STRUCTURED PREDICTION: STATISTICAL AND COMPUTATIONAL GUARANTEES IN LEARNING AND INFERENCE

by

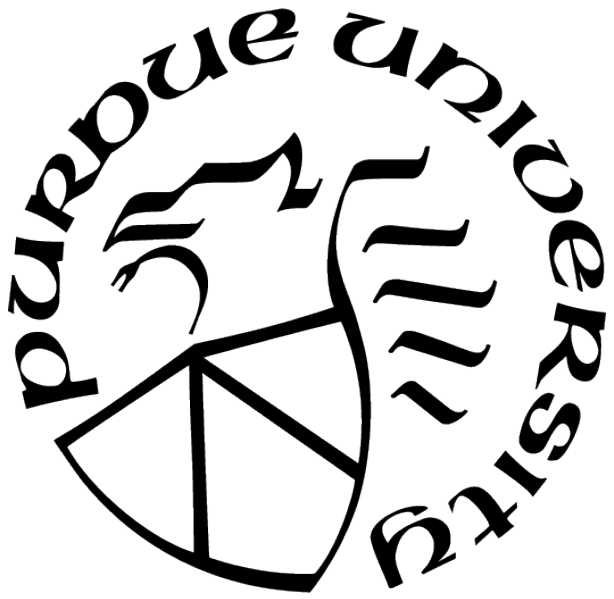
Kevin Bello

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy



Department of Computer Science

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Jean Honorio, Chair

Department of Computer Science

Dr. Jennifer Neville

Department of Computer Science

Dr. Dan Goldwasser

Department of Computer Science

Dr. Elena Grigorescu

Department of Computer Science

Approved by:

Dr. Kihong Park

To my parents Greta & Segundo

ACKNOWLEDGMENTS

First, I am immensely grateful to my advisor Prof. Jean Honorio for his valuable support throughout this five years at Purdue. Particularly, I thank him for his vast patience on the many times I got stuck in a research question; for his technical knowledge and imperative feedback that ensured the culmination of all the projects we worked together; and for his thoughtful guidance on events related to academic development and life in general. I will surely reminisce about all the time we spent in front of the whiteboard discussing ideas and trying to figure out solutions for the unknown. A large portion of the reason I enjoyed so much my doctoral journey is because of Jean.

I also extend my gratitude to my Ph.D. committee members, Dan Goldwasser, Jeniffer Neville, and Elena Grigorescu for putting the time to examine my research outcomes. Especially Prof. Goldwasser, for his keen comments and questions that helped me improve my dissertation.

When I arrived to Lafayette on August 1st, 2016, I was prepared for a lonely voyage. Little did I know I was going to find a lovely community of friends and colleagues that made life much more bearable. Be it watching soccer games of the Peruvian national team, dancing, or enjoying drinks with fun conversations, I thank Jorge, Eliana, Andres, Johnny, Grady, Clara, Eloy, and the rest of the Peruvian Community at Purdue (PCP) for the all the cheerful memories. I shall mention here that the PCP was founded by, the then Purdue Master student, Sulyn Gomez, who I would marry four years after starting my graduate studies. Nonetheless, she deserves her own paragraph of acknowledgments. Finally, I thank Andres for introducing me to Kevin Ro who allowed me to join the training sessions of the West Lafayette FC team in 2018. I will always cherish those moments of joy.

I would also like to thank Asish, Chuyang, Adarsh, Hanbyul, Gregory, and other members of our research group under Prof. Honorio for their time and feedback on presentation rehearsals and for being remarkable co-authors.

No words will ever exist to express the gratitude I feel towards my parents Segundo and Greta and my brother Noel for their undying love and support that have helped me come out victorious against the hardships of the Ph.D. Having been born in San Ramon, a small

town from the Peruvian jungle, I was always mindful of all the hard work my parents went through to secure that my brother and I received the best of health, food, and education. My parents always emphasized that education would open me many doors in life, and they were right. They gave me everything they could have given me and I am here today in great part because of them, thus, I dedicate this dissertation to them.

Lastly, I would like to thank my dear wife Sulyn who has been my partner in crime since 2018. I was proud and wretched when that same year she went to pursue her Ph.D. at Berkeley; instantly however, every holiday or break would translate to a trip to California to see her, which gave rise to an uncountable series of lighthearted memories. At the same time, being a Ph.D. student myself meant I spent several extra hours and weekends working on research, for which I am grateful to her for bear with me during such stressful times. Furthermore, I am thankful to her as during my last semester at Purdue she gave me the best gift a human being can ask for, the gift of life, the light of my life, my daughter Emilia. Both are helping me appreciate the little pleasures of life and are shaping my life in new ways. With all my heart, I look forward to a healthy and happy life with them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Structured prediction consists of receiving a structured input and producing a combinatorial structure such as trees, clusters, networks, sequences, permutations, among others. From the computational viewpoint, structured prediction is in general considered *intractable* because of the size of the output space being exponential in the input size. For instance, in image segmentation tasks, the number of admissible segments is exponential in the number of pixels. A second factor is the combination of the input dimensionality along with the amount of data under availability. In structured prediction it is common to have the input live in a high-dimensional space, which involves to jointly reason about thousands or millions of variables, and at the same time contend with limited amount of data. Thus, learning and inference methods with strong computational and statistical guarantees are desired. The focus of our research is then to propose *principled methods* for structured prediction that are both polynomial time, i.e., *computationally efficient*, and require a polynomial number of data samples, i.e., *statistically efficient.*

The main contributions of this thesis are as follows:

i. We develop an *efficient* and *principled* learning method of latent variable models for structured prediction under Gaussian perturbations. We derive a Rademacher-based generalization bound and argue that the use of non-convex formulations in learning latent-variable models leads to tighter bounds of the Gibbs decoder distortion.

ii. We study the *fundamental limits* of structured prediction, i.e., we characterize the necessary sample complexity for learning factor graph models in the context of structured prediction. In particular, we show that the finiteness of our *novel* PAIR-dimension is necessary for learning. Lastly, we show a connection between the PAIR-dimension and the VC-dimension—which allows for using existing results on VC-dimension to calculate the PAIR-dimension.

iii. We analyze a generative model based on connected graphs, and find the structural conditions of the graph that allow for the exact recovery of the node labels. In particular, we show that exact recovery is realizable in polynomial time for a large class of graphs.

Our analysis is based on convex relaxations, where we thoroughly analyze a semidefinite program and a degree-4 sum-of-squares program. Finally, we extend this model to consider linear constraints (e.g., fairness), and formally explain the effect of the added constraints on the probability of exact recovery.

1. INTRODUCTION

Over the last decade, artificial intelligence (AI) has played a critical role in the progress of several industries, and is sometimes regarded as the "new electricity" by established leaders of the field [1]. In particular, machine learning, a sub-field of AI, has received a lot of attention by the industry and is currently been applied in a wide range of domains, for example, transportation, medicine, speech and image processing, forecasting, and robotics, to name a few. This implies that new problems, packed with their own challenges, are constantly arising, which calls for novel and better algorithms. However, at a high-level, one can discriminate two important factors that drive the complexity of a machine learning problem. One factor is related to what is being predicted, for instance, it could be a real number indicating the likelihood that tomorrow is going to rain, the estimation of the share price of a company in the next quarter, or the classification of a product review into a negative, neutral, or positive category. In this regard, structured prediction consists of receiving a structured input and producing a combinatorial structure such as trees, clusters, networks, sequences, permutations, among others. From the computational viewpoint, structured prediction is in general considered *intractable* because of the size of the output space being exponential in the input size. For instance, in image segmentation tasks, the number of admissible segments is exponential in the number of pixels. A second factor is the combination of the input dimensionality along with the amount of data under availability. In structured prediction it is common to have the input live in a high-dimensional space, which involves to jointly reason about thousands or millions of variables, and at the same time contend with limited amount of data. For instance, consider the bioinformatics problem of predicting the three-dimensional structure of a protein given its amino acid sequence [2, 3], in this case the input sequence generally is of hundreds to thousands amino acids long [4], which requires to jointly process all that information to predict a complex three-dimensional structure. Moreover, due to massively parallel sequencing technology, we know many more protein sequences than protein three-dimensional structures, and the gap is widening rather than diminishing. Another interesting example is dependency parsing [5–7], in which one is given a sentence and the goal is to predict a parse tree that represents its grammatical structure. In this case, a

sentence and its parse tree are usually jointly embedded in a high-dimensional feature vector, a decade ago this construction relied on feature engineering, nowadays deep learning is being applied to automatically create such embeddings. In addition, the number of parse trees of a sentence is exponential in the length of the sentence, where only a few parse trees are actually good candidates. However, in practice we receive only one parse tree for each sentence and this is usually manually annotated, which requires quite a bit of human work to create a single training set. For this reason, not all languages enjoy the luxury of having large training sets for different language applications. Other examples of the applicability of structure prediction include: part-of-speech tagging [8], object detection [9], scene understanding [10–12], phoneme/speech recognition [13, 14], and text-to-speech mapping [15]. Thus, learning and inference methods with strong computational and statistical guarantees are desired. The focus of this research is then to propose *principled methods* for structured prediction that are both polynomial time, i.e., *computationally efficient*, and require a polynomial number of data samples, i.e., *statistically efficient.* Finally, to further motivate this line of work, there is a lack of foundational research in structured prediction as opposed to other machine learning problems, such as binary classification and regression. In what follows, we formally and succinctly introduce the structured prediction problem, and later summarize in a high level the main technical contributions of this thesis.

1.1 Structured Prediction

A large fragment of machine learning models are considered *discriminative* models. *Discriminative learning* is an approach that aims to directly learn a function $f : \mathcal{X} \to \mathcal{Y}$ that maps inputs $x \in \mathcal{X}$ to outputs $y \in \mathcal{Y}$. When $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \mathbb{R}$, we are facing the long-established binary and regression problems, respectively. However, in the structured prediction framework, the output space $\mathcal{Y}$ is some type of structured output such as graphs, sequences, sets, among others. As a concrete example, consider that the input space $\mathcal{X}$ consists of square images, where these images are represented by matrices in $\mathbb{R}^{n \times n}$; let also the output space $\mathcal{Y}$ consists of all matrices in $\{0, 1\}^{n \times n}$. The previous example is an instance of the image segmentation task, where each binary entry in a matrix $y \in \mathcal{Y}$ corresponds to

assigning a pixel to foreground or background. The reader can rapidly notice that the size of the output space is exponential in n; and also that, for a given input x, definitely not all outputs y are equally "good". Then, for a given image x, it is natural to attempt to assign a score to each possible segmentation $y \in \mathcal{Y}$, and choose as prediction the output y that attains the best score—which hopefully corresponds to the best possible segmentation. More formally, let $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a score function that assigns a real number to each given pair (x, y). Moreover, assume that s is parameterized by θ. Then, our prediction function f can be described as:

$$f(x) = \arg\max_{y \in \mathcal{Y}} s(x, y \mid \theta). \tag{1.1}$$

Eq.(1.1) is regarded as the *inference* problem. That is, given a fixed θ, one seeks to efficiently find the output y that maximizes s. Finally, given a set of m observations of input-output pairs $S = \{(x_i, y_i)\}_{i=1}^{m}$, the task of estimating θ from S corresponds to the *learning* problem. We note that the learning problem typically requires performing an inference step during the learning stage.

In this thesis, we aim to answer the following questions:

i. **Learning:**

   (i) How to *efficiently* learn the mapping f from a finite set of samples S?

   (ii) How many observations suffice to learn f so that it generalizes to unseen observations?

   (iii) What is the minimum number of samples needed to learn f so that it performs better than a random predictor?

ii. **Inference:**

   (i) Is it possible to solve eq.(1.1) exactly in polynomial time?

   (ii) What classes of structures allow for efficient exact inference?

   (iii) If one further constrains the inference problem in eq.(1.1), what role do the additional constraints play in exact inference?

1.2 Contributions

The overall contribution of this thesis is to demonstrate that, when facing intractable problems in machine learning, one can aim to *exactly* solve subset of instances where there exists polynomial time methods, or develop approximate algorithms—with guarantees—that yield reasonable results to deal with intractability. Specifically:

i. We propose an *efficient* and *principled* learning method of latent variable models for structured prediction under Gaussian perturbations. Our method is based on cleverly sampling a polynomial number of objects from some proposal distribution that would then guarantee a good approximation of the loss at training time. We derive a Rademacher-based generalization bound and argue that the use of non-convex formulations in learning latent-variable models leads to tighter upper bounds of the Gibbs decoder distortion.

ii. We characterize the *necessary* sample complexity for learning factor graph models in the context of structured prediction. Specifically, we introduce a *novel* type of dimension, named PAIR-dimension, and show that its finiteness is necessary for learning. We further show the connection of the PAIR-dimension to the VC-dimension, which could allow us computing the PAIR-dimension from the several known results on VC-dimension.

iii. We analyze a generative model based on connected graphs, and aim to discover the structural conditions that allow for the exact recovery of the node labels. We show that exact recovery is possible and achievable in polynomial time for multiple classes of graphs of n nodes where their Cheeger constant grows in at least $\mathcal{O}(\log n)$. Our analysis is based on continuous relaxations of a combinatorial problem, where we also thoroughly study the problem under the sum-of-squares hierarchy. Finally, we extend this model to consider linear constraints (e.g., in the context of fairness), and formally explain the effect of the added constraints on the probability of exact recovery.

1.3 Outline and Previously Published Work

The rest of the manuscript is organized as follows. Chapter 2 is concerned with efficient learning of latent-variable models for structured prediction, and describes in detail our main results, proofs, and comparison with previous work. Chapter 3 discusses the fundamental limits of structured prediction based on factor graph models, and describes our main results for the same, along with detailed comparison with prior work. Finally, Chapter 4 delves into understanding exact inference in structured prediction, and describes our main results, proofs, and detailed comparison with prior work.

The bulk of this manuscript is based on the following five papers [16–20], which are joint work of mine with my advisor Jean Honorio. The first paper [16] contains our results for efficient learning in structured prediction. The main results on the fundamental limits of structured prediction are described in [19]. The results on exact inference are contained in [17, 18, 20]. Finally, our work [21] was also published but is not included in this dissertation.

2. EFFICIENT LEARNING OF LATENT VARIABLE MODELS WITH GAUSSIAN PERTURBATIONS

In many tasks it is crucial to take into account latent variables. For example, in machine translation, one is usually given a sentence x and its translation y, but not the linguistic structure h that connects them (e.g., alignments between words). Even if h is not observable, it is important to include this information in the model in order to obtain better prediction results. Examples also arise in computer vision, for instance, most images in indoor scene understanding [22] are cluttered by furniture and decorations, whose appearances vary drastically across scenes, and can hardly be modeled (or even hand-labeled) consistently. In this application, the input x is an image, the structured output y is the layout of the faces (floor, ceiling, walls) and furniture, while the latent structure h assigns a binary label to each pixel (clutter or non-clutter).

2.1 Preliminaries

We denote the input space as $\mathcal{X}$, the output space as $\mathcal{Y}$, and the latent space as $\mathcal{H}$. We assume a distribution D over the observable space $\mathcal{X} \times \mathcal{Y}$. We further assume that we are given a training set S of n i.i.d. samples drawn from the distribution D, i.e., $S \sim D^n$.

Let $\mathcal{Y}_x \neq \emptyset$ denote the countable set of feasible outputs or *decodings* of x. In general, $|\mathcal{Y}_x|$ is exponential with respect to the input size. Likewise, let $\mathcal{H}_x \neq \emptyset$ denote the countable set of feasible latent decodings of x.

We consider a fixed mapping $\mathbf{\Phi}$ from triples to feature vectors to describe the relation among input x, output y, and latent variable h, i.e., for any triple (x, y, h), we have the feature vector $\mathbf{\Phi}(x, y, h) \in \mathbb{R}^k \setminus \{0\}$. For a parameter $\boldsymbol{w} \in \mathcal{W} \subseteq \mathbb{R}^k \setminus \{0\}$, we consider linear decoders of the form:

$$f_{\boldsymbol{w}}(x) = \operatorname*{arg\,max}_{(y,h) \in \mathcal{Y}_x \times \mathcal{H}_x} \langle \mathbf{\Phi}(x, y, h), \boldsymbol{w} \rangle. \tag{2.1}$$

The problem of computing this arg max is typically referred as the *inference* or *prediction* problem. In practice, very few cases of the above general inference problem are tractable,

while most are NP-hard and also hard to approximate within a fixed factor. (For instance, see Section 6.1 in [23] for a thorough discussion).

We denote the *distortion* function by $d : \mathcal{Y} \times \mathcal{Y} \times \mathcal{H} \to [0, 1]$, which measures the dissimilarity among two elements of the output space $\mathcal{Y}$ and one element of the latent space $\mathcal{H}$. (Note that the distortion function is general in the sense that the latent element may not be used in some applications). Therefore, the goal is to find a $\boldsymbol{w} \in \mathcal{W}$ that minimizes the decoder distortion, that is:

$$\min_{\boldsymbol{w} \in \mathcal{W}} \mathop{\mathbb{E}}_{(x,y)\sim D} \left[d(y, (f_{\boldsymbol{w}}(x))) \right]. \tag{2.2}$$

In the above equation, the inner parentheses surrounding $f_{\boldsymbol{w}}(x)$ indicate that we are inserting a pair $(\hat{y}, \hat{h}) = f_{\boldsymbol{w}}(x)$ into the distortion function. From the computational point of view, the above optimization problem is intractable since $d(y, (f_{\boldsymbol{w}}(x)))$ is discontinuous with respect to $\boldsymbol{w}$. From the statistical viewpoint, eq.(2.2) requires access to the data distribution D and would require an infinite amount of data. In practice, one only has access to a finite number of samples.

Furthermore, even if one were able to compute $\boldsymbol{w}$ using the objective in eq.(2.2), this parameter $\boldsymbol{w}$, while achieving low distortion, could potentially be in a neighborhood of parameters with high distortion. Therefore, we can optimize a more *robust* objective that takes into account perturbations. In this chapter we consider Gaussian perturbations. More formally, let $\alpha > 0$ and let $Q(\boldsymbol{w})$ be a unit-variance Gaussian distribution centered at $\alpha\boldsymbol{w}$ of parameters $\boldsymbol{w}' \in \mathcal{W}$. The Gibbs decoder distortion of the perturbation distribution $Q(\boldsymbol{w})$ and data distribution D, is defined as:

$$L(Q(\boldsymbol{w}), D) = \mathop{\mathbb{E}}_{(x,y)\sim D} \left[\mathop{\mathbb{E}}_{\boldsymbol{w}'\sim Q(\boldsymbol{w})} \left[d(y, (f_{\boldsymbol{w}'}(x))) \right] \right]. \tag{2.3}$$

Then, the optimization problem using the Gibbs decoder distortion can be written as:

$$\min_{\boldsymbol{w}\in W} L(Q(\boldsymbol{w}), D).$$

We define the margin $m(x, y, y', h', \boldsymbol{w})$ as follows:

$$m(x, y, y', h', \boldsymbol{w}) = \max_{h \in \mathcal{H}_x} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle - \langle \boldsymbol{\Phi}(x, y', h'), \boldsymbol{w} \rangle.$$

Note that since we are considering latent variables, our definition of margin differs from that of McAllester [24], and Honorio and Jaakkola [23]. For a given pair (x, y) and parameter $\boldsymbol{w}$, let $h^* = \arg\max_{h \in \mathcal{H}_x} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle$. In this case h^* can be interpreted as the latent variable that best explains the pair (x, y). Then, for a fixed $\boldsymbol{w}$, the margin computes the amount by which the pair (y, h^*) is preferred to the pair (y', h').

Next we introduce the concept of *parts*, also used in the work of [24]. Let $c(p, x, y, h)$ be a nonnegative integer that represents the number of times that the part $p \in \mathcal{P}$ appears in the triple (x, y, h). For a part $p \in \mathcal{P}$, we define the feature p as follows:

$$\boldsymbol{\Phi}_p(x, y, h) \equiv c(p, x, y, h).$$

We let $\mathcal{P}_x \neq \emptyset$ denote the set of $p \in \mathcal{P}$ such that there exists $(y, h) \in \mathcal{Y}_x \times \mathcal{H}_x$ with $c(p, x, y, h) > 0$.

2.2 Related Work

During past years, there has been several solutions to address the problem of latent variables in structured prediction. In the field of computer vision, hidden conditional random fields (HCRF) [25–27] have been widely applied for object recognition and gesture detection. In natural language processing, there are also works in applying discriminative probabilistic latent variable models, for example, the training of probabilistic context free grammars with latent annotations in a discriminative manner [28]. The work of Yu and Joachims [29] extends the margin re-scaling SSVM in [30] by introducing latent variables (LSSVM) and obtains a formulation that is optimized using the concave-convex procedure (CCCP) [31]. The work of Ping, Liu, and Ihler [32] considers a smooth objective in LSSVM by incorporating marginal maximum *a posteriori* inference that "averages" over the latent space.

Some of the scarce works in deriving generalization bounds for structured prediction include the work of McAllester [24], which provides PAC-Bayesian guarantees for arbitrary losses; and the work of Cortes, Kuznetsov, Mohri, and Yang [33], which provides data-dependent margin guarantees for a general family of hypotheses with an arbitrary factor graph decomposition. However, with the exception of Honorio and Jaakkola [23], both aforementioned works do not focus on producing computationally tractable methods. Moreover, prior generalization bounds have not focused on latent variables.

2.2.1 Structural Support Vector Machines with Latent Variables

Yu and Joachims [29] extended the formulation of *margin re-scaling* given in [30] by incorporating latent variables. The motivation to extend such formulation is that it leads to a difference of two convex functions, which then allows the use of CCCP [31]. The aforementioned formulation is:

$$\begin{aligned}\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C \cdot \sum_{(x,y)\in S} \max_{(\hat{y},\hat{h})\in\mathcal{Y}_x\times\mathcal{H}_x} \{\langle \boldsymbol{\Phi}(x,\hat{y},\hat{h}), \boldsymbol{w}\rangle + d(y,\hat{y},\hat{h})\} \\ - C \cdot \sum_{(x,y)\in S} \max_{h\in\mathcal{H}_x} \langle \boldsymbol{\Phi}(x,y,h), \boldsymbol{w}\rangle. \end{aligned} \tag{2.4}$$

In the case of standard SSVMs (without latent variables), Tsochantaridis, Joachims, Hofmann, and Altun [30] discussed two advantages of the *slack re-scaling* formulation over the margin re-scaling formulation, these are: the slack re-scaling formulation is *invariant* to the scaling of the distortion function, and the margin re-scaling potentially gives significant score to structures that are not even close to being confusable with the target structures. Altun and Hofmann [34], Collins and Roark [35], and Taskar, Guestrin, and Koller [36] proposed similar formulations to the slack re-scaling formulation. Despite its theoretical advantages, the slack re-scaling has been less popular than the margin re-scaling approach due to computational requirements. In particular, both formulations require optimizing over the output space, but while margin re-scaling preserves the structure of the score and error functions, the slack re-scaling does not. This results in harder inference problems during training. Honorio and Jaakkola [23] also analyzed the slack re-scaling approach and formally showed

that using random structures one can obtain a tighter upper bound of the Gibbs decoder distortion. However, none of these works take into account latent variables.

The following formulation corresponds to the slack re-scaling approach with latent variables:

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{(x,y)\in S} \max_{(\hat{y},\hat{h})\in\mathcal{Y}_x\times\mathcal{H}_x} d(y,\hat{y},\hat{h})\ \mathbb{1}\Big[m(x,y,\hat{y},\hat{h},\boldsymbol{w}) \leq 1\Big] + \lambda\|\boldsymbol{w}\|_2^2. \tag{2.5}$$

We take into account the loss of structures whose margin is less than one (i.e., $m(\cdot) \leq 1$) instead of the Hamming distance, as done in [23]. This is because the former gave better results in preliminary experiments. Also, it is more related to current practice (e.g., [29]). In order to obtain an SSVM-like formulation, the hinge loss is used instead of the discontinuous 0/1 loss in the above formulation. However, note that both eq.(2.4) and eq.(2.5) are now non-convex problems with respect to the learning parameter $\boldsymbol{w}$, even if the hinge loss is used.

2.3 The Maximum Loss Over All Structured Outputs and Latent Variables

In this section we extend the work of McAllester [24] by including latent variables. In the following theorem, we show that the slack re-scaling objective function, eq.(2.5), is an upper bound of the Gibbs decoder distortion, eq.(2.3), up to an statistical accuracy of $\mathcal{O}(\sqrt{\log n/n})$ for n training samples.

Theorem 2.3.1. *Assume that there exists a finite integer value r such that $|\mathcal{Y}_x \times \mathcal{H}_x| \leq r$ for all $(x,y) \in S$. Assume also that $\|\boldsymbol{\Phi}(x,y,h)\|_2 \leq \gamma$ for any triple (x,y,h). Fix $\delta \in (0,1)$. With probability at least $1-\delta/2$ over the choice of n training samples, simultaneously for all parameters $\boldsymbol{w} \in \mathcal{W}$ and unit-variance Gaussian perturbation distributions $Q(\boldsymbol{w})$ centered at $\boldsymbol{w}\gamma\sqrt{8\log(rn/\|\boldsymbol{w}\|_2^2)}$, we have:*

$$L(Q(\boldsymbol{w}), D) \leq \frac{1}{n} \sum_{(x,y)\in S} \max_{(\hat{y},\hat{h})\in\mathcal{Y}_x\times\mathcal{H}_x} d(y,\hat{y},\hat{h})\ \mathbb{1}\Big[m(x,y,\hat{y},\hat{h},\boldsymbol{w}) \leq 1\Big] + \frac{\|\boldsymbol{w}\|_2^2}{n} + \sqrt{\frac{4\|\boldsymbol{w}\|_2^2\,\gamma^2\log(rn/\|\boldsymbol{w}\|_2^2) + \log(2n/\delta)}{2(n-1)}}$$

(See Appendix A for all detailed proofs).

For the proof of the above, we used the PAC-Bayes theorem and well-known Gaussian concentration inequalities. Note that the average sum in the right-hand side, i.e., the objective function, can be equivalently written as:

$$\frac{1}{n}\sum_{(x,y)\in S}\max_{(\hat{y},\hat{h})\in\mathcal{Y}_x\times\mathcal{H}_x}\min_{h\in\mathcal{H}_x} d(y,\hat{y},\hat{h})\ \mathbb{1}\Big[\langle\boldsymbol{\Phi}(x,y,h),\boldsymbol{w}\rangle-\langle\boldsymbol{\Phi}(x,\hat{y},\hat{h}),\boldsymbol{w}\rangle\leq 1\Big].$$

Remark 2.3.2. *It is clear that the above formulation is tight with respect to the latent space $\mathcal{H}_x$ due to the minimization. This is an interesting observation because it reinforces the idea that a non-convex formulation is required in models using latent variables, i.e., an attempt to* convexify *the formulation will result in looser upper bounds and consequently might produce worse predictions. Some other examples of non-convex formulations for latent-variable models are found in [29, 37].*

Note also that the upper bound has a maximization over $\mathcal{Y}_x\times\mathcal{H}_x$ (usually exponential in size) and a minimization over $\mathcal{H}_x$ (potentially in exponential size). We state two important observations in the following remark.

Remark 2.3.3. *First, for the minimization, it is clear that the use of a subset of $\mathcal{H}_x$ would lead to a looser upper bound. However, using a superset $\widetilde{\mathcal{H}}_x\supseteq\mathcal{H}_x$ would lead to a tighter upper bound. The latter relaxation not only can tighten the bound but also can allow the margin to be computed in polynomial time. See for instance some analyses of LP-relaxations in [38–40]. Second, for the maximization, using a subset of $\mathcal{Y}_x\times\mathcal{H}_x$ would lead to a tighter upper bound.*

From the first observation above, we will now introduce a new definition of margin, $\widetilde{m}$, which performs a maximization over a superset $\widetilde{\mathcal{H}}_x\supseteq\mathcal{H}_x$.

$$\widetilde{m}(x,y,y',h',\boldsymbol{w})=\max_{h\in\widetilde{\mathcal{H}}_x}\langle\boldsymbol{\Phi}(x,y,h),\boldsymbol{w}\rangle-\langle\boldsymbol{\Phi}(x,y',h'),\boldsymbol{w}\rangle.$$

Several examples are NP-hard m for $\mathcal{H}$ (DAGs, trees or cardinality constrained sets), but poly-time $\widetilde{m}$ for $\widetilde{\mathcal{H}}$ being a set of binary strings. That is, we can encode any DAG (in $\mathcal{H}$)

as a binary string (in $\widetilde{\mathcal{H}}$), but not all binary strings are DAGs. Later, in Section 2.6, we provide an empirical comparison of the use of m and $\widetilde{m}$. We next present a similar upper bound to the one obtained in Theorem 2.3.1 but now using the margin $\widetilde{m}$.

Theorem 2.3.4 (Relaxed margin bound.)**.** *Assume that there exists a finite integer value r such that $|\mathcal{Y}_x \times \mathcal{H}_x| \leq r$ for all $(x, y) \in S$. Assume also that $\|\boldsymbol{\Phi}(x, y, h)\|_2 \leq \gamma$ for any triple (x, y, h). Fix $\delta \in (0, 1)$. With probability at least $1 - \delta/2$ over the choice of n training samples, simultaneously for all parameters $\boldsymbol{w} \in \mathcal{W}$ and unit-variance Gaussian perturbation distributions $Q(\boldsymbol{w})$ centered at $\boldsymbol{w}\gamma\sqrt{8 \log (rn/\|\boldsymbol{w}\|_2^2)}$, we have:*

$$L(Q(\boldsymbol{w}), D) \leq \frac{1}{n} \sum_{(x,y)\in S} \max_{(\hat{y},\hat{h})\in\mathcal{Y}_x\times\mathcal{H}_x} d(y, \hat{y}, \hat{h})\ \mathbb{1}\Big[\widetilde{m}(x, y, \hat{y}, \hat{h}, \boldsymbol{w}) \leq 1\Big] + \frac{\|\boldsymbol{w}\|_2^2}{n} + \sqrt{\frac{4\|\boldsymbol{w}\|_2^2\, \gamma^2 \log (rn/\|\boldsymbol{w}\|_2^2) + \log (2n/\delta)}{2(n-1)}}.$$

From the second observation in Remark 2.3.3, it is natural to ask what elements should constitute this subset in order to control the statistical accuracy with respect to the Gibbs decoder. Finally, if the number of elements is polynomial then we also have an efficient computation of the maximum. We provide answers to these questions in the next section.

2.4 The Maximum Loss Over Random Structured Outputs and Latent Variables

In this section, we show the relation between PAC-Bayes bounds and the maximum loss over random structured outputs and latent variables sampled i.i.d. from some proposal distribution.

2.4.1 A More Efficient Evaluation

Instead of using a maximization over $\mathcal{Y}_x \times \mathcal{H}_x$, we will perform a maximization over a set $T(\boldsymbol{w}, x)$ of random elements sampled i.i.d. from some proposal distribution $R(\boldsymbol{w}, x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$. More explicitly, our new formulation is:

$$\min_{\boldsymbol{w}} \frac{1}{n} \sum_{(x,y)\in S} \max_{(\hat{y},\hat{h})\in T(\boldsymbol{w},x)} d(y, \hat{y}, \hat{h})\ \mathbb{1}\Big[\widetilde{m}(x, y, \hat{y}, \hat{h}, \boldsymbol{w}) \leq 1\Big] + \lambda \|\boldsymbol{w}\|_2^2. \tag{2.6}$$

We make use of the following two assumptions in order for $|T(\boldsymbol{w}, x)|$ to be polynomial, even when $|\mathcal{Y}_x \times \mathcal{H}_x|$ is exponential with respect to the input size.

Assumption 2.4.1 (Maximal distortion [23])**.** *The proposal distribution $R(\boldsymbol{w}, x)$ fulfills the following condition. There exists a value $\beta \in [0, 1)$ such that for all $(x, y) \in S$ and $\boldsymbol{w} \in \mathcal{W}$:*

$$\mathop{\mathbb{P}}_{(y',h')\sim R(\boldsymbol{w},x)} \Big[d(y, y', h') = 1\Big] \geq 1 - \beta.$$

Assumption 2.4.2 (Low norm)**.** *The proposal distribution $R(\boldsymbol{w}, x)$ fulfills the condition for all $(x, y) \in S$ and $\boldsymbol{w} \in \mathcal{W}$:*[1]

$$\left\| \mathop{\mathbb{E}}_{(y',h')\sim R(\boldsymbol{w},x)} \Big[\boldsymbol{\Phi}(x, y, h^*) - \boldsymbol{\Phi}(x, y', h')\Big] \right\|_2 \leq \frac{1}{2\sqrt{n}} \leq \frac{1}{2\|\boldsymbol{w}\|_2},$$

where $h^* = \arg\max_{h\in\mathcal{H}_x} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w}\rangle$.

In Section 2.5 we provide examples for Assumptions 2.4.1 and 2.4.2 which allow us to obtain $|T(\boldsymbol{w}, x)| = \mathcal{O}\Big(\frac{1}{\log(1/(\beta+\mathrm{e}^{-1/(\gamma^2\|\boldsymbol{w}\|_2^2)}))}\Big)$. Note that β plays an important role in the number of samples that we need to draw from the proposal distribution $R(\boldsymbol{w}, x)$.

2.4.2 Statistical Analysis

In this approach, randomness comes from two sources, from the training data S and the random set $T(\boldsymbol{w}, x)$. That is, in Theorem 2.3.1, randomness only stems from the training

[1]↑The second inequality follows from an implicit assumption made in Theorem 2.3.1, i.e., $\|\boldsymbol{w}\|_2^2/n \leq 1$ since the distortion function d is at most 1.

set S. Now we need to produce generalization results that hold for all the sets $T(\boldsymbol{w}, x)$, and for all possible proposal distributions $R(\boldsymbol{w}, x)$. The following assumption will allow us to upper-bound the number of possible proposal distributions $R(\boldsymbol{w}, x)$.

Assumption 2.4.3 (Linearly inducible ordering [23])**.** *The proposal distribution $R(\boldsymbol{w}, x)$ depends solely on the linear ordering induced by the parameter $\boldsymbol{w} \in \mathcal{W}$ and the mapping $\boldsymbol{\Phi}(x, \cdot, \cdot)$. More formally, let $r(x) \equiv |\mathcal{Y}_x \times \mathcal{H}_x|$ and thus $\mathcal{Y}_x \times \mathcal{H}_x \equiv \{(y_1, h_1), \ldots, (y_{r(x)}, h_{r(x)})\}$. Let $\boldsymbol{w}, \boldsymbol{w}' \in \mathcal{W}$ be any two arbitrary parameters. Let $\boldsymbol{\pi}(x) = (\pi_1, \ldots, \pi_{r(x)})$ be a permutation of $\{1, \ldots, r(x)\}$ such that $\langle \boldsymbol{\Phi}(x, y_{\pi_1}, h_{\pi_1}), \boldsymbol{w} \rangle < \cdots < \langle \boldsymbol{\Phi}(x, y_{\pi_{r(x)}}, h_{\pi_{r(x)}}), \boldsymbol{w} \rangle$. Also, let $\boldsymbol{\pi}'(x) = (\pi'_1, \ldots, \pi'_{r(x)})$ be a permutation of $\{1, \ldots, r(x)\}$ such that $\langle \boldsymbol{\Phi}(x, y_{\pi'_1}, h_{\pi'_1}), \boldsymbol{w}' \rangle < \cdots < \langle \boldsymbol{\Phi}(x, y_{\pi'_{r(x)}}, h_{\pi'_{r(x)}}), \boldsymbol{w}' \rangle$. For all $\boldsymbol{w}, \boldsymbol{w}' \in \mathcal{W}$ and $x \in \mathcal{X}$, if $\boldsymbol{\pi}(x) = \boldsymbol{\pi}'(x)$ then $\mathbb{KL}\left(R(\boldsymbol{w}, x) \| R(\boldsymbol{w}', x)\right) = 0$. In this case, we say that the proposal distribution fulfills $R(\boldsymbol{\pi}(x), x) \equiv R(\boldsymbol{w}, x)$.*

In Assumption 2.4.3, geometrically speaking, for a fixed x, we first project the feature vectors $\boldsymbol{\Phi}(x, y, h)$ of all $(y, h) \in \mathcal{Y}_x \times \mathcal{H}_x$ onto the lines $\boldsymbol{w}$ and $\boldsymbol{w}'$. Let $\boldsymbol{\pi}(x)$ and $\boldsymbol{\pi}'(x)$ be the resulting ordering of the structured outputs after projecting them onto $\boldsymbol{w}$ and $\boldsymbol{w}'$ respectively. Two proposal distributions $R(\boldsymbol{w}, x)$ and $R(\boldsymbol{w}', x)$ are the same provided that $\boldsymbol{\pi}(x) = \boldsymbol{\pi}'(x)$. That is, the specific values of $\langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle$ and $\langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w}' \rangle$ are irrelevant, and only their ordering matters.

In Section 2.5 we show an example that fulfills Assumption 2.4.3, which corresponds to a generalization of Algorithm 2 proposed in [23] for any structure with computationally efficient local changes.

In the following theorem, we show that our new formulation in eq.(2.6) is related to an upper bound of the Gibbs decoder distortion up to statistical accuracy of $\mathcal{O}(\log^2 n / \sqrt{n})$ for n training samples.

Theorem 2.4.4. *Assume that there exist finite integer values r, $\tilde{r}$, ℓ, and γ such that $|\mathcal{Y}_x \times \mathcal{H}_x| \leq r$ and $|\widetilde{\mathcal{H}}_x| \leq \tilde{r}$ for all $(x, y) \in S$, $|\cup_{(x,y)\in S} \mathcal{P}_x| \leq \ell$, and $\|\boldsymbol{\Phi}(x, y, h)\|_2 \leq \gamma$ for any triple (x, y, h). Assume that the proposal distribution $R(\boldsymbol{w}, x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.1 with value β, as well as Assumptions 2.4.2 and 2.4.3. Assume that $\|\boldsymbol{w}\|_2^2 \leq \frac{1}{128\gamma^2 \log(1/(1-\beta))}$. Fix $\delta \in (0, 1)$ and an integer $\mathfrak{s}$ such that $3 \leq 2\mathfrak{s} + 1 \leq$*

$\frac{9}{20}\sqrt{\ell(r+1)+1}$. *With probability at least* $1-\delta$ *over the choice of both* n *training samples and* n *sets of random structured outputs and latent variables, simultaneously for all parameters* $\boldsymbol{w} \in \mathcal{W}$ *with* $\|\boldsymbol{w}\|_0 \leq \mathfrak{s}$*, unit-variance Gaussian perturbation distributions* $Q(\boldsymbol{w})$ *centered at* $\boldsymbol{w}\gamma\sqrt{8\log\left(rn/\|\boldsymbol{w}\|_2^2\right)}$*, and for sets of random structured outputs* $T(\boldsymbol{w}, x)$ *sampled i.i.d. from the proposal distribution* $R(\boldsymbol{w}, x)$ *for each training sample* $(x, y) \in S$*, such that* $|T(\boldsymbol{w}, x)| = \left\lceil \frac{1}{2} \frac{\log n}{\log(1/(\beta + \mathrm{e}^{-1/(128\gamma^2\|\boldsymbol{w}\|_2^2)}))} \right\rceil$*, we have:*

$$
\begin{aligned}
L(Q(\boldsymbol{w}), D) \leq \frac{1}{n} \sum_{(x,y)\in S} \max_{(\hat{y},\hat{h})\in T(\boldsymbol{w},x)} d(y, \hat{y}, \hat{h}) \; \mathbb{1}\Big[\widetilde{m}(x, y, \hat{y}, \hat{h}, \boldsymbol{w}) \leq 1\Big] + \frac{\|\boldsymbol{w}\|_2^2}{n} \\
+ \sqrt{\frac{4\|\boldsymbol{w}\|_2^2 \gamma^2 \log \frac{rn}{\|\boldsymbol{w}\|_2^2} + \log \frac{2n}{\delta}}{2(n-1)}} + \sqrt{\frac{1}{n}} + 3\sqrt{\frac{\mathfrak{s}(\log \ell + 2\log(nr)) + \log(4/\delta)}{n}} \\
+ \frac{1}{\log(1/(\beta + \mathrm{e}^{-1/(128\gamma^2\|\boldsymbol{w}\|_2^2)}))} \sqrt{\frac{(2\mathfrak{s}+1)\log(\ell(n\tilde{r}+1)+1)\log^3(n+1)}{n}}.
\end{aligned}
$$

The proof of the above is based on Theorem 2.3.4 as a starting point. In order to account for the computational aspect of requiring sets $T(\boldsymbol{w}, x)$ of polynomial size, we use Assumptions 2.4.1 and 2.4.2 for bounding a *deterministic* expectation. In order to account for the statistical aspects, we use Assumption 2.4.3 and Rademacher complexity arguments for bounding a *stochastic* quantity for all sets $T(\boldsymbol{w}, x)$ of random structured outputs and latent variables, and all possible proposal distributions $R(\boldsymbol{w}, x)$.

Remark 2.4.5. *A straightforward application of Rademacher complexity in the analysis of [23] leads to a bound of* $\mathcal{O}(|\mathcal{H}_x|/\sqrt{n})$*. Technically speaking, a classical Rademacher complexity states that: let* $\mathcal{F}$ *and* $\mathcal{G}$ *be two hypothesis classes. Let* $\min(\mathcal{F}, \mathcal{G}) = \{\min(f, g) \mid f \in \mathcal{F}, g \in \mathcal{G}\}$*. Then* $\mathfrak{R}(\min(\mathcal{F}, \mathcal{G})) \leq \mathfrak{R}(\mathcal{F}) + \mathfrak{R}(\mathcal{G})$*. If we were to use such result, then Theorem 2.4.4 would contain a* $\mathcal{O}(|\mathcal{H}_x|/\sqrt{n})$ *term, or equivalently* $\mathcal{O}(r/\sqrt{n})$*. This would be prohibitive since* r *is typically exponential size, and one would require a very large number of samples* n *in order to have a useful bound, i.e., to make* $\mathcal{O}(r/\sqrt{n})$ *close to zero. In the proof of Theorem 2.4.4, we show a way to tighten the bound to* $\mathcal{O}(\sqrt{\log|\mathcal{H}_x|/n})$*.*

2.5 Examples

Here we provide several examples that fulfill the three main assumptions of our theoretical results.

2.5.1 Examples for Assumption 2.4.1

First we argue that we can perform a change of measure between different proposal distributions. This allows us to focus on uniform proposals afterwards.

Claim 2.5.1 (Change of measure)**.** *Let $R(\boldsymbol{w}, x)$ and $R'(\boldsymbol{w}, x)$ two proposal distributions, both with support on $\mathcal{Y}_x \times \mathcal{H}_x$. Assume that $R(\boldsymbol{w}, x)$ fulfills Assumption 2.4.1 with value β_1. Let $r_{\boldsymbol{w},x}(\cdot)$ and $r'_{\boldsymbol{w},x}(\cdot)$ be the probability mass functions of $R(\boldsymbol{w}, x)$ and $R'(\boldsymbol{w}, x)$ respectively. Assume that the total variation distance between $R(\boldsymbol{w}, x)$ and $R'(\boldsymbol{w}, x)$ fulfills for all $(x, y) \in S$ and $\boldsymbol{w} \in \mathcal{W}$:*

$$\mathsf{TV}(R(\boldsymbol{w}, x) \mid R'(\boldsymbol{w}, x)) \equiv \frac{1}{2} \sum_{(y,h)} |r_{\boldsymbol{w},x}(y, h) - r'_{\boldsymbol{w},x}(y, h)| \leq \beta_2.$$

Then $R'(\boldsymbol{w}, x)$ fulfills Assumption 2.4.1 with $\beta = \beta_1 + \beta_2$ provided that $\beta_1 + \beta_2 \in [0, 1)$.

Proof. For all $(x, y) \in S$ and $\boldsymbol{w} \in \mathcal{W}$, by definition of the total variation distance, we have for any event $\mathcal{A}(x, y, y', h', \boldsymbol{w})$:

$$\left| \mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} [\mathcal{A}(x, y, y', h', \boldsymbol{w})] - \mathop{\mathbb{P}}_{(y',h') \sim R'(\boldsymbol{w},x)} [\mathcal{A}(x, y, y', h', \boldsymbol{w})] \right| \leq \mathsf{TV}(R(\boldsymbol{w}, x) \| R'(\boldsymbol{w}, x))$$

Let the event $\mathcal{A}(x, y, y', h', \boldsymbol{w}) : d(y, y', h') = 1$ and $1 - m(x, y, y', h', \boldsymbol{w}) \geq 0$. Since $R(\boldsymbol{w}, x)$ fulfills Assumption 2.4.1 with value β_1 and since $\mathsf{TV}(R(\boldsymbol{w}, x) \| R'(\boldsymbol{w}, x)) \leq \beta_2$, we have that for all $(x, y) \in S$ and $\boldsymbol{w} \in \mathcal{W}$:

$$\begin{aligned} \mathop{\mathbb{P}}_{(y',h') \sim R'(\boldsymbol{w},x)} [\mathcal{A}(x, y, y', h', \boldsymbol{w})] &\geq \mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} [\mathcal{A}(x, y, y', h', \boldsymbol{w})] - \mathsf{TV}(R(\boldsymbol{w}, x) \| R'(\boldsymbol{w}, x)) \\ &\geq 1 - \beta_1 - \beta_2, \end{aligned}$$

which proves our claim. □

Next, we present a new result for permutations and for a distortion that returns the number of different positions. We later use this result for an image matching application in the experiments section.

Claim 2.5.2 (Permutations)**.** *Let $\mathcal{Y}_x$ be the set of all permutations of v elements, such that $v > 1$. Let y_i be the i-th element in the permutation y. Let $d(y, y', h) = \frac{1}{v}\sum_{i=1}^{v} \mathbb{1}[y_i \neq y'_i]$. The uniform proposal distribution $R(\boldsymbol{w}, x) = R(x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.1 with $\beta = 2/3$.*

Proof. Since $\mathcal{Y}_x$ is the set of all permutations of v elements, then $|\mathcal{Y}_x| = v!$. In addition, since $d(y, y', h) = \frac{1}{v}\sum_{i=1}^{v} \mathbb{1}[y_i \neq y'_i]$ and since $R(x)$ is a uniform proposal distribution with support on $\mathcal{Y}_x \times \mathcal{H}_x$, we have:

$$\begin{aligned} \mathop{\mathbb{P}}_{(y',h')\sim R(x)}[d(y, y', h') = 1] &= \mathop{\mathbb{P}}_{y'}[d(y, y') = 1] \\ &= \frac{F(v)}{v!} \\ &\geq 1 - 2/3. \end{aligned} \tag{2.7.a}$$

For a fixed y, the function $F(v)$ in step eq.(2.7.a) represents the number of permutations $y' \in \mathcal{Y}_x$ such that $d(y, y', h) = 1$. Moreover, $F(v)$ can be computed through the following recursion: $F(v) = (v-1)! \times (1 + \sum_{i=1}^{v-2} \frac{F(i)}{i!})$. The probability is then $F(v)/v!$, it can be seen that this probability converges as $v \to \infty$ through the following: $\lim_{v\to\infty} \frac{F(v+1)}{(v+1)!} - \frac{F(v)}{v!} = 0$. The probability converges to 0.3679 approximately, while achieving a minimum value of $1/3$ at $v = 3$. Hence $\beta = 2/3$. □

Honorio and Jaakkola [23] presented several examples of distortion functions of the form $d(y, y')$, for directed spanning trees, directed acyclic graphs and cardinality-constrained sets, and a distortion function that returns the number of different edges/elements; as well as, for any type of structured output and binary distortion functions. For completeness, we next present the examples provided in [23] since we make use of the suggested β values in our synthetic experiments. Although their proofs are given without using latent variables, it is straightforward to extend their claims by marginalizing over h.

i. *Any type of structured output for binary distortion functions.* Let $\mathcal{Y}_x \times \mathcal{H}_x$ be an arbitrary countable set of feasible decodings of x, such that $|\mathcal{Y}_x| \geq 2$ for all $(x, y) \in S$. Let $d(y, y', h) = \mathbb{1}[y \neq y']$. The uniform proposal distribution $R(\boldsymbol{w}, x) = R(x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.1 with $\beta = 1/2$.

ii. *Directed spanning trees for a distortion function that returns the number of different edges.* Let $\mathcal{Y}_x$ be the set of directed spanning trees of v nodes. Let $A(y)$ be the adjacency matrix of $y \in \mathcal{Y}_x$. Let $d(y, y', h) = \frac{1}{2(v-1)} \sum_{ij} |A(y)_{ij} - A(y')_{ij}|$. The uniform proposal distribution $R(\boldsymbol{w}, x) = R(x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.1 with $\beta = \frac{v-2}{v-1}$.

iii. *Directed acyclic graphs for a distortion function that returns the number of different edges.* Let $\mathcal{Y}_x$ be the set of directed acyclic graphs of v nodes and b parents per node, such that $2 \leq b \leq v - 2$. Let $A(y)$ be the adjacency matrix of $y \in \mathcal{Y}_x$. Let $d(y, y', h) = \frac{1}{b(2v-b-1)} \sum_{ij} |A(y)_{ij} - A(y')_{ij}|$. The uniform proposal distribution $R(\boldsymbol{w}, x) = R(x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.1 with $\beta = \frac{b^2+2b+2}{b^2+3b+2}$.

iv. *Cardinality-constrained sets for a distortion function that returns the number of different elements.* Let $\mathcal{Y}_x$ be the set of sets of b elements chosen from v possible elements, such that $b \leq v/2$. Let $d(y, y', h) = \frac{1}{2b}(|y - y'| + |y' - y|)$. The uniform proposal distribution $R(\boldsymbol{w}, x) = R(x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.1 with $\beta = 1/2$.

2.5.2 Examples for Assumption 2.4.2

The claim below is for a particular instance of a sparse mapping and a uniform proposal distribution.

Claim 2.5.3 (Sparse mapping)**.** *Let $b > 0$ be an arbitrary integer value. For all $(x, y) \in S$ with $h^* = \arg\max_{h \in \mathcal{H}_x} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle$, let $\Upsilon_x = \cup_{p \in \mathcal{P}_x} \Upsilon_x^p$, where the partition Υ_x^p is defined as follows for all $p \in \mathcal{P}_x$:*

$$\Upsilon_x^p \equiv \{(y', h') \mid |\boldsymbol{\Phi}_p(x, y, h^*) - \boldsymbol{\Phi}_p(x, y', h')| \leq b \text{ and } (\forall q \neq p)\ \boldsymbol{\Phi}_q(x, y, h^*) = \boldsymbol{\Phi}_q(x, y', h')\}.$$

If $n \leq |\mathcal{P}_x|/(4b^2)$ for all $(x, y) \in S$, then the uniform proposal distribution $R(\boldsymbol{w}, x) = R(x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$ fulfills Assumption 2.4.2.

Proof. Let $\Delta \equiv \boldsymbol{\Phi}(x, y, h^*) - \boldsymbol{\Phi}(x, y', h')$. Let $p \in \mathcal{P}_x$ be a superindex denoting the partitions, i.e., for all $p \in \mathcal{P}_x$, let $\Delta^p \equiv \boldsymbol{\Phi}(x, y, h^*) - \boldsymbol{\Phi}(x, y', h')$ for some $(y', h') \in \Upsilon_x^p$. By assumption, since $(y', h') \in \Upsilon_x^p$ then $|\Delta_p^p| \leq b$ and $(\forall q \neq p)\ \Delta_q^p = 0$. Therefore:

$$
\begin{aligned}
\left\| \mathop{\mathbb{E}}_{(y',h') \sim R(x)} [\Delta] \right\|_2 &= \sqrt{\sum_{q \in \mathcal{P}_x} \mathop{\mathbb{E}}_{(y',h') \sim R(x)} \left[\Delta_q\right]^2} \\
&\leq \sqrt{\sum_{q \in \mathcal{P}_x} \mathop{\mathbb{E}}_{(y',h') \sim R(x)} \left[|\Delta_q|\right]^2} \\
&= \sqrt{\sum_{q \in \mathcal{P}_x} \left(\sum_{p \in \mathcal{P}_x} \mathop{\mathbb{P}}_{(y',h') \sim R(x)} \left[(y', h') \in \Upsilon_x^p\right] |\Delta_q^p| \right)^2} \\
&= \sqrt{\sum_{q \in \mathcal{P}_x} \left(\mathop{\mathbb{P}}_{(y',h') \sim R(x)} \left[(y', h') \in \Upsilon_x^q\right] |\Delta_q^q| \right)^2} \\
&\leq \sqrt{|\mathcal{P}_x| \left(\frac{b}{|\mathcal{P}_x|} \right)^2} \\
&= b/\sqrt{|\mathcal{P}_x|},
\end{aligned}
$$

where we used the fact that for a uniform proposal distribution $R(x)$, we have: $\mathbb{P}_{(y',h') \sim R(\boldsymbol{w},x)} \left[(y', h') \in \Upsilon_x^q\right] = 1/|\mathcal{P}_x|$. Finally, since we assume that $n \leq |\mathcal{P}_x|/(4b^2)$, we have $b/\sqrt{|\mathcal{P}_x|} \leq 1/(2\sqrt{n})$ and we prove our claim. □

The claim below is for a particular instance of a dense mapping and an *arbitrary* proposal distribution.

Claim 2.5.4 (Dense mapping)**.** *Let a finite $b > 0$ be an arbitrary integer value. Let $|\boldsymbol{\Phi}_p(x, y, h^*) - \boldsymbol{\Phi}_p(x, y', h')| \leq \frac{b}{|\mathcal{P}_x|}$ for all $(x, y) \in S$ with $h^* = \arg\max_{h \in \mathcal{H}_x} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle$, $(y', h') \in \mathcal{Y}_x \times \mathcal{H}_x$ and $p \in \mathcal{P}_x$. If $n \leq |\mathcal{P}_x|/(4b^2)$ for all $(x, y) \in S$, then any arbitrary proposal distribution $R(\boldsymbol{w}, x)$ fulfills Assumption 2.4.2.*

Proof. Let $\Delta \equiv \boldsymbol{\Phi}(x, y, h^*) - \boldsymbol{\Phi}(x, y', h')$. By assumption $|\Delta_p| \leq b/|\mathcal{P}_x|$, for all $p \in \mathcal{P}_x$. Therefore:

$$\begin{aligned}
\left\| \mathop{\mathbb{E}}_{(y',h') \sim R(\boldsymbol{w},x)} [\Delta] \right\|_2 &= \sqrt{\sum_{p \in \mathcal{P}_x} \mathop{\mathbb{E}}_{(y',h') \sim R(\boldsymbol{w},x)} \left[\Delta_p\right]^2} \\
&\leq \sqrt{\sum_{p \in \mathcal{P}_x} \mathop{\mathbb{E}}_{(y',h') \sim R(\boldsymbol{w},x)} \left[|\Delta_p|\right]^2} \\
&\leq \sqrt{|\mathcal{P}_x| \left(\frac{b}{|\mathcal{P}_x|}\right)^2} \\
&= b/\sqrt{|\mathcal{P}_x|}
\end{aligned}$$

Finally, since we assume that $n \leq |\mathcal{P}_x|/(4b^2)$, we have $b/\sqrt{|\mathcal{P}_x|} \leq 1/(2\sqrt{n})$ and we prove our claim. □

2.5.3 Examples for Assumption 2.4.3

In the case of modeling without latent variables, Zhang, Lei, Barzilay, and Jaakkola [7] and Zhang, Li, Barzilay, and Darwish [8] presented an algorithm for directed spanning trees in the context of dependency parsing in natural language processing. Later, Honorio and Jaakkola [23] extended the previous algorithm to any structure with computationally efficient local changes, which includes directed acyclic graphs (traversed in post-order) and cardinality-constrained sets. Next, we generalize Algorithm 2 in [23] by including latent variables.

Algorithm 1 Procedure for sampling a structured output $(y', h') \in \mathcal{Y}_x \times \mathcal{H}_x$ from a greedy local proposal distribution $R(\boldsymbol{w}, x)$

1: **Input:** parameter $\boldsymbol{w} \in \mathcal{W}$, observed input $x \in \mathcal{X}$
2: Draw uniformly at random a structured output $(\hat{y}, \hat{h}) \in \mathcal{Y}_x \times \mathcal{H}_x$
3: **repeat**
4: Make a local change to $(\hat{y}, \hat{h})$ in order to increase $\langle \boldsymbol{\Phi}(x, \hat{y}, \hat{h}), \boldsymbol{w} \rangle$
5: **until** no refinement in last iteration
6: **Output:** structured output and latent variable $(y', h') \leftarrow (\hat{y}, \hat{h})$

The algorithm above has the following property:

Claim 2.5.5 (Sampling for any type of structured output and latent variable)**.** *Algorithm 1 fulfills Assumption 2.4.3.*

Proof. Algorithm 1 depends solely on the linear ordering induced by the parameter $\boldsymbol{w}$ and the mapping $\boldsymbol{\Phi}(x, \cdot)$. That is, at any point in time, Algorithm 1 executes comparisons of the form $\langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle > \langle \boldsymbol{\Phi}(x, \hat{y}, \hat{h}), \boldsymbol{w} \rangle$ for any two pair of structured outputs and latent variables (y, h) and $(\hat{y}, \hat{h})$. □

2.6 Experiments

In this section, we illustrate the use of our approach by using the formulation in eq.(2.6). The goal of the synthetic experiments is to show the improvement in prediction results and runtime of our method. While the goal of the real-world experiment is to show the usability of our method in practice.

2.6.1 Synthetic Experiments

We present experimental results for directed spanning trees, directed acyclic graphs and cardinality-constrained sets. We performed 30 repetitions of the following procedure. We generated a ground truth parameter $\boldsymbol{w}^*$ with independent zero-mean and unit-variance Gaussian entries. Then, we generated a training set of $n = 100$ samples. Our mapping $\boldsymbol{\Phi}(x, y, h)$ is as follows. For every pair of possible edges/elements i and j, we define $\boldsymbol{\Phi}_{ij}(x, y, h) = \mathbb{1}\Big[(h_{ij} \text{ xor } x_{ij}) \text{ and } i \in y \text{ and } j \in y\Big]$. In order to generate each training sample $(x, y) \in S$, we generated a random vector x with independent Bernoulli entries, each with equal probability of being 1 or 0. The latent space $\mathcal{H}$ is the set of binary strings with two entries being 1, where these two entries share a common edge or element, i.e., $h_{ij} = h_{ik} = 1, \forall\ i, j, k$. To the best of our knowledge there is no efficient way to *exactly* compute the maximization in the margin m under this latent space. Thus, we define $\widetilde{\mathcal{H}}$ (relaxed set) as the set of all binary strings with exactly *two* entries being 1. We then can efficiently compute the margin $\tilde{m}$ by a greedy approach since our feature vector is constructed using linear operators. After

generating x, we set $(y, h) = f_{\boldsymbol{w}^*}(x)$. That is, we solved eq.(2.1) in order to produce the structured output y, and disregard h.

We replaced the discontinuous 0/1 loss $\mathbb{1}[z \geq 0]$ with the convex hinge loss $\max(0, 1 + z)$, as it is customary. Note however, that even by using the hinge loss, the objective functions in eq.(2.4), eq.(2.5) and in eq.(2.6) are still non-convex with respect to $\boldsymbol{w}$. This is due to the maximization over the latent space in the definition of the margin. We used $\lambda = 1/n$ as suggested by Theorems 2.3.1 and 2.4.4, and we performed 30 iterations of the subgradient descent method with a decaying step size $1/\sqrt{t}$ for iteration t. For sampling random structured outputs and latent variables in eq.(2.6), we implemented Algorithm 1 for directed spanning trees, directed acyclic graphs and cardinality-constrained sets. We performed the local changes in Algorithm 1 as follows. Given a pair $(\hat{y}, \hat{h})$, making a local change to $(\hat{y}, \hat{h})$ consists on iterating through all pairs (y', h') where $\hat{y}$ and y' differ only in one edge/element, and where the single entries in $\hat{h}$ and h' are contiguous. Finally, we used $\beta = 0.67$ for directed spanning trees, $\beta = 0.84$ for directed acyclic graphs, and $\beta = 0.5$ for cardinality-constrained sets, as prescribed by the examples given in Section 2.5.

We compared three training methods: the maximum loss over *all* possible structured outputs and latent variables with slack re-scaling as in eq.(2.5). We also evaluated the maximum loss over *random* structured outputs and latent variables, using the original latent space, as well as, the superset relaxation as in eq.(2.6). We considered directed spanning trees of 4 nodes, directed acyclic graphs of 4 nodes and 2 parents per node, and sets of 3 elements chosen from 9 possible elements. After training, for inference on an independent test set, we used eq.(2.1) for the maximum loss over *all* possible structured outputs and latent variables. For the maximum loss over random structured outputs and latent variables, we use the following *approximate* inference approach:

$$\tilde{f}_{\boldsymbol{w}}(x) \equiv \underset{(y,h) \in T(\boldsymbol{w}, x)}{\arg\max} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle. \tag{2.8}$$

Note that we used small structures and latent spaces in order to compare to exact learning, i.e., going through all possible structures as in eq.(2.5) and eq.(2.4). Bigger structures would result in exponential number of structures, making exact methods intractable to compare

against our method. For purposes of testing, we tried cardinality constrained sets of 4 elements out of 100 (note that in this case $|\mathcal{Y}| \approx 10^8$, $|\mathcal{H}| \approx 10^{16}$) and training took only 11 minutes under our approach.

Table 2.1 shows the runtime, the training distortion as well as the test distortion in an independently generated set of 100 samples. In the different study cases, the maximum loss over *random* structured outputs and latent variables obtains similar test performance than the maximum loss over *all* possible structured outputs and latent variables. However, note that our method is considerable faster.

2.6.2 Image Matching

We illustrate our approach for image matching on video frames from the Buffy Stickmen dataset (http://www.robots.ox.ac.uk/~vgg/data/stickmen/). The goal of the experiment is to match the keypoints representing different body parts between two images. Each frame contains 18 keypoints representing different parts of the body. From a total of 187 image pairs (from different episodes and people), we randomly selected 120 pairs for training and the remaining 67 pairs for testing. We performed a total of 30 repetitions. Ground truth keypoint matching is provided in the dataset.

Following the experiments of Gane, Hazan, and Jaakkola [41], and Volkovs and Zemel [42], we represent the matching as a permutation of keypoints. Let $x = (I, I')$ be a pair of images, and let y be a permutation of $\{1, \ldots, 18\}$. We model the latent variable h as a $\mathbb{R}^{2\times 2}$ matrix representing an affine transformation of a keypoint, where $h_{11}, h_{22} \in \{0.8, 1, 1.2\}$, and $h_{12}, h_{21} \in \{-0.2, 0, 0.2\}$. Our mapping $\boldsymbol{\Phi}(x, y, h)$ uses SIFT features, and the distance between coordinates after using h.

The authors in [41, 42] did not use latent variables, and considered the mapping $\boldsymbol{\Phi}(x, y) = \frac{1}{18}\sum_{i=1}^{18}(\boldsymbol{\Psi}(I, i) - \boldsymbol{\Psi}(I', y_i))^2$, where $\boldsymbol{\Psi}(I, k) \in \mathbb{R}^{128}$ are the SIFT descriptors at scale 5 evaluated at keypoint k. We properly centered the coordinates independently on each frame to avoid modeling translations in h. We use the mapping $\boldsymbol{\Phi}(x, y, h) = (\boldsymbol{\Phi}(x, y), \frac{1}{18}\sum_{i=1}^{18}\|c(I, i)\times h - c(I', y_i)\|_2^2)$, where $c(I, k) \in \mathbb{R}^2$ are the coordinates of keypoint k. Intuitively, we are adding

one extra feature that summarizes the change in rotation and scaling of the keypoints, i.e., $\mathbf{\Phi}(x, y, h) \in \mathbb{R}^{129}$.

The learning is performed using the randomized formulation as in eq.(2.6), and using local changes as in Algorithm 1 for sampling from the proposal distribution. As in the synthetic experiments, we also replaced the discontinuous 0/1 loss $\mathbb{1}[z \geq 0]$ with the convex hinge loss $\max(0, 1 + z)$, and followed the local changes in Algorithm 1 for sampling from the proposal distribution. The neighborhoods of the structures and latent variables were defined as follow: for a given permutation y, we considered y' to be its neighbor, and vice versa, if they have only two mismatched entries. Similarly, for a given h, we considered h' to be its neighbor, and vice versa, if they have only one different entry.

We used the distortion function and $\beta = 2/3$ as prescribed by Claim 2.5.2. After learning, for a given x from the test set, we performed 100 iterations of randomized inference as in eq.(2.8). We obtained an average error of 0.3878 (6.98 incorrectly matched keypoints) in the test set, which is an improvement to the values of 8.47 for maximum-a-posteriori perturbations and 8.69 for max-margin, as reported in [41]. Finally, we show an example from the test set in Figure 2.1.

Figure 2.1. Image matching on the Buffy Stickmen dataset, predicted by our randomized approach with latent variables. The problem is challenging since the dataset contains different episodes and people.

2.7 Discussion

2.7.1 Inference on Test Data

The upper bound in Theorem 2.4.4 holds simultaneously for all parameters $\boldsymbol{w} \in \mathcal{W}$. Therefore, our result implies that after learning the optimal parameter $\hat{\boldsymbol{w}} \in \mathcal{W}$ in eq.(2.6) from *training* data, we can bound the decoder distortion when performing *exact* inference on *test* data. More formally, Theorem 2.4.4 can be additionally invoked for a *test* set S', also with probability at least $1 - \delta$. Thus, under the same setting as of Theorem 2.4.4, the Gibbs decoder distortion is upper-bounded with probability at least $1 - 2\delta$ over the choice of S and S'. In this chapter, we focused on learning the model parameters.

2.7.2 A Non-Convex Formulation

As mentioned in Section 2.1, all formulations with latent variables (eq.(2.4), eq.(2.5), and eq.(2.6)) are non-convex objectives. The motivation to use the margin re-scaling approach in the work of Yu and Joachims [29] is that the non-convex objective leads to a difference of two convex functions, which allows the use of CCCP [31]. In the case of models without latent variables, Sarawagi and Gupta [43] propose a method to reduce the problem of slack re-scaling to a series of modified margin re-scaling problems. However, there are two main caveats in their approach. First, the optimization is only heuristic, that is, it is not guaranteed to solve the slack rescaling objective exactly. Second, their method is specific to the cutting plane training algorithm and does not easily extend to stochastic algorithms. Choi, Meshi, and Srebro [44] propose efficient methods for finding the most-violating-label in a slack re-scaling formulation, given an oracle that returns the most-violating-label in a (slightly modified) margin re-scaling formulation. However, in the case of latent models, it is still unclear if this sort of reductions are possible for the slack re-scaling approach because of the maximization in the margin with respect to the latent space.

We also note that one way to make the objective in eq.(2.5) convex is to replace the maximization in the margin by the latent variable $\hat{h}$. However, this not only results in a

looser upper bound of the Gibbs decoder distortion but also under performs with respect to the methods mentioned in this chapter.

2.7.3 Randomizing the Latent Space

We note that in the definition of the margin, there is a maximization over the latent space $\mathcal{H}$. In this chapter, we sample structured outputs and latent variables from some proposal distribution and these samples are used in the outer maximization in eq.(2.6). While sampling latent variables from some proposal distribution in the maximization of the margin might be computationally appealing, the main issue is that this will lead to a looser upper bound of the Gibbs decoder distortion.

2.8 Summary

We focused on the learning aspects of structured prediction problems using latent variables. We first extended the work of McAllester [24] by including latent variables, and showed that the non-convex formulation using the slack re-scaling approach with latent variables is related to a tight upper bound of the *Gibbs decoder distortion*. This motivates the apparent need of the non-convexity in different formulations using latent variables (e.g., [29, 37]). Second, we provided a tighter upper bound of the Gibbs decoder distortion by randomizing the search space of the optimization problem. That is, instead of having a formulation over all possible structures and latent variables (usually exponential in size), we proposed a formulation that uses i.i.d. samples coming from some proposal distribution.

Our approach is computationally appealing in cases where the margin can be computed in poly-time since it would lead to a fully polynomial time evaluation of the formulation. We provided a method to obtain an upper bound that is logarithmic in the size of the latent space as the use of standard Rademacher arguments (e.g., [23]) would lead to a prohibitive upper bound that is proportional to the size of the latent space. Finally, we provided experimental results in synthetic data and in a computer vision application, where we obtained competitive results in the average test error with respect to the values reported in [41].

Table 2.1. Average over 30 repetitions, and standard error at 95% confidence level. *All (LSSVM)* indicates the use of exact learning and exact inference. *Rand* and *Rand/All* indicate use of randomized learning, and randomized and exact inference respectively. The mark *(S)* indicates the use of superset $\widetilde{\mathcal{H}}$ in the calculation of the margin. Rand/All obtains a similar or sightly better test performance than All in the different study cases. Note that the runtime for learning using the randomized approach is much less than exact learning, while still having a good test performance.

Problem	**Method**	**Training runtime**	**Training distortion**	**Test runtime**	**Test distortion**
Directed spanning trees	All (LSSVM)	1000 ± 15	$8.4\% \pm 1.4\%$	18.9 ± 0.1	$8.2\% \pm 1.3\%$
	Rand (S)	$\mathbf{44 \pm 1}$	$22\% \pm 2.2\%$	0.92 ± 0	$22\% \pm 1.9\%$
	Rand/All (S)			19 ± 0.1	$8.2\% \pm 1.3\%$
	Rand	126 ± 5	$23\% \pm 3.0\%$	3 ± 0.4	$24\% \pm 3.2\%$
	Rand/All			17 ± 0.8	$8.2\% \pm 1.4\%$
Directed acyclic graphs	All (LSSVM)	1000 ± 21	$17\% \pm 1.7\%$	19 ± 0.2	$21\% \pm 2.4\%$
	Rand (S)	$\mathbf{63 \pm 0}$	$24\% \pm 1.5\%$	1.5 ± 0	$28\% \pm 1.9\%$
	Rand/All (S)			19 ± 0.2	$20\% \pm 1.9\%$
	Rand	353 ± 5	$21\% \pm 1.1\%$	8 ± 1	$25\% \pm 1.4\%$
	Rand/All			15 ± 0.2	$19\% \pm 1.6\%$
Cardinality constrained sets	All (LSSVM)	1000 ± 5	$6.3\% \pm 1.0\%$	19.5 ± 0.1	$6\% \pm 1.2\%$
	Rand (S)	$\mathbf{75 \pm 0}$	$18\% \pm 1.8\%$	1.7 ± 0	$18\% \pm 1.8\%$
	Rand/All (S)			19.5 ± 0.1	$6\% \pm 1.3\%$
	Rand	182 ± 3	$15\% \pm 3.2\%$	3.1 ± 1	$17\% \pm 1.2\%$
	Rand/All			19.4 ± 0.1	$6\% \pm 2.2\%$

3. THE FUNDAMENTAL LIMITS OF STRUCTURED PREDICTION

A common approach to structured prediction is to exploit local features to infer the global structure. For instance, one could include a feature that encourages two individuals of a social network to be assigned to different clusters whenever there is a strong disagreement in opinions about a particular subject. Then, one can define a posterior distribution over the set of possible labelings conditioned on the input. The output structure and corresponding loss function make these problems significantly different from the (unstructured) binary or multiclass classification problems extensively studied in learning theory.

Some classical algorithms for learning the parameters of the model include conditional random fields [45], structured support vector machines [30, 34, 36], kernel-regression algorithm [46], search-based structured prediction [47]. More recently, deep learning algorithms have been developed for specific tasks such as image annotation [48], part-of-speech-tagging [49, 50], and machine translation [51].

However, in contrast to the various algorithms proposed throughout the years, there have been only a small handful of studies devoted to the theoretical understanding of structured prediction. From the scarce theory literature, the most studied property of structure prediction models has been the generalization error. Cortes, Kuznetsov, and Mohri [52], Collins [53], and Taskar, Guestrin, and Koller [54] provided learning guarantees that hold primarily for losses such as the Hamming loss. Cortes, Kuznetsov, Mohri, and Yang [33] presented generalization bounds for more general losses and scoring functions based on factor graphs. Similar to [33], in this chapter we also study factor graph models, with the difference that we focus on lower bounds and not upper bounds. Bello and Honorio [16], Honorio and Jaakkola [23], McAllester [24], and Ghoshal and Honorio [55] provided PAC-Bayesian guarantees for arbitrary losses through the analysis of randomized algorithms using count-based hypotheses.

Results on lower bounding the sample complexity for structure prediction is scarcer even for specific classes of predictors. Information-theoretic bounds have been studied in the context of binary graphical models [56, 57] and Gaussian Markov random fields [58]. Never-

theless, the aforementioned works apply to the modeling of the input x and not the prediction of y from x.

In this chapter, our main contribution consists of characterizing the necessary sample complexity for learning factor graph models in the context of structured prediction, which to the best of our knowledge, we are the first to find such characterization. Specifically, we show that the finiteness of the PAIR-dimension (see Definition 3.2.2) is necessary for learning. We further show the connection of the PAIR-dimension to the VC-dimension [59], which could allow us to compute the PAIR-dimension from existing results on VC-dimension.

3.1 Preliminaries

Let $\mathcal{X}$ denote the input space and $\mathcal{Y}$ the output space. In structured prediction, the output space usually consists of a large (e.g., exponential) set of discrete objects admitting some possibly overlapping structure. For example, sequences, graphs, images, parse trees, etc. Thus, we consider the output space $\mathcal{Y}$ to be decomposable into l substructures: $\mathcal{Y} = \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_l$. Here, $\mathcal{Y}_i$ is the set of possible labels that can be assigned to substructure i. For example, in a webpage collective classification task [60], each $\mathcal{Y}_i$ is a webpage label, whereas $\mathcal{Y}$ is a joint label for an entire website. In this work we assume that $\mathcal{Y}_i \in \{0, 1\}$, that is, $|\mathcal{Y}_i| = 2$ for all i. In this case, the number of possible assignments to $\mathcal{Y}$ is exponential in the number of substructures l, i.e., $|\mathcal{Y}| = 2^l$.

3.1.1 The Hamming Loss

In order to measure the success of a prediction, we use the Hamming loss throughout this work. Specifically, for two outputs $y, y' \in \mathcal{Y}$, with $y = (y_1, \ldots, y_l)$ and $y' = (y'_1, \ldots, y'_l)$, the Hamming loss, L_H, is defined as:

$$L_H(y, y') = \sum_{i=1}^{l} \mathbb{1}\left[y_i \neq y'_i\right].$$

The use of Hamming loss in this work is motivated for being widely used in structured prediction problems, for instance, in image segmentation one may count the number of

pixels that are incorrectly assigned as foreground/background; in graphs, one may count the number of different edges between the prediction and the true label. For this reason, Globerson, Roughgarden, Sontag, and Yildirim [61] also focused on the Hamming loss for analyzing approximate inference.

3.1.2 Factor Graphs and Scoring Functions

We adopt a common approach in structured prediction where predictions are based on a scoring function $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$, where $\mathbb{R}_+$ denotes the set of non-negative real numbers. Let $\mathcal{F}$ be a family of scoring functions. For any $f \in \mathcal{F}$, we denote by $\mathsf{f}(x)$ the predictor induced by the scoring function f: for any $x \in \mathcal{X}$,

$$\mathsf{f}(x) \;=\; \arg\max_{y \in \mathcal{Y}} f(x, y).$$

We denote the class of induced predictors by F. Furthermore, we assume that each score function $f \in \mathcal{F}$ can be decomposed as a sum, as is standard in structured prediction. We consider the most general case for such decompositions through the notion of factor graphs, motivated also in [33]. A factor graph G is a bipartite graph, and is represented as a tuple $G = (V, \Phi, E)$, where V is a set of variable nodes, Φ a set of factor nodes, and E a set of undirected edges between a variable node and a factor node. In our context, V can be identified with the set of substructure indices, that is $V = \{1, \ldots, l\}$. We further assume that G is connected.

For any factor node $\phi \in \Phi$, denote by $\mathrm{Scope}(\phi) \subseteq V$ the set of variable nodes connected to ϕ via an edge and define $\mathcal{Y}_\phi$ as the substructure set cross-product $\mathcal{Y}_\phi = \bigtimes_{i \in \mathrm{Scope}(\phi)} \mathcal{Y}_i$. Then, f decomposes as a sum of functions f_ϕ , each taking as argument an element of the input space $x \in \mathcal{X}$ and an element of $\mathcal{Y}_\phi$, $y_\phi \in \mathcal{Y}_\phi$:

$$f(x, y) = \sum_{\phi \in \Phi} f_\phi(x, y_\phi).$$

We further use $\mathcal{F}(G)$ to denote the set of scoring functions that are decomposable with respect to the graph G, and use $\mathsf{F}(G)$ to denote the set of predictors induced by $\mathcal{F}(G)$. Note

also that while all $f \in \mathcal{F}(G)$ decompose with respect to same graph G, the scoring functions f_ϕ and f'_ϕ are allowed to be different for any $\phi \in \Phi$, $f, f' \in \mathcal{F}(G)$. For instance, f_ϕ can be a linear function, while f'_ϕ can be a kernel-based function. Figure 3.1 shows different examples of factor graphs.

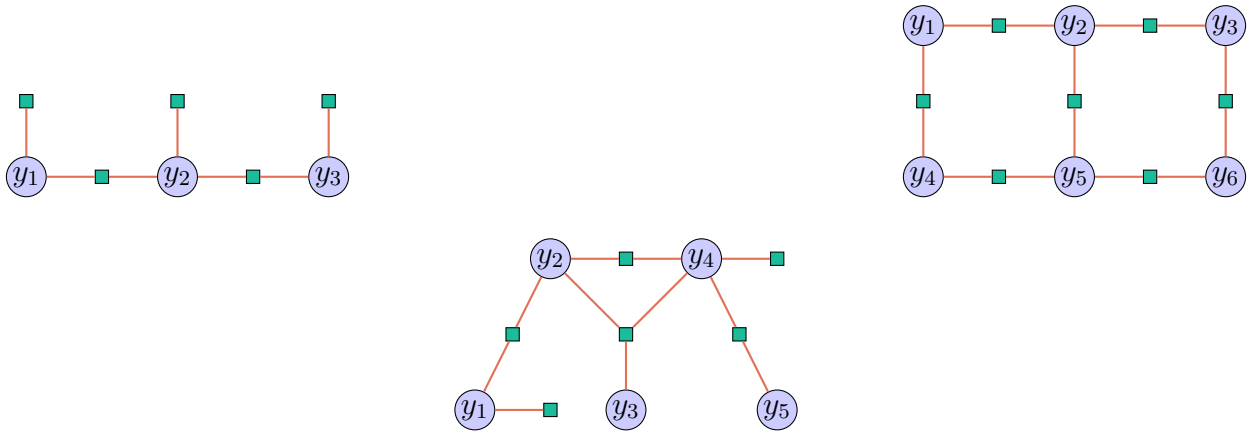


Figure 3.1. Three examples of factor graphs. *(Left)* Tree-structured factor graph. *(Center)* Arbitrary factor graph with decomposition: $f(x, y) = f_{\phi_1}(x, y_1) + f_{\phi_4}(x, y_4) + f_{\phi_{12}}(x, y_1, y_2) + f_{\phi_{45}}(x, y_4, y_5) + f_{\phi_{24}}(x, y_2, y_4) + f_{\phi_{234}}(x, y_2, y_3, y_4)$. *(Right)* Grid-structured factor graph.

3.1.3 Learning

We receive a training set $S = ((x_1, y_1), \ldots, (x_m, y_m))$ of m i.i.d. samples drawn according to some distribution P over $\mathcal{X} \times \mathcal{Y}$. We denote by $R_P(\mathsf{f})$ the *expected Hamming loss* and by $R_S(\mathsf{f})$ the *empirical Hamming loss* of f:

$$R_P(\mathsf{f}) = \mathop{\mathbb{E}}_{(x,y)\sim P}[L_H(\mathsf{f}(x), y)], \tag{3.1}$$

$$R_S(\mathsf{f}) = \frac{1}{m} \sum_{(x,y)\in S} L_H(\mathsf{f}(x), y). \tag{3.2}$$

Our learning scenario consists of using the sample S to select a scoring function $f \in \mathcal{F}(G)$ with small expected Hamming loss $R_P(\mathsf{f})$.

Next, we introduce the definition of *Bayes-Hamming loss*, which in words is the minimum attainable expected Hamming loss by any predictor.

Definition 3.1.1 (Bayes-Hamming loss)**.** *For any given distribution P over $\mathcal{X} \times \mathcal{Y}$, the Bayes-Hamming loss is defined as the minimum achievable expected Hamming loss among all possible predictors $\mathsf{f} : \mathcal{X} \to \mathcal{Y}$. That is, $R^* = \min_{\mathsf{f}} R_P(\mathsf{f})$.*

Then the *Bayes-Hamming predictor*, f^*, is defined as the function that achieves the Bayes-Hamming loss, that is, $R_P(\mathsf{f}^*) = R^*$.

The following proposition shows how the Bayes-Hamming predictor makes its decision with respect to the Hamming loss.

Proposition 3.1.1. *For any given distribution P over $\mathcal{X} \times \mathcal{Y}$, the Bayes-Hamming predictor f^* is:*

$$(\mathsf{f}^*(x))_i = \begin{cases} 1 & \text{if } \eta_i(x) \geq 1/2, \\ 0 & \text{otherwise,} \end{cases}$$

where $\eta_i(x) = \mathbb{P}[y_i = 1|x]$ is the marginal probability of substructure y_i.

Proof. Recall that $\eta_i(x) = \mathbb{P}[y_i = 1|x]$. From eq.(3.1) and Definition 3.1.1, the Bayes-Hamming predictor f^* minimizes the following expression (with respect to f).

$$\begin{aligned} R_P(\mathsf{f}) &= \mathop{\mathbb{E}}_{(x,y)\sim P}[L_H(\mathsf{f}(x), y)] \\ &= \mathop{\mathbb{E}}_{(x,y)\sim P}\left[\sum_{i=1}^{l} \mathbb{1}[(\mathsf{f}(x))_i \neq y_i]\right] \\ &= \sum_{i=1}^{l} \mathop{\mathbb{E}}_{(x,y)\sim P}\left[\mathbb{1}[(\mathsf{f}(x))_i \neq y_i]\right] \\ &= \sum_{i=1}^{l} \mathop{\mathbb{E}}_{x}\left[\mathbb{P}[y_i = 1|x](1 - (\mathsf{f}(x))_i) + (1 - \mathbb{P}[y_i = 1|x])(\mathsf{f}(x))_i\right] \\ &= \sum_{i=1}^{l} \mathop{\mathbb{E}}_{x}\left[\eta_i(x)(1 - (\mathsf{f}(x))_i) + (1 - \eta_i(x))(\mathsf{f}(x))_i\right]. \end{aligned}$$

In order to minimize the above expression, for any x we choose $(\mathsf{f}(x))_i = 1$ if $\eta_i(x) \geq 1/2$, and $(\mathsf{f}(x))_i = 0$ otherwise. □

We emphasize that the above proposition considers the Hamming loss, L_H, as defined in Section 3.1.1. For other types of loss functions, the Bayes predictor can have different optimal decisions.

3.1.4 A Review of the General Minimax Risk Framework

In this section we briefly review the minimax framework in the context of general statistical problems. The minimax framework consists of a well defined objective that aims to shed light about the optimality of algorithms and has been widely used in statistics and machine learning [62, 63]. The standard minimax risk considers a family of distributions $\mathcal{Q}$ over a sample space $\mathcal{Z}$, and a function $\theta : \mathcal{Q} \to \Theta$ defined on $\mathcal{Q}$, that is, a mapping $Q \mapsto \theta(Q)$. Here we call $\theta(Q)$ parameter of the distribution Q. We aim to estimate the parameter $\theta(Q)$ based on a sequence of m i.i.d. observations $Z = (z_1, \ldots, z_m)$ drawn from the (unknown) distribution Q, that is, $Z \in \mathcal{Z}^m$. To evaluate the quality of an estimator $\widehat{\theta}$, we let $\rho : \Theta \times \Theta \to \mathbb{R}_+$ denote a semi-metric on the space Θ, which we use to measure the error of an estimator $\widehat{\theta}$ with respect to the parameter $\theta(Q)$. For a distribution $Q \in \mathcal{Q}$ and for a given estimator $\widehat{\theta} : \mathcal{Z}^m \to \Theta$, we assess the quality of the estimate $\widehat{\theta}(Z)$ in terms of the (expected) risk:

$$\mathbb{E}_{Z \sim Q^m} \left[\rho(\widehat{\theta}(Z), \theta(Q)) \right].$$

A common approach, first suggested by [64], for choosing an estimator $\widehat{\theta}$ is to select the one that minimizes the maximum risk, that is,

$$\sup_{Q \in \mathcal{Q}} \mathbb{E}_{Z \sim Q^m} \left[\rho(\widehat{\theta}(Z), \theta(Q)) \right].$$

An optimal estimator for this semi-metric then gives the minimax risk, which is defined as:

$$\mathfrak{M}_m(\mathcal{Q}, \rho) := \inf_{\widehat{\theta}} \sup_{Q \in \mathcal{Q}} \mathbb{E}_{Z \sim Q^m} \left[\rho(\widehat{\theta}(Z), \theta(Q)) \right],$$

where we take the supremum (worst-case) over distributions $Q \in \mathcal{Q}$, and the infimum is taken over all estimators $\widehat{\theta}$.

3.1.5 Minimax Risk in Structured Prediction

We now apply the framework above to our context and study a specialized notion of risk appropriate for prediction problems. In this setting, we aim to estimate a scoring function $f \in \mathcal{F}(G)$ by using samples from a distribution P. For any sample $(x, y) \sim P$, we will measure the quality of our estimation, f, by comparing the prediction $\mathsf{f}(x)$ to the structure y drawn from P through the Hamming loss. By taking expectation, we obtain the expected risk or expected Hamming loss, $R_P(\mathsf{f})$, defined in eq.(3.1). We then compare this risk to the best possible Hamming loss, i.e., the Bayes-Hamming loss (Definition 3.1.1). That is, we assume that at least one scoring function in $\mathcal{F}(G)$ achieves the Bayes-Hamming loss. Finally, recall that $S \in (\mathcal{X} \times \mathcal{Y})^m$ is the training set consisting of m i.i.d. samples drawn from P. Thus, we arrive to the following *minimax excess risk*:

$$\mathfrak{M}_m(\mathcal{P}) = \inf_{\mathcal{A}} \sup_{P \in \mathcal{P}} \mathop{\mathbb{E}}_{S \sim P^m} \left[R_P(\mathcal{A}(S)) - R_P(\mathsf{f}^*) \right], \tag{3.3}$$

where f^* is the induced predictor by the scoring function $f^* = \arg\min_{f \in \mathcal{F}(G)} R_P(\mathsf{f})$[1], and $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^m \to \mathcal{F}(G)$ is any algorithm that returns a predictor given m training samples from P. Moreover, $\mathcal{P}$ defines a family of distributions over $\mathcal{X} \times \mathcal{Y}$.

Intuitively speaking, for a fixed distribution $P \in \mathcal{P}$, the quantity $\mathfrak{M}_m(\mathcal{P})$ represents the minimum expected excess loss achievable by any algorithm with respect to the factor graph G. Then $\mathfrak{M}_m(\mathcal{P})$ looks into the distribution that attains the worst expected excess loss.

3.2 An Information-Theoretic Lower Bound for Structured Prediction

We are interested on finding a lower bound to the minimax risk (3.3) presented in Section 3.1.5. By doing this, we characterize the necessary number of samples to have any hope in achieving learning.

Before presenting our main result for this chapter, we introduce a new type of dimension that will show up in our lower bound and will help to characterize learnability. Note that it is known that different notions of dimension of predictor classes help to characterize learnability

[1]↑Recall that f denotes the induced predictor by $f \in \mathcal{F}(G)$.

in certain prediction problems. For example, in binary classification with the 0/1-loss, the finiteness of the VC dimension [59] is necessary for learning [65]. For multiclass classification, it was shown that the finiteness of the Natarajan dimension is necessary for learning [66]. General notions of dimensions for multiclass classification has also been study in [67].

For a given predictor class $\mathcal{G} \subseteq \{\mathsf{g} \mid \mathsf{g} : \mathcal{X} \to \{0,1\}^2\}$, and dataset S of m samples, we use the following shorthand notation:

$$\mathcal{G}(S) = \{(\mathsf{g}(x_1), \ldots, \mathsf{g}(x_m)) \in \{0,1\}^{m\times 2} \mid \mathsf{g} \in \mathcal{G}\}.$$

That is, $\mathcal{G}(S)$ contains all the matrices in $\{0,1\}^{m\times 2}$ that can be produced by applying all functions in $\mathcal{G}$ to the dataset S. Next we define PAIR-shattering.

Definition 3.2.1 (PAIR-shattering)**.** *A function class, $\mathcal{G}$, PAIR-shatters a finite set S of m samples if $\mathcal{G}(S)$ produces all possible binary matrices in $\{0,1\}^{m\times 2}$. That is, $|\mathcal{G}(S)| = 2^{2m}$.*

Definition 3.2.2 (PAIR-dimension)**.** *The PAIR-dimension of a function class $\mathcal{G}$, denoted PAIRDIM($\mathcal{G}$), is the maximal size of a set S that can be PAIR-shattered by $\mathcal{G}$. If $\mathcal{G}$ can shatter sets of arbitrarily large size we say that $\mathcal{G}$ has infinite PAIR-dimension.*

The above dimension applies to predictors with output in $\{0,1\}^2$. Next, we define the MAX-PAIR-dimension for classes of predictors $\mathcal{H} \subseteq \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}^l\}$.

Definition 3.2.3 (MAX-PAIR-dimension)**.** *For a predictor class $\mathcal{H} \subseteq \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}^l\}$, the MAX-PAIR-dimension of $\mathcal{H}$, denoted as MAX-PAIRDIM($\mathcal{H}$), is defined as:*

$$\text{MAX-PAIRDIM}(\mathcal{H}) = \max_{\substack{u,v\in\{1,\ldots,l\}\\ u\neq v}} \text{PAIRDIM}(\mathcal{H}_{u,v}),$$

where $\mathcal{H}_{u,v} = \{\mathsf{h}_{u,v} \mid \mathsf{h} \in \mathcal{H}, \mathsf{h}_{u,v} : \mathcal{X} \to \{0,1\}^2,\ \mathsf{h}_{u,v}(x) = (\mathsf{h}(x)_u, \mathsf{h}(x)_v)\}$, that is, the predictor $\mathsf{h}_{u,v}$ only takes into account the output of h at positions u and v, and becomes a mapping from $\mathcal{X}$ to $\{0,1\}^2$.

We remark that Definition 3.2.3 is stated for general classes of predictors with output in $\{0,1\}^l$. However, in our context we consider predictors induced by scoring functions based

on factor graphs. That is, for a predictor $\mathsf{f} : \mathcal{X} \to \{0,1\}^l$ induced by the scoring function f, we will create predictors with output in $\{0,1\}^2$ as follows. Let

$$f^{(0)}_{u,v}(x, y_u, y_v) \stackrel{\text{def}}{=} f(x, (0, \ldots, 0, y_u, 0, \ldots, 0, y_v, 0, \ldots, 0))$$

denote the scoring function $f(x, y)$ with $y_i = 0$ for all $i \in \{1, \ldots, l\} \setminus \{u, v\}$. Then, let

$$\mathsf{f}^{(0)}_{u,v}(x) = \arg\max_{y_u, y_v} f^{(0)}_{u,v}(x, y_u, y_v)$$

be the induced predictor by $f^{(0)}_{u,v}(x, y_u, y_v)$, i.e., the output of $\mathsf{f}^{(0)}_{u,v}(x)$ is in $\{0,1\}^2$.

Remark 3.2.1. *For a given factor graph $G = (V, \Phi, E)$ such that $T = \{(u, v) \mid u \neq v, \{u, v\} \subseteq \text{Scope}(\phi), \phi \in \Phi\}$, $\mathcal{F}^{(0)}_{u,v}(G) = \{f^{(0)}_{u,v} \mid f \in \mathcal{F}(G)\}$, and let $\mathsf{F}^{(0)}_{u,v}(G) = \{\mathsf{f}^{(0)}_{u,v} \mid f^{(0)}_{u,v} \in \mathcal{F}^{(0)}_{u,v}(G)\}$ denote the set of predictors induced by $\mathcal{F}^{(0)}_{u,v}(G)$. Then, the* MAX-PAIR*-dimension of a class of scoring functions is given by the* MAX-PAIR*-dimension of the class of predictors it induces, i.e.,*

$$\textsc{max-PairDim}(\mathsf{F}(G)) = \max_{(u,v) \in T} \textsc{PairDim}(\mathsf{F}^{(0)}_{u,v}(G)).$$

Next, we present our main result which provides a characterization on the necessary number of samples for learning.

Theorem 3.2.2. *Let $G = (V, \Phi, E)$ be a factor graph and let $\mathcal{F}(G)$ denote a class of scoring functions where each $f \in \mathcal{F}(G)$ decomposes according to G. Let $\mathsf{F}(G)$ be the induced class of predictors by $\mathcal{F}(G)$, where $\mathsf{f} : \mathcal{X} \to \{0,1\}^l$ for each $\mathsf{f} \in \mathsf{F}(G)$, and let $d =$ $\textsc{max-PairDim}(\mathsf{F}(G)) \geq 2$. Then, we have that for any $\gamma \in [0, 1/3]$ and any $m \geq d$:*

$$\mathfrak{M}_m(\mathcal{P}) \geq \frac{1}{81} \min\left(\frac{d-1}{\gamma m}, \sqrt{\frac{d-1}{m}}\right).$$

The proof of the theorem above can be found in Appendix B.1. As prescribed by Theorem 3.2.2, the MAX-PAIR-dimension needs to be finite in order for the predictor class to be learnable.

3.3 Relation of the Pair-Dimension to the VC-Dimension

In this section, we show a connection of our defined PAIR-dimension to the classical VC-dimension [59]. The following theorem shows that for a function class $\mathcal{G} \subseteq \{\mathsf{g} \mid \mathsf{g} : \mathcal{X} \to \{0,1\}^2\}$, the PAIR-dimension of $\mathcal{G}$ is related to the minimum VC-dimension of a subclass of functions derived from $\mathcal{G}$.

Theorem 3.3.1. *Let $\mathcal{G} \subseteq \{\mathsf{g} \mid \mathsf{g} : \mathcal{X} \to \{0,1\}^2\}$ be a function class. Let $\mathcal{H}_{11}, \mathcal{H}_{10}, \mathcal{H}_{01}$, $\mathcal{H}_{00} \subseteq \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}\}$ be four function classes defined as:*

$$\begin{aligned}
\mathcal{H}_{11} &= \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}, \mathsf{h}(x) = \mathsf{g}(x)_1 \mathsf{g}(x)_2, \mathsf{g} \in \mathcal{G}\},\\
\mathcal{H}_{10} &= \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}, \mathsf{h}(x) = \mathsf{g}(x)_1 (1 - \mathsf{g}(x)_2), \mathsf{g} \in \mathcal{G}\},\\
\mathcal{H}_{01} &= \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}, \mathsf{h}(x) = (1 - \mathsf{g}(x)_1) \mathsf{g}(x)_2, \mathsf{g} \in \mathcal{G}\},\\
\mathcal{H}_{00} &= \{\mathsf{h} \mid \mathsf{h} : \mathcal{X} \to \{0,1\}, \mathsf{h}(x) = (1 - \mathsf{g}(x)_1)(1 - \mathsf{g}(x)_2), \mathsf{g} \in \mathcal{G}\}.
\end{aligned}$$

Then, we have that $\textsc{PairDim}(\mathcal{G}) = \min_{i,j \in \{0,1\}} \textsc{VC-Dim}(\mathcal{H}_{ij})$.

Proof. Recall that for a dataset S of m samples, $\mathcal{G}(S) = \{(\mathsf{g}(x_1), \ldots, \mathsf{g}(x_m)) \in \{0,1\}^{m \times 2} \mid \mathsf{g} \in \mathcal{G}\}$. Similarly, define $\mathcal{H}_{ij}(S) = \{(\mathsf{h}(x_1), \ldots, \mathsf{h}(x_m)) \in \{0,1\}^m \mid \mathsf{h} \in \mathcal{H}_{ij}\}$ for all $i, j \in \{0,1\}$. Let $\textsc{PairDim}(\mathcal{G}) = d$.

There exists a dataset S of d samples such that $|\mathcal{G}(S)| = 2^{2d}$. Thus for all $i, j \in \{0,1\}$ we have $|\mathcal{H}_{ij}(S)| = 2^d$, which implies that for all $i, j \in \{0,1\}$ we have $\text{VC-}\textsc{Dim}(\mathcal{H}_{ij}) \geq d$. Therefore,

$$d \leq \min_{i,j \in \{0,1\}} \text{VC-}\textsc{Dim}(\mathcal{H}_{ij}).$$

Also, for any dataset S of $d+1$ samples we have $|\mathcal{G}(S)| < 2^{2(d+1)}$. Thus there exists $i, j \in \{0,1\}$ such that $|\mathcal{H}_{ij}(S)| < 2^{d+1}$, implying that there exists $i, j \in \{0,1\}$ such that $\text{VC-}\textsc{Dim}(\mathcal{H}_{ij}) < d+1$. Therefore,

$$\min_{i,j \in \{0,1\}} \text{VC-}\textsc{Dim}(\mathcal{H}_{ij}) < d+1.$$

From the above, $\min_{i,j \in \{0,1\}} \text{VC-}\textsc{Dim}(\mathcal{H}_{ij}) = d$. □

3.4 Summary

In this chapter, we studied the problem of finding the necessary number of samples for learning of scoring functions based on factor graphs in the context of structured prediction. Our work was based on the minimax framework, that is, in obtaining a lower bound to the minimax risk. We showed a lower bound that requires the MAX-PAIR-dimension (Definition 3.2.3) to be finite in order for a function class to be learnable. We also note that in the proof of Theorem 3.2.2, our choice of setting a value of zero to many y's was for clarity purposes. In principle, one can create such distributions by fixing y's to arbitrary values in $\{0,1\}^{l-2}$. This would result in a slightly different notion of dimension, which would take the maximum across the 2^{l-2} different values. However, our focus was on providing a clear guideline to obtain lower bounds in structured prediction, hence, we opted for simplicity. In addition, in Theorem 3.3.1, we showed the connection of the PAIR-dimension to the VC-dimension, for which there are several known results for different types of hypothesis classes.

An interesting future work is the analysis of tightness. For example, regarding tightness for linear classifiers, consider inputs $x \in \mathbb{R}^k$. We observe that our lower bound in Theorem 3.2.2 is tight with respect to k and m. Specifically, consider non-sparse linear classifiers as the scoring functions, Theorem 2 in [33] gives $\mathcal{O}(\sqrt{k/m})$. In this case, the PAIR-dimension is equal to the VC-dimension, and the latter is equal to k. Thus, we obtain a lower bound with rate $\sqrt{k/m}$ for some γ. Similarly, consider sparse linear classifiers as the scoring functions. Then, Theorem 2 of [33] gives $\mathcal{O}(\sqrt{\log k/m})$. In this case, the VC-dimension is $\mathcal{O}(\log k)$ [68], thus, we obtain a lower bound with rate $\sqrt{\log k/m}$ for some γ. However, an analysis for general functions remains open, where perhaps, one possible approach is to find an upper bound to the *factor graph Rademacher complexity* [33] in terms of the PAIR-dimension, similar in spirit to the known result of the VC-dimension being an upper bound to the classical Rademacher complexity (see for instance, [69]).

4. EXACT INFERENCE IN STRUCTURED PREDICTION

In this chapter, we focus on the inference problem and assume that the model parameters have already been learned. In the context of Markov random fields (MRFs), for an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, one is interested in finding a solution to the following inference problem:

$$\max_{\boldsymbol{y} \in \mathcal{M}^{|\mathcal{V}|}} \sum_{v \in \mathcal{V}, m \in \mathcal{M}} c_v(m) \mathbb{1}[y_v = m] + \sum_{\substack{(u,v) \in \mathcal{E} \\ s,t \in \mathcal{M}}} c_{u,v}(s,t) \mathbb{1}[y_u = s, y_v = t] \,, \tag{4.1}$$

where $\mathcal{M}$ is the set of possible labels, $c_v(m)$ is the cost of assigning label m to node v, and $c_{u,v}(s,t)$ is the cost of assigning s and t to the neighbors u, v respectively.[1] Similar inference problems arise in the context of statistical physics, sociology, community detection, average case analysis, and graph partitioning. Very few cases of the general MRF inference problem are known to be exactly solvable in polynomial time. For example, Chandrasekaran, Srebro, and Harsha [70] showed that problem (4.1) can be solved exactly in polynomial time for a graph $\mathcal{G}$ with low treewidth via the junction tree algorithm. While in the case of Ising models, Schraudolph and Kamenetsky [71] showed that the inference problem can also be solved exactly in polynomial time for planar graphs via perfect matchings. Finally, polynomial-time solvability can also stem from properties of the pairwise potential, under this view, the inference problem can be solved exactly in polynomial time via graph cuts for binary labels and sub-modular pairwise potentials [72].

Despite the intractability of maximum likelihood estimation, maximum a-posteriori estimation, and marginal inference for most models in the worst case, the inference task seems to be easier in practice than the theoretical worst case. Approximate inference algorithms can be extremely effective, often obtaining state-of-the-art results for these structured prediction tasks. Some important theoretical and empirical work on approximate inference include [38, 47, 61, 73–75].

Globerson, Roughgarden, Sontag, and Yildirim [61] analyzed the hardness of approximate inference in the case where performance is measured through the Hamming error, and provided conditions for the minimum-achievable Hamming error by studying a generative

[1] ↑In the literature, the cost functions c_v and $c_{u,v}$ are also known as unary and pairwise potentials, respectively.

model. Similar to the objective (4.1), the authors in [61] considered unary and pairwise noisy observations. As a concrete example [73], consider the problem of trying to recover opinions of individuals in social networks. Suppose that every individual in a social network can hold one of two opinions labeled by -1 or $+1$. One observes a measurement of whether neighbors in the network have an agreement in opinion, but the value of each measurement is flipped with probability p (pairwise observations). Additionally, one receives estimates of the opinion of each individual, perhaps using a classification model on their profile, but these estimates are corrupted with probability q (unary observations). Foster, Sridharan, and Reichman [73] generalized the work of Globerson, Roughgarden, Sontag, and Yildirim [61], who provided results for grid lattices, by providing results for trees and general graphs that allow tree decompositions (e.g., hypergrids and ring lattices).

Note that the above problem is challenging since there is a *statistical* and *computational* trade-off, as in several machine learning problems. The statistical part focuses on giving highly accurate labels while ignoring computational constraints. In practice this is unrealistic, one cannot afford to wait long times for each prediction, which motivated several studies on this trade-off (e.g., [16, 76]).

While the statistical and computational trade-off comes into sight in general, an interesting question is whether there are conditions for when recovery of the true labels is achievable in polynomial time. That is, conditions for when the Hamming error of the prediction is *zero* and can be obtained *efficiently*. The present chapter addresses this question. Finally, Chen, Kamath, Suh, and Tse [77] and Abbe, Bandeira, and Hall [78] also studied exact recovery. The former analyzed edges on sparse graphs—such as grids and rings— where one has multiple i.i.d. observations for each edge label; while the latter studied exact inference in the context of community detection, where there is a single (noisy) observation of each edge of the graph—in this case a complete graph.

4.1 Preliminaries

Vectors and matrices are denoted by lowercase and uppercase bold faced letters respectively (e.g., $\boldsymbol{a}, \boldsymbol{A}$), while scalars are in normal font weight (e.g., a). For a vector $\boldsymbol{a}$, and a

matrix $\boldsymbol{A}$, their entries are denoted by a_i and $A_{i,j}$ respectively. Indexing starts at 1, with $\boldsymbol{A}_{i,:}$ and $\boldsymbol{A}_{:,i}$ indicating the i-th row and i-th column of $\boldsymbol{A}$ respectively. Finally, sets and tuples are both expressed in uppercase blackboard bold and calligraphic fonts respectively. For example, $\mathbb{R}$ will denote the set of real numbers. The eigenvalues of a $n \times n$ matrix $\boldsymbol{A}$ are denoted as $\lambda_i(\boldsymbol{A})$, where λ_1 and λ_n correspond to the minimum and maximum eigenvalue respectively. Finally, the set of integers $\{1, \dots, n\}$ is represented as $[n]$.

We now present the inference task. We consider a similar problem setting to the one in [61], with the only difference that we consider general undirected graphs. That is, the goal is to predict a vector of n node labels $\widehat{\boldsymbol{y}} = (\widehat{y}_1, \dots, \widehat{y}_n)^\top$, where $\widehat{y}_i \in \{+1, -1\}$, from a set of observations $\boldsymbol{X}$ and $\boldsymbol{c}$, where $\boldsymbol{X}$ and $\boldsymbol{c}$ correspond to corrupted measurements of edges and nodes respectively. These observations are assumed to be generated from a ground truth labeling $\overline{\boldsymbol{y}}$ by a generative process defined via an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an edge noise $p \in (0, 0.5)$, and a node noise $q \in (0, 0.5)$. For each edge $(u, v) \in \mathcal{E}$, the edge observation $X_{u,v}$ is independently sampled to be $\overline{y}_u \overline{y}_v$ (*good edge*) with probability $1 - p$, and $-\overline{y}_u \overline{y}_v$ (*bad edge*) with probability p. While for each edge $(u, v) \notin \mathcal{E}$, the observation $X_{u,v}$ is always 0. Similarly, for each node $u \in \mathcal{V}$, the node observation c_u is independently sampled to be $\overline{y}_u$ (*good node*) with probability $1 - q$, and $-\overline{y}_u$ (*bad node*) with probability q. Thus, we have a *known* undirected connected graph $\mathcal{G}$, an *unknown* ground truth label vector $\overline{\boldsymbol{y}} \in \{+1, -1\}^n$, and noisy observations $\boldsymbol{X} \in \{-1, 0, +1\}^{n \times n}$ and $\boldsymbol{c} \in \{-1, +1\}^n$, and our goal is to find sufficient conditions for which we can predict, in polynomial time and with high probability, a vector label $\widehat{\boldsymbol{y}} \in \{-1, +1\}^n$ such that $\widehat{\boldsymbol{y}} = \overline{\boldsymbol{y}}$.

Definition 4.1.1 (Biased Rademacher variable)**.** *Let* $\mathrm{z}_p \in \{+1, -1\}$ *such that* $\mathbb{P}(\mathrm{z}_p = +1) = 1 - p$*, and* $\mathbb{P}(\mathrm{z}_p = -1) = p$*. We call* z_p *a biased Rademacher random variable with parameter* p *and expected value* $1 - 2p$*.*

From the definition above, we can write the edge observations as $X_{u,v} = \overline{y}_u \, \overline{y}_v \, \mathrm{z}_p^{(u,v)} \cdot \mathbb{1}[(u, v) \in \mathcal{E}]$, where $\mathrm{z}_p^{(u,v)}$ is a biased Rademacher with parameter p. While the node observation is $c_u = \overline{y}_u \, \mathrm{z}_q^{(u)}$, where $\mathrm{z}_q^{(u)}$ is a biased Rademacher with parameter q.

Given the generative process, we aim to solve the following optimization problem, which is based on the maximum likelihood estimator that returns the label $\arg\max_{\boldsymbol{y}} \mathbb{P}(\boldsymbol{X}, \boldsymbol{y})$ (see [61]):

$$\widehat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \ \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{y} + \alpha \boldsymbol{c}^\top \boldsymbol{y} \quad \text{subject to} \quad y_i = \pm 1, \tag{4.2}$$

where $\alpha = {}^{\log \frac{1-q}{q}}/_{\log \frac{1-p}{p}}$. In general, the above combinatorial problem is NP-hard to compute (e.g., see [79] for results on grids). Our goal is to find what structural properties of the graph $\mathcal{G}$ suffice to achieve, with high probability, exact recovery in polynomial time.

4.2 On Exact Recovery of Node Labels

Our approach consists of two stages, similar in spirit to [61]. We first use only the quadratic term from (4.2), which will give us two possible solutions, and then as a second stage, the linear term is used to decide the best between these two solutions.

4.2.1 First Stage

We analyze a semidefinite program (SDP) relaxation to the following combinatorial problem (4.3), motivated by the techniques in [78].

$$\widehat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} \ \frac{1}{2}\boldsymbol{y}^\top \boldsymbol{X}\boldsymbol{y} \quad \text{subject to} \quad y_i = \pm 1, \tag{4.3}$$

We denote the degree of node i as Δ_i, and the maximum node degree as $\Delta_{\max} = \max_{i\in\mathcal{V}} \Delta_i$. For any subset $\mathcal{S} \subset \mathcal{V}$, we denote its complement by $\mathcal{S}^C$ such that $\mathcal{S} \cup \mathcal{S}^C = \mathcal{V}$ and $\mathcal{S} \cap \mathcal{S}^C = \emptyset$. Furthermore, let $\mathcal{E}(\mathcal{S}, \mathcal{S}^C) = \{(i,j) \in \mathcal{E} \mid i \in \mathcal{S}, j \in \mathcal{S}^C \text{ or } j \in \mathcal{S}, i \in \mathcal{S}^C\}$, i.e., $|\mathcal{E}(\mathcal{S}, \mathcal{S}^C)|$ denotes the number of edges between $\mathcal{S}$ and $\mathcal{S}^C$.

Definition 4.2.1 (Edge Expansion)**.** *For a set $\mathcal{S} \subset \mathcal{V}$ with $|\mathcal{S}| \leq {}^{n}/_{2}$, its edge expansion, $\phi_{\mathcal{S}}$, is defined as: $\phi_{\mathcal{S}} = {}^{|\mathcal{E}(\mathcal{S},\mathcal{S}^C)|}/_{|\mathcal{S}|}$. Then, the edge expansion of a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined as: $\phi_{\mathcal{G}} = \min_{\mathcal{S}\subset\mathcal{V}, |\mathcal{S}| \leq n/2} \phi_{\mathcal{S}}$.*

In the literature, $\phi_{\mathcal{G}}$ is also known as the *Cheeger constant*, due to the geometric analogue defined by Cheeger in [80]. Next, we define the Laplacian matrix of a graph and the Rayleigh quotient which are also used throughout this section.

Definition 4.2.2 (Laplacian matrix)**.** *For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of n nodes. The Laplacian matrix $\boldsymbol{L}$ is defined as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{A}$ is the adjacency matrix.*

Definition 4.2.3 (Rayleigh quotient)**.** *For a given symmetric matrix $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ and non-zero vector $\boldsymbol{a} \in \mathbb{R}^n$, the Rayleigh quotient $R_{\boldsymbol{M}}(\boldsymbol{a})$, is defined as: $R_{\boldsymbol{M}}(\boldsymbol{a}) = \frac{\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a}}{\boldsymbol{a}^\top \boldsymbol{a}}$.*

We now define a signed Laplacian matrix.

Definition 4.2.4 (Signed Laplacian matrix)**.** *For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of n nodes. A signed Laplacian matrix, $\boldsymbol{M}$, is a symmetric matrix that satisfies $\boldsymbol{x}^\top \boldsymbol{M}\boldsymbol{x} = \sum_{(i,j)\in\mathcal{E}}(y_i x_i - y_j x_j)^2$, where $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{M}$ with eigenvalue 0, and $y_i \in \{+1, -1\}$.*

Note that the typical Laplacian matrix, as in Definition 4.2.2, fulfills the conditions of Definition 4.2.4 with $y_i = +1$ for all i. Next, we present an intermediate result for later use.

Lemma 4.2.1. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph of n nodes with Laplacian $\boldsymbol{L}$. Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a signed Laplacian with eigenvector $\boldsymbol{y}$ as in Definition 4.2.4, and let $\boldsymbol{a} \in \mathbb{R}^n$ be a vector such that $\langle \boldsymbol{y}, \boldsymbol{a}\rangle = 0$. Finally, let $\mathbf{1} \in \mathbb{R}^n$ be a vector of ones. Then we have that, for a given $\delta \in \mathbb{R}$, $R_{\boldsymbol{L}}(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1}) \leq R_{\boldsymbol{M}}(\boldsymbol{a})$, where the operator $\circ$ denotes the Hadamard product.*

Proof. First, note that $\boldsymbol{L}$ has a 0 eigenvalue with corresponding eigenvector $\mathbf{1}$. Also, we have that $\boldsymbol{x}^\top \boldsymbol{L}\boldsymbol{x} = \sum_{(i,j)\in\mathcal{E}}(x_i - x_j)^2$, for any vector $\boldsymbol{x}$. Then, $(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1})^\top \boldsymbol{L}(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1}) = \sum_{(i,j)\in\mathcal{E}}((y_i a_i + \delta) - (y_j a_j + \delta))^2 = (y_i a_i - y_j a_j)^2 = \boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a}$. Therefore, we have that the numerators of $R_{\boldsymbol{L}}(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1})$ and $R_{\boldsymbol{M}}(\boldsymbol{a})$ are equal. For the denominators, one can observe that: $(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1})^\top(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1}) = (\boldsymbol{a} \circ \boldsymbol{y})\top(\boldsymbol{a} \circ \boldsymbol{y}) + 2\delta\langle \mathbf{1}, \boldsymbol{a} \circ \boldsymbol{y}\rangle + \delta^2 \mathbf{1}^\top \mathbf{1} = \sum_i a_i y_i a_i y_i + 2\delta\langle \boldsymbol{a}, \boldsymbol{y}\rangle + \delta^2 n = \boldsymbol{a}^\top \boldsymbol{a} + \delta^2 n \geq \boldsymbol{a}^\top \boldsymbol{a}$, which implies that $R_{\boldsymbol{L}}(\boldsymbol{a} \circ \boldsymbol{y} + \delta\mathbf{1}) \leq R_{\boldsymbol{M}}(\boldsymbol{a})$. □

In what follows, we present our first result of this chapter, which has a connection to Cheeger's inequality [80].

Theorem 4.2.2. *Let $\mathcal{G}, \boldsymbol{M}, \boldsymbol{L}, \boldsymbol{y}$ be defined as in Lemma 4.2.1, and let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of $\boldsymbol{M}$. Then, we have that $\frac{\phi_{\mathcal{G}}^2}{4\Delta_{\max}} \leq \lambda_2$.*

Remark 4.2.3. *For a given undirected graph $\mathcal{G}$, its Laplacian matrix $\boldsymbol{L}$ fulfills the conditions of Lemma 4.2.1 and Theorem 4.2.2. That is, if $\boldsymbol{M} = \boldsymbol{L}$ in Theorem 4.2.2 then it becomes the known Cheeger's inequality. Therefore, our result in Theorem 4.2.2 apply for more general matrices and is of use for our next result.*

We now provide the SDP relaxation of problem (4.3). Let $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\top$, we have that $\boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{y} = \mathrm{Tr}(\boldsymbol{X}\boldsymbol{Y}) = \langle \boldsymbol{X}, \boldsymbol{Y} \rangle$. Since our prediction is a column vector $\boldsymbol{y}$, we have that $\boldsymbol{y}\boldsymbol{y}^\top$ is rank-1 and symmetric, which implies that $\boldsymbol{Y}$ is a positive semidefinite matrix. Therefore, our relaxation to the combinatorial problem (4.3) results in the following primal formulation[2]:

$$\widehat{\boldsymbol{Y}} = \arg\max_{\boldsymbol{Y}} \quad \langle \boldsymbol{X}, \boldsymbol{Y} \rangle \quad \text{subject to} \quad Y_{ii} = 1, \ \boldsymbol{Y} \succeq 0. \tag{4.4}$$

We will make use of the following matrix concentration inequality for our main proof.

Lemma 4.2.4 (Matrix Bernstein inequality, Theorem 1.4 in [81])**.** *Consider a finite sequence $\{\boldsymbol{N}_k\}$ of independent, random, self-adjoint matrices with dimension n. Assume that each random matrix satisfies $\mathbb{E}[\boldsymbol{N}_k] = 0$ and $\lambda_{\max}(\boldsymbol{N}_k) \leq R$ almost surely. Then, for all $t \geq 0$, $P\Big(\lambda_{\max}\big(\sum_k \boldsymbol{N}_k\big) \geq t\Big) \leq n \cdot \exp\left(\frac{-t^2/2}{\sigma^2 + Rt/3}\right)$, where $\sigma^2 = \|\sum_k \mathbb{E}[\boldsymbol{N}_k^2]\|$.*

The next theorem provides the conditions for exact recovery of labels with high probability.

Theorem 4.2.5. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected connected graph with n nodes, Cheeger constant $\phi_{\mathcal{G}}$, and maximum node degree $\Delta_{\max}$. Then, for the combinatorial problem (4.3), a solution $\widehat{\boldsymbol{y}} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$ is achievable in polynomial time by solving the SDP based relaxation (4.4), with probability at least $1 - \epsilon_1(\phi_{\mathcal{G}}, \Delta_{\max}, p)$, where p is the edge noise from our model, and*

$$\epsilon_1(\phi_{\mathcal{G}}, \Delta_{\max}, p) = 2n \cdot \mathrm{e}^{\frac{-3(1-2p)^2\phi_{\mathcal{G}}^4}{1536\Delta_{\max}^3 p(1-p) + 32(1-2p)(1-p)\phi_{\mathcal{G}}^2\Delta_{\max}}}.$$

[2]↑Here we dropped the constant 1/2 since it does not change the decision problem.

Regarding the statistical part from Theorem 4.2.5, it is natural to ask under what conditions we obtain a high probability statement. For example, one can observe that if $\phi_{\mathcal{G}}^2/\Delta_{\max} \in \Omega(n)$ then there is an exponential decay in the probability of error. Another example would be that if $\Delta_{\max} \in \mathcal{O}(\sqrt{n})$ and $\phi_{\mathcal{G}}^2/\Delta_{\max} \in \Omega(\sqrt{n})$ then we also obtain high probability statement. Thus, we are interested in finding what classes of graphs fulfill these or other structural properties so that we obtain a high probability bound in Theorem 4.2.5. Regarding the computational complexity of exact recovery, from Theorem 4.2.5, we are solving a SDP, and any SDP can be solved in polynomial time using methods such as the interior point method.

4.2.2 Second Stage

After the first stage, we obtain two feasible solutions for problem (4.3), that is, $\widehat{\boldsymbol{y}} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$. To decide which solution is correct we will use the node observations $\boldsymbol{c}$. Specifically, we will output the vector $\widehat{\boldsymbol{y}}$ that maximizes the score $\boldsymbol{c}^\top \widehat{\boldsymbol{y}}$. The next theorem formally states that, with high probability, $\widehat{\boldsymbol{y}} = \overline{\boldsymbol{y}}$ maximizes the score $\boldsymbol{c}^\top \widehat{\boldsymbol{y}}$ for a sufficiently large n.

Theorem 4.2.6. *Let $\widehat{\boldsymbol{y}} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$. Then, with probability at least $1 - \epsilon_2(n, q)$, we have that: $\boldsymbol{c}^\top \overline{\boldsymbol{y}} = \max_{\widehat{\boldsymbol{y}} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}} \boldsymbol{c}^\top \widehat{\boldsymbol{y}}$, where $\epsilon_2(n, q) = \mathrm{e}^{-\frac{n}{2}(1-2q)^2}$ and q is the node noise.*

Proof. We are interested in upper bounding the probability of predicting the wrong vector $\boldsymbol{y}$, that is,

$$\begin{aligned} P(\boldsymbol{c}^\top \boldsymbol{y}^* \leq -\boldsymbol{c}^\top \boldsymbol{y}^*) &= P(\boldsymbol{c}^\top \boldsymbol{y}^* \leq 0) \\ &= P\Big(\sum_{u \in \mathcal{V}} z_q^{(u)} \leq 0\Big) \\ &\leq \mathrm{e}^{-\frac{n}{2}(1-2q)^2}, \end{aligned}$$

where for the last equation we applied Hoeffding's inequality. □

Remark 4.2.7. *From Theorems 4.2.5 and 4.2.6, we obtain that exact recovery (i.e., $\widehat{\boldsymbol{y}} = \overline{\boldsymbol{y}}$) is achievable with probability at least $1 - \epsilon_1(\phi_{\mathcal{G}}, \Delta_{\max}, p) - \epsilon_2(n, q)$. Finally, from Theorem 4.2.6, it is clear that since the parameter $q \in (0, 0.5)$, for a sufficiently large n we have an*

exponential decay of the probability of error ϵ_2. Thus, we focus on the conditions of the first stage and provide examples in the next section.

4.2.3 Examples of Classes of Graphs

In this section, we provide examples of classes of graphs that yield high probability in Theorem 4.2.5.

Perhaps the most important example we provide in this section is related to the smoothed analysis on connected graphs [82]. Consider any fixed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and let $\widetilde{\mathcal{E}}$ be a random set of edges over the same set of vertices $\mathcal{V}$, where each edge $e \in \widetilde{\mathcal{E}}$ is independently drawn according to the Erdős-Rényi model with probability ε/n and where ε is a small (fixed) positive constant. We denote this as $\widetilde{\mathcal{E}} \sim \mathtt{ER}(n, \varepsilon/n)$, then let $\widetilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \cup \widetilde{\mathcal{E}})$ denote the random graph with the edge set $\widetilde{\mathcal{E}}$ added.

The model above can be considered a generalization of the classical Erdős-Rényi random graph, where one starts from an empty graph (i.e., $\mathcal{G} = (\mathcal{V}, \emptyset)$) and adds edges between all possible pairs of vertices independently with a given probability. The focus on "small" ε means that we are interested in the effect of a rather gentle random perturbation. In particular, it is known that graphs with bad expansion properties are not suitable for exact inference (see for instance [83]), but certain classes such as grids or planar graphs can yield good approximation under some regimes despite being bad expanders as shown by Globerson, Roughgarden, Sontag, and Yildirim [61]. Here we consider the graph $\mathcal{G}$ to be a bad expander and show that with a small perturbation, exact inference is achievable.

The following result was presented by Krivelevich, Reichman, and Samotij [82] in an equivalent fashion. Specifically, we set $\alpha = 1/2, \delta = \varepsilon/256$, $K = 128/\varepsilon$, $C = 1$, $s = K \log n$, which results with all the conditions being fulfilled in the proof of Theorem 2 in [82].

Lemma 4.2.8 (Theorem 2 in [82]). *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a connected graph, choose $\widetilde{\mathcal{E}} \sim \mathtt{ER}(n, \varepsilon/n)$, and let $\widetilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \cup \widetilde{\mathcal{E}})$. Then, for every $\varepsilon \in [1, n]$, we have that $\phi_{\widetilde{\mathcal{G}}} \geq \frac{\varepsilon}{256+256 \log n}$, with probability at least $1 - n^{-2.2-\frac{\log \varepsilon}{2}}$.*

The above lemma allows us to lower bound the Cheeger constant of the random graph $\widetilde{\mathcal{G}}$ with high probability, and is of use for our first example.

Corollary 4.2.9. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be any connected graph, choose $\widetilde{\mathcal{E}} \sim \texttt{ER}(n, \log^8 n/n)$, let $\widetilde{\mathcal{G}} = (\mathcal{V}, \mathcal{E} \cup \widetilde{\mathcal{E}})$ and let $\Delta_{\max}^{\widetilde{\mathcal{G}}}$ be the maximum node degree of $\widetilde{\mathcal{G}}$. Then, we have that $\phi_{\widetilde{\mathcal{G}}}^2/\Delta_{\max}^{\widetilde{\mathcal{G}}} \in \Omega(\log^5 n)$ and $\Delta_{\max}^{\widetilde{\mathcal{G}}} \in \mathcal{O}(\log^9 n)$ with high probability. Therefore, exact recovery in polynomial time is achievable with high probability.*

Proof. Fix $\varepsilon = \log^8 n$. Let $\epsilon_r(n, \varepsilon) = n^{-2.2 - \frac{\log \varepsilon}{2}}$, then from Lemma 4.2.8 we get $\phi_{\widetilde{\mathcal{G}}} \in \Omega(\log^7 n)$ with probability at least $1 - \epsilon_r(n, \varepsilon)$. Let $\Delta_{\max}$ be the maximum node degree of graph $\mathcal{G}$, then it is clear that $\Delta_{\max}^{\widetilde{\mathcal{G}}}$ is a random variable with expected value $\mathbb{E}\left[\Delta_{\max}^{\widetilde{\mathcal{G}}}\right] \leq \Delta_{\max} + \log^8 n$. By applying Markov's inequality we obtain $P(\Delta_{\max}^{\widetilde{\mathcal{G}}} \geq t) \leq \mathbb{E}\left[\Delta_{\max}^{\widetilde{\mathcal{G}}}\right]/t \leq (\Delta_{\max} + \log^8 n)/t$ for $t > 0$. Set $t = \log^9 n$, then let $\epsilon_\Delta(\Delta_{\max}, n) = (\Delta_{\max} + \log^8 n)/\log^9 n$, we have that $\Delta_{\max}^{\widetilde{\mathcal{G}}} \leq \log^9 n$ with probability at least $1 - \epsilon_\Delta(\Delta_{\max}, n)$.

By using the union bound and noting that $\epsilon_r \to 0$ and $\epsilon_\Delta \to 0$ as $n \to \infty$, we have that $\phi_{\widetilde{\mathcal{G}}}^2/\Delta_{\max}^{\widetilde{\mathcal{G}}} \in \Omega(\log^5 n)$ and $\Delta_{\max}^{\widetilde{\mathcal{G}}} \in \mathcal{O}(\log^9 n)$ with high probability. Finally, this leads to $\epsilon_1 \to 0$ as $n \to \infty$, thus, exact inference is achievable in polynomial time. □

We emphasize the nice property of random graphs $\widetilde{\mathcal{G}}$ shown in Corollary 4.2.9, that is, by adding a small perturbation—edges from the Erdős-Rényi model with small probability—we are able to obtain exact inference in spite of $\mathcal{G}$ being a bad expander. Our next two examples include complete graphs and d-regular expanders. The following corollary shows that, with high probability, exact recovery of labels for complete graphs is possible in polynomial time.

Corollary 4.2.10 (Complete graphs)**.** *Let $\mathcal{G} = \mathcal{K}_n$, where $\mathcal{K}_n$ denotes a complete graph of n nodes. Then, we have that $\phi_{\mathcal{G}}^2/\Delta_{\max} \in \Omega(n)$. Therefore, exact recovery in polynomial time is achievable with high probability.*

Proof. For any set $\mathcal{S} \subset \mathcal{V}$ with $|\mathcal{S}| \leq n/2$, we have that:

$$\phi_{\mathcal{S}} = \frac{|\mathcal{E}(\mathcal{S}, \mathcal{S}^C)|}{|\mathcal{S}|} = \frac{|\mathcal{S}| \cdot |\mathcal{S}^C|}{|\mathcal{S}|} = |\mathcal{S}^C| \quad \Longrightarrow \quad \phi_{\mathcal{G}} = \lceil \frac{n}{2} \rceil.$$

Since $\mathcal{G}$ is a complete graph, we have that $\Delta_{\max} = n - 1$, which yields $\phi_{\mathcal{G}}^2/\Delta_{\max} \in \Omega(n)$. Thus, from Theorem 4.2.5, we have that $\epsilon_1(\phi_{\mathcal{G}}, \Delta_{\max}, p) \to 0$ as $n \to \infty$. □

Another important class of graphs that admits exact recovery is the family of d-regular expanders [84], which is defined below.

Definition 4.2.5 (d-regular expander)**.** *A d-regular graph with n nodes is an expander with constant $c > 0$ if, for every set $\mathcal{S} \subset \mathcal{V}$ with $|\mathcal{S}| \leq n/2$, $|\mathcal{E}(\mathcal{S}, \mathcal{S}^C)| \geq c \cdot d \cdot |\mathcal{S}|$.*

Corollary 4.2.11 (Expanders graphs)**.** *Let $\mathcal{G}$ be a d-regular expander with constant c. Then, we have that $\phi_{\mathcal{G}}^2/\Delta_{\max} \in \Omega(d)$. If $d \in \Omega(\log n)$ then exact recovery in polynomial time is achievable with high probability.*

Proof. From Definition 4.2.5, we have that $\phi_{\mathcal{G}} \geq c \cdot d$. Since the graph is regular, we have that $\Delta_{\max} = d$. Therefore, $\phi_{\mathcal{G}}^2/\Delta_{\max} \in \Omega(d)$. Finally, if $d \in \Omega(\log n)$, then $\epsilon_1(\phi_{\mathcal{G}}, \Delta_{\max}, p)$ decays in at least n^{-c_1} for some constant $c_1 > 0$. That is, $\epsilon_1(\phi_{\mathcal{G}}, \Delta_{\max}, p) \to 0$ as $n \to \infty$. □

4.3 Exact Inference from the Degree-4 Sum-of-Squares Hierarchy

In the previous section, we studied the sufficient conditions for realizing exact recovery from a SDP viewpoint. In contrast, we now study the same problem under the sum-of-squares (SoS) hierarchy of relaxations [85–87], which is a sequential tightening of convex relaxations based on SDP. We study the SoS hierarchy because it is tighter than other known hierarchies such as the Sherali-Adams and Lovász-Schrijver hierarchies [88]. In addition, our motivation to study the level-2 or degree-4 SoS relaxation stems from three reasons. First, higher-levels of the hierarchy, while polynomial time solvable, are already computationally very costly. This is one of the reasons the SoS hierarchy have been mostly used as a proof system for finding lower bounds in hard problems (e.g., for the planted clique problem, see [89]). Second, little is still known about the level-2 SoS relaxation, where [90] and [91] are attempts to understand its geometry. Third, there is empirical evidence on the improvement in exact recoverability with respect to SDP, an example of which is depicted in Figure 4.1.

While it is known that the level-2 SoS relaxation has a tighter search space than that of SDP, it is not obvious why it can perform better than SDP for *exact recovery*. In this section, we aim to understand the origin of such improvement from a graph theoretical perspective. We will show that the solution of the dual of the SoS relaxed problem is related to finding edge weights of the Johnson and Kneser graphs, where the weights fulfill the SoS constraints and intuitively allow the input graph to increase its algebraic connectivity. Finally, as byproduct

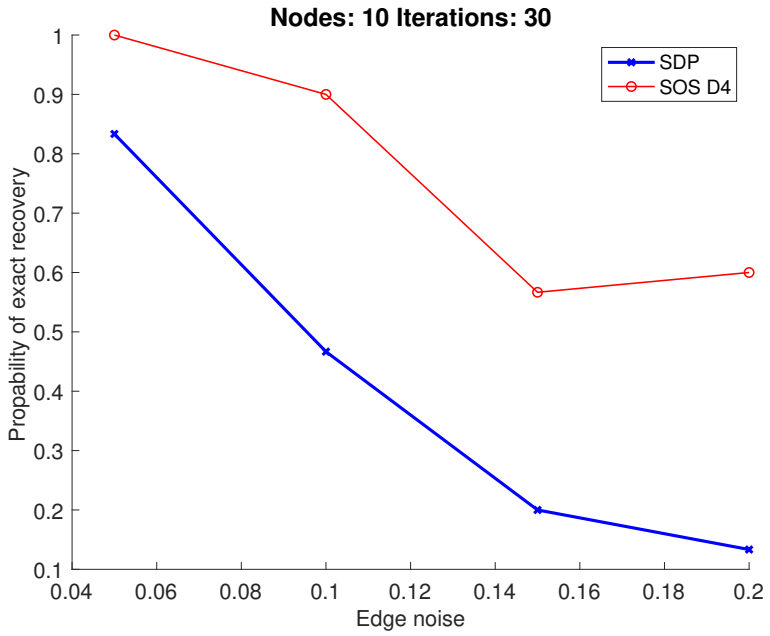


Figure 4.1. A comparison between the degree-4 SoS and SDP relaxations in the context of structured prediction. We observe that SoS attains a higher probability of exact recovery, for different levels of edge noise p. (See Section 4.3.1 for a formal problem definition).

of our analysis, we derive a novel Cheeger-type lower bound for the algebraic connectivity of graphs with *signed* edge weights.

We emphasize that the objective of this section is on the *understanding* of exact recoverability by using the degree-4 SoS. Scalability of the SoS hierarchy is an important open problem that is actively under study [92, 93] and is beyond the scope of this analysis.

4.3.1 Problem Definition

For clarity purposes, we restate the problem under analysis and slightly modify our notation for convenience. We aim to predict a vector of n node labels $\boldsymbol{y} = (y_1, \ldots, y_n)^\top$, where $y_i \in \{+1, -1\}$, from a set of observations $\boldsymbol{X}$, where $\boldsymbol{X}$ corresponds to noisy measurements of edges. These observations are assumed to be generated from a ground truth labeling $\overline{\boldsymbol{y}}$ by a generative process defined via an undirected connected graph $G = (V, E)$, where $V = [n]$, and an edge noise $p \in (0, 0.5)$. For each edge $(u, v) \in E$, we have a *single* independent

edge observation $X_{u,v} = \overline{y}_u \overline{y}_v$ with probability $1-p$, and $X_{u,v} = -\overline{y}_u \overline{y}_v$ with probability p. While for each edge $(u,v) \notin E$, the observation $X_{u,v}$ is always 0. Thus, we have a *known* undirected connected graph G, an *unknown* ground truth label vector $\overline{\boldsymbol{y}} \in \{+1,-1\}^n$, noisy observations $\boldsymbol{X} \in \{-1,0,+1\}^{n \times n}$. Given that we consider only edge observations, our goal is to understand when one can predict, in polynomial time and with high probability, a vector label $\boldsymbol{y} \in \{-1,+1\}^n$ such that $\boldsymbol{y} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$.

Given the aforementioned generative process, our focus will be to solve the following optimization problem, which stems from using maximum likelihood estimation [61]:

$$\max_{\boldsymbol{y}} \quad \boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{y}, \quad \text{subject to} \quad y_i = \pm 1, \ \forall i \in [n]. \tag{4.5}$$

Recall that, in general, the above combinatorial problem is NP-hard to compute [79]. Let $\boldsymbol{y}^{\mathsf{dis}}$ denote the optimizer of eq.(4.5). It is clear that for any label vector $\boldsymbol{y}$, the negative label vector $-\boldsymbol{y}$ attains the same objective value in eq.(4.5). Thus, we say that one can achieve exact recovery by solving eq.(4.5) if $\boldsymbol{y}^{\mathsf{dis}} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$. Given the computational hardness of solving eq.(4.5), we next revise approaches that relax problem (4.5) to one that can be solved in polynomial time. Then, our focus will be to understand the effects of the structural properties of the graph G in achieving, with high probability, exact recovery in the continuous problem.

Semidefinite Programming Relaxation

As explained in Section 4.2, a popular approach for approximating problem (4.5) is to consider a larger search space that is simpler to describe and is convex. In particular, let $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\top$, that is, $Y_{i,j} = y_i y_j$ and noting that $\boldsymbol{Y}$ is a rank-1 positive semidefinite matrix. We can rewrite the objective of problem (4.5) in matrix terms as follows, $\boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{y} = \mathrm{Tr}(\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^\top) = \mathrm{Tr}(\boldsymbol{X}\boldsymbol{Y}) = \langle \boldsymbol{X}, \boldsymbol{Y} \rangle$. Thus, we have

$$\max_{\boldsymbol{Y}} \quad \langle \boldsymbol{X}, \boldsymbol{Y} \rangle, \quad \text{subject to} \quad \boldsymbol{Y} \succeq 0, \quad Y_{i,i} = 1, \forall i \in [n]. \tag{4.6}$$

Let $\boldsymbol{Y}^*$ denote the optimizer of the problem above, then, in this case, we say that exact recovery is realized by solving eq.(4.6) if $\boldsymbol{Y}^* = \overline{\boldsymbol{y}}\overline{\boldsymbol{y}}^\top$. The only constraint dropped in problem (4.6) with respect to problem (4.5) is the rank-1 constraint, which makes problem (4.6) convex. The above relaxation is known as semidefinite programming (SDP) relaxation and is typically used as an approximation algorithm. That is, after obtaining a continuous solution $\boldsymbol{Y}^* \in \mathbb{R}^{n\times n}$, a rounding procedure is performed to recover an approximate solution in $\{\pm 1\}^n$, e.g., see [94, 95]. We now introduce tighter levels of relaxations known as the SoS hierarchy, and we will see that an SDP relaxation corresponds to the first level of the SoS hierarchy.

Sum-of-Squares Hierarchy

Let $[n]^{\leq d} = \{\emptyset\} \cup [n]^1 \cup \ldots \cup [n]^d$ denote the set of (possibly empty) tuples, of length up to d, composed of the integers from 1 to n, e.g., $[2]^{\leq 2} = \{\emptyset, (1), (2), (1,1), (1,2), (2,1), (2,2)\}$. Also, let the summation between two tuples be the concatenation of all the elements in them, e.g., for $\mathcal{C}_1 = (1,1,2), \mathcal{C}_2 = (3,1)$ we have $\mathcal{C}_1 + \mathcal{C}_2 = (1,1,2,3,1)$. We use $\psi(\mathcal{C})$ to denote the tuple with elements from $\mathcal{C}$ sorted in ascending order, e.g., for $\mathcal{C} = (2,1,1,3)$ we have $\psi(\mathcal{C}) = (1,1,2,3)$. We also use $|\mathcal{C}|$ to denote the cardinality of $\mathcal{C}$. For two distinct tuples $\mathcal{C}_1$ and $\mathcal{C}_2$, the expression $\mathcal{C}_1 < \mathcal{C}_2$ means that either $|\mathcal{C}_1| < |\mathcal{C}_2|$, or $|\mathcal{C}_1| = |\mathcal{C}_2|$ and $\exists i$ such that the i-ith entry of $\mathcal{C}_2$ is greater than the i-th entry of $\mathcal{C}_1$. Then, for a set of tuples $\mathfrak{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$, we say that $\mathfrak{C}$ is in lexicographical order if $\mathcal{C}_i < \mathcal{C}_j$ for all $i < j$. Finally, for a matrix $\boldsymbol{Y} \in \mathbb{R}^{[n]^{\leq \ell} \times [n]^{\leq \ell}}$, we index its rows and columns by using tuples in $[n]^{\leq \ell}$ ordered lexicographically, e.g., for $\boldsymbol{Y} \in \mathbb{R}^{[5]^{\leq 3}\times[5]^{\leq 3}}$ we have that $\boldsymbol{Y}_{(1,1,2),(5)}$ corresponds to the entry at row $(1,1,2)$ and column (5).

It is convenient to rewrite the objective of problem (4.5) as a polynomial optimization problem, i.e., $\sum_i \sum_j X_{i,j} y_i y_j$, so that the standard machinery of SoS optimization [85, 86, 96] can be applied to formulate the degree-d relaxation. Then, for an even number d, the degree-d (or level $d/2$) SoS relaxation of problem (4.5) takes the form

$$\max_{\boldsymbol{Y} \in \mathbb{R}^{[n]^{\leq \frac{d}{2}} \times [n]^{\leq \frac{d}{2}}}} \sum_{i=1}^{n}\sum_{j=1}^{n} X_{i,j} Y_{(i),(j)}, \tag{4.7}$$

$$\text{subject to } \boldsymbol{Y} \succeq 0; \quad \boldsymbol{Y}_{(\emptyset)(\emptyset)} = 1; \quad \boldsymbol{Y}_{(i)+\mathcal{C}_1,(i)+\mathcal{C}_2} = \boldsymbol{Y}_{\mathcal{C}_1,\mathcal{C}_2}, \ \forall i \in [n], \ |\mathcal{C}_1|, |\mathcal{C}_2| \leq d/2 - 1;$$
$$\boldsymbol{Y}_{\mathcal{C}_1,\mathcal{C}_2} = \boldsymbol{Y}_{\mathcal{C}_1,\mathcal{C}_2}, \ \forall \psi(\mathcal{C}_1 + \mathcal{C}_2) = \psi(\mathcal{C}_1 + \mathcal{C}_2), \ |\mathcal{C}_1|, |\mathcal{C}_2|, |\mathcal{C}_1|, |\mathcal{C}_2| \leq d/2.$$

In the problem above, each entry of the matrix $\boldsymbol{Y}$ corresponds to a reparametrization that takes the form $\boldsymbol{Y}_{\mathcal{C}_1,\mathcal{C}_2} = \prod_{i\in\mathcal{C}_1} y_i \prod_{j\in\mathcal{C}_2} y_j = \prod_{i\in\mathcal{C}_1+\mathcal{C}_2} y_i$, which is also known as a pseudomoment matrix [86, 96]. In problem (4.7), the second constraint can be thought as a normalization constraint. The third list of constraints corresponds to $\prod_{j\in\mathcal{C}} y_j \cdot y_i^2 = \prod_{j\in\mathcal{C}} y_j, \forall |\mathcal{C}| \leq d-2$, which is equivalent to $y_i = \pm 1$ in problem (4.5). Finally, the last list of constraints corresponds to $\prod_{j\in\mathcal{C}_1+\mathcal{C}_2} y_j = \prod_{j\in\mathcal{C}_1+\mathcal{C}_2} y_j, \forall \psi(\mathcal{C}_1 + \mathcal{C}_2) = \psi(\mathcal{C}_1 + \mathcal{C}_2)$, and $|\mathcal{C}_1|, |\mathcal{C}_2|, |\mathcal{C}_1|, |\mathcal{C}_2| \leq d/2$, which states that $\boldsymbol{Y}_{\mathcal{C}_1,\mathcal{C}_2}$ should be invariant to all permutations of the tuple $\mathcal{C}_1 + \mathcal{C}_2$. One can note that, for $d = 2$, the degree-2 (or level 1) SoS relaxation is equivalent to the SDP relaxation in eq.(4.6). It is clear that for a larger d, the degree-d SoS relaxation gives a tighter convex relaxation of problem (4.5). While one can solve problem (4.7) to a fixed accuracy using general-purpose SDP algorithms in polynomial time in n, the computational complexity will be of order $n^{\mathcal{O}(d)}$. Thus, it is important that d be of low order.

As the focus of this section is on the degree-4 SoS relaxation, we start by formulating the corresponding optimization problem. In problem (4.7), for $d = 4$, the matrix $\boldsymbol{Y}$ is in $\mathbb{R}^{[n]^{\leq 2}\times[n]^{\leq 2}}$, that is, $\boldsymbol{Y}$ is a matrix of dimension $(1 + n + n^2) \times (1 + n + n^2)$. Bandeira and Kunisky [90, Appendix A] showed that one can write an equivalent formulation by using only the principal submatrix of $\boldsymbol{Y}$ indexed by $[n]^2 \times [n]^2$ (i.e., a matrix of dimension $n^2 \times n^2$). The reduced formulation takes the form:

$$\max_{\boldsymbol{Y}\in\mathbb{R}^{[n]^2\times[n]^2}} \sum_{i=1}^{n}\sum_{j=1}^{n} X_{i,j} Y_{(1,1),(i,j)}, \tag{4.8}$$
$$\text{subject to } \boldsymbol{Y} \succeq 0; \quad \boldsymbol{Y}_{(i,i)(j,j)} = 1, \ \forall i, j \in [n]; \quad \boldsymbol{Y}_{(i,i)(j,k)} = \boldsymbol{Y}_{(i,i)(j,k)}, \ \forall i, i, j, k \in [n];$$
$$\boldsymbol{Y}_{(i,j)(k,\ell)} = \boldsymbol{Y}_{(\pi_1,\pi_2)(\pi_3,\pi_4)}, \forall \ i, j, k, \ell \in [n], \ \boldsymbol{\pi} \in \Pi(i, j, k, \ell),$$

where $\Pi(i, j, k, \ell)$ is the set of all permutations of (i, j, k, ℓ). We will go one step further in the reduction and show that one can indeed cast an equivalent formulation to problem (4.8) by using only the principal submatrix of $\boldsymbol{Y} \in \mathbb{R}^{[n]^2\times[n]^2}$ indexed by $\binom{[n]}{2} \times \binom{[n]}{2}$, i.e., a

matrix of dimension $\frac{n(n-1)}{2} \times \frac{n(n-1)}{2}$. Here, it will be more convenient to use sets instead of tuples for indexing the rows and columns of $\boldsymbol{Y}$, where $\binom{[n]}{2}$ denotes the set of all unordered combinations of length 2 from the numbers in $[n]$, e.g., $\binom{[3]}{2} = \{\{1,2\},\{1,3\},\{2,3\}\}$. For further distinction against the matrix $\boldsymbol{Y} \in \mathbb{R}^{[n]^2 \times [n]^2}$, we will use $\widetilde{\boldsymbol{Y}}$ to denote the matrix indexed by $\binom{[n]}{2} \times \binom{[n]}{2}$.

We will also make use of the next set of definitions, which are important for stating our results.

Definition 4.3.1 (The level-2 vector)**.** *For any vector $\boldsymbol{v} \in \mathbb{R}^n$, its level-2 vector, denoted by $\boldsymbol{v}^{(2)} \in \mathbb{R}^{\binom{n}{2}}$ and indexed by $\binom{[n]}{2}$, is defined as $v^{(2)}_{\{i,j\}} = v_i v_j$.*

We also define the level-2 version of a graph as follows.

Definition 4.3.2 (The level-2 graph)**.** *Let $G = (V, E)$, where $V = [n]$, be any undirected graph of n nodes with adjacency matrix $\boldsymbol{A} \in \{0,1\}^{n \times n}$. The level-2 graph of G, denoted by $G^{(2)} = (\binom{[n]}{2}, E^{(2)})$ and with adjacency matrix $\boldsymbol{A}^{(2)} \in \{0,1\}^{\binom{n}{2} \times \binom{n}{2}}$, has its adjacency matrix defined as $A^{(2)}_{\{i,k\},\{k,j\}} = 1$ if $(i,j) \in E$ for all $i < j < k \in [n]$, and $A^{(2)}_{\{i,j\},\{k,\ell\}} = 0$ for all $i < j < k < \ell \in [n]$.*

The next type of graphs have been studied for several years within the graph theory community and we will later show how they relate to the solution of the level-2 SoS relaxation.

Definition 4.3.3 (Johnson graph [97])**.** *For a set $[n]$, the Johnson graph $\mathcal{J}(n,k)$ has all the k-element subsets of $[n]$ as vertices, and two vertices are adjacent if and only if the intersection of the two vertices (subsets) contains $(k-1)$-elements.*

Definition 4.3.4 (Kneser graph [98])**.** *For a set $[n]$, the Kneser graph $\mathcal{K}(n,k)$ has all the k-element subsets of $[n]$ as vertices, and two vertices are adjacent if and only if the two vertices (subsets) are disjoint.*

From Definitions 4.3.3 and 4.3.4, we are interested in $\mathcal{J}(n,2)$ and $\mathcal{K}(n,2)$, where we first note that $\mathcal{K}(n,2)$ is the complement of $\mathcal{J}(n,2)$. We also note that for a graph G of n nodes, by construction, the level-2 graph of G is always a subgraph of the Johnson graph $\mathcal{J}(n,2)$, and is equal to $\mathcal{J}(n,2)$ if and only if G is the complete graph of n nodes. Finally,

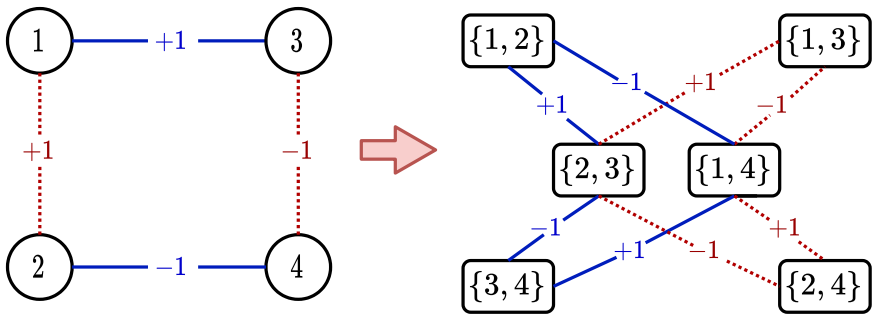


Figure 4.2. Illustration of the level-2 construction of $\boldsymbol{X}$. The edge values in the grid graph correspond to the observation $\boldsymbol{X}$, while the edge values on the right graph correspond to level-2 matrix $\boldsymbol{X}^{(2)}$. The solid blue and dotted red lines indicate that the observation is correct and corrupted, respectively.

since our observation matrix $\boldsymbol{X}$ depends on a graph G, one can also extend $\boldsymbol{X}$ to a matrix in $\{-1, 0, +1\}^{\binom{n}{2}\times\binom{n}{2}}$. We will use $\boldsymbol{X}^{(2)}$ to denote the level-2 version of $\boldsymbol{X}$. Specifically, $X^{(2)}_{\{i,k\},\{k,j\}} = X_{i,j}$ for all $i < j < k \in [n]$, and $X^{(2)}_{\{i,j\},\{k,\ell\}} = 0$ for all $i < j < k < \ell \in [n]$. For further clarity, we illustrate the level-2 construction of $\boldsymbol{X}$ in Figure 4.2, where the input graph is a 2 by 2 grid.

Next, we present an optimization problem that is equivalent to problem (4.8) but in terms of the level-2 constructions defined above. For notational convenience, we will use $\boldsymbol{S}^{+(ijk\ell)}_{-(stuv)}$ to denote a sparse *symmetric* matrix such that the only non-zero entries are $\boldsymbol{S}_{\{i,j\},\{k,\ell\}} = 1$, and $\boldsymbol{S}_{\{s,t\},\{u,v\}} = -1$.

$$\max_{\widetilde{\boldsymbol{Y}}\in\mathbb{R}^{\binom{[n]}{2}\times\binom{[n]}{2}}} \frac{1}{n-2}\langle \boldsymbol{X}^{(2)}, \widetilde{\boldsymbol{Y}}\rangle, \tag{4.9}$$

$$\text{subject to } \widetilde{\boldsymbol{Y}} \succeq 0;\ \widetilde{\boldsymbol{Y}}_{\mathcal{C},\mathcal{C}} = 1, \forall \mathcal{C} \in \tbinom{[n]}{2};\ \langle \boldsymbol{S}^{+(ikkj)}_{-(ikkj)}, \widetilde{\boldsymbol{Y}}\rangle = 0, \forall i < j < k < k \in [n];$$

$$\langle \boldsymbol{S}^{+(ijk\ell)}_{-\boldsymbol{\pi}(ijk\ell)}, \widetilde{\boldsymbol{Y}}\rangle = 0, \forall i < j < k < l \in [n],\ \boldsymbol{\pi} \in \Pi(i,j,k,l).$$

Proposition 4.3.1. *Problem* (4.9) *is equivalent to problem* (4.8).

Proof. By construction of the level-2 matrix $\boldsymbol{X}^{(2)}$, we have that each entry $X_{i,j}$ is repeated $n-2$ times. Thus, it follows that the objectives in problems (4.8) and (4.9) are equal.

Let $\boldsymbol{Y}$ be a feasible solution to problem (4.8), then clearly the principal submatrix indexed by $\binom{[n]}{2} \times \binom{[n]}{2}$ is a feasible solution to problem (4.9). It remains to verify that if $\widetilde{\boldsymbol{Y}}$ is a

feasible solution to problem (4.9) then there exists a matrix $\boldsymbol{Y}$ such that it is feasible to problem (4.8) and has $\widetilde{\boldsymbol{Y}}$ as a principal submatrix. We define the entries of $\boldsymbol{Y}$ as follows,

$$
\begin{aligned}
Y_{(i,i)(j,j)} &= \widetilde{Y}_{\{i,j\},\{i,j\}} \\
Y_{(i,i)(j,k)} &= \widetilde{Y}_{\{i,j\},\{i,k\}} \\
Y_{(i,j)(k,\ell)} &= \widetilde{Y}_{\{i,j\},\{k,\ell\}}.
\end{aligned}
$$

Clearly, $\boldsymbol{Y}$ will fulfill the constraints of problem (4.8) if $\widetilde{\boldsymbol{Y}}$ is feasible to problem (4.9). In particular, one can verify that $\boldsymbol{v}^\top \boldsymbol{Y} \boldsymbol{v} \geq 0$ for any $\boldsymbol{v}$ if $\widetilde{\boldsymbol{Y}} \succeq 0$, which concludes our proof. □

Remark 4.3.1. *Let $\overline{\boldsymbol{y}}^{(2)}$ be the level-2 vector of the ground-truth labeling $\overline{\boldsymbol{y}}$, and let $\widetilde{\boldsymbol{Y}}^*$ be the optimizer of problem (4.9). Then, we say that exact recovery is realized if $\widetilde{\boldsymbol{Y}}^* = \overline{\boldsymbol{y}}^{(2)} \overline{\boldsymbol{y}}^{(2)\top}$.*

4.3.2 The Dual Problem

A key ingredient for our analysis is the dual formulation of problem (4.9), which takes the following form

$$
\begin{aligned}
&\min_{\widetilde{\boldsymbol{V}},\, \boldsymbol{\mu}} \ \operatorname{Tr}(\widetilde{\boldsymbol{V}}), \qquad (4.10) \\
&\text{subject to } \widetilde{\boldsymbol{V}} \text{ is diagonal,} \\
&\qquad \widetilde{\boldsymbol{\Lambda}} = \widetilde{\boldsymbol{V}} - \frac{\boldsymbol{X}^{(2)}}{n-2} + \sum_{i<j<k<k} \mu_{ikkj}^{ikkj} \boldsymbol{S}_{-(ikkj)}^{+(ikkj)} + \sum_{\substack{i<j<k<\ell \\ \pi \in \Pi(i,j,k,\ell)}} \mu_{\pi(ijk\ell)}^{ijk\ell} \boldsymbol{S}_{-\pi(ijk\ell)}^{+(ijk\ell)} \succeq 0,
\end{aligned}
$$

where $\widetilde{\boldsymbol{V}} \in \mathbb{R}^{\binom{[n]}{2} \times \binom{[n]}{2}}$, $\mu_{ikkj}^{ikkj} \in \mathbb{R}$ and $\mu_{\pi(ijk\ell)}^{ijk\ell} \in \mathbb{R}$ are the dual variables of the second constraint, and the third and fourth list of constraints from the primal formulation (4.9), respectively. The dual variable $\boldsymbol{\mu}$ denotes all the scalars μ_{ikkj}^{ikkj} and $\mu_{\pi(ijk\ell)}^{ijk\ell}$.

We have that if there exists $\widetilde{\boldsymbol{Y}}, \widetilde{\boldsymbol{V}}, \boldsymbol{\mu}$ that satisfy the Karush-Kuhn-Tucker (KKT) conditions [99], then $\widetilde{\boldsymbol{Y}}$ and $\widetilde{\boldsymbol{V}}, \boldsymbol{\mu}$ are primal and dual optimal, and strong duality holds in this case. Let $\overline{\boldsymbol{y}}^{(2)}$ be the level-2 vector of the *ground-truth* labeling $\overline{\boldsymbol{y}}$. Since we are interested

in exact recovery, we will consider the solution $\widetilde{\boldsymbol{Y}} = \overline{\boldsymbol{y}}^{(2)}\overline{\boldsymbol{y}}^{(2)\top}$ for the rest of our analysis, where it is clear that such setting satisfies the primal constraints. Let

$$\widetilde{\boldsymbol{V}} = \frac{\operatorname{diag}\Big(\boldsymbol{X}^{(2)}\widetilde{\boldsymbol{Y}}\Big)}{n-2} - \operatorname{diag}\Big(\sum_{i<j<k<k} \mu_{ikkj}^{ikkj} \boldsymbol{S}_{-(ikkj)}^{+(ikkj)} \widetilde{\boldsymbol{Y}}\Big) - \operatorname{diag}\Big(\sum_{\substack{i<j<k<\ell \\ \pi\in\Pi(ijk\ell)}} \mu_{\pi(ijk\ell)}^{ijk\ell} \boldsymbol{S}_{-\pi(ijk\ell)}^{+(ijk\ell)} \widetilde{\boldsymbol{Y}}\Big), \tag{4.11}$$

where, for a matrix $\boldsymbol{M}$, $\operatorname{diag}(\boldsymbol{M})$ denotes the diagonal matrix formed from the diagonal entries of $\boldsymbol{M}$. Complementary slackness and stationarity require the trace of $\widetilde{\boldsymbol{V}}$ to be equal to the trace of the r.h.s. of eq.(4.11), which is clearly satisfied by construction. Thus, if we find an assignment of $\boldsymbol{\mu}$ such that $\widetilde{\boldsymbol{\Lambda}} \succeq 0$, we would have an optimal solution since all KKT conditions are fulfilled. Nevertheless, we are also interested in $\widetilde{\boldsymbol{Y}} = \overline{\boldsymbol{y}}^{(2)}\overline{\boldsymbol{y}}^{(2)\top}$ being the *unique* optimal solution, where we note that having $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) > 0$ suffices to guarantee a unique solution. The argument follows from the fact that, by the setting of eq.(4.11), we have $\widetilde{\boldsymbol{\Lambda}}\overline{\boldsymbol{y}}^{(2)} = 0$. Thus, if $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) > 0$ then $\overline{\boldsymbol{y}}^{(2)}$ spans all of the null-space of $\widetilde{\boldsymbol{\Lambda}}$. Combined with the KKT conditions, we have that $\widetilde{\boldsymbol{Y}}$ should be a multiple of $\overline{\boldsymbol{y}}^{(2)}\overline{\boldsymbol{y}}^{(2)\top}$. Since $\widetilde{\boldsymbol{Y}}$ has diagonal entries equal to 1, we must have that $\widetilde{\boldsymbol{Y}} = \overline{\boldsymbol{y}}^{(2)}\overline{\boldsymbol{y}}^{(2)\top}$.

Putting all pieces together, we have that under eq.(4.11), if for some $\boldsymbol{\mu}$ we have that $\widetilde{\boldsymbol{\Lambda}} \succeq 0$ and $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) > 0$, then the optimizer of problem (4.9) is $\overline{\boldsymbol{y}}^{(2)}\overline{\boldsymbol{y}}^{(2)\top}$, i.e., we obtain exact recovery. Since $\overline{\boldsymbol{y}}^{(2)}$ is an eigenvector of $\widetilde{\boldsymbol{\Lambda}}$ with eigenvalue zero, we focus on controlling the quantity $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = \min_{\boldsymbol{v}\perp\overline{\boldsymbol{y}}^{(2)}} \frac{\boldsymbol{v}^\top\widetilde{\boldsymbol{\Lambda}}\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}}$. [3] Also, as $\widetilde{\boldsymbol{\Lambda}}$ depends on the noisy observation $\boldsymbol{X}^{(2)}$, we have that $\widetilde{\boldsymbol{\Lambda}}$ is a random quantity. Then, by using Weyl's theorem on eigenvalues, we have

$$\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = \lambda_2(\widetilde{\boldsymbol{\Lambda}} - \mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right] + \mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) \geq \lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) + \lambda_1(\widetilde{\boldsymbol{\Lambda}} - \mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]). \tag{4.12}$$

In eq.(4.12), let t be a lower bound to $\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right])$, i.e., $\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) \geq t$. Then, the second summand can be lower bounded by using matrix concentration inequalities. Specifically, by using matrix Bernstein inequality [81], one can obtain that $\mathbb{P}[\lambda_1(\widetilde{\boldsymbol{\Lambda}} - \mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) \leq -t] \leq \mathcal{O}(n^2\mathrm{e}^{-t})$. Thus, we can now focus on the first summand, which will be lower bounded by a

[3] ↑This expression comes from the variational characterization of eigenvalues.

novel Cheeger-type inequality. In the next subsections, we look at the expected value of $\widetilde{\mathbf{\Lambda}}$ in more detail.

4.3.3 The Expected Value and the Algebraic Connectivity of the Level-2 Graph

We will show how $\mathbb{E}[\widetilde{\mathbf{\Lambda}}]$ is related to the Laplacian matrix of $G^{(2)}$ (the level-2 version of G). To do so, we will use the following definitions and notation.

For a *signed* weighted graph $H = (U, F)$, we use $\boldsymbol{W}^H$ to denote its weight matrix, that is, the entry $W^H_{i,j} \in \mathbb{R}$ is the weight of edge $(i,j) \in F$ and is zero if $(i,j) \notin F$. For any set $T \subset U$, its boundary is defined as $\partial T = \{(i,j) \mid i \in T \text{ and } j \notin T\}$; while its boundary weight is defined as $\omega(\partial T) = \sum_{i \in T, j \notin T} W^H_{i,j}$. The number of nodes in T is denoted by $|T|$. The degree of a node is defined as $\deg(i) = \sum_{j \neq i} W^H_{i,j}$.

Definition 4.3.5. *Let H be a graph with degree matrix $\boldsymbol{D}^H$ and weight matrix $\boldsymbol{W}^H$, where $\boldsymbol{D}^H$ is a diagonal matrix such that $D_{i,i} = \deg(i)$. The Laplacian matrix of H is defined as $\boldsymbol{L}^H = \boldsymbol{D}^H - \boldsymbol{W}^H$.*

Definition 4.3.6 (Cheeger constant [80])**.** *For a graph $H = (U, F)$ of n nodes, its Cheeger constant is defined as $\phi(H) = \min_{T \subset U,\, |T| \leq n/2} \frac{\omega(\partial T)}{|T|}$.*

Remark 4.3.2. *For unweighted graphs, the definitions above match the standard definitions for node degree, boundary of a set, and Laplacian matrix, as in Section 4.2.*

Next, we analyze the scenario where all the scalar dual variables in $\boldsymbol{\mu}$ are zero, we defer the case when they are not for the next subsection.

The $\boldsymbol{\mu} = \mathbf{0}$ scenario. From eq.(4.11) we have that $\widetilde{\boldsymbol{V}} = \operatorname{diag}(\boldsymbol{X}^{(2)}\widetilde{\boldsymbol{Y}})/(n-2)$. Hence, for all $i < j \in [n]$, we have $\mathbb{E}[\widetilde{V}_{\{i,j\},\{i,j\}}] = {}^{(1-2p)}/_{(n-2)} \cdot \deg(\{i,j\})$. In addition, we have $\mathbb{E}[X^{(2)}_{\{i,k\},\{k,j\}}] = (1-2p)\ \bar{y}_i\ \bar{y}_j \mathbb{1}[(i,j) \in E]$, for all $i < j < k \in [n]$. [4] Finally, since $\boldsymbol{\mu} = \mathbf{0}$, we have $\widetilde{\mathbf{\Lambda}} = \widetilde{\boldsymbol{V}} - \frac{\boldsymbol{X}^{(2)}}{n-2}$. Therefore,

$$\mathbb{E}[\widetilde{\mathbf{\Lambda}}] = \frac{1-2p}{n-2} \widetilde{\mathbf{\Upsilon}} \boldsymbol{L}^{G^{(2)}} \widetilde{\mathbf{\Upsilon}}, \tag{4.13}$$

[4]↑ Recall that if $(i,j) \in E$, then $X_{i,j} = -\bar{y}_i \bar{y}_j$ with probability p, and $X_{i,j} = \bar{y}_i \bar{y}_j$ otherwise. If $(i,j) \notin E$ then $X_{i,j} = 0$.

where $\widetilde{\boldsymbol{\Upsilon}}$ is a diagonal matrix with entries equal to the entries in $\overline{\boldsymbol{y}}^{(2)}$. Recall that $\overline{y}^{(2)}_{\{i,j\}} = \overline{y}_i \overline{y}_j$ and $\overline{y}_i \in \{\pm 1\}$ for all $i \in [n]$. Then, we have that $\widetilde{\boldsymbol{\Upsilon}}^{-1} = \widetilde{\boldsymbol{\Upsilon}}$ and, thus, the matrix $\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]$ and $\frac{1-2p}{n-2}\boldsymbol{L}^{G^{(2)}}$ are similar. The latter means that both matrices share the same spectrum, i.e.,

$$\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) = \frac{1-2p}{n-2}\lambda_2(\boldsymbol{L}^{G^{(2)}}). \tag{4.14}$$

Notice that the level-2 graph $G^{(2)}$ is unweighted since G is unweighted. That implies that one can lower bound $\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right])$ by using existing lower bounds for the second eigenvalue[5] of the Laplacian matrix of $G^{(2)}$. In particular, one can have [100]

$$\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) = \frac{1-2p}{n-2}\lambda_2(\boldsymbol{L}^{G^{(2)}}) \geq \frac{(1-2p)\phi(G^{(2)})^2}{2(n-2)\deg_{\max}}. \tag{4.15}$$

Finally, we note that considering $\boldsymbol{\mu} = \mathbf{0}$ is equivalent to not having the third and fourth list of constraints in problem (4.9). At this point, the reader might wonder if, setting $\boldsymbol{\mu} = \mathbf{0}$ and solving problem (4.9) yields in any better chances of exact recovery than solving problem (4.6). We answer the latter in the negative.

Proposition 4.3.2. *Without the third and fourth list of constraints, problem* (4.9) *does not improve exact recoverability with respect to problem* (4.6)*.*

Proof. We will show the equivalence between problem (4.8), *without the third and fourth list of constraints*, and problem (4.6). Then, by Proposition 4.3.1, our claim follows.

It is clear that the objectives in problems (4.6) and (4.8) are equal. Let $\boldsymbol{Y}^{\mathsf{sdp}}$ be a feasible solution to problem (4.6), then we define $\boldsymbol{Y}^{\mathsf{sos}}$ as follows,

$$Y^{\mathsf{sos}}_{(i,i)(j,k)} = Y^{\mathsf{sdp}}_{i,j}, \quad Y^{\mathsf{sos}}_{(i,j)(k,\ell)} = 0.$$

Since $\boldsymbol{Y}^{\mathsf{sdp}} \succeq 0$, it follows that $\boldsymbol{Y}^{\mathsf{sos}} \succeq 0$ and, thus, $\boldsymbol{Y}^{\mathsf{sos}}$ is feasible to problem (4.8) without the third and fourth list of constraints. Similarly, in the other direction, let $\boldsymbol{Y}^{\mathsf{sos}}$ be a feasible solution to problem (4.8) without the third and fourth list of constraints, and define $\boldsymbol{Y}^{\mathsf{sdp}}$ to

[5]↑The second eigenvalue of the Laplacian matrix is also known as the algebraic connectivity.

be the principal submatrix of $\boldsymbol{Y}^{\mathsf{sos}}$ with the first n rows and columns. Then, it follows that if $\boldsymbol{Y}^{\mathsf{sos}} \succeq 0$ then $\boldsymbol{Y}^{\mathsf{sdp}} \succeq 0$, which is feasible to problem (4.6). □

The purpose of Proposition 4.3.2 is to highlight the role that a $\boldsymbol{\mu} \neq \mathbf{0}$ will play in showing the improvement in exact recoverability of the degree-4 SoS relaxation with respect to the SDP relaxation, which is discussed next.

4.3.4 Systems of Sets and a Novel Cheeger-Type Lower Bound

We next show how the third and fourth list of constraints of problem (4.9) relate to finding edge weights of the Johnson and Kneser graphs, respectively, so that the Laplacian matrix of a *new* graph is positive semidefinite (PSD).

Note that the third and fourth list of constraints in the SoS relaxation (4.9) do not depend on the input graph, nor on the edge observations or the ground-truth node labels. Instead, they are constraints coming from the SoS relaxation, as explained in the subsequent paragraphs to problem (4.7). That means that they depend only on the number of nodes, n, and on the degree of the relaxation, $d = 4$. We will illustrate in detail the case of $n = 4$ as it is easier to generalize from there to any value of n.

Recall that $\boldsymbol{S}^{+(ijk\ell)}_{-(stuv)}$ is a symmetric matrix that has non-zero entries $\boldsymbol{S}_{\{i,j\},\{k,\ell\}} = 1$, and $\boldsymbol{S}_{\{s,t\},\{u,v\}} = -1$. By taking advantage of the implicit symmetry constraint from $\widetilde{\boldsymbol{Y}} \succeq 0$, for $n = 4$, one can realize that the third list of constraints in problem (4.9) has six different constraints in total (with their respective dual variables), which are:

$$\mu^{1224}_{1334} : \langle \boldsymbol{S}^{+(1224)}_{-(1334)}, \widetilde{\boldsymbol{Y}} \rangle = 0, \quad \mu^{2113}_{2443} : \langle \boldsymbol{S}^{+(2113)}_{-(2443)}, \widetilde{\boldsymbol{Y}} \rangle = 0, \quad \mu^{1332}_{1442} : \langle \boldsymbol{S}^{+(1332)}_{-(1442)}, \widetilde{\boldsymbol{Y}} \rangle = 0,$$
$$\mu^{3114}_{3224} : \langle \boldsymbol{S}^{+(3114)}_{-(3224)}, \widetilde{\boldsymbol{Y}} \rangle = 0, \quad \mu^{1223}_{1443} : \langle \boldsymbol{S}^{+(1223)}_{-(1443)}, \widetilde{\boldsymbol{Y}} \rangle = 0, \quad \mu^{2334}_{2114} : \langle \boldsymbol{S}^{+(2334)}_{-(2114)}, \widetilde{\boldsymbol{Y}} \rangle = 0.$$

Similarly, from the fourth list of constraints we have:

$$\mu^{1324}_{1234} : \langle \boldsymbol{S}^{+(1324)}_{-(1234)}, \widetilde{\boldsymbol{Y}} \rangle = 0, \quad \mu^{2314}_{1234} : \langle \boldsymbol{S}^{+(2314)}_{-(1234)}, \widetilde{\boldsymbol{Y}} \rangle = 0.$$

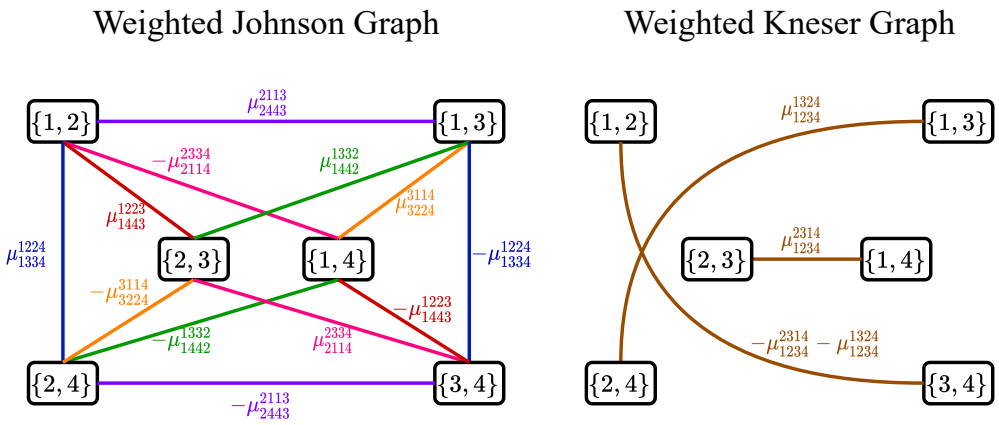


Figure 4.3. Johnson and Kneser graphs for $n = 4$, where each edge weight is related to some dual variables from the SoS constraints. Edge weights with the same color sum to zero, see eq.(4.16).

In the dual formulation (4.10), for both lists above, the matrices $\boldsymbol{S}$ are weighted by the dual variables μ. Then, the two weighted summations can be thought of as weight matrices of some graphs. Interestingly, such graphs happen to be the Johnson and Kneser graphs [6] for the first and second list of constraints above, respectively. In Figure 4.3, we show an illustration of the Johnson and Kneser graphs with edge weights corresponding to the dual variables.

Let $\oplus$ denote the symmetric difference of sets. Also, let $\boldsymbol{W}^{\mathcal{J}}$ and $\boldsymbol{W}^{\mathcal{K}}$ denote the weight matrices of the Johnson and Kneser graphs, respectively. Then, for any n, the third and fourth list of constraints of problem (4.9) translate to having the following constraints on $\boldsymbol{W}^{\mathcal{J}}$ and $\boldsymbol{W}^{\mathcal{K}}$,

$$\sum_{\substack{\mathcal{C}_1,\mathcal{C}_2 \\ \mathcal{C}_1\oplus\mathcal{C}_2=\{i,j\}}} \boldsymbol{W}^{\mathcal{J}}_{\mathcal{C}_1,\mathcal{C}_2} = 0,\ \forall\ i<j\in[n], \qquad \sum_{\substack{\mathcal{C}_1,\mathcal{C}_2 \\ \mathcal{C}_1\oplus\mathcal{C}_2=\{i,j,k,\ell\}}} \boldsymbol{W}^{\mathcal{K}}_{\mathcal{C}_1,\mathcal{C}_2} = 0,\ \forall\ i<j<k<\ell\in[n]. \tag{4.16}$$

Thus, by using the construction in eq.(4.11), we have that the PSD constraint of the dual formulation (4.10) can be rewritten in terms of $\boldsymbol{W}^{\mathcal{J}}$ and $\boldsymbol{W}^{\mathcal{K}}$ as follows,

$$\widetilde{\boldsymbol{\Lambda}} = \frac{\operatorname{diag}\left(\boldsymbol{X}^{(2)}\widetilde{\boldsymbol{Y}}\right)}{n-2} - \frac{\boldsymbol{X}^{(2)}}{n-2} + \left(\operatorname{diag}(\boldsymbol{W}^{\mathcal{J}}\widetilde{\boldsymbol{Y}}) - \boldsymbol{W}^{\mathcal{J}}\right) + \left(\operatorname{diag}(\boldsymbol{W}^{\mathcal{K}}\widetilde{\boldsymbol{Y}}) - \boldsymbol{W}^{\mathcal{K}}\right) \succeq 0.$$

[6] For any n, whenever we write the Johnson and Kneser graphs, we refer to $\mathcal{J}(n,2)$ and $\mathcal{K}(n,2)$, respectively.

Let $\widetilde{\mathcal{G}} = G^{(2)} \cup \mathcal{J} \cup \mathcal{K}$ such that $\boldsymbol{W}^{\widetilde{\mathcal{G}}} = \frac{1-2p}{n-2}\boldsymbol{W}^{G^{(2)}} + \boldsymbol{W}^{\mathcal{J}} + \boldsymbol{W}^{\mathcal{K}}$, and noting that w.l.o.g. one can multiply the weights in eq.(4.16) by $\bar{y}_i\bar{y}_j$ and $\bar{y}_i\bar{y}_j\bar{y}_k\bar{y}_\ell$, respectively. We can use a similar argument to that of eq.(4.14) and obtain

$$\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) = \lambda_2(\boldsymbol{L}^{\widetilde{\mathcal{G}}}). \tag{4.17}$$

The subtlety for lower bounding eq.(4.17) is that, unless all edge weights are zero, the Johnson and Kneser graphs will both have at least one negative edge weight in order to fulfill eq.(4.16). In other words, the Laplacian matrix $\boldsymbol{L}^{\widetilde{\mathcal{G}}}$ is no longer guaranteed to be PSD. That fact alone rules out almost all existing results on lower bounding the algebraic connectivity as it is mostly assumed that all edge weights are positive. Among the few works that study the Laplacian matrix with negative weights, one can find [101, 102]; however, their results focus on finding conditions for positive semidefiniteness of the Laplacian matrix in the context of electrical circuits and not in finding a lower bound. Our next result, generalizes the lower bound in [100] by considering negative edge weights.

Theorem 4.3.3. *Let $H = H^+ \cup H^-$ be a weighted graph such that H^+ and H^- denote the disjoint subgraphs of H with positive and negative weights, respectively. Also, let $\deg_{\max}^{H^+}$ denote the maximum node degree of H^+. Then, we have that $\lambda_2(\boldsymbol{L}^H) \geq \frac{\phi(H^+)^2}{2\deg_{\max}^{H^+}} + 2 \cdot \text{mincut}(H^-)$.*

Remark 4.3.4. *We remark that the reason we do not consider other versions of the Laplacian matrix (e.g., the normalized Laplacian matrix which is guaranteed to be PSD even in the presence of negative weights) is because how our primal/dual construction (see Section 4.3.2) leads to a valid solution of the constraints in eq.(4.10) which also satisfies the KKT conditions. That is, using other notions of Laplacian matrix (see e.g., [103–108]) would* not *satisfy the optimality conditions needed for exact recovery—in particular, stationarity and complementary slackness. In fact, one of the challenges we face in our analysis is that by having the standard Laplacian matrix, its minimum eigenvalue can be negative, as shown in our example in Section 4.3.5 and also discussed in [107], which motivated the search of a more general lower bound for the algebraic connectivity of signed graphs (Theorem 4.3.3).*

In the case when there are positive weights only, the theorem above yields the typical Cheeger bound [100]. When there is at least one negative weight, the bound shows an interesting trade-off between the Cheeger constant of the positive subgraph and the minimum cut of the negative subgraph. By applying Theorem 4.3.3 to eq.(4.17), we obtain

$$\lambda_2(\mathbb{E}\left[\widetilde{\boldsymbol{\Lambda}}\right]) \geq \phi(\widetilde{\mathcal{G}}^+)^2/(2\deg_{\max}^{\widetilde{\mathcal{G}}^+}) + 2 \cdot \text{mincut}(\widetilde{\mathcal{G}}^-). \tag{4.18}$$

Without the weights of the Johnson and Kneser graphs, the lower bound above is equal to that of eq.(4.15). Also, recall that, by construction, the edge set of the level-2 graph $G^{(2)}$ is a subset of the edge set of the Johnson graph, and that the Kneser graph is the complement of the Johnson graph. That means that $\widetilde{\mathcal{G}}$ will be a complete graph of $\binom{n}{2}$ vertices, where the edge weights of the Kneser graph are exclusively related to the dual variables μ, while the edge weights of the Johnson graph might have an interaction between the noisy edge observations and the dual variables μ. Intuitively, the SoS solution will try to find negative weights for the Johnson and Kneser graphs of as low magnitude as possible, so that the minimum-cut of the negative subgraph does not make the algebraic connectivity negative. From the concentration argument stated after eq.(4.12), we conclude that as the lower bound in eq.(4.18) increases then the more likely to realize exact recovery.

4.3.5 Example

The goal of this section is to provide a concrete example where the SoS relaxation (4.8) *achieves* exact recovery but the SDP relaxation (4.6) *does not*. Since for any input graph with n vertices, its level-2 version has $\binom{n}{2}$ vertices, we select a value of $n = 5$ so that the level-2 graph has 10 nodes and the plots can still be visually inspected in detail. Figure (4.4a) shows the ground-truth labels of a graph with 5 nodes and 8 edges. Figure (4.4b) corresponds to the observation matrix $\boldsymbol{X}$. In this case, only one edge is corrupted (the red edge). Figure (4.4c) shows the graph where an edge label of -1 or 1 indicates whether the observed edge value was corrupted or not, respectively. The latter graph is obtained by $\boldsymbol{\Upsilon}\boldsymbol{X}\boldsymbol{\Upsilon}$, where $\boldsymbol{\Upsilon}$ denotes a diagonal matrix with entries from $\overline{\boldsymbol{y}}$, similar to the procedure in eq.(4.13). Let $\boldsymbol{\Lambda}$ be the dual variable of the PSD constraint in the SDP relaxation (4.6). Then, under a

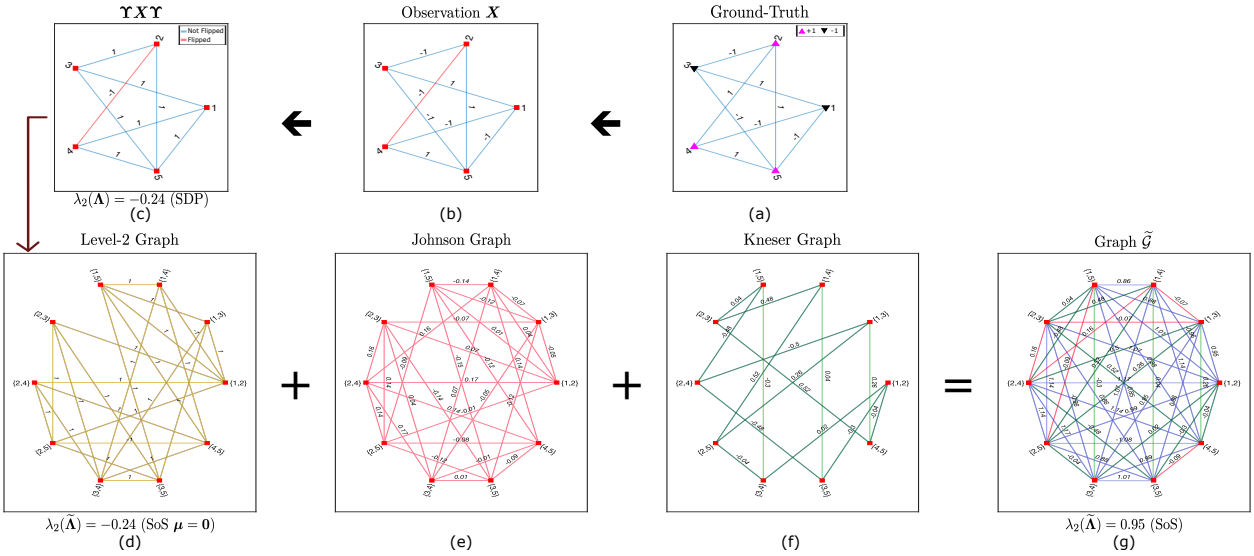


Figure 4.4. Detailed example of how the level-2 SoS relaxation results in improving the algebraic connectivity of the input graph through a combination of weights of its level-2 version, and the Johnson and Kneser graphs. In the final graph $\widetilde{\mathcal{G}}$, green and red lines indicate that their weights remain unchanged w.r.t. the Kneser and Johnson edge weights, respectively; while blue lines indicate that their weights resulted from the summation of weights from the Level-2 and Johnson graphs.

similar dual construction to the one in [17, 78], we have that $\lambda_2(\boldsymbol{\Lambda}) = \min_{\boldsymbol{v}\perp\overline{\boldsymbol{y}}} \frac{\boldsymbol{v}^\top \boldsymbol{\Lambda} \boldsymbol{v}}{\boldsymbol{v}^\top \boldsymbol{v}}$ is equal to the second eigenvalue of the Laplacian matrix of Figure (4.4c). Thus, we can observe that, for SDP, the ground-truth solution $\overline{\boldsymbol{Y}}$ attains a value of $\lambda_2(\boldsymbol{\Lambda}) = -0.24 < 0$, hence, exact recovery fails.

In Figure (4.4d), we show the level-2 graph of $\boldsymbol{\Upsilon}\boldsymbol{X}\boldsymbol{\Upsilon}$, i.e., $\widetilde{\boldsymbol{\Upsilon}}\boldsymbol{X}^{(2)}\widetilde{\boldsymbol{\Upsilon}}$. As argued by Proposition 4.3.2, by setting $\boldsymbol{\mu} = \mathbf{0}$ the SoS does not do any better than SDP, which is verified by obtaining $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = -0.24 < 0$, hence, exact recovery also fails in this case. However, by solving problem (4.9), we obtain $\boldsymbol{\mu} \neq \mathbf{0}$ which, as discussed in Section 4.3.4, relates to edge weights in the Johnson and Kneser graphs. Those edge weights are depicted in Figures (4.4e) and (4.4f), respectively. Finally, after summing all the weights of the level-2 graph, Johnson and Kneser graphs, we obtain a complete graph depicted in Figure (4.4g). In the latter, we have that $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = 0.95 > 0$, which guarantees that $\widetilde{\boldsymbol{Y}} = \overline{\boldsymbol{y}}^{(2)}\overline{\boldsymbol{y}}^{(2)\top}$, i.e., exact recovery succeeds.

Motivated by eq.(4.16) and Theorem 4.3.3, in Appendix C.4, we show a non-trivial construction of the Kneser graph weights based on only the node degrees of the level-2 graph.

4.4 Exact Inference Under Fairness Constraints

As the use of machine learning in decision making increases in our society [109], researchers have shown interest in developing methods that can mitigate unfair decisions or avoid bias amplification. With the existence of several notions of fairness [110–113], and some of them being simultaneously incompatible [114], the first step is to define the notion of fairness, which is commonly dependent upon the task on hand. For our purposes, we will adapt the notion of statistical parity and apply it to the exact inference problem. Several notions of statistical parity have been studied in prior works [115–117], where, in general, statistical parity enforces a predictor to be independent of the protected attribute. In particular, in regression, Agarwal, Dudik, and Wu [115] relaxed the principle of statistical parity and studied ε-away difference of marginal CDF and conditional CDF on the protected attribute. Finally, unlike the works on supervised learning [118–120], the work of Chierichetti, Kumar, Lattanzi, and Vassilvitskii [121] is among the first to adapt the disparate impact doctrine (related to statistical parity) to unsupervised learning, specifically, to the clustering problem.

For the rest of this chapter we will study the generative model described in the previous sections with the addition of a fairness constraint.

4.4.1 Statistical Parity

In a few words, statistical (or demographic) parity enforces a predictor to be independent of the protected attributes. While the definition has been mostly used in supervised learning, in this work we try to adapt this notion of fairness to an inference problem. Specifically, we say that, given a vector attribute $\boldsymbol{a}$, the assignment $\overline{\boldsymbol{y}}$ is fair under statistical parity if $\overline{\boldsymbol{y}}^\top \boldsymbol{a} = 0$. In particular, we will consider $\bar{y}_i \in \{-1, +1\}$ to be the node labels of a graph, as described in the next section. That is, we would like the partitions (or clusters) to have

the same sum of the attribute $\boldsymbol{a}$.[7] As an example, we can consider the nodes of a graph to be individuals, and the node label to represent the community an individual belongs to. Then, given a vector of resources $\boldsymbol{a}$, demographic parity will enforce to output a labeling that assigns the same amount of resources to each community.

4.4.2 Problem Definition

We consider a similar generative model as the one studied in previous sections. For clarity purposes, we next provide a complete description of the problem to be studied in the next sections. We aim to predict a vector of n node labels $\widehat{\boldsymbol{y}} = (\widehat{y}_1, \dots, \widehat{y}_n)^\top$, where $\widehat{y}_i \in \{+1, -1\}$, from a set of observations $\boldsymbol{X}$ and $\boldsymbol{c}$, where $\boldsymbol{X}$ and $\boldsymbol{c}$ correspond to noisy measurements of edges and nodes respectively. These observations are assumed to be generated from a *fair* ground truth labeling $\overline{\boldsymbol{y}}$ by a generative process defined via an undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an edge noise $p \in (0, 0.5)$, and a node noise $q \in (0, 0.5)$. For each edge $(u, v) \in \mathcal{E}$, we have a *single* independent edge observation $X_{u,v} = \overline{y}_u \overline{y}_v$ with probability $1 - p$, and $X_{u,v} = -\overline{y}_u \overline{y}_v$ with probability p. While for each edge $(u, v) \notin \mathcal{E}$, the observation $X_{u,v}$ is always 0. Similarly, for each node $u \in \mathcal{V}$, we have an independent node observation $c_u = \overline{y}_u$ with probability $1 - q$, and $c_u = -\overline{y}_u$ with probability q. In addition, we are given a set of attributes $\mathbb{A} = \{\boldsymbol{a}_1, \dots, \boldsymbol{a}_k\}$ such that $\boldsymbol{a}_i \in \mathbb{R}^n$ and $\langle \boldsymbol{a}_i, \overline{\boldsymbol{y}} \rangle = 0$ for all $i \in [k]$, i.e., for each i we have $\sum_{j|\overline{y}_j=1} (a_i)_j = \sum_{j|\overline{y}_j=-1} (a_i)_j$. In other words, we say that the ground truth labeling $\overline{\boldsymbol{y}}$ is fair under statistical parity with respect to the set of attributes $\mathbb{A}$. Thus, we have a *known* undirected connected graph $\mathcal{G}$, an *unknown* fair ground truth label vector $\overline{\boldsymbol{y}} \in \{+1, -1\}^n$, noisy observations $\boldsymbol{X} \in \{-1, 0, +1\}^{n \times n}$ and $\boldsymbol{c} \in \{-1, +1\}^n$, a set $\mathbb{A}$ of k attributes $\boldsymbol{a}_i \in \mathbb{R}^n$, and our goal is to find sufficient conditions for which we can predict, in polynomial time and with high probability, a vector label $\widehat{\boldsymbol{y}} \in \{-1, +1\}^n$ such that $\widehat{\boldsymbol{y}} = \overline{\boldsymbol{y}}$.

Given the generative process, our prediction $\widehat{\boldsymbol{y}}$ is given by the following combinatorial problem:

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \{-1,+1\}^n}{\arg\max} \quad \frac{1}{2} \boldsymbol{y}^\top \boldsymbol{X} \boldsymbol{y} + \alpha \cdot \boldsymbol{c}^\top \boldsymbol{y} \tag{4.19}$$

[7] ↑Note that the elements of the attribute can already be divided by the size of the clusters they belong to, in which case it would represent equal averages. Here we make no assumptions on the elements of $\boldsymbol{a}$.

$$\text{subject to} \quad \langle \boldsymbol{a}_i, \boldsymbol{y} \rangle = 0, \ \forall i \in [k]$$
$$y_i = \pm 1, \ \forall i \in [n].$$

where $\alpha = \log \frac{1-q}{q} / \log \frac{1-p}{p}$. Thus, we have a similar objective to that of eq.(4.2) with the addition of a linear constraint related to fairness.

Remark 4.4.1. *The optimization problem 4.19 is clearly NP-hard to compute in general. For instance, consider the case where $k = 1$, and $(a_1)_j$ is a positive integer for all $j \in [n]$, i.e., there is a single attribute with positive entries. Also, let $\boldsymbol{X} = \boldsymbol{0}$ and $\boldsymbol{c} = \boldsymbol{0}$, that is, any vector $\boldsymbol{y}$ will attain the same objective value. Then, the problem reduces to finding an assignment $\boldsymbol{y}$ such that $\langle \boldsymbol{a}_1, \boldsymbol{y} \rangle = 0$, which is equivalent to the known NP-complete partition problem. Another example is the case when $\boldsymbol{a}_1 = \boldsymbol{1}$, that is, a feasible solution has to have the same number of positive and negative labels. Thus, if $\boldsymbol{X}$ is such that it encourages minimizing the number of edges between clusters, the problem reduces to the minimum bisection problem, which is known to be NP-complete [122]. Finally, consider also the case in which $k = 0$, then it is known that when the graph $\mathcal{G}$ is a grid, the problem is NP-hard [79].*

In the next section, we relax the combinatorial problem 4.19 to a continuous problem, and formally show how the addition of some fairness constraints such as that of statistical parity (as described above) can increase the exact recovery rate of Theorem 4.2.5.

4.4.3 The Effect of Linear Constraints on Exact Recovery

Our approach to analyze exact recovery will focus on the quadratic term of problem (4.19). This is because if $\widehat{\boldsymbol{y}} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$ from solving only the quadratic term with the constraints, then by using majority vote with respect to the observation $\boldsymbol{c}$ one can decide which of $\{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$ is optimal, as done in Section 4.2.2. We will show sufficient conditions for exact recovery in polynomial time through the use of semidefinite programming (SDP) relaxations similar to that of Section 4.2.1.

Next, we provide the SDP relaxation of problem (4.19).

$$\widehat{\boldsymbol{Y}} = \underset{\boldsymbol{Y} \in \mathbb{R}^{n \times n}}{\arg\max} \quad \langle \boldsymbol{X}, \boldsymbol{Y} \rangle \tag{4.20}$$

$$\begin{aligned} \text{subject to} \quad & Y_{ii} = 1, \ i \in [n], \\ & \boldsymbol{a}_i^\top \boldsymbol{Y} \boldsymbol{a}_i = 0, \ i \in [k], \\ & \boldsymbol{Y} \succeq 0. \end{aligned}$$

Next, we present an intermediate result that is of use for the proof of Theorem 4.4.4.

Lemma 4.4.2. *Let $\boldsymbol{M} \in \mathbb{R}^{n\times n}$ be a positive semidefinite matrix and let $\boldsymbol{N} \in \mathbb{R}^{n\times n}$ be a rank-l positive semidefinite matrix, and consider a non-negative $\alpha \in \mathbb{R}$. Define $\Delta = \lambda_2(\boldsymbol{M}) - \lambda_1(\boldsymbol{M})$, where $\lambda_1(\cdot)$ and $\lambda_2(\cdot)$ represent the minimum and second minimum eigenvalue, respectively. Also, let $\boldsymbol{q}_1$ denote the first eigenvector of $\boldsymbol{M}$, and let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ denote the eigenvectors of $\boldsymbol{N}$ related to $\lambda_1(\boldsymbol{N}), \ldots, \lambda_n(\boldsymbol{N})$ respectively. Then, we have that:*

$$\lambda_1(\boldsymbol{M} + \alpha \cdot \boldsymbol{N}) \geq \lambda_1(\boldsymbol{M}) + \max_i \left(\frac{\alpha_i + \Delta}{2} - \sqrt{\left(\frac{\alpha_i + \Delta}{2}\right)^2 - \alpha_i \cdot \Delta \cdot (\boldsymbol{v}_i^\top \boldsymbol{q}_1)^2} \right),$$

where $\alpha_i = \alpha \cdot \lambda_i(\boldsymbol{N})$.

Proof. Let $\boldsymbol{M} = \boldsymbol{Q}\boldsymbol{D}\boldsymbol{Q}^\top$ and $\boldsymbol{N} = \sum_{i=n-l+1}^{n} \lambda_i(N)\boldsymbol{v}_i\boldsymbol{v}_i^\top$ be the eigendecomposition of $\boldsymbol{M}$ and $\boldsymbol{N}$ respectively. Let us define $\boldsymbol{T} = \boldsymbol{Q}^\top(\boldsymbol{M} + \alpha \cdot \boldsymbol{N})\boldsymbol{Q}$. Since $\boldsymbol{T}$ and $(\boldsymbol{M} + \alpha \cdot \boldsymbol{N})$ are similar matrices, their spectrum is the same, which means that $\lambda_1(\boldsymbol{M} + \alpha \cdot \boldsymbol{N}) = \lambda_1(\boldsymbol{T})$. By letting $\boldsymbol{p}_i = \boldsymbol{Q}^\top \boldsymbol{v}_i$ and $\alpha_i = \alpha \cdot \lambda_i(\boldsymbol{N})$, we can express $\boldsymbol{T} = \boldsymbol{D} + \sum_{i=n-l+1}^{n} \alpha_i \cdot \boldsymbol{p}_i\boldsymbol{p}_i^\top$. Without loss of generality, consider the elements of the diagonal matrix $\boldsymbol{D}$ to be in non-decreasing order, i.e., $D_{11} = \lambda_1(\boldsymbol{M}) \leq D_{22} = \lambda_2(\boldsymbol{M}) \leq \ldots \leq D_{nn} = \lambda_n(\boldsymbol{M})$. Choose any $r \in \{n-l+1, \ldots, n\}$ and let $\widetilde{\boldsymbol{D}} = \mathrm{diag}(D_{11}, D_{22}, \ldots, D_{22})$, and $\widetilde{\boldsymbol{T}} = \widetilde{\boldsymbol{D}} + \alpha_r \cdot \boldsymbol{p}_r\boldsymbol{p}_r^\top$. Then, we have that $\lambda_1(\boldsymbol{T}) \geq \lambda_1(\widetilde{\boldsymbol{T}})$. Denote by $\tilde{\lambda}_i$ the eigenvalues of $\widetilde{\boldsymbol{T}}$, since $\boldsymbol{p}_r\boldsymbol{p}_r^\top$ is a rank-1 matrix and $\widetilde{\boldsymbol{D}}$ has only two different eigenvalues, we have that $\tilde{\lambda}_2 = \ldots = \tilde{\lambda}_{n-1} = D_{22}$. Now,

$$\begin{aligned} \tilde{\lambda}_1\tilde{\lambda}_n D_{22}^{n-2} = \det(\widetilde{\boldsymbol{D}} + \alpha_r \cdot \boldsymbol{p}_r\boldsymbol{p}_r^\top) &= \det(\widetilde{\boldsymbol{D}}) \det(\boldsymbol{I} + \alpha_r \cdot \widetilde{\boldsymbol{D}}^{-1}\boldsymbol{p}_r\boldsymbol{p}_r^\top) \\ &= (1 + \alpha_r \cdot \boldsymbol{p}_r^\top \widetilde{\boldsymbol{D}}^{-1}\boldsymbol{p}_r) \det(\widetilde{\boldsymbol{D}}) \\ &= D_{11}D_{22}^{n-1}\left(1 + \alpha_r \frac{p_{r_1}^2}{D_{11}} + \alpha_r \frac{1}{D_{22}}(1 - p_{r_1}^2)\right), \end{aligned}$$

where the third equality comes from $\det(\boldsymbol{I} + \boldsymbol{AB}) = \det(\boldsymbol{I} + \boldsymbol{BA})$, and the last equality is due to $\|\boldsymbol{p}_r\|_2 = 1$. Simplifying on both ends, we obtain:

$$\tilde{\lambda}_1 \tilde{\lambda}_n = \alpha_r D_{11} + D_{11} D_{22} + \alpha_r p_{r_1}^2 \Delta \tag{4.21}$$

From calculating the trace we have:

$$\tilde{\lambda}_1 + (n-2) D_{22} + \tilde{\lambda}_n = \operatorname{Tr}(\widetilde{\boldsymbol{T}}) = \operatorname{Tr}(\widetilde{\boldsymbol{D}}) + \alpha_r \operatorname{Tr}(\boldsymbol{p}_r \boldsymbol{p}_r^\top) = D_{11} + (n-1) D_{22} + \alpha_r.$$

Simplifying on both ends, we obtain:

$$\tilde{\lambda}_1 + \tilde{\lambda}_n = D_{11} + D_{22} + \alpha_r. \tag{4.22}$$

Combining eq.(4.21) and eq.(4.22), and simplifying for $\tilde{\lambda}_1$ we have, $\tilde{\lambda}_1 = D_{11} + \frac{\alpha_r + \Delta}{2} \pm \sqrt{(\frac{\alpha_r + \Delta}{2})^2 - \alpha_r \cdot \Delta \cdot p_{r_1}^2}$. Finally, since $\lambda_1(\boldsymbol{T}) \geq \lambda_1(\widetilde{\boldsymbol{T}}) = \tilde{\lambda}_1$ and the choice of r was arbitrary, we take the negative sign of the square root for a lower bound and we can maximize over the choice of r for the tightest lower bound. □

Remark 4.4.3. *Note that Lemma 4.4.2 is tighter than general eigenvalue inequalities such as Weyl's inequality. Lemma 4.4.2 is tight with respect to Δ in the sense that when $\boldsymbol{N}$ is rank-1 and $\Delta = 0$, i.e., when $\lambda_1(\boldsymbol{M}) = \lambda_2(\boldsymbol{M})$, our lower bound yields $\lambda_1(\boldsymbol{M})$, which is exactly the case as the minimum eigenvalue cannot be perturbed by a rank-1 matrix under this scenario. Similarly, our bound is tight with respect to α. When $\alpha = 0$, i.e., no perturbation, our lower bound results in $\lambda_1(\boldsymbol{M})$.*

Recall from Definition 4.2.1 that $\phi_{\mathcal{G}}$ is the *Cheeger constant* of $\mathcal{G}$, and let $\boldsymbol{L}$ be the Laplacian matrix of $\mathcal{G}$. Then, the second smallest eigenvalue of $\boldsymbol{L}$ and its respective eigenvector are known as the *algebraic connectivity* and the *Fiedler vector*[8], respectively. The following theorem corresponds to our main result for this section, where we formally show how the effect of the statistical parity constraint improves the probability of exact recovery.

[8]↑If the multiplicity of the algebraic connectivity is greater than one then we have a set of Fiedler vectors.

Theorem 4.4.4. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected connected graph with n nodes, Cheeger constant $\phi_{\mathcal{G}}$, Fiedler vector $\boldsymbol{\pi}_2$, and maximum node degree $\deg_{\max}(\mathcal{G})$. Also let Δ denote the gap between the third minimum and second minimum eigenvalue of the Laplacian of $\mathcal{G}$, namely, $\Delta = \lambda_3(\boldsymbol{L}) - \lambda_2(\boldsymbol{L})$. Let $\boldsymbol{N} = \sum_{i=1}^{k} \boldsymbol{a}_i \boldsymbol{a}_i^\top$ with eigenvalues $\lambda_i(\boldsymbol{N})$ and related eigenvectors $\boldsymbol{v}_i$ for $i \in [n]$. Then, for the combinatorial problem (4.19), a solution $\boldsymbol{y} \in \{\overline{\boldsymbol{y}}, -\overline{\boldsymbol{y}}\}$ is achievable in polynomial time by solving the SDP based relaxation (4.20), with probability at least $1 - 2n \cdot \mathrm{e}^{\frac{-3(\epsilon_1+\epsilon_2)^2}{24\sigma^2+8R(\epsilon_1+\epsilon_2)}}$, where*

$$\epsilon_1 = \max_{i=n-k+1\ldots n} \left(\frac{n\lambda_i(\boldsymbol{N}) + \Delta}{2} - \sqrt{\left(\frac{n\lambda_i(\boldsymbol{N}) + \Delta}{2}\right)^2 - n\lambda_i(\boldsymbol{N}) \cdot \Delta \cdot (\boldsymbol{v}_i^\top \boldsymbol{\pi}_2)^2} \right),$$

$$\epsilon_2 = (1-2p)\frac{\phi_{\mathcal{G}}^2}{4\deg_{\max}(\mathcal{G})}, \quad \sigma^2 = 4p(1-p)\deg_{\max}(\mathcal{G}), \quad R = 2(1-p),$$

and p is the edge noise from our model.

4.4.4 Discussion

We start by contrasting our result in Theorem 4.4.4 to that of Theorem 4.2.5. Following the notation in Theorem 4.4.4, Theorem 4.2.5 states that the probability of error for exact recovery is $2n \cdot \mathrm{e}^{\frac{-3\epsilon_2^2}{24\sigma^2+8R\epsilon_2}}$, while our result in Theorem 4.4.4 is $2n \cdot \mathrm{e}^{\frac{-3(\epsilon_1+\epsilon_2)^2}{24\sigma^2+8R(\epsilon_1+\epsilon_2)}}$. Then, we can conclude that, whenever $\epsilon_1 > 0$, the *probability of error* when adding a statistical parity constraint (our model) is *strictly less* than the case with no fairness constraint whatsoever (models studied in [17, 61, 73, 78]).

The above argument poses the question on when $\epsilon_1 > 0$. Recall from Theorem 4.4.4 that $\epsilon_1 = \max_{i=n-k+1\ldots n} \left(\frac{n\lambda_i(\boldsymbol{N})+\Delta}{2} - \sqrt{\left(\frac{n\lambda_i(\boldsymbol{N})+\Delta}{2}\right)^2 - n\lambda_i(\boldsymbol{N}) \cdot \Delta \cdot (\boldsymbol{v}_i^\top \boldsymbol{\pi}_2)^2} \right)$. For clarity purposes, we discuss the case of a single fairness constraint, that is, $\boldsymbol{N} = \boldsymbol{a}_1 \boldsymbol{a}_1^\top$, and let $\|\boldsymbol{a}_1\|_2^2 = s$. Then we have that $\epsilon_1 = \frac{n \cdot s + \Delta}{2} - \sqrt{\left(\frac{n \cdot s + \Delta}{2}\right)^2 - n \cdot \Delta \cdot (\boldsymbol{a}_1^\top \boldsymbol{\pi}_2)^2}$, from this expression, it is clear that whenever $\Delta > 0$ and $\langle \boldsymbol{a}_1, \boldsymbol{\pi}_2 \rangle \neq 0$ then $\epsilon_1 > 0$. In other words, to observe improvement in the probability of exact recovery, it suffices to have a non-zero scalar projection of the attribute $\boldsymbol{a}_1$ onto the Fiedler vector $\boldsymbol{\pi}_2$, and an algebraic connectivity of

multiplicity 1.[9] Finally, note that since $\langle \boldsymbol{a}_1, \boldsymbol{\pi}_2 \rangle$ depends on $\boldsymbol{a}_1$, which is a given attribute, one can safely assume that $\langle \boldsymbol{a}_1, \boldsymbol{\pi}_2 \rangle \neq 0$. However, the eigenvalue gap Δ depends solely on the graph $\mathcal{G}$ and raises the question on what classes of graphs we observe (or do not) $\Delta = 0$.

4.4.5 On the Multiplicity of the Algebraic Connectivity

Since $\Delta > 0$ if and only if the multiplicity of the algebraic connectivity is 1, we devote this section to discuss in which cases this condition does or does not occur. After the seminal work of Fiedler [123], which unveiled relationships between graph properties and the second minimum eigenvalue of the Laplacian matrix, several researchers aimed to find additional connections. In the graph theory literature, one can find analyses on the complete spectrum of the Laplacian (e.g., [100, 124–127]), where the main focus is to find bounds for the Laplacian eigenvalues based on structural properties of the graph. Another line of work studies the changes on the Laplacian eigenvalues after adding or removing edges in $\mathcal{G}$ [128–130]. To our knowledge the only work who attempts to characterize families of graphs that have algebraic connectivity with certain multiplicity is the work of Barik and Pati [130]. Let $\boldsymbol{\pi}$ be a Fiedler vector of $\mathcal{G}$, we denote the entry of $\boldsymbol{\pi}$ corresponding to vertex u as $\boldsymbol{\pi}_u$. A vertex u is called a *characteristic vertex* of $\mathcal{G}$ if $\boldsymbol{\pi}_u = 0$ and if there exists a vertex w adjacent to u such that $\boldsymbol{\pi}_w \neq 0$. An edge (u, w) is called a *characteristic edge* of $\mathcal{G}$ if $\boldsymbol{\pi}_u \boldsymbol{\pi}_w < 0$. The *characteristic set* of $\mathcal{G}$ is denoted by $\mathbb{C}_{\mathcal{G}}(\boldsymbol{\pi})$ and consists of all the characteristic vertices and characteristic edges of $\mathcal{G}$. Let $\mathbb{W}$ be any proper subset of the vertex set of $\mathcal{G}$, by a branch at $\mathbb{W}$ of $\mathcal{G}$ we mean a component of $\mathcal{G} \setminus \mathbb{W}$. A branch at $\mathbb{W}$ is called a *Perron branch* if the principal submatrix of $\boldsymbol{L}$, corresponding to the branch, has an eigenvalue less than or equal to $\lambda_2(\boldsymbol{L})$. The following was presented in [130] and characterizes graphs that have algebraic connectivity with certain multiplicity.

Theorem 4.4.5 (Theorem 10 in [130])**.** *Let $\mathcal{G}$ be a connected graph and $\boldsymbol{\pi}$ be a Fiedler vector with $\mathbb{W} = \mathbb{C}_{\mathcal{G}}(\boldsymbol{\pi})$ consisting of vertices only. Suppose that there are $t \geq 2$ Perron branches $\mathcal{G}_1, \ldots, \mathcal{G}_t$ of $\mathcal{G}$ at $\mathbb{W}$. Then the following are equivalent.*

1. *The multiplicity of $\lambda_2(\boldsymbol{L})$ is exactly $t - 1$.*

[9] ↑Specifically, we refer to the algebraic multiplicity. Having an algebraic connectivity with multiplicity greater than 1 will imply that $\Delta = 0$.

2. *For each Fiedler vector* $\boldsymbol{\psi}$, $\mathbb{C}_{\mathcal{G}}(\boldsymbol{\psi}) = \mathbb{W}$.

3. *For each Fiedler vector* $\boldsymbol{\psi}$, *the set* $\mathbb{C}_{\mathcal{G}}(\boldsymbol{\psi})$ *consists of vertices only.*

The above characterization is very limited in the sense that authors in [130] are able to show only one example of graph family that satisfies the conditions above. Specifically, their example correspond to the class $\mathcal{G} = (\mathcal{K}_{n-t}^C + \mathcal{H}_t^C)^C$, where $\mathcal{K}_i$ denotes the complete graph of order i and $\mathcal{H}_j$ is a graph of j isolated vertices, and for $\mathcal{G}_1 = (\mathcal{V}_1, \mathcal{E}_1), \mathcal{G}_2 = (\mathcal{V}_2, \mathcal{E}_2)$, the operation $\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_2$ is defined as $\mathcal{G} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E}_1 \cup \mathcal{E}_2)$. A particularly known instance of this class is $t = n - 1$, which corresponds to the star graph and has algebraic connectivity with multiplicity $n - 2$ and therefore $\Delta = 0$ for $n > 3$.

Another known example where $\Delta = 0$ is the complete graph $\mathcal{K}_n$ of order n where there is only one non-zero eigenvalue equal to n and with multiplicity $n - 1$. We now turn our attention to graphs with poor expansion properties such as grids. A $m \times n$ grid, denoted by $\text{Grid}(m, n)$, is a connected graph such that it has 4 corner vertices which have two edges each, $m - 2$ vertices that have 3 edges which make up the short "edge of a rectangle" and $n - 2$ vertices that have 3 edges each which make up the "long edge of a rectangle" and $(n - 2)(m - 2)$ inner vertices which each have four edges. Edwards [131] characterizes the full Laplacian spectrum for grid graphs as follows: the eigenvalues of the Laplacian matrix of $\text{Grid}(m, n)$ are of the form $\lambda_{i,j} = (2\sin(\frac{\pi i}{2n}))^2 + (2\sin(\frac{\pi j}{2m}))^2$, where i and j are non-negative integers. Next, we present a corollary showing the behavior of Δ in grids.

Corollary 4.4.6. *Let* $\mathcal{G}$ *be a grid graph,* $\text{Grid}(m, n)$*, then we have:*

- *If* $m = n$ *then* $\Delta = 0$.
- *If* $m \neq n$ *then* $\Delta > 0$.

Proof. Since $\lambda_{i,j} = (2\sin(\frac{\pi i}{2n}))^2 + (2\sin(\frac{\pi j}{2m}))^2$, then $\lambda_{i,j} = 0$ if and only if $(i, j) = (0, 0)$ and corresponds to the first eigenvalue of the Laplacian. It is clear that the next minimum should be of the form $\lambda_{0,j}$ and $\lambda_{i,0}$. By taking derivatives we obtain: $\frac{d\lambda_{i,0}}{di} = \frac{2\pi}{n}\sin(\frac{\pi i}{n})$ and $\frac{d\lambda_{0,j}}{dj} = \frac{2\pi}{m}\sin(\frac{\pi j}{m})$. We observe that the minimums are attained at $\lambda_{1,0} = (2\sin(\frac{\pi}{2n}))^2$ and $\lambda_{0,1} = (2\sin(\frac{\pi}{2m}))^2$ respectively. Thus, when $m = n$ we have $\Delta = 0$ and when $m \neq n$ we have $\Delta > 0$. □

That is, Corollary 4.4.6 states that square grids have $\Delta = 0$, while rectangular grids have $\Delta > 0$. To conclude our discussion on Δ, we empirically show that the family of Erdős-Rényi graphs exhibit $\Delta > 0$ with high probability. Specifically, we let $\mathcal{G} \sim \mathsf{ER}(n, r)$, where r is the edge probability. When $r = 1$, $\mathcal{G}$ is the complete graph of order n and $\Delta > 0$ with probability zero. Interestingly, when $r = 0.9$ or $r = 0.99$, that is, values close to 1, the probability of $\Delta > 0$ tends to 1 as n increases. Also, we analyze the case when $r = {}^{2\log n}/_{n}$,[10] and also observe high probability of $\Delta > 0$. The aforementioned results are depicted in Figure 4.5 (Left). Intuitively, this suggests that the family of graphs where $\Delta > 0$ is much larger than the families where $\Delta = 0$. Finally, in Figure 4.5 (Right), we also plot the expected value of the gap, where we note an interesting concentration of the gap to 0.5 for $r = {}^{2\log n}/_{n}$. An explanation of the latter gap behavior remains an open question.

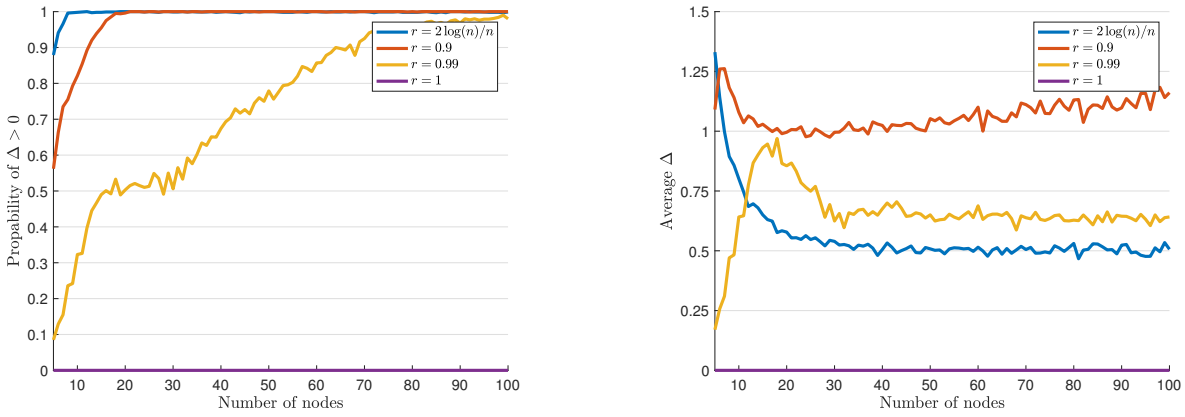


Figure 4.5. Graphs drawn from an Erdős-Rényi model with n nodes and edge probability r. *(Left)* Probability of $\Delta > 0$ for each number of nodes, we draw 1000 graphs and compute Δ, then, we count an event as success whenever $\Delta > 0$, and failure when $\Delta = 0$. *(Right)* Expected value of Δ computed across the 1000 random graphs for each number of nodes.

4.4.6 Experiments

In this section, we corroborate our theoretical results through synthetic experiments. Graphs with high expansion properties such as complete graphs and d-regular expanders are

[10]↑Our motivation for the choice of $r = {}^{2\log n}/_{n}$ is that for $r > {}^{(1+\varepsilon)\log n}/_{n}$ then the graph is connected almost surely [132].

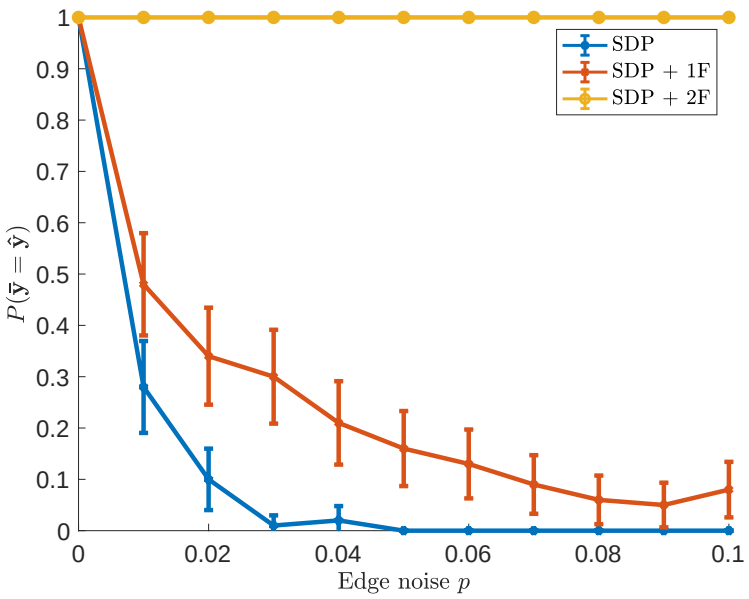


Figure 4.6. Probability of exact recovery for Grid$(4, 16)$ computed across 30 observations $\boldsymbol{X}$ for different values of $p \in [0, 0.1]$. We observe how the addition of fairness constraints helps exact recovery, where SDP+1F refers to the addition of a single constraint, and SDP+2F the addition of two constraints.

known to manifest high probability of exact recovery as their Cheeger constant increases with respect to n or d [17].

That is, in those graphs, the effect of the fairness constraint will not be noticeable. In contrast, graphs with poor expansion properties such as grids, which have a Cheeger constant in the order of $\mathcal{O}(1/n)$ for a Grid(n, n), can only be recovered approximately [61], or exactly if the graph can be perturbed with additional edges [17]. Thus, we focus our experiments on grids and empirically show how the inclusion of the fairness constraint boosts the probability of exact recovery. In Figure 4.6, we first randomly set $\bar{\boldsymbol{y}}$ by independently sampling each $\bar{y}_i$ from a Rademacher distribution. We consider a graph of 64 nodes, specifically, Grid$(4, 16)$, i.e., Δ is guaranteed to be greater than 0. Finally, we compute 30 observations for $p \in [0, 0.1]$. When there is no fairness constraint, we observe that the probability of exact recovery decreases at a very high rate, while the addition of fairness constraints improves the exact recovery probability. In particular, we note that while the addition of a *single* fairness constraint (SDP + 1F) helps to achieve exact recovery, the tendency is to still decrease as p increases, in this case the attribute $\boldsymbol{a}_1$ was randomly sampled from the nullspace of $\bar{\boldsymbol{y}}^\top$

so that $\overline{\boldsymbol{y}}^\top \boldsymbol{a}_1 = 0$. We also show the case when two fairness constraints are added (SDP + 2F), were we observe that exact recovery happens almost surely, here the two attributes also come randomly from the nullspace of $\overline{\boldsymbol{y}}^\top$.

4.5 Summary

In this chapter, we studied a generative model where we receive a single noisy observation for each edge and each node of a graph with the goal of recovering the ground-truth node labels exactly.

In Section 4.2, our approach consisted of two stages. The first stage consisted of solving solely the quadratic term of an optimization problem (based on a SDP relaxation) in order to find the structural properties of a graph that guarantee exact recovery with high probability. Given two possible solutions from the first stage, the second stage consisted in using solely the node observations and simply outputting the labeling with higher score. We showed that for any graph $\mathcal{G}$, the term $\phi^2_{\mathcal{G}}/\deg_{\max}(\mathcal{G})$ is related to achieving exact recovery in polynomial time. Examples include complete graphs and d-regular expanders, that are guaranteed to recover the correct labeling with high probability. While perhaps the most interesting example is related to smoothed analysis on connected graphs, where, even for a graph with bad expansion properties, the node labels can still be exactly recovered by adding small perturbations (edges coming from an Erdős-Rényi model with small probability).

In Section 4.3, we applied a powerful hierarchy of relaxations, known as the sum-of-squares (SoS) hierarchy, to the combinatorial problem. Motivated by empirical evidence on the improvement in exact recoverability, we centered our attention on the degree-4 SoS relaxation and set out to understand the origin of such improvement from a graph theoretical perspective. We showed that the solution of the dual of the relaxed problem is related to finding edge weights of the Johnson and Kneser graphs, where the weights fulfill the SoS constraints and intuitively allow the input graph to increase its algebraic connectivity. Finally, as byproduct of our analysis, we derived a novel Cheeger-type lower bound for the algebraic connectivity of graphs with *signed* edge weights.

In Section 4.4, we studied the effect of adding fairness constraints[11] to the aforementioned generative model, specifically, under a notion of statistical parity, and showed how they can help increase the probability of exact recovery even for graphs with poor expansion properties such as grids. In our analysis, we assumed that the ground-truth labeling is fair. While the linear constraints reduce the search space in the relaxed continuous problem, before our results, it was unclear how these constraints would affect the probability of exact recovery, which we formally show in Theorem 4.4.4. We argue that even in the scenario of having "fair data" one should not rule out the possibility of adding fairness constraints as there is a chance that it can help increase the performance. For instance, a practitioner could use one of the several *preprocessing* methods for debiasing a dataset with respect to a particular metric [133–136], assuming that the data is now fair, the practitioner might be tempted not to use any fairness constraint anymore. However, as showed in this chapter, when the data is fair, adding fairness constraints could improve performance.

[11]↑Note that, given that our definition of statistical parity is a linear constraint, our analysis and results will hold for any linear constraint not necessarily attached to a fairness viewpoint.

5. CONCLUSION

This dissertation took a detailed view at different combinatorial aspects of structured prediction problems. As structured prediction encompasses several important problems from different domains (e.g., computer vision, biology, social networks, and natural language processing to name a few), it is key to develop methods with strong theoretical guarantees.

In particular, in Chapter 2, we tackled on the problem of learning latent-variable models for structured prediction, where the key challenge we faced was the exponential size of the search space in the max-margin formulation. To that end, we proposed a computationally appealing method that allows for a fully polynomial time evaluation of the formulation, in cases where the margin can be computed in poly-time. Our work showed that the non-convex formulation using the slack re-scaling approach with latent variables is related to a tight upper bound of the Gibbs decoder distortion, and provided a tighter upper bound of the Gibbs decoder distortion by randomizing the search space of the optimization problem. Finally, we presented experimental results in synthetic data and in a computer vision application, where we obtained competitive results in average test error with respect to previous work, but with a much lower computational time.

In Chapter 3, we considered the problem of finding the necessary number of samples for learning of scoring functions based on factor graphs in the context of structured prediction. Our work was based on the minimax framework, and showed a lower bound that requires a new dimension (which we call MAX-PAIR-dimension) to be finite in order for a function class to be learnable. In addition, we showed a connection of the PAIR-dimension to the classical VC-dimension, for which there are several known results for different types of function classes. Finally, it remains an open question to analyze the optimality of our bound for general functions, where one possible attempt is perhaps to find an upper bound to the factor graph Rademacher complexity in terms of the PAIR-dimension, similar in spirit to the known result of the VC-dimension being an upper bound of the classical Rademacher complexity.

Lastly, in Chapter 4, we studied the statistical problem of exact inference in graphs, given noisy measurements. By formulating a SDP relaxation of the discrete combinatorial problem, which is solvable in polynomial time, we showed conditions on the input graph that suffice

to realize exact recovery. Those conditions relate to structural properties of the input graph such as the Cheeger constant. Moreover, one of the most intriguing examples we provided that satisfies such conditions is related to smoothed analysis on connected graphs. That is, given any connected graph that possibly does not fulfill the conditions, one can obtain a graph that satisfies the conditions by adding a few edges from an Erdős-Rényi graph. This understanding enables the possibility to drive modeling decisions, where graph models that satisfy such conditions are perhaps preferable. Finally, we also studied the problem under the SoS hierarchy of relaxations, and considered the effect of linear constraints.

In the above, we briefly discussed the different analyses presented in this dissertation. To obtain such results, we relied on techniques such as randomization, convex relaxations, minimax theory, and graph theory. While the aforementioned technical tools have been largely used by the machine learning community, this work employed such techniques to understand different aspects of structured prediction. Thus, the presented work includes the first set of results on characterizing the necessary number of samples for structured prediction, a randomized learning method with computational and statistical guarantees, and an analysis of the exact inference problem from a continuous relaxation viewpoint.

REFERENCES

[1] S. Lynch, *Andrew Ng: Why AI Is the New Electricity*, 2017. [Online]. Available: https://www.gsb.stanford.edu/insights/andrew-ng-why-ai-new-electricity.

[2] Y. Liu, E. Xing, and J. Carbonell, "Predicting protein folds with structural repeats using a chain graph model," *International Conference on Machine Learning*, pp. 513–520, 2005.

[3] Y. Liu, J. Carbonell, V. Gopalakrishnan, and P. Weigele, "Protein quaternary fold recognition using conditional graphical models," *International Joint Conference on Artificial Intelligence*, pp. 937–943, 2007.

[4] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell. 4th edition.* 2017. [Online]. Available: Available%20online%20at:%20https://www.ncbi.nlm.nih.gov/books/NBK21054.

[5] A. Martins, M. Almeida, and N. Smith, "Turning on the turbo: Fast third-order non-projective turbo parsers," *Annual Meeting of the Association for Computational Linguistics*, pp. 617–622, 2013.

[6] A. Rush, D. Sontag, M. Collins, and T. Jaakkola, "On dual decomposition and linear programming relaxations for natural language processing," *Empirical Methods in Natural Language Processing*, pp. 1–11, 2010.

[7] Y. Zhang, T. Lei, R. Barzilay, and T. Jaakkola, "Greed is good if randomized: New inference for dependency parsing," *Empirical Methods in Natural Language Processing*, pp. 1013–1024, 2014.

[8] Y. Zhang, C. Li, R. Barzilay, and K. Darwish, "Randomized greedy inference for joint segmentation, POS tagging and dependency parsing," *North American Chapter of the Association for Computational Linguistics*, pp. 42–52, 2015.

[9] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 1627–1645, 2010.

[10] V. Hedau, D. Hoiem, and D. Forsyth, "Recovering the spatial layout of cluttered rooms," *IEEE International Conference on Computer Vision*, pp. 1849–1856, 2009.

[11] D. C. Lee, A. Gupta, M. Hebert, and T. Kanade, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces," *Neural Information Processing Systems*, vol. 23, pp. 1288–1296, 2010.

[12] H. Wang, S. Gould, and D. Koller, "Discriminative learning with latent variables for cluttered indoor scene understanding," *European Conference on Computer Vision*, vol. 6312, pp. 435–449, 2010.

[13] J. Keshet, D. McAllester, and T. Hazan, "PAC-Bayesian approach for minimization of phoneme error rate," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2224–2227, 2011.

[14] S. Zhang and M. Gales, "Structured SVMs for automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 3, pp. 544–555, 2013.

[15] H. Tang, J. Keshet, and K. Livescu, "Discriminative pronunciation modeling: A large-margin, feature-rich approach," *Annual Meeting of the Association for Computational Linguistics*, pp. 194–203, 2012.

[16] K. Bello and J. Honorio, "Learning latent variable structured prediction models with gaussian perturbations," *NeurIPS*, 2018.

[17] K. Bello and J. Honorio, "Exact inference in structured prediction," *Advances in Neural Information Processing Systems*, 2019.

[18] K. Bello and J. Honorio, "Fairness constraints can help exact inference in structured prediction," *Advances in Neural Information Processing Systems*, 2020.

[19] K. Bello, A. Ghoshal, and J. Honorio, "Minimax bounds for structured prediction based on factor graphs," in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 213–222.

[20] K. Bello, C. Ke, and J. Honorio, "A thorough view of exact inference in graphs from the degree-4 sum-of-squares hierarchy," *arXiv preprint arXiv:2102.08019*, 2021.

[21] K. Bello and J. Honorio, "Computationally and statistically efficient learning of causal bayes nets using path queries," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 931–10 941.

[22] H. Wang, S. Gould, and D. Roller, "Discriminative learning with latent variables for cluttered indoor scene understanding," *Communications of the ACM*, vol. 56, no. 4, pp. 92–99, 2013.

[23] J. Honorio and T. Jaakkola, "Structured prediction: From gaussian perturbations to linear-time principled algorithms," in *Uncertainty in Artificial Intelligence*, 2016.

[24] D. McAllester, "Generalization bounds and consistency," in *Predicting Structured Data*, MIT Press, 2007, pp. 247–261.

[25] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell, "Hidden conditional random fields," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 10, 2007.

[26] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Computer Vision and Pattern Recognition*, IEEE, vol. 2, 2006, pp. 1521–1527.

[27] A. Quattoni, M. Collins, and T. Darrell, "Conditional random fields for object recognition," in *Advances in neural information processing systems*, 2005, pp. 1097–1104.

[28] S. Petrov and D. Klein, "Discriminative log-linear grammars with latent variables," in *Advances in neural information processing systems*, 2008, pp. 1153–1160.

[29] C. Yu and T. Joachims, "Learning structural SVMs with latent variables," *International Conference on Machine Learning*, pp. 1169–1176, 2009.

[30] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of machine learning research*, vol. 6, no. Sep, pp. 1453–1484, 2005.

[31] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (cccp)," in *Advances in neural information processing systems*, 2002, pp. 1033–1040.

[32] W. Ping, Q. Liu, and A. Ihler, "Marginal structured SVM with hidden variables," *International Conference on Machine Learning*, pp. 190–198, 2014.

[33] C. Cortes, V. Kuznetsov, M. Mohri, and S. Yang, "Structured prediction theory based on factor graph complexity," in *Advances in Neural Information Processing Systems*, 2016, pp. 2514–2522.

[34] Y. Altun and T. Hofmann, "Large margin methods for label sequence learning," *European Conference on Speech Communication and Technology*, pp. 145–152, 2003.

[35] M. Collins and B. Roark, "Incremental parsing with the perceptron algorithm," *Annual Meeting of the Association for Computational Linguistics*, pp. 111–118, 2004.

[36] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," *Neural Information Processing Systems*, vol. 16, pp. 25–32, 2003.

[37] G. E. Hinton, "A practical guide to training restricted boltzmann machines," in *Neural networks: Tricks of the trade*, Springer, 2012, pp. 599–619.

[38] A. Kulesza and F. Pereira, "Structured learning with approximate inference," *Neural Information Processing Systems*, vol. 20, pp. 785–792, 2007.

[39] B. London, O. Meshi, and A. Weller, "Bounding the integrality distance of lp relaxations for structured prediction," *NIPS workshop on Optimization for Machine Learning*, 2016.

[40] O. Meshi, M. Mahdavi, A. Weller, and D. Sontag, "Train and test tightness of lp relaxations in structured prediction," *International Conference on Machine Learning*, 2016.

[41] A. Gane, T. Hazan, and T. Jaakkola, "Learning with maximum a-posteriori perturbation models," in *Artificial Intelligence and Statistics*, 2014, pp. 247–256.

[42] M. Volkovs and R. S. Zemel, "Efficient sampling for bipartite matching problems," in *Advances in Neural Information Processing Systems*, 2012, pp. 1313–1321.

[43] S. Sarawagi and R. Gupta, "Accurate max-margin training for structured output spaces," in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 888–895.

[44] H. Choi, O. Meshi, and N. Srebro, "Fast and scalable structural svm with slack rescaling," in *Artificial Intelligence and Statistics*, 2016, pp. 667–675.

[45] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[46] C. Cortes, M. Mohri, and J. Weston, "A general regression framework for learning string-to-string mappings," *Predicting Structured Data*, vol. 2, no. 4, 2007.

[47] H. Daumé, J. Langford, and D. Marcu, "Search-based structured prediction," *Machine learning*, vol. 75, no. 3, pp. 297–325, 2009.

[48] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3156–3164, 2015.

[49] D. Jurafsky and J. H. Martin, *Speech and language processing*. Pearson London, 2014, vol. 3.

[50] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," *Advances in neural information processing systems*, pp. 2773–2781, 2015.

[51] D. Zhang, L. Sun, and W. Li, "A structured prediction approach for statistical machine translation," in *Proceedings of the Third International Joint Conference on Natural Language Processing*, 2008.

[52] C. Cortes, V. Kuznetsov, and M. Mohri, "Ensemble methods for structured prediction," in *International Conference on Machine Learning*, 2014, pp. 1134–1142.

[53] M. Collins, "Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods," in *New developments in parsing technology*, Springer, 2004, pp. 19–55.

[54] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *Advances in neural information processing systems*, 2004, pp. 25–32.

[55] A. Ghoshal and J. Honorio, "Learning maximum-a-posteriori perturbation models for structured prediction in polynomial time," in *International Conference on Machine Learning*, 2018.

[56] N. P. Santhanam and M. J. Wainwright, "Information-theoretic limits of selecting binary graphical models in high dimensions," *IEEE Transactions on Information Theory*, vol. 58, no. 7, pp. 4117–4134, 2012.

[57] R. Tandon, K. Shanmugam, P. K. Ravikumar, and A. G. Dimakis, "On the information theoretic limits of learning ising models," in *Advances in Neural Information Processing Systems*, 2014, pp. 2303–2311.

[58] W. Wang, M. J. Wainwright, and K. Ramchandran, "Information-theoretic bounds on model selection for gaussian markov random fields," in *IEEE International Symposium on Information Theory*, 2010, pp. 1373–1377.

[59] V. Vapnik, *The nature of statistical learning theory*. Springer science & business media, 2013.

[60] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., 2002, pp. 485–492.

[61] A. Globerson, T. Roughgarden, D. Sontag, and C. Yildirim, "How hard is inference for structured prediction?" In *International Conference on Machine Learning*, 2015.

[62] M. J. Wainwright, *High-dimensional statistics: A non-asymptotic viewpoint*. Cambridge University Press, 2019.

[63] L. Wasserman, *All of nonparametric statistics*. Springer Science & Business Media, 2006.

[64] A. Wald, "Contributions to the theory of statistical estimation and testing hypotheses," *The Annals of Mathematical Statistics*, vol. 10, no. 4, pp. 299–326, 1939.

[65] P. Massart, É. Nédélec, *et al.*, "Risk bounds for statistical learning," *The Annals of Statistics*, vol. 34, no. 5, pp. 2326–2366, 2006.

[66] A. Daniely, S. Sabato, S. Ben-David, and S. Shalev-Shwartz, "Multiclass learnability and the erm principle," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2377–2404, 2015.

[67] S. Ben-David, N. Cesa-Bianchi, D. Haussler, and P. M. Long, "Characterizations of learnability for classes of $\{0, \ldots, n\}$-valued functions," *Journal of Computer and System Sciences*, vol. 50, no. 1, pp. 74–86, 1995.

[68] T. Neylon, "Sparse solutions for linear prediction problems," Ph.D. dissertation, New York University, May 2006.

[69] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[70] V. Chandrasekaran, N. Srebro, and P. Harsha, "Complexity of inference in graphical models," in *Proceedings of the Twenty-Fourth Conference on Uncertainty in Artificial Intelligence*, 2008.

[71] N. N. Schraudolph and D. Kamenetsky, "Efficient exact inference in planar ising models," in *Advances in Neural Information Processing Systems*, 2009, pp. 1417–1424.

[72] Y. Boykov and O. Veksler, "Graph cuts in vision and graphics: Theories and applications," in *Handbook of mathematical models in computer vision*, Springer, 2006, pp. 79–96.

[73] D. Foster, K. Sridharan, and D. Reichman, "Inference in sparse graphs with pairwise measurements and side information," in *International Conference on Artificial Intelligence and Statistics*, 2018, pp. 1810–1818.

[74] D. Sontag, D. K. Choe, and Y. Li, "Efficiently searching for frustrated cycles in map inference," *arXiv preprint arXiv:1210.4902*, 2012.

[75] T. Koo, A. M. Rush, M. Collins, T. Jaakkola, and D. Sontag, "Dual decomposition for parsing with non-projective head automata," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, 2010, pp. 1288–1298.

[76] V. Chandrasekaran and M. I. Jordan, "Computational and statistical tradeoffs via convex relaxation," *Proceedings of the National Academy of Sciences*, p. 201 302 293, 2013.

[77] Y. Chen, G. Kamath, C. Suh, and D. Tse, "Community recovery in graphs with locality," in *International Conference on Machine Learning*, 2016, pp. 689–698.

[78] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Transactions on Information Theory*, 2016.

[79] F. Barahona, "On the computational complexity of ising spin glass models," *Journal of Physics A: Mathematical and General*, 1982.

[80] J. Cheeger, "A lower bound for the smallest eigenvalue of the laplacian," in *Proceedings of the Princeton conference in honor of Professor S. Bochner*, 1969.

[81] J. A. Tropp, "User-friendly tail bounds for sums of random matrices," *Foundations of computational mathematics*, 2012.

[82] M. Krivelevich, D. Reichman, and W. Samotij, "Smoothed analysis on connected graphs," *SIAM Journal on Discrete Mathematics*, 2015.

[83] E. Abbe, A. S. Bandeira, A. Bracher, and A. Singer, "Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery," *IEEE Transactions on Network Science and Engineering*, 2014.

[84] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.

[85] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization," Ph.D. dissertation, California Institute of Technology, 2000.

[86] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on optimization*, 2001.

[87] B. Barak and D. Steurer, "Sum-of-squares proofs and the quest toward optimal algorithms," *ArXiv 1404.5236*, 2014.

[88] M. Laurent, "A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0–1 programming," *Mathematics of Operations Research*, 2003.

[89] R. Meka, A. Potechin, and A. Wigderson, "Sum-of-squares lower bounds for planted clique," in *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, 2015, pp. 87–96.

[90] A. S. Bandeira and D. Kunisky, "A gramian description of the degree 4 generalized elliptope," *ArXiv preprint ArXiv:1812.11583*, 2018.

[91] D. Cifuentes, C. Harris, and B. Sturmfels, "The geometry of sdp-exactness in quadratic optimization," *Mathematical Programming*, vol. 182, no. 1, pp. 399–428, 2020.

[92] T. Weisser, J.-B. Lasserre, and K.-C. Toh, "A bounded degree sos hierarchy for large scale polynomial optimization with sparsity," 2016.

[93] M. A. Erdogdu, Y. Deshpande, and A. Montanari, "Inference in graphical models via semidefinite programming hierarchies," *Advances in Neural Information Processing Systems*, 2017.

[94] M. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, 1995.

[95] Y. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optimization methods and software*, 1998.

[96] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging applications of algebraic geometry*, Springer, 2009.

[97] D. Holton and J. Sheehan, *The Petersen Graph*, ser. Australian Mathematical Society Lecture Series. Cambridge University Press, 1993.

[98] L. Lovász, "Kneser's conjecture, chromatic number, and homotopy," *Journal of Combinatorial Theory, Series A*, 1978.

[99] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[100] B. Mohar, "The laplacian spectrum of graphs," *Graph theory, combinatorics, and applications*, 1991.

[101] D. Zelazo and M. Bürger, "On the definiteness of the weighted laplacian and its connection to effective resistance," in *53rd IEEE Conference on Decision and Control*, IEEE, 2014.

[102] Y. Chen, S. Z. Khong, and T. T. Georgiou, "On the definiteness of graph laplacians with negative weights: Geometrical and passivity-based approaches," in *2016 American Control Conference (ACC)*, IEEE, 2016.

[103] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak, "Spectral analysis of signed graphs for clustering, prediction and visualization," in *Proceedings of the 2010 SIAM International Conference on Data Mining*, SIAM, 2010, pp. 559–570.

[104] P. Mercado, F. Tudisco, and M. Hein, "Clustering signed networks with the geometric mean of laplacians," *NIPS 2016-Neural Information Processing Systems*, 2016.

[105] M. Cucuringu, P. Davies, A. Glielmo, and H. Tyagi, "Sponge: A generalized eigenproblem for clustering signed networks," in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 1088–1098.

[106] K.-Y. Chiang, J. J. Whang, and I. S. Dhillon, "Scalable clustering of signed networks using balance normalized cut," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, 2012, pp. 615–624.

[107] A. V. Knyazev, "Signed laplacian for spectral clustering revisited," *arXiv preprint arXiv:1701.01394*, vol. 1, 2017.

[108] F. M. Atay and S. Liu, "Cheeger constants, structural balance, and spectral clustering analysis for signed graphs," *Discrete Mathematics*, vol. 343, no. 1, p. 111 616, 2020.

[109] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan, "Human decisions and machine predictions," *The quarterly journal of economics*, vol. 133, no. 1, pp. 237–293, 2018.

[110] P. Gajane and M. Pechenizkiy, "On formalizing fairness in prediction with machine learning," *arXiv preprint arXiv:1710.03184*, 2017.

[111] S. Verma and J. Rubin, "Fairness definitions explained," in *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, IEEE, 2018, pp. 1–7.

[112] S. Barocas and A. D. Selbst, "Big data's disparate impact," *Calif. L. Rev.*, vol. 104, p. 671, 2016.

[113] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact," in *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 259–268.

[114] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*, 2016.

[115] A. Agarwal, M. Dudik, and Z. S. Wu, "Fair Regression: Quantitative Definitions and Reduction-based Algorithms," *arXiv preprint arXiv:1905.12843*, 2019.

[116] K. D. Johnson, D. P. Foster, and R. A. Stine, "Impartial predictive modeling: Ensuring fairness in arbitrary models," *arXiv preprint arXiv:1608.00528*, 2016.

[117] T. Calders, A. Karim, F. Kamiran, W. Ali, and X. Zhang, "Controlling attribute effect in linear regression," in *2013 IEEE 13th international conference on data mining*, IEEE, 2013, pp. 71–80.

[118] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," in *Advances in neural information processing systems*, 2016, pp. 3315–3323.

[119] B. T. Luong, S. Ruggieri, and F. Turini, "K-nn as an implementation of situation testing for discrimination discovery and prevention," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 502–510.

[120] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, and H. Wallach, "A reductions approach to fair classification," *arXiv preprint arXiv:1803.02453*, 2018.

[121] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Advances in Neural Information Processing Systems*, 2017, pp. 5029–5037.

[122] M. R. Garey and D. S. Johnson, *Computers and intractability*. freeman San Francisco, 1979, vol. 174.

[123] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak mathematical journal*, vol. 23, no. 2, pp. 298–305, 1973.

[124] R. Grone, R. Merris, and V. S. Sunder, "The laplacian spectrum of a graph," *SIAM Journal on matrix analysis and applications*, vol. 11, no. 2, pp. 218–238, 1990.

[125] R. Grone and R. Merris, "The laplacian spectrum of a graph ii," *SIAM Journal on discrete mathematics*, vol. 7, no. 2, pp. 221–229, 1994.

[126] M. W. Newman, "The laplacian spectrum of graphs," 2001.

[127] K. C. Das, "The laplacian spectrum of a graph," *Computers & Mathematics with Applications*, vol. 48, no. 5-6, pp. 715–724, 2004.

[128] S. Kirkland, "Completion of laplacian integral graphs via edge addition," *Discrete mathematics*, vol. 295, no. 1-3, pp. 75–90, 2005.

[129] S. Kirkland, "Algebraic connectivity for vertex-deleted subgraphs, and a notion of vertex centrality," *Discrete Mathematics*, vol. 310, no. 4, pp. 911–921, 2010.

[130] S. Barik and S. Pati, "On algebraic connectivity and spectral integral variations of graphs," *Linear algebra and its applications*, vol. 397, pp. 209–222, 2005.

[131] T. Edwards, *The discrete laplacian of a rectangular grid*, 2013.

[132] P. Erdős and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hung. Acad. Sci*, vol. 5, no. 1, pp. 17–60, 1960.

[133] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork, "Learning fair representations," in *International Conference on Machine Learning*, 2013, pp. 325–333.

[134] F. Calmon, D. Wei, B. Vinzamuri, K. N. Ramamurthy, and K. R. Varshney, "Optimized pre-processing for discrimination prevention," in *Advances in Neural Information Processing Systems*, 2017, pp. 3992–4001.

[135] C. Louizos, K. Swersky, Y. Li, M. Welling, and R. Zemel, "The variational fair autoencoder," *arXiv preprint arXiv:1511.00830*, 2015.

[136] P. Gordaliza, E. Del Barrio, G. Fabrice, and J.-M. Loubes, "Obtaining fairness using optimal transport theory," in *International Conference on Machine Learning*, 2019, pp. 2357–2365.

[137] J. Bennett, "Determination of the number of independent parameters of a score matrix from the examination of rank orders," *Psychometrika*, vol. 21, no. 4, pp. 383–393, 1956.

[138] J. Bennett and W. Hays, "Multidimensional unfolding: Determining the dimensionality of ranked preference data," *Psychometrika*, vol. 25, no. 1, pp. 27–43, 1960.

[139] T. Cover, "The number of linearly inducible orderings of points in d-space," *SIAM Journal on Applied Mathematics*, vol. 15, no. 2, pp. 434–439, 1967.

[140] M. Grant and S. Boyd, *CVX: Matlab software for disciplined convex programming, version 2.1*, Mar. 2014.

A. APPENDIX TO CHAPTER 2

A.1 Proof of Theorem 2.3.1

First, we derive an intermediate lemma needed for the final proof.

Lemma A.1.1 (Adapted from Lemma 5 in [24]). *Assume that there exists a finite integer value r such that, $|\mathcal{Y}_x \times \mathcal{H}_x| \leq r$ for all $(x, y) \in S$. Assume also that $\|\mathbf{\Phi}(x, y, h)\|_2 \leq \gamma$ for any triple (x, y, h). Let $Q(\boldsymbol{w})$ be a unit-variance Gaussian distribution centered at $\alpha\boldsymbol{w}$ for $\alpha = \gamma\sqrt{8 \log \frac{rn}{\|\boldsymbol{w}\|_2^2}}$. Then for all $(x, y) \in S$, and all $\boldsymbol{w} \in \mathcal{W}$, we have:*

$$\mathop{\mathbb{P}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [m(x, y, (f_{\boldsymbol{w}'}(x)), \boldsymbol{w}) \geq 1] \leq \|\boldsymbol{w}\|_2^2/n$$

or equivalently:

$$\mathop{\mathbb{P}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [m(x, y, (f_{\boldsymbol{w}'}(x)), \boldsymbol{w}) \leq 1] \geq 1 - \|\boldsymbol{w}\|_2^2/n \tag{A.1}$$

Proof. Note that the randomness in the statement comes from the variable $\boldsymbol{w}'$, then by a union bound on the elements of $\mathcal{Y}_x \times \mathcal{H}_x$ it suffices to show that for any given $(\widehat{y}, \hat{h})$ with $m(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \geq 1$, the probability that $f_{\boldsymbol{w}'}(x) = (\widehat{y}, \hat{h})$ is at most $\|\boldsymbol{w}\|_2^2/(rn)$.

Consider a fixed $(\widehat{y}, \hat{h}) \in \mathcal{Y}_x \times \mathcal{H}_x$ with $m(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \geq 1$. First, by well-know concentration inequalities we have that for any vector $\mathbf{\Psi} \in \mathbb{R}^\ell$ with $\|\mathbf{\Psi}\|_2 = 1$ and $\epsilon \geq 0$:

$$\mathop{\mathbb{P}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [\langle (\alpha\boldsymbol{w} - \boldsymbol{w}'), \mathbf{\Psi} \rangle \geq \epsilon] \leq \mathrm{e}^{-\epsilon^2/2} \tag{A.2}$$

Let $h^* = \arg\max_{h \in \mathcal{H}_x} \langle \mathbf{\Phi}(x, y, h), \boldsymbol{w} \rangle$, and let $\Delta(x, y, h^*, \widehat{y}, \hat{h}) = \mathbf{\Phi}(x, y, h^*) - \mathbf{\Phi}(x, \widehat{y}, \hat{h})$. Then, $m(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) = \Delta(x, y, h^*, \widehat{y}, \hat{h}) \cdot \boldsymbol{w}$.
Using $\mathbf{\Psi} = \Delta(x, y, h^*, \widehat{y}, \hat{h})/\|\Delta(x, y, h^*, \widehat{y}, \hat{h})\|_2$ in (A.2) we have:

$$\mathop{\mathbb{P}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [m(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}') \leq \alpha m(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) - \epsilon \|\Delta(x, y, h^*, \widehat{y}, \hat{h})\|_2] \leq \mathrm{e}^{-\epsilon^2/2}$$

$$\mathop{\mathbb{P}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [m(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}') \leq \alpha - \epsilon \|\Delta(x, y, h^*, \widehat{y}, \hat{h})\|_2] \leq \mathrm{e}^{-\epsilon^2/2}$$

$$\mathop{\mathbb{P}}_{\boldsymbol{w}'\sim Q(\boldsymbol{w})}\left[m(x,y,\widehat{y},\hat{h},\boldsymbol{w}')\leq 0\right]\leq \mathrm{e}^{-\alpha^2/(8\gamma^2)} \tag{A.3}$$

$$\mathop{\mathbb{P}}_{\boldsymbol{w}'\sim Q(\boldsymbol{w})}\left[f_{\boldsymbol{w}'}(x)=(\widehat{y},\hat{h})\right]\leq \mathrm{e}^{-\alpha^2/(8\gamma^2)}$$

where the step in (A.3) follows from $\epsilon=\alpha/\|\Delta(x,y,h^*,\widehat{y},\hat{h})\|_2$ and $\|\Delta(x,y,h^*,\widehat{y},\hat{h})\|_2\leq 2\gamma$. Thus, we prove our claim. □

Next, we provide the final proof.

Proof of Theorem 2.3.1. Define the Gibbs decoder *empirical* distortion of the perturbation distribution $Q(\boldsymbol{w})$ and training set S as:

$$L(Q(\boldsymbol{w}),S)=\frac{1}{n}\sum_{(x,y)\in S}\mathop{\mathbb{E}}_{\boldsymbol{w}'\sim Q(\boldsymbol{w})}\left[d(y,(f_{\boldsymbol{w}'}(x)))\right]$$

In PAC-Bayes terminology, $Q(\boldsymbol{w})$ is the *posterior* distribution. Let the *prior* distribution P be the unit-variance zero-mean Gaussian distribution. Fix $\delta\in(0,1)$ and $\alpha>0$. By well-known PAC-Bayes proof techniques, Lemma 4 in [24] shows that with probability at least $1-\delta/2$ over the choice of n training samples, simultaneously for all parameters $\boldsymbol{w}\in\mathcal{W}$, and unit-variance Gaussian posterior distributions $Q(\boldsymbol{w})$ centered at $\boldsymbol{w}\alpha$, we have:

$$\begin{aligned}L(Q(\boldsymbol{w}),D)&\leq L(Q(\boldsymbol{w}),S)+\sqrt{\frac{KL(Q(\boldsymbol{w})\|P)+\log\left(2n/\delta\right)}{2(n-1)}}\\&=L(Q(\boldsymbol{w}),S)+\sqrt{\frac{\|w\|_2^2\,\alpha^2/2+\log\left(2n/\delta\right)}{2(n-1)}}\end{aligned} \tag{A.4}$$

Thus, an upper bound of $L(Q(\boldsymbol{w}),S)$ would lead to an upper bound of $L(Q(\boldsymbol{w}),D)$. In order to upper-bound $L(Q(\boldsymbol{w}),S)$, we can upper-bound each of its summands, i.e., we can upper-bound $\mathbb{E}_{\boldsymbol{w}'\sim Q(\boldsymbol{w})}\left[d(y,(f_{\boldsymbol{w}'}(x)))\right]$ for each $(x,y)\in S$. Define the distribution $Q(\boldsymbol{w},x)$ with support on $\mathcal{Y}_x\times\mathcal{H}_x$ in the following form for all $y\in\mathcal{Y}_x$ and $h\in\mathcal{H}_x$:

$$\mathop{\mathbb{P}}_{(y',h')\sim Q(\boldsymbol{w},x)}\left[(y',h')=(y,h)\right]\equiv\mathop{\mathbb{P}}_{\boldsymbol{w}'\sim Q(\boldsymbol{w})}\left[f_{\boldsymbol{w}'}(x)=(y,h)\right] \tag{A.5}$$

For clarity of presentation, define:

$$u(x, y, y', h', \boldsymbol{w}) \equiv 1 - m(x, y, y', h', \boldsymbol{w})$$

Let $u \equiv u(x, y, (f_{\boldsymbol{w}'}(x)), \boldsymbol{w})$. Simultaneously for all $(x, y) \in S$, we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [d(y, (f_{\boldsymbol{w}'}(x))] &= \mathop{\mathbb{E}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [d(y, (f_{\boldsymbol{w}'}(x)))\ \mathbb{1}[u \geq 0] + d(y, (f_{\boldsymbol{w}'}(x)))\ \mathbb{1}[u < 0]\,] \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [d(y, (f_{\boldsymbol{w}'}(x)))\ \mathbb{1}[u \geq 0] + \mathbb{1}[u < 0]\,] && \text{(A.6.a)} \\
&= \mathop{\mathbb{E}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} d(y, (f_{\boldsymbol{w}'}(x))\ \mathbb{1}[u \geq 0] + \mathop{\mathbb{P}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [u < 0] \\
&\leq \mathop{\mathbb{E}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} d(y, (f_{\boldsymbol{w}'}(x))\ \mathbb{1}[u \geq 0] + \|\boldsymbol{w}\|_2^2 / n && \text{(A.6.b)} \\
&= \mathop{\mathbb{E}}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} d(y, (f_{\boldsymbol{w}'}(x))\ \mathbb{1}[u(x, y, (f_{\boldsymbol{w}'}(x)), \boldsymbol{w}) \geq 0] + \|\boldsymbol{w}\|_2^2 / n \\
&= \mathop{\mathbb{E}}_{(y', h') \sim Q(\boldsymbol{w}, x)} d(y, y', h')\ \mathbb{1}\Big[u(x, y, y', h', \boldsymbol{w}) \geq 0\Big] + \|\boldsymbol{w}\|_2^2 / n && \text{(A.6.c)} \\
&\leq \max_{(\widehat{y}, \hat{h}) \in \mathcal{Y}_x \times \mathcal{H}_x} d(y, \widehat{y}, \hat{h})\ \mathbb{1}\Big[u(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \geq 0\Big] + \|\boldsymbol{w}\|_2^2 / n && \text{(A.6.d)}
\end{aligned}$$

where the step in eq.(A.6.a) holds since $d : \mathcal{Y} \times \mathcal{Y} \times \mathcal{H} \to [0, 1]$. The step in eq.(A.6.b) follows from Lemma A.1.1 which states that $\mathbb{P}_{\boldsymbol{w}' \sim Q(\boldsymbol{w})} [u(x, y, (f_{\boldsymbol{w}'}(x)), \boldsymbol{w}) < 0] \leq \|\boldsymbol{w}\|_2^2 / n$ for $\alpha = \gamma \sqrt{8 \log (rn / \|\boldsymbol{w}\|_2^2)}$, for all $(x, y) \in S$ and all $\boldsymbol{w} \in \mathcal{W}$. By the definition in eq.(A.5), then the step in eq.(A.6.c) holds. Let $\lambda : \mathcal{Y} \times \mathcal{H} \to [0, 1]$ be some arbitrary function, the step in eq.(A.6.d) uses the fact that $\mathbb{E}_{(y,h)} [\lambda(y, h)] \leq \max_{(y,h)} \lambda(y, h)$.
By eq.(A.4) and eq.(A.6.d), we prove our claim. □

A.2 Proof of Theorem 2.3.4

Proof. The proof follows similar steps to that of Theorem 2.3.1. Note that the relaxed margin, $\widetilde{m}$, also fulfills the bound in Lemma A.1.1. Hence, following the steps of Proof A.1 we obtain an upper bound with same constants. □

A.3 Proof of Theorem 2.4.4

First, we derive an intermediate lemma needed for the final proof.

Lemma A.3.1. *Let $\Delta \in \mathbb{R}^\ell$ be a random variable with $\|\Delta\|_2 \leq 2\gamma$, and $\boldsymbol{w} \in \mathbb{R}^\ell$ be a constant. If $\langle \mathbb{E}[\Delta], \boldsymbol{w} \rangle \leq 1/2$ then we have:*

$$\mathbb{P}\left[\langle \Delta, \boldsymbol{w} \rangle > 1\right] \leq \exp\left(\frac{-1}{128\gamma^2 \|\boldsymbol{w}\|_2^2}\right)$$

Proof. Let $t > 0$, we have that:

$$\begin{aligned}
\mathbb{P}\left[\langle \Delta, \boldsymbol{w} \rangle > 1\right] &= \mathbb{P}\left[\langle (\Delta - \mathbb{E}[\Delta]), \boldsymbol{w} \rangle > 1 - \langle \mathbb{E}[\Delta], \boldsymbol{w} \rangle\right] \\
&\leq \mathbb{P}\left[\langle (\Delta - \mathbb{E}[\Delta]), \boldsymbol{w} \rangle \geq 1/2\right] && \text{(A.7.b)} \\
&= \mathbb{P}\left[\exp\left(t\langle (\Delta - \mathbb{E}[\Delta]), \boldsymbol{w} \rangle\right) \geq \mathrm{e}^{t/2}\right] \\
&\leq \mathrm{e}^{-t/2}\, \mathbb{E}\left[\exp\left(t\langle (\Delta - \mathbb{E}[\Delta]), \boldsymbol{w} \rangle\right)\right] && \text{(A.7.c)} \\
&\leq \exp\left(-t/2 + 8t^2\gamma^2\|\boldsymbol{w}\|_2^2\right) && \text{(A.7.d)}
\end{aligned}$$

The step in eq.(A.7.b) follows from $\langle \mathbb{E}[\Delta], \boldsymbol{w} \rangle \leq 1/2$ and thus $1 - \langle \mathbb{E}[\Delta], \boldsymbol{w} \rangle \geq 1/2$. The step in eq.(A.7.c) follows from Markov's inequality. The step in eq.(A.7.d) follows from Hoeffding's lemma and the fact that the random variable $z = \langle (\Delta - \mathbb{E}[\Delta]), \boldsymbol{w} \rangle$ fulfills $\mathbb{E}[z] = 0$ as well as $z \in [-4\gamma\|\boldsymbol{w}\|_2, +4\gamma\|\boldsymbol{w}\|_2]$. In more detail, note that $\|\Delta\|_2 \leq 2\gamma$ and by Jensen's inequality $\|\mathbb{E}[\Delta]\|_2 \leq \mathbb{E}[\|\Delta\|_2] \leq 2\gamma$. Then, note that by Cauchy-Schwarz inequality $|\langle (\Delta - \mathbb{E}[\Delta]), \boldsymbol{w} \rangle| \leq \|\Delta - \mathbb{E}[\Delta]\|_2 \|\boldsymbol{w}\|_2 \leq (\|\Delta\|_2 + \|\mathbb{E}[\Delta]\|_2)\|\boldsymbol{w}\|_2 \leq 4\gamma\|\boldsymbol{w}\|_2$. Finally, let $g(t) = -t/2 + 8t^2\gamma^2\|\boldsymbol{w}\|_2^2$. By making $\partial g / \partial t = 0$, we get the optimal setting $t^* = 1/(32\gamma^2\|\boldsymbol{w}\|_2^2)$. Thus, $g(t^*) = -1/(128\gamma^2\|\boldsymbol{w}\|_2^2)$ and we prove our claim. □

Next, we provide the final proof.

Proof of Theorem 2.4.4. Note that sampling from the distribution $Q(\boldsymbol{w}, x)$ as defined in eq.(A.5) is NP-hard in general, thus our plan is to upper-bound the expectation in eq.(A.6.c) by using the maximum over random structured outputs and latent variables sampled independently from a proposal distribution $R(\boldsymbol{w}, x)$ with support on $\mathcal{Y}_x \times \mathcal{H}_x$.

Let $T(\boldsymbol{w}, x)$ be a set of n' i.i.d. random structured outputs and latent variables drawn from the proposal distribution $R(\boldsymbol{w}, x)$, i.e., $T(\boldsymbol{w}, x) \sim R(\boldsymbol{w}, x)^{n'}$. Furthermore, let $\mathbb{T}(\boldsymbol{w})$ be the collection of the n sets $T(\boldsymbol{w}, x)$ for all $(x, y) \in S$, i.e., $\mathbb{T}(\boldsymbol{w}) \equiv \{T(\boldsymbol{w}, x)\}_{(x,y)\in S}$ and thus $\mathbb{T}(w) \sim \{R(\boldsymbol{w}, x)^{n'}\}_{(x,y)\in S}$. For clarity of presentation, define:

$$v(x, y, y', h', \boldsymbol{w}) \equiv d(y, y', h') \; \mathbb{1}\Big[\widetilde{m}(x, y, y', h', \boldsymbol{w}) \leq 1\Big]$$

For sets $T(\boldsymbol{w}, x)$ of sufficient size n', our goal is to upper-bound eq.(A.6.c) in the following form for all parameters $\boldsymbol{w} \in \mathcal{W}$:

$$\frac{1}{n} \sum_{(x,y)\in S} \mathop{\mathbb{E}}_{(y',h')\sim Q(\boldsymbol{w},x)} [v(x, y, y', h', \boldsymbol{w})] \leq \frac{1}{n} \sum_{(x,y)\in S} \max_{(\widehat{y},\hat{h})\in T(\boldsymbol{w},x)} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) + \mathcal{O}(\log^2 n/\sqrt{n})$$

Note that the above expression would produce a tighter upper bound than the maximum loss over all possible structured outputs and latent variables since $\max_{(\widehat{y},\hat{h})\in T(\boldsymbol{w},x)} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \leq \max_{(\widehat{y},\hat{h})\in \mathcal{Y}_x\times\mathcal{H}_x} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w})$. For analysis purposes, we decompose the latter equation into two quantities:

$$A(\boldsymbol{w}, S) \equiv \frac{1}{n} \sum_{(x,y)\in S} \left(\mathop{\mathbb{E}}_{(y',h')\sim Q(\boldsymbol{w},x)} [v(x, y, y', h', \boldsymbol{w})] - \mathop{\mathbb{E}}_{T(\boldsymbol{w},x)\sim R(\boldsymbol{w},x)^{n'}} \left[\max_{(\widehat{y},\hat{h})\in T(\boldsymbol{w},x)} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \right] \right) \tag{A.8}$$

$$B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) \equiv \frac{1}{n} \sum_{(x,y)\in S} \left(\mathop{\mathbb{E}}_{T(\boldsymbol{w},x)\sim R(\boldsymbol{w},x)^{n'}} \left[\max_{(\widehat{y},\hat{h})\in T(\boldsymbol{w},x)} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \right] - \max_{(\widehat{y},\hat{h})\in T(\boldsymbol{w},x)} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \right) \tag{A.9}$$

Thus, we will show that $A(\boldsymbol{w}, S) \leq \sqrt{1/n}$ and $B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) \leq \mathcal{O}(\log^2 n/\sqrt{n})$ for all parameters $\boldsymbol{w} \in \mathcal{W}$, any training set S and all collections $\mathbb{T}(\boldsymbol{w})$, and therefore $A(\boldsymbol{w}, S)+B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) \leq \mathcal{O}(\log^2 n/\sqrt{n})$. Note that while the value of $A(\boldsymbol{w}, S)$ is deterministic, the value of $B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w}))$ is stochastic given that $\mathbb{T}(\boldsymbol{w})$ is a collection of sampled random structured outputs.

Fix a specific $\boldsymbol{w} \in \mathcal{W}$. If data is separable then $v(x, y, y', h', \boldsymbol{w}) = 0$ for all $(x, y) \in S$ and $(y', h') \in \mathcal{Y}_x \times \mathcal{H}_x$. Thus, we have $A(\boldsymbol{w}, S) = B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) = 0$ and we complete our proof for the separable case.[1] In what follows, we focus on the non-separable case.

Bounding the deterministic expectation. Here, we show that in eq.(A.8), $A(\boldsymbol{w}, S) \leq \sqrt{1/n}$ for all parameters $\boldsymbol{w} \in \mathcal{W}$ and any training set S, provided that we use a sufficient number n' of random structured outputs sampled from the proposal distribution.

By well-known identities, we can rewrite:

$$
\begin{aligned}
A(\boldsymbol{w}, S) &= \frac{1}{n} \sum_{(x,y) \in S} \int_0^1 \Big(\mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} \left[v(x, y, y', h', \boldsymbol{w}) < z \right]^{n'} \\
&\qquad\qquad - \mathop{\mathbb{P}}_{(y',h') \sim Q(\boldsymbol{w},x)} \left[v(x, y, y', h', \boldsymbol{w}) < z \right] \Big) dz \qquad \text{(A.10.a)} \\
&\leq \frac{1}{n} \sum_{(x,y) \in S} \mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} \Big[v(x, y, y', h', \boldsymbol{w}) < 1 \Big]^{n'} \\
&= \frac{1}{n} \sum_{(x,y) \in S} \mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} \Big[d(y, y', h') < 1 \vee \widetilde{m}(x, y, y', h', \boldsymbol{w}) > 1 \Big]^{n'} \\
&\leq \frac{1}{n} \sum_{(x,y) \in S} \Bigg(\Big(1 - \mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} \Big[d(y, y', h') = 1 \Big] \Big) \\
&\qquad\qquad + \mathop{\mathbb{P}}_{(y',h') \sim R(\boldsymbol{w},x)} \Big[\widetilde{m}(x, y, y', h', \boldsymbol{w}) > 1 \Big] \Bigg)^{n'} \\
&\leq \left(\beta + \exp\left(\frac{-1}{128 \gamma^2 \|\boldsymbol{w}\|_2^2} \right) \right)^{n'} \qquad \text{(A.10.b)} \\
&= \sqrt{1/n} \qquad \text{(A.10.c)}
\end{aligned}
$$

where the step in eq.(A.10.a) holds since for two independent random variables $g, h \in [0, 1]$, we have $\mathbb{E}[g] = 1 - \int_0^1 \mathbb{P}[g < z] dz$ and $\mathbb{P}[\max(g, h) < z] = \mathbb{P}[g < z] \mathbb{P}[h < z]$. Therefore, $\mathbb{E}[\max(g, h)] = 1 - \int_0^1 \mathbb{P}[g < z] \mathbb{P}[h < z] dz$. For the step in eq.(A.10.b), we used Assumption 2.4.1 for the first term in the sum. For the second term in the sum, let $\Delta \equiv \boldsymbol{\Phi}(x, y, h^*) - \boldsymbol{\Phi}(x, y', h')$ where $h^* = \arg\max_{h \in \widetilde{\mathcal{H}}_x} \langle \boldsymbol{\Phi}(x, y, h), \boldsymbol{w} \rangle$, then $\widetilde{m}(x, y, y', h', \boldsymbol{w}) =$

[1]↑ The same result can be obtained for any subset of S for which the "separability" condition holds. Therefore, our analysis with the "non-separability" condition can be seen as a worst case scenario.

$\langle \Delta, \boldsymbol{w} \rangle$. From $\|\boldsymbol{\Phi}(x, y, h)\|_2 \leq \gamma$, we have that $\|\Delta\|_2 \leq 2\gamma$. By Assumption 2.4.2, we have that $\|\mathbb{E}[\Delta]\|_2 \leq 1/(2\sqrt{n}) \leq 1/(2\|\boldsymbol{w}\|_2)$. By Cauchy-Schwarz inequality we have $\langle \mathbb{E}[\Delta], \boldsymbol{w} \rangle \leq \|\mathbb{E}[\Delta]\|_2 \|\boldsymbol{w}\|_2 \leq \|\boldsymbol{w}\|_2 / (2\|\boldsymbol{w}\|_2) \leq 1/2$. Since $\langle \mathbb{E}[\Delta], \boldsymbol{w} \rangle \leq 1/2$ and $\|\Delta\|_2 \leq 2\gamma$, we apply Lemma A.3.1 in the step in eq.(A.10.b). For the step in eq.(A.10.c), let $\lambda \equiv \frac{1}{\log(1/(\beta + e^{-1/(128\gamma^2 \|\boldsymbol{w}\|_2^2)}))}$. Furthermore, let $n' = \frac{1}{2}\lambda \log n$.

Therefore, $\left(\beta + \exp\left(\frac{-1}{128\gamma^2 \|\boldsymbol{w}\|_2^2}\right)\right)^{n'} = \sqrt{1/n}$.

Bounding the stochastic quantity. Here, we show that in eq.(A.9), $B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) \leq \mathcal{O}(\log^2 n/\sqrt{n})$ for all parameters $\boldsymbol{w} \in \mathcal{W}$, any training set S and all collections $\mathbb{T}(\boldsymbol{w})$. For clarity of presentation, define:

$$g(x, y, T, \boldsymbol{w}) \equiv \max_{(\widehat{y}, \hat{h}) \in T} v(x, y, \widehat{y}, \hat{h}, \boldsymbol{w})$$

Thus, we can rewrite:

$$B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) = \frac{1}{n} \sum_{(x,y) \in S} \left(\mathop{\mathbb{E}}_{T(\boldsymbol{w},x) \sim R(\boldsymbol{w},x)^{n'}} [g(x, y, T(\boldsymbol{w}, x), \boldsymbol{w})] - g(x, y, T(\boldsymbol{w}, x), \boldsymbol{w}) \right)$$

Let $r_x \equiv |\mathcal{Y}_x \times \mathcal{H}_x|$ and thus $\mathcal{Y}_x \times \mathcal{H}_x \equiv \{(y_1, h_1) \ldots (y_{r_x}, h_{r_x})\}$. Let $\boldsymbol{\pi}(x) = (\boldsymbol{\pi}_1 \ldots \boldsymbol{\pi}_{r_x})$ be a permutation of $\{1 \ldots r_x\}$ such that $\langle \boldsymbol{\Phi}(x, y_{\boldsymbol{\pi}_1}, h_{\boldsymbol{\pi}_1}), \boldsymbol{w} \rangle < \cdots < \langle \boldsymbol{\Phi}(x, y_{\boldsymbol{\pi}_{r_x}}, h_{\boldsymbol{\pi}_{r_x}}), \boldsymbol{w} \rangle$. Let Π be the collection of the n permutations $\boldsymbol{\pi}(x)$ for all $(x, y) \in S$, i.e., $\Pi = \{\boldsymbol{\pi}(x)\}_{(x,y) \in S}$. From Assumption 2.4.3, we have that $R(\boldsymbol{\pi}(x), x) \equiv R(\boldsymbol{w}, x)$. Similarly, we rewrite $T(\boldsymbol{\pi}(x), x) \equiv T(\boldsymbol{w}, x)$ and $\mathbb{T}(\Pi) \equiv \mathbb{T}(\boldsymbol{w})$.

Furthermore, let $\mathcal{W}_{\Pi,S}$ be the set of all $\boldsymbol{w} \in \mathcal{W}$ that induce Π on the training set S. For the parameter space $\mathcal{W}$, collection Π and training set S, define the function class $\mathbb{G}_{\mathcal{W},\Pi,S}$ as follows:

$$\mathbb{G}_{\mathcal{W},\Pi,S} \equiv \{g(x, y, T, \boldsymbol{w}) \mid \boldsymbol{w} \in \mathcal{W}_{\Pi,S} \text{ and } (x, y) \in S\}$$

Note that since $|\mathcal{Y}_x \times \mathcal{H}_x| \leq r$ for all $(x, y) \in S$, then $|\cup_{(x,y) \in S} \mathcal{Y}_x \times \mathcal{H}_x| \leq \sum_{(x,y) \in S} |\mathcal{Y}_x \times \mathcal{H}_x| \leq nr$. Note that each ordering of the nr structured outputs completely determines a collection

Π and thus the collection of proposal distributions $R(\boldsymbol{w}, x)$ for each $(x, y) \in S$. Note that since $|\cup_{(x,y)\in S} \mathcal{P}_x| \leq \ell$, we consider $\boldsymbol{\Phi}(x, y, h) \in \mathbb{R}^\ell$. Although we can consider $\boldsymbol{w} \in \mathbb{R}^\ell$, the vector $\boldsymbol{w}$ is sparse with at most $\mathfrak{s}$ non-zero entries. Thus, we take into account all possible subsets of $\mathfrak{s}$ features from ℓ possible features. From results in [137–139], we can conclude that there are at most $(nr)^{2(\mathfrak{s}-1)}$ linearly inducible orderings, for a fixed set of $\mathfrak{s}$ features. Therefore, there are at most $\binom{\ell}{\mathfrak{s}}(nr)^{2(\mathfrak{s}-1)} \leq \ell^{\mathfrak{s}}(nr)^{2\mathfrak{s}}$ collections Π.

Fix $\delta \in (0, 1)$. By Rademacher-based uniform convergence[2] and by a union bound over all $\ell^{\mathfrak{s}}(nr)^{2\mathfrak{s}}$ collections Π, with probability at least $1 - \delta/2$ over the choice of n sets of random structured outputs, simultaneously for all parameters $\boldsymbol{w} \in \mathcal{W}$:

$$B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w})) \leq 2\ \mathfrak{R}_{\mathbb{T}(\Pi)}(\mathbb{G}_{\mathcal{W},\Pi,S}) + 3\sqrt{\frac{\mathfrak{s}(\log \ell + 2\log(nr)) + \log(4/\delta)}{n}}, \tag{A.11}$$

where $\mathfrak{R}_{\mathbb{T}(\Pi)}(\mathbb{G}_{\mathcal{W},\Pi,S})$ is the *empirical* Rademacher complexity of the function class $\mathbb{G}_{\mathcal{W},\Pi,S}$ with respect to the collection $\mathbb{T}(\Pi)$ of the n sets $T(\boldsymbol{\pi}(x), x)$ for all $(x, y) \in S$. Let σ be an n-dimensional vector of independent Rademacher random variables indexed by $(x, y) \in S$, i.e., $\mathbb{P}[\sigma_{(x,y)} = +1] = \mathbb{P}[\sigma_{(x,y)} = -1] = 1/2$. The empirical Rademacher complexity is defined as:

$$\begin{aligned}
\mathfrak{R}_{\mathbb{T}(\Pi)}(\mathbb{G}_{\mathcal{W},\Pi,S}) &\equiv \mathop{\mathbb{E}}_{\sigma}\left[\sup_{g\in\mathbb{G}_{\mathcal{W},\Pi,S}}\left(\frac{1}{n}\sum_{(x,y)\in S}\sigma_{(x,y)}g(x, y, T(\boldsymbol{\pi}(x), x), \boldsymbol{w})\right)\right] \\
&= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\boldsymbol{w}\in\mathcal{W}_{\Pi,S}}\left(\frac{1}{n}\sum_{(x,y)\in S}\sigma_{(x,y)}\max_{(\widehat{y},\hat{h})\in T(\boldsymbol{\pi}(x),x)} d(y, \widehat{y}, \hat{h})\ \mathbb{1}\big[1 - \widetilde{m}(x, y, \widehat{y}, \hat{h}, \boldsymbol{w}) \geq 0\big]\right)\right] \\
&= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\boldsymbol{w}\in\mathcal{W}_{\Pi,S}}\left(\frac{1}{n}\sum_{(x,y)\in S}\sigma_{(x,y)}\max_{(\widehat{y},\hat{h})\in T(\boldsymbol{\pi}(x),x)} d(y, \widehat{y}, \hat{h})\ \mathbb{1}\left[1 \geq \max_{h\in\widetilde{\mathcal{H}}_x}\langle\boldsymbol{\Phi}(x, y, h), \boldsymbol{w}\rangle - \langle\boldsymbol{\Phi}(x, \widehat{y}, \hat{h}), \boldsymbol{w}\rangle\right]\right)\right]
\end{aligned}$$

[2]↑ Note that for the analysis of $B(\boldsymbol{w}, S, \mathbb{T}(\boldsymbol{w}))$, the training set S is fixed and randomness stems from the collection $\mathbb{T}(\boldsymbol{w})$. Also, note that for applying McDiarmid's inequality, independence of each set $T(\boldsymbol{w}, x)$ for all $(x, y) \in S$ is a sufficient condition, and identically distributed sets $T(\boldsymbol{w}, x)$ are not necessary.

$$= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{\boldsymbol{w}\in\mathbb{R}^{\ell}\setminus\{0\}}\left(\frac{1}{n}\sum_{i\in\{1\ldots n\}}\sigma_i\max_{j\in\{1\ldots n'\}}d_{ij}\;\mathbb{1}\left[1\geq\max_{h\in\{1\ldots|\widetilde{\mathcal{H}}_x|\}}\langle z'_{ih},\boldsymbol{w}\rangle-\langle z_{ij},\boldsymbol{w}\rangle\right]\right)\right] \tag{A.12.a}$$

$$\leq \sum_{j\in\{1\ldots n'\}}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\boldsymbol{w}\in\mathbb{R}^{\ell}\setminus\{0\}}\left(\frac{1}{n}\sum_{i\in\{1\ldots n\}}\sigma_i\;d_{ij}\;\mathbb{1}\left[1\geq\max_{h\in\{1\ldots|\widetilde{\mathcal{H}}_x|\}}\langle z'_{ih},\boldsymbol{w}\rangle-\langle z_{ij},\boldsymbol{w}\rangle\right]\right)\right] \tag{A.12.b}$$

$$\leq \sum_{j\in\{1\ldots n'\}}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\boldsymbol{w}\in\mathbb{R}^{\ell}\setminus\{0\}}\left(\frac{1}{n}\sum_{i\in\{1\ldots n\}}\sigma_i\;\mathbb{1}\left[1\geq\max_{h\in\{1\ldots|\widetilde{\mathcal{H}}_x|\}}\langle z'_{ih},\boldsymbol{w}\rangle-\langle z_{ij},\boldsymbol{w}\rangle\right]\right)\right] \tag{A.12.c}$$

$$\leq \sum_{j\in\{1\ldots n'\}}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{\tilde{\boldsymbol{w}}\in\mathbb{R}^{\ell(|\widetilde{\mathcal{H}}|+1)+1}\setminus\{0\}}\left(\frac{1}{n}\sum_{i\in\{1\ldots n\}}\sigma_i\;\mathbb{1}\left[\langle z_{ij}^{\widetilde{\mathcal{H}}},\tilde{\boldsymbol{w}}\rangle\geq 0\right]\right)\right] \tag{A.12.d}$$

$$\leq 2n'\sqrt{\frac{(2\mathfrak{s}+1)\log\left(\ell(n\tilde{r}+1)+1\right)\log\left(n+1\right)}{n}} \tag{A.12.e}$$

where in the step in eq.(A.12.a), the terms σ_i, d_{ij}, z'_{ih}, z_{ij} correspond to $\sigma_{(x,y)}$, $d(y,\widehat{y},\hat{h})$, $\boldsymbol{\Phi}(x,y,h)$ and $\boldsymbol{\Phi}(x,\widehat{y},\hat{h})$ respectively. Thus, we assume that index i corresponds to the training sample $(x,y)\in S$, and that index j corresponds to the structured output and latent variable $(\widehat{y},\hat{h})\in T(\boldsymbol{\pi}(x),x)$. Note that since $\boldsymbol{\Phi}(x,y,h)\in\mathbb{R}^{\ell}$, thus the step in eq.(A.12.a) considers $\boldsymbol{w}, z'_{ih}, z_{ij}\in\mathbb{R}^{\ell}\setminus\{0\}$ without loss of generality. The step in eq.(A.12.b) follows from the fact that for any two function classes $\mathbb{G}$ and $\mathcal{H}$, we have that $\mathfrak{R}(\{\max(g,h)\mid g\in\mathbb{G}\text{ and }h\in\mathcal{H}\})\leq\mathfrak{R}(\mathbb{G})+\mathfrak{R}(\mathcal{H})$. The step in eq.(A.12.c) follows from the composition lemma and the fact that $d_{ij}\in[0,1]$ for all i and j. The step in eq.(A.12.d) considers a larger function class, we consider $\tilde{\boldsymbol{w}}, z_{ij}^{\widetilde{\mathcal{H}}}\in\mathbb{R}^{\ell(|\widetilde{\mathcal{H}}|+1)+1}\setminus\{0\}$. More detailed, for a fixed i,j, and $\boldsymbol{w}\in\mathbb{R}^{\ell}$, we can construct the vectors $z_{ij}^{\widetilde{\mathcal{H}}}=(1,-z'_{i1},\ldots,-z'_{i|\widetilde{\mathcal{H}}|},z_{ij})$ and $\tilde{\boldsymbol{w}}^{(t)}=(1,\boldsymbol{w}^{(1)},\ldots,\boldsymbol{w}^{(|\widetilde{\mathcal{H}}|)},\boldsymbol{w})$, where $\boldsymbol{w}^{(l)}=\boldsymbol{w}$ if $l=t$, and $\boldsymbol{w}^{(l)}=\mathbf{0}$ otherwise. The step in eq.(A.12.e) follows from the Massart lemma, the Sauer-Shelah lemma and the VC-dimension of sparse linear classifiers. That is, for any function class $\mathbb{G}$, we have that $\mathfrak{R}(\mathbb{G})\leq\sqrt{\frac{2VC(\mathbb{G})\log(n+1)}{n}}$ where $VC(\mathbb{G})$ is the VC-dimension of $\mathbb{G}$. Finally, note that $|\widetilde{\mathcal{H}}_x|\leq\tilde{r},\forall(x,y)\in S$, and $|\widetilde{\mathcal{H}}|=|\cup_{(x,y)\in S}\widetilde{\mathcal{H}}_x|\leq n\tilde{r}$. Also, since $\boldsymbol{w}$ is $\mathfrak{s}$-sparse, we have that $\tilde{\boldsymbol{w}}$ is $(2\mathfrak{s}+1)$-sparse. Then, by Theorem 20 of

[68], $VC(\mathbb{G}) \leq 2(2\mathfrak{s}+1)\log\left(\ell(|\widetilde{\mathcal{H}}|+1)+1\right)$ for the class $\mathbb{G}$ of sparse linear classifiers on $\mathbb{R}^{\ell(|\widetilde{\mathcal{H}}|+1)+1}$, with $3 \leq 2\mathfrak{s}+1 \leq \frac{9}{20}\sqrt{\ell(|\widetilde{\mathcal{H}}|+1)+1}$.

By eq.(A.4), eq.(A.6.c), eq.(A.10.c), eq.(A.11) and eq.(A.12.e), we prove our claim. □

B. APPENDIX TO CHAPTER 3

B.1 Proof of Theorem 3.2.2

Proof. The proof is motivated by the work of [65] for binary classifiers. As a first step it is clear that one can lower bound eq.(3.3) by defining the maximum over a subset of $\mathcal{P}$. That is, we create a collection of family of distributions $\mathbb{D}_\gamma$, where $|\mathbb{D}_\gamma| = |\Phi|$. Each family distribution $\mathcal{D}_{\gamma,u,v} \in \mathbb{D}_\gamma$ is further indexed by $(u,v) \in T = \{(u,v) \mid u \neq v, \{u,v\} \subseteq \text{Scope}(\phi), \phi \in \Phi\}$. Then we have,

$$\mathfrak{M}_m(\mathcal{P}) \geq \max_{(u,v)\in T} \mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}).$$

Our approach consists of first defining the families of distributions $\mathcal{D}_{\gamma,u,v} \subset \mathcal{P}$ such that its elements can be naturally indexed by the vertices of a binary hypercube. We will then relate the expected excess risk problem to an estimation of binary strings in order to apply Assouad's lemma.

Construction of $\mathcal{D}_{\gamma,u,v}$. Consider a fixed $(u,v) \in T$. We first focus on constructing a family of distributions, $\mathcal{D}_{\gamma,u,v}$, parameterized by $\gamma > 0$. Each distribution $D_{\gamma,u,v,B} \in \mathcal{D}_{\gamma,u,v}$ is further indexed by a binary matrix $B \in \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$, where $d^{(0)}_{u,v}$ is the PAIR-dimension of $\mathsf{F}^{(0)}_{u,v}$. To construct these distributions, we will first pick the marginal distribution $D^{(x)}_{\gamma,u,v,B}$ of the feature x, and then specify the conditional distributions $D^{(y|x)}_{\gamma,u,v,B}$ of y given x, for each $B \in \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$.

We construct $D^{(x)}_{\gamma,u,v,B}$ as follows. Since $\mathsf{F}^{(0)}_{u,v}$ is a class with PAIR-dimension $d^{(0)}_{u,v}$, there exists a set of points $\{x_1, \ldots, x_{d^{(0)}_{u,v}}\} \in \mathcal{X}$ that is shattered by $\mathsf{F}^{(0)}_{u,v}$, that is, for any binary matrix $B \in \{0,1\}^{d^{(0)}_{u,v}\times 2}$ there exists at least one function $\mathsf{f}^{(0)}_{u,v} \in \mathsf{F}^{(0)}_{u,v}$ such that $\mathsf{f}^{(0)}_{u,v}(x_i) = B_{i*}$, for all $i \in \{1, \ldots, d^{(0)}_{u,v}\}$.

We now define the marginal distribution $D^{(x)}_{\gamma,u,v,B}$ such that its support is the shattered set

$\{x_1, \ldots, x_{d^{(0)}_{u,v}}\}$, i.e., $\mathbb{P}^{(x)}_{\gamma,u,v,B}\left[\{x_1, \ldots, x_{d^{(0)}_{u,v}}\}\right] = 1$. For a given parameter $p \in [0, 1/(d^{(0)}_{u,v}-1)]$, whose value is set later, we have:

$$\mathbb{P}^{(x)}_{\gamma,u,v,B}[x_i] = \begin{cases} p, & \text{if } i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ 1 - (d^{(0)}_{u,v} - 1)p, & \text{otherwise.} \end{cases}$$

Next, for a fixed $B \in \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$, the conditional distribution of y given x, $D^{(y|x)}_{\gamma,u,v,B}$, is defined as:

$$\mathbb{P}^{(y|x)}_{\gamma,u,v,B}\left[y|x\right] = \begin{cases} \frac{1-3\gamma}{4}, & \text{if } x = x_i, y_u = 1 - B_{i1}, y_v = 1 - B_{i2}, \\ & y_k = 0 \text{ for } k \in V \setminus \{u, v\}, i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ \frac{1+\gamma}{4}, & \text{if } x = x_i, (y_u \neq 1 - B_{i1} \text{ or } y_v \neq 1 - B_{i2}), \\ & y_k = 0 \text{ for } k \in V \setminus \{u, v\}, i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ 0, & \text{otherwise,} \end{cases}$$

here we implicitly assume that $\gamma \in (0, 1/3]$ in order to obtain a valid distribution. The above definition produces the following marginal probabilities:

$$\eta_j^{(\gamma,u,v,B)}(x) \equiv \mathbb{P}^{(y_j|x)}_{\gamma,u,v,B}\left[y_j = 1|x\right] = \begin{cases} \frac{1-\gamma}{2}, & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ & ((j = u, B_{i1} = 0) \text{ or } (j = v, B_{i2} = 0)), \\ \frac{1+\gamma}{2}, & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ & ((j = u, B_{i1} = 1) \text{ or } (j = v, B_{i2} = 1)), \\ 0, & \text{otherwise,} \end{cases} \quad \text{(B.1)}$$

where we note that for each $j \in V$ and any x we have that $|2\eta_j^{(\gamma,u,v,B)}(x) - 1| \geq \gamma$. Given the above marginals, the corresponding Bayes-Hamming predictor for substructure y_j for a given input x (see Proposition 3.1.1), which we denote by $(\mathsf{f}^*_{B,u,v}(x))_j$, is given by:

$$(\mathsf{f}^*_{B,u,v}(x))_j = \begin{cases} 0, & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ & ((j = u \text{ and } B_{i1} = 0) \text{ or } (j = v \text{ and } B_{i2} = 0)) \\ 1, & \text{if } x = x_i \text{ for some } i \in \{1, \ldots, d^{(0)}_{u,v} - 1\}, \\ & ((j = u \text{ and } B_{i1} = 1) \text{ or } (j = v \text{ and } B_{i2} = 1)) \\ 0, & \text{otherwise.} \end{cases} \tag{B.2}$$

That is, we have that the output of the Bayes-Hamming predictor on each x_i for $i \in \{1 \ldots d^{(0)}_{u,v} - 1\}$, for each substructure y_j for $j \in \{u, v\}$, is equal to the bit value B_{i1} or B_{i2}, and zero otherwise.

Reduction to estimation of binary strings. For any distribution $D_{\gamma,u,v,B} \in \mathcal{D}_{\gamma,u,v}$, we can further express the expected excess risk in eq.(3.3) as follows:

$$\begin{aligned} R_{B,u,v}(\mathcal{A}(S)) - R_{B,u,v}(\mathsf{f}^*_{B,u,v}) &= \mathop{\mathbb{E}}_{(x,y)\sim D_{\gamma,u,v,B}} \left[\sum_{j=1}^{l} (1 - 2y_j)((\hat{\mathsf{f}}_m(x))_j - (\mathsf{f}^*_{B,u,v}(x))_j) \right] \\ &= \sum_{j=1}^{l} \mathop{\mathbb{E}}_{x\sim D^{(x)}_{\gamma,u,v,B}} \left[\mathop{\mathbb{E}}_{y_j\sim D^{(y_j|x)}_{\gamma,u,v,B}} \left[(1 - 2y_j)((\hat{\mathsf{f}}_m(x))_j - (\mathsf{f}^*_{B,u,v}(x))_j) \right] \right] \\ &= \sum_{j=1}^{l} \mathop{\mathbb{E}}_{x\sim D^{(x)}_{\gamma,u,v,B}} \left[\left| 2\eta_j^{(\gamma,u,v,B)}(x) - 1 \right| \cdot \left| (\hat{\mathsf{f}}_m(x))_j - (\mathsf{f}^*_{B,u,v}(x))_j \right| \right] \\ &\geq \gamma \cdot \mathop{\mathbb{E}}_{x\sim D^{(x)}_{\gamma,u,v,B}} \left[\sum_{j=1}^{l} \left| (\hat{\mathsf{f}}_m(x))_j - (\mathsf{f}^*_{B,u,v}(x))_j \right| \right] \qquad \text{(B.3)} \\ &= \gamma \cdot \sum_{i=1}^{d^{(0)}_{u,v}} \sum_{j=1}^{l} \left| (\hat{\mathsf{f}}_m(x_i))_j - (\mathsf{f}^*_{B,u,v}(x_i))_j \right| \cdot \mathbb{P}^{(x)}_{\gamma,u,v,B}[x_i], \\ &\stackrel{\text{def}}{=} \gamma \cdot \|\hat{\mathsf{f}}_m - \mathsf{f}^*_{B,u,v}\|_{1,1}, \qquad \text{(B.4)} \end{aligned}$$

where $R_{B,u,v}$ denotes the expected risk and $\mathsf{f}^*_{B,u,v}$ the Bayes-Hamming predictor, both with respect to $D_{\gamma,u,v,B}$. Here $\hat{\mathsf{f}}_m$ is the output of $\mathcal{A}(S)$, with $(\hat{\mathsf{f}}_m(x))_j$ denoting the j-th substructure of the output $\hat{\mathsf{f}}_m(x)$, and $\eta_j^{(\gamma,u,v,B)}(x)$ denotes the marginal probability $\mathbb{P}_{D^{(y_j|x)}_{\gamma,u,v,B}}[y_j = 1|x]$.

Equation (B.3) follows from our definition of $D^{(y_j|x)}_{\gamma,u,v,B}$ (see eq.(B.1)), and the $L_{1,1}$ matrix norm in eq.(B.4) is computed with respect to $D^{(x)}_{\gamma,u,v,B}$. Thus, we have that:

$$\begin{aligned}\mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) &= \inf_{\hat{\mathsf{f}}_m} \max_{B\in\{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}} \mathbb{E}_{B,u,v}\left[R_{B,u,v}(\hat{\mathsf{f}}_m) - R_{B,u,v}(\mathsf{f}^*_{B,u,v})\right] \\ &\geq \gamma \cdot \inf_{\hat{\mathsf{f}}_m} \max_{B\in\{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}} \mathbb{E}_{B,u,v}\left[\|\hat{\mathsf{f}}_m - \mathsf{f}^*_{B,u,v}\|_{1,1}\right], \end{aligned} \tag{B.5}$$

where $\mathbb{E}_{B,u,v}[\,\cdot\,]$ denotes the expectation with respect to $S \sim D^m_{\gamma,u,v,B}$. Equation (B.5) follows from eq.(B.4). Given any candidate estimation $\hat{\mathsf{f}}_m$, let $\widehat{B}_m \in \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$ be defined as follows:

$$\widehat{B}_m \stackrel{\text{def}}{=} \operatorname*{arg\,min}_{B\in\{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}} \|\hat{\mathsf{f}}_m - \mathsf{f}^*_{B,u,v}\|_{1,1}. \tag{B.6}$$

Intuitively, $\widehat{B}_m \in \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$ is the binary matrix that indexes the element of $\{\mathsf{f}^*_{B,u,v} : B \in \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}\}$ that is closest to $\hat{\mathsf{f}}_m$ in $L_{1,1}$ norm. Then, for any B, we have

$$\begin{aligned}\|\mathsf{f}^*_{\widehat{B}_m,u,v} - \mathsf{f}^*_{B,u,v}\|_{1,1} &\leq \|\mathsf{f}^*_{\widehat{B}_m,u,v} - \hat{\mathsf{f}}_m\|_{1,1} + \|\hat{\mathsf{f}}_m - \mathsf{f}^*_{B,u,v}\|_{1,1} \\ &\leq 2\|\hat{\mathsf{f}}_m - \mathsf{f}^*_{B,u,v}\|_{1,1},\end{aligned}$$

where we first applied the triangle inequality, and then used eq.(B.6). Applying this to eq.(B.5), we obtain:

$$\mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) \geq \frac{\gamma}{2} \min_{\widehat{B}_m} \max_{B\in\{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}} \mathbb{E}_{B,u,v}\left[\|\mathsf{f}^*_{\widehat{B}_m,u,v} - \mathsf{f}^*_{B,u,v}\|_{1,1}\right], \tag{B.7}$$

here the infimum is over all estimators that take values in $\{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$ based on m samples, i.e., over $\widehat{B}_m : (\mathcal{X}\times\mathcal{Y})^m \to \{0,1\}^{(d^{(0)}_{u,v}-1)\times 2}$. We now compute $\|\mathsf{f}^*_{B,u,v} - \mathsf{f}^*_{B',u,v}\|_{1,1}$ for any two B, B'. Using eq.(B.2) we have:

$$\|\mathsf{f}^*_{B,u,v} - \mathsf{f}^*_{B',u,v}\|_{1,1} = \sum_{i=1}^{d^{(0)}_{u,v}} \sum_{j=1}^{l} \left|(\mathsf{f}^*_{B,u,v}(x_i))_j - (\mathsf{f}^*_{B',u,v}(x_i))_j\right| \cdot \mathbb{P}^{(x)}_{\gamma,u,v,B}[x_i]$$

$$= p \cdot \sum_{i=1}^{d_{u,v}^{(0)}-1} \sum_{j=1}^{2} \left| B_{ij} - B'_{ij} \right|$$
$$= p \cdot L_H(B, B').$$

In the last equality we abuse notation and consider the matrix $B \in \{0,1\}^{(d_{u,v}^{(0)}-1)\times 2}$ as a vector of dimension $2(d_{u,v}^{(0)}-1)$. Replacing this result into eq.(B.7), we get:

$$\mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) \geq \frac{p\,\gamma}{2} \min_{\widehat{B}_m} \max_{B\in\{0,1\}^{(d_{u,v}^{(0)}-1)\times 2}} \mathbb{E}_{B,u,v}[L_H(\widehat{B}_m, B)],$$

which is related to an estimation problem in the $\{0,1\}^{2(d_{u,v}^{(0)}-1)}$ hypercube.

Applying Assouad's lemma. In order to apply Assouad's lemma, we need an upper bound on the squared Hellinger distance $H^2(D_{\gamma,u,v,B}, D_{\gamma,u,v,B'})$ for all B, B' with $L_H(B, B') = 1$. For any two $B, B' \in \{0,1\}^{(d_{u,v}^{(0)}-1)\times 2}$ we have:

$$H^2(D_{\gamma,u,v,B}, D_{\gamma,u,v,B'}) = \sum_{i=1}^{d_{u,v}^{(0)}} \sum_{y\in\{0,1\}^l} \left(\sqrt{\mathbb{P}_{\gamma,u,v,B}(x_i, y)} - \sqrt{\mathbb{P}_{\gamma,u,v,B'}(x_i, y)} \right)^2$$
$$= p \sum_{i=1}^{d_{u,v}^{(0)}-1} \sum_{y\in\{0,1\}^l} \left(\sqrt{\mathbb{P}^{(y|x)}_{\gamma,u,v,B'}(y|x_i)} - \sqrt{\mathbb{P}^{(y|x)}_{\gamma,u,v,B}(y|x_i)} \right)^2.$$

In the above summation, the inner sum is zero if $B_{i*} = B'_{i*}$. Since we are interested on B and B' such that $L_H(B, B') = 1$, this implies that for only one row i from $\{1, \ldots, d_{u,v}^{(0)} - 1\}$ we have $B_{i*} \neq B'_{i*}$ with exactly one bit different. Then, the Hellinger distance results in: $H^2(D_{\gamma,u,v,B}, D_{\gamma,u,v,B'}) = p \cdot (1 - \gamma - \sqrt{1 - 2\gamma - 3\gamma^2}) \leq 6p\gamma^2$. Applying Assouad's lemma we obtain:

$$\mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) \geq \frac{p\gamma}{2} \min_{\widehat{B}_m} \max_{B\in\{0,1\}^{(d_{u,v}^{(0)}-1)\times 2}} \mathbb{E}_{B,u,v} \left[L_H(\widehat{B}_m, B) \right]$$
$$\geq \frac{p\gamma(d_{u,v}^{(0)} - 1)}{2} \left(1 - \sqrt{6p\gamma^2 m} \right) \tag{B.8}$$

Let $p = {}^{2}/_{(27\gamma^2 m)}$, and noting that if $\gamma \geq \sqrt{(d^{(0)}_{u,v} - 1)/m}$ then the condition $p \leq {}^{1}/_{(d^{(0)}_{u,v}-1)}$ holds. Replacing p in eq.(B.8) we have:

$$\mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) \geq \frac{d^{(0)}_{u,v} - 1}{81\gamma m}. \tag{B.9}$$

If $\gamma \leq \sqrt{\frac{d^{(0)}_{u,v}-1}{m}}$, and using the same construction as above with $\widetilde{\gamma} = \sqrt{\frac{d^{(0)}_{u,v}-1}{m}}$, we see that:

$$\mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) \geq \frac{d^{(0)}_{u,v} - 1}{81\widetilde{\gamma} m} = \frac{1}{81}\sqrt{\frac{d^{(0)}_{u,v} - 1}{m}}. \tag{B.10}$$

Therefore, combining equations (B.9) and (B.10), and since the choice of (u, v) was arbitrary, we have that:

$$\begin{aligned}
\mathfrak{M}_m(\mathcal{P}) &\geq \max_{(u,v)\in T} \mathfrak{M}_m(\mathcal{D}_{\gamma,u,v}) \\
&\geq \max_{(u,v)\in T} \frac{1}{81} \min\left(\frac{d^{(0)}_{u,v} - 1}{\gamma m}, \sqrt{\frac{d^{(0)}_{u,v} - 1}{m}}\right) \\
&= \frac{1}{81} \min\left(\frac{d-1}{\gamma m}, \sqrt{\frac{d-1}{m}}\right),
\end{aligned}$$

which concludes our proof. □

C. APPENDIX TO CHAPTER 4

C.1 Proof of Theorem 4.2.2

Proof. Since $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{M}$ with eigenvalue 0, and $\boldsymbol{M}$ is a symmetric matrix, we can express λ_2 using the variational characterization of eigenvalues as follows:

$$\lambda_2 = \min_{\boldsymbol{a}\in\mathbb{R}^n,\ \boldsymbol{a}^\top \boldsymbol{y}=0} R_{\boldsymbol{M}}(\boldsymbol{a}), \tag{C.1}$$

where we used the fact that $\boldsymbol{y}$ is orthogonal to all the other eigenvectors, by the Spectral Theorem.

Assume that $\boldsymbol{a}$ is the eigenvector associated with λ_2, i.e., we have that $\boldsymbol{M}\boldsymbol{a} = \lambda_2\boldsymbol{a}$ and $\boldsymbol{a}^\top\boldsymbol{y} = 0$. Then, by Lemma 4.2.1, we have that:

$$R_{\boldsymbol{L}}(\boldsymbol{a}\circ\boldsymbol{y} + \delta\mathbf{1}) \leq R_{\boldsymbol{M}}(\boldsymbol{a}) = \lambda_2. \tag{C.2}$$

Next, we choose $\delta \in \mathbb{R}$ such that $\{a_1y_1 + \delta, a_2y_2 + \delta, \ldots, a_ny_n + \delta\}$ has median 0. The reason for the zero median is to later ensure that the subset of vertices $\mathcal{S}$ has less than $n/2$ vertices. Let $\boldsymbol{w} = \boldsymbol{a}\circ\boldsymbol{y} + \delta\mathbf{1}$. From equation (C.2), we have that $R_{\boldsymbol{L}}(\boldsymbol{w}) \leq \lambda_2$.

Let $\boldsymbol{w}^+ = (w_i^+)\top$ such that $w_i^+ = w_i$ if $w_i \geq 0$ and $w_i^+ = 0$ otherwise. Let $\boldsymbol{w}^- = (w_i^-)\top$ such that $w_i^- = w_i$ if $w_i \leq 0$ and $w_i^- = 0$ otherwise. Then, we have that either $R_{\boldsymbol{L}}(\boldsymbol{w}^+) \leq 2R_{\boldsymbol{L}}(\boldsymbol{w})$ or $R_{\boldsymbol{L}}(\boldsymbol{w}^-) \leq 2R_{\boldsymbol{L}}(\boldsymbol{w})$. Now suppose that w.l.o.g. $R_{\boldsymbol{L}}(\boldsymbol{w}^+) \leq 2R_{\boldsymbol{L}}(\boldsymbol{w})$, then, it follows that $R_{\boldsymbol{L}}(\boldsymbol{w}^+) \leq 2\lambda_2$.

Let us scale $\boldsymbol{w}^+$ by some constant $\beta \in \mathbb{R}$ so that: $\{\beta w_1, \beta w_2, \ldots, \beta w_m\} \subseteq [0,1]$. It is clear that $R_{\boldsymbol{L}}(\boldsymbol{w}^+) = R_{\boldsymbol{L}}(\beta\boldsymbol{w}^+)$, therefore, we will still use $\boldsymbol{w}^+$ to denote the rescaled vector. That is, now the entries of vector $\boldsymbol{w}^+$ are in between 0 and 1.

Next, we will show that there exists a set $\mathcal{S} \subset \mathcal{V}$ with $|\mathcal{S}| \leq n/2$ such that: $\frac{\mathbb{E}[|\mathcal{E}(\mathcal{S},\mathcal{S}^C)|]}{\mathbb{E}[|\mathcal{S}|]} \leq \sqrt{2R_{\boldsymbol{L}}(\boldsymbol{w}^+)\Delta_{\max}}$. We construct the set $\mathcal{S}$ as follows. We choose $t \in [0,1]$ uniformly at random and let $\mathcal{S} = \{i \mid (w_i^+)^2 \geq t\}$. Let $B_{i,j} = 1$ if $i \in \mathcal{S}$ and $j \in \mathcal{S}^C$ or if $j \in \mathcal{S}$ and $i \in \mathcal{S}^C$, and $B_{i,j} = 0$ otherwise. Then, $\mathbb{E}[|\mathcal{E}(\mathcal{S},\mathcal{S}^C)|] = \mathbb{E}[\sum_{(i,j)\in\mathcal{E}} B_{i,j}] = \sum_{(i,j)\in\mathcal{E}} \mathbb{E}[B_{i,j}] =$

$\sum_{(i,j)\in\mathcal{E}} P((w_j^+)^2 \le t \le (w_i^+)^2)$. Recall that $(w_i^+)^2 \in [0,1]$, therefore, the probability above is $|(w_i^+)^2 - (w_j^+)^2|$. Thus,

$$\mathbb{E}\left[|\mathcal{E}(\mathcal{S},\mathcal{S}^C)|\right] = \sum_{(i,j)\in\mathcal{E}} |w_i^+ - w_j^+|\,|w_i^+ + w_j^+| \le \sqrt{\sum_{(i,j)\in\mathcal{E}} (w_i^+ - w_j^+)^2}\sqrt{\sum_{(i,j)\in\mathcal{E}} (w_i^+ + w_j^+)^2} \tag{C.3}$$

$$\le \sqrt{\sum_{(i,j)\in\mathcal{E}} (w_i^+ - w_j^+)^2}\sqrt{2\sum_{(i,j)\in\mathcal{E}} (w_i^+)^2 + (w_j^+)^2} \le \sqrt{\sum_{(i,j)\in\mathcal{E}} (w_i^+ - w_j^+)^2}\sqrt{2\Delta_{\max}\sum_i (w_i^+)^2}, \tag{C.4}$$

where eq.(C.3) is due to Cauchy-Schwarz inequality and eq.(C.4) uses the maximum-degree of a node for an upper bound.

Now consider another random variable b_i such that $b_i = 1$ if $i \in \mathcal{S}$, and $b_i = 0$ otherwise. Therefore, we have that $\mathbb{E}\left[|\mathcal{S}|\right] = \mathbb{E}\left[\sum_i b_i\right] = \sum_i \mathbb{E}\left[b_i\right] = \sum_i P(t \le (w_i^+)^2) = \sum_i (w_i^+)^2$. Thus, $\frac{\mathbb{E}\left[|\mathcal{E}(\mathcal{S},\mathcal{S}^C)|\right]}{\mathbb{E}\left[|\mathcal{S}|\right]} \le \frac{\sqrt{\sum_{(i,j)\in\mathcal{E}}(w_i^+ - w_j^+)^2}\sqrt{2\Delta_{\max}\sum_i (w_i^+)^2}}{\sum_i (w_i^+)^2} = \frac{\sqrt{\sum_{(i,j)\in\mathcal{E}}(w_i^+ - w_j^+)^2}\sqrt{2\Delta_{\max}}}{\sqrt{\sum_i (w_i^+)^2}} = \sqrt{2R_{\boldsymbol{L}}(\boldsymbol{w}^+)\Delta_{\max}} \le 2\sqrt{\lambda_2\Delta_{\max}}$. The above implies that there exists some $\mathcal{S}$ such that $\frac{|\mathcal{E}(\mathcal{S},\mathcal{S}^C)|}{|\mathcal{S}|} \le 2\sqrt{\lambda_2\Delta_{\max}}$. Therefore, $\phi_{\mathcal{G}} \le 2\sqrt{\lambda_2\Delta_{\max}}$ or equivalently $\frac{\phi_{\mathcal{G}}^2}{4\Delta_{\max}} \le \lambda_2$. □

C.2 Proof of Theorem 4.2.5

Proof. Without loss of generality assume that $\boldsymbol{y} = \overline{\boldsymbol{y}}$. The first step of our proof corresponds to finding sufficient conditions for when $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\top$ is the unique optimal solution to SDP (4.4), for which we make use of the Karush-Kuhn-Tucker (KKT) optimality conditions [99]. In the following we write the dual formulation of SDP (4.4):

$$\min_{\boldsymbol{V}} \quad \mathrm{Tr}(\boldsymbol{V}) \quad \text{subject to} \quad \boldsymbol{V} \succeq X^{(2)}, \boldsymbol{V} \text{ is diagonal.} \tag{C.5}$$

Thus, we have that $\boldsymbol{Y} = \boldsymbol{y}\boldsymbol{y}^\top$ is guaranteed to be an optimal solution under the following conditions:

1. $\boldsymbol{y}\boldsymbol{y}^\top$ is a feasible solution to the primal problem (4.4).

2. There exists a matrix $\boldsymbol{V}$ feasible for the dual formulation such that $\mathrm{Tr}(X^{(2)}\boldsymbol{y}\boldsymbol{y}^\top) = \mathrm{Tr}(\boldsymbol{V})$.

The first point is trivially verified. For the second point, we assume strong duality in order to find a dual certificate. To achieve that, we make $\boldsymbol{V}_{i,i} = (X^{(2)}\boldsymbol{Y})_{i,i}$. If $\boldsymbol{V} - X^{(2)} \succeq 0$ then the matrix $\boldsymbol{V}$ is a feasible solution to the dual formulation. Thus, our first condition is to have $\boldsymbol{V} - X^{(2)} \succeq 0$, and we conclude that $\boldsymbol{y}\boldsymbol{y}^\top$ is an optimal solution to SDP (4.4).

For showing that $\boldsymbol{y}\boldsymbol{y}^\top$ is the unique optimal solution, it suffices to have $\lambda_2(\boldsymbol{V} - X^{(2)}) > 0$. Suppose that $\widehat{\boldsymbol{Y}}$ is another optimal solution to SDP (4.4). Then, from complementary slackness we have that $\langle \boldsymbol{V} - X^{(2)}, \widehat{\boldsymbol{Y}} \rangle = 0$, and from primal feasibility $\widehat{\boldsymbol{Y}} \succeq 0$. Moreover, notice that we have $(\boldsymbol{V} - X^{(2)})\boldsymbol{y} = 0$, i.e., $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{V} - X^{(2)}$ with eigenvalue 0. By assumption, the second smallest eigenvalue of $\boldsymbol{V} - X^{(2)}$ is greater than 0, therefore, $\boldsymbol{y}$ spans all of its null space. This fact combined with complementary slackness, primal and dual feasibility, entail that $\widehat{\boldsymbol{Y}}$ is a multiple of $\boldsymbol{y}\boldsymbol{y}^\top$. Thus, we must have that $\widehat{\boldsymbol{Y}} = \boldsymbol{y}\boldsymbol{y}^\top$ because $\widehat{Y}_{i,i} = 1$.

From the points above we arrived to the two following sufficient conditions:

$$\boldsymbol{V} - X^{(2)} \succeq 0 \quad \text{and} \quad \lambda_2(\boldsymbol{V} - X^{(2)}) > 0. \tag{C.6}$$

Our next step is to show when condition (C.6) is fulfilled with high probability. Since we have that $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{V} - X^{(2)}$ with eigenvalue zero, showing that $\lambda_2(\boldsymbol{V} - X^{(2)}) > 0$ will imply that $\boldsymbol{V} - X^{(2)}$ is positive semidefinite. Therefore, we focus on controlling its second smallest eigenvalue. Next, we have that:

$$\begin{aligned} \lambda_2(\boldsymbol{V} - X^{(2)}) > 0 \quad &\iff \quad \lambda_2(\boldsymbol{V} - X^{(2)} - \mathbb{E}\left[\boldsymbol{V} - X^{(2)}\right] + \mathbb{E}\left[\boldsymbol{V} - X^{(2)}\right]) > 0 \\ &\impliedby \quad \lambda_1(\boldsymbol{V} - \mathbb{E}\left[\boldsymbol{V}\right]) + \lambda_1(\mathbb{E}\left[X^{(2)}\right] - X^{(2)}) + \lambda_2(\mathbb{E}\left[\boldsymbol{V} - X^{(2)}\right]) > 0. \end{aligned} \tag{C.7}$$

We now focus on condition (C.7) since it implies that $\lambda_2(\boldsymbol{V} - X^{(2)}) > 0$. For the first two summands of condition (C.7) we make use of Lemma 4.2.4, while for the third summand we

make use of Theorem 4.2.2. From $\boldsymbol{V}_{i,i} = (X^{(2)}\boldsymbol{Y})_{i,i}$, we have that $\boldsymbol{V}_{i,i} = y_i X^{(2)}_{i,:}\boldsymbol{y}$, thus, $\boldsymbol{V}_{i,i} = \sum_{j=1}^{n} y_i y_j X_{i,j} = \sum_{j=1}^{n} \mathrm{z}_p^{(i,j)} \mathbb{1}[(i,j) \in \mathcal{E}]$. Then, its expected value is: $\mathbb{E}[\boldsymbol{V}_{i,i}] = \Delta_i(1-2p)$.

Bounding the third summand of condition (C.7). Our goal is to find a non-zero lower bound for the second smallest eigenvalue of $\mathbb{E}[\boldsymbol{V} - X^{(2)}]$. Notice that $\mathbb{E}[\boldsymbol{V} - X^{(2)}] \succeq 0$ since it is a diagonally dominant matrix, and $\boldsymbol{y}$ is its first eigenvector with eigenvalue 0, i.e., $\lambda_1(\mathbb{E}[\boldsymbol{V} - X^{(2)}]) = 0$.

Then, we write $\boldsymbol{M} = \mathbb{E}[\boldsymbol{V} - \boldsymbol{X}]$. Now we focus on finding a lower bound for $\lambda_2(\boldsymbol{M})$. We use the fact that for any vector $\boldsymbol{a} \in \mathbb{R}^n$, we have that $\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a} = (1-2p)\sum_{(i,j)\in\mathcal{E}}(y_i a_i - y_j a_j)^2$.

We also note that $\boldsymbol{M}$ has a 0 eigenvalue with eigenvector $\boldsymbol{y}$. Thus, the matrix $\boldsymbol{M}/(1-2p)$ satisfies the conditions of Theorem 4.2.2 and we have that $\lambda_2(\boldsymbol{M}/(1-2p)) \geq \frac{\phi_{\mathcal{G}}^2}{4\Delta_{\max}}$. We conclude that,

$$\lambda_2(\mathbb{E}[\boldsymbol{V} - X^{(2)}]) \geq (1-2p)\frac{\phi_{\mathcal{G}}^2}{4\Delta_{\max}}. \tag{C.8}$$

Bounding the first summand of condition (C.7). Let $\boldsymbol{N}_p^{(i,j)} = \mathrm{z}_p^{(i,j)}(\mathbf{e}_i\mathbf{e}_i^\top + \mathbf{e}_j\mathbf{e}_j^\top)$, where $\mathbf{e}_i$ is the standard basis, i.e., the vector of all zeros except the i-th entry which is 1. We can now write $\boldsymbol{V} = \sum_{(i,j)\in\mathcal{E}} \boldsymbol{N}_p^{(i,j)}$. Then, we have a sequence of independent random matrices $\{\mathbb{E}[\boldsymbol{N}_p^{(i,j)}] - \boldsymbol{N}_p^{(i,j)}\}$, where we obtain the following: $\lambda_{\max}(\mathbb{E}[\boldsymbol{N}_p^{(i,j)}] - \boldsymbol{N}_p^{(i,j)}) \leq 2(1-p)$, and also $\|\sum_{(i,j)\in\mathcal{E}} \mathbb{E}[(\mathbb{E}[\boldsymbol{N}_p^{(i,j)}] - \boldsymbol{N}_p^{(i,j)})^2]\| \leq 4\Delta_{\max}p(1-p)$.

Next, we use the fact that $\lambda_{\max}(\boldsymbol{A}) = -\lambda_1(-\boldsymbol{A})$ for any matrix $\boldsymbol{A}$. Then, by applying Lemma 4.2.4, we obtain:

$$P\Big(\lambda_1\big(\boldsymbol{V} - \mathbb{E}[\boldsymbol{V}]\big) \leq \frac{-(1-2p)\phi_{\mathcal{G}}^2}{8\Delta_{\max}}\Big) \leq n \cdot \mathrm{e}^{\frac{-3(1-2p)^2\phi_{\mathcal{G}}^4}{1536\Delta_{\max}^3 p(1-p) + 32(1-2p)(1-p)\phi_{\mathcal{G}}^2\Delta_{\max}}} \tag{C.9}$$

Bounding the second summand of condition (C.7). Using similar arguments to the concentration above, we now analyze $\lambda_1(\mathbb{E}[\boldsymbol{X}] - \boldsymbol{X})$. Let $\boldsymbol{H}^{(i,j)} = X_{i,j}(\mathbf{e}_i\mathbf{e}_j^\top + \mathbf{e}_j\mathbf{e}_i^\top)$. Then, we have a sequence of independent random matrices $\{\boldsymbol{H}^{(i,j)} - \mathbb{E}[\boldsymbol{H}^{(i,j)}]\}$ and we can write $\boldsymbol{X} = \sum_{(i,j)\in\mathcal{E}} \boldsymbol{H}^{(i,j)}$. Finally, we have that $\lambda_{\max}(\boldsymbol{H}^{(i,j)} - \mathbb{E}[\boldsymbol{H}^{(i,j)}]) \leq 2(1-p)$, and

$\mathbb{E}\left[(\boldsymbol{H}^{(i,j)} - \mathbb{E}\left[\boldsymbol{H}^{(i,j)}\right])^2\right] = 4p(1-p)(\mathbf{e}_i\mathbf{e}_i^\top + \mathbf{e}_j\mathbf{e}_j^\top)$. Thus, $\|\sum_{(i,j)\in\mathcal{E}} \mathbb{E}\left[(\boldsymbol{H}^{(i,j)} - \mathbb{E}\left[\boldsymbol{H}^{(i,j)}\right])^2\right]\| \leq 4\Delta_{\max}p(1-p)$ and by applying Lemma 4.2.4 we obtain:

$$P\Big(\lambda_1\big(\mathbb{E}\left[\boldsymbol{X}\right] - \boldsymbol{X}\big) \leq \frac{-(1-2p)\phi_{\mathcal{G}}^2}{8\Delta_{\max}}\Big) \leq n \cdot \mathrm{e}^{\frac{-3(1-2p)^2\phi_{\mathcal{G}}^4}{1536\Delta_{\max}^3 p(1-p) + 32(1-2p)(1-p)\phi_{\mathcal{G}}^2\Delta_{\max}}} \tag{C.10}$$

Note that the thresholds in the concentrations above are motivated by equation (C.8). Finally, combining equations (C.8), (C.9), and (C.10), we have that:

$$P\Big(\lambda_2(\boldsymbol{V} - X^{(2)}) > 0\Big) \geq 1 - 2n\mathrm{e}^{\frac{-3(1-2p)^2\phi_{\mathcal{G}}^4}{1536\Delta_{\max}^3 p(1-p) + 32(1-2p)(1-p)\phi_{\mathcal{G}}^2\Delta_{\max}}},$$

which concludes our proof. □

C.3 Proof of Theorem 4.3.3

For simplicity, let $\boldsymbol{W}$ and $\boldsymbol{L}$ be the weight matrix and Laplacian matrix of an undirected connected graph H of m nodes. Also, let $\boldsymbol{W}^+$ and $\boldsymbol{W}^-$ be the weight matrices of H^+ and H^-. For a matrix $\boldsymbol{M}$ and vector $\boldsymbol{v}$, we use $R_{\boldsymbol{M}}(\boldsymbol{v})$ to denote their Rayleigh quotient, i.e., $R_{\boldsymbol{M}}(\boldsymbol{v}) = \frac{\boldsymbol{v}^\top\boldsymbol{M}\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}}$. It follows that $R_{\boldsymbol{L}}(\boldsymbol{v}) := \frac{\boldsymbol{v}^\top\boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^\top\boldsymbol{v}} = \frac{\sum_{i<j} W_{i,j}(v_i - v_j)^2}{\boldsymbol{v}^\top\boldsymbol{v}}$, and $\lambda_2(\boldsymbol{L}) = \min_{\boldsymbol{v}\perp\mathbf{1}} R_{\boldsymbol{L}}(\boldsymbol{v})$. Similarly, we define $R^+_{\boldsymbol{L}}(\boldsymbol{v}) := \frac{\sum_{i<j} W^+_{i,j}(v_i - v_j)^2}{\boldsymbol{v}^\top\boldsymbol{v}}$, $R^-_{\boldsymbol{L}}(\boldsymbol{v}) := \frac{\sum_{i<j} W^-_{i,j}(v_i - v_j)^2}{\boldsymbol{v}^\top\boldsymbol{v}}$. Note that $R_{\boldsymbol{L}}(\boldsymbol{v}) = R^+_{\boldsymbol{L}}(\boldsymbol{v}) + R^-_{\boldsymbol{L}}(\boldsymbol{v})$. Next, we state a lemma that will be of use for the proof of Theorem 4.3.3.

Lemma C.3.1. *Let $\boldsymbol{L}$ be a Laplacian matrix of dimension $m \times m$. Let also $\mathbf{1}$ denote a vector of ones. Then, for any $\delta \in \mathbb{R}, \boldsymbol{v} \in \mathbb{R}^m, \sum_i v_i \geq 0$, it follows that*

$$R^+_{\boldsymbol{L}}(\boldsymbol{v}) \geq R^+_{\boldsymbol{L}}(\boldsymbol{v} + \delta\mathbf{1})\,.$$

Proof. Starting from the right-hand side, we have

$$R^+_{\boldsymbol{L}}(\boldsymbol{v} + \delta\mathbf{1}) = \frac{\sum_{i<j} W^+_{i,j}\left((v_i + \delta) - (v_j + \delta)\right)^2}{\sum_i (v_i + \delta)^2}$$

$$
\begin{aligned}
&= \frac{\sum_{i<j} W^+_{i,j} \left(v_i - v_j\right)^2}{\sum_i \left(v_i + \delta\right)^2} \\
&= \frac{\sum_{i<j} W^+_{i,j} \left(v_i - v_j\right)^2}{\sum_i \left(v_i^2 + \delta^2 + 2\delta v_i\right)} \\
&= \frac{\sum_{i<j} W^+_{i,j} \left(v_i - v_j\right)^2}{\sum_i v_i^2 + m\delta^2 + 2\delta \sum_i v_i} \\
&\overset{(a)}{\leq} \frac{\sum_{i<j} W^+_{i,j} \left(v_i - v_j\right)^2}{\sum_i v_i^2 + m\delta^2} \\
&\leq \frac{\sum_{i<j} W^+_{i,j} \left(v_i - v_j\right)^2}{\sum_i v_i^2} \\
&= R^+_{\boldsymbol{L}}(\boldsymbol{v}),
\end{aligned}
$$

where (a) holds by the fact that $\sum_i v_i \geq 0$. □

We now present the proof of Theorem 4.3.3.

Proof. Let $\boldsymbol{v}$ be the eigenvector related to the eigenvalue $\lambda_2(\boldsymbol{L})$. Without loss of generality, we assume $\|\boldsymbol{v}\| = 1$ and $v_1 \leq v_2 \leq \cdots \leq v_m$. Recall that $\mathbf{1}^\top \boldsymbol{v} = 0$. Then, we have that

$$
\lambda_2(\boldsymbol{L}) = R_{\boldsymbol{L}}(\boldsymbol{v}) = R^+_{\boldsymbol{L}}(\boldsymbol{v}) + R^-_{\boldsymbol{L}}(\boldsymbol{v}).
$$

Lower bounding $R^+_{\boldsymbol{L}}(\boldsymbol{v})$. Set $\delta = v_1$ and denote $\boldsymbol{u} = \boldsymbol{v} - \delta\mathbf{1}$. Then, we have that $0 = u_1 \leq \cdots \leq u_m$. Also note that $\delta^2 \leq 1$. Then, by Lemma C.3.1, it follows that $R^+_{\boldsymbol{L}}(\boldsymbol{v}) \geq R^+_{\boldsymbol{L}}(\boldsymbol{u})$.

We now define a random variable t on the support $[0, u_m]$, with probability density function $f(t) = \frac{2}{u_m^2} t$. One can verify that $\int_{t=0}^{u_m} \frac{2}{u_m^2} t \; dt = 1$, thus $f(t)$ is a valid probability density function. Then, for any interval $[a, b]$, it follows that the probability of t falling in the interval is

$$
\mathbb{P}[a \leq t \leq b] = \int_{t=a}^{b} \frac{2}{u_m^2} t \; dt = \frac{1}{u_m^2}(b^2 - a^2).
$$

Next, for some t, construct a random set $S_t = \{i \mid u_i \geq t\}$. Let $\omega^+(\partial S_t) = \sum_{i \in S_t, j \notin S_t} W^+_{i,j}$.

It follows that

$$\begin{aligned}
\mathbb{E}\left[w^+(\partial S_t)\right] &= \mathbb{E}\left[\sum_{i\in S_t, j\notin S_t} W^+_{i,j}\right] \\
&= \sum_{i<j} \mathbb{P}[u_j \le t \le u_i] W^+_{i,j} \\
&= \frac{1}{u_m^2} \sum_{i<j} (u_i - u_j)(u_i + u_j) W^+_{i,j} \\
&\le \frac{1}{u_m^2} \sqrt{\sum_{i<j} (u_i - u_j)^2 W^+_{i,j}} \sqrt{\sum_{i<j} (u_i + u_j)^2 W^+_{i,j}} \\
&= \frac{1}{u_m^2} \sqrt{R^+_{\boldsymbol{L}}(\boldsymbol{u}) \sum_i u_i^2} \sqrt{\sum_{i<j} (u_i + u_j)^2 W^+_{i,j}} \\
&\le \frac{1}{u_m^2} \sqrt{R^+_{\boldsymbol{L}}(\boldsymbol{u}) \sum_i u_i^2} \sqrt{2 \sum_i u_i^2 \deg^{H^+}(i)} \\
&\le \frac{1}{u_m^2} \sqrt{R^+_{\boldsymbol{L}}(\boldsymbol{u}) \sum_i u_i^2} \sqrt{2 \deg^{H^+}_{\max} \sum_i u_i^2} \\
&= \frac{\sum_i u_i^2}{u_m^2} \sqrt{2 \deg^{H^+}_{\max} R^+_{\boldsymbol{L}}(\boldsymbol{u})}.
\end{aligned}$$

Also note that $\mathbb{E}\left[|S_t|\right] = \sum_i \mathbb{P}[u_i \ge t] = \sum_i \frac{u_i^2}{u_m^2}$. As a result we obtain

$$\mathbb{E}\left[\omega^+(\partial S_t)\right] \le \mathbb{E}\left[|S_t|\right] \sqrt{2 \deg^{H^+}_{\max} R^+_{\boldsymbol{L}}(\boldsymbol{u})}.$$

Thus, we have $\mathbb{E}\left[\omega^+(\partial S_t) - |S_t| \sqrt{2 \deg^{H^+}_{\max} R^+_{\boldsymbol{L}}(\boldsymbol{u})}\right] \le 0$. This implies that $\exists S_t$ such that $\omega^+(\partial S_t) - |S_t| \sqrt{2 \deg^{H^+}_{\max} R^+_{\boldsymbol{L}}(\boldsymbol{u})} \le 0$. Rearranging we have,

$$R^+_{\boldsymbol{L}}(\boldsymbol{v}) \ge R^+_{\boldsymbol{L}}(\boldsymbol{u}) \ge \frac{\omega^+(\partial S_t)^2}{2 \deg^{H^+}_{\max} |S_t|^2} \tag{C.11}$$

Lower bounding $R^-_{\boldsymbol{L}}(\boldsymbol{v})$. Set $\alpha = \sqrt{\frac{1}{v_1^2 + v_m^2}}$ and denote $\boldsymbol{u} = \alpha \boldsymbol{v}$. Then, we have that $u_1^2 + u_m^2 = 1$. Note also that $R^-_{\boldsymbol{L}}(\boldsymbol{v}) = R^-_{\boldsymbol{L}}(\boldsymbol{u})$.

We now define a random variable t on the support $[u_1, u_m]$, with probability density function $f(t) = 2|t|$. One can verify that $\int_{t=u_1}^{u_m} 2|t| \; dt = 1$, thus $f(t)$ is a valid probability

density function. Then, for any interval $[a, b]$, it follows that the probability of t falling in the interval is

$$\mathbb{P}[a \leq t \leq b] = \int_a^b 2|t| \; dt = b^2 \operatorname{sign}(b) - a^2 \operatorname{sign}(a).$$

Since $[u_1, u_m] \subset [-1, 1]$, one can verify that $(a-b)^2/2 \leq \mathbb{P}[a \leq t \leq b]$. Let $\omega^-(\partial S_t) = \sum_{i \in S_t, j \notin S_t} W^-_{i,j}$. For some t, construct a random set $S_t = \{i \mid u_i \leq t\}$. It follows that

$$\begin{aligned}
\mathbb{E}\left[\omega^-(\partial S_t)\right] &= \mathbb{E}\left[\sum_{i \in S_t, j \notin S_t} W^-_{i,j}\right] \\
&= \sum_{i<j} \mathbb{P}[u_i \leq t \leq u_j] W^-_{i,j} \\
&\leq \frac{1}{2} \sum_{i<j} (u_i - u_j)^2 W^-_{i,j} \\
&= \frac{1}{2} R^-_{\boldsymbol{L}}(\boldsymbol{u}) \sum_i u_i^2 \\
&\leq \frac{1}{2} R^-_{\boldsymbol{L}}(\boldsymbol{u}),
\end{aligned}$$

where the last inequality follows from having $\sum_i u_i^2 \geq 1$ and $R^-_{\boldsymbol{L}}(\boldsymbol{u}) \leq 0$. Thus, we have $\mathbb{E}\left[\omega^-(\partial S_t) - \frac{1}{2} R^-_{\boldsymbol{L}}(\boldsymbol{u})\right] \leq 0$. This implies that $\exists S_t$ such that $\omega^-(\partial S_t) - \frac{1}{2} R^-_{\boldsymbol{L}}(\boldsymbol{u}) \leq 0$. Rearranging we have,

$$R^-_{\boldsymbol{L}}(\boldsymbol{v}) = R^-_{\boldsymbol{L}}(\boldsymbol{u}) \geq 2\omega^-(\partial S_t). \tag{C.12}$$

By minimizing (C.11) and (C.12) independently, and combining them, we conclude our proof. □

C.4 A Degree-Based Construction of the Kneser Graph

In Section 4.3.5, we used CVX [140] to solve problem (4.9) and, thus, obtain the dual variables $\boldsymbol{\mu}$ in problem (4.10) from which we construct the weights of the Johnson and Kneser graphs. Motivated by the trade-off between Cheeger constants of the positive and negative subgraphs, shown in Theorem 4.3.3, we show a simple non-trivial way (not necessarily op-

Algorithm 2 A construction of Kneser graph weights

Input: Level-2 weight matrix $\boldsymbol{M} = \widetilde{\boldsymbol{\Upsilon}}\boldsymbol{X}^{(2)}\widetilde{\boldsymbol{\Upsilon}}$, constant $c \in \mathbb{R}$.

1: $\deg(\mathcal{C}_1) \leftarrow \sum_{\mathcal{C}_2} M_{\mathcal{C}_1,\mathcal{C}_2},\ \forall \mathcal{C}_1 \in \binom{[n]}{2}$
2: Initialize $\boldsymbol{W}^{\mathcal{K}}$ as a zero matrix
3: **for all** $i < j < k < \ell \in [n]$ **do**
4: Assign the following such that $\psi_1 \geq \psi_2 \geq \psi_3$
5: $\psi_1 \leftarrow \deg(\{i,j\}) + \deg(\{k,\ell\})$
6: $\psi_2 \leftarrow \deg(\{i,k\}) + \deg(\{j,\ell\})$
7: $\psi_3 \leftarrow \deg(\{i,\ell\}) + \deg(\{j,k\})$
8: **if** $\psi_1 = \psi_2 = \psi_3$ **then**
9: $W^{\mathcal{K}}_{\{i,j\},\{k,\ell\}} \leftarrow 0,\ \ W^{\mathcal{K}}_{\{i,k\},\{j,\ell\}} \leftarrow 0,\ \ W^{\mathcal{K}}_{\{i,\ell\},\{j,k\}} \leftarrow 0$
10: **else if** $\psi_1 = \psi_2$ **then**
11: $W^{\mathcal{K}}_{\{i,j\},\{k,\ell\}} \leftarrow -c,\ \ W^{\mathcal{K}}_{\{i,k\},\{j,\ell\}} \leftarrow -c,\ \ W^{\mathcal{K}}_{\{i,\ell\},\{j,k\}} \leftarrow 2c$
12: **else**
13: $W^{\mathcal{K}}_{\{i,j\},\{k,\ell\}} \leftarrow -2c,\ \ W^{\mathcal{K}}_{\{i,k\},\{j,\ell\}} \leftarrow c,\ \ W^{\mathcal{K}}_{\{i,\ell\},\{j,k\}} \leftarrow c$
14: **end if**
15: **end for**
16: $\boldsymbol{W}^{\mathcal{K}} \leftarrow \boldsymbol{W}^{\mathcal{K}} + (\boldsymbol{W}^{\mathcal{K}})^\top$ {To symmetrize.}

Output: $\boldsymbol{W}^{\widetilde{\mathcal{G}}} \leftarrow \boldsymbol{M} + \boldsymbol{W}^{\mathcal{K}}$

timal) to directly construct the weights of the Kneser graph. The reason why we focus in the Kneser graph weights is because the fourth list of constraints in problem (4.9) can be expressed by two constraints for any $i < j < k < \ell$, as noted in Section 4.3.4. The latter fact implies that, for any $i < j < k < \ell$, the edge weights $W^{\mathcal{K}}_{\{i,j\},\{k,\ell\}}$, $W^{\mathcal{K}}_{\{i,k\},\{j,\ell\}}$, and $W^{\mathcal{K}}_{\{i,\ell\},\{j,k\}}$ need to sum to zero in order to fulfill the SoS constraints. As also noted in Section 4.3.4, at least one of the previous weights need to be negative unless all three are zero. With these considerations, we present our construction in Algorithm 2, which relies only on the node degrees and a constant real value.

The intuition behind Algorithm 2 is that the negative weight will be assigned to the edge that connects the two nodes that have the highest combined node degree. In Lines 8-9, if all three edges have the same combined node degree then we set all three weights to zero. In Lines 10-11, the edge with lowest combined node degree is set to $2c$, while the other edges that attain the same combined node degree are set to $-c$. In Line 13, the edge with highest combined node degree is set to $-2c$, while the other edges are set to c. It is clear that the SoS constraints will be fulfilled for each quadruple $i < j < k < \ell$. Finally, we note that if

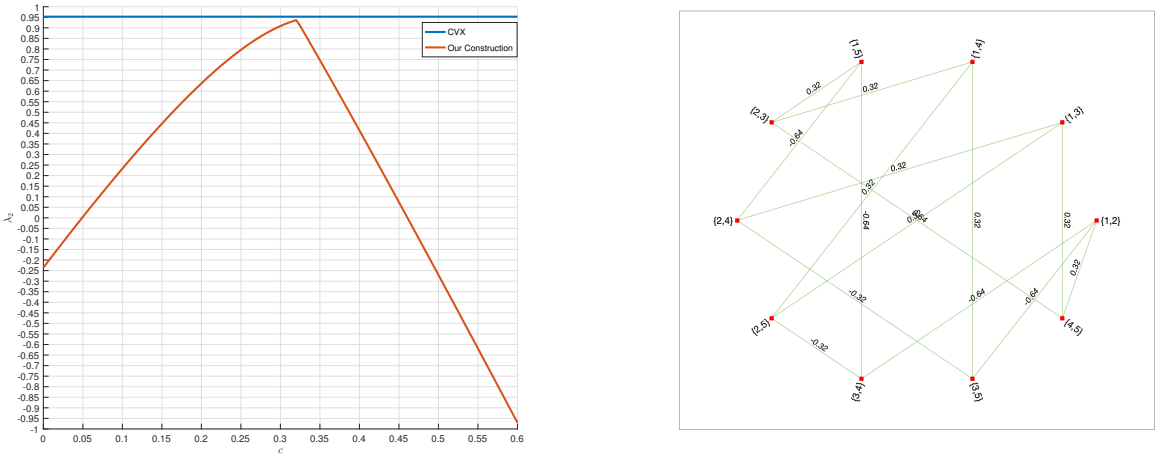


Figure C.1. *(Left)* The blue line is the algebraic connectivity found by CVX, i.e., 0.95 as pointed in Figure (4.4g). The red line is the algebraic connectivity of our construction in Algorithm 2 for different values of $c \in [0, 0.6]$. *(Right)* The Kneser graph weights for the optimal $c = 0.32$, which in effect differs from the weights found by CVX in Figure (4.4f).

$c = 0$ then simply the same input, $\boldsymbol{M}$, is returned. The latter implies that, for the *optimal* value of c, Algorithm 2 cannot return a weight matrix with lower algebraic connectivity than that of $\boldsymbol{M}$.[1]

Recall that $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = \lambda_2(\boldsymbol{L}^{\widetilde{\mathcal{G}}})$. In Figure C.1, we ran Algorithm 2 with input graph $\widetilde{\boldsymbol{\Upsilon}}\boldsymbol{X}^{(2)}\widetilde{\boldsymbol{\Upsilon}}$ equal to the graph in Figure (4.4d), and $c \in [0, 0.6]$. For each c, we plotted the algebraic connectivity of our construction. We observe that when $c = 0$, in effect $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = -0.24$ as pointed in Figure (4.4d). In this example, the optimal value of c is 0.32 and attains a $\lambda_2(\widetilde{\boldsymbol{\Lambda}})$ of 0.9368, which is very close to the value $\lambda_2(\widetilde{\boldsymbol{\Lambda}}) = 0.95$ found by CVX (see Figure (4.4g)). Finally, we also plot the Kneser graph weights for $c = 0.32$ following the construction in Algorithm 2.

[1]↑Recall that the SDP problem (4.6) would attain an algebraic connectivity equal to that of $\boldsymbol{M}$ if the optimal solution is $\overline{\boldsymbol{y}}\overline{\boldsymbol{y}}^\top$.

C.5 Proof of Theorem 4.4.4

Proof. The dual of problem 4.20 is given by:

$$\begin{aligned} \min_{\boldsymbol{V},\,\boldsymbol{\rho}} \quad & \mathrm{Tr}(\boldsymbol{V}) \\ \text{subject to} \quad & \boldsymbol{V} - \boldsymbol{X} - \sum_{i=1}^{k} \rho_i \cdot \boldsymbol{a}_i \boldsymbol{a}_i^\top \succeq 0, \\ & \boldsymbol{V} \text{ is diagonal.} \end{aligned} \tag{C.13}$$

Letting $\boldsymbol{\Lambda} \stackrel{\text{def}}{=} \boldsymbol{\Lambda}(\boldsymbol{V}, \boldsymbol{\rho}) = \boldsymbol{V} - \boldsymbol{X} - \sum_{i=1}^{k} \rho_i \cdot \boldsymbol{a}_i \boldsymbol{a}_i^\top$, with $\boldsymbol{V}$ diagonal. The Karush-Kuhn-Tucker (KKT) [99] optimality conditions are:

1. Primal Feasibility: $Y_{ii} = 1,\ \boldsymbol{a}_i^\top \boldsymbol{Y} \boldsymbol{a}_i = 0,\ \boldsymbol{Y} \succeq 0.$
2. Dual Feasibility: $\boldsymbol{\Lambda} \succeq 0.$
3. Complementary Slackness: $\langle \boldsymbol{\Lambda}, \boldsymbol{Y} \rangle = 0.$

Our approach is to find a pair of primal and dual solutions that simultaneously satisfy all KKT conditions above. Then, the pair witnesses strong duality between the primal and dual problems, which means that the pair is optimal. It is clear that $\boldsymbol{Y} = \overline{\boldsymbol{Y}} = \overline{\boldsymbol{y}}\overline{\boldsymbol{y}}^\top$ satisfies the primal constraints. Let $V_{ii} = (\boldsymbol{X}\overline{\boldsymbol{Y}})_{ii}$ and $\rho_i = -n$, if $\boldsymbol{\Lambda} \succeq 0$ then $\boldsymbol{V}$ and ρ satisfy the dual constraints. Thus, we conclude that if the condition $\boldsymbol{\Lambda} \succeq 0$ is met then $\overline{\boldsymbol{Y}}$ is an optimal solution.

For arguing about uniqueness, let us consider that $\lambda_2(\boldsymbol{\Lambda}) > 0$ and let $\widetilde{\boldsymbol{Y}}$ be another optimal solution to problem 4.20. From dual feasibility and complementary slackness we have that $\boldsymbol{\Lambda}\overline{\boldsymbol{y}} = 0$, which implies that $\overline{\boldsymbol{y}}$ spans all the null space of $\boldsymbol{\Lambda}$ since $\lambda_2(\boldsymbol{\Lambda}) > 0$. Finally, from primal feasibility we have that $\widetilde{\boldsymbol{Y}} = \overline{\boldsymbol{y}}\overline{\boldsymbol{y}}^\top$. Thus, $\lambda_2(\boldsymbol{\Lambda}) > 0$ is a sufficient condition for uniqueness.

From the arguments above, showing the condition $\lambda_2(\boldsymbol{\Lambda}) > 0$ suffices to guarantee that $\boldsymbol{Y} = \overline{\boldsymbol{Y}}$ is optimal and unique. As $\boldsymbol{X}$ and $\boldsymbol{V}$ are random variables by construction, we

next show when this condition is satisfied with high probability. By Weyl's theorem on eigenvalues, we have

$$\lambda_2(\boldsymbol{\Lambda}) = \lambda_2(\boldsymbol{\Lambda} - \mathbb{E}[\boldsymbol{\Lambda}] + \mathbb{E}[\boldsymbol{\Lambda}]) \geq \lambda_2(\mathbb{E}[\boldsymbol{\Lambda}]) + \lambda_1(\boldsymbol{\Lambda} - \mathbb{E}[\boldsymbol{\Lambda}]).$$

Let $\boldsymbol{M} = \boldsymbol{V} - \boldsymbol{X}$ and $\boldsymbol{N} = \sum_{i=1}^{k} \boldsymbol{a}_i \boldsymbol{a}_i^\top$, then we have $\mathbb{E}[\boldsymbol{\Lambda}] = \mathbb{E}[\boldsymbol{M}] + n \cdot \boldsymbol{N}$, where we remove the expectation on $\boldsymbol{N}$ since it is not a random matrix. To lower bound $\lambda_2(\mathbb{E}[\boldsymbol{M}] + n \cdot \boldsymbol{N})$, we first note that $\overline{\boldsymbol{y}} \in \{\mathsf{Null}(\boldsymbol{M}) \cap \mathsf{Null}(\boldsymbol{N})\}$, which means that we can invoke Lemma 4.4.2 for λ_2 instead of λ_1. Thus, we have

$$\lambda_2(\mathbb{E}[\boldsymbol{M}] + n \cdot \boldsymbol{N}) \geq \lambda_2(\mathbb{E}[\boldsymbol{M}]) + \epsilon_1 \tag{C.14}$$

$$\geq \epsilon_2 + \epsilon_1, \tag{C.15}$$

where $\epsilon_1 = \max_{i=n-k+1\ldots n} \left(\frac{n\lambda_i(\boldsymbol{N})+\Delta}{2} - \sqrt{\left(\frac{n\lambda_i(\boldsymbol{N})+\Delta}{2}\right)^2 - n\lambda_i(\boldsymbol{N}) \cdot \Delta \cdot (\boldsymbol{v}_i^\top \boldsymbol{\pi}_2)^2} \right)$ in eq.(C.14) follows from Lemma 4.4.2, and $\epsilon_2 = (1-2p)\frac{\phi_{\mathcal{G}}^2}{4\deg_{\max}(\mathcal{G})}$ in eq.(C.15) follows from Theorem 4.2.2. The term $\boldsymbol{\pi}_2$ in ϵ_1 corresponds to the Fiedler vector of $\mathcal{G}$ because the matrix $\boldsymbol{M}$ is a signed Laplacian of $\mathcal{G}$, that is, the matrix $\boldsymbol{L}$ and $\boldsymbol{M}$ share the same spectrum, and the i-th eigenvector of $\boldsymbol{M}$ is equal to the i-th eigenvector of $\boldsymbol{L}$ multiplied by $\bar{y}_i$. Since $\bar{y}_i^2 = 1$, only the second eigenvector of $\boldsymbol{L}$ appears in the expression, i.e., $\boldsymbol{\pi}_2$.

To lower bound $\lambda_1(\boldsymbol{\Lambda} - \mathbb{E}[\boldsymbol{\Lambda}])$, we first observe that $\boldsymbol{\Lambda} - \mathbb{E}[\boldsymbol{\Lambda}] = \boldsymbol{V} - \boldsymbol{X} \ - \mathbb{E}[\boldsymbol{V} - \boldsymbol{X}]$. Thus, we can further decompose the lower bound as follows: $\lambda_1(\boldsymbol{V} - \boldsymbol{X} \ - \mathbb{E}[\boldsymbol{V} - \boldsymbol{X}]) \geq \lambda_1(\boldsymbol{V} - \mathbb{E}[\boldsymbol{V}]) + \lambda_1(\mathbb{E}[\boldsymbol{X}] - \boldsymbol{X})$. Finally, for $\lambda_1(\boldsymbol{V} - \mathbb{E}[\boldsymbol{V}])$ and $\lambda_1(\mathbb{E}[\boldsymbol{X}] - \boldsymbol{X})$ we use Bernstein's inequality [81] with a similar setting to the one in the proof of Theorem 4.2.5 and obtain:

$$P\left(\lambda_1(\boldsymbol{V} - \mathbb{E}[\boldsymbol{V}]) \leq -\frac{\epsilon_1 + \epsilon_2}{2}\right) \leq n \cdot \mathrm{e}^{\frac{-3(\epsilon_1+\epsilon_2)^2}{24\sigma^2 + 8R(\epsilon_1+\epsilon_2)}}, \tag{C.16}$$

$$P\left(\lambda_1(\mathbb{E}[\boldsymbol{X}] - \boldsymbol{X}) \leq -\frac{\epsilon_1 + \epsilon_2}{2}\right) \leq n \cdot \mathrm{e}^{\frac{-3(\epsilon_1+\epsilon_2)^2}{24\sigma^2 + 8R(\epsilon_1+\epsilon_2)}}, \tag{C.17}$$

where $\sigma^2 = 4p(1-p)\deg_{\max}(\mathcal{G})$ and $R = 2(1-p)$. Combining equations (C.15), (C.16) and (C.17) we conclude our proof. □