning tree of $\mathcal{H}$. The minimum value of d is equal to d^*, and the edges in that basis need to be removed from $\mathcal{H}$ for obtaining $\mathcal{K}^*$. In other words, if $\mathcal{I}$ is the basis of $\bar{\mathcal{M}} \cap \mathcal{M}_{d^*}$, then $\mathcal{K}^* = (\mathcal{X}, \mathcal{Y}, \mathcal{F} \setminus \mathcal{I})$.

We define an arbitrary labeling of edges of $\mathcal{H}$ as $\mathcal{F} = \{e_1, e_2, \ldots, e_{|\mathcal{F}|}\}$, then we can use Algorithm 1 to get d^* and $\mathcal{K}^*$ for a given connected graph $\mathcal{H}$.

Theorem 3.3.4. *Given any (connected) bipartite graph $\mathcal{H}$, Algorithm 1 finds d^* and spanning tree $\mathcal{K}^*$ for $\mathcal{H}$, with a computational complexity of $O(\left(1 + \frac{K}{M}\right)|\mathcal{F}|^2)$.*

Proof. Refer to Appendix A.7. □

Algorithm 1 Finding d^* and $\mathcal{K}^*$ for connected graph $\mathcal{H}$

Initialize: $\mathcal{H} = (\mathcal{X}, \mathcal{Y}, \mathcal{F})$, $\mathcal{F} = \{e_1, e_2, \ldots, e_{|\mathcal{F}|}\}$
  for $d = \lceil (|\mathcal{F}| - K - M + 1)/M \rceil$ to $\lfloor |\mathcal{F}|/M \rfloor$ **do**
    $\mathcal{I} \leftarrow \emptyset$
    for $i = 1$ to $|\mathcal{F}|$ **do**
      if $\mathcal{I} \cup \{e_i\} \in \bar{\mathcal{M}} \cap \mathcal{M}_d$ **then**
        $\mathcal{I} \leftarrow \mathcal{I} \cup \{e_i\}$
      end if
    end for
    if $|\mathcal{I}| = |\mathcal{F}| - K - M + 1$ **then**
      break
    end if
  end for
Output: $d^* = d$, $\mathcal{K}^* = (\mathcal{X}, \mathcal{Y}, \mathcal{F} \backslash \mathcal{I})$

In case $\mathcal{H}$ is not a connected component, we can apply Algorithm 1 on its disjoint components separately and obtain their corresponding optimal values of d and optimal spanning trees. Then d^* is the maximum of the optimal values of d obtained for the disjoint components, and $\bar{\mathcal{H}}^*$ can be obtained using an edge removal process from the set of disjoint optimal trees similar to the one used for the case of connected graph $\mathcal{H}$. Also, if number of disjoint components of $\mathcal{H}$ is c, the time complexity for running Algorithm 1 over $\mathcal{H}$ is $O(c\left(1 + \frac{K}{M}\right)|\mathcal{F}|^2)$.

Summary: We describe a systematic mechanism for providing guarantees on achievable data rates for networks employing linear network coding with multiple multicast sessions. We use the PBNA scheme for designing codebooks that use finite number of time-slots for networks with acyclic interference graphs. For networks with cyclic interference graphs, we present a graph sparsification approach that optimally removes cycles and gives reasonable achievable rates.

Chapter 4

Learning Structure of Markov Networks

The usefulness of Markov networks in efficiently encoding probability distributions with large number of random variables in form of undirected graphs has led to its widespread adoption for modeling and designing applications in fields like social network analysis [53, 54], image processing/computer vision [55, 56] and computational biology [57, 58]. The problem of learning the graph structure of any Markov network from samples generated by its underlying probability distribution is a well-studied one and is referred to as the problem of learning Markov graphs or graphical model selection. A tractable approach of tackling this learning problem is to interpret it as a channel coding problem, as done in information theory, and derive the necessary and sufficient conditions for error-free recovery of Markov graphs with respect to a given learning algorithm. There is diverse literature associated with the two aspects of graphical model selection for specific ensembles of Markov networks – the achievability aspect dealing with the design and analysis of efficient and near-optimal (in terms of sample and computational complexity) learning algorithms/estimators, and the converse aspect dealing with deriving information-theoretic lower bounds on sample complexity for any learning algorithm to correctly reconstruct the structure of Markov networks.

Main Results: We analyze the problem of learning Markov graphs from a rate-distortion theoretic perspective, and provide lower bounds on the number of samples required for any algorithm to learn the Markov graph structure of a probability distribution, up to a pre-specified edit distance. In particular, for both discrete and Gaussian models on p variables with degree at most d, we show that at least $\Omega\left(\left(d - \frac{s}{p}\right)\log_2 p\right)$ samples are required for any algorithm to learn the graph structure up to edit distance s. Our bounds represent a strong converse; i.e., we show that for a smaller number of samples, the probability of error of any learning algorithm goes to one as the problem size increases. In this sense, our results have stronger consequences than the traditional ones obtained using Fano's inequality (the typical result here is that the probability of error is bounded away from zero, like $\geq \frac{1}{2}$, with increasing problem size). Moreover, our bounds indicate that substantial gains in sample complexity may not be possible without paying a significant price in edit distance error. We also take a look at the problem of accurately learning discrete Markov networks based on power-law graphs generated by the configuration model, i.e., networks whose degree sequence follow a power-law distribution. It has been observed that power-law graphs crop up in a number of real-world scenarios, e.g., internet graphs, biological networks and gene associations. We examine the effect of power-law exponent on the limits of sample complexity, and use the converse aspect to show that learning algorithms require more samples (in order-wise sense) to exactly recover Markov graphs having low values of power-law exponents. In the context of designing algorithms for power-law Markov networks, a major challenge faced in the learning process is that the

degrees may not always be bounded by constants and the maximum degree could scale with number of nodes. We design an algorithm, similar to conditional variation distance thresholding in structure, and show that ferromagnetic Ising model on power-law graphs with p variables and exponent greater than 3 can be learnt exactly using $\Omega(\log_2 p)$ samples that is order-wise optimal if the minimum degree scales like a constant. In case the power-law exponent lies between 2 and 3, we get a sample complexity requirement that is poly-log in the number of nodes ($\Omega((\log_2 p)^3)$, to be precise) under certain constraints on the degree sequence.

4.1 System Model and Preliminaries

We consider an undirected graph $G = (V, E)$, where $V = \{1, \ldots, p\}$ is the set of nodes and E is the set of edges. A Markov network is obtained by associating a random variable X_i to $i \in V$, that takes values from alphabet $\mathcal{A}$, and specifying a joint probability distribution $f(\cdot)$ over vector $X = (X_1, X_2, \ldots, X_p)$ that satisfies

$$f(x_A, x_B | x_C) = f(x_A | x_C) f(x_B | x_C),$$

where A, B and C are any disjoint subsets of V such that every path from a node in A to a node in B passes through a node in C (C is also called a separator set for A, B), and x_A, x_B, x_C denote the restrictions of $(x_1, \ldots, x_p) \in \mathcal{A}^p$ to indices in A, B, C respectively. Note that $f(\cdot)$ denotes the probability mass function (p.m.f.) for the case of discrete Markov networks (i.e., $|\mathcal{A}| < \infty$) and probability density function (p.d.f.) for the case of continuous Markov networks (i.e., $|\mathcal{A}| = \mathbb{R}$ or $\mathbb{C}$). Next, we present examples of families of discrete and continuous Markov networks.

Ising Model: This is a well-known example of family of discrete Markov networks that is studied in a multitude of fields like statistical physics [77], computer vision [78] and algorithmic game theory [79]. This discrete model is obtained by setting $\mathcal{A} = \{-1, 1\}$, and assigning real-valued node potentials h_i to $i \in V$ and edge potentials θ_{ij} to $(i, j) \in E$. Then the p.m.f. of X satisfies the following relation:

$$f(x) \propto \exp\left(\sum_{i \in V} h_i x_i + \sum_{(i,j) \in E} \theta_{ij} x_i x_j\right).$$

The Ising model is said to be zero-field if $h_i = 0$ for all $i \in V$. Also, the Ising model is said to be ferromagnetic if $\theta_{ij} > 0$ for all $(i, j) \in E$. Note that the normalization constant depends on the graph structure and values of edge/node potentials.

Gaussian Markov Networks: This is a well-known example of family of continuous Markov networks; here, X possesses a real-valued multivariate Gaussian distribution (i.e., $\mathcal{A} = \mathbb{R}$). Without loss of generality, we assume that X has the zero vector as its mean. Given a $p \times p$ positive definite matrix Θ satisfying $\Theta(i, j) = 0$ iff $(i, j) \notin E$ ($\Theta(i, j)$ is the (i, j)th entry of Θ), the p.d.f. of X is given by

$$f(x) = \frac{1}{\sqrt{(2\pi)^p |\Theta^{-1}|}} \exp\left(-\frac{1}{2} x^T \Theta x\right),$$

Note that Θ serves as the inverse covariance matrix of X; it is also referred to as the potential matrix, since $\Theta(i, j)$ acts as the potential of $(i, j) \in E$. We also define

$$\lambda^*(\Theta) \triangleq \min_{(i,j) \in E} \frac{|\Theta(i, j)|}{\sqrt{\Theta(i, i)\Theta(j, j)}}.$$

Note that this quantity is invariant to rescaling of the random variables, and it plays an important role in determining the bounds on sample complexity.

We restrict our attention to Markov networks based on two graph ensembles. The first ensemble is that of degree-bounded graphs – the set of graphs whose node degrees are bounded above by a pre-specified quantity. The second ensemble is that of power-law graphs generated using the configuration model [16]. Our focus on degree-bounded graphs is in light of the fact that there has been extensive work on learning Markov networks based on these graphs. Our focus on power-law graphs stems from the observation of the emergence of power-law behavior in natural networks like social networks and protein interaction networks.

4.1.1 Learning Algorithm and Error Criterion

We denote the set of all undirected graphs on p nodes by $\mathcal{U}_p$. Given two undirected graphs H and H' on the same set of nodes, we define edit distance $\Delta(H,H')$ as the minimum number of edge additions and/or deletions needed for changing H to H'. Next, we describe the system setup for analyzing the problem of learning structure of Markov networks. We consider an ensemble of M undirected graphs on a common set of p nodes, $\mathcal{G} = \{G_1,\ldots,G_M\}$, and an ensemble of M Markov networks $\mathcal{K} = \{K_1,\ldots,K_M\}$, such that K_i has G_i as its underlying graph and the random variables in $X = (X_1,\ldots,X_p)$ draw values from alphabet $\mathcal{A}$. We choose a Markov network $K \in \mathcal{K}$ uniformly at random and obtain n i.i.d. vector samples $X^n = (X^{(1)},\ldots,X^{(n)})$ from the distribution of K. Our objective is to reconstruct G, the underlying graph of K, using the samples X^n. A learning algorithm is any function $\phi : \mathcal{A}^{np} \to \mathcal{U}_p$ that maps the observed samples to a graph estimate $\hat{G} = \phi(X^n) \in \mathcal{U}_p$. Given a non-negative integer s, we define the error

event for the learning algorithm as $\{\Delta(G,\hat{G}) \geq s\} = \{\Delta(G,\phi(X^n)) \geq s\}$, i.e., error occurs if the edit distance between the actual graph and the reconstructed version is at least s. Therefore, the probability of error of learning algorithm ϕ is given by

$$P_{e,s}^{(n)}(\phi) = P(\Delta(G,\phi(X^n)) \geq s) = \frac{1}{M}\sum_{i=1}^{M} P\left(\Delta(G_i,\phi(X^n)) \geq s | K = K_i\right).$$

A learning algorithm requires $s = 0$ to ensure exact recovery of the Markov graph; we denote the probability of error as $P_e^{(n)}(\phi) = P_{e,0}^{(n)}(\phi)$. As mentioned before, the converse aspect of the graphical model selection problem is concerned with finding lower bounds on the probability of error for any ϕ, in terms of n and the parameters associated with $\mathcal{K}$. Also, the achievability aspect is concerned with designing ϕ such that its probability of error can be made arbitrarily small.

4.2 Learning Markov Graphs up to Edit Distance

In this section, we present strong converse results for the problem of learning Markov graphs up to a pre-specified edit distance for families of discrete and Gaussian Markov networks based on degree-bounded graphs. First, we derive general lower bounds on the sample complexity for ensemble of Markov networks based on arbitrary graphs. Thereafter, we use these bounds to derive results for the specialized ensembles of Markov networks on degree-bounded graphs.

4.2.1 Markov Networks on General Graphs

We define the following quantities for graph ensemble $\mathcal{G}$ and any $G \in \mathcal{G}$:

$$\mathcal{B}(s,G) \triangleq \{H : \Delta(G,H) < s, H \in \mathcal{U}_p\}, \quad B(s,\mathcal{G}) \triangleq \max_{G\in\mathcal{G}} |\mathcal{B}(s,G)|.$$

Note that $\mathcal{B}(s, G)$ denotes the set of graphs that are at an edit distance less than s from G and $B(s, \mathcal{G})$ denotes the maximum set size among such $\mathcal{B}(s, G)$'s, $G \in \mathcal{G}$. We also define another quantity, analogous to mutual information in structure:

$$I(K_i; X^{(1)}) \triangleq \begin{cases} H(X^{(1)}) - H(X^{(1)}|K = K_i), & |\mathcal{A}| < \infty, \\ h(X^{(1)}) - h(X^{(1)}|K = K_i), & \mathcal{A} = \mathbb{R}. \end{cases}$$

Note that $H(\cdot)$ and $h(\cdot)$ represent the entropy and differential entropy functions respectively. Given $\mathcal{K}, \mathcal{G}$, we define bounds R, C_1, C_2 on the following quantities:

$$R \leq \log_2 M - \log_2 B(s, \mathcal{G}), \quad C_1 \geq p \log_2 |\mathcal{A}|, \quad C_2 \geq \max_{1 \leq i \leq M} I(K_i; X^{(1)}).$$

While C_1 is well-defined only if $\mathcal{A}$ is a finite set, C_2 is well-defined for general alphabets, like $\mathcal{A} = \mathbb{R}$. The following result holds for discrete Markov networks:

Theorem 4.2.1. *Consider an ensemble of M discrete Markov networks $\mathcal{K}$ on p random variables and the ensemble of their underlying undirected graphs $\mathcal{G}$ on p nodes. Suppose the random variables take values from some finite alphabet $\mathcal{A}$. Then we have the following lower bound on the probability of error for any learning algorithm ϕ:*

$$P_{e,s}^{(n)}(\phi) \geq 1 - 2^{-(R - nC_1)}.$$

Proof. Refer to Appendix B.1. □

We also have the following result for graph recovery in general Markov networks:

Theorem 4.2.2. *Consider an ensemble of M discrete Markov networks $\mathcal{K}$ on p random variables and the ensemble of their underlying undirected graphs $\mathcal{G}$ on p nodes.*

Suppose the random variables take values from a general alphabet $\mathcal{A}$. Then we have the following lower bound on the probability of error for any learning algorithm ϕ:

$$P_{e,s}^{(n)}(\phi) \geq 1 - \frac{4A(\mathcal{K})}{(R - nC_2)^2} - 2^{-\frac{(R-nC_2)}{2}},$$

where $A(\mathcal{K}) \triangleq \max_{1 \leq i \leq M} \operatorname{var}\left(\log_2 \frac{f\left(X^{(1)} \middle| K=K_i\right)}{f(X^{(1)})} \middle| K = K_i\right)$*, and* $f(\cdot)$ *denotes p.m.f. and p.d.f. for discrete and continuous Markov networks/distributions respectively.*

Proof. Refer to Appendix B.2. □

A consequence of Theorem 4.2.1 is that the probability of error of any learning algorithm is lower bounded by $1 - 2^{-\frac{R}{2}}$, if $n < \frac{R}{2C_1}$. Therefore, the probability of error approaches one as $p \to \infty$, if R scales with p. Likewise, Theorem 4.2.2 states that the probability of error of any learning algorithm is lower bounded by $1 - \frac{8A(\mathcal{K})}{RC_2} - 2^{-\frac{R}{4}}$, if $n < \frac{R}{2C_1}$; the probability of error approaches one as $p \to \infty$, only if R scales with p and ensemble $\mathcal{K}$ satisfies $A(\mathcal{K}) = o(RC_2)$. In this sense, Theorem 4.2.2 seems to be weaker than Theorem 4.2.1 as $(1 - P_e^{(n)})$ is upper bounded by polynomial decaying term (in R) in Theorem 4.2.2 for $n < \frac{R}{2C_2}$, whereas it is upper bounded by an exponential decaying term (in R) in Theorem 4.2.1 for $n < \frac{R}{2C_1}$. Nevertheless, Theorem 4.2.2 yields a more general result since it is applicable to ensembles of Markov networks with any alphabet set, be it finite or infinite.

Next, we make use of these results to derive high-dimensional results for discrete and Gaussian Markov networks based on graphs in $\mathcal{G}_{p,d}$, the set of undirected graphs on p nodes where degree of each node is bounded above by d. The main idea is to choose $\mathcal{G}$ and $\mathcal{K}$ in an intelligent way ($\mathcal{G}$ should be a subset of the

graph ensemble), so that reasonable values for the bounds R, C_1, C_2 can be derived. Thereafter, Theorem 4.2.1 and Theorem 4.2.2 can be used to obtain the necessary conditions/bounds on sample complexity associated with these ensembles.

4.2.2 Discrete Markov Networks on Degree-bounded Graphs

We consider any family of discrete Markov networks (for example, Ising model) with finite alphabet $\mathcal{A}$ and based on graphs in $\mathcal{G}_{p,d}$. For each $G \in \mathcal{G}_{p,d}$, we choose a Markov network from the family whose underlying graph is G. We refer to this ensemble of discrete Markov networks as $\mathcal{K}^D_{p,d}$, with $|\mathcal{K}^D_{p,d}| = |\mathcal{G}_{p,d}|$. We choose $\mathcal{G} = \mathcal{G}_{p,d}$, $\mathcal{K} = \mathcal{K}^D_{p,d}$ and $C_1 = p \log_2 |\mathcal{A}|$. Also, the following lemma holds:

Lemma 4.2.3. *Given $\mathcal{G} = \mathcal{G}_{p,d}$, $d \leq \frac{p-1}{2}$, $0 < s \leq \frac{p(p-1)}{4}$, the following bounds hold:*

$$\log_2 |\mathcal{G}| \geq \frac{pd}{4} \log_2 \left(\frac{p}{8d}\right), \quad B(s, \mathcal{G}) < s \binom{\frac{p^2}{2}}{s}.$$

Proof. Refer to Appendix B.3. □

This lemma gives the value of bound R for $\mathcal{G}_{p,d}$. Then the following result holds:

Theorem 4.2.4. *Suppose K is chosen uniformly from $\mathcal{K}^D_{p,d}$. If for some $\alpha < 1$, $d = o(p^\alpha)$, $2 \leq s < (1-\alpha)\frac{pd}{16}$, and the number of i.i.d. samples, generated from K, satisfies*

$$n < \frac{1}{2 \log_2 |\mathcal{A}|} \left(\left(\frac{d}{4} - \frac{2s}{p} \right) \log_2 p - \frac{d}{4} \log_2 8d \right) = \Omega \left(\left((1-\alpha)d - \frac{8s}{p} \right) \log_2 p \right),$$

then for any learning algorithm ϕ, we observe that $P^{(n)}_{e,s}(\phi) \to 1$ as $p \to \infty$.

Proof. Refer to Appendix B.4. □

The use of Theorem 4.2.4 for small values of s (like $s = 2$) gives a sample complexity requirement of $n = \Omega(d \log_2 p)$, that is consistent with the sample complexity bound for ensuring exact recovery of Markov graph ($s = 0$) [59]. The theorem also leads to an interesting observation – we require $s = \Theta(pd)$, the order of the maximum number of edges of a graph in $\mathcal{G}_{p,d}$, to reduce the scaling of n, as compared to its optimal scaling of $n = \Omega(d \log_2 p)$ for $s = 0$. In other words, providing additional scope for improvement in form of permitted non-zero distortion does not help a learning algorithm in reducing the scaling of sample complexity for discrete Markov networks based on graphs in $\mathcal{G}_{p,d}$, unless the permitted edit-distance based distortion is comparable to the maximum number of edges.

4.2.3 Gaussian Markov Networks on Degree-bounded Graphs

Next, we consider the family of Gaussian Markov networks based on graphs in $\mathcal{G}_{p,d}$. Here, we construct $\mathcal{G}$ and $\mathcal{K}$ as follows. Without any loss of generality, we assume that p is even. We choose d perfect matchings on p nodes, each perfect matching chosen uniformly at random, and form a multi-graph resulting from the union of the edges in the matchings. We refer to the set of all such multi-graphs as $\mathcal{H}_{p,d}$. The uniform distribution over the set of perfect matchings defines a probability distribution over $\mathcal{H}_{p,d}$. We have the following result for this distribution:

Lemma 4.2.5 ([80])**.** *Consider a multi-graph H formed from the union of d random perfect matchings on p nodes, that are chosen according to a uniform distribution. Suppose the eigenvalues of the weighted adjacency matrix of H, denoted by $\mathbf{A}$, are $d = \lambda_1(\mathbf{A}) \geq \lambda_2(\mathbf{A}) \geq \cdots \geq \lambda_p(\mathbf{A})$. If $\rho(\mathbf{A}) \triangleq \max_{2 \leq i \leq p} |\lambda_i(\mathbf{A})|$, then we have*

$P(\rho(\mathbf{A}) < 3d^{1/2}) \geq 1 - cp^{-\tau}$, *where* c *is positive constant and* $\tau = \left\lceil \frac{(d-1)^{1/2}+1}{2} \right\rceil - 1$.

We eliminate those multi-graphs from $\mathcal{H}_{p,d}$ whose weighted adjacency matrices $\mathbf{A}$ satisfy $\rho(\mathbf{A}) \geq 3d^{1/2}$ and get a reduced subset $\mathcal{H}'_{p,d}$. By Lemma 4.2.5, $\mathcal{H}_{p,d} \backslash \mathcal{H}'_{p,d}$ forms a small fraction of $\mathcal{H}_{p,d}$. We fix constants $\lambda \in \left(0, \frac{1}{4d^{1/2}}\right)$, $\delta > 0$ and define $\mu \triangleq \frac{\delta}{\lambda^{-1} - 4d^{1/2}}$. For every multi-graph $H \in \mathcal{H}'_{p,d}$, we generate a $p \times p$ matrix $\Theta = (4d^{1/2}\mu + \delta)\mathbf{I}_p + \mu\mathbf{A}$, where $\mathbf{I}_p$ is the $p \times p$ identity matrix and $\mathbf{A}$ is the weighted adjacency matrix of H. We refer to the resulting set of these matrices as $\mathcal{T}_{p,d}$. Note that every matrix in $\mathcal{T}_{p,d}$ is symmetric and positive definite, since the minimum eigenvalue of $\Theta \in \mathcal{T}_{p,d}$ is at least $4d^{1/2}\mu + \delta - \rho(\mathbf{A})\mu > d^{1/2}\mu + \delta > 0$, which implies all eigenvalues of Θ are positive. Also, the choice of μ ensures $\lambda^*(\Theta) = \lambda$ for $\Theta \in \mathcal{T}_{p,d}$. Thus, the matrices of $\mathcal{T}_{p,d}$ can be inverse covariance matrices of Gaussian Markov networks. By construction, the underlying graphs of these Markov networks are in $\mathcal{G}_{p,d}$. We denote this ensemble of Gaussian Markov networks by $\mathcal{K}^G_{p,d}$ and its graphical ensemble by $\mathcal{G}'_{p,d} \subseteq \mathcal{G}_{p,d}$. We choose $\mathcal{G} = \mathcal{G}'_{p,d}$ and $\mathcal{K} = \mathcal{K}^G_{p,d}$. Then the following lemmas give the values of bounds R and C_2 for these ensemble choices:

Lemma 4.2.6. *Given* $\mathcal{G} = \mathcal{G}'_{p,d}$, $0 < s \leq \frac{p(p-1)}{4}$, *large* p, *the following bounds hold:*

$$\log_2 |\mathcal{G}| \geq \frac{pd}{2} \log_2 \left(\frac{p}{4d^2}\right) - 1, \quad B(s, \mathcal{G}) < s \binom{\frac{p^2}{2}}{s}.$$

Proof. Refer to Appendix B.5 □

Lemma 4.2.7. *If* K *is chosen uniformly at random from* $\mathcal{K} = \mathcal{K}^G_{p,d} = \{K_1, \ldots, K_M\}$,

$$\max_{1 \leq i \leq M} I(K_i; X^{(1)}) \leq \frac{p}{2} \log_2 \left(1 + \frac{4d^{1/2}}{\lambda^{-1} - 4d^{1/2}}\right).$$

Proof. Refer to Appendix B.6. □

The use of these lemmas allows us to derive strong converse result as stated below:

Theorem 4.2.8. *Suppose K is chosen uniformly from $\mathcal{K}^G_{p,d}$. If for some $\alpha < \frac{1}{2}$, $d = o(p^\alpha)$, $2 \leq s < (1-2\alpha)\frac{pd}{8}$ and number of i.i.d. samples, generated from K, satisfies*

$$n < \frac{\left(d - \frac{4s}{p}\right)\log_2 p - 2d\log_2 2d}{2\log_2\left(1 + \frac{4d^{1/2}}{\lambda^{-1} - 4d^{1/2}}\right)} = \Omega\left(\frac{\left((1-2\alpha)d - \frac{4s}{p}\right)\log_2 p}{\log_2\left(1 + \frac{4d^{1/2}}{\lambda^{-1} - 4d^{1/2}}\right)}\right),$$

then for any learning algorithm ϕ, we observe that $P^{(n)}_{e,s}(\phi) \to 1$ as $p \to \infty$.

Proof. Refer to Appendix B.7. □

The use of Theorem 4.2.8 gives a sample complexity requirement of $n = \Omega\left(\left(d - \frac{s}{p}\right)\log_2 p\right)$ for $\lambda = \Theta\left(\frac{1}{d^{1/2}}\right)$, and $n = \Omega\left(d^{1/2}\left(d - \frac{s}{p}\right)\log_2 p\right)$ for $\lambda = \Theta\left(\frac{1}{d}\right)$. For small values of s, these reduce to $n = \Omega(d\log_2 p)$ for $\lambda = \Theta\left(\frac{1}{d^{1/2}}\right)$, and $n = \Omega(d^{3/2}\log_2 p)$ for $\lambda = \Theta\left(\frac{1}{d}\right)$. While the sample complexity bound for ensuring exact recovery of Markov graph matches for the first case, it is off by a factor of $d^{1/2}$ for the second case [64] (the sample complexity of exact recovery with $\lambda = \Theta\left(\frac{1}{d}\right)$ is $n = \Omega(d^2\log_2 p)$). Analogous to the case of discrete Markov networks with graphs in $\mathcal{G}_{p,d}$, the theorem also indicates the following fact – we require $s = \Theta(pd)$ to reduce the scaling of n, as compared to its optimal scalings for small values of s. Thus, the scaling of sample complexity cannot be reduced, unless the given edit-distance based distortion is comparable to the maximum number of edges.

Summary: We develop a rate-distortion framework for the problem of learning Markov graphs, where we characterize lower bounds on sample complexity using edit-distance based distortion criterion. Our results suggest that for both discrete and Gaussian Markov networks based on ensemble of degree-bounded graphs, substantial gains in sample complexity may not be possible unless the distortion limit is made a constant fraction of the number of edges in the graph.

4.3 Markov Networks based on Power-Law Graphs

In this section, we present results related to both the converse and achievability aspects of the problem of learning structure of Markov networks based on power-law graphs obtained using the configuration model. We restrict ourselves to discrete Markov networks for deriving lower bounds on sample complexity for learning algorithms targeting exact recovery of underlying graphs. Thereafter, we design an efficient algorithm for recovering the structure of power-law graph-based Ising models. We start by describing the configuration model and the way to generate power-law graphs, and examine its structural properties. Thereafter, we look into the converse and achievability aspects of the learning problem.

4.3.1 Configuration Model and Power-Law Graphs

We consider a degree sequence $\mathbf{d} = (d_1, d_2, \ldots, d_p)$ for an undirected graph on p nodes, and a set of configuration points $W = \{1, 2, \ldots, 2m\}$, where $2m = \sum_{i=1}^{n} d_i$. We define $W_k = \{\sum_{i=1}^{k-1} d_i + 1, \sum_{i=k}^{k-1} d_i + 2 \ldots, \sum_{i=1}^{k} d_i\}$, $k = 1, 2, \ldots, p$ (we set $d_0 = 1$). Thus, $\{W_k : 1 \leq k \leq p\}$ forms a partition of W with $|W_k| = d_k$. We

define mapping $\psi : W \to \{1, 2, \ldots, p\}$ such that $\psi(x) = k$ for $x \in W_k$. Given a (perfect) matching $\mathcal{F}$ for W (i.e., a partition of W into m pairs $\{x, y\}$), we obtain a multi-graph $G(\mathcal{F}) = (V, E)$ with $V = \{1, 2, \ldots, p\}$ and $(\psi(x), \psi(y)) \in E$ for each $\{x, y\} \in \mathcal{F}$. Therefore, choosing a matching $\mathcal{F}$ for W uniformly at random results in the generation of a multi-graph $G(\mathcal{F})$, where $i \in V$ has degree d_i. We refer to this model as the configuration model and designate the ensemble as $\mathcal{G}(\mathbf{d})$.

The number of distinct matchings $\mathcal{F}$ of the $2m$ points in W is given by $N_{2m} = \frac{(2m)!}{m!2^m}$. We call a multi-graph simple if it has no self-loops or multiple edges between nodes. An important point to note is that the number of matchings corresponding to each simple graph in $\mathcal{G}(\mathbf{d})$ is the same, i.e., simple graphs are equiprobable in the space of multi-graphs. We refer to the subset of simple graphs as $\mathcal{G}_s(\mathbf{d}) \subset \mathcal{G}(\mathbf{d})$. Furthermore, we define $d_{\min} = \min_{i \in V} d_i$, $d_{\max} = \max_{i \in V} d_i$, and

$$D_l = \sum_{i \in V} d_i^l, \quad l = 0, 1, 2, \ldots,$$

so that $D_0 = p$, $D_1 = 2m$. We assume that $d_{\max} = o(p^{\frac{1}{3}})$ and $d_{\min} = o(d_{\max})$ – under these constraints, it is known that the probability of $G(\mathcal{F})$ being simple for a uniformly chosen $\mathcal{F}$ asymptotically goes to $q_s = \exp(-\frac{\nu}{2} - \frac{\nu^2}{4})$ as $p \to \infty$, where $\nu = \frac{D_2}{D_1} - 1$ [81]. The use of this fact gives the following lower bound on $|\mathcal{G}(\mathbf{d})|$:

Lemma 4.3.1. *Given a degree sequence* $\mathbf{d}$ *and large* p*, the ensemble* $\mathcal{G}_s(\mathbf{d})$ *satisfies*

$$|\mathcal{G}_s(\mathbf{d})| \geq \frac{q_s}{2} \frac{N_{2m}}{\prod_{i \in V} d_i!} \geq \frac{q_s}{2} \prod_{i \in V} \left(\frac{m^{1/2}}{2d_i} \right)^{d_i}.$$

Proof. Refer to Appendix B.8. □

Next, we state a lemma that we use for getting tight bounds on summations:

Lemma 4.3.2 ([82]). *Given a continuous and positive function $f : \mathbb{R} \to \mathbb{R}$, we have*

$$\int_a^b f(x)dx \leq \sum_{k=a}^{b} f(k) \leq \int_a^b f(x)dx + f(b+1), \quad \text{if } f \text{ is an increasing function,}$$

$$\int_a^b f(x)dx \leq \sum_{k=a}^{b} f(k) \leq \int_a^b f(x)dx + f(a-1), \quad \text{if } f \text{ is a decreasing function,}$$

for all positive integers a, b with $a < b$, and f well-defined over $[a-1, b+1]$.

We consider the generation of simple power-law graphs using the configuration model, i.e., simple graphs whose degree sequence follow a power-law or Pareto distribution. Given $\alpha > 1$, a power-law graph with exponent α has the property that the number of nodes with degree k is proportional to $k^{-\alpha}$. For p nodes and given values of $d_{\min}, d_{\max}$, we define $\zeta(\alpha) = (\sum_{k=d_{\min}}^{d_{\max}} k^{-\alpha})^{-1}$. Then the number of nodes with degree k is approximately $p\zeta(\alpha)k^{-\alpha}$, where $d_{\min} \leq k \leq d_{\max}$. For the sake of simplicity and convenience, we assume that $p\zeta(\alpha)k^{-\alpha}$, $d_{\min} \leq k \leq d_{\max}$, are integers. Note that this gives the constraint $d_{\max} \leq (p\zeta(\alpha))^{\frac{1}{\alpha}}$, since there is at least one node with degree $d_{\max}$. Therefore, we impose a stronger restriction $d_{\max} = o(p^{\min(\frac{1}{3}, \frac{1}{\alpha})})$, and define the degree sequence $\mathbf{d}$ for power-law graphs as

$$d_j = l, \quad p\zeta(\alpha)\left(\sum_{k=d_{\min}}^{l-1} k^{-\alpha}\right) < j \leq p\zeta(\alpha)\left(\sum_{k=d_{\min}}^{l} k^{-\alpha}\right), \quad d_{\min} \leq l \leq d_{\max}.$$

We denote the ensemble of graphs generated using the configuration model and having the power-law degree sequence as defined above, by $\mathcal{G}_\alpha$, and its subset

of simple graphs by $\mathcal{G}_{s,\alpha}$. Application of Lemma 4.3.2 to function $f(x) = x^{t-\alpha}$ gives

$$S_t = \sum_{k=d_{\min}}^{d_{\max}} k^{t-\alpha} \in \begin{cases} \left(\frac{d_{\max}^{t+1-\alpha}}{2(t+1-\alpha)}, \frac{2d_{\max}^{t+1-\alpha}}{(t+1-\alpha)} \right) & , \quad t > \alpha - 1, \\ \left(\frac{d_{\min}^{t+1-\alpha}}{2(\alpha-t-1)}, \frac{2d_{\min}^{t+1-\alpha}}{(\alpha-t-1)} \right) & , \quad t < \alpha - 1, \end{cases} \tag{4.1}$$

where t is any real, and $d_{\min}$ is greater than some constant (that depends only on α). Note that the bounds on S_t allow us to compute order-wise tight bounds on $\zeta(\alpha)$ and D_l, as $\zeta(\alpha) = \frac{1}{S_0}, D_l = p\zeta(\alpha)S_l$. We also define the following quantities:

$$\bar{d} = \begin{cases} \frac{(\alpha-1)}{(2-\alpha)} d_{\max}^{2-\alpha} d_{\min}^{\alpha-1} & , \ 1 < \alpha < 2 \\ \frac{(\alpha-1)}{(\alpha-2)} d_{\min} & , \ \alpha > 2 \end{cases}, \quad \tilde{d} = \begin{cases} \frac{(2-\alpha)}{(3-\alpha)} d_{\max} & , \ 1 < \alpha < 2 \\ \frac{(\alpha-2)}{(3-\alpha)} d_{\max}^{3-\alpha} d_{\min}^{\alpha-2} & , \ 2 < \alpha < 3 \\ \frac{(\alpha-2)}{(\alpha-3)} d_{\min} & , \ \alpha > 3 \end{cases}.$$

One can check that $\bar{d}$ is close (up to some constant factor) to average degree of the power-law graph (i.e., $\frac{D_1}{p}$), and $\tilde{d}$ is close in value (up to some constant factor) to the ratio of average squared degree divided by average degree (i.e., $\frac{D_2}{D_1}$). Then the following theorem gives a lower bound on number of power-law graphs in $\mathcal{G}_{s,\alpha}$:

Lemma 4.3.3. *There exists constant $c_0 > 0$ s.t. for $d_{min} \geq c_0$ and large p, we have*

$$\log_2 |\mathcal{G}_{s,\alpha}| \geq \frac{p\bar{d}}{9} \log_2 \left(\frac{(\alpha-1)}{|\alpha-2|} p \right).$$

Proof. Refer to Appendix B.9. □

Note that we inherently assume that $d_{\min}$ is larger than some suitable constant and p is large enough in all the subsequent results concerning the graphs in $\mathcal{G}_{s,\alpha}$.

4.3.2 Structural Properties of Power-Law Graphs

Next, we examine some structural properties of power-law graphs in $\mathcal{G}_{s,\alpha}$ with $\alpha > 2$. We consider a stronger set of assumptions on the scalings of $d_{\min}, d_{\max}$:

(A1) $d_{\min} = \Theta(1)$, $d_{\max} = \Theta((\log_2 p)^{\delta_1})$ for some $\delta_1 < \frac{1}{2(3-\alpha)}$, $2 < \alpha < 3$,

(A2) $d_{\min} = \Theta(1)$, $d_{\max} = \Theta(p^{\delta_2})$ for some $\delta_2 < \min\left(\frac{1}{6}, \frac{1}{\alpha}\right)$, $\alpha > 3$.

The choice of these scalings ensure that $\nu = o((\log_2 p)^{1/2})$, so that the probability of getting a simple graph is $\exp(-o(\log_2 p))$. Thus, as long as some property holds for a uniformly generated graph from $\mathcal{G}_\alpha$ with probability $\geq 1 - p^{-\Theta(1)}$, it also holds for a uniformly selected simple graph from $\mathcal{G}_{s,\alpha}$ with probability $\geq 1 - p^{-\Theta(1)}$.

We show that power-law graphs generated using the configuration model tend to be tree-like. Given a positive integer r, we define the r-neighborhood of a node in a graph as the subgraph resulting from the set of nodes that reachable from it via at most r edges. We define $r_0 = \frac{1}{2}\frac{\log_2 p}{\log_2(64\tilde{d})}$ and assume it is an integer. Then the r_0-neighborhoods of nodes of graphs in $\mathcal{G}_{s,\alpha}$ satisfy the following property:

Lemma 4.3.4. *Given $\alpha > 2$ and assumptions (A1), (A2) hold, if a graph from $\mathcal{G}_{s,\alpha}$ is selected uniformly at random, there is at most one cycle in the r_0-neighborhood of any node with probability $\geq 1 - p^{-\Theta(1)}$ (i.e., r_0-neighborhood is almost a tree).*

Proof. Refer to Appendix B.10. □

An alternate way of restating the consequence of Lemma 4.3.4 is that there exist at most two paths, of length at most r_0, between any two nodes in a graph of $\mathcal{G}_{s,\alpha}$,

with high probability. We define $B(i,r)$ as the number of nodes in r-neighborhood of $i \in V$. Then the following result provides bounds on $B(i,r_0)$ for graphs in $\mathcal{G}_{s,\alpha}$:

Lemma 4.3.5. *Given $\alpha > 2, \epsilon > 0$, and assumptions (A1), (A2) hold, if a graph from $\mathcal{G}_{s,\alpha}$ is selected uniformly at random, then the r-neighborhoods, $1 \le r \le r_0$, satisfy*

$$P(B(i,r) < p^{\frac{1}{4}+\frac{\epsilon}{2}} d_i (64\tilde{d})^{(r-1)} \ \forall i \in V) \ge 1 - p^{-\epsilon}, \quad 2 < \alpha < 3,$$
$$P(B(i,r) < p^{\frac{1}{4}+\frac{\epsilon}{8}} d_i (64\tilde{d})^{(r-1)} \ \forall i \in V) \ge 1 - p^{-\epsilon}, \quad \alpha > 3.$$

Proof. Refer to Appendix B.11 □

4.3.3 Lower Bounds on Sample Complexity

We analyze the converse aspect of the learning problem and derive lower bounds on number of samples required for any algorithm to accurately learn the structure of a discrete Markov network with its graph in $\mathcal{G}_{s,\alpha}$. For this, we consider any family of discrete Markov networks with alphabet $\mathcal{A}$ and based on graphs in $\mathcal{G}_{s,\alpha}$. For each $G \in \mathcal{G}_{s,\alpha}$, we choose a Markov network whose underlying graph is G. We refer to this ensemble of Markov networks as $\mathcal{K}^D_{s,\alpha}$. Then the use of Theorem 4.2.1 gives the following bound on sample complexity for exact graph recovery:

Theorem 4.3.6. *Suppose a discrete Markov network is chosen uniformly from $\mathcal{K}^D_{s,\alpha}$. If the number of i.i.d. samples, generated from the discrete Markov network, satisfies*

$$n < \frac{\bar{d}}{10 \log_2 |\mathcal{A}|} \log_2 \left(\frac{(\alpha-1)}{|\alpha-2|} p \right),$$

then for any learning algorithm ϕ, we observe that $P_e^{(n)}(\phi) \to 1$ as $p \to \infty$.

Proof. Refer to Appendix B.12. □

Thus, the use of Theorem 4.3.6 gives a sample complexity requirement of $n = \Omega(d_{\max}^{2-\alpha} d_{\min}^{\alpha-1} \log_2 p)$ for $1 < \alpha < 2$, and $n = \Omega(d_{\min} \log_2 p)$ for $\alpha > 2$, for ensuring exact recovery of the Markov graph. Note that the sample complexity result for $\alpha > 3$ matches the one derived in [68] in order-wise sense, where a slightly modified version of configuration model is used and $d_{\min}$ is set as 1. In case $d_{\max}$ scales with p, Theorem 4.3.6 implies that a larger number of samples (in order-wise sense) is needed by a learning algorithm to reconstruct the underlying graph of a discrete Markov graph when α is less than 2, as compared to when it is greater than 2. Furthermore, the transition in the sample complexity requirement is sharp and observed at $\alpha = 2$. A potential reason for this phenomenon is that the fraction of high degree nodes decreases as α increases. In other words, for a fixed number of samples, it is inherently difficult to learn power-law graph-based discrete Markov networks with lower exponent values (less than 2, to be precise); moreover, this issue aggravates as the power-law exponent value goes down from 2 to 1.

4.3.4 Learning Algorithm for Ising Model

Next, we examine the achievability aspect of the learning problem and work on designing an algorithm for learning the graph structure of discrete Markov networks based on graphs in $\mathcal{G}_{s,\alpha}$. In particular, we focus on the ferromagnetic Ising model family with that has finite node weights and edge weights. As observed in Section 4.3.3, the problem of learning power-law Markov graphs tends to be challenging, in terms of sample complexity, if the power-law exponent is

less than 2, since then the average degree depends on the maximum degree, that can scale with the number of nodes. Therefore, we restrict ourselves to the regime $\alpha > 2$, and also allow assumptions (A1) and (A2) to hold. As pointed out in [68], one of the major challenges that a learning algorithm faces in the context of power-law Markov networks is to tackle the large variation in degrees of nodes. We use a learning algorithm, similar to the empirical conditional variation distance thresholding algorithm, studied in [61, 62, 83] for reconstructing Markov networks based on degree-bounded, large-girth, Watts-Strogatz and Erdös-Rényi graphs.

Given $0 < \theta_{\min} \leq \theta_{\max}$, we consider the family of ferromagnetic Ising models on p random variables, with finite node potentials and edge potentials lying in $[\theta_{\min}, \theta_{\max}]$. We choose any Ising model from this family with p.m.f. $f(\cdot)$ and $G = (V, E) \in \mathcal{G}_{s,a}$ as its underlying graph. We define the following mapping, that can be thought of as some measure of distance between random variables:

$$\rho(i,j) = \min_{U \subseteq V : |U| \leq 2} \max_{x_i, x_U} |f(x_i | x_j = 1, x_U) - f(x_i | x_j = -1, x_U)|, \quad i,j \in V.$$

As described in Section 4.1.1, we are provided with n i.i.d. samples from $f(\cdot)$ – we rename this collection as $\mathbf{x}^n = (x^{(1)}, x^{(2)}, \ldots, x^{(n)})$, as the samples are deterministic when used by the learning algorithm. We also define the empirical p.m.f. $\hat{f}(\cdot)$ as

$$\hat{f}(x) = \hat{f}(x_1, x_2, \ldots, x_p) = \frac{1}{n} \sum_{l=1}^{n} \mathbb{I}(x_i = x_i^{(l)}, \, 1 \leq i \leq p),$$

where $\mathbb{I}(\cdot)$ is the indicator function. Note that $\hat{f}(\cdot)$ can be used to compute/define empirical marginal and conditional p.m.f.'s related to $f(\cdot)$. This allows us to define

$$\hat{\rho}(i,j) = \min_{U \subseteq V : |U| \leq 2} \max_{x_i, x_U} |\hat{f}(x_i | x_j = 1, x_U) - \hat{f}(x_i | x_j = -1, x_U)|, \quad i,j \in V,$$

Algorithm 2 Learning algorithm ϕ^* to obtain $\hat{G}$ from $\mathbf{x}^n$

Initialize: $V = \{1, 2, \ldots, p\}$, $\hat{E} = \emptyset$
  for all $i, j \in V$ **do**
    if $\hat{\rho}(i,j) > \zeta_{n,p}$ **then**
      $\hat{E} \leftarrow \hat{E} \cup \{(i,j)\}$
    end if
  end for
Output: $\hat{G} = (V, \hat{E})$

which serves as the empirical counterpart (or approximation) of $\rho(\cdot,\cdot)$.

The learning algorithm ϕ^* for obtaining $\hat{G}$, the estimate of G, is tabulated in form of Algorithm 2. The value of threshold $\zeta_{n,p}$ influences the sample complexity required for exact recovery, and is stated in the subsequent subsection. The motivation behind this learning algorithm comes from the observation that $\rho(i,j)$ tends to be larger when edge exists between i and j than when the edge does not exist. In other words, the influence of X_j on X_i is more when i, j are neighbors versus when they are non-neighbors. Also, $\rho(\cdot,\cdot)$ and $\hat{\rho}(\cdot,\cdot)$ are close in value if the number of samples n is large – we corroborate all of these facts below.

Non-neighboring nodes: If $i, j \in V$ are non-neighboring nodes in G, then by Lemma 4.3.4 there exists at most two short paths of length at most r_0 connecting i to j with high probability. We define the l-separator set for two nodes as the minimum number of nodes that need to be removed for eliminating all paths of length $\leq l$ between them. This means the r_0-separator set size for i, j is at most two with high probability. We use the following strong correlation decay result, related to separator sets, for ferromagnetic Ising model to show that $\rho(i,j)$ is small:

Lemma 4.3.7 ([83]). *Consider a ferromagnetic Ising model based on graph $G = (V, E)$, with p random variables and p.m.f. $f(x_1, x_2, \ldots, x_p)$. Given non-neighboring nodes $i, j \in V$ and positive integer l, let $U \subseteq V$ be the l-separator set for i, j and let $A(i, l)$ be the number of nodes that are at a distance of l from i in $G_{i,SAW}$, the self-avoiding walk (SAW) tree of G with i as root. Then for $x_i, x_j \in \{-1, 1\}$, $x_U \in \{-1, 1\}^{|U|}$,*

$$|f(x_i|x_j = 1, x_U) - f(x_i|x_j = -1, x_U)| \leq A(i, l)(\tanh \theta_{max})^l.$$

The SAW tree rooted at node i of a graph is the tree with i as its root and generated via self-avoiding walks starting at i in the graph. In the context of Ising model, a node that closes a cycle is made the leaf of the tree (this generates potentially multiple copies of a node) and is assigned value of 1, if the node label ending the cycle is larger than the node label starting the cycle, otherwise it is assigned value of -1. The advantage of using a SAW tree is that it transforms a non-tree Ising model into a tree-structured Ising model, whose analysis is more tractable. A description about SAW trees and their properties can be found in [61].

The application of Lemma 4.3.7 to our Ising model learning setup yields:

Theorem 4.3.8. *Consider a ferromagnetic Ising model based on a uniformly selected graph from $\mathcal{G}_{s,\alpha}$, where $\alpha > 2$, assumptions (A1), (A2) hold, and $\tanh \theta_{max} < \frac{1}{(64\tilde{d})^2}$. Then $P(\rho(i, j) = o(p^{-\kappa})) \geq 1 - p^{-\Theta(1)}$, for some $\kappa > 0$ and non-neighbors i, j.*

Proof. Refer to Appendix B.13. □

Neighboring nodes: If $i, j \in V$ are neighbors, the following result holds:

Theorem 4.3.9. *Consider a ferromagnetic Ising model based on a uniformly selected graph from $\mathcal{G}_{s,\alpha}$, where $\alpha > 2$. Then $\rho(i,j) \geq \frac{1}{16}(1 - e^{-4\theta_{min}})$ for neighbors i, j.*

Proof. Refer to Appendix B.14. □

4.3.5 Performance Analysis

We have the following concentration result that relates ρ to its estimate $\hat{\rho}$:

Lemma 4.3.10. *For n i.i.d. samples, we get the following concentration inequality:*

$$P(|\hat{\rho}(i,j) - \rho(i,j)| \leq \gamma \ \forall i,j \in V) \geq 1 - 192p^4 \exp\left(-\frac{n\gamma^2 f_{\min}^2}{2(\gamma+4)^2}\right),$$

where $\gamma > 0$ can be any arbitrary real value and $f_{\min} = \min_{x_U : |U| \leq 2} f(x_U)$.

Proof. Refer to Appendix B.15. □

We choose $\zeta_{n,p} = \frac{1}{32}(1 - e^{-4\theta_{\min}})$ for ϕ^*. We also assume $\theta_{\min} = \Omega\left(\frac{1}{(\log p)^r}\right)$ for some constant $r > 0$. Then we have the following performance-related result:

Theorem 4.3.11. *Consider a ferromagnetic Ising model based on a uniformly selected graph from $\mathcal{G}_{s,\alpha}$, where $\alpha > 2$, assumptions (A1), (A2) hold, $\tanh\theta_{max} < \frac{1}{(64\tilde{d})^2}$ and $\theta_{min} = \Omega\left(\frac{1}{(\log p)^r}\right)$ for some constant $r > 0$. Suppose we choose $\zeta_{n,p} = \frac{1}{32}(1 - e^{-4\theta_{min}})$ for ϕ^*. Given p is large enough and the number of i.i.d. samples n satisfies*

$$n > \frac{2^{18}}{(1 - e^{-4\theta_{min}})^2 f_{\min}^2} \log_2\left(\frac{p}{3}\right)$$

then ϕ^* recovers the correct underlying graph with probability $\geq 1 - p^{-\Theta(1)}$. Furthermore, the computational complexity for running the learning algorithm is $O(p^4)$.

Proof. Refer to Appendix B.16. □

Results of Theorem 4.3.11: Since we have the restriction that $\tanh\theta_{\min} \leq \tanh\theta_{\max} \leq \frac{1}{(64\tilde{d})^2} \leq \frac{1}{2^{12}}$, we can approximate $1 - e^{-4\theta_{\min}} \approx 4\theta_{\min}$. Also, it can be shown that $f_{\min}$ is greater than some constant, along the lines of [83] (the result in [83] is demonstrated for Erdos-Renyi graphs, but the same proof technique can be used to prove that $f_{\min}$ is bounded in our case). Thus, it is sufficient to have a sample complexity of $\Omega(\theta_{\min}^{-2}\log_2 p)$ for learning algorithm ϕ^* to recover the correct graph. For the scaling $\theta_{\min} = \Theta\left(\frac{1}{\tilde{d}^2}\right)$, we get the sample complexity requirement of $n = \Omega(\tilde{d}^4\log_2 p)$ – this transforms to $\theta_{\min} = \Theta\left(\frac{1}{(\log p)^{(3-\alpha)\delta_1}}\right)$ with $n = \Theta((\log_2 p)^3)$ for $2 < \alpha < 3$, and $\theta_{\min} = \Theta\left(\frac{1}{d_{\min}^2}\right)$ with $n = \Theta(d_{\min}^4\log_2 p)$ for $\alpha > 3$. Keeping in mind that $n = \Omega(d_{\min}\log_2 p)$ is the information-theoretic lower bound on sample complexity for $\alpha > 2$ to ensure accurate recovery, one can note that the constraints are more restrictive and the sample complexity result is worse for the case $2 < \alpha < 3$ than for the case $\alpha > 3$. However, since assumption (A2) makes $d_{\min}$ a constant, we can conclude the the sample complexity associated with the converse and achievability aspects match in an order-wise sense for $\alpha > 3$. A probable reason for the relative poor performance of the learning algorithm for $2 < \alpha < 3$ could be the structural nature of power-law graphs in that regime – they generally have a big core with many high degree nodes residing in it [69]. So, there is scope for designing learning algorithms with better performance (in both sample and computational complexity) in this regime of power-law exponent.

Comparison with previous results: The statistical guarantees provided by some well-known algorithms in the context of learning power-law graphical

models are examined in [68] – two generative models of power-law graphs are considered, the configuration model and Chung-Lu model [69]. The ℓ_1-regularization based learning algorithm [84] needs a sample complexity of $n = \Omega(d_{\max}^3 \log_2 p)$ for both configuration and Chung-Lu power-law graphs (the average degree is assumed to be $\Theta(1)$). The greedy algorithm, described in [85], performs slightly better and guarantees accurate recovery with $n = \Omega(d_{\max}^2 \log_2 p)$ samples. The performance analysis of the conditional variation distance thresholding estimator [61], the motivation behind our learning algorithm, exhibits a trade-off in the context of learning Chung-Lu power-law graph-based Ising model – restricting the algorithm to run in polynomial time shoots up the sample complexity requirement to $\Omega(\text{poly}(p) \log_2 p)$. In contrast, by performing a careful analysis, we show that our learning algorithm performs reasonably well in the regime $\alpha > 3$. On the other hand, there is a additional $(\log_2 p)^3$ factor in the sample complexity for the regime $2 < \alpha < 3$ when $d_{\max}$ is restricted to have $\Theta(\text{poly}(\log_2 p))$ scaling.

Summary: We study the problem of learning discrete Markov networks based on power-law graphs generated by configuration model and show that the learning problem is inherently difficult in terms of sample complexity when the power-law exponent is less than 2. Thereafter, we design a algorithm for learning the power-law structure of ferromagnetic Ising model. The algorithm proves to be order-optimal when the minimum degree is constant for power-law exponent exceeding 3; on the other hand, the sample complexity is sub-optimal when the power-law exponent lies between 2 and 3 and maximum degree is restricted to have a poly-log scaling nature (this is a limitation of the generative model).

Chapter 5

Conclusion

This dissertation seeks to analyze structured high-dimensional problems using information-theoretic tools and techniques. We address two problems that have an inherent structured high-dimensional flavor to them – the problems of communication over networks that employ linear network coding and learning Markov networks from observed samples generated from underlying probability distributions. An information-theoretic analysis of these problems gives intuition about good coding architectures as well as the limitations of transmission/learning. For the problem of communication over networks, we design linear network coding schemes based on interference alignment techniques that guarantee good throughput in networks with multiple sources and destinations communicating with each other. For the problem of learning Markov networks, we provide strong lower bounds on probability of error of any learning algorithm in terms of the number of available samples and parameters of the family of Markov networks based on degree-bounded graphs and power-law graphs. We also examine the achievability problem for Markov graphs generated by the configuration power-law graph model. Thus, the use of information-theoretic methods can prove to be useful in understanding issues as well as providing solutions to potentially difficult problems/issues in the structured high-dimensional framework.

Appendices

Appendix A

Proofs for Chapter 3

A.1 Proof of Theorem 3.2.1

Using relations (3.4)–(3.6), we get $\mathbf{V}_2 = \mathbf{M}_{31}\mathbf{M}_{32}^{-1}\mathbf{V}_1, \mathbf{V}_3 = \mathbf{M}_{21}\mathbf{M}_{23}^{-1}\mathbf{T}\mathbf{V}_1$. $\mathbf{V}_1$ is full rank since it has the structure of Vandermonde matrix; this implies $\dim(\mathcal{U}_i) = v$ for all i. We also have $\dim(\mathcal{W}_i) \leq v+1$ for all i, because of the following relations:

$$\text{rank}([\mathbf{M}_{12}\mathbf{V}_2 \ \mathbf{M}_{13}\mathbf{V}_3]) = \text{rank}(\mathbf{M}_{12}\mathbf{M}_{31}\mathbf{M}_{32}^{-1}[\mathbf{V}_1 \ \mathbf{V}_1]) = \text{rank}([\mathbf{V}_1 \ \mathbf{V}_1]) = v,$$

$$\text{rank}([\mathbf{M}_{21}\mathbf{V}_1 \ \mathbf{M}_{23}\mathbf{V}_3]) = \text{rank}(\mathbf{M}_{21}[\mathbf{V}_1 \ \mathbf{T}\mathbf{V}_1]) = \text{rank}([\mathbf{V}_1 \ \mathbf{T}\mathbf{V}_1]) = v + 1,$$

$$\text{rank}([\mathbf{M}_{31}\mathbf{V}_1 \ \mathbf{M}_{32}\mathbf{V}_2]) = \text{rank}(\mathbf{M}_{31}[\mathbf{V}_1 \ \mathbf{V}_1]\mathbf{A}) = \text{rank}([\mathbf{V}_1 \ \mathbf{V}_1]) = v.$$

Note that $\text{rank}([\mathbf{M}_{11}\mathbf{V}_1 \ \mathbf{M}_{12}\mathbf{V}_2]) = \text{rank}([\mathbf{P}_1\mathbf{V}_1 \ \mathbf{V}_1])$, where $\mathbf{P}_1 = \mathbf{M}_{11}\mathbf{M}_{32}\mathbf{M}_{12}^{-1}\mathbf{M}_{31}^{-1}$ is a $(2v+1) \times (2v+1)$ diagonal matrix with $p_1(\underline{\xi}^{(k)})$ as its (k,k)th entry. Since $p_1(\underline{\xi}) \notin \mathcal{S}_v$, there does not exist linear combinations of columns of $\mathbf{P}_1\mathbf{V}_1$ and $\mathbf{V}_1$ that are equal. In other words, $\text{rank}([\mathbf{P}_1\mathbf{V}_1 \ \mathbf{V}_1]) = 2v$, i.e., $[\mathbf{M}_{11}\mathbf{V}_1 \ \mathbf{M}_{12}\mathbf{V}_2]$ is full rank. Likewise, we can show that $[\mathbf{M}_{11}\mathbf{V}_1 \ \mathbf{M}_{13}\mathbf{V}_3]$ is full rank, which implies $\dim(\mathcal{U}_1 \cap \mathcal{W}_1) = 0$. This way it is possible to show that $\dim(\mathcal{U}_i \cap \mathcal{W}_i) = 0$, $i = 2, 3$, as well. Thus, the relations in (3.3) are satisfied over $\mathbb{F}_q[\underline{\delta}]$. Therefore, it remains to show that there exists an assignment of $\underline{\delta}$, say $\underline{\delta}_0$, for which the relations still hold. Since

any v rows of $\mathbf{V}_1$ are also the rows of a Vandermonde matrix, it is full rank only if

$$a(\underline{\delta}) = \prod_{i \neq j} (t(\underline{\xi}^i) - t(\underline{\xi}^{(j)}))$$

is non-zero-valued at $\underline{\delta} = \underline{\delta}_0$. Also, $[\mathbf{M}_{ii}\mathbf{V}_i \ \mathbf{M}_{ij}\mathbf{V}_j]$, $i \neq j$ is full rank, if the determinant of the first $2v$ rows, denoted by polynomial $r_{ij}(\underline{\delta})$, is non-zero at $\underline{\delta} = \underline{\delta}_0$. Therefore, we require the following polynomial gives a non-zero element at $\underline{\delta} = \underline{\delta}_0$:

$$f(\underline{\delta}) = a(\underline{\delta}) \prod_k \prod_{i,j} m_{ij}(\underline{\xi}^{(k)}) \prod_{i \neq j} r_{ij}(\underline{\delta}).$$

The existence of $\underline{\delta}_0$, such that $f(\underline{\delta}_0) \neq 0$, is guaranteed by the application of Schwartz-Zippel Lemma [29], for large enough $\mathbb{F}_q$. Hence, the relations in (3.3) are satisfied for some assignment and each source can transmit at rate of $\frac{v}{2v+1}$.

A.2 Proof of Theorem 3.2.3

One can check that the choice of the precoding matrices in (3.7) satisfies the alignment constraints, i.e., $\mathbf{M}_{ij}\mathbf{V}_j = \mathbf{M}_{ik}\mathbf{V}_k$ for all $i \neq j,k$. Also, the constraint $\dim(\mathcal{U}_i \cap \mathcal{W}_i) = 0$ reduces to $p_i(\underline{\xi}^{(1)})/p_i(\underline{\xi}^{(2)})$ not being equal to some constant in $\mathbb{F}_q$, $i = 1,2,3$. These facts hold good unless at least one of $p_i(\underline{\xi})$'s is identically a constant in $\mathbb{F}_q$. Analogous to the proof of Theorem 3.2.1, the existence of a feasible PBNA scheme reduces to finding an assignment of variables that makes a non-trivial polynomial non-zero in $\mathbb{F}_q$. The existence of such an assignment of variables is guaranteed by Schwartz-Zippel Lemma for large enough $\mathbb{F}_q$, and this allows each source to transmit at rate of $\frac{1}{2}$ via coding across two successive time-slots.

A.3 Proof of Theorem 3.2.4

Consider an example where we assume that $m_{12}(\underline{\xi}) \equiv 0$ and rest of the transfer functions are non-zero polynomials. Then we choose the precoding matrices as $\mathbf{V}_1 = [1 \quad 1]^t$, $\mathbf{V}_2 = \mathbf{M}_{31}\mathbf{M}_{32}^{-1}[1 \quad 1]^t$, $\mathbf{V}_3 = \mathbf{M}_{21}\mathbf{M}_{23}^{-1}[1 \quad 1]^t$. One can verify that rate of $\frac{1}{2}$ per source is achievable here if $p_2(\underline{\xi})$ and $p_3(\underline{\xi})$ are non-constant polynomials. Likewise, other cases can be handled using similar arguments.

A.4 Proof of Theorem 3.3.1

We prove this achievability result by setting $n = (L+1)$, $a = b = 1$, and designing precoding matrices $\mathbf{V}_i$, $i = 1, 2, \ldots, K$, that satisfy the relations in (3.3). Note that since $\mathcal{H}$ has no cycles, it is either a tree or a collection of disjoint trees (i.e., a forest graph). We assume there are $c \geq 1$ disjoint trees and denote them by $\mathcal{T}_l = (\mathcal{X}_l, \mathcal{Y}_l, \mathcal{F}_l)$, $l = 1, 2, \ldots, c$. Thus, $\{\mathcal{X}_l : l = 1, 2, \ldots, c\}$, $\{\mathcal{Y}_l : l = 1, 2, \ldots, c\}$, $\{\mathcal{F}_l : l = 1, 2, \ldots, c\}$ are partitions of $\mathcal{X}, \mathcal{Y}, \mathcal{F}$ respectively, and $\mathcal{H} = \cup_{l=1}^{c} \mathcal{T}_l$. Note that if $\mathcal{H}$ is a single spanning tree graph, then we have $c = 1$ and $\mathcal{H} = \mathcal{T}_1$.

We handle the disjoint trees separately, i.e., precoding vectors for sources in $\mathcal{X}_l$ are designed independently of those for sources in $\mathcal{X}_k$, $k \neq l$. Given $l \in \{1, 2, \ldots, c\}$, we choose any $S_{a_l} \in \mathcal{X}_l$ as the tree root. Next, we define $\mathcal{L}_0^{(l)} = \{S_{a_l}\}$, $\mathcal{L}_1^{(l)}$ as the set of neighbor nodes of S_{a_l} in $\mathcal{T}_l$, and $\mathcal{L}_{k+1}^{(l)}$ as the set of neighbors of nodes in $\mathcal{L}_k^{(l)}$ for $k \geq 1$ (these sets are levels of the BFS tree rooted at S_{a_l}). Since $\mathcal{T}_l$ is a bipartite graph, $\mathcal{L}_{2k+1}^{(l)} \subseteq \mathcal{Y}_l$ and $\mathcal{L}_{2k}^{(l)} \subseteq \mathcal{X}_l$ for $k \geq 0$. Thereafter, we associate

a $(L+1) \times (L+1)$ matrix $\mathbf{H}_{ij}$ with $(S_j, \mathcal{W}_i) \in \mathcal{F}_l$, that is related to $\mathbf{M}_{ij}$ as follows:

$$\mathbf{H}_{ij} := \begin{cases} \mathbf{M}_{ij}, & S_j \in \mathcal{L}_{2k}^{(l)},\ \mathcal{W}_i \in \mathcal{L}_{2k+1}^{(l)},\ k \geq 0, \\ \mathbf{M}_{ij}^{-1}, & S_j \in \mathcal{L}_{2k+2}^{(l)},\ \mathcal{W}_i \in \mathcal{L}_{2k+1}^{(l)},\ k \geq 0. \end{cases}$$

Therefore, by construction, $\mathbf{H}_{ij}$ is a diagonal matrix with $h_{ij}(\underline{\xi}^{(t)})$ as its (t,t)th entry, such that $h_{ij}(\underline{\xi}) \equiv m_{ij}(\underline{\xi})$ for $S_j \in \mathcal{L}_{2k}^{(l)}$, $\mathcal{W}_i \in \mathcal{L}_{2k+1}^{(l)}$ with $(S_j, \mathcal{W}_i) \in \mathcal{F}_l$, and $h_{ij}(\underline{\xi}) \equiv (m_{ij}(\underline{\xi}))^{-1}$ for $S_j \in \mathcal{L}_{2k+2}^{(l)}, \mathcal{W}_i \in \mathcal{L}_{2k+1}^{(l)}$ with $(S_j, \mathcal{W}_i) \in \mathcal{F}_l$, for $k \geq 0$.

We set $\mathbf{V}_{a_l} = [\theta_l^{(1)}\ \theta_l^{(2)}\ \cdots\ \theta_l^{(L+1)}]^T$, where $\theta_l^{(k)}$, $k = 1, 2, \ldots, L+1$, are variables that take values from $\mathbb{F}_q$. Since $\mathcal{H}$ has the structure of a collection of trees, there exists a unique path between S_u and S_v, say $\mathcal{P}_{uv}$, if they are connected to each other via edge(s). Then, given $i \neq a_l$ and $S_i \in \mathcal{X}_l$, we assign $\mathbf{V}_i = \mathbf{T}_i \mathbf{V}_{a_l}$, where

$$\mathbf{T}_i = \prod_{(u,v):(S_v, \mathcal{W}_u) \in \mathcal{P}_{i,a_l}} \mathbf{H}_{uv}.$$

Note that $\mathbf{T}_i$ is $(L+1) \times (L+1)$ diagonal matrix with (k,k)th entry as $t_i(\underline{\xi}^{(k)})$, where

$$t_i(\underline{\xi}) \equiv \prod_{(u,v):(S_v, \mathcal{W}_u) \in \mathcal{P}_{i,a_l}} h_{uv}(\underline{\xi}).$$

This choice of precoding vectors ensures $\mathbf{M}_{ij}\mathbf{V}_j = \mathbf{M}_{ik}\mathbf{V}_k$ if $S_j, S_k \in \mathcal{B}_i$ and $|\mathcal{B}_i| \geq 2$. Therefore, the constraint $\dim(\mathcal{W}_i) = 1$ is satisfied for all i (this is trivially satisfied if $|\mathcal{B}_i| = 1$). Also, the constraints $\dim(\mathcal{W}_i) = 1$, $\dim(\mathcal{U}_i \cap \mathcal{W}_i) = 0$ are satisfied if and only if the set of vectors $\{\mathbf{M}_{ij}\mathbf{V}_j : S_j \in \mathcal{A}_i\}$ and $\mathbf{M}_{ik}\mathbf{V}_k$, for any $k \in \mathcal{B}_i$, form a full rank $(L+1) \times (L+1)$ matrix, say $\mathbf{R}_{ik}$. Note that the entries of $\mathbf{R}_{ik}$ are rational functions based on multivariate polynomials in $\mathbb{F}_q[\underline{\delta}]$, where $\underline{\delta}$ comprises of variables in $\underline{\xi}^{(k)}$ and the new variables $\theta_l^{(k)}$, $k = 1, \ldots, L+1$, $l = 1, \ldots, c$.

The fact whether $\mathbf{R}_{ik}$ is full rank or not can be checked by computing the determinant of $\mathbf{R}_{ik}$ – if the determinant is a rational function such that the product of its numerator and denominator, say $r_{ik}(\underline{\delta}) \in \mathbb{F}_q[\underline{\delta}]$, is non-trivial, then $\mathbf{R}_{ik}$ is full rank, else it is not. Also, computing these determinant values require time that is polynomial in $L, |\mathcal{F}|$ and the transfer function degrees. Therefore, we need the following polynomial to be non-trivial so that all constraints in (3.3) are satisfied:

$$f(\underline{\delta}) = \prod_{k=1}^{L+1} \prod_{(i,j):m_{ij}(\underline{\xi})\not\equiv 0} m_{ij}(\underline{\xi}^{(k)}) \prod_{i=1}^{K} \prod_{k\notin\mathcal{A}_i} r_{ik}(\underline{\delta}).$$

An assignment of $\underline{\delta}$ from $\mathbb{F}_q^{(L+1)(s+c)}$ that makes $f(\underline{\delta})$ non-zero is guaranteed for large enough $\mathbb{F}_q$ using Schwartz-Zippel Lemma. Therefore, this assignment of $\underline{\delta}$ enables each source to transmit at rate of $\frac{1}{L+1}$, and achieve sum rate of $\frac{K}{L+1}$.

A.5 Proof of Theorem 3.3.2

Note that $L = 2$ for this case, therefore, we require $a = b$ and $n = 2a + b$ in order to achieve a rate of $\frac{a}{n} = \frac{1}{3}$ per source. Then the constraints $\dim(\mathcal{W}_i) = a$ for all i, as given in (3.3), imply that precoding matrices satisfy the following relations:

$$\mathbf{M}_{11}\mathbf{V}_1\mathbf{A}_1 = \mathbf{M}_{12}\mathbf{V}_2, \quad \mathbf{M}_{22}\mathbf{V}_2\mathbf{A}_2 = \mathbf{M}_{23}\mathbf{V}_3,$$
$$\mathbf{M}_{33}\mathbf{V}_3\mathbf{A}_3 = \mathbf{M}_{34}\mathbf{V}_4, \quad \mathbf{M}_{44}\mathbf{V}_4\mathbf{A}_4 = \mathbf{M}_{41}\mathbf{V}_1,$$

where $\mathbf{A}_i$, $i = 1,2,3,4$, are full rank $a \times a$ matrices. These relations result in the equation $\mathbf{T}\mathbf{V}_1 = \mathbf{V}_1\mathbf{A}$, where $\mathbf{T} = \mathbf{M}_{12}\mathbf{M}_{23}\mathbf{M}_{34}\mathbf{M}_{41}(\mathbf{M}_{11}\mathbf{M}_{22}\mathbf{M}_{33}\mathbf{M}_{44})^{-1}$ and $\mathbf{A} = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4$; this imposes restrictions on choices of $\mathbf{V}_1$. $\mathbf{T}$ is a diagonal matrix with its (k,k)th entry as $t(\underline{\xi}^{(k)})$. Thus, we have $t(\underline{\xi}^{(k)})\mathbf{v}_k = \mathbf{v}_k\mathbf{A}$, where $\mathbf{v}_k$ is the kth row

of $\mathbf{V}_1$, $k = 1, 2, \ldots, n$. This means if $\mathbf{v}_k$ is not the zero vector for some k, then it is one of the left eigenvectors of $\mathbf{A}$ and $t(\underline{\xi}^k)$ is the corresponding eigenvalue [86]. Since $t(\underline{\xi})$ is not a constant in $\mathbb{F}_q$, the eigenvectors form a linearly independent set and $\mathbf{A}$ is full rank. This implies $\mathbf{v}_k$ is the zero vector for $(n-a)$ instances of k, i.e., $(n-a)$ rows of $\mathbf{V}_1$ are zero vectors. Then the four alignment relations stated above imply that the corresponding $(n-a)$ rows of $\mathbf{V}_i$, $i = 2, 3, 4$, are also zero vectors. One can check that these precoding matrices satisfy $\dim(\mathcal{U}_i \cap \mathcal{W}_i) > 0$ for all i, that makes recovery of desired messages impossible at each destination. Therefore, the sources cannot achieve a rate of $\frac{1}{3}$ each, with $a = b$ using a PBNA-based coding approach. However, if $a < b$ and the relations in (3.3) could be satisfied by some choice of precoding matrices, the achievable rate per source would be at most $\frac{a}{2a+b} = \frac{1}{2+(b/a)}$. Hence, the only possibility for achieving rate close to $\frac{1}{3}$ per source is to choose a, b large enough such that $\frac{b}{a}$ is very close to one; this in turn introduces the requirement that the number of transmissions n should be very large.

A.6 Proof of Theorem 3.3.3

Note that $|\bar{\mathcal{A}}_i| = L + |\mathcal{E}_i| \leq L + \min(d, |\mathcal{B}_i|)$, and $|\bar{\mathcal{A}}_i| = L + d$ for at least one D_i. If $|\bar{\mathcal{A}}_i| = L + d$ and $\bar{\mathcal{B}}_i \neq \emptyset$ for all i, since $\bar{\mathcal{H}}$ has no cycles, we can directly apply Theorem 3.3.1 to achieve a rate of $\frac{1}{L+d+1}$ per source using PBNA scheme under certain constraints. The only other case is that $|\bar{\mathcal{A}}_i| < L + d$ and/or $\bar{\mathcal{B}}_i = \emptyset$ for some values of i (note that $\bar{\mathcal{B}}_i = \emptyset$ implies D_i has chosen to decode messages from all sources in $\mathcal{B}_i$). Then we can introduce unique artificial transfer functions and auxiliary sources to make $|\bar{\mathcal{A}}_i| = L + d$ and $\bar{\mathcal{B}}_i \neq \emptyset$ for each such i.

For example, if $|\bar{\mathcal{A}}_i| = L + d - 2$ and $\bar{\mathcal{B}}_i = \emptyset$ for some i, we construct three dummy sources, say S'_1, S'_2, S'_3, and assume that the corresponding transfer functions with respect to D_i are a set of new variables η_{i1}, η_{i2}, γ_{i3} respectively (that take values from $\mathbb{F}_q$). Thereafter, we make the updates $\bar{\mathcal{A}}_i \leftarrow \bar{\mathcal{A}}_i \cup \{S'_1, S'_2\}$ and $\bar{\mathcal{B}}_i \leftarrow \{S'_3\}$, so that $|\bar{\mathcal{A}}_i| = L + d$ and $|\bar{\mathcal{B}}_i| = 1$. Thus, this procedure ensures $|\bar{\mathcal{A}}_i| = L + d$, $\bar{\mathcal{B}}_i \neq \emptyset$ for all i, and we can use Theorem 3.3.1 to complete the achievability proof.

A.7 Proof of Theorem 3.3.4

Note that d^* lies between $\lceil (|\mathcal{F}| - K - M + 1)/M \rceil$ and $\lfloor |\mathcal{F}|/M \rfloor$ since $\mathcal{H}$ is connected, the number of edges in its spanning tree is $K + M - 1$, and at most Md edges are removed from $\mathcal{H}$ to obtain $\mathcal{K}^*$ (this implies $|\mathcal{F}| - K - M + 1 < Md^* < |\mathcal{F}|$). Also, the edge-set of the complement of $\mathcal{K}^*$ is a basis of $\bar{\mathcal{M}} \cap \mathcal{M}_{d^*}$. The inner loop of the algorithm corresponds to the greedy approach for generating a basis of $\bar{\mathcal{M}} \cap \mathcal{M}_d$ for given d. The outer loop of the algorithm checks if the complement of the basis forms a spanning tree for $\mathcal{H}$ by examining if the size of the obtained basis is $|\mathcal{F}| - K - M + 1$ or not. This, along with the fact that all bases of a matroid have the same size, shows that the algorithm returns d^*, $\mathcal{K}^*$ as answers.

As far as the computational complexity of the algorithm is concerned, the membership of $\mathcal{I} \cup \{e_i\}$ in $\bar{\mathcal{M}} \cup \mathcal{M}_d$ for each i and instance of d can be checked by running BFS (or DFS) algorithm, that requires $O(|\mathcal{F}|)$ time. The inner for-loop of the algorithm runs $|\mathcal{F}|$ times and the outer for-loop runs at most $\lfloor |\mathcal{F}|/M \rfloor - \lceil (|\mathcal{F}| - K - M + 1)/M \rceil \leq \left(1 + \frac{K}{M}\right)$ times. Therefore, the algorithm has an overall computational complexity of $O(\left(1 + \frac{K}{M}\right) |\mathcal{F}|^2)$, if $\mathcal{H}$ is a connected graph.

Appendix B

Proofs for Chapter 4

B.1 Proof of Theorem 4.2.1

We define $\mathcal{R}$ as the range of ϕ, i.e., the set of all graphs in $\mathcal{U}_p$ that can be returned as output by ϕ. Note that the domain of ϕ is $\mathcal{A}^{np}$, that consists of $|\mathcal{A}|^{np}$ entries. Since the graph estimate can be at an edit distance of at most s from the true graph, we have $|\mathcal{R}| \leq B(s,\mathcal{G})|\mathcal{A}|^{np}$. We define $\mathcal{I} = \{i : G_i \in \mathcal{R}\}$; then we get

$$\begin{aligned}
P_{e,s}^{(n)}(\phi) &= \frac{1}{M}\sum_{i=1}^{M} P(\Delta(G_i,\phi(X^n)) \geq s|K = K_i) \\
&\geq \frac{1}{M}\sum_{i\in\mathcal{I}^c} P(\Delta(G_i,\phi(X^n)) \geq s|K = K_i) \\
&= \frac{|\mathcal{I}^c|}{M} \\
&\geq 1 - \frac{|\mathcal{R}|}{M} \\
&\geq 1 - \frac{B(s,\mathcal{G})|\mathcal{A}|^{np}}{M},
\end{aligned}$$

where we use the fact that $P(\Delta(G_i,\phi(X^n)) \geq s|K = K_i) = 1$ for $i \in \mathcal{I}^c$, as then $G_i \in \mathcal{R}^c$, $\phi(X^n) \in \mathcal{R}$. We get the desired result using the definitions of R_1, C.

B.2 Proof of Theorem 4.2.2

We prove the result for the case when $\mathcal{A}$ is a finite set and $\mathcal{K}$ is an ensemble of discrete Markov networks. The proof for the case when $\mathcal{A} = \mathbb{R}$ and the Markov networks have continuous p.d.f.'s is along the same lines. We fix $\epsilon > 0$ and define

$$\mathcal{B}_i = \left\{ x^n \in \mathcal{A}^{np} : \log_2 \frac{f(x^n|K = K_i)}{f(x^n)} \geq n(C_2 + \epsilon) \right\}, \ i = 1, 2, \ldots, M.$$

Thus, $\mathcal{B}_i$ attempts to capture those points in sample space where random variable $\log_2 \frac{f(X^n|K=K_i)}{f(X^n)}$ exceeds its mean conditioned on $K = K_i$ (strictly speaking, it only contains a subset of those points as C_2 is an upper bound on the mean of the random variable). Given any learning algorithm ϕ, we define the following sets:

$$\mathcal{R}_i = \{x^n \in \mathcal{A}^{np} : \Delta(\phi(x^n), G_i) < s\}, \quad \mathcal{S}_i = \{x^n \in \mathcal{A}^{np} : \phi(x^n) = G_i\},$$

for $i = 1, 2, \ldots, M$. If $\mathcal{B}(s, G_i) = \{G_{i_1}, \ldots, G_{i_k}\}$, note that $\mathcal{R}_i = \cup_{t=1}^{k} \mathcal{S}_{i_t}$. Also, $\mathcal{S}_i \cap \mathcal{S}_j = \phi$ for $i \neq j$. Thus, the probability of error-free decoding by ϕ is given by

$$\begin{aligned} P_{c,s}^{(n)}(\phi) &= \frac{1}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i} f(x^n|K = K_i) \\ &= \frac{1}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i \cap \mathcal{B}_i^c} f(x^n|K = K_i) + \frac{1}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i \cap \mathcal{B}_i} f(x^n|K = K_i). \end{aligned}$$

The first term, involving the points in $\mathcal{R}_i \cap \mathcal{B}_i^c$, $i = 1, 2, \ldots, M$, can be bounded as

$$\begin{aligned} \frac{1}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i \cap \mathcal{B}_i^c} f(x^n|K = K_i) &\leq \frac{2^{n(C_2+\epsilon)}}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i \cap \mathcal{B}_i^c} f(x^n) \\ &\leq \frac{2^{n(C_2+\epsilon)}}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i} f(x^n) \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{2^{n(C_2+\epsilon)}}{M} \sum_{i=1}^{M} \sum_{t=1}^{|\mathcal{B}(s,G_i)|} \sum_{x^n \in \mathcal{S}_{i_t}} f(x^n) \\
&\leq \frac{2^{n(C_2+\epsilon)}}{M} \max_i |\mathcal{B}(s,\mathcal{G}_i)| \\
&= \frac{2^{n(C_2+\epsilon)}}{M} B(s,\mathcal{G}) \\
&\leq 2^{n(C_2+\epsilon)-R}.
\end{aligned}
$$

The second term, involving points in $\mathcal{R}_i \cap \mathcal{B}_i$, $i = 1, 2, \ldots, M$, can be bounded as

$$
\begin{aligned}
\frac{1}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{R}_i \cap \mathcal{B}_i} f(x^n | K = K_i) &\leq \frac{1}{M} \sum_{i=1}^{M} \sum_{x^n \in \mathcal{B}_i} f(x^n | K = K_i) \\
&= \frac{1}{M} \sum_{i=1}^{M} P\left(\log_2 \frac{f(X^n | K = K_i)}{f(X^n)} \geq n(C_2 + \epsilon) \,\middle|\, K = K_i \right).
\end{aligned}
$$

Note that $X^{(1)}, \ldots, X^{(n)}$ are i.i.d. vectors, so we have the following simplification:

$$
\begin{aligned}
\operatorname{var}\left(\log \frac{f(X^n | K = K_i)}{f(X^n)} \,\middle|\, K = K_i \right) &= \operatorname{var}\left(\sum_{j=1}^{n} \log \frac{f(X^{(j)} | K = K_i)}{f(X^{(j)})} \,\middle|\, K = K_i \right) \\
&= n \operatorname{var}\left(\log \frac{f(X^{(1)} | K = K_i)}{f(X^{(1)})} \,\middle|\, K = K_i \right),
\end{aligned}
$$

where $\operatorname{var}(\cdot)$ is the variance of random variable. We define the following quantity:

$$
A(\mathcal{K}) = \max_{1 \leq i \leq M} \operatorname{var}\left(\log \frac{f(X^{(1)} | K = K_i)}{f(X^{(1)})} \,\middle|\, K = K_i \right).
$$

Thereafter, using the definition of C_2 and applying Chebyshev's inequality gives

$$
P\left(\log \frac{f(X^n | K = K_i)}{f(X^n)} \geq n(C_2 + \epsilon) \,\middle|\, K = K_i \right) \leq \frac{A(\mathcal{K})}{n\epsilon^2},
$$

for $i = 1, 2, \ldots, M$. Choosing $\epsilon = \frac{R - nC_2}{2n}$ and the relation $P_{e,s}^{(n)}(\phi) = 1 - P_{c,s}^{(n)}(\phi)$ gives

$$
P_{e,s}^{(n)}(\phi) \geq 1 - \frac{4nA(\mathcal{K})}{(R - nC_2)^2} - 2^{-\frac{(R - nC_2)}{2}}.
$$

B.3 Proof of Lemma 4.2.3

The first inequality follows from the proof of Theorem 1 in [60]. For the second inequality, note that for any undirected graphs $G = (V, E(G)) \in \mathcal{G}_{p,d}$, $H = (V, E(H)) \in \mathcal{U}_p$ with $\Delta(G,H) < s$, we have $|E(G,H)| < s$, where $E(G,H)$ is the symmetric difference between the edge sets $E(G)$ and $E(H)$. In other words, $(V, E(G,H))$ is a graph on p nodes and at most $s-1$ edges. Therefore, $B(s, \mathcal{G}_{p,d})$ is no more than the number of graphs on p nodes and at most $s-1$ edges. This gives

$$B(s, \mathcal{G}_{p,d}) \leq \sum_{i=0}^{s-1} \binom{\frac{p(p-1)}{2}}{i} \leq s \binom{\frac{p(p-1)}{2}}{s} < s \binom{\frac{p^2}{2}}{s},$$

where we use the facts that $\binom{m}{i} \leq \binom{m}{j}$ for $0 \leq i \leq j \leq \frac{m}{2}$, and $0 < s \leq \frac{p(p-1)}{4}$.

B.4 Proof of Theorem 4.2.4

We make use of the relations proven in Lemma 4.2.3 to choose bound R as

$$R = \frac{pd}{4} \log_2 \frac{p}{8d} - \log_2 \left[s \binom{\frac{p^2}{2}}{s} \right].$$

Using the facts that $\binom{a}{b} \leq \left(\frac{a \cdot e}{b}\right)^b$ and $s \geq 2$, we obtain the following lower bound:

$$\frac{R}{C_1} \geq \frac{1}{\log_2 |\mathcal{A}|} \left(\left(\frac{d}{4} - \frac{2s}{p} \right) \log_2 p - \frac{d}{4} \log_2 8d \right).$$

By hypothesis of the theorem, we have $n < \frac{R}{2C_1}$. Then Theorem 4.2.1 implies that

$$P_{e,s}^{(n)}(\phi) \geq 1 - 2^{-\frac{R}{2}}. \tag{B.1}$$

Since $d = o(p^\alpha)$ for some $\alpha < 1$ and $2 \leq s < (1-\alpha)\frac{pd}{16}$, we have $R = \Theta(pd \log p)$. This shows that the second term of the RHS of (B.1) goes to 0 as $p \to \infty$ and hence the probability of error of any ϕ approaches 1 as the problem size increases.

B.5 Proof of Lemma 4.2.6

The upper bound on $B(s, \mathcal{G}'_{p,d})$ follows from arguments similar to those used in the proof of Lemma 4.2.3. For the lower bound on $\log_2 |\mathcal{G}'_{p,d}|$, note that there are $N_p = \frac{p!}{2^{p/2}(p/2)!}$ possible perfect matchings on a set of p nodes. Therefore, a multigraph composed of d perfect matchings can be formed in $(N_p)^d$ ways. It is possible that multiple copies of the same multigraph get generated during this construction. Using Lemma 4.2.5, atleast $\left(1 - \frac{c}{p^\tau}\right)(N_p)^d$ of these multigraphs have (weighted) adjacency matrix $\mathbf{A}$ satisfying $\rho(\mathbf{A}) < 3d^{1/2}$, for some constant $c > 0$. Note that any given multigraph generated by d perfect matchings is a d-regular graph and has $\frac{pd}{2}$ edges. In general, each of the edges can come from any of the d perfect matchings. Therefore, a single multigraph can potentially be generated by atmost $d^{\frac{pd}{2}}$ sets of d perfect matchings. Also, we desire that the multigraphs have different underlying undirected graph structures in $\mathcal{G}_{p,d}$. The fact that there can be atmost d edges between two nodes of the multigraph and there are $\frac{pd}{2}$ edges in total, gives the lower bound $|\mathcal{G}'_{p,d}| \geq \left(1 - \frac{c}{p^\tau}\right) \frac{1}{d^{pd}} (N_p)^d$. Choosing $p > (2c)^{\frac{1}{\tau}}$ ensures that $\left(1 - \frac{c}{p^\tau}\right) \geq \frac{1}{2}$; thereafter, we use the fact that $p! \geq \left(\frac{p}{2}\right)^{\frac{p}{2}} \left(\frac{p}{2}\right)!$ to get

$$\log_2 |\mathcal{G}'_{p,d}| \geq \log_2 \left[\frac{1}{d^{pd}} \left(1 - \frac{c}{p^\tau}\right) \left(\frac{p!}{2^{p/2}(p/2)!} \right)^d \right] \geq \frac{pd}{2} \log_2 \frac{p}{4d^2} - 1.$$

B.6 Proof of Lemma 4.2.7

By definition we have $I(K_i; X^{(1)}) = h(X^{(1)}) - h(X^{(1)}|K = K_i)$. Note that the differential entropy of $X^{(1)}$ is upper bounded by the differential entropy of a Gaussian random vector with the same covariance matrix [1]. This gives $h(X^{(1)}) \leq$

$\frac{1}{2}\log_2(2\pi e)^p|\bar{\Sigma}|$, where $\bar{\Sigma} = \frac{1}{M}\sum_{i=1}^{M}\Sigma_i$ and $\Sigma_i = \Theta_i^{-1}$ is the covariance matrix associated with K_i. We also have $h(X^{(1)}|K = K_i) = \frac{1}{2}\log_2(2\pi e)^p|\Sigma_i|$. Thus, this implies

$$I(K_i; X^{(1)}) \leq \frac{1}{2}(\log_2|\bar{\Sigma}| - \log_2|\Sigma_i|) = \frac{1}{2}(\log_2|\bar{\Sigma}| + \log_2|\Theta_i|).$$

By construction, the diagonal entries of Θ_i are same and equal to $4d^{1/2}\mu + \delta$. Therefore, by Hadamard's Inequality, $|\Theta_i| \leq (4d^{1/2}\mu + \delta)^p$. Also, the minimum eigenvalue of Θ_i is atleast δ. This means that the maximum eigenvalue of Σ_i is atmost $\frac{1}{\delta}$ or $||\Sigma_i||_2 \leq \frac{1}{\delta}$. Hence, the maximum eigenvalue of $\bar{\Sigma}$ does not exceed $\frac{1}{\delta}$ as $||\bar{\Sigma}||_2 \leq \frac{1}{M}\sum_{i=1}^{M}||\Sigma_i||_2$. This gives $|\bar{\Sigma}| \leq ||\bar{\Sigma}||_2^p \leq \frac{1}{\delta^p}$. The use of these bounds give

$$I(K_i; X^{(1)}) \leq \frac{p}{2}\log_2\left(1 + \frac{4d^{1/2}\mu}{\delta}\right) = \frac{p}{2}\log_2\left(1 + \frac{4d^{1/2}}{\lambda^{-1} - 4d^{1/2}}\right),$$

where we substitute $\mu = \frac{\delta}{\lambda^{-1} - 4d^{1/2}}$ to get the last term in the above relation.

B.7 Proof of Theorem 4.2.8

We use the relations in Lemmas 4.2.6 and 4.2.7 to choose bounds R, C_2 as

$$R = \frac{pd}{2}\log_2\frac{p}{4d^2} - \log_2\left[s\binom{p^2/2}{s}\right] - 1,$$
$$C_2 = \frac{p}{2}\log_2\left(1 + \frac{4d^{1/2}}{\lambda^{-1} - 4d^{1/2}}\right).$$

Using the facts that $\binom{a}{b} \leq \left(\frac{a \cdot e}{b}\right)^b$ and $s \geq 2$, we obtain the following lower bound:

$$\frac{R}{C_2} \geq \frac{\left(d - \frac{4s}{p}\right)\log_2 p - 2d\log_2 2d}{\log_2\left(1 + \frac{4d^{1/2}}{\lambda^{-1} - 4d^{1/2}}\right)}.$$

By hypothesis of the theorem, we have $n < \frac{R}{2C_2}$. Then Theorem 4.2.2 implies that

$$P_{e,s}^{(n)}(\phi) \geq 1 - \frac{8A(\mathcal{K}_{p,d}^G)}{RC_2} - 2^{-\frac{R}{4}}. \tag{B.2}$$

Analogous to the proof of Theorem B.4, the last term in the RHS of (B.2) goes to 0 as $p \to \infty$, as $d = o(p^\alpha)$ for some $\alpha < \frac{1}{2}$ and $2 \le s < (1-2\alpha)\frac{pd}{8}$ implies $R = \Theta(pd \log_2 p)$. It remains to show that the second term in RHS of (B.2) goes to 0 as $p \to \infty$; we do this by deriving a suitable upper bound for $A(\mathcal{K}_{p,d}^G)$.

We derive a deterministic bound on the variance of $\log_2 \frac{f(X^{(1)}|K=K_i)}{f(X^{(1)})}$. We define $\delta_{\max} = \max_i |\Theta_i|$, $\delta_{\min} = \min_i |\Theta_i|$, $\lambda_{\max}$ to be the maximum among the eigenvalues of Θ_i, $i = 1, 2, \ldots, M$, and $\bar{\Theta} = \frac{1}{M}\sum_{i=1}^M \Theta_i$. Then given $x \in \mathbb{R}^p$, we have

$$\begin{aligned}
\frac{f(x|K=K_i)}{f(x)} &= \frac{|\Theta_i|^{1/2}\exp(-\frac{1}{2}x^T\Theta_i x)}{\frac{1}{M}\sum_{j=1}^M |\Theta_j|^{1/2}\exp(-\frac{1}{2}x^T\Theta_j x)} \\
&\le \frac{\delta_{\max}^{1/2}}{\delta_{\min}^{1/2}} \frac{1}{\frac{1}{M}\sum_{j=1}^M \exp(-\frac{1}{2}x^T\Theta_j x)} \\
&\le \frac{\delta_{\max}^{1/2}}{\delta_{\min}^{1/2}} \exp\left(\frac{1}{2}x^T\bar{\Theta}x\right) \\
&\le \frac{\delta_{\max}^{1/2}}{\delta_{\min}^{1/2}} \exp\left(\frac{\lambda_{\max}}{2}x^Tx\right) \\
\Rightarrow \log_2 \frac{f(x|K=K_i)}{f(x)} &\le \frac{1}{2}\log_2 \frac{\delta_{\max}}{\delta_{\min}} + \frac{\lambda_{\max}}{2}x^Tx, \qquad \text{(B.3)}
\end{aligned}$$

$$\begin{aligned}
\frac{f(x|K=K_i)}{f(x)} &= \frac{|\Theta_i|^{1/2}\exp(-\frac{1}{2}x^T\Theta_i x)}{\frac{1}{M}\sum_{j=1}^M |\Theta_j|^{1/2}\exp(-\frac{1}{2}x^T\Theta_j x)} \\
&\ge \frac{\delta_{\min}^{1/2}}{\delta_{\max}^{1/2}} \exp\left(-\frac{1}{2}x^T\Theta_i x\right) \\
&\ge \frac{\delta_{\min}^{1/2}}{\delta_{\max}^{1/2}} \exp\left(-\frac{\lambda_{\max}}{2}x^Tx\right) \\
\Rightarrow \log_2 \frac{f(x|K=K_i)}{f(x)} &\ge -\frac{1}{2}\log_2 \frac{\delta_{\max}}{\delta_{\min}} - \frac{\lambda_{\max}}{2}x^Tx. \qquad \text{(B.4)}
\end{aligned}$$

Thus, the joint use of inequalities (B.3) and (B.4) for random variable $X \in R^p$ gives

$$\left|\log_2 \frac{f(X|K=K_i)}{f(X)}\right| \leq \frac{1}{2}\log_2 \frac{\delta_{\max}}{\delta_{\min}} + \frac{\lambda_{\max}}{2} X^T X,$$

which in turn leads to the following upper bound on the variance of $\log_2 \frac{f(X|K=K_i)}{f(X)}$:

$$\text{var}\left(\log_2 \frac{f(X|K=K_i)}{f(X)} \,\middle|\, K=K_i\right) \leq \frac{1}{2}\left(\log_2 \frac{\delta_{\max}}{\delta_{\min}}\right)^2 + \frac{\lambda_{\max}^2}{2} E[(X^T X)^2 | K=K_i].$$

For the given ensemble, we have $\delta_{\max} \leq (4d^{1/2}\mu + \delta)^p$, $\delta_{\min} \geq \delta^p$, $\lambda_{\max} = (d + 4d^{1/2})\mu + \delta \leq 5d\mu + \delta$, where $\mu = \frac{\delta}{\lambda^{-1} - 4d^{1/2}}$. We also have $E[(X^T X)^2 | K = K_i] = (\text{Tr}(\Theta_i^{-1}))^2 + \text{Tr}(2\Theta_i^{-2}) \leq \frac{p^2+2p}{\delta^2} \leq \frac{2p^2}{\delta^2}$. Using these, we compute the desired bound:

$$A(\mathcal{K}_{p,d}^G) = \max_{1\leq i\leq M} \text{var}\left(\log_2 \frac{f(X|K=K_i)}{f(X)} \,\middle|\, K=K_i\right) \leq \frac{3p^2}{2}\left(1 + \frac{5d}{\lambda^{-1} - 4d^{1/2}}\right)^2.$$

For $\lambda = O\left(\frac{1}{d^{1/2}}\right)$, $A(\mathcal{K}_{p,d}^G) = O(p^2 d)$. Using the definitions of R, C_2 and scaling of $A(\mathcal{K}_{p,d}^G)$, we see that the second term of RHS of (B.2) goes to 0 as $p \to \infty$; hence, the probability of error of any ϕ approaches 1 as the problem size increases.

B.8 Proof of Lemma 4.3.1

The number of matchings that give rise to the same simple graph in $\mathcal{G}(\mathbf{d})$ is $\prod_{i\in V} d_i!$. Therefore, the probability of choosing a specific simple graph, provided $\mathcal{F}$ is selected uniformly at random, from $\mathcal{G}(\mathbf{d})$ is equal to $\frac{1}{N_{2m}} \prod_{i\in V} d_i!$. Also, as mentioned in [81], the probability of a graph being simple in $\mathcal{G}(\mathbf{d})$ is at least $\frac{q_s}{2}$ for large p. Thus, we have $|\mathcal{G}_s(\mathbf{d})|(\frac{1}{N_{2m}} \prod_{i\in V} d_i!) \geq \frac{q_s}{2}$ – rearranging this gives the first inequality. The second inequality follows from the first inequality and the facts that $N_{2m} = \prod_{j=1}^m (\frac{m+j}{2}) \geq (\frac{m}{2})^m$, $d_i! \leq d_i^{d_i}$ for $i \in V$, and $D_1 = 2m = \sum_{i\in V} d_i$.

B.9 Proof of Lemma 4.3.3

The use of Lemma 4.3.1 and definition of power-law degree sequence gives

$$\log_2 |\mathcal{G}_{s,\alpha}| \geq p\zeta(\alpha) \sum_{k=d_{\min}}^{d_{\max}} k^{1-\alpha} \log_2 \left(\frac{m^{1/2}}{2k}\right) - \nu - \frac{\nu^2}{2} - 1.$$

Given $d_{\min}$ exceeds some constant in value and p is chosen large enough, we can make use of Lemma 4.3.2, the lower and upper bounds given in (4.1), and assumptions on scalings of $d_{\min}, d_{\max}$, with respect to p, to obtain the following relations:

$$\zeta(\alpha) = \frac{1}{S_0} \geq \frac{(\alpha-1)}{2} d_{\min}^{\alpha-1}, \quad 2m = D_1 = p\frac{S_1}{S_0} \geq \frac{p\bar{d}}{4}, \quad \nu \leq \frac{D_2}{D_1} = \frac{S_2}{S_1} \leq 4\tilde{d},$$

$$\sum_{k=d_{\min}}^{d_{\max}} k^{1-\alpha} \log_2 \left(\frac{m^{1/2}}{2k}\right) \geq \int_{d_{\min}}^{d_{\max}} x^{1-\alpha} \log_2 \left(\frac{m^{1/2}}{2x}\right) \geq \frac{\bar{d} d_{\min}^{1-\alpha}}{4(\alpha-1)} \log_2 \left(\frac{(\alpha-1)}{|\alpha-2|} p\right).$$

Using the above inequalities and considering large p give the desired bound.

B.10 Proof of Lemma 4.3.4

We prove that the property in the theorem statement holds with high probability for a uniformly generated graph in $\mathcal{G}_\alpha$; then this would imply that the property holds with high probability for a uniformly chosen graph from $\mathcal{G}_{s,\alpha}$, due to the assumptions made on $d_{\min}, d_{\max}$ (that ensures $\mathcal{G}_{s,\alpha}$ is a subset of $\mathcal{G}_\alpha$ of significant size). Note that if this property does not hold for a graph in $\mathcal{G}_\alpha$, it means there exist two cycles in the r_0-neighborhood of some node. To be precise, there exists a path of nodes $(i_1, i_2, \ldots, i_k)$, $i_l \in V$, with $4 \leq k \leq l_0 = 2r_0 + 1$, alongside edges $(i_1, i_u), (i_k, i_v)$ for some $1 < u, v < k$. Also, for large values of p, $m\,(= \Theta(p)) \geq 3k\,(= O(\log_2 p))$, so that $2m - i \geq m$, $1 \leq i \leq 3k$. Thus, the

probability that such a path exists in uniform choice from $\mathcal{G}_\alpha$ is bounded above by

$$\frac{d_{i_1}d_{i_2}}{2m-1}\frac{(d_{i_2}-1)d_{i_3}}{2m-3}\cdots\frac{(d_{i_{k-1}}-1)d_{i_k}}{2m-2k+1}\frac{(d_{i_1}-2)(d_{i_u}-2)}{2m-2k-1}\frac{(d_{i_k}-2)(d_{i_v}-2)}{2m-2k-3}$$
$$\leq\sum_{k=4}^{l_0}\sum_{1<u,v<k}\sum_{i_1,i_2,\ldots,i_k}\frac{d_{i_1}d_{i_u}}{m}\frac{d_{i_k}d_{i_v}}{m}\prod_{l=1}^{k}\frac{d_{i_l}^2}{m}\leq\sum_{k=4}^{l_0}\frac{k^2D_3^2D_2^{k-1}}{m^{k+2}}=\frac{6l_0^2D_3^2}{D_2m^2}\left(\frac{D_2}{m}\right)^{l_0}.$$

Since $m=\frac{D_1}{2}$, $D_l=p\frac{S_l}{S_0}$, we make can use of the relations in (4.1) to bound the above probability as $O(p^{-\frac{1}{2}}(\log_2 p)^2 d_{\max}^{5-\alpha}d_{\min}^{\alpha-3})$ for $\alpha\in(2,3)$, $O(p^{-\frac{1}{2}}(\log_2 p)^2 d_{\max}^{2(4-\alpha)}d_{\min}^{2\alpha-6})$ for $\alpha\in(3,4)$, and $O(p^{-\frac{1}{2}}(\log_2 p)^2 d_{\min}^2)$ for $\alpha\in(4,\infty)$. The scalings of $d_{\min},d_{\max}$ by assumptions (A1), (A2) ensure that the probability scaling is at most $p^{-\Theta(1)}$ for graphs in $\mathcal{G}_\alpha$. This demonstrates that the property holds for uniformly generated/selected graphs from $\mathcal{G}_\alpha$ and $\mathcal{G}_{s,\alpha}$ with high probability ($\geq 1-p^{-\Theta(1)}$).

B.11 Proof of Lemma 4.3.5

Analogous to the proof of Lemma 4.3.4, we show the result holds for a graph in $\mathcal{G}_\alpha$ generated by choosing a matching uniformly at random. Given $i\in V$ with degree d_i, we consider the spanning tree of its r_0-neighborhood (in case multiple edges exist between two nodes, we replace it by a single edge for the r_0-neighborhood). Note that by Theorem 4.3.4, the r_0-neighborhood of i is almost a tree having at most $B(i,r_0)+1$ edges with high probability. Therefore, using a well-established result the spanning tree structure (over uniform choice of matching) is stochastically dominated by the following two-stage (truncated) branching process (provided $d_{\min}$ exceeds some constant and p is sufficiently large) [16]– a root node connected to d_i offspring nodes in the first generation, and

each node connected to N offspring nodes in the subsequent generations (we consider r_0 generations in total), where N is an integral random variable with p.m.f. $P(N = k) \propto (k+1)f_{k+1}$, $d_{\min} - 1 \leq k \leq d_{\max} - 1$, and f_k is the fraction of nodes with degree k in the power-law configuration model. We define Z_r, $r \geq 0$, as the number of nodes in the rth generation of the branching process. Then an alternate way of describing the branching process is via the sequence $\{Z_r : r \geq 0\}$ – we set $Z_0 = 1$, $Z_1 = d_i$, $Z_r = N_1 + N_2 + \cdots + N_{Z_{r-1}}$, $r \geq 2$, where N_j's are i.i.d. realizations of N. Thus, we have $P(B(i,r) > x) \leq P(Z_r > x)$, for $x \in \mathbb{R}$ and $0 \leq r \leq r_0$.

Note that $P(N = k) = (k+1)^{1-\alpha} S_1^{-1}$, $d_{\min} - 1 \leq k \leq d_{\max} - 1$; this implies

$$E[N^t] \in \begin{cases} \left(\dfrac{(\alpha-2)}{4(t+2-\alpha)} d_{\max}^{t+2-\alpha} d_{\min}^{\alpha-2}, \dfrac{4(\alpha-2)}{(t+2-\alpha)} d_{\max}^{t+2-\alpha} d_{\min}^{\alpha-2} \right) & , \alpha \in (2, t+2), \\ \left(\dfrac{(\alpha-2)}{4(\alpha-t-2)} d_{\min}^{t}, \dfrac{4(\alpha-2)}{(\alpha-t-2)} d_{\min}^{t} \right) & , \alpha \in (t+2, \infty), \end{cases}$$

for $t \in \mathbb{R}$, where we utilize the bounds in (4.1). $\{Z_r, r \geq 1\}$, is a bounded sequence as $d_{\min} \leq Z_r \leq d_{\max}^r$; therefore, $E[Z_r] = d_i (E[N])^{r-1} \leq d_i (4\tilde{d})^{r-1}$, by property of branching processes. Given positive integers s, t, using Theorem 1 from [87] gives

$$E[(N_1 + N_2 + \cdots + N_s)^t] \leq (2e^2)^t s^t (E[N])^t + (t+1)^t s E[N^t]).$$

This result allows us to bound $E[Z_r^t]$, $r \geq 2$, in terms of $E[Z_{r-1}]$, $E[Z_{r-1}^t]$, as follows:

$$E[Z_r^t] \leq (2e^2)^t E[Z_{r-1}^t](E[N])^t + (t+1)^t E[Z_{r-1}] E[N^t],$$

where we use the definition $Z_r = N_1 + \cdots + N_{Z_{r-1}}$. Solving these recursively gives

$$E[Z_r^t] \leq d_i^t (2e^2 E[N])^{(r-1)t} \left(1 + 2 \left(\frac{t+1}{2} \right)^t e^{-2t} \frac{E[N^t]}{(E[N])^t} \right).$$

If we choose $t > (\alpha - 2)$, then for large $d_{\max}$ we get the following upper bounds:

$$E[Z_r^t] \leq \begin{cases} \dfrac{9(3-\alpha)^{t-1}}{(\alpha-2)^{t-1}}(t+1)^t \left(\dfrac{d_{\max}}{d_{\min}}\right)^{t(\alpha-2)} d_i^t (64\tilde{d})^{(r-1)t} & , \ \alpha \in (2,3), \\ \dfrac{9(\alpha-3)}{(t+2-\alpha)}(t+1)^t \left(\dfrac{d_{\max}}{d_{\min}}\right)^{t+2-\alpha} d_i^t (64\tilde{d})^{(r-1)t} & , \ \alpha \in (3,\infty). \end{cases}$$

Note that $P(B(i,r) < x) \leq P(Z_r < x) = P(Z_r^t < x^t) \leq \frac{E[Z_r^t]}{x^t}$. As assumptions (A1), (A2) hold, setting $t = 4$, $x = p^{\frac{1}{4}+\frac{\epsilon}{2}} d_i (64\tilde{d})^{(r-1)}$ for $\alpha \in (2,3)$, and $t = 16\lceil \alpha - 2 \rceil$, $x = p^{\frac{1}{4}+\frac{\epsilon}{8}} d_i (64\tilde{d})^{(r-1)}$ for $\alpha \in (3,\infty)$, leads to a concentration result for the r-neighborhood of node i, $1 \leq r \leq r_0$. The desired concentration result follows from using the union bound inequality on these individual concentration results.

B.12 Proof of Theorem 4.3.6

The proof follows from the application of Theorem 4.2.1 and Lemma 4.3.3 with $s = 0$, since we require exact recovery by the learning algorithm ϕ. Note that $B(0, \mathcal{G}_{s,\alpha}) = 1$, so we choose $R = \log_2 |\mathcal{G}_{s,\alpha}|$ and $C_1 = p \log_2 |\mathcal{A}|$. The given bound on n leads to $P_e^{(n)}(\phi) \geq 1 - 2^{-\Theta(p\tilde{d} \log_2 p)}$ – this value approaches 1 as $p \to \infty$.

B.13 Proof of Theorem 4.3.8

The result of Lemma 4.3.4 implies that the size of r_0-separator set for non-neighboring nodes i, j in a graph from $\mathcal{G}_{s,\alpha}$ is at most two with high probability ($\geq 1 - p^{-\Theta(1)}$), since the presence of at most one cycle in any r_0-neighborhood means that there are at most two distinct paths of length at most r_0 between any two non-neighboring nodes. We define the r_0-separator set for i, j as $U^* \subseteq V$ ($|U^*| \leq 2$). By construction of the SAW tree, $A(i, r_0)$, the number of the nodes at a

distance of r_0 from i in the SAW tree with i as its root satisfies $A(i,r_0) \leq 2B(i,r_0)$ with probability $\geq 1 - p^{-\Theta(1)}$. Thus, applying Lemma 4.3.7 to i,j and set U^* gives

$$|f(x_i|x_j = 1, x_{U^*}) - f(x_i|x_j = -1, x_{U^*})| \leq 2B(i,r_0)(\tanh\theta_{\max})^{r_0},$$

for all $x_i \in \{-1,1\}$, $x_{U^*} \in \{-1,1\}^{|U^*|}$. This implies $\rho(i,j) \leq 2B_{i,r_0}(\tanh\theta_{\max})^{r_0}$, since we can take maximum over x_i, x_{U^*}, followed by minimum over all node sets U satisfying $|U| \leq 2$, both on LHS. Given assumptions (A1), (A2) hold, the use of constraint $\tanh\theta_{\max} < \frac{1}{(64\tilde{d})^2}$ and the probabilistic upper bounds on $B(i,r_0)$, derived in Lemma 4.3.5, gives $\rho(i,j) = o(p^{-\frac{1}{5}})$ for $\alpha \in (2,3)$, and $\rho(i,j) = o(p^{-\frac{1}{50}})$ for $\alpha \in (3,\infty)$ ($\epsilon = \frac{1}{2}$ in both cases) for all pairs of non-neighboring nodes i,j with high probability ($\geq 1 - p^{-\Theta(1)}$); thereby completing the proof of the theorem.

B.14 Proof of Theorem 4.3.9

We use a result from [83] for ferromagnetic Ising model which states that

$$\max_{x_i \in \{-1,1\}} |f(x_i|x_j = 1) - f(x_i|x_j = -1)| \geq \frac{1 - e^{-4\theta_{\min}}}{16}, \tag{B.5}$$

if i,j are neighboring nodes. An interesting property of ferromagnetic Ising model is that for any pair of nodes u,v, $f(\cdot|x_U)$, $x_U \in \{-1,1\}^{|U|}$, is also a ferromagnetic Ising model with the same edge weights, but modified node potentials. This implies

$$\max_{x_i \in \{-1,1\}} |f(x_i|x_j = 1, x_U) - f(x_i|x_j = -1, x_U)| \geq \frac{1 - e^{-4\theta_{\min}}}{16}.$$

The desired bound on $\rho(i,j)$ is obtained after taking maximum over x_U on RHS, followed by taking minimum over all node sets U with $|U| \leq 2$ on RHS again.

B.15 Proof of Theorem 4.3.10

We make use of the following concentration result, that is derived in [61]:

$$P(\max_{x_i,x_j,x_U} |f(x_i|x_j,x_U) - \hat{f}(x_i|x_j,x_U)| > \gamma) \leq 2^{|U|+3} \exp\left(-\frac{n\gamma^2 f_{\min,U}^2}{2(\gamma+2)^2}\right),$$

where i,j are any pair of nodes, U is any node set, and $f_{\min,U} = \min_{x_U} f(x_U)$. Thus, restricting the choice of U to $|U| \leq 2$ and the fact $f_{\min} = \min_{x_U:|U|\leq 2} f(x_U)$ gives

$$P(\max_{x_i,x_j,x_U} |f(x_i|x_j,x_U) - \hat{f}(x_i|x_j,x_U)| > \gamma) \leq 32 \exp\left(-\frac{n\gamma^2 f_{\min}^2}{2(\gamma+2)^2}\right).$$

Note that absolute difference between $|f(x_i|x_j = 1,x_U) - f(x_i|x_j = -1,x_U)|$ and $|\hat{f}(x_i|x_j = 1,x_U) - \hat{f}(x_i|x_j = -1,x_U)|$ is at most 2γ if $|f(x_i|x_j = 1,x_U) - \hat{f}(x_i|x_j = 1,x_U)| \leq \gamma$ and $|f(x_i|x_j = -1,x_U) - \hat{f}(x_i|x_j = -1,x_U)| \leq \gamma$. Then the desired result follows after application of union bound inequality over all choices of i,j,U.

B.16 Proof of Theorem 4.3.11

Note that since Assumptions (A1), (A2) hold, $\theta_{\min} = \Omega(\frac{1}{(\log p)^r})$ for some constant $r > 0$, and Theorems 4.3.8, 4.3.9 are satisfied, as long as $|\hat{\rho}(i,j) - \rho(i,j)| < \frac{\zeta_{n,p}}{2}$, the decision whether edge (i,j) exists or not is made correctly by learning algorithm ϕ^*. Therefore, the probability of exact graph recovery by ϕ^* is at least

$$P\left(|\hat{\rho}(i,j) - \rho(i,j)| < \frac{\zeta_{n,p}}{2} \ \forall i,j \in V\right) \geq 1 - 192p^4 \exp\left(-\frac{n\zeta_{n,p}^2 f_{\min}^2}{2(8+\zeta_{n,p})^2}\right).$$

Thus, to ensure that the probability of correct recovery is $\geq 1 - p^{-\Theta(1)}$, it suffices to have the bound on n as mentioned in the theorem statement. The computational complexity of $O(p^4)$ results from the fact that we need to compute all empirical marginal probabilities having at most 4 variables and provide them to ϕ^*.

Bibliography

[1] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley Series, 1991.

[2] T. M. Cover. Broadcast channels. *IEEE Trans. Inform. Theory*, 18:2–14, 1972.

[3] T. Cover, A. El Gamal, and M. Salehi. Multiple-access channel with arbitrarily correlated sources. *IEEE Trans. Inform. Theory*, 26:648–659, 1980.

[4] R. Kindermann and L. Snell. *Markov Random Fields and their Applications*. American Mathematical Society, 1980.

[5] C. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover, 1998.

[6] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Trans. Inform. Theory*, 46:1204–1216, 2000.

[7] R. Koetter and M. Médard. Beyond routing: An algebraic approach to network coding. *IEEE INFOCOM*, 2002.

[8] A. Das, S. Vishwanath, S. Jafar, and A. Markopoulou. Network coding for multiple unicasts: An interference alignment approach. In *IEEE ISIT*, 2010, http://arxiv.org/abs/1008.0235.

[9] A. Ramakrishnan, A. Das, H. Maleki, A. Markopoulou, S. Jafar, and S. Vishwanath. Network coding for three unicast sessions: interference alignment approaches. In *Allerton*, 2010.

[10] V. R. Cadambe and S. A. Jafar. Interference alignment and the degrees of freedom for the k-user interference channel. *IEEE Transactions on Information Theory*, 54(8):3425–3441, August 2008.

[11] B. Nazer, M. Gastpar, S. A. Jafar, and S. Vishwanath. Ergodic interference alignment. In *IEEE ISIT*, 2009.

[12] M. Jackson. *Social and economic Networks*. Princeton Univ. Press, 2008.

[13] T. Ideker and R. Sharan. Protein networks in disease. *Genome Research*, 18:644–652, 2008.

[14] S. Wu and X. Gu. Gene network: Model, dynamics and simulation. *Computing and Combinatorics, 2005*, 3595:12–21, 2005.

[15] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM*, 1999.

[16] B. Bollobas. *Random graphs*. Cambridge Studies in Advanced Mathematics, 2001.

[17] C. Meng, A. Ramakrishnan, S. Jafar, and A. Markopoulou. On the feasibility of precoding-based network alignment for three unicast sessions. *IEEE ISIT*, July 2012.

[18] A. Das, S. Banerjee, and S. Vishwanath. Linear network coding for multiple groupcast sessions: An interference alignment approach. In *IEEE ITW*, 2013.

[19] A. Das, P. Netrapalli, S. Sanghavi, and S. Vishwanath. Learning markov graphs up to edit distance. In *IEEE ISIT*, 2012.

[20] T. S. Han and M. H. M. Costa. Broadcast channels with arbitrarily correlated sources. *IEEE Trans. Inform. Theory*, 33(5):641–650, Sep. 1987.

[21] R. Ahlswede. Multi-way communication channels. In *Proceedings of 2nd International Symposium on Information Theory*, pages 23–52, 1971.

[22] D. Slepian and J. Wolf. A coding theorem for multiple access channels with correlated sources. *Bell Labs Tech. J.*, 52:1037–1076, 1973.

[23] T. M. Cover and A. El Gamal. Capacity theorems for the relay channel. *IEEE Trans. Inform. Theory*, 25:572–584, 1979.

[24] B. Wang, J. Zhang, and A. Høst-Madsen. On capacity bounds of MIMO relay channel. *Submitted to IEEE Trans. Inform. Theory*, 2004. available at *http://www.eas.asu.edu/ junshan/publications.html*.

[25] M. Costa and A. E. Gamal. The capacity region of the discrete memoryless interference channel with strong interference. *IEEE Trans. Info. Theory*.

[26] R. Etkin. Gaussian interference channel capacity to within one bit. *IEEE Trans. Info. Theory*.

[27] L. Song and R. W. Yeung. Network information flow - multiple sources. *Proc. IEEE Intl. Symp. Inform. Theory*, page 102, 2001.

[28] T. Ho and D. Lun. *Network Coding: An introduction.* Cambridge, 2008.

[29] Ralf Koetter and Muriel Médard. An algebraic approach to network coding. *IEEE/ACM Transactions on Networking*, 11(5):782–795, Oct. 2003.

[30] Sidharth Jaggi, Peter Sanders, Philip A. Chou, Michelle Effros, Sebastian Egner, Kamal Jain, and Ludo M. G. M. Tolhuizen. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, 51(6):1973–1982, 2005.

[31] Tracey Ho, Muriel Médard, Ralf Koetter, David R. Karger, Michelle Effros, Jun Shi, and Ben Leong. A random linear network coding approach to multicast. *IEEE Transactions on Information Theory*, 52, 2006.

[32] A. Rasala-Lehman and E. Lehman. Complexity classification of network information flow problems. In *15th Annual ACM-SIAM SODA*.

[33] S. Riis. Linear versus non-linear boolean functions in network flow. In *Proc. of CISS*.

[34] M. Médard, M. Effros, T. Ho, and D. Karger. On coding for nonmulticast networks. In *Proc. of 41st Allerton Conference*, Oct 2003.

[35] R. Dougherty, C. Freiling, and K. Zeger. Linearity and solvability in multicast networks. *Proc. CISS*, 2004.

[36] Z. Li and B. Li. Network coding: The case of multiple unicast sessions. In *the Proceedings of the 42nd Annual Allerton Conference on Communication, Control, and Computing*, 2004.

[37] Nicholas J. A. Harvey, Robert Kleinberg, and April Rasala Lehman. On the capacity of information networks. *Special Issue of the IEEE Transactions on Information Theory and the IEEE/ACM Transactions on Networking*, 52(6):2345–2364, June 2006.

[38] N. Ratnakar, R. Koetter, and T. Ho. Linear flow equations for network coding in the multiple unicast case. In *Proc. DIMACS Working Group Network Coding*.

[39] D. Traskov, N. Ratnakar, D. S. Lun, R. Koetter, and M. Médard. Network coding for multiple unicasts: An approach based on linear optimization. In *Proc. IEEE ISIT 2006*.

[40] T. Ho, Y. H. Chang, and K. J. Han. On constructive network coding for multiple unicasts. In *Proc. Allerton Conference on Comm., Control and Computing*, 2006.

[41] M. Effros, T. Ho, and S. Kim. A tiling approach to network code design for wireless networks. In *Proc. of IEEE (ITW 2006)*.

[42] S. Huang and A. Ramamoorthy. A note on the multiple unicast capacity of directed acyclic networks. In *IEEE ICC*, 2011.

[43] C. Wang and N. Shroff. Intersession network coding for two simple multicast sessions. *IEEE Allerton*, Sept. 2007.

[44] W. Song, K. Cai, R. Feng, and W. Rui. Solving the two simple multicast network coding problem in time O(E). *IEEE ICCSN*, May 2011.

[45] H. Weingarten, S. Shamai, and G. Kramer. On the compound mimo broadcast channel. In *Proceedings of Annual Information Theory and Applications Workshop UCSD*, 2007.

[46] C. Suh and D. Tse. Interference alignment for cellular networks. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 1037–1044. Ieee, 2008.

[47] N. Lee and J.B. Lim. A novel signaling for communication on mimo y channel: Signal space alignment for network coding. In *Information Theory, 2009. ISIT 2009. IEEE International Symposium on*, pages 2892–2896. IEEE, 2009.

[48] H. Maleki, V. Cadambe, and S. Jafar. Index coding - an interference alignment perspective. *e-print arXiv:1205.1483*, 2012.

[49] S. A. Jafar. Topological interference management through index coding. *e-print ArXiv:1301.3106*, Jan 2013.

[50] C. Suh and K. Ramchandran. Exact-repair mds code construction using interference alignment. *IEEE Transactions on Information Theory*, 57(3):1425–1442, 2011.

[51] V. R. Cadambe, S. A. Jafar, H. Maleki, K. Ramchandran, and C. Suh. Asymptotic interference alignment for optimal repair of mds codes in distributed data storage.

[52] D. Papailiopoulos and A. Dimakis. Distributed storage codes through hadamard designs. In *IEEE ISIT 2011*, pages 1230–1234. IEEE, 2011.

[53] A. Grabowski and R. Kosinski. Ising-based model of opinion formation in a complex network of interpersonal interactions. *Physica A: Statistical Mechanics and its Applications*, 361:651–664, 2006.

[54] F. Vega-Redondo. *Complex social networks*. Cambridge Press, 2007.

[55] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society Series B*, 48:259–279, 1986.

[56] M. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchial context on a large database of object categories. In *IEEE CVPR*, 2010.

[57] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, Feb 2004.

[58] A. Ahmedy, L. Song, and E. P. Xing. Time-varying networks: Recovering temporally rewiring genetic networks during the life cycle of drosophila melanogaster. Technical report, 2008. arXiv.

[59] G. Bresler, E. Mossel, and A. Sly. Reconstruction of Markov random fields from samples: Some observations and algorithms. In *APPROX*, pages 343–356, 2008.

[60] N. Santhanam and M. J. Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *arXiv*, 2009.

[61] A. Anandkumar, V. Y. F. Tan, and A. Willsky. High-dimensional structure learning of Ising models: Tractable graph families. *arXiv Preprint*, 2011.

[62] P. Netrapalli, S. Banerjee, S. Sanghavi, and S. Shakkottai. Greedy learning of markov network structure. In *IEEE Allerton*, 2010.

[63] A. Ray, S. Sanghavi, and S. Shakkottai. Greedy learning of graphical models with small girth. In *IEEE Allerton*, 2012.

[64] W. Wang, M. J. Wainwright, and K. Ramchandran. Information-theoretic bounds on model selection for Gaussian Markov random fields. In *IEEE ISIT*, 2010.

[65] P. Ravikumar and M. Wainwright. High-dimensional ising model selection using l1-regularized logistic regression. *Annals of Statistics*, 38:1287–1319.

[66] A. Anandkumar, V. Y. F. Tan, and A. Willsky. High-dimensional Gaussian graphical model selection: Tractable graph families. *arXiv Preprint*, 2011.

[67] I. Mitliagkas and S. Vishwanath. Strong information-theoretic limits for source/model recovery. In *Proc. of Allerton Conf. on Communication, Control and Computing, Monticello, USA*, 2010.

[68] R. Tandon and P. Ravikumar. On the difficulty of learning power law graphical models. In *IEEE ISIT*, 2013.

[69] F. Chung and L. Lu. *Complex graphs and networks*. American Mathematical Society, 2004.

[70] R. Koetter and M. Médard. An algebraic approach to network coding. *IEEE/ACM Trans. Networking*, 11:782–795, 2003.

[71] F. Kschischang and R. Koetter. Coding for errors and erasures in random network coding. arXiv:cs/0703061v2.

[72] C. Meng, A. Ramakrishnan, A. Markopoulou, and S. A. Jafar. On the feasibility of network alignment for three unicast sessions. Technical report.

[73] C. Meng, A. Das, A. Ramakrishnan, S. Jafar, A. Markopoulou, and S. Vishwanath. Precoding-based network alignment for three unicast sessions. *e-print ArXiv:1305.0868*, May 2013.

[74] H. Maleki, V. Cadambe, and S. Jafar. Index coding: An interference alignment perspective. *IEEE ISIT*, July 2012.

[75] S. Jafar. Topological interference management through index coding. *e-print ArXiv:1301.3106*, Jan 2013.

[76] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press 3rd edition, 2009.

[77] L.E. Reichl and J.H. Luscombe. A modern course in statistical physics. *American Journal of Physics*, 67:1285, 1999.

[78] S. Geman and C. Graffigne. Markov random field image models and their applications to computer vision. In *Proceedings of the International Congress of Mathematicians*, volume 1, page 2. AMS, Providence, RI, 1986.

[79] Y.C. Zhang. Modeling market mechanism with evolutionary games. *Arxiv preprint cond-mat/9803308*, 1998.

[80] J. Friedman. A proof of Alon's second eigenvalue conjecture and related problems. *arXiv*, 2004.

[81] M. Abdullah, C. Cooper, and A. Frieze. Cover time of a random graph with given degree sequence. *Discrete Mathematics*, Nov 2012.

[82] S. Child and J. Barnard. *Higher Algebra*. Macmillan, 1947.

[83] R. Wu, R. Srikant, and J. Ni. Learning graph structure in discrete Markov random fields. In *IEEE NetSciCom*, 2012.

[84] P. Ravikumar, M. J. Wainwright, and J. Lafferty. High-dimensional Ising model selection using l1-regularized logistic regression. *Annals of Statistics*, 2008.

[85] C. C. Johnson A. Jalali and P. K. Ravikumar. On learning discrete graphical models using greedy methods. In *NIPS*, 2012.

[86] D. Dummitt and R. Foote. *Abstract Algebra*. Wiley, 3 edition, 2003.

[87] R. Latala. Estimation of moments of sums of independent real random variables. *The Annals of Probability*, 1997.

Vita

Abhik Kumar Das is a doctoral candidate in the department of Electrical and Computer Engineering at The University of Texas at Austin. He received his B.Tech. degree in Electrical Engineering from the Indian Institute of Technology (I.I.T.) Kanpur, India, in 2008, and M.S. degree in Electrical and Computer Engineering from The University of Texas at Austin in 2010. He has worked as a summer intern at Qualcomm Inc., Santa Clara, in 2012. His research interests are in graphical models, social network analysis, and network information theory.

Permanent address: akdas@utexas.edu

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.