Graph Partitioning and Semi-definite Programming Hierarchies

Ali Kemal Sinop

CMU-CS-12-121

May 15, 2012

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Venkatesan Guruswami, Chair
Anupam Gupta
Ryan O'Donnell
Ravi Kannan, M.S.R. India

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant numbers CCF-0635257, CCF-0747250, CCF-0915893, CCF-0953155, and CCF-1115525. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: approximation algorithms, certificate of infeasibility, column selection, graph partitioning, graph spectrum, Lasserre hierarchy, local rounding, semi-definite programming, strong duality.

To my mother, father and sister.

Abstract

Graph partitioning is a fundamental optimization problem that has been intensively studied. Many graph partitioning formulations are important as building blocks for divide-and-conquer algorithms on graphs as well as to many applications such as VLSI layout, packet routing in distributed networks, clustering and image segmentation. Unfortunately such problems are notorious for the huge gap between known best known approximation algorithms and hardness of approximation results. In this thesis, we study approximation algorithms for graph partitioning problems using a strong hierarchy of relaxations based on semi-definite programming, called Lasserre Hierachy.

Our main contribution in this thesis is a propagation based rounding framework for solutions arising from such relaxations. We present a novel connection between the quality of solutions it outputs and column based matrix reconstruction problem. As part of our work, we derive optimal bounds on the number of columns necessary together with efficient randomized and deterministic algorithms to find such columns. Using this framework, we derive approximation schemes for many graph partitioning problems with running times dependent on how fast the graph spectrum grows.

Our final contribution is a fast SDP solver for this rounding framework: Even though SDP relaxation has $n^{O(r)}$ many variables, we achieve running times of the form $2^{O(r)}\,\mathrm{poly}(n)$ by only partially solving the relevant part of relaxation. In order to achieve this, we present a new ellipsoid algorithm that returns certificate of infeasibility.

Acknowledgments

First and foremost, I owe my deepest gratitude to my advisor Venkat Guruswami. During the most turmoiled time of my PhD, just as I was planning to drop out of grad school; I was in for one of the most pleasant surprises of my life: Venkat transferred from UW to CMU and agreed to take me as a student. In fact, I could never have asked for a better advisor: Any of my success is due to him, and all my failures are due to me. Thanks, Venkat!

I will be forever indebted to Engin Dikici, my lifelong friend whom I met at UPenn; and Leo Grady, my mentor at SCR. If it was not for their help, I wouldn't even have been able to start PhD at CMU. Similarly I can never overstate the support of Gary Miller in starting the grad school.

I also want to thank Anupam Gupta, Ryan O'Donnell and Ravi Kannan for serving on my thesis committee. It was a reading group organized by Ryan in which gave me the inspiration behind my thesis. I also thank Alan Frieze, from whom I learnt a lot.

I couldn't have made it through all these years without the support of my friends. I especially want to thank Ravishankar Krishnaswamy, B. Aditya Prakash, Bekir Mugayitoğlu and Dafna Shahaf for all the good times at CMU and Pittsburgh. I also thank Deborah Cavlovich and Catherine Copetas for their continuous help in my PhD at CMU.

I do research solely to satiate my curiosity: An infinite scientific curiosity instilled by my parents. It should be no surprise then that my biggest thanks goes to them and my sister: Nazife, Yafes and Azize Sinop. Their continuous love, sacrifice, support and encouragement have allowed me to pursue my ambitions.

Contents

List of Tables

List of Algorithms

Chapter 1

Introduction

Graphs are ubiquitous structures in computer science, mathematics, and the natural and social sciences. For example, they are useful for modeling various networks like the internet, genetic networks, and social networks. An amazingly vast array of tasks arising in computer science, operations research, biology, social sciences, chemistry and physics can be cast as a certain class of combinatorial optimization problems on graphs, where we want to find a solution with **minimum cost** where cost is a function of how many edges are cut in various graphs. We collectively refer to this class as **graph partitioning** problems. For an example, consider sparsest cut problem.

Non-Uniform Sparsest Cut. Given two graphs G and H on n nodes, partition the nodes into two non-empty sets so as to:

$$\text{Minimize } \frac{\text{number of edges cut in } G}{\text{number of edges cut in } H}.$$

If we replace H with a clique graph, K, we obtain what is known as **Uniform Sparsest Cut** problem. This problem arises as a building block for divide-and-conquer algorithms on graphs as well as to many applications such as: Image segmentation [Shi and Malik, 2000, Sinop and Grady, 2007], VLSI layout [Bhatt and Leighton, 1984], packet routing in distributed networks [Awerbuch and Peleg, 1990], etc.

Instead of penalizing cut edges, we can also penalize *not* cutting an edge:

Minimum Uncut (Maximum Cut). Given a graph G on n nodes, partition the nodes into two sets so as to:

$$\text{Minimize the number of uncut edges in } G.$$

Similar to sparsest cut, there is a diverse list of applications ranging from solid state physics to printed circuit board design for this problem [Barahona et al., 1988].

Unfortunately, for an overwhelming majority of combinatorial optimization problems, finding the optimal solution turns out to be NP-hard. Therefore, unless P = NP, there are no efficient algorithms to solve any of the above problems optimally.

1.1 Approximation Algorithms

To cope with this intractability, one settles for solutions that are approximately optimal. For instance, can we design an efficient algorithm that always outputs a solution whose cost is at most twice that of the optimum? Formally, we define α-factor **approximation algorithm** for a problem as an algorithm which on every instance, outputs a solution whose cost is at most α times that of the optimal solution.

Over the last thirty-five years, this approach has been fruitful, giving rise to the field of approximation algorithms and leading to practical algorithms for a plethora of real-world optimization problems. We refer the reader to the book of Vazirani [2001] for an overview of this vast area. We will now review a common paradigm for obtaining approximation algorithms.

1.1.1 Convex Relaxations and Rounding Algorithms

A common approach for designing approximation algorithms is to give a convex formulation on which any optimal integer solution has a corresponding feasible solution with the same objective value. Such formulations are known as convex relaxations of the original problem and one can solve such convex optimization problems optimally (or near optimally for arbitrarily small error) using algorithms such as ellipsoid method Grötschel et al. [1993] in polynomial time.

This means in polynomial time, one can obtain a lower bound for the optimum value of original NP-hard problem. But our initial goal was to find an approximate integral solution. Furthermore there is no guarantee on the quality of estimate obtained from convex relaxation. If we think of the optimal solution for convex relaxation as a "fractional solution", we can try to "round" it to an integral solution. If we can also make sure that the rounding procedure produces solutions whose cost is at most α times that of the relaxation optimum, it means

solving the convex relaxation together with our rounding algorithm is an α-factor approximation for the original integral problem.

The use of relaxations based on linear programming (LP) in approximation algorithms, is a well-established approach which has spawned much work and a large variety of techniques. On the other hand, the use of semi-definite programming (SDP) relaxations in approximation algorithms is a more recent development, starting with the seminal work of Goemans and Williamson [1995] for Maximum Cut problem achieving a better approximation than is achievable by known LP approaches.

1.1.2 Integrality Gaps as Limitations

Suppose $\mathsf{OPT}_{\text{int}}(I)$ and $\mathsf{OPT}_{\text{convex}}(I)$ are optimal values of integer problem and its convex relaxation respectively on instance I. Being a convex relaxation, we always have $\frac{\mathsf{OPT}_{\text{int}}}{\mathsf{OPT}_{\text{convex}}} \geq 1$. Any rounding algorithm as described above that achieves a factor α approximation implies that:

$$\text{Integrality Gap} \stackrel{\text{def}}{=} \max_{I} \frac{\mathsf{OPT}_{\text{int}}(I)}{\mathsf{OPT}_{\text{convex}}(I)} \leq \alpha.$$

In the other direction, we can also try to lower bound this ratio. In order to do so, we need to exhibit an instance I such that:

- I has no integral solution with value $\leq s$ which means $\mathsf{OPT}_{\text{int}}(I) \geq s$,
- I has a feasible solution under our convex relaxation whose value is $\leq c$ which means $\mathsf{OPT}_{\text{convex}}(I) \leq c$.

Then for such instance,

$$\text{Integrality Gap} \geq \frac{\mathsf{OPT}_{\text{int}}(I)}{\mathsf{OPT}_{\text{convex}}(I)} \geq \frac{s}{c}.$$

This means there is no factor-$\frac{s}{c}$ rounding algorithm for our convex relaxation.

1.1.3 Hierarchies of Relaxations

Towards obtaining better approximations, a natural avenue is to utilize stronger relaxations that include greater number of constraints. There are numerous choices

of additional constraints that can be included to strengthen a given convex relaxation. Relaxation hierarchies are systematic procedures which work round-by-round: At each round, they produce a stronger convex relaxation at the cost of larger problem size. First such hierarchy was given by Sherali and Adams [1990] followed by Lovász and Schrijver [1991], both based on linear programming. The strongest known hierarchy is based on semi-definite relaxation given by Lasserre [2002], which forms the basis for all our approximation algorithms also. Typically these at r^{th} round, these hierarchies produce problems of size $n^{O(r)}$.

Even few rounds of Lasserre is already as strong as the SDPs used to obtain the best known approximation algorithms for several problems — for example, 3 rounds of Lasserre is enough to capture the ARV SDP relaxation for Sparsest Cut [Arora et al., 2009], and Chlamtac [2007] used the third level of the Lasserre hierarchy to get improvements for coloring 3-colorable graphs.

Furthermore such hierarchies are known to converge to a $0/1$ solution, i.e. have an integrality gap of 1, as number of rounds gets closer to n. However in such case, it takes an exponential time to solve these hierarchies anyway. The interesting question is then to characterize the problems for which small number of rounds of these hierarchies yields a better approximation. On the other hand, a lower bound showing that the integrality gap of the program obtained after many (say even $\Omega(n)$) levels of a hierarchy remains large, is a strong lower bound against a class of algorithms capturing most known ones.

For weaker hierarchies, many strong integrality gaps were known (see a recent survey of Chlamtac and Tulsiani [2011]). Starting with the seminal works of Schoenebeck [2008] and Tulsiani [2009], integrality gaps matching various known hardness of approximation results for Lasserre Hierachy were also found. These were extended to some graph partitioning problems such as densest sub-graph Bhaskara et al. [2012], uniform sparsest cut, balanced separator and maximum cut Guruswami et al. [2012]. However these integrality gaps are still not any close to ruling out potential of Lasserre Hierarchy to obtain, say, constant factor approximation.

In terms of positive results that use a larger (growing) number of Lasserre rounds, to the best of our knowledge, only two results existed prior to our work. Chlamtac and Singh [2008] used $O(1/\gamma^2)$ rounds of Lasserre hierarchy to find an independent set of size $\Omega(n^{\gamma^2/8})$ in 3-uniform hyper-graphs with an independent set of size γn. Karlin et al. [2010] showed that $1/\varepsilon$ rounds of Lasserre SDP gives a $(1+\varepsilon)$ approximation to the Knapsack problem.

1.2 Hardness of Approximation

What if beating the known approximation factors is NP-hard? Formally we say finding α-factor approximation is hard, if beating factor α is as hard as solving 3-SAT for example. This would mean that, unless P=NP, there can be no efficient algorithm to find α-approximate solutions. Starting with the breakthrough result known as PCP theorem [Arora et al., 1998, Arora and Safra, 1998] through the seminal works of Raz [1998] and Håstad [2001], it turned out indeed this was the case for many combinatorial optimization problems – the known approximation algorithms were also the best possible, assuming P$\neq$NP.

Unfortunately proving hardness of approximation for almost all NP-hard graph partitioning problems remained elusive. In fact, to the best of our knowledge, the best hardness factor we know is for minimum uncut problem which is $\frac{5}{4}$ due to Trevisan et al. [2000]. The difficulty faced in obtaining strong hardness of approximation results motivated two conjectures.

1.2.1 Unique Games Conjecture

Definition 1.1 (Unique Games Problem). *Given $\mathbb{L} = (G, k, \pi)$ where k is a positive integer, $G = (V, E)$ is a graph, and $\pi = (\pi_{u,v})_{(u,v)\in E}$ is a collection of permutation constraints associated with each edge $e = (u, v) \in E$ of the form $\pi_{u,v} : \{1, \ldots, k\} \to \{1, \ldots, k\}$ where $\pi_{u,v}$ is a bijection, the fraction of constraints satisfied by a k-labeling $f : V \to \{1, 2, \ldots, k\}$ is defined as the fraction of edges $(u, v) \in E$ for which $\pi_{u,v}(f(u)) = f(v)$.*

Khot [2002] conjectured that for any $\varepsilon > 0$ and k, it is NP-hard to decide if given $\mathbb{L}$ has a labeling that satisfies at least $1-\varepsilon$ fraction of constraints or no labeling can satisfy more than ε-fraction.

Surprisingly this conjecture led to a flurry of inapproximability results matching known approximation algorithms, thus proving their optimality; such as vertex cover [Khot, 2002], maximum cut [Khot et al., 2007], later extended by Raghavendra [2008] to all constraint satisfaction problems, non-uniform sparsest cut [Chawla et al., 2006], multi-way cut and 0-extension problems [Manokaran et al., 2008], ordering problems [Guruswami et al., 2011], independent set on bounded degree graphs [Austrin et al., 2011] and so on.

1.2.2 Small Set Expansion Conjecture

However a certain class of problems related to graph expansion resisted any attempts for proving hardness even under UGC. One such problem is uniform sparsest cut, which asks for a partitioning with minimum **sparsity**. Here sparsity of a partitioning on graph G is the ratio of total weight of separated edges to the total number of separated pairs. For Uniform Sparsest Cut problem as well as related problems, the best hardness of approximation factor we know is only $1 + \alpha$ for some constant $0 < \alpha < 0.001$ assuming a weaker form of Exponential Time Hypothesis [1] due to Khot [2006] and Ambühl et al. [2011].

Definition 1.2 (Small Set Expansion Problem)**.** *Given a positive constant $\mu \in (0, 1)$, for any graph $G = (V, E)$, small set expansion (SSE) on G is defined as the minimum sparsity over all binary partitions whose smaller side has size $\mu(1 \pm \varepsilon)|V|$.*

Raghavendra and Steurer [2010] conjectured that the decision problem for Small Set Expansion is hard, which was shown by Raghavendra et al. [2012] to be equivalent to the following: For any constant $\Phi > 0$ there exists $\mu > 0$ such that it is NP-hard to decide whether if there exists a set of size $\mu|V|$ whose expansion is $\leq \Phi$ or any set of size $(\mu/2, 2\mu)|V|$ has expansion $> 1 - \Phi$.

1.3 Our Motivation

A rich body of recent research has shown that for many optimization problems, the Unique Games conjecture (UGC) serves as a barrier to further improvements to the approximation factor achieved by efficient algorithms. In many cases, including all constraint satisfaction problems and various graph partitioning problems, the best algorithms are based on fairly simple semi-definite programming (SDP) relaxations. The UGC foretells that for these problems, no tighter relaxation than these simple SDPs will yield a better approximation ratio in the worst-case.

Hierarchies of convex relaxations. A natural question thus is to understand the power and limitations of potentially stronger SDP relaxations, for example those from various *hierarchies* of relaxations. These hierarchies are parameterized by an integer r (called rounds/levels) which capture higher order correlations between (roughly r-tuples of) variables (the basic SDP captures only pairwise correlations, and certain extensions like triangle inequalities pose constraints on triples). Larger the r, tighter the relaxation. The optimum of n'th level of the hierarchy, where n

[1] $\mathrm{NP} \not\subseteq \cap_{\varepsilon>0}\mathrm{BPTIME}(2^{n^{\varepsilon}})$.

is the number of variables in the underlying integer program, usually equals the integral optimum.

There are several hierarchies of relaxations that have been studied in the literature, such as Sherali-Adams Hierarchy of linear programs by Sherali and Adams [1990], Lóvasz-Schrijver Hierarchy by Lovász and Schrijver [1991], a "mixed" hierarchy combining Sherali-Adams linear programs with the base level SDP, and Lasserre Hierachy by Lasserre [2002] (see a recent survey of Chlamtac and Tulsiani [2011] focusing on their use in approximate combinatorial optimization). Of these hierarchies, the most powerful one is the Lasserre Hierachy (see [Laurent, 2003] for a comparison), and therefore holds the most potential for new breakthroughs in approximation algorithms. Arguably, Lasserre SDPs pose the currently strongest known threat to the k-Unique Games conjecture, as even the possibility of the 4'th level of Lasserre SDP relaxation improving upon the Goemans-Williamson 0.878 approximation factor for Maximum Cut has *not* been ruled out. Recently, it has also been shown that $O(1)$ rounds of Lasserre Hierachy are able to solve all candidate gap instances of Unique Games by Barak et al. [2012]. (On the other hand, for some of the weaker hierarchies, integrality gaps for super-constant rounds *are* known for various Unique-Games hard problems [Khot and Saket, 2009, Raghavendra and Steurer, 2009].) In light of the above, the power and limitations of the Lasserre Hierachy merit further investigation.

1.4 Lasserre Hierarchy Relaxation

Suppose we are given two degree-d multi-linear polynomials over variables V, $\mathbf{X}_V = [\mathbf{X}_u]_{u \in V}$: $\mathsf{p}, \mathsf{q} \in \mathbb{R}[\mathbf{X}_V]$. Our goal is to find an assignment, $\mathbf{x} \in \{0,1\}^V$, which minimizes eq. (1.1):

$$\begin{array}{ll} \text{Minimize} & \mathsf{p}(\mathbf{x}) \\ \text{subject to} & \mathsf{q}(\mathbf{x}) \geq 0. \\ & \mathbf{x} \in \{0,1\}^V. \end{array} \tag{1.1}$$

Observe that we can convert majority of our $0/1$ programming problems to this form easily, which means finding optimal solution to eq. (1.1) or even a feasible one is NP-hard. But is it at least possible to formulate eq. (1.1) as convex problem, say with size at most exponential in n? We will give a sequence of transformations and end up with an equivalent Semi-Definite Programming (SDP) problem with size $2^{O(n)}$. Rather than proving equivalence our transformations, we only prove it for the final problem eq. (1.4). We give a formal study later in Chapter 3.

Substituting Moments. First, we will express objective and constraints from eq. (1.1) as linear functions. To do so, we replace each $\prod_{u\in S} \mathbf{x}_u$ with a new variable, x_S, over all S. Here the sequence $[x_S]_{S\subseteq V}$ is intended to be a **moment sequence** for some optimal $\mathbf{x}$. Note $x_\emptyset = 1$. Since $x_S \in \{0,1\}$, it satisfies $x_S^2 = x_S$ for any subset S. In particular for any $A, B \subseteq V$, we should have:

$$x_A \cdot x_B = x_{A\cup B}.$$

One can easily check that if such reals, $[x_S]$, exist then indeed we have $x_S \in \{0,1\}$ for all S. If we use p_S to denote the coefficient of polynomial p on monomial $\prod_{i\in S} x_i$:

$$\mathsf{p}(\mathbf{x}) = \sum_S \mathsf{p}_S x_S$$

with a similar relation for q_S as well. Thus we get the following re-formulation for eq. (1.1):

$$\begin{array}{ll} \text{Minimize} & \sum_S \mathsf{p}_S x_S \\ \text{subject to} & \sum_S \mathsf{q}_S x_S \geq 0, \\ & x_\emptyset = 1, \\ & x_A \cdot x_B = x_{A\cup B} \text{ for all } A, B \subseteq V, \\ & x_A \in \mathbb{R}, \text{ for all } A. \end{array} \tag{1.2}$$

Introducing Vectors. Our next transformation is a technique common in semi-definite relaxations: Introduce a vector $\vec{x}_A$ for each subset A and use $\langle \vec{x}_A, \vec{x}_B\rangle$ instead of $x_A \cdot x_B$. We will refer to these vectors as **moment vectors**. Then:

$$\begin{array}{ll} \text{Minimize} & \sum_S \mathsf{p}_S x_S \\ \text{subject to} & \sum_S \mathsf{q}_S x_S \\ & x_\emptyset = 1, \\ & \langle \vec{x}_A, \vec{x}_B\rangle = x_{A\cup B} \text{ for all } A, B \subseteq V, \\ & \operatorname{rank}([\vec{x}_A]_A) = 1,\ [\vec{x}_A]_A \in \mathbb{R}^{\Upsilon, 2^V}. \end{array} \tag{1.3}$$

Handling the Polynomial Constraints. Recall we intended x_A's to be $\{0,1\}$ variables (and they still are, though this is not relevant at the time being). Thus in any feasible solution, for each P, the vectors $[\vec{y}(\mathsf{q})_A]$ defined as $\vec{y}(\mathsf{q})_A \leftarrow \sqrt{\langle \mathsf{q}, x\rangle}\vec{x}_A$ also satisfies

$$\langle \vec{y}(\mathsf{q})_A, \vec{y}(\mathsf{q})_B\rangle = \sum_S \mathsf{q}_S x_{A\cup B\cup S}.$$

Moment Matrix. Basic linear algebra tells us that such vectors exist iff their Gram matrix is positive semi-definite (PSD), denoted by $\succeq 0$. Here Gram matrix refers to the matrix whose entries represent inner products:

$$[\vec{x}_A]_A \text{ exists iff } \mathbf{M}(x) \stackrel{\text{def}}{=} [x_{A\cup B}]_{A,B\subseteq V} = \begin{bmatrix} x_\emptyset & x_{\{u\}} & x_{\{v\}} & & x_S & \\ x_{\{u\}} & x_{\{u\}} & x_{\{u,v\}} & \cdots & x_{S\cup\{u\}} & \cdots \\ x_{\{v\}} & x_{\{u,v\}} & x_{\{v\}} & & x_{S\cup\{v\}} & \\ & \vdots & & \ddots & \vdots & \\ x_S & x_{S\cup\{u\}} & x_{S\cup\{v\}} & \cdots & x_S & \cdots \\ & \vdots & & & \vdots & \ddots \end{bmatrix} \succeq 0.$$

After some algebra, we can express the non-negativity constraint as:

$$\mathbf{M}(P * x) \succeq 0,$$

where $* : \mathbb{R}^{2^V} \times \mathbb{R}^{2^V} \to \mathbb{R}^{2^V}$ is a bi-linear operator with $[\mathsf{q} * x]_A = \sum_B \mathsf{q}_B x_{A\cup B}$.

Obtaining an SDP. We are only one "rank constraint " away from an SDP. We will simply throw it away and obtain the following SDP:

$$\begin{array}{ll} \text{Minimize} & \langle Q, x\rangle \\ \text{subject to} & \mathbf{M}(P * x) \succeq 0 \text{ for all } P \in \mathcal{P}, \\ & \mathbf{M}(x) \succeq 0, \\ & x_\emptyset = 1,\ x \in \mathbb{R}^{2^V}. \end{array} \tag{1.4}$$

Rather surprisingly, Theorem 3.36 shows that this is still equivalent to eq. (1.1).

Lasserre Hierarchy Relaxation. Unfortunately eq. (1.4) is too large: it has 2^n variables. Therefore we relax the problem by imposing PSD-ness constraint only on the principal minor of $\mathbf{M}(x)$ with rows and columns corresponding to $\leq r$-subsets instead. The resulting formulation is called Lasserre Hierachy relaxation and was introduced by Lasserre [2002].

1.5 Our Contributions and Thesis Structure

In Chapter 2, we begin by reviewing our notation, then present basic mathematical background on linear algebra, generalized spectrum of two matrices, graphs

and basic matrices related to graphs, such as adjacency matrix, degree matrix, node-edge incidence matrix and most important of all, Laplacian matrix. Finally we end this chapter by giving some basic probabilistic inequalities: Markov inequality, Chernoff and Hoeffding bounds.

In Chapter 3, we first introduce some (minimal) algebraic background necessary to formally introduce $0/1$ programming problems over polynomials and their SDP relaxations based on moments. We end this chapter by presenting constructions for SDP vectors corresponding to the relaxations of indicator variables for all possible configurations and prove certain properties on them using the algebraic connection: These labeling vectors and their properties form a crucial part of our rounding algorithms.

In Chapter 4, we do a case study of approximating minimum bisection problem on a simple setting and present main ideas behind our rounding algorithm along with its analysis, where we relate the solution quality to column based matrix reconstruction. We intend this chapter to be an introduction for our rounding algorithm, and not as a formal treatment of minimum bisection problem. We will present a formal treatment later in Chapter 7 including comparison with existing literature.

In Chapter 5, we propose a simple algorithmic framework which turns out to be general enough to capture not only our rounding, but also other rounding algorithms known in the literature. The benefit of this abstraction becomes clear when we demonstrate how to, in principle, avoid constructing the whole solution, which has size $n^{O(r)}$, and instead only compute relevant portions of the solution with size $2^{O(r)}n^{O(1)}$ as need arises. Our main technical contribution is a separation oracle based ellipsoid algorithm which can also output a certificate of infeasibility. Using this algorithm, we show how to implement this framework so that we reduce the final running time to something of the form $2^{O(r)}n^{O(1)}$.

Chapters 7 and 8 are continuation of Chapter 4 and we present various approximation guarantees. However all our final algorithms and theorem statements now take advantage of the fast solver framework we developed in Chapter 5.

In Chapter 7, we analyze it in the context of quadratic integer programming problems. Then we re-state the rounding procedure in terms of our framework from Chapter 5 and bound the running time. Finally we end our chapter with an application of this method to individual problems: Minimum bisection, small set expansion, their k-way generalizations and independent set.

In Chapter 8, we consider the problems of minimum uncut and unique games: A direct application of the rounding from previous chapter yields poor bounds due to a dependence on lifted graph. In order to remove this dependence, we extend our rounding method from previous chapter using an embedding, enabling us to by-pass the lifted graph and relate the analysis to original constraint graph instead.

In Chapter 9, we present an extension of our basic rounding procedure for the problem of generalized sparsest cut.. In the special case of uniform sparsest cut, we compare the guarantees of our algorithm and another one based subspace enumeration and cut improvement. Finally we argue why subspace enumeration based methods will fail in the case of non-uniform sparsest cut. In the analysis of all our rounding methods, the crucial step is always a relation between the cost of solution for a specific seed set to how well the corresponding columns for the seeds approximate a related matrix in terms of Frobenius norm.

In Chapter 10, we present our contribution for the problem of choosing minimum number of columns from a matrix so as to minimize the reconstruction error in Frobenius norm. Basically we prove upper bounds for the number of columns necessary and show how to find such columns efficiently by presenting both randomized and deterministic algorithms. Finally we prove that our upper bounds are best possible up to low order terms by exhibiting a class of matrices.

In Chapter 11, we analyze the structure of primal and dual formulations for moment based SDP relaxations (including Lasserre hierarchy) which we introduced back in Chapter 3. First note that there are two potential pitfalls associated with primals and duals of convex programs: (1) There might be a positive duality gap; (2) Even in the absence of a duality gap, primal or dual optima might not be attained. Unlike LP's, there are many SDP formulations for which these problems occur. Our main contribution in this chapter is an analysis of our relaxations from a dual perspective and prove that above issues do not occur.

In Chapter 12, we conclude our thesis, summarize our contributions and discuss possible directions for future research.

1.6 Bibliographic Note

Most of the research that appears in this thesis was either published elsewhere in some form, or is under submission. Chapters 4, 7 and 8 are based on [Guruswami and Sinop, 2011]. Chapter 5 is based on [Guruswami and Sinop, 2012c]. Chapter 9

is based on [Guruswami and Sinop, 2012a]. Chapter 10 is based on [Guruswami and Sinop, 2012b].

Chapter 2

Background

We start by presenting basic mathematical background and notations we use throughout the whole thesis.

2.1 Sets and Families

For any positive integer n, let $[n] \stackrel{\text{def}}{=} \{1, 2, \ldots, n\}$. We will use $\emptyset$ to denote empty set.

Notation 2.1 (Subsets and Power Sets)**.** *Given set A, let 2^A be its **power set**, i.e. set of all subsets. For any real k, we will use $\binom{A}{k}$, $\binom{A}{\leq k}$ and $\binom{A}{\geq k}$ (equivalently $A_{=k}, A_{\leq k}, A_{\geq k}$) to denote the set of all subsets of A having size exactly k, at most k and at least k respectively.*

Observe that $2^A = \binom{A}{\geq 0} = A_{\geq 0}$, $\emptyset = \binom{A}{0} = A_{=0}$ and $\binom{A}{\geq 1} = A_{\geq 1}$ is the set of non-empty subsets of A.

Notation 2.2 (Family of Subsets)**.** *Given V, $\mathcal{F}$ is a **family** over V if $\mathcal{F} \subseteq 2^V$.*

Notation 2.3 (Elementwise Union)**.** *For any pair of families $\mathcal{F}, \mathcal{G} \subseteq 2^V$, let $\mathcal{F} \biguplus \mathcal{G} \subseteq 2^V$ be the family obtained by element-wise unions of $\mathcal{F}$ and $\mathcal{G}$:*

$$\mathcal{F} \biguplus \mathcal{G} \stackrel{\text{def}}{=} \left\{ A \cup B \middle| A \in \mathcal{F}, B \in \mathcal{G} \right\}.$$

Definition 2.4 (Downward Closedness)**.** *For any set V, given a family of its subsets $\mathcal{F} \subseteq 2^V$, $\mathcal{F}$ is a **downward closed family** if whenever $S \in \mathcal{F}$, $\mathcal{F}$ also contains all subsets of S:*

$$S \in \mathcal{F} \iff 2^S \subseteq \mathcal{F}.$$

*We call such $\mathcal{F}$ a **down family over** V.*

Example 2.5.
- $\mathcal{F} \leftarrow \{\emptyset, \{1\}, \{1,2\}\} \not\ni \{2\}$ *is not a down family.*
- $\mathcal{G} \leftarrow \{\emptyset, \{3\}\}$ *is a down family.*
- $\mathcal{F} \biguplus \mathcal{G} = \{\emptyset, \{1\}, \{1,2\}, \{3\}, \{1,3\}, \{1,2,3\}\}$.

Claim 2.6.
- $2^S \biguplus 2^T = 2^{S \cup T}$.
- $V_{\leq p} \biguplus V_{\leq q} = V_{\leq p+q}$.
- $\mathcal{F} \biguplus \mathcal{F}$ *iff* $\mathcal{F} = 2^S$ *for some* S.

Definition 2.7 (k-way Partitionings)**.** *Given set V, for any k-collection of its subsets, $(U_1 \subset V, \ldots, U_k \subset V)$ we say it is a **k-way partitioning** of V if and only if all U_i's are disjoint with their union equal to V:*

$$V = U_1 \sqcup U_2 \sqcup \ldots \sqcup U_k.$$

*We will refer to it as a **proper partitioning** if all U_i's are non-empty in addition.*

2.2 Linear Algebra

Notation 2.8 (Reals, Rationals and Integers)**.** *Let $\mathbb{R}$, $\mathbb{Q}$, $\mathbb{Z}$ and $\mathbb{N}$ be the set of **reals**, **rationals**, **integers** and **natural numbers**. Given a subset of reals $\mathbb{F} \subseteq \mathbb{R}$, we use $\mathbb{F}_+$ and $\mathbb{F}_{++}$ to denote set $\mathbb{F}$ restricted to non-negative and positive numbers respectively.*

Notation 2.9 (Vectors and Matrices)**.** *Given finite sets A, B and a subset of reals $R \subseteq \mathbb{R}$, we will use R^A and $R^{A,B}$ to denote set of **vectors** and **matrices** over R whose **rows** and **columns** are identified with elements of A and B respectively. For any function $f : A \to R$ (resp. $g : A \times B \to R$), we will use $[f(u)]_{u \in A}$ (resp. $[g(u,v)]_{(u,v) \in A \times B}$) to denote the vector (resp. matrix) whose value at row u (resp. row u and column v) is equal to $f(u)$ (resp. $g(u,v)$).*

Notation 2.10 (Matrices as Collection of Vectors)**.** *Given A, B and a vector valued function $\vec{f} : B \to \mathbb{R}^A$, we use $[\vec{f}(u)]_{u \in B} \in \mathbb{R}^{A,B}$ to refer to the matrix whose columns are $\vec{f}(u)$ over all $u \in A$.*

Notation 2.11 (Minors)**.** *Given vector $x \in \mathbb{F}^A$ and matrix $Y \in \mathbb{F}^{A \times B}$, for any $C \subseteq A$ and $D \subseteq B$ let $x_C \in \mathbb{F}^C$ and $Y_{C,D} \in \mathbb{F}^{C \times D}$ denote the minors of x, Y on rows C and columns D.*

Notation 2.12 (Direct Sum). *Given two sets $K \subseteq \mathbb{R}^A, L \subseteq \mathbb{R}^B$, we define their **direct sum** as:*

$$K \circ L \stackrel{\text{def}}{=} \left\{(x_A, y_{B\setminus A}) \middle| x \in K, y \in L\right\} \subseteq \mathbb{R}^{A\cup B}.$$

Notation 2.13 (Norms, Inner Products and Normalized Vectors). *Let $\|x\|_p$ be its p-**norm** with $\|x\| \stackrel{\text{def}}{=} \|x\|_2$. For any $x \neq 0$, we will use $\overline{x} \stackrel{\text{def}}{=} x/\|x\|$ to denote the **normalized vector** for x. As a convention, if $x = 0$, we will take $\overline{x} = 0$ as well. For any $x, y \in \mathbb{R}^A$, let $\langle x, y\rangle = x^T y$ be their **inner product** $\sum_{a\in A} x_a y_a$.*

Notation 2.14 (Standard Matrix Functions). *We will use $\|Y\|_F$, $\mathrm{Tr}(Y)$, $|Y|$, Y^T, Y^{-1}, $Y^\dagger$ and Y^p to denote Frobenius norm of a matrix Y, its trace, transpose, inverse, pseudo-inverse and p^{th} power respectively, whenever defined.*

Notation 2.15 (Symmetric Matrices). *Given finite A, let $\mathbb{S}^A$ be the set of real symmetric matrices over rows and columns A.*

Notation 2.16 (Constant Valued Vectors, Matrices and Identity Matrix). *Given finite set of rows A and columns B, we use 0_A and $\mathbb{1}_A$ to denote all 0's and all 1's vector over A. Similarly we use $0_{A,B}$, $\mathbb{1}_{A,B}$ and $I_{A,B}$ to denote all 0's, all 1's and **identity matrix** over A and B. When there is no room for ambiguity, we will drop the subscripts and use $0, \mathbb{1}, I$ instead.*

Notation 2.17 (Indicator Vectors and Canonical Basis). *Given finite A, for any **predicate** of the form $f : A \to \{\text{false}, \text{true}\}$, we use $\mathbb{1}_f \in \{0,1\}^A$ to denote the **indicator vector** of f so that for any $j \in A$:*

$$(\mathbb{1}_f)_j = \begin{cases} 1 & \text{if } f(j) \text{ is true,} \\ 0 & \text{else.} \end{cases}$$

*If f is the membership predicate for some set B, then $\mathbb{1}_f$ corresponds to the **indicator vector** for B, which we will denote by $\mathbb{1}_B$ instead.*

*We denote the **canonical basis** of $\mathbb{R}^A$ as $\{\mathbb{1}_a \mid a \in A\}$.*

Notation 2.18 (Indicator Functions). *Given a predicate of the form $f : A \to \{\text{false}, \text{true}\}$ for some infinite set R, we use $[\![f]\!]$ to denote the **indicator function** of f. For any $x \in R$:*

$$[\![f]\!](x) = \begin{cases} 1 & \text{if } f(j) \text{ is true,} \\ 0 & \text{else.} \end{cases}$$

Definition 2.19 (Partitioning Representation and k-labeling Functions)**.** *We represent any partitioning $U_1 \sqcup \ldots \sqcup U_k = V$ with its **indicator vector** $\mathbf{x} = [\mathbf{x}_{(u,i)}] \in \{0,1\}^{V \times [k]}$ where $\mathbf{x}_{(u,i)}$ is 1 if $u \in U_i$ and 0 else. When $k = 2$, we will use $\mathbf{x} = [\mathbf{x}_u] \in \{0,1\}^V$ instead.*

*We also associate partitions with **labeling functions**: We want to assign a label from $[k]$ to each element of V. For any $S \subseteq V$, we use $[k]^S$ to denote the set of all k-labelings of S, $[k]^S \stackrel{\text{def}}{=} \{f \big| f : S \to [k]\}$.*

Definition 2.20 (Positive (Semi)Definite Ordering)**.** *We say a matrix $Y \in \mathbb{S}^A$ is positive semi-definite (PSD), denoted by $Y \succeq 0$ if $\forall x : x^T Y x \geq 0$. Further we say Y is positive definite (PD), $Y \succ 0$, if $\forall x : x^T A x > 0$ whenever $x \neq 0$.*

Finally we use $\mathbb{S}^A_+$ and $\mathbb{S}^A_{++}$ to denote the set of all PSD and PD matrices on rows and columns A respectively. Note $\mathbb{S}^A_{++} \subset \mathbb{S}^A_+ \subset \mathbb{S}^A$.

The following are well known characterizations of PSD-ness, therefore we skip their proofs.

Proposition 2.21. *Given $X \in \mathbb{S}^A$, $X \succeq 0$ iff $\operatorname{Tr}[X \cdot Y] \geq 0$ for all $Y \succeq 0$.*

Theorem 2.22 (Schur's Complement Criteria)**.** *Given disjoint sets A, B and matrices $X \in \mathbb{S}^A, Y \in \mathbb{R}^{A,B}, Z \in \mathbb{S}^B$:*

$$W \stackrel{\text{def}}{=} \begin{bmatrix} X & Y^T \\ Y & Z \end{bmatrix} \succeq 0 \iff X \succeq 0 \text{ and } \underbrace{Z - Y^T X^\dagger Y}_{\text{Schur's complement}} \succeq 0.$$

$\mathbb{S}^A_+$ can be thought of as generalization of $\mathbb{R}_+$. For example, we can take *square roots*:

Theorem 2.23. *Given $Y \in \mathbb{S}^V$, $Y \succeq 0$ iff there exists matrix $\vec{X} = [\vec{x}_u]_{u \in V} \in \mathbb{R}^{\Upsilon, V}$ such that $\vec{X}^T \vec{X} = Y$ for some set $\Upsilon : |\Upsilon| \leq |V|$. Given such $\vec{X}$, we refer to Y as the **Gram matrix** of $\vec{X}$. Similarly we refer to $\vec{X}$ as the **Gram decomposition** of Y.*

Theorem 2.24. *Given $Y \in \mathbb{S}^V$, there exists reals $\lambda_1 \leq \ldots \leq \lambda_{|V|}$ and unit vectors $\vec{z}_1, \vec{z}_2, \ldots, \vec{z}_{|V|} \in \mathbb{R}^V$ such that:*

$$Y = \sum_i \lambda_i \vec{z}_i \vec{z}_i^T, \ \langle \vec{z}_i, \vec{z}_j \rangle = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else.} \end{cases}$$

*We call this **eigen decomposition** of Y. Moreover $\lambda_1 \geq 0 (\text{resp. } \lambda_1 > 0)$ iff $Y \succeq 0 (\text{resp. } Y \succ 0)$.*

Notation 2.25 (Projection Operators). *For any matrix $X \in \mathbb{R}^{A,B}$, we will use X^{Π} and $X^{\perp}$ to denote the **projection matrices** onto the column span of X and onto its orthogonal complement respectively.*

Notation 2.26 (Support). *For any vector $x \in \mathbb{R}^A$, we will use $\mathrm{support}(x)$ to denote the set of its non-zero coordinates:*

$$\mathrm{support}(x) \stackrel{\mathrm{def}}{=} \{i \in A : x_i \neq 0\}.$$

Observe that $\|x\|_0 = |\mathrm{support}(x)|$.

2.3 Geometry

Definition 2.27 (Convex Set). *Given $X \subseteq \mathbb{R}^A$, X is a **convex set** if for any $y, z \in X$ and real $\theta \in [0,1]$, $\theta y + (1-\theta)z \in X$.*

Definition 2.28 (Convex Hull). *Given $X \subseteq \mathbb{R}^A$, convex hull of X is defined as:*

$$\mathrm{convex}(X) \stackrel{\mathrm{def}}{=} \bigcap_{\substack{C \text{ is convex} \\ C \supseteq X}} C.$$

Notation 2.29 (Minkowski Sum). *Given two sets $K, L \subseteq \mathbb{R}^A$, their **Minkowski sum** is defined as the following set:*

$$K + L \stackrel{\mathrm{def}}{=} \left\{x + y \middle| x \in K, y \in L\right\} \subseteq \mathbb{R}^A.$$

Notation 2.30 (Balls). *Given a set $K \subseteq \mathbb{R}^A$ and non-negative real $\varepsilon \geq 0$, we define $\mathbb{B}(K, \pm\varepsilon)$ in the following way.*

$$\mathbb{B}(K, \varepsilon) \stackrel{\mathrm{def}}{=} \left\{x \in \mathbb{R}^A | \exists y \in K \text{ s.t. } \|y - x\|_2 \leq \varepsilon\right\}.$$

$$\mathbb{B}(K, -\varepsilon) \stackrel{\mathrm{def}}{=} K \setminus \mathbb{B}(\mathbb{R}^A \setminus K, \varepsilon).$$

Observe that for $y \in \mathbb{R}^A$, $\mathbb{B}(y, \varepsilon)$ is the $|A|$-dimensional sphere with origin y, with $\mathbb{B}(K, \varepsilon)$ being Minkowski addition of sphere of radius ε to K and $\mathbb{B}(K, -\varepsilon)$ being Minkowski subtraction of sphere of radius ε from K.

Observation 2.31. *For any convex body $K \subseteq \mathbb{R}^A$ and non-negative reals $\varepsilon, \varepsilon_1, \varepsilon_2$, the following hold:*

1. $\mathbb{B}(\mathbb{B}(K,\varepsilon),-\varepsilon) = K,\ \mathbb{B}(\mathbb{B}(K,-\varepsilon),\varepsilon) \subseteq K.$
2. $\mathbb{B}(\mathbb{B}(K,\varepsilon_1),\varepsilon_2) = \mathbb{B}(K,\varepsilon_1+\varepsilon_2).$
3. $\mathbb{B}(\mathbb{B}(K,-\varepsilon_1),-\varepsilon_2) = \mathbb{B}(K,-\varepsilon_1-\varepsilon_2).$

Proof. [See Grötschel et al., 1993] □

Notation 2.32 (Volumes). *Given $K \subseteq \mathbb{R}^A$ with $|A| = d$, we will use $\mathrm{vol}_d(K)$ to denote d-dimensional volume of K, provided it exists. Furthermore for any non-negative real $\varepsilon \geq 0$, let $\mathrm{vol}_d(\varepsilon)$ be the volume of d-dimensional ball of radius ε. We will use $\mathrm{vol}_d^{-1}(K)$ to denote the radius of a d-dimensional sphere whose volume is equal to $\mathrm{vol}_d(K)$ so that*

$$\mathrm{vol}_d(K) = \mathrm{vol}_d(\mathrm{vol}_d^{-1}(K)).$$

2.4 Convex Optimization and Semi-definite Programming

As mentioned in the introduction, our approach for approximating graph partitioning problems is to express a certain convex relaxation for them, called semi-definite programming: These problems form one of the strongest convex formulations we know and they have been indispensable in designing approximation algorithms, starting with the seminal work of Goemans and Williamson [1995]. We first define basic terminology associated with convex optimization problems.

Definition 2.33 (Convex Optimization). *Given a convex set $K \subseteq \mathbb{R}^A$ and a convex function $f : \mathbb{R}^A \to \mathbb{R}$, consider the following:*

$$\textit{Infimum } f(x) \textit{ subject to } x \in K. \tag{2.1}$$

*We call such problems as **convex optimization problems**. Here f is the **objective function** and K is the **feasible set**. For any $x \in \mathbb{R}^A$ if $x \in K$ we say x is a **feasible solution** to eq. (2.1). We refer to the objective value achieved by x, $f(x)$, as the **value of solution** x. If no such x exists, i.e. $K = \emptyset$, we say eq. (2.1) is **infeasible**. Let $\mathsf{OPT} \in \mathbb{R} \cup \{\pm\infty\}$ denote the optimum value of eq. (2.1). We assume the following convention:*

$$\mathsf{OPT} = \begin{cases} +\infty & \textit{if eq. (2.1) is infeasible,} \\ -\infty & \textit{if eq. (2.1) is unbounded from below,} \\ \in \mathbb{R} & \textit{else.} \end{cases}$$

*Finally if there exists feasible solution x with $f(x) = \mathsf{OPT}$ we say x is an **optimal solution** to eq. (2.1).*

Definition 2.34 (SDPs). *Given a linear function $M : \mathbb{R}^A \to \mathbb{S}^B$, which maps vectors from $\mathbb{R}^A$ to symmetric matrices $\mathbb{S}^B$, and a symmetric matrix $C \in \mathbb{S}^B$ consider the following convex optimization problem:*

$$\textit{Infimum } \operatorname{Tr}[C \cdot M(x)] \textit{ subject to } M(x) \succeq 0. \tag{2.2}$$

*We refer to problems of the form eq. (2.2) as **semi-definite programming (SDP)** problems.*

Remark 2.35. *Equation (2.2) might seem rather unusual as the standard definition is:*

$$\textit{Infimum } \langle c, x \rangle \textit{ subject to } M(x) \succeq 0 \tag{2.3}$$

for some vector $c \in \mathbb{R}^A$. It is an easy exercise to show that eqs. (2.2) and (2.3) are equivalent. However we prefer eq. (2.2) as it naturally fits in the geometric theme of our thesis.

The way we formulated in eq. (2.2), it is rather difficult to "imagine" what feasible (let alone optimal) solutions look like. But together with $M(x) \succeq 0$, we can use Theorem 2.23 to interpret $M(x)$ as Gram matrix of some vectors and obtain the following more intuitive characterization:

Proposition 2.36. *The following is equivalent to eq. (2.2):*

$$\textit{Infimum } \sum_{i,j \in B} C_{i,j} \langle \vec{x}_i, \vec{x}_j \rangle \textit{ subject to } M(x) = \Big[\langle \vec{x}_i, \vec{x}_j \rangle \Big]_{i,j} \textit{ for some } x \in \mathbb{R}^A. \tag{2.4}$$

Remark 2.37 (SDP Duality). *SDP problems, such as the one given in eq. (2.2), are part of a certain class of convex optimization problems, called linear conic programming, which we will present later in Chapter 11. This abstraction will prove to be extremely useful when we want to talk about duality.*

2.5 Conic Ordering

When we are talking about polynomials, it will be more convenient to work without an explicit embedding into a real space.

Definition 2.38 (Bilinear Form). *Given two linear spaces E_1, E_2 and a function $f : E_1 \times E_2 \to \mathbb{R}$, we call f a **bilinear form** (or **inner product**) between E_1 and E_2 provided that:*

- *$f(p+q, r) = f(p, r) + f(q, r)$ for any $p, q \in E_1, r \in E_2$;*

- $f(p, q + r) = f(p, q) + f(p, r)$ *for any* $p \in E_1, q, r \in E_2$*;*
- $f(\alpha p, q) = f(p, \alpha q) = \alpha f(p, q)$ *for any* $p \in E_1, q \in E_2, \alpha \in \mathbb{R}$*.*

Lemma 2.39 (Adjoint)**.** *Given any inner product* $f : E_1 \times E_2 \to \mathbb{R}$*, any linear transformation* $T : E_1 \to E_1$ *(resp.* $U : E_2 \to E_2$*) has an* ***adjoint*** *transformation over* f*,* $\widehat{T} : E_2 \to E_2$ *(resp.* $\widehat{U} : E_2 \to E_2$*) such that for any* $p \in E_1, q \in E_2$*:*

$$f(T(p), q) = f(p, \widehat{T}(q)) (resp. f(p, U(q)) = f(\widehat{U}(p), q)).$$

Definition 2.40 (Cone)**.** *Given a linear space* E_1*,* $K \subseteq E_1$ *is a* ***cone*** *if for any* $x \in K$ *and non-negative real* α*,* $\alpha x \in K$*.*

Definition 2.41 (Dual Cone)**.** *Given linear spaces* E_1, E_2 *and subset* $K \subseteq E_1$*; for any inner product* $f : E_1 \times E_2 \to \mathbb{R}$*,* ***dual cone*** *of* K *over* f *is defined as:*

$$K^* \stackrel{\text{def}}{=} \{q \in E_2 \mid f(p, q) \geq 0 \text{ for all } p \in K\} \subseteq E_2.$$

Notation 2.42 (Conic Ordering)**.** *Given a convex cone* $K \subseteq E_1$*, for any* $p, q \in E_1$*, we say*

$$x \geq_K y \iff x - y \in K.$$

We define $\leq_K, >_K$ *and* $<_K$ *similarly.*

2.6 Generalized Eigenvalues

In this section, we will introduce the generalized eigenvalues and eigenvectors of a pair of symmetric matrices, $X \in \mathbb{S}^A$ and $Y \in \mathbb{S}^A_+$.

Definition 2.43 (Generalized Eigenvalues and Eigenvectors)**.** *Given* $X \in \mathbb{S}^A$ *and* $Y \in \mathbb{S}^A_+$*, for any positive integer* $j \leq \operatorname{rank}(Y)$ *we inductively define* j^{th} *smallest* ***generalized eigenvalue*** *of* X *and* Y*,* λ_j *along with corresponding* ***generalized eigenvector*** z_j *as:*

$$z_j \stackrel{\text{def}}{=} \operatorname{argmin} \left\{ \frac{z^T X z}{z^T Y z} \middle| z \in \mathbb{R}^A,\ z^T Y z = 1 \text{ and } z^T Y z_i = 0 \text{ for all } i < j. \right\},$$

$$\lambda_j(X; Y) \stackrel{\text{def}}{=} \frac{z_j^T X z_j}{z_j^T Y z_j}.$$

We refer to $(\lambda_1, \lambda_2, \ldots, \lambda_{\operatorname{rank}(Y)})$ *as the* ***generalized spectrum*** *of matrices* X *and* Y*. As a shorthand, we use:*

$$\lambda_{\min}(X; Y) \stackrel{\text{def}}{=} \lambda_1,\ \lambda_{\max}(X; Y) \stackrel{\text{def}}{=} \lambda_{\operatorname{rank}(Y)}.$$

Definition 2.44 (Eigenvalues)**.** *When Y is the identity matrix, $Y = I$, we refer to $\lambda_i = \lambda_i(X, I)$ simply as **eigenvalues** of X.*

Definition 2.45 (Normalized Eigenvalues)**.** *When $Y = \operatorname{diag}(X)$, we refer to $\lambda_i = \lambda_i(X, \operatorname{diag}(X))$ as **normalized eigenvalues** of X.*

Observe that the above definition coincides with the variational (Courant-Fischer) characterization of eigenvectors when Y is the identity matrix.

Remark 2.46. *At the first glance our variational definition might seem odd, as the traditional way to define eigenvalues and eigenvectors is through the solutions of equation $Xz = \lambda Yz$. Even though these definitions coincide in the case when Y is non-singular (i.e. $Y \in \mathbb{S}^A_{++}$) variational characterization is stronger when Y is a singular matrix.*

Theorem 2.47. *Given $X \in \mathbb{S}^A_+$ and $Y \in \mathbb{S}^A_+$, the following holds. There exists $\operatorname{rank}(Y)$ many generalized eigenvectors which satisfies the following:*

$$X \succeq \sum_{j=1}^{\operatorname{rank}(Y)} \lambda_j z_j \cdot {z_j}^T; \;\; Y = \sum_{j=1}^{\operatorname{rank}(Y)} z_j \cdot {z_j}^T; \;\; z_i^T Y z_j = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else;} \end{cases}$$

Here $z_j = z_j(X; Y)$ and $\lambda_j = \lambda_j(X; Y)$.

Provided that $\operatorname{range}(X) \subseteq \operatorname{range}(Y)$, this inequality is tight and:

$$\lambda_{\min}(X; Y) \cdot Y \preceq X \preceq \lambda_{\max}(X; Y) \cdot Y. \tag{2.5}$$

2.7 Graphs and Laplacian Matrices

Notation 2.48 (Graphs)**.** *All our graphs will be: (1) Loop-less, (2) Undirected, (3) Non-negatively weighted. We will use $G = (V, E, W)$ to denote such a graph defined on the set of nodes V and **edges** $E \subseteq \binom{V}{2}$ with **edge weights** $W = [w^G_{u,v} \in \mathbb{R}_+]_{u,v \in V}$. We assume there is an edge $\{u, v\}$ if and only if the corresponding weight is positive i.e. $\operatorname{support}(W) = E$. The **degree of a node** $u \in [n]$, d^G_u, is defined as $d^G_u \stackrel{\text{def}}{=} \sum_v w^G_{u,v}$, with $d^G_{\min} \stackrel{\text{def}}{=} \min_u d^G_u, d^G_{\max} \stackrel{\text{def}}{=} \max_u d^G_u$ being the minimum and maximum degrees respectively.*

For any two subsets $A, B \subseteq V$, the weight of edges between A and B, $w^G(A, B)$ is defined as:

$$w^G(A, B) = \sum_{u \in A, v \in B} w^G_{u,v}.$$

When dealing with unweighted graphs we will use $G = (V, E)$ with edge weights being implicitly defined as $w^G_{u,v} = \begin{cases} 1 & \text{if } \{u, v\} \in E, \\ 0 & \text{else.} \end{cases}$ When there is no room for ambiguity, we will drop the superscript G and use $d_u, d_{\min}, d_{\max}, w_{u,v}$ instead.

Most of the problems we study revolve around the relaxation of "disconnectedness" which involve measuring how close a given partitioning is to being disconnected measured by its cut cost, formalized below:

Definition 2.49 (Cut Cost)**.** *Given $G = (V, E, W)$ and a k-way partitioning of V, $(U_1, \ldots, U_k)$ we define cut cost of $(U_1, \ldots, U_k)$ as the total weights of edges crossing different subsets:*

$$\mathrm{cost}^G(U_1, \ldots, U_k) \stackrel{\text{def}}{=} \sum_{u \in U_i, v \notin U_i} w^G_{u,v} = \frac{1}{2} \sum_i w^G(U_i, V \setminus U_i).$$

A convenient way to algebrize cut costs is to introduce Laplacian matrix of a graph.

Definition 2.50 (Node-Edge Incidence and Laplacian Matrices)**.** *Let $B = [B_{\{a,b\},c}] \in \mathbb{R}^{\binom{V}{2}, V}$ be the node-edge incidence matrix of a complete graph where*

$$B_{\{a,b\},c} = \begin{cases} +1 & \text{if } c = a \text{ with } a < b\,, \\ -1 & \text{else.} \end{cases}$$

Given graph G, we define the diagonal matrix of edge weights, $W_G \in \mathbb{R}^{\binom{V}{2}, \binom{V}{2}}$ as

$$(W_G)_{\{a,b\},\{c,d\}} = \begin{cases} w^G_{a,b} & \text{if } \{a, b\} = \{c, d\}, \\ 0 & \text{else.} \end{cases}$$

Finally the Laplacian matrix of graph G, $L_G \in \mathbb{S}^V_+$ is defined as $L_G \stackrel{\text{def}}{=} B^T W_G B$ which is equal to

$$(L_G)_{u,v} \stackrel{\text{def}}{=} \begin{cases} \sum_{a \in V \setminus \{u\}} w^G_{u,a} & \text{if } u = v, \\ -w^G_{u,v} & \text{else.} \end{cases}$$

Definition 2.51 (Degree and Adjacency Matrix)**.** *Given $G = (V, E, W)$, we use $D_G \in \mathbb{S}^V$ and $A_G \in \mathbb{S}^V$ to denote G's diagonal matrix of degrees and adjacency matrix:*

$$(D_G)_{u,v} = \begin{cases} d^G_u & \text{if } u = v, \\ 0 & \text{else.} \end{cases}, \quad (A_G)_{u,v} = w^G_{u,v}.$$

Observe that $L_G = D_G - A_G$.

Claim 2.52. *For any subset $U \subseteq V$ whose indicator vector is given by $\mathbf{x} \in \{0,1\}^V$, the cut cost of partitioning $(U, V \setminus U)$ is equal to the following:*

$$\mathrm{cost}^G(U, V \setminus U) = \mathbf{x}^T L_G \mathbf{x}.$$

Similarly, given a k-way partitioning of V with indicator vector $\mathbf{x} \in \{0,1\}^{V \times [k]}$ so that $\mathbf{x}_{(V,i)} \in \{0,1\}^V$ over all $i \in [k]$, we can express its cut cost on graph G as:

$$\frac{1}{2} \sum_i \mathbf{x}_{(V,i)}^T L_G \mathbf{x}_{(V,i)}.$$

Notation 2.53 (Clique graph)**.** *Let K be the n-clique where for any $u \neq v$, $w^K_{u,v} = \frac{1}{n}$, and $w^K_{u,u} = 0$ so that its Laplacian matrix L_K satisfies $L_K = I_{V,V} - \frac{1}{n}\mathbb{1}_{V,V} = I - \frac{1}{n}\mathbb{1}\mathbb{1}^T$.*

Definition 2.54 (Generalized Eigenvalues for Graphs)**.** *Given two graphs G and H, for any positive integer j, we define $\lambda_j(\mathcal{G})$ and $\lambda_j(G;H)$ as the j^{th} smallest normalized eigenvalue of L_G and j^{th} smallest generalized eigenvalue of L_G and L_H, i.e.*

$$\lambda_j(\mathcal{G}) \stackrel{\text{def}}{=} \lambda_j(\mathcal{L}_\mathcal{G}),\ \lambda_j(G;H) \stackrel{\text{def}}{=} \lambda_j(L_G; L_H).$$

Observation 2.55. *For any pair of graphs G and H, the following hold:*

1. $L_G \mathbb{1} = L_H \mathbb{1} = 0$;
2. $z_1(\mathcal{G}) = D_G^{1/2}\mathbb{1}$;
3. *For any matrix $X = [X_u]_{u \in V}$,* $\mathrm{Tr}(X^T X L_G) = \sum_{u<v} w^G_{u,v} \|X_u - X_v\|^2$;
4. $L_G, L_H \succeq 0$;
5. $0 = \lambda_1(G;H) \leq \lambda_2(G;H) \leq \ldots \leq \lambda_n(G;H)$;
6. $\sum_i \lambda_i(\mathcal{G}) = |V| = n$.

2.8 Some Probabilistic Inequalities

We will review some basic probabilistic inequalities.

Theorem 2.56 (Markov Inequality)**.** *Given a random variable Y over non-negative numbers, $\mathbb{R}_+$ the following holds:*

$$\mathrm{Prob}\left[Y \geq \frac{1}{\varepsilon}\mathbb{E}\big[Y\big]\right] \leq \varepsilon.$$

Theorem 2.57 (Hoeffding Bound). *If X is a distribution over $\{0,1\}^A$ with each coordinate being **independent**, then for any vector $a \in \mathbb{R}^A$ and positive real $\varepsilon > 0$ we have*

$$\mathrm{Prob}_{x\sim X}\left[\left|\langle x,a\rangle - \mathbb{E}_{x'\sim X}\left[\langle x',a\rangle\right]\right| \geq O\left(\sqrt{\log 1/\varepsilon}\|a\|_2\right)\right] \leq \varepsilon,$$

where $\mu \stackrel{\text{def}}{=} \mathbb{E}_{x'\sim X}\left[\langle x',a\rangle\right]$.

Theorem 2.58 (Chernoff Bound). *If X is a distribution over $\{0,1\}^A$ with each coordinate being **independent**, then for any vector $a \in \mathbb{R}^A_+$ with $\|a\|_\infty \leq \frac{\mu}{\log(1/\varepsilon)}$, we have:*

$$\mathrm{Prob}_{x\sim X}\left[\left|\langle x,a\rangle - \mathbb{E}_{x'\sim X}\left[\langle x',a\rangle\right]\right| \geq O\left(\sqrt{\|a\|_\infty \mu \log 1/\varepsilon}\right)\right] \leq \varepsilon.$$

Chapter 3

Moment Based SDP Relaxations

In this chapter, we will formally introduce a class of semi-definite programming (SDP) based relaxations for $0/1$ problems, based on **pseudo-moments**. This class contains Lasserre Hierachy for which we presented a simpler derivation back in Section 1.4. We start with some algebraic background.

3.1 Algebraic Background

Throughout the whole chapter, we always denote the set of variables with V, $\mathbf{X} = [\mathbf{X}_u]_{u \in V}$.

Definition 3.1 (Real Polynomials)**.** *Let $\mathbb{R}[\mathbf{X}]$ be the set of polynomials with real coefficients over **variables** $\mathbf{X} = [\mathbf{X}_u]_{u \in V}$ where $\mathsf{f} \in \mathbb{R}[\mathbf{X}]$ if it can be expressed as*

$$\mathsf{f} = \sum_{\alpha: V \to \mathbb{Z}_+} \mathsf{f}_\alpha \underbrace{\prod_{u \in V} \mathbf{X}_u^{\alpha(u)}}_{\stackrel{\text{def}}{=} \mathbf{X}^{(\alpha)}},$$

for some $[\mathsf{f}_\alpha \in \mathbb{R}]_{\alpha: V \to \mathbb{Z}_+}$ such that at most finitely many f_α are non-zero. For any $S \subseteq V$, we will use $\mathbf{X}^{(S)} \stackrel{\text{def}}{=} \mathbf{X}^{(\mathbb{1}_S)} = \prod_{u \in S} \mathbf{X}_u$ and $\mathsf{f}_S \stackrel{\text{def}}{=} \mathsf{f}_{\mathbb{1}_S}$. For small cardinality sets, we will directly use $\mathbf{X}^{(u,v)}$ instead of $\mathbf{X}^{(\{u,v\})}$.

Given $\mathbf{x} \in \mathbb{R}^V$, we use $\mathsf{f}(\mathbf{x})$ to denote evaluation of f at $\mathbf{x} \in \mathbb{R}^V$:

$$\mathsf{f}(\mathbf{x}) = \sum_{\alpha: V \to \mathbb{Z}_+} \mathsf{f}_\alpha \prod_{u \in V} \mathbf{x}_u^{\alpha(u)}.$$

Notation 3.2 (Degree)**.** *We define* ***degree*** *of* f *as*

$$\operatorname{degr}(\mathsf{f}) \stackrel{\text{def}}{=} \max_{\substack{\alpha: V \to \mathbb{Z}_+ \\ \mathsf{f}_\alpha \neq 0}} \sum_{u \in V} \alpha_u.$$

Definition 3.3 (Multilinear Polynomials)**.** *Given* $\mathsf{f} \in \mathbb{R}[\mathbf{X}]$ *we call* f *a* ***multilinear polynomial*** *if* f *is linear in each variable, i.e., when no variable occurs to a power of* 2 *or higher. We use* $\mathbb{ML}[\mathbf{X}]$ *to denote the set of* ***multilinear polynomials****. In particular, for any* $\mathsf{f} \in \mathbb{R}[\mathbf{X}]$*:*

$$\mathsf{f} \in \mathbb{ML}[\mathbf{X}] \iff \mathsf{f} = \sum_S f_S \mathbf{X}^{(S)} \text{. for some } [f_S]_S \in \mathbb{R}^{2^V}.$$

Observation 3.4. *For any multilinear polynomial* $\mathsf{f} \in \mathbb{ML}[\mathbf{X}]$*,* $\operatorname{degr}(\mathsf{f}) = \max_{S:\mathsf{f}_S \neq 0} |S|$*.*

Notation 3.5 (Support of a Multilinear Polynomial)**.** *We define the* ***support of multilinear polynomial*** f *as:*

$$\operatorname{support}(\mathsf{f}) \stackrel{\text{def}}{=} \left\{ S \subseteq V \mid \mathsf{f}_S \neq 0 \right\} \subseteq 2^S.$$

Notation 3.6 (Multilinear Polynomials on Restricted Support)**.** *Given a set of variables* V*, family* $\mathcal{F}$ *over* V*, we use*

$$\mathbb{ML}_{\mathcal{F}}[\mathbf{X}] \stackrel{\text{def}}{=} \left\{ \mathsf{p} \in \mathbb{ML}[\mathbf{X}] \mid \operatorname{support}(\mathsf{p}) \subseteq \mathcal{F} \right\}$$

to denote the set of multilinear polynomials whose support is contained in $\mathcal{F}$*.*

Notation 3.7 (Coefficient Vectors for Multilinear Polynomials)**.** *Given family* $\mathcal{F}$ *over* V*, let* $\vec{\cdot} \colon \mathbb{ML}_{\mathcal{F}}[\mathbf{X}] \to \mathbb{R}^{\mathcal{F}}$ *be the* ***coefficient vector*** *operator. Given* $\mathsf{f} \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$*:*

$$\vec{\mathsf{f}} \stackrel{\text{def}}{=} [\mathsf{f}_S]_{S \in \mathcal{F}} \in \mathbb{R}^{\mathcal{F}}.$$

Lemma 3.8. *Given* $\mathsf{f} \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$*,*

$$\mathsf{f}(\mathbb{1}_S) = 0 \text{ for all } S \in \mathcal{F} \iff \mathsf{f} = 0.$$

Here $\mathbb{1}_S = [(\mathbb{1}_S)_u]_{u \in V} \in \{0,1\}^V$ *denotes the indicator vector for set* S*:*

$$(\mathbb{1}_S)_u = \begin{cases} 1 & \text{if } u \in S, \\ 0 & \text{else.} \end{cases}$$

Proof. ($\Rightarrow$) Consider the matrix $M = [\mathbf{X}^{(S)}(\mathbb{1}_T)]_{T\in\mathcal{F},S\in\mathcal{F}}$. For $\vec{\mathsf{f}} = [f_S]_{S\in\mathcal{F}}$ being the coefficient vector of f, we have $(M\vec{\mathsf{f}})_T = \mathsf{f}(\mathbb{1}_S)$. We will prove that determinant of M is zero, $|M| = 0$, which will imply $\mathrm{rank}(M) = |\mathcal{F}|$. In other words $M\vec{\mathsf{f}} = 0 \iff \vec{\mathsf{f}} = 0 \iff \mathsf{f} = 0$.

$$|M| = \sum_{\pi\in\mathrm{sym}(\mathcal{F})} (-1)^{\mathrm{sign}(\pi)} \prod_{S\in\mathcal{F}} M_{\pi(S),S} = \sum_{\pi\in\mathrm{sym}(\mathcal{F})} (-1)^{\mathrm{sign}(\pi)} \prod_{S\in\mathcal{F}} \mathbf{X}^{(S)}(\mathbb{1}_{\pi(S)})$$
$$= \sum_{\pi\in\mathrm{sym}(\mathcal{F})} (-1)^{\mathrm{sign}(\pi)} [\![S \subseteq \pi(S) \text{ for all } S \in \mathcal{F}]\!]$$

Whenever π is identity, this product is one. We will show that for any other π, this product is zero by contradiction; which means $|M| = 1$. Suppose π is non-identity where this product is non-zero. consider non-identity permutation π. Let $S^* \in \mathcal{F}$ be a largest set with $\pi(S^*) \neq S^*$: $S^* \leftarrow \mathrm{argmax}_{S\in\mathcal{F}:S\neq\pi(S)} |S|$. For such S^*, consider $T \stackrel{\text{def}}{=} \pi(S^*)$:

- $T \in \mathcal{F}$ since π is a permutation on $\mathcal{F}$.
- $|T| > |S^*|$ since $S^* \subseteq \pi(S^*)$ but $S^* \neq \pi(S^*) = T$.

This contradicts maximality of S^*.

($\Leftarrow$) Trivial.

□

3.1.1 Polynomial Ideals and Quotient Algebra

All our problems are based on finding some $\mathbf{x} \in \{0,1\}^V$ subject to polynomial constraints of the form $\mathsf{f}(\mathbf{x}) \geq 0$ or $\mathsf{g}(\mathbf{x}) = 0$. Therefore we need focus on polynomials restricted to $\{0,1\}^V$ and roots of such polynomials. In this section, we will review polynomial ideals and quotient algebra which will allow us to reason about such polynomials and solutions algebraically.

Polynomial Ideals. Let's start with the most basic question: When are two polynomials $\mathsf{f}, \mathsf{g} \in \mathbb{R}[\mathbf{X}]$ equal when restricted to $\{0,1\}^V$? In $\mathbb{R}^V$, checking when two polynomials are equal is easy – we can simply compare each coefficient. But this is not true anymore with the simplest example being $\mathbf{X}_u^2 \equiv \mathbf{X}_u$ for any $u \in V$. One way to check such equivalence is to find some other polynomial $\mathsf{h} \in \mathbb{R}[\mathbf{X}]$ such that

$$\mathsf{h}(\{0,1\}^V) = 0 \text{ and } \mathsf{f} = \mathsf{g} + \mathsf{h}.$$

Let $\mathcal{I}(\{0,1\}^V)$ be the set of such polynomials:

$$\mathcal{I}(\{0,1\}^V) \stackrel{\text{def}}{=} \left\{ \mathsf{h} \in \mathbb{R}[\mathbf{X}] \mid \mathsf{h}(\{0,1\}^V) = 0 \right\}.$$

It turns out $\mathcal{I}$ can be expressed in terms of "linear subspace" over polynomials (see Lemma 3.15): $\mathsf{h} \in \mathcal{I}$ iff there exists $\mathsf{h}^{(u)} \in \mathbb{R}[\mathbf{X}]$ for each $u \in V$ such that

$$\mathsf{h} = \sum_u \mathsf{h}^{(u)} \cdot (\mathbf{X}_u - \mathbf{X}_u^2).$$

This motivates the definition of ideals:

Definition 3.9 (Ideals)**.** *A subset $\mathcal{I} \subseteq \mathbb{R}[\mathbf{X}]$ is an **ideal** if it satisfies the following:*

- $\mathsf{p} \in \mathcal{I} \implies \mathsf{f} \cdot \mathsf{p} \in \mathcal{I}$ *for any* $\mathsf{f} \in \mathbb{R}[\mathbf{X}]$ *(note this implies $0 \in \mathcal{I}$,)*
- $\mathsf{p}, \mathsf{q} \in \mathcal{I} \implies \mathsf{p} + \mathsf{q} \in \mathcal{I}$.

Given an ideal as sets of polynomials, we can define the "dual" set in terms of their vanishing points:

Definition 3.10 (Varieties)**.** *Given a polynomial ideal $\mathcal{I}$, we define its **real variety** as the set of real solutions to $\mathsf{p}(\mathbf{x}) = 0$ for all $\mathsf{p} \in \mathcal{I}$:*

$$\text{Variety}(\mathcal{I}) \stackrel{\text{def}}{=} \left\{ \mathbf{x} \in \mathbb{R}^V \mid \mathsf{p}(\mathbf{x}) = 0 \text{ for all } \mathsf{p} \in \mathcal{I} \right\}.$$

Definition 3.11 (Two Ideal Constructions)**.** *The **ideal generated** by a finite set of polynomials $\{\mathsf{p}_1, \ldots, \mathsf{p}_m\} \subset \mathbb{R}[\mathbf{X}]$ is defined as set of all polynomial combinations of p_i's:*

$$\left(\mathsf{p}_1, \ldots, \mathsf{p}_m\right) \stackrel{\text{def}}{=} \left\{ \sum_i \mathsf{p}_i \mathsf{h}_i \mid \mathsf{h}_i \in \mathbb{R}[\mathbf{X}] \text{ for all } i \right\}.$$

*Similarly we define **vanishing ideal** of a set $X \subseteq \mathbb{R}^V$ as:*

$$\mathcal{I}(X) \stackrel{\text{def}}{=} \{ \mathsf{p} \in \mathbb{R}[\mathbf{X}] \mid \mathsf{p}(\mathbf{x}) = 0 \text{ for all } \mathbf{x} \in X \}.$$

Notation 3.12 (Binary Ideal)**.** *Given a down family $\mathcal{F}$ over V, we will use $\mathcal{B}_V(\mathcal{F})$ to denote the binary ideal,*

$$\mathcal{B}_V(\mathcal{F}) \stackrel{\text{def}}{=} \Big(\left\{ \mathbf{X}_u^2 - \mathbf{X}_u \;\middle|\; u \in V \right\} \cup \left\{ \mathbf{X}^{(S)} \;\middle|\; S \notin \mathcal{F} \right\} \Big).$$

We use $\mathcal{B}_V$ as a shorthand for $\mathcal{B}_V \stackrel{\text{def}}{=} \mathcal{B}_V(2^V)$. It is easy to see that $\text{Variety}[\mathcal{B}_V(\mathcal{F})] \subseteq \{0,1\}^V = \text{Variety}(\mathcal{B}_V)$.

Definition 3.13 (Quotient Algebra)**.** *Given an ideal $\mathcal{I}$, we use $\mathbb{R}[\mathbf{X}]/\mathcal{I}$ to denote its* ***quotient algebra*** *under the equivalence relation:*

$$\mathsf{f} \equiv \mathsf{g} \pmod{\mathcal{I}} \text{ iff } \mathsf{f} = \mathsf{g} + \mathsf{h} \text{ for some } \mathsf{h} \in \mathcal{I}.$$

For any $\mathsf{f} \in \mathbb{R}[\mathbf{X}]$, we use $\big[\mathsf{f}\big]$ to denote a ***representative*** *chosen in some canonical way from its equivalence class so that $\big[\mathsf{f}\big] \equiv \mathsf{f}$ and $\big[\mathsf{f}\big] = \mathsf{f}$.*

Notation 3.14 (Multilinear Representatives)**.** *We define* ***representatives*** *for $\mathbb{R}[\mathbf{X}]/\mathcal{B}_V(\mathcal{F})$ with the following linear map, $[\cdot] : \mathbb{R}[\mathbf{X}] \to \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$. For any $\mathsf{p} \in \mathbb{R}[\mathbf{X}]$,*

$$\big[\mathsf{p}\big] \stackrel{\text{def}}{=} \sum_{S \in \mathcal{F}} \Bigg(\sum_{\alpha : S \to \mathbb{N}} \mathsf{p}_\alpha \Bigg) \mathbf{X}^{(S)} \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}],$$

so that $\big[\big[\mathsf{p}\big]\big] = \big[\mathsf{p}\big]$. We will prove $[\mathsf{p}] \equiv \mathsf{p} \pmod{\mathcal{B}_V(\mathcal{F})}$ in the next lemma.

Lemma 3.15. *The following hold for any $\mathsf{f}, \mathsf{g} \in \mathbb{R}[\mathbf{X}]$.*

(i) For any $u \in V$ if $\{u\} \in \mathcal{F}$ then for any $k \in \mathbb{N}$, $\mathbf{X}_u^k \equiv \mathbf{X}_u \pmod{\mathcal{B}_V(\mathcal{F})}$.

(ii) For any $S \notin \mathcal{F}$, $\mathbf{X}^{(S)} \equiv 0 \pmod{\mathcal{B}_V(\mathcal{F})}$.

(iii) For any $\alpha : V \to \mathbb{Z}_+$, $\mathbf{X}^\alpha \equiv \mathbf{X}^{\text{support}(\alpha)}$.

(iv) $\mathsf{f} \equiv \big[\mathsf{f}\big] \pmod{\mathcal{B}_V(\mathcal{F})}$.

(v) If $\mathsf{f}, \mathsf{g} \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$ and $\mathsf{f} \equiv \mathsf{g} \pmod{\mathcal{B}_V(\mathcal{F})}$ then $\mathsf{f} = \mathsf{g}$.

(vi) $\mathsf{f}(\mathbf{x}) = \mathsf{g}(\mathbf{x})$ for all $\mathbf{x} \in \{0,1\}^V \iff \mathsf{f} \equiv \mathsf{g} \pmod{\mathcal{B}_V(\mathcal{F})}$.

Therefore $\big[\mathsf{f}\big]$ is the **unique** *multilinear polynomial with $\mathsf{f} \equiv \big[\mathsf{f}\big]$[1] and*

$$\mathcal{I}(\{0,1\}^V) = \mathcal{B}_V(\mathcal{F}).\ ^{[2]}$$

Proof. (i) Follows from $\mathbf{X}_a = \mathbf{X}_a^k + \left(\mathbf{X}_a^{k-2} + \mathbf{X}_a^{k-3} + \ldots + 1\right)(\mathbf{X}_a - \mathbf{X}_a^2)$.

(ii) By induction on $|\{u \mid \alpha(u) \geq 2\}|$ and using the previous item.

(iii) By induction on $|\{\alpha : V \to \mathbb{N} \mid \alpha(u) \geq 2 \text{ for some } u \text{ and } \mathsf{f}_\alpha > 0\}|$.

[1] Therefore $\mathbb{R}[\mathbf{X}]/\mathcal{B}_V(\mathcal{F})$ is isomorphic to $\mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$, $\mathbb{R}[\mathbf{X}]/\mathcal{B}_V(\mathcal{F}) \cong \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$.
[2] $\mathcal{B}_V(\mathcal{F})$ is radical ideal.

(iv) Then $\mathsf{f}-\mathsf{g} \in \mathbb{ML}[\mathbf{X}]$ and $\mathsf{f}-\mathsf{g} \equiv 0 \pmod{\mathcal{B}_V(\mathcal{F})}$ which implies $(\mathsf{f}-\mathsf{g})(\mathbf{x}) = 0$ for all $\mathbf{x} \in \{0,1\}^V$. By Lemma 3.8, $\mathsf{f}-\mathsf{g} = 0$.

(v) ($\Leftarrow$) Trivial.

($\Rightarrow$) $(\mathsf{f}-\mathsf{g})(\mathbf{x}) = 0$ for all $\mathbf{x} \in \{0,1\}^V$. Then $\left[\mathsf{f}-\mathsf{g}\right] \equiv 0 \pmod{\mathcal{B}_V(\mathcal{F})}$ and unique by previous items. Since $0 \in \mathbb{ML}[\mathbf{X}]$, this means $\left[\mathsf{f}\right] = \left[\mathsf{g}\right] \implies \mathsf{f} \equiv \mathsf{g} \pmod{\mathcal{B}_V(\mathcal{F})}$. □

Above claim says that for the quotient algebra of $\mathcal{B}_V(\mathcal{F})$, we can choose our representatives as multilinear polynomials, which turns out to be unique.

Example 3.16. *We can algebraically express the fact that any partitioning of an odd cycle graph must leave at least one edge uncut as follows.*

Let $V = \{1, 2, \ldots, 2k+1\}$ for some positive integer k. Then

$$(\mathbf{X}_1 - \mathbf{X}_2)(\mathbf{X}_2 - \mathbf{X}_3)\ldots(\mathbf{X}_{2k+1} - \mathbf{X}_1) \equiv 0 \pmod{\mathcal{B}_V}.$$

Proof. Assume there is a non-root $\mathbf{x} \in \{0,1\}^V$. Then:

$$\begin{aligned}
\mathbf{x}_1 \neq \mathbf{x}_2 &\implies \mathbf{x}_1 + \mathbf{x}_2 = 1 \\
\mathbf{x}_2 \neq \mathbf{x}_3 &\implies \mathbf{x}_2 + \mathbf{x}_3 = 1 \\
&\vdots \qquad\qquad \vdots \\
\mathbf{x}_{2k+1} \neq \mathbf{x}_1 &\implies \mathbf{x}_{2k+1} + \mathbf{x}_1 = 1 \\
& \quad + \\
& \quad \overline{\qquad\qquad\qquad\qquad} \\
& \quad 2(\textstyle\sum_i \mathbf{x}_i) = 2k+1.
\end{aligned}$$

A contradiction. Using Lemma 3.15, we conclude that $(\mathbf{X}_1-\mathbf{X}_2)(\mathbf{X}_2-\mathbf{X}_3)\ldots(\mathbf{X}_{2k+1}-\mathbf{X}_1) \equiv 0$. □

Claim 3.17. *Given polynomials $\mathsf{f}, \mathsf{g} \in \mathbb{R}[\mathbf{X}]$ and down family $\mathcal{F}$ over V:*

$$\left[\mathsf{f}\cdot\mathsf{g}\right] = \sum_{S\in\mathcal{F}} \left(\sum_{A,B:A\cup B=S} \left[\mathsf{f}\right]_A \cdot \left[\mathsf{g}\right]_B \right) \mathbf{X}^{(S)}.$$

3.1.2 Pseudo-Moments

Definition 3.18 (Moment Matrix and Moment Sequence)**.** *Given a set of variables V, let **moment matrix** of $x \in \mathbb{R}^{2^V}$, $\mathbf{M}_V : \mathbb{R}^{2^V} \to \mathbb{S}^{2^V}$ be the following linear map:*

$$\mathbf{M}_V(x) = [x_{A\cup B}]_{A,B\in\subseteq V}.$$

*We call $x \in \mathbb{R}^{2^V}$ a **moment sequence** iff $\mathbf{M}_V(x) \succeq 0$.*

For subsets of small cardinality such as $\{a, b\}$, we will use $x_{a,b}$ instead of $x_{\{a,b\}}$.

As mentioned in Section 1.4, our relaxations are based on enforcing positive semi-definiteness condition only on certain principal minors of $\mathbf{M}_V(x)$ whose rows and columns correspond to set families over V.

Notation 3.19 (Principal Minors of Moment Matrix)**.** *Given down family $\mathcal{F}$ over V, we define the linear map $\mathbf{M}_\mathcal{F} : \mathbb{R}^{\mathcal{F} \uplus \mathcal{F}} \to \mathbb{S}^\mathcal{F}$ as the function corresponding to the principal minor $\mathcal{F}$ of $\mathbf{M}_V$ so that:*

$$\mathbf{M}_\mathcal{F}(x) = [x_{S \cup T}]_{S \in \mathcal{F}, T \in \mathcal{F}}.$$

Observe that $\mathbf{M}_\mathcal{F}(x)$ is only a function of $x_{\mathcal{F} \uplus \mathcal{F}}$.

Example 3.20. *Suppose $V = \{a, b\}$ and $\mathcal{F} = \{\emptyset, \{a\}, \{b\}\}$. Then*

$$\mathbf{M}_\mathcal{F}(x) = \begin{bmatrix} x_\emptyset & x_a & x_b \\ x_a & x_a & x_{a,b} \\ x_{a,b} & x_{a,b} & x_b \end{bmatrix}.$$

Now we can formally define pseudo-moments:

Definition 3.21 (Pseudo-Moments)**.** *Given a down family $\mathcal{F}$ over V, we call $x \in \mathbb{R}^{\mathcal{F} \uplus \mathcal{F}}$ a **pseudo-moment** sequence if*

$$\mathbf{M}_\mathcal{F}(x) \succeq 0.$$

We denote the set of such pseudo-moments by $\Sigma^*_\mathcal{F}$:

$$\Sigma^*_\mathcal{F} \overset{\text{def}}{=} [3] \left\{ x \in \mathbb{R}^{\mathcal{F} \uplus \mathcal{F}} \middle| \, \mathbf{M}_\mathcal{F}(x) \succeq 0 \right\}.$$

Claim 3.22. *Given a family $\mathcal{F}$ over V and pseudo-moments $x \in \Sigma^*_\mathcal{F}$, for any $A, B \in \mathcal{F}$:*

$$x_A x_B \geq x^2_{A \cup B}.$$

In particular, if $x_\emptyset = 0$, then $x = 0$.

Proof. Since $(\mathbf{M}_\mathcal{F}(x))_{\{A,B\},\{A,B\}} = \begin{bmatrix} x_A & x_{A \cup B} \\ x_{A \cup B} & x_B \end{bmatrix} \succeq 0$:

$$0 \leq \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} x_A & x_{A \cup B} \\ x_{A \cup B} & x_B \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = x_A x_B - x^2_{A \cup B}. \qquad \square$$

[3] We will prove in Theorem 3.34 that $\Sigma^*_\mathcal{F}$ indeed corresponds to the dual cone for a natural family of polynomials.

In some sense, a pseudo-moment sequence will corresponding to the expectation of $\mathbf{X}^{(S)}(\mathbf{x})$ for $\mathbf{x}$ drawn from an unknown distribution. We will make this connection formal later in Theorem 3.36.

Definition 3.23 (Pseudo-Moment Vectors)**.** *Given down family $\mathcal{F}$ over V, we call $\vec{X} = [\vec{x}_S \in \mathbb{R}^{\Upsilon}]_{S\in\mathcal{F}} \in \mathbb{R}^{\Upsilon,\mathcal{F}}$ **pseudo-moment vectors** if*

$$\langle \vec{x}_A, \vec{x}_B\rangle = \langle \vec{x}_S, \vec{x}_T\rangle \text{ for all } A, B, S, T \in \mathcal{F} \text{ with } A \cup B = S \cup T.$$

*We say $\vec{X}$ **represents** $x \in \mathbb{R}^{\mathcal{F}\uplus\mathcal{F}}$ if $x_{S\cup T} = \langle \vec{x}_S, \vec{x}_T\rangle$ for all $S, T \in \mathcal{F}$ (i.e. $\vec{X}^T\vec{X} = \mathbf{M}_{\mathcal{F}}(x)$).*

Up to rotations, there is a one-to-one correspondence between the set of pseudo-moment vectors on $\mathcal{F}$ and pseudo-moment sequences:

Proposition 3.24. *$x \in \Sigma^*_{\mathcal{F}}$ iff there exists representing moment vectors $\vec{X} = [\vec{x}_S]$.*

Proof. Immediate from Gram decomposition (Theorem 2.23). □

Remark 3.25. *Proposition 3.24 says we can prove properties on moment sequences by proving them on moment vectors and vice versa. One such example is the proof of eq. (3.4) from Theorem 3.31*

Definition 3.26 (Pseudo-Evaluation)**.** *Let $\langle\!\langle \cdot, \cdot \rangle\!\rangle : \mathbb{R}[\mathbf{X}] \times \Sigma^*_{\mathcal{F}} \to \mathbb{R}$ be the following bilinear form. For any $\mathsf{p} \in \mathbb{R}[\mathbf{X}]$ and $x \in \Sigma^*_{\mathcal{F}}$:*

$$\langle\!\langle \mathsf{p}, x\rangle\!\rangle \stackrel{\text{def}}{=} \sum_{S\in\mathcal{F}} \left[\mathsf{p}\right]_S x_S.$$

*We refer to $\langle\!\langle \mathsf{p}, x\rangle\!\rangle$ as **pseudo-evaluation** of f over x.*

Remark 3.27. *Suppose $x \in \mathbb{R}^{\mathcal{F}}$ corresponds to the moments of some $\mathbf{x} \in \{0,1\}^V$. Then for any $\mathsf{f} \in \mathbb{R}[\mathbf{X}]$ with $\left[\mathsf{f}\right] \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$, $\langle\!\langle \mathsf{f}, x\rangle\!\rangle = \mathsf{f}(\mathbf{x})$.*

It is trivial to verify that $\langle\!\langle \cdot, \cdot \rangle\!\rangle$ is a bilinear form. Then we can ask what adjoints of some operators are:

Definition 3.28 (Adjoint of Moment Function)**.** *Given down family $\mathcal{F}$ over V, let $\widehat{\cdot} : \mathbb{R}^{\mathcal{F},\mathcal{F}} \to \mathbb{R}[\mathbf{X}]$ be the following linear map. For any $Q \in \mathbb{R}^{\mathcal{F},\mathcal{F}}$:*

$$\widehat{Q} \stackrel{\text{def}}{=} \sum_{A\in\mathcal{F}, B\in\mathcal{F}} Q_{A,B}\mathbf{X}^A\mathbf{X}^B.$$

Claim 3.29. $\mathbb{ML}_{\mathcal{F}\uplus\mathcal{F}}[\mathbf{X}] = \left[\widehat{\mathbb{R}^{\mathcal{F},\mathcal{F}}}\right]$.

We also define the adjoint of polynomial multiplication operator as follows.

Definition 3.30 (Adjoint of Polynomial Multiplication)**.** *Given pair of down families $\mathcal{F}, \mathcal{G}$ over V let $* : \mathbb{ML}_{\mathcal{G}}[\mathbf{X}] \times \mathbb{R}^{\mathcal{F}\uplus\mathcal{G}} \to \mathbb{R}^{\mathcal{F}}$ be the following linear map. For any $\mathsf{f} \in \mathbb{ML}_{\mathcal{G}}[\mathbf{X}]$ and $x \in \mathbb{R}^{\mathcal{F}\uplus\mathcal{G}}$:*

$$\mathsf{f} * x \stackrel{\text{def}}{=} \left[\sum_{T\in\mathcal{G}} \mathsf{f}_T x_{S\cup T}\right]_{S\in\mathcal{F}}.$$

Having defined $\widehat{\cdot}$ and $*$, we will state simple identities in Theorem 3.31.

Theorem 3.31. *1. For any $Q \in \mathbb{R}^{\mathcal{F},\mathcal{F}}$ we have $\left[\widehat{Q}\right] \in \mathbb{ML}_{\mathcal{F}\uplus\mathcal{F}}[\mathbf{X}]$ and for any $x \in \mathbb{R}^{\mathcal{F}\uplus\mathcal{F}}$,*

$$\operatorname{Tr}\left[Q \cdot \mathbf{M}_{\mathcal{F}}(x)\right] = \langle\!\langle \widehat{Q}, x\rangle\!\rangle. \tag{3.1}$$

2. For any $\mathsf{f}, \mathsf{g} \in \mathbb{ML}[\mathbf{X}]$, $\widehat{\cdot}$ maps outer product of coefficient vectors $[\mathsf{f}_A]$ and $[\mathsf{g}_B]$ to their product:

$$\widehat{[\mathsf{f}_A][\mathsf{g}_B]^T} = \mathsf{f} \cdot \mathsf{g}. \tag{3.2}$$

3. For any $\mathsf{f}, \mathsf{g} \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$ and $x \in \mathbb{R}^{\mathcal{F}\uplus\mathcal{F}}$:

$$\langle\!\langle \mathsf{f} \cdot \mathsf{g}, x\rangle\!\rangle = [\mathsf{f}_A]^T \mathbf{M}_{\mathcal{F}}(x)[\mathsf{g}_B] = \langle\!\langle \mathsf{f}, \mathsf{g} * x\rangle\!\rangle. \tag{3.3}$$

4.

$$\langle\!\langle \mathsf{p}^2, x\rangle\!\rangle = 0 \iff \mathsf{p} * x = 0 \iff \sum_S \vec{x}_S \left[\mathsf{p}\right]_S = 0. \tag{3.4}$$

Proof. 1. Note that $\left[\widehat{Q}\right] = \sum_{A\in\mathcal{F},B\in\mathcal{F}} Q_{A,B}\mathbf{X}^{A\cup B} \in \mathbb{ML}_{\mathcal{F}\uplus\mathcal{F}}[\mathbf{X}]$. Second claim follows from $\operatorname{Tr}\left[Q \cdot \mathbf{M}_{\mathcal{F}}(x)\right] = \sum_{A\in\mathcal{F},B\in\mathcal{F}} x_{A\cup B}Q_{A,B} = \sum_{S\in\mathcal{F}\uplus\mathcal{F}} x_S \left(\sum_{\substack{A\in\mathcal{F},B\in\mathcal{F}\\ A\cup B=S}} Q_{A,B}\right) = \langle\!\langle \widehat{Q}, x\rangle\!\rangle$.

2. For any S, $\left(\widehat{[\mathsf{f}_A]\cdot[\mathsf{g}_B]^T}\right)_S = \sum_{A\in\mathcal{F},B\in\mathcal{F}:A\cup B=S} \mathsf{f}_A\mathsf{g}_B = (\mathsf{f}\cdot\mathsf{g})_S$.

3. $\langle\!\langle \mathsf{f}\cdot\mathsf{g}, x\rangle\!\rangle = \langle\!\langle \widehat{[\mathsf{f}_A]\cdot[\mathsf{g}_B]^T}, x\rangle\!\rangle = \operatorname{Tr}\left[[\mathsf{f}_A]\cdot[\mathsf{g}_B]^T \mathbf{M}_{\mathcal{F}}(x)\right] = [\mathsf{f}_A]^T \mathbf{M}_{\mathcal{F}}(x)[\mathsf{g}_B]$. Here we used item 2 followed by item 1. This is equal to

$$= \sum_{A,B\in\mathcal{F}} \mathsf{f}_A\mathsf{g}_B x_{A\cup B} = \sum_{A\in\mathcal{F}} \mathsf{f}_A \left(\mathsf{g} * x\right)_A = \langle\!\langle \mathsf{f}, \mathsf{g} * x\rangle\!\rangle. \qquad \square$$

Proof of eq. (3.4). First we prove $\langle\!\langle \mathsf{p}^2, x \rangle\!\rangle = 0 \implies \sum_S \vec{x}_S \big[\mathsf{p}\big]_S = 0$:

$$\langle\!\langle \mathsf{p}^2, x \rangle\!\rangle = \Big\| \sum_S \big[\mathsf{p}\big]_S \vec{x}_S \Big\|^2 = 0 \implies \sum_S \big[\mathsf{p}\big]_S \vec{x}_S = 0.$$

Next we prove that $\sum_S \vec{x}_S \big[\mathsf{p}\big]_S = 0 \implies \mathsf{p} * x = 0$. For any $S \in \mathcal{F}$:

$$(\mathsf{p} * x)_S = \sum_T \big[\mathsf{p}\big]_T x_{S \cup T} = \sum_T \big[\mathsf{p}\big]_T \langle\!\langle \vec{x}_S, \vec{x}_T \rangle\!\rangle = \langle\!\langle \vec{x}_S, \underbrace{\sum_T \big[\mathsf{p}\big]_T \vec{x}_T}_{=0} \rangle\!\rangle = 0.$$

Finally $\mathsf{p} * x = 0 \implies \langle\!\langle \mathsf{p}^2, x \rangle\!\rangle$ follows easily from $\langle\!\langle \mathsf{p}, \mathsf{p} * x \rangle\!\rangle = \langle\!\langle \mathsf{p}^2, x \rangle\!\rangle = 0$ where we used eq. (3.3). □

3.1.3 Sum of Squares Ordering

From a dual perspective, our problems can be expressed as finding non-negative matrices maximizing various functions. Our relaxation for primal corresponds to finding sum of squares type polynomials on support $\mathcal{F}$ instead.

Definition 3.32 (Sum of Squares)**.** *Given family $\mathcal{F}$ over V, we use $\Sigma_{\mathcal{F}} \subseteq \mathbb{ML}_{\mathcal{F} \uplus \mathcal{F}}[\mathbf{X}]$ to denote the set of multilinear polynomials on $\mathcal{F}$ equivalent to **sum of squares** of polynomials from $\mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$ under quotient algebra $\mathbb{R}[\mathbf{X}]/\mathcal{B}_V(\mathcal{F})$:*

$$\Sigma_{\mathcal{F}} \stackrel{\text{def}}{=} \left\{ \mathsf{f} \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}] \;\middle|\; \mathsf{f} \equiv \sum_i \mathsf{g}_i^2 \pmod{\mathcal{B}_V(\mathcal{F})} \text{ for some } \mathsf{g}_1, \ldots, \mathsf{g}_m \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}] \right\}.$$

*We call $\mathsf{f} \in \Sigma_{\mathcal{F}}$ an **SoS**. Since $\Sigma_{\mathcal{F}}$ is a convex cone, we can define a partial ordering:*

$$\mathsf{f} \geq_{\Sigma_{\mathcal{F}}} \mathsf{g} \iff \mathsf{f} - \mathsf{g} \in \Sigma_{\mathcal{F}}.$$

$\leq_{\Sigma_{\mathcal{F}}}$, $=_{\Sigma_{\mathcal{F}}}$, $>_{\Sigma_{\mathcal{F}}}$ and $<_{\Sigma_{\mathcal{F}}}$ are defined similarly.

Lemma 3.33. *The linear operator $\widehat{\cdot}$ as given in Definition 3.28 is a surjective map between PSD-matrices and SoS-polynomials:*

$$\mathsf{f} \in \Sigma_{\mathcal{F}} \text{ iff } \mathsf{f} \equiv \widehat{G} \pmod{\mathcal{B}_V(\mathcal{F})} \text{ for some } G \in \mathbb{S}_+^{\mathcal{F}}.$$

Proof. ($\Rightarrow$) Given such f let g_i's be such that $\sum_i \mathsf{g}_i^2 \equiv \mathsf{f}$. Define G_i as the following outer product:

$$G_i \stackrel{\text{def}}{=} [(\mathsf{g}_i)_A][(\mathsf{g}_i)_A]^T \in \mathbb{S}_+^{\mathcal{F}} \implies \widehat{G_i} = \mathsf{g}_i^2 \text{ by eq. (3.2)}.$$

For $G \leftarrow \sum_i G_i$ we have $G \in \mathbb{S}_+^{\mathcal{F}}$. By linearity, $\widehat{G} = \sum_i \widehat{G_i} = \sum_i \mathsf{g}_i{}^2$ which means $\mathsf{f} \equiv \widehat{G}$.

($\Leftarrow$) Given $G \in \mathbb{S}^{\mathcal{F}}_+$ consider its by eigen-decomposition (see Theorem 2.24), which says that exists vectors $\vec{g}_1, \ldots, \vec{g}_m \in \mathbb{R}^{\mathcal{F}}$ such that

$$G = \sum_i \vec{g}_i \vec{g}_i^T.$$

For each such $\vec{g}_i$, let $\mathsf{g}_i \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$ be the following:

$$\mathsf{g}_i \stackrel{\text{def}}{=} \sum_{S \in \mathcal{F}} (\vec{g}_i)_S \mathbf{X}^{(S)} \implies \widehat{\vec{g}_i \vec{g}_i^T} = \mathsf{g}_i^2.$$

In particular $\widehat{G} = \sum_i \mathsf{g}_i^2$ by eq. (3.2) which implies $\mathsf{f} \equiv \widehat{G} \equiv \sum_i \mathsf{g}_i^2$. □

Using Lemma 3.33, we can characterize pseudo-moment sequences purely in terms of SoS.

Theorem 3.34. *$x \in \Sigma^*_{\mathcal{F}}$ iff for all $\mathsf{f} \in \Sigma_{\mathcal{F}}$ we have $\langle\langle \mathsf{f}, x \rangle\rangle \geq 0$. In particular, $\Sigma^*_{\mathcal{F}}$ is the dual cone of $\Sigma_{\mathcal{F}}$ with respect to $\langle\langle \cdot, \cdot \rangle\rangle$.*

Proof. $x \in \Sigma^*_{\mathcal{F}}$ is equivalent to $\mathbf{M}_{\mathcal{F}}(x) \succeq 0$, which in turn is equivalent to

$$\forall Q \in \mathbb{S}^{\mathcal{F}}_+ : \operatorname{Tr}\left[Q \cdot \mathbf{M}_{\mathcal{F}}(x)\right] \geq 0$$

by Proposition 2.21. Using Lemma 3.33 we have $Q \in \mathbb{S}^{\mathcal{F}}_+ \iff \widehat{Q} \in \Sigma_{\mathcal{F}}$. By eq. (3.1):

$$\operatorname{Tr}\left[Q \cdot \mathbf{M}_{\mathcal{F}}(x)\right] = \langle\langle \widehat{Q}, x \rangle\rangle.$$

Hence $\operatorname{Tr}\left[Q \cdot \mathbf{M}_{\mathcal{F}}(x)\right] \geq 0$ iff $\langle\langle \widehat{Q}, x \rangle\rangle \geq 0$. □

The following appears in Chapter 11.

Corollary 3.35 (Restatement of Corollary 11.12)**.** *$\mathsf{f} \in \Sigma_{\mathcal{F}}$ iff $\langle\langle \mathsf{f}, x \rangle\rangle \geq 0$ for all $x \in \Sigma^*_{\mathcal{F}}$.*

3.2 Moment Based SDP Relaxations

We can finally prove the following, which implies enforcing PSD-ness on principal minors is a relaxation.

Theorem 3.36. *For any* $x \in \mathbb{R}^{2^V}$

$$\mathbf{M}_V(x) \succeq 0 \iff \forall S \subseteq V : \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathbf{X}^{(S)}(\mathbf{x})\right] = x_S \text{ for some distribution } \mathcal{D} \text{ on } \{0,1\}^V.$$

Furthermore, if $\mathbf{M}_V(x) \succeq 0$ *then for any* $\mathsf{p} \in \mathbb{R}^{2^V}$*:*

$$\mathbf{M}_V(\mathsf{p} * x) \succeq 0 \iff \mathrm{Prob}_{\mathbf{x}\sim\mathcal{D}}\left[\mathsf{p}(\mathbf{x}) < 0\right] = 0.$$

In this section, we will only prove the existence of such x given distribution $\mathcal{D}$ as it is sufficient for Corollary 3.37. We postpone the proof of harder direction till next section.

Proof of $\Leftarrow$ *(easy direction).* Given such $\mathcal{D}$, consider the following $\vec{X} = [\vec{x}_S]$ where each $\vec{x}_S \in \mathbb{R}^{\{0,1\}^V}$ is defined as:

$$\vec{x}_S \stackrel{\text{def}}{=} \left[\sqrt{\mathrm{Prob}_{\widetilde{y}\sim\mathcal{D}}\left[\mathbf{x} = \widetilde{y}\right]}\mathbf{X}^{(S)}(\mathbf{x})\right]_{\mathbf{x}\in\{0,1\}^V}.$$

It is trivial to check that this is indeed a collection of moment vectors. For the second property, given such p, consider $\mathbf{Y} = [\vec{y}_S]$ where each $\vec{y}_S \in \mathbb{R}^{\{0,1\}^V}$ is given by:

$$\vec{y}_S \stackrel{\text{def}}{=} \left[\sqrt{\mathrm{Prob}_{\widetilde{y}\sim\mathcal{D}}\left[\mathbf{x} = \widetilde{y}\right]\mathsf{p}(\mathbf{x})}\mathbf{X}^{(S)}(\mathbf{x})\right]_{\mathbf{x}\in\{0,1\}^V}.$$

Again we can check easily that these are moment vectors. We will now prove that $\mathsf{p} * [\|\vec{x}_A\|^2]_A = [\|\vec{y}_A\|^2]_A$: For any $A \subseteq V$:

$$(\mathsf{p} * [\|\vec{x}_S\|^2]_S)_A = \langle \vec{x}_A, \sum_B \mathsf{p}_B \vec{x}_B \rangle = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathbf{X}^{(A)}(\mathbf{x})\mathsf{p}(\mathbf{x})\right] = \|\vec{y}_A\|^2. \qquad \square$$

Proof of $\Rightarrow$ *(hard direction).* Given in Lemma 3.44. $\square$

Now we can formally state moment relaxations for eq. (1.1). Lasserre relaxation follows as a special case.

Corollary 3.37. *Given variable set V, $\mathsf{a} \in \mathbb{ML}_{V_{\leq d}}[\mathbf{X}]$, $B \subset \mathbb{ML}_{V_{\leq d}}[\mathbf{X}]$ and $C \subset \mathbb{ML}_{V_{\leq d}}[\mathbf{X}]$ consider the following $0/1$ problem (same with eq. (1.1)):*

$$\begin{array}{lll} \textit{Minimize} & \mathsf{a}(\mathbf{x}) & \\ \textit{subject to} & \mathsf{b}(\mathbf{x}) \geq 0 & \textit{for all } \mathsf{b} \in B, \\ & \mathsf{c}(\mathbf{x}) = 0 & \textit{for all } \mathsf{c} \in C, \\ & \mathbf{x} \in \{0,1\}^n. & \end{array} \tag{3.5}$$

For any family of subsets $\mathcal{F} \subseteq 2^V$, the following is a relaxation for eq. (3.5):

$$\begin{array}{lll} \textit{Minimize} & \langle\!\langle \mathsf{a}, x \rangle\!\rangle & \\ \textit{subject to} & \mathbf{M}_{\mathcal{F}}(\mathsf{b} * x) \succeq 0 & \textit{for all } \mathsf{b} \in \mathcal{B}, \\ & \langle\!\langle \mathsf{c}^2, x \rangle\!\rangle = 0 & \textit{for all } \mathsf{c} \in \mathcal{C}, \\ & \mathbf{M}_{\mathcal{F} \uplus V_{\leq d}}(x) \succeq 0, & \\ & x_\emptyset = 1 \textit{ and } x \in \mathbb{R}^{\mathcal{F} \uplus \mathcal{F} \uplus \binom{V}{\leq 2d}}. & \end{array} \tag{3.6}$$

If $\mathcal{F} = \binom{V}{\leq r}$, this corresponds to r rounds of Lasserre Hierachy *relaxation as given by Lasserre [2002].*

Proof of Corollary 3.37. Note that we can express each constraint $\mathsf{c}(\mathbf{x}) = 0$ as

$$\mathsf{c}(\mathbf{x}) \geq 0, \ -\mathsf{c}(\mathbf{x}) \geq 0.$$

Then by $\Leftarrow$ of Theorem 3.36, it is clear that the following is a relaxation:

$$\begin{array}{lll} \text{Minimize} & \langle\!\langle \mathsf{a}, x \rangle\!\rangle & \\ \text{subject to} & \mathbf{M}_{\mathcal{F}}(\mathsf{b} * x) \succeq 0 & \text{for all } \mathsf{b} \in \mathcal{B}, \\ & \mathbf{M}_{\mathcal{F}}(\mathsf{c} * x) \succeq 0 & \\ & \text{and } \mathbf{M}_{\mathcal{F}}((-\mathsf{c}) * x) \succeq 0 & \text{for all } \mathsf{c} \in \mathcal{C}, \\ & \mathbf{M}_{\mathcal{F} \uplus V_{\leq d}}(x) \succeq 0, & \\ & x_\emptyset = 1 \text{ and } x \in \mathbb{R}^{\mathcal{F} \uplus \mathcal{F} \uplus \binom{V}{\leq d}}. & \end{array} \tag{3.7}$$

But $\mathbf{M}(\mathsf{c} * x) \succeq 0, \ \mathbf{M}(-\mathsf{c} * x) \succeq 0$ iff $\mathsf{c} * x = 0$ which is equivalent to $\langle\!\langle \mathsf{c}^2, x \rangle\!\rangle = 0$ by eq. (3.4). □

3.3 Labeling Vectors

In the previous section, we said pseudo-moment vectors can be thought as random variables corresponding to indicator function of each monomial, $\mathbf{X}^{(S)}(\mathbf{x})$. In

this section, we will show how to construct vectors corresponding to indicator functions of a class of polynomials. The most powerful property of Lasserre Hierachy reveals itself here: Any such vector will behave consistently with respect to the quotient algebra induced by given pseudo-moment sequence.

Definition 3.38. *Given V, $\mathcal{F} \subseteq 2^V$ and $x \in \Sigma^*_{\mathcal{F}}$ with moment vectors $\vec{X} = [\vec{x}_S]$, we define the **vector for polynomial** $\mathsf{p} \in \mathbb{R}[\mathbf{X}]$ as:*

$$\vec{x}(\mathsf{p}) \stackrel{\text{def}}{=} \vec{X} \cdot \vec{[\mathsf{p}]} = \sum_{S \in \mathcal{F}} [\mathsf{p}]_S \vec{x}_S.$$

Using identities from Theorem 3.31, we can show that these vectors are consistent in the following way:

Corollary 3.39. *Given down family $\mathcal{F}$ over V and pseudo-moment sequence $x \in \Sigma^*_{\mathcal{F}}$ with vectors $\vec{X} = [\vec{x}_S]_{S \in \mathcal{F}}$, for any pair of polynomials $\mathsf{p}, \mathsf{q} \in \mathbb{R}[\mathbf{X}]$,*

$$\langle \vec{x}(\mathsf{p}), \vec{x}(\mathsf{q}) \rangle = \langle\!\langle \mathsf{p} \times \mathsf{q}, x \rangle\!\rangle.$$

Proof. Follows from definitions of $\vec{x}(\mathsf{p}), \vec{x}(\mathsf{q})$ and Theorem 3.31. □

3.3.1 Binary Labeling Vectors

Notation 3.40 (Indicator Functions of Labelings)**.** *Given non-empty subset $S \subseteq V$ and labeling of S, $f : S \to \{0,1\}$, the indicator function for labeling f is defined as the predicate $\mathbf{X}_S \stackrel{?}{=} f$ which we denote by:*

$$[\![\mathbf{X}_S = f]\!] \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } \mathbf{X}_u = f(u) \text{ for all } u \in S, \\ 0 & \text{else.} \end{cases}$$

Lemma 3.41. *Over $\{0,1\}^V$, $[\![\mathbf{X}_S = f]\!]$ is equal to the following polynomial:*

$$[\![\mathbf{X}_S = f]\!] = \prod_{u \in S} (\mathbf{X}_u + f(u) - 1)^2.$$

They have the following properties:

1. $[\![\mathbf{X}_S = f]\!] \in \Sigma_{2s}$.

2. *For any $f : S \to \{0,1\}$, $g : T \to \{0,1\}$ if f and g are inconsistent, i.e. there is some $u \in S \cap T$ with $f(u) \neq g(u)$, then:*

$$[\![\mathbf{X}_S = f]\!] \times [\![\mathbf{X}_T = g]\!] \equiv 0 \pmod{\mathcal{B}_V}.$$

Otherwise

$$[\![\mathbf{X}_S = f]\!] \times [\![\mathbf{X}_T = g]\!] \equiv [\![\mathbf{X}_{S\cup T} = f \circ g]\!] \pmod{\mathcal{B}_V}.$$

3. *For any $S, T : S \subseteq T$ and $f : S \to \{0,1\}$:*

$$\sum_{g:T\to\{0,1\}, g|_S = f} [\![\mathbf{X}_T = g]\!] \equiv [\![\mathbf{X}_S = f]\!] \pmod{\mathcal{B}_V}.$$

4. *For any S, $[\![\mathbf{X}_S = \mathbb{1}]\!] = \mathbf{X}^{(S)}$.*

Proof. 1. By construction.

2. Observe that $(\mathbf{x}_u + f(u) - 1)^2 \in \{0,1\}$ for any $\mathbf{x}_u \in \{0,1\}$. Hence

$$(\mathbf{X}_u + f(u) - 1)^4 \equiv (\mathbf{X}_u + f(u) - 1)^2.$$

Consequently:

$$[\![\mathbf{X}_S = f]\!] \times [\![\mathbf{X}_T = g]\!] = \prod_{u\in S\setminus T} (\mathbf{X}_u + f(u) - 1)^2 \prod_{v\in T\setminus S} (\mathbf{X}_v + g(v) - 1)^2 \prod_{w\in S\cap T} (\mathbf{X}_w + f(w) - 1)^2 (\mathbf{X}_w + g(w) - 1)^2.$$

If $f(w) = g(w)$ for any $w \in S \cap T$,

$(\mathbf{X}_w + f(w) - 1)^2 (\mathbf{X}_w + g(w) - 1)^2 \equiv (\mathbf{X}_w + (f\circ g)(w) - 1)^4 \equiv (\mathbf{X}_w + (f\circ g)(w) - 1)^2$.

Otherwise: $(\mathbf{X}_w + f(w) - 1)(\mathbf{X}_w + g(w) - 1) \equiv (\mathbf{X}_w - 1)\,\mathbf{X}_w \equiv 0$. Hence the claim follows.

3. First note that

$$(1 - \mathbf{X}_u)^2 + \mathbf{X}_u^2 \equiv 1. \tag{3.8}$$

Given some $f : S \to \{0,1\}$, let $g, h : S \cup \{u\} \to \{0,1\}$ be defined as $g\big|_S = h\big|_S = f$, $g(u) = 0$ and $h(u) = 1$ respectively. Then, using the previous item:

$$\begin{aligned}
[\![\mathbf{X}_{S\cup\{u\}} = g]\!] + [\![\mathbf{X}_{S\cup\{u\}} = h]\!] &\equiv [\![\mathbf{X}_S = f]\!] \left[[\![\mathbf{X}_u = 0]\!] + [\![\mathbf{X}_u = 1]\!]\right] \\
&\equiv [\![\mathbf{X}_S = f]\!] \left[(\mathbf{X}_u - 1)^2 + (\mathbf{X}_u)^2\right] \\
&\equiv [\![\mathbf{X}_S = f]\!] \qquad \text{(by eq. (3.8))}.
\end{aligned}$$

Now claim follows by induction.

4. By construction. □

All our rounding algorithms use $\vec{x}(\llbracket \mathbf{X}_S = f \rrbracket)$. So it will be more convenient to define a short hand notation:

Definition 3.42 ($0/1$-Labeling Vectors)**.** *Given subset $S \subseteq V$ and any labeling $f \in \{0,1\}^S$:*

$$\vec{x}_{S(f)} \stackrel{\text{def}}{=} \vec{x}(\llbracket \mathbf{X}_S = f \rrbracket).$$

The following appears as decomposition theorem in Karlin et al. [2010].

Theorem 3.43. *Given variable set V, family of* **downward closed** *subsets $\mathcal{F} \subseteq 2^V$ and moment sequence $x \in \mathbb{R}^{\mathcal{F} \uplus \mathcal{F}}$ with corresponding moment vectors $\vec{X} = [\vec{x}_S]_{S \in \mathcal{F}}$:*

1. *For any S, T with $S \cup T \in \mathcal{F}$:*

$$\langle \vec{x}_{S(f)}, \vec{x}_{T(g)} \rangle = \begin{cases} \|\vec{x}_{S \cup T(f \circ g)}\|^2 & \text{if } f|_{S \cap T} = g|_{S \cap T}, \\ 0 & \text{else.} \end{cases}$$

2. *For any $S \in \mathcal{F}$:*

$$\sum_{f \in \{0,1\}^S} \vec{x}_{S(f)} = \vec{x}_\emptyset.$$

Proof. 1. We will first prove that $\langle\!\langle \llbracket \mathbf{x}_A = h \rrbracket, x \rangle\!\rangle = \|\vec{x}_{A(h)}\|^2$. Since $\llbracket \mathbf{x}_A = h \rrbracket(\mathbf{x}) \in \{0,1\}$, we have $(\llbracket \mathbf{x}_A = h \rrbracket)^2 = \llbracket \mathbf{x}_A = h \rrbracket$. Therefore:

$$\langle\!\langle \llbracket \mathbf{x}_A = h \rrbracket, x \rangle\!\rangle = \langle\!\langle (\llbracket \mathbf{x}_A = h \rrbracket)^2, x \rangle\!\rangle = \|\vec{x}(\llbracket \mathbf{x}_A = h \rrbracket)\|^2 = \|\vec{x}_{A(h)}\|^2.$$

Now claim follows easily by the following:

$$\begin{aligned} \langle \vec{x}_{S(f)}, \vec{x}_{T(g)} \rangle =& \langle \vec{x}\llbracket \mathbf{X}_S = f \rrbracket, \vec{x}\llbracket \mathbf{X}_T = g \rrbracket \rangle \\ =& \langle\!\langle \llbracket \mathbf{X}_S = f \rrbracket \llbracket \mathbf{X}_T = g \rrbracket, x \rangle\!\rangle && \text{(by Corollary 3.39)} \\ =& \langle\!\langle \llbracket \mathbf{X}_{S \cup T} = f \circ g \rrbracket, x \rangle\!\rangle && \text{(by Lemma 3.41).} \end{aligned}$$

2. Observe that $\vec{x}(\mathbf{X}_\emptyset) = \vec{x}_\emptyset$. By previous item, we see that

$$\langle \vec{x}_{S(f)}, \vec{x}_{S(g)} \rangle = \begin{cases} \|\vec{x}_{S(f)}\|^2 & \text{if } f = g, \\ 0 & \text{else;} \end{cases} \quad \text{and } \langle \vec{x}_{S(f)}, \vec{x}_\emptyset \rangle = \|\vec{x}_{S(f)}\|^2.$$

Hence:

$$\begin{aligned}\left\|\vec{x}_\emptyset - \sum_f \vec{x}_{S(f)}\right\|^2 =& \langle \vec{x}_\emptyset, \vec{x}_\emptyset - \sum_f \vec{x}_{S(g)}\rangle \\ =& \langle \vec{x}_\emptyset, \vec{x}\big(\mathbf{X}_\emptyset - \sum_f [\![\mathbf{X}_S = f]\!]\big)\rangle \\ =& \langle\!\langle \mathbf{X}_\emptyset - \sum_f [\![\mathbf{X}_S = f]\!], x\rangle\!\rangle \qquad \text{(by Lemma 3.41)} \\ =& \langle\!\langle 0, x\rangle\!\rangle = 0 \qquad \text{(by Corollary 3.39).} \quad \square\end{aligned}$$

Using Theorem 3.43, we are ready to prove the hard direction of Theorem 3.36.

Lemma 3.44. *For any $x \in \Sigma^*_{2^V}$ with $x_\emptyset = 1$, there exists a distribution $\mathcal{D}$ on $\{0,1\}^V$ such that:*

$$\forall S \subseteq V : \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\Big[\mathbf{X}_S(\mathbf{x})\Big] = x_S.$$

Furthermore, whenever $\mathbf{M}_V(\mathsf{p} * x) \succeq 0$ for $\mathsf{p} \in \mathbb{R}^{2^V}$:

$$\mathrm{Prob}_{\mathbf{x}\sim\mathcal{D}}\Big[\mathsf{p}(\mathbf{x}) < 0\Big] = 0.$$

Proof. Using Theorem 3.43, we have vectors $\vec{x}_{V(f)}$ for each partitioning $f : V \to \{0,1\}$. Observe that for any S:

$$\langle \vec{x}_S, \vec{x}_{V(f)}\rangle = \|\vec{x}_{V(f)}\|^2 \underbrace{\mathbf{X}^{(S)}(f)}_{\prod_{u\in S} f(u)}.$$

Consider the distribution $\mathcal{D}$ where $f : V \to \{0,1\}$ is chosen with probability $\|\vec{x}_{V(f)}\|^2$:

$$\begin{aligned}\sum_{f:\mathbf{X}^{(S)}(f)=1} \|\vec{x}_{V(f)}\|^2 =& \sum_f \langle \vec{x}([\![\mathbf{X}_S = \mathbb{1}]\!]), \vec{x}_{V(f)}\rangle \\ =& \langle\!\langle \sum_f [\![\mathbf{X}_V = f]\!][\![\mathbf{X}_S = \mathbb{1}]\!], x\rangle\!\rangle \\ =& \langle\!\langle \mathbf{X}^\emptyset \mathbf{X}^{(S)}, x\rangle\!\rangle = x_S,\end{aligned}$$

which proves the first claim.

For the second claim observe that by $[\![\mathbf{X}_V = f]\!] \in \Sigma_{2^V}$ and Theorem 3.34:

$$\begin{aligned} 0 \leq & \langle\!\langle [\![\mathbf{X}_V = f]\!], \mathsf{p} * \vec{x} \rangle\!\rangle = \langle \vec{x}_{V(f)}, \vec{x}(\mathsf{p}) \rangle \\ = & \sum_S [\mathsf{p}]_S \underbrace{\langle \vec{x}_{V(f)}, \vec{x}_S \rangle}_{= \|\vec{x}_{V(f)}\|^2 \mathbf{X}^{(S)}(f)} = \|\vec{x}_{V(f)}\|^2 \sum_S [\mathsf{p}]_S \mathbf{X}^{(S)}(f) = \|\vec{x}_{V(f)}\|^2 \mathsf{p}(f). \end{aligned}$$

Hence whenever $\|\vec{x}_{V(f)}\|^2 \neq 0$ we have $\mathsf{p}(f) \geq 0$. □

3.3.2 k-Labeling Vectors

We can easily generalize eq. (5.12) to handle k-labeling problems in the following way. Suppose the set of variables is V_0.

1. Let the problem domain be $V \leftarrow V_0 \times [k]$ with variables $\mathbf{X} = [\mathbf{X}_{u(i)}]_{u \in V_0, i \in [k]} \in \{0,1\}^{V_0 \times [k]}$. Here $\mathbf{X}_{u(i)}$ denotes the indicator variable for labeling u with i.
2. For each $u \in V_0$ add the following constraint:

$$\sum_{i \in [k]} \mathbf{X}_{u(i)} = 1.$$

We saw how to enforce item 2 in eq. (3.4) so that:

$$\vec{x}_\emptyset = \sum_i \vec{x}_{u(i)}.$$

Notation 3.45 (Indicator Polynomials for k-Labelings)**.** *Given subset $S \subseteq V$ and k-labeling $f : S \to [k]$, we define $[\![\mathbf{X}_S = f]\!]$ as the following polynomial:*

$$[\![\mathbf{X}_S = f]\!](\mathbf{x}) = \prod_{u \in S} \mathbf{X}^2_{u(f(u))}.$$

Definition 3.46 (k-Labeling Vectors)**.** *Given subset $S \subseteq V$ and any labeling $f : S \to [k]$, let:*

$$\vec{x}_{S(f)} \stackrel{\text{def}}{=} \vec{x}([\![\mathbf{x}_S = f]\!]).$$

Theorem 3.47. *Given variable set V_0, positive integers k, with $V \stackrel{\text{def}}{=} V_0 \times [k]$; down family $\mathcal{F}$ over V and moment sequence $x \in \Sigma^*_{\mathcal{F}}$; provided that*

$$\mathcal{F} \supseteq (V)_{\leq 2}$$

and

$$\vec{x}\left(\sum_i \mathbf{X}_{u(i)} - 1\right) = 0 \text{ for all } u \in V_0,$$

the vectors as given in Definition 3.46 satisfy the following:

1. *For any S, T with $[k]^{S\cup T} \in \mathcal{F}$:*

$$\langle\!\langle \vec{x}_{S(f)}, \vec{x}_{T(g)} \rangle\!\rangle = \begin{cases} \|\vec{x}_{S\cup T(f\circ g)}\|^2 & \text{if } f|_{S\cap T} = g|_{S\cap T}, \\ 0 & \text{else.} \end{cases}$$

2. *For any $S : [k]^S \in \mathcal{F}$:*

$$\sum_{f:S\to[k]} \vec{x}_{S(f)} = \vec{x}_{\emptyset}.$$

Proof. We will prove that

$$[\![\mathbf{X}_u = f]\!][\![\mathbf{X}_u = g]\!] \equiv \begin{cases} [\![\mathbf{X}_u = f]\!] & \text{if } f = g, \\ 0 & \text{else.} \end{cases}$$

By construction, $[\![\mathbf{X}_u = f]\!] \in \{0, 1\}$ from which we see that

$$[\![\mathbf{X}_u = f]\!][\![\mathbf{X}_u = f]\!] = [\![\mathbf{X}_u = f]\!]^2 = [\![\mathbf{X}_u = f]\!].$$

Suppose $f(u) = i$ and $g(u) = j$ with $i \neq j$. Then

$$[\![\mathbf{X}_u = f]\!][\![\mathbf{X}_u = g]\!] \equiv \mathbf{X}_{u(i)}\mathbf{X}_{u(j)}.$$

Let $\mathcal{J}$ be the following ideal:

$$\mathcal{J} \leftarrow \left(\mathbf{X}_{u(1)} - \mathbf{X}_{u(1)}^2, \ldots, \mathbf{X}_{u(k)} - \mathbf{X}_{u(k)}^2, 1 - \sum_i \mathbf{X}_{u(i)}\right).$$

One can easily check that

$$(\mathbf{x}_{u(1)}, \ldots, \mathbf{x}_{u(k)}) \in \text{Variety}(\mathcal{J}) \implies \mathbf{x}_{u(i)}\mathbf{x}_{u(j)} = 0 \text{ for any } i \neq j,$$

and that $\mathcal{I}(\text{Variety}(\mathcal{J})) = \mathcal{J}$. Consequently

$$\mathbf{X}_{u(i)}\mathbf{X}_{u(j)} \equiv 0 \text{ whenever } i \neq j.$$

Given this, our theorem can be proven exactly in the same way with Lemma 3.41 and Theorem 3.43. □

Chapter 4

Case Study: Minimum Bisection

In this chapter, we will illustrate the main ideas involved in our work in a simplified setting, by working out progressively better approximation ratios for the following basic, well-studied problem: Given as input a graph $G = (V, E)$ with $n = |V|$, and an integer size parameter μ, find a subset $U \subset V$ with $|U| = \mu$ that minimizes the number of edges between U and $V \setminus U$. The special case of $\mu = |n|/2$ when we want to partition the vertex set into two equal parts is the minimum bisection problem. We will loosely refer to the general μ case also as minimum bisection.[1]

This chapter is intended to be somewhat informal introduction to our basic rounding algorithms and main ideas behind their analysis. We will present a more formal treatment of this problem including a bibliography in Chapter 7.

For simplicity we will assume G is unweighted and d-regular, however all our results given in Chapters 7 to 9 are for any non-negative weighted undirected graph G. We can formulate minimum bisection as a binary integer programming problem as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{e=\{u,v\} \in E} (\mathbf{x}_u - \mathbf{x}_v)^2, \\ \text{st} \quad & \sum_u \mathbf{x}_u = \mu \quad \text{and} \quad \mathbf{x} \in \{0,1\}^V. \end{aligned} \tag{4.1}$$

If we let L be the Laplacian matrix for G, we can rewrite the objective as $\mathbf{x}^T L \mathbf{x}$.

Note that the above is a quadratic integer programming (QIP) problem with linear constraints. The somewhat peculiar formulation is in anticipation of the

[1] We will be interested in finding a set of size $\mu \pm o(\mu)$, so we avoid the terminology *Balanced Separator* which typically refers to the variant where $\Omega(n)$ slack is allowed in the set size.

Lasserre semi-definite programming relaxation for this problem, which we describe below.

4.1 Lasserre Hierarchy Relaxation

Recall from Chapter 3, in particular Corollary 3.37, that r'-rounds of Lasserre Hierarchy relaxation corresponding to eq. (4.1) can be written as the following SDP for any given positive integer r':

$$\begin{array}{ll} \min & \sum_{e=\{u,v\}\in E(G)} \|\vec{x}_u - \vec{x}_v\|^2 \\ \text{st} & \|\sum_u \vec{x}_u - \mu\vec{x}_\emptyset\|^2 = 0, \\ & \|\vec{x}_\emptyset\|^2 = 1, \\ & \langle \vec{x}_S, \vec{x}_T\rangle = x_{S\cup T} \text{ for all } S,T \in \binom{V}{\le r'}, \\ & x = [x_S] \in \mathbb{R}^{\binom{V}{\le 2r'}}. \end{array} \tag{4.2}$$

It is easy to see that this is indeed a relaxation of our original QIP formulation given in eq. (4.1).

4.2 Main Theorem on Rounding

Let x be an (optimal) solution to the above r'-round Lasserre SDP with labeling vectors $[\vec{x}_{S(f)}]_{S\in\binom{V}{\le r'},f\in\{0,1\}^S}$ as in Definition 3.42. We will always use OPT in this section to refer to the objective value of x, i.e., $\mathsf{OPT} = \sum_{e=\{u,v\}\in E(G)} \|\vec{x}_u - \vec{x}_v\|^2$.

Our ultimate goal in this section is to give an algorithm to round the SDP solution x to a good cut U of size very close to μ, and prove the below theorem.

Theorem 4.1. *For all $r \ge 1$ and $\varepsilon > 0$, there exists $r' = O\left(\frac{r}{\varepsilon^2}\right)$, such that given a feasible solution x to eq. (4.2) with objective value equal to* OPT*, one can find in randomized $2^{O(r/\varepsilon^2)}n^{O(1)}$ time, a partitioning $\mathbf{x} \in \{0,1\}^V$ satisfying the following two properties w.h.p:*

1. $\mathbf{x}^T L\mathbf{x} \le \frac{1+\varepsilon}{\min(1,\lambda_{r+1}(G)/d)}\mathsf{OPT}$.
2. $\mu(1-o(1)) = \mu - O\left(\sqrt{\mu\log(1/\varepsilon)}\right) \le \|\mathbf{x}\|_1 \le \mu + O\left(\sqrt{\mu\log(1/\varepsilon)}\right) = \mu(1+o(1))$.

Since one can solve the Lasserre relaxation in $n^{O(r')}$ time, we get the result claimed in the introduction: an $n^{O(r/\varepsilon^2)}$ time factor $(1+\varepsilon)/\min\{\lambda_r(G),1\}$ approximation algorithm. Note that if $t = \operatorname{argmin}_r\{r \mid \lambda_r(G)/d \ge 1-\varepsilon/2\}$, then this gives

an $n^{O_\varepsilon(t)}$ time algorithm for approximating minimum bisection to within a $(1+\varepsilon)$ factor, provided we allow $O(\sqrt{n})$ imbalance. The formal theorem for general (non-regular, weighted) graphs appears as Corollary 7.14 in Chapter 7, where we will also demonstrate how to combine this with the faster solver from Chapter 5 and decrease running time to $2^{O_\varepsilon(t)} n^{O_\varepsilon(1)}$.

4.3 The Rounding Algorithm

Recall that labeling vectors $[\vec{x}_{S(f)}]$ corresponding to moment sequence x contains a vector $\vec{x}_{T(f)}$ for each $T \in \binom{V}{\leq r'}$ and every possible labeling of T, $f \in \{0,1\}^T$ of T. Our approach to round x to a solution $\mathbf{x}$ to the integer program eq. (4.1) is similar to the label propagation approach used by Arora et al. [2008a].

Consider fixing a set of r' nodes, $S \in \binom{V}{\leq r'}$, and assigning a label $f(s)$ to every $s \in S$ by choosing $f \in \{0,1\}^S$ with probability $\|\vec{x}_{S(f)}\|^2$. (The best choice of S can be found by brute-forcing over all of $\binom{V}{\leq r'}$, since solving the Lasserre SDP takes $n^{O(r')}$ time anyway. But there is also a faster method to find a good S, as mentioned in Theorem 10.1.) Conditional on choosing a specific labeling f to S, we propagate the labeling to other nodes as follows: Independently for each $u \in V$, assign $\mathbf{x}_u \leftarrow i$ where $i \in \{0,1\}$ with probability

$$\text{Prob}\left[\mathbf{x}_u = i \middle| f\right] = \frac{\langle \vec{x}_{S(f)}, \vec{x}_{u(i)} \rangle}{\|\vec{x}_{S(f)}\|^2} = \frac{\langle \overline{\vec{x}_{S(f)}}, \vec{x}_{u(i)} \rangle}{\|\vec{x}_{S(f)}\|} = \frac{1}{\|\vec{x}_{S(f)}\|} \overline{\vec{x}_{S(f)}}^T \vec{x}_{u(i)}.$$

Observe that if $u \in S$, label of u will always be $f(u)$. Finally, we output $\mathbf{x} \in \{0,1\}^V$ as the final partitioning.

Before analyzing the partition size and number of edges cut, we will define a specific matrix based on S in Definition 4.2 below.

Definition 4.2. *Given labeling vectors $[\vec{x}_{S(f)}]_{S,f}$ representing a moment sequence x we define $\Pi_S \in \mathbb{R}^{\Upsilon \times \Upsilon}$ as the projection matrix onto the span of $\{\vec{x}_{S(f)}\}_{f \in \{0,1\}^S}$ for given S:*

$$\Pi_S \stackrel{\text{def}}{=} \sum_{f \in \{0,1\}^S} \overline{\vec{x}_{S(f)}} \cdot \overline{\vec{x}_{S(f)}}^T.$$

Define $\Pi_S^\perp = I - \Pi_S$ to be the projection matrix onto the orthogonal complement of the span of $\{\vec{x}_{S(f)}\}_{f \in \{0,1\}^S}$.

Rather than diving into the properties of Definition 4.2 right away, we will defer them based on need.

Lemma 4.3. *For the above rounding procedure, the cost of the partitioning produced* $\mathbf{x}^T L\mathbf{x}$ *satisfies*

$$\mathbb{E}\left[\mathbf{x}^T L\mathbf{x}\right] = \mathsf{OPT} + \sum_{(u,v)\in E} \langle \Pi_S^{\perp}\vec{x}_u, \Pi_S^{\perp}\vec{x}_v\rangle \ . \tag{4.3}$$

Proof. Note that for $u \neq v$, and $i, j \in \{0,1\}$,

$$\begin{aligned}\mathrm{Prob}\left[\mathbf{x}_u = i \wedge \mathbf{x}_v = j\right] &= \sum_f \|\vec{x}_{S(f)}\|^2 \frac{\langle \overline{\vec{x}_{S(f)}}, \vec{x}_{u(i)}\rangle}{\|\vec{x}_{S(f)}\|} \frac{\langle \overline{\vec{x}_{S(f)}}, \vec{x}_{v(j)}\rangle}{\|\vec{x}_{S(f)}\|} \\ &= \sum_f \langle \overline{\vec{x}_{S(f)}}, \vec{x}_{u(i)}\rangle \langle \overline{\vec{x}_{S(f)}}, \vec{x}_{v(j)}\rangle.\end{aligned}$$

Since $\{\overline{\vec{x}_{S(f)}}\}_f$ is an orthonormal basis, the above expression can be written as the inner product of *projections* of $\vec{x}_{u(i)}$ and $\vec{x}_{v(j)}$ onto the span of $\{\vec{x}_{S(f)}\}_{f\in\{0,1\}^S}$, which we denote by Π_S. Let us now calculate the expected number $\mathbf{x}^T L\mathbf{x}$ of edges cut by this rounding. It is slightly more convenient to treat edges $e = \{u, v\}$ as two directed edges (u, v) and (v, u), and count directed edges (u, v) with $u \in U$ and $v \in V \setminus U$ in the cut. Therefore,

$$\begin{aligned}\mathbb{E}\left[\text{number of edges cut}\right] &= \sum_{(u,v)\in E} \langle \Pi_S\vec{x}_u, \Pi_S\vec{x}_{v(2)}\rangle = \sum_{(u,v)\in E} \langle \Pi_S\vec{x}_u, \Pi_S(\vec{x}_\emptyset - \vec{x}_v)\rangle \\ &= \sum_{(u,v)\in E} \langle \Pi_S\vec{x}_u, \Pi_S\vec{x}_\emptyset\rangle - \langle \Pi_S\vec{x}_u, \Pi_S\vec{x}_v)\rangle \end{aligned} \tag{4.4}$$

By using the fact that $\langle \Pi_S\vec{x}_u, \Pi_S\vec{x}_{\emptyset=\langle\vec{x}_u}, \Pi_S\vec{x}_\emptyset\rangle = \langle \vec{x}_u, \vec{x}_\emptyset\rangle = \|\vec{x}_u\|^2$, we can rewrite eq. (4.4) in the following way:

$$\begin{aligned}&= \sum_{(u,v)\in E} \|\vec{x}_u\|^2 - \langle \Pi_S\vec{x}_u, \Pi_S\vec{x}_v\rangle \\ &= \sum_{(u,v)\in E} \|\vec{x}_u\|^2 - \langle \vec{x}_u, \vec{x}_v\rangle + \langle \Pi_S^{\perp}\vec{x}_u, \Pi_S^{\perp}\vec{x}_v\rangle \\ &= \mathsf{OPT} + \sum_{(u,v)\in E} \langle \Pi_S^{\perp}\vec{x}_u, \Pi_S^{\perp}\vec{x}_v\rangle. \end{aligned}$$

□

Note that the matrix Π_S depends on vectors $\vec{x}_{S(f)}$ which are hard to control because we do not have any constraint relating $\vec{x}_{S(f)}$ to a known matrix. The main driving force behind all our results is the following fact, which follows since given any $u \in S$ and $i \in \{0,1\}$, $\vec{x}_{u(i)} = \sum_{f:f(u)=i} \vec{x}_{S(f)}$ by Lasserre constraints.

Observation 4.4. *For all* $S \in \binom{V}{\leq r'}$,

$$\operatorname{span}\left(\{\vec{x}_{S(f)}\}_{f\in\{0,1\}^S}\right) \supseteq \operatorname{span}\left(\{\vec{x}_{u(i)}\}_{u\in S, i\in\{0,1\}}\right) .$$

Equivalently for P_S *being the projection matrix onto span of* $\{\vec{x}_{u(i)}\}_{u\in S,i\in 0,1}$, $P_S \preceq \Pi_S$.

Thus we will try to upper bound the term in eq. (4.3) by replacing $\Pi_S^\perp$ with $P_S^\perp$, but we cannot directly perform this switch: $\langle P_S^\perp \vec{x}_{u(i)}, P_S^\perp \vec{x}_{v(j)}\rangle$ might be negative while $\Pi_S^\perp \vec{x}_{u(i)} = 0$.

4.4 Factor $1 + \frac{1}{\lambda_r}$ Approximation of Cut Value

Our first bound is by directly upper bounding eq. (4.3) in terms of $\|\Pi_S^\perp \vec{x}_{u(i)}\|^2 \leq \|P_S^\perp \vec{x}_{u(i)}\|^2$. Using Cauchy-Schwarz and Arithmetic-Geometric Mean inequalities, eq. (4.3) implies that the expected number of edges cut is upper bounded by

$$\mathsf{OPT} + \frac{1}{2}\sum_{e=(u,v)\in E} \|\Pi_S^\perp \vec{x}_u\|^2 + \|\Pi_S^\perp \vec{x}_v\|^2 = \mathsf{OPT} + d\sum_u \|\Pi_S^\perp \vec{x}_u\|^2 \leq \mathsf{OPT} + d\sum_u \|P_S^\perp \vec{x}_u\|^2 . \tag{4.5}$$

Now define $\vec{X} \stackrel{\text{def}}{=} [\vec{x}_u]_{u\in V} \in \mathbb{R}^{\Upsilon\times V}$ as the matrix whose columns correspond to vectors $\vec{x}_u$. By eq. (4.2), we have the objective value $\mathsf{OPT} = \operatorname{Tr}(\vec{X}^T \vec{X} L)$. Let $\vec{X}_S^\Pi$ be the projection matrix onto the span of $\{\vec{X}_u\}_{u\in S}$. Since this set is a subset of $\{\vec{x}_{u(i)}\}_{u\in S,i\in\{0,1\}}$, we have $\vec{X}_S^\Pi \preceq P_S$. Therefore, we can bound eq. (4.5) further as

$$\mathbb{E}\left[\text{number of edges cut}\right] \leq \mathsf{OPT} + d\sum_u \|\vec{X}_S^\perp \vec{X}_u\|^2 = \mathsf{OPT} + d\cdot\operatorname{Tr}(\vec{X}^T \vec{X}_S^\perp \vec{X}) . \tag{4.6}$$

To get the best upper bound, we want to pick $S \in \binom{V}{\leq r'}$ to minimize $\sum_{u\in V} \|\vec{X}_S^\perp \vec{X}_u\|^2$. It is a well known fact that among *all* projection matrices M of rank r' (not necessarily restricted to projection onto columns of $\vec{X}$), the minimum value of $\sum_u \|M^\perp \vec{X}_u\|^2 = \operatorname{Tr}(\vec{X}^T M^\perp \vec{X})$ is achieved by matrix M projecting onto the space of the largest r' singular vectors of $\vec{X}$. Further, this minimum value equals $\sum_{i\geq r'+1}\sigma_i$ where $\sigma_i = \sigma_i(\vec{X})$ denotes the *squared* i^{th} largest singular value of $\vec{X}$ (equivalently $\sigma_i(\vec{X})$ is the i^{th} largest eigenvalue of $\vec{X}^T\vec{X}$). Hence $\operatorname{Tr}(\vec{X}^T\vec{X}_S^\perp\vec{X}) \geq \sum_{i\geq r'+1}\sigma_i$ for every choice of S. The following theorem, which is a restatement of Theorem 10.12, shows the existence of S which comes close to this lower bound:

Theorem 4.5 (Restatement of Theorem 10.12)**.** *For every real matrix $\vec{X}$ with column set V, and positive integers $r \le r'$, one can find a subset $S \in \binom{V}{r'}$ deterministically in time $\mathrm{poly}(n)$ such that*

$$\delta_{r'}(\vec{X}) \stackrel{\text{def}}{=} \min_{C \in \binom{V}{\le r'}} \mathrm{Tr}(\vec{X}^T \vec{X}_C^{\perp} \vec{X}) \le \mathrm{Tr}(\vec{X}^T \vec{X}_S^{\perp} \vec{X}) \le \frac{r'+1}{r'-r+1} \Big(\sum_{i \ge r+1} \sigma_i \Big) .$$

In particular, for all $\varepsilon \in (0,1)$, one can find $r' \le r + r/\varepsilon - 1$ columns in polynomial time whose squared reconstruction error is within $1+\varepsilon$ times best rank-r error.

Remark 4.6. *Boutsidis et al. [2011] showed that $\delta_{r(2+\varepsilon)/\varepsilon} \le (1+\varepsilon)\Big(\sum_{i \ge r+1} \sigma_i\Big)$. The improvement in the bound on r' from $2r/\varepsilon$ to r/ε to achieve $(1+\varepsilon)$ approximation is not of major significance to our application, but since the tight bound is now available, we decided to state and use it.* □

Remark 4.7 (Running time of our algorithms)**.** *For a naïve implementation of our rounding algorithm, existence bound of Theorem 4.5 is sufficient since we can find such S by simply enumerating all $r' = O(r/\varepsilon)$-subsets in time $n^{O_\varepsilon(r)}$: Even storing the full Lasserre Hierarchy relaxation will take time $n^{O_\varepsilon(r)}$ anyway, so running time will not be affected.*

However it is possible to bypass computing or storing the full SDP solution by carefully exploiting the structure of our rounding algorithm, in turn opening up possibilities for improving the running time. We give one such solver in Chapter 5 which only partially constructs SDP solution achieving a running time of $2^{O_\varepsilon(r)} n^{O_\varepsilon(1)}$. For this case, finding good set of columns, S (deterministically) in $\mathrm{poly}(n)$ time becomes crucial. □

Picking the subset $\mathcal{S}^* \in \binom{V}{\le r'}$ that achieves the bound guaranteed in eq. (4.6), say by using Algorithm 10.2, we end up with

$$\mathrm{Tr}(\vec{X}^T \vec{X}_{\mathcal{S}^*}^{\perp} \vec{X}) = \delta_{r/\varepsilon}(\vec{X}) \le (1-\varepsilon)^{-1} \sum_{i>r} \sigma_i .$$

In order to relate this quantity to the SDP objective value $\mathsf{OPT} = \mathrm{Tr}(\vec{X}^T \vec{X} L)$, we use the fact that $\mathrm{Tr}(\vec{X}^T \vec{X} L)$ is minimized when eigenvectors of $\vec{X}^T \vec{X}$ and L are matched in reverse order: i^{th} largest eigenvector of $\vec{X}^T \vec{X}$ corresponds to i^{th} smallest eigenvector of L. Letting $0 = \lambda_1(G) \le \lambda_2(G) \le \ldots \le \lambda_n(G) \le 2d$ be the eigenvalues of graph Laplacian matrix, L, we have

$$\mathsf{OPT} = \mathrm{Tr}(\vec{X}^T \vec{X} L) \ge \sum_i \sigma_i(\vec{X}) \lambda_i(G) \ge \sum_{i \ge r+1} \sigma_i(\vec{X}) \lambda_{r+1}(G) \ge (1-\varepsilon) \lambda_{r+1}(G) \delta_{r/\varepsilon}(\vec{X}).$$

Plugging this into eq. (4.6), we can conclude our first bound:

Theorem 4.8. *For all positive integers r and $\varepsilon \in (0,1)$, given a feasible solution $x \in \mathbb{R}^{\binom{V}{\leq \lceil r/\varepsilon \rceil}}$ to the SDP problem eq. (4.2), the rounding algorithm given in Section 4.3 cuts at most*

$$\left(1 + \frac{d}{(1-\varepsilon)\lambda_{r+1}(G)}\right) \sum_{e=(u,v)\in E} \|\vec{x}_u - \vec{x}_v\|^2$$

edges in expectation.

In particular, the algorithm cuts at most a factor $\left(1 + \frac{d}{(1-\varepsilon)\lambda_{r+1}(G)}\right)$ more edges than the optimal cut with μ nodes on one side.[2]

Note that $\lambda_n(G) \leq 2$, hence even if we use n-rounds of Lasserre relaxation, for which x is an integral solution, we can only show an upper bound $\geq \frac{3}{2}$. Although this is too weak by itself for our purposes, this bound will be crucial to obtain our final bound.

4.5 Improved Analysis and Factor $\frac{1}{\lambda_r}$ Approximation on Cut Value

First notice that eq. (4.3) can be written as

$$\begin{aligned}\mathbb{E}\Big[\text{number of edges cut}\Big] &= \mathrm{Tr}(\vec{X}^T\vec{X}L) + \mathrm{Tr}(\vec{X}^T\Pi_S^{\perp}\vec{X}(I-L)) \\ &= \mathrm{Tr}(\vec{X}^T\Pi_S^{\perp}\vec{X}) + \mathrm{Tr}(\vec{X}^T\Pi_S\vec{X}L). \qquad (4.7)\end{aligned}$$

If value of this expression is larger than $\frac{\mathsf{OPT}}{(1-\varepsilon)\lambda_{r+1}} + \mathsf{OPT}\varepsilon$, then value of $\mathrm{Tr}(\vec{X}^T\Pi_S\vec{X}L)$ has to be larger than $\varepsilon\mathsf{OPT}$ due to the bound we proved on $\mathrm{Tr}(\vec{X}^T\Pi_S^{\perp}\vec{X})$. Consider choosing another subset T that achieves the bound $\delta_r(\Pi_S^{\perp}\vec{X})$. The crucial observation is that distances between neighboring nodes on vectors $\Pi_S^{\perp}\vec{X}$ has decreased by an additive factor of $\mathsf{OPT}\varepsilon$,

$$\mathrm{Tr}(\vec{X}^T\Pi_S^{\perp}\vec{X}L) = \mathrm{Tr}(\vec{X}^T\vec{X}L) - \mathrm{Tr}(\vec{X}^T\Pi_S\vec{X}L) < \mathsf{OPT}(1-\varepsilon)$$

so that $\mathrm{Tr}(\vec{X}^T\Pi_{S\cup T}^{\perp}\vec{X}) < (1-\varepsilon)\frac{\mathsf{OPT}}{(1-\varepsilon)\lambda_{r+1}}$. Now, if we run the rounding algorithm with $S \cup T$ as the seed set, and eq. (4.7) with $S \cup T$ in place of S is larger than $\frac{\mathsf{OPT}}{(1-\varepsilon)\lambda_{r+1}} + \mathsf{OPT}\varepsilon$, then $\mathrm{Tr}(\vec{X}^T\Pi_{S\cup T}\vec{X}L) > 2\varepsilon\mathsf{OPT}$. Hence

$$\mathrm{Tr}(\vec{X}^T\Pi_{S\cup T}^{\perp}\vec{X}L) \leq \mathrm{Tr}(\vec{X}^T\vec{X}L) - \mathrm{Tr}(\vec{X}^T\Pi_{S\cup T}\vec{X}L) < \mathsf{OPT}(1-2\varepsilon)\,.$$

[2]We will later argue that the cut will also meet the balance requirement up to $o(\mu)$ vertices.

Picking another set T', we will have $\mathrm{Tr}(\vec{X}^T \Pi^{\perp}_{S \cup T \cup T'} \vec{X}) < (1-2\varepsilon)\frac{\mathsf{OPT}}{(1-\varepsilon)\lambda_{r+1}}$. Continuing this process, if the quantity eq. (4.7) is not upper bounded by $\frac{\mathsf{OPT}}{(1-\varepsilon)\lambda_{r+1}} + \mathsf{OPT}\varepsilon$ after $\lceil \frac{1}{\varepsilon} \rceil$ many such iterations, then the total projection distance becomes

$$\mathrm{Tr}(\vec{X}^T \Pi^{\perp}_{S \cup T \cup \ldots} \vec{X}) < (1 - \lceil 1/\varepsilon \rceil \varepsilon) \frac{\mathsf{OPT}}{(1-\varepsilon)\lambda_{r+1}} \leq 0$$

which is a contradiction. For formal statement and proof in a more general setting, see Theorem 7.7 in Chapter 7.

Theorem 4.9. *For all positive integer r and $\varepsilon \in (0,1)$, letting $r' = O\left(\frac{r}{\varepsilon^2}\right)$, given a feasible solution $x \in \mathbb{R}^{\binom{V}{\leq 2r'}}$ to the SDP problem eq. (4.2), the expected number of edges cut by the above rounding algorithm is at most $(1+\varepsilon)/\min\{1, \lambda_{r+1}(G)\}$ times the size of the optimal cut with μ nodes on one side. (Here $\lambda_{r+1}(G)$ is the $(r+1)$'th smallest eigenvalue of the Laplacian matrix L for graph G.)*

4.6 Bounding Set Size

We now analyze the balance of the cut, and show that we can ensure that $|U| = \mu \pm o(\mu)$ in addition to the cut cost, $\mathbf{x}^T L \mathbf{x}$, being close to the expected bound of Theorem 4.9 (and similarly for Theorem 4.8).

Let $\mathcal{S}^*$ fixed to be $\mathrm{argmin}_{S \in \binom{V}{\leq r'}} \mathrm{Tr}(\vec{X}^T \vec{X}_S^{\perp} \vec{X})$. We will show that conditioned on finding cuts with small cut cost, $\mathbf{x}^T L \mathbf{x}$, the probability that one of them has $|U| \approx \mu$ is bounded away from zero. We can use a simple Markov bound to show that there is a non-zero probability that both cut size and set size are within 3-factor of corresponding bounds. But by exploiting the independence in our rounding algorithm and Lasserre relaxations of linear constraints, we can do much better. Note that in the r'-round Lasserre relaxation, for each $f \in \{0,1\}^{\mathcal{S}^*}$, due to the set size constraint in original IP formulation, $\vec{x}$ satisfies:

$$\sum_u \mathbf{x}_u = \mu \implies \sum_u \langle \vec{x}_{\mathcal{S}^*(f)}, \vec{x}_u \rangle = \mu \|\vec{x}_{\mathcal{S}^*(f)}\|^2 .$$

This implies that conditioned on the choice of f, the expectation of $\sum_u \mathbf{x}_u$ is μ and events $\mathbf{x}_u = 1$ over all $u \in V$ are independent. Applying the Chernoff bound, we get

$$\mathrm{Prob}_{\mathbf{x}}\left[\left| \sum_u \mathbf{x}_u - \mu \right| \geq 2\sqrt{\mu \log \frac{1}{\zeta}} \right] \leq o(\zeta) \leq \frac{\zeta}{3}.$$

Consider choosing $f \in \{0,1\}^{\mathcal{S}^*}$ so that

$$\mathbb{E}\big[\text{number of edges cut} \,\big|\, f\big] \leq \mathbb{E}\big[\text{number of edges cut}\big] \stackrel{\text{def}}{=} b.$$

By Markov inequality, if we pick such an f, $\mathsf{Pr}\big[\text{number of edges cut} \geq (1+\zeta)b\big] \leq 1 - \frac{\zeta}{2}$, where the probability is over the random propagation once $\mathcal{S}^*$ and f are fixed.

Hence with probability at least $\frac{\zeta}{6}$, the solution $\mathbf{x}$ will yield a partitioning $\mathbf{x}$ with $\mathbf{x}^T L \mathbf{x} \leq (1+\zeta)b$ and size $\|\mathbf{x}\|_1$ in the range $\mu \pm 2\sqrt{\mu \log \frac{1}{\zeta}}$. Taking $\zeta = \varepsilon$ and repeating this procedure $O\left(\varepsilon^{-1} \log n\right)$ times, we get a high probability statement and finish our main Theorem 4.1 on minimum bisection.

Chapter 5

Local Rounding Framework and Faster Solvers

In this chapter, we present a rounding framework that captures all our rounding algorithms. For this framework, we present a faster solver of the underlying convex relaxation.

5.1 Introduction

A (near)-optimal solution to the r'th level Lasserre relaxation can be found in $n^{O(r)}$ time. So understanding the power of these relaxations for small values of r, such as $r = \Theta(\log n)$, is of particular interest. The main contribution of this work is to improve the running time of various Lasserre-based approximation algorithms to $2^{O(r)}n^{O(1)}$ (from the default $n^{O(r)}$). In particular, the guarantees achieved by $O(\log n)$ rounds of Lasserre SDPs can be realized in polynomial time. Another use of the Lasserre hierarchy to find graph bisections was given by Raghavendra and Tan [2012], which we can also similarly speed up. A table of several algorithms whose algorithms we are able to improve is given in Chapter 6.

Theorem 5.1 (Informal)**.** *Given an undirected graph G, we can find a bisection cutting at most $O(\sqrt{\mathsf{OPT}})$ edges in time $2^{\mathrm{poly}(1/\mathsf{OPT})}n^{O(1)}$ where OPT is the fraction of edges cut by the minimum bisection. A similar result holds for Maximum Bisection, where we find a cut of size at least $1 - O(\sqrt{\eta})$ when the optimum bisection cuts a fraction $1 - \eta$ of the edges, in time $2^{\mathrm{poly}(1/\eta)}n^{O(1)}$.*

In our thesis, from this chapter on, all our rounding algorithms in Chapters 7

to 9 will be presented in terms of this framework and their running times will be bounded assuming Algorithm 5.2 is used.

Our techniques might also be useful in the context of fixed-parameter tractability, which we leave as a potentially interesting avenue for future research.

Local rounding algorithms. Note that even writing down the full r-round Lasserre solution takes $n^{\Omega(r)}$ time. The hope to speed-up the algorithms to a runtime dependence of $2^{O(r)}$ is based on the observation that many of the rounding algorithms have a "local" character that uses only a small portion of the SDP solution. In the simplest setting, the rounding algorithm proceeds in two steps: (i) find a "seed set" S^* of $\approx r$ nodes based only the solution to the base (1-round) SDP, and (ii) use the value of r-round Lasserre solution on the set S^* to sample a partial assignment to S^* and then propagate it to the other nodes. Thus the rounding algorithm only uses the portion of the Lasserre SDP corresponding to the subsets $S^* \cup \{u\}$ for various u. Further, the analysis of the rounding algorithm also relies only on Lasserre consistency constraints for subsets of S^*. More generally, the algorithms might pick a sequence of seed subsets $S_1, S_2, \ldots, S_\ell$ iteratively and the SDP solution restricted to subsets of $S_1 \cup S_2 \cup \cdots S_\ell$ is used for rounding.

Note that the needed portion of the solution (corresponding to S^*, or more generally $S_1 \cup S_2 \cup \cdots S_\ell$) itself depends on certain other parts of the solution. So one cannot simply project the space down to the relevant dimensions to find the required part of the Lasserre solution. Our main technical contribution is an ellipsoid algorithm that can find the needed partial solution (which satisfies all the local constraints induced on those variables) in time polynomial in the number of variables in the partial solution. We stress that the partial solution we find *may not* extend to a full Lasserre solution. This, however, does not matter for the approximation guarantee as it will *"fool" the rounding algorithm* which can't distinguish the solution we find from a global Lasserre solution.

There are two examples of hierarchy based approximation algorithms which have been sped up to $2^{O(r)}$ dependence on the number of rounds, both of which rely on weaker hierarchies than Lasserre: (i) the algorithm of Chlamtac et al. [2010] for sparsest cut on bounded tree-width graphs using the Sherali-Adams hierarchy and (ii) the Unique Games algorithm of Barak et al. [2011] based on the "mixed" hierarchy. The faster algorithm is for the former case is immediate as the required portion of the solution only depends on the input graph, so one can simply find that part using any LP solver. For the Unique Games algorithm, the seed set S^* depends on the vector solution to the basic SDP relation. The goal is to extend the solution to local distributions of labels on subsets $S^* \cup \{u\}$ for various nodes u, whose 2-way marginals agree with the vector inner products. As briefly sketched

by Barak et al. [2011], these constraints form a linear program, and if infeasible, by Farkas' lemma, one can get a new constraint for the vector inner products, which can be fed into an ellipsoid algorithm for solving the basic SDP. Our situation is more complicated as we handle several iterations of seed set selection, and the "extension" problems we solve are no longer simple linear programs. Also, the runtime of the Unique Games algorithm of Barak et al. [2011] had an exponential dependence on the number of labels, as opposed to our polynomial dependence.

Main technique: Separation oracle with restricted support. We describe the high level ideas behind our method for finding adequate partial solutions to Lasserre SDPs in Section 5.2. Our approach applies in a fairly general set-up, and therefore we describe our methods in an abstract framework for clarity, both in Section 5.2 and later in Section 5.4 where the formal details appear.

In addition to the runtime improvements, our results contribute a useful, and to our knowledge new, basic tool in convex optimization, which is an ***efficient ellipsoid algorithm based separation oracle that can output a certificate of infeasibility with restricted support*** (or more generally belonging to a restricted subspace). For instance, suppose we are given a convex body $K \subseteq \mathbb{R}^n$ via a separation oracle for it. Given a point $y \in \mathbb{R}^U$ (a potential partial solution) for some $U \subset \{1, 2, \ldots, n\}$, we give an algorithm to either find $x \in K$ such that $\mathrm{proj}_U(x) = y$ (if one exists[1]), or find a separating constraint *that is supported on* U.

5.2 Our Rounding Framework and Method Overview

Consider a rounding algorithm with following property: Given an optimal solution $x \in \mathbb{R}^N$ as input, it only reads a much smaller part of this solution, say $T \subseteq N$ with $|T| = o(|N|)$. We call these "local" rounding algorithms: Even though this setting might sound too restrictive and/or unrealistic, observe that all our rounding algorithms as well as other algorithms which use "hierarchies" fit into this framework such as [Chlamtac and Singh, 2008, Karlin et al., 2010, Arora and Ge, 2011, Raghavendra and Tan, 2012]. See Chapter 6 for details.

We first start by outlining a generic *iterative rounding* algorithm. This framework depends on two application specific deterministic[2] procedures, **SEED** and **FEASIBLE**. Without going into the formal details, at a high level, $\mathbf{SEED}_S$ procedure chooses next "seed set" designating which fragment of the solution we will

[1] Actually, we need the volume of $K \cap \mathrm{proj}_U^{-1}(y)$ to be at least some small ε

[2] We can allow randomization also, but we stick to the deterministic case for simplicity, since all the seed selection procedures used by the known algorithms can be derandomized.

read based on current seeds S and $\textbf{FEASIBLE}_S(y)$ is a *strong separation oracle* for a convex body K_S representing the induced solutions on seeds S.

At the end, final seeds and induced solution are fed into another application specific rounding procedure.

1. Let $x \in \mathbb{R}^N$ be a vector representing an optimal solution for some convex optimization problem, $x \in K_N$.
2. Let $S(0)$ be the initial solution fragment and $y(0) \leftarrow x_{S(0)}$ be the induced solution.
3. For $i \leftarrow 0$ to ℓ:
   (a) Fail if $\textbf{FEASIBLE}_{S(i)}(y(i))$ asserts infeasible (i.e. $y(i) \notin K_{S(i)}$).
   (b) If $i < \ell$, read next part of solution: $S(i+1) \leftarrow \textbf{SEED}_{S(i)}(y(i))$ and $y(i+1) \leftarrow x_{S(i+1)}$.
4. Perform rounding using $S(\ell)$ and $y(\ell)$.

Suppose $|S(\ell)| \ll |N|$ – the algorithm reads only a negligible portion of the full solution. Then can we find an equivalent rounding algorithm which runs in time $\text{poly}(|S(\ell)|)$ as opposed to $\text{poly}(N)$? Claim 5.2 shows this can be expected:

Claim 5.2. *Above rounding algorithm can not distinguish between the following two cases, i.e. any properties satisfied by the output assuming 1 still holds under a weaker condition, 2:*

1. *There exists a feasible solution $x \in \mathbb{R}^N$, i.e. $\textbf{FEASIBLE}_N(x)$ asserts feasible.*
2. *For all $i \in \{0, \ldots, \ell\}$:*
   - *$y(i) \in K_{S(i)}$: $\textbf{FEASIBLE}_{S(i)}(y(i))$ asserts feasible,*
   - *$S(i+1) = \textbf{SEED}_{S(i)}(y(i))$ if $i < \ell$,*
   - *$y(i+1)_{S(i)} = y(i)$ if $i < \ell$.*

Using this insight, we first consider a simple case and give an algorithm whose running time depends on $|S(\ell)|$ instead of $|N|$.

5.2.1 An Algorithm for a Simple Case

Suppose that **SEED** procedure does not depend on y. Then the above conditions can easily be expressed as a convex problem of size $|S(\ell)|$, which is much smaller than the original problem. Then we can solve this convex problem using standard ellipsoid procedure and execute the above procedure on this solution instead.

5.2.2 Our Algorithm

Unfortunately for all algorithms we consider in this thesis, the procedure **SEED** heavily depends on y. In particular, at the i^{th} level, $0 \leq i < \ell$, we are trying to solve the following induced problem on $S(i+1)$. Given $y(i) \in K_{S(i)}$:

$$\begin{aligned}
\text{Find} \quad & y(i+1) \\
\text{st} \quad & y(i+1)_{S(i)} = y(i),\ y(i+1) \in K_{S(i+1)}; \\
& \exists y(i+2) \in K_{S(i+1)} : y(i+2)_{S(i+1)} = y(i+1) \\
& \qquad\qquad \text{where } S(i+2) = \mathbf{SEED}_{S(i+1)}(y(i+1)); \\
& \quad \vdots \\
& \exists y(\ell) \in K_{S(\ell)} : y(\ell)_{S(\ell-1)} = y(\ell-1) \\
& \qquad\qquad \text{where } S(\ell) = \mathbf{SEED}_{S(\ell-1)}(y(\ell-1)).
\end{aligned} \tag{5.1}$$

Observe that if we can construct a weak separation oracle for eq. (5.1) at $(i+1)^{th}$ level, then we can combine it with ellipsoid algorithm to solve the problem at i^{th} level also. Thus if we can convert this ellipsoid algorithm to a weak separation oracle, then we can call these separation oracles recursively starting from 0^{th} level all the way down to ℓ^{th} level:

Recursive Separation Oracle. (Template for i^{th} level)

1. Given $S(i)$ and $y(i)$, if $\mathbf{FEASIBLE}_{S(i)}(y(i))$ asserts infeasible and returns c, then assert infeasible and return c (to the $(i-1)^{th}$ level).
2. If $i = \ell$, then return the solution $y(\ell)$.
3. Let $S(i+1) \leftarrow \mathbf{SEED}_{S(i)}(y(i))$.
4. Use ellipsoid method to find $y(i+1)$ such that $y(i+1)_j = y(i)_j$ for all $j \in S(i)$ with separation oracle being a recursive call for the $(i+1)^{th}$ level (which takes inputs $S(i+1)$ and $y(i+1)$).
5. If ellipsoid method fails to find such solution $y(i+1)$, return a separating hyperplane.

The key question now is how one might implement (the currently vague) step 5. Let us inspect a simple option, and see what goes wrong with it.

Return an arbitrary hyperplane seen so far. Any inequality returned by the recursive separation oracle call is a valid separating hyperplane, so consider returning an arbitrary one. What goes wrong in this case? The problem is that the running time now might be as large as polynomial in $|N|$. To see this, suppose that $\mathbf{FEASIBLE}_{S(\ell)}(y(\ell))$ returned an inequality on support $S(\ell)$. Then the parent ellipsoid procedure needs to keep track of the additional variables from this particular

$S(\ell)$, call it $\tilde{S}$. At some later stage, the algorithm may backtrack and change an earlier seed set, say $S(\ell-5)$, which will need to a new $S(\ell)$. But the algorithm would still need to keep the values of variables from the $\tilde{S}$, the old value of $S(\ell)$. Continuing in this fashion, the set of variables the algorithm has to track might end up being N, which is equivalent to constructing the whole solution on $\mathbb{R}^N$!

This attempt has not been futile though, as it shows what kind of hyperplanes we need:

Any hyperplane returned by step 5 at $(i+1)^{st}$ level should have support $S(i)$. (5.2)

We outline our proposed solution in the next section.

5.2.3 Our Contribution: A Separation Oracle with Restricted Support

Our solution to 5.2 is based on a new ellipsoid algorithm for finding separating constraints with restricted support. Specifically, the main technical contribution of this paper is Algorithm 5.1 with the following guarantee: Given a feasibility problem of the form

$$\text{Find } y \in \mathbb{R}^n \text{ subject to } \Pi y = y_0,\, y \in \operatorname{int}(K),$$

where Π is a projection matrix, $y_0 \in \operatorname{span}(\Pi) \subseteq \mathbb{R}^n$; along with separation oracle for convex body K; it either finds feasible y or asserts that the problem is infeasible and outputs a separating hyperplane $c \in \operatorname{span}(\Pi)$. This algorithm coupled with the recursive separation oracle meets both our correctness and running time requirements. In particular, the running time instead of being the trivial bound of $|N|^{O(1)}$ will be roughly $|S(\ell)|^{O(\ell)}$. Assuming the exponential-time hypothesis, the exponential dependence on the number of seed selection stages ℓ cannot be avoided (a sub-exponential dependence would lead to a $f(k)n^{o(k)}$ time algorithm to decide if an n-vertex graph has a k-clique).

Remark 5.3. *Our algorithm can be thought as a weak separation oracle for eq. (5.1) at level i given a weak separation oracle for level $i+1$. When each convex body $K_{S(2)}, \ldots, K_{S(\ell)}$ is guaranteed to be a polytope, such as Sherali-Adams Hierarchy, it is known that one can obtain a strong separation oracle at level i by using only using a strong separation oracle at level $i+1$ ([see Grötschel et al., 1993, Corollary 6.5.13]). However in the case of semi-definite programming, it is an open question [see Porkolab and Khachiyan, 1997, for example] whether if one can obtain a strong separation oracle from another strong separation oracle in polynomial time.*

5.3 Preliminaries

In this section, we give relevant background on convex geometry and ellipsoid method which is used heavily in this chapter.

The main crux of our algorithm relies on an ellipsoid solver method which can also return a certificate of infeasibility.

5.3.1 Convex Geometry

Notation 5.4 (Projection). *We will use $\Pi \in \mathbb{S}_+^{[n]}$ to denote a projection matrix representing some linear subspace $\mathrm{span}(\Pi) \subseteq \mathbb{R}^{[n]}$ and $\Pi^\perp$ to denote the projection matrix onto null space of Π, i.e. $\Pi^\perp = \mathrm{identity} - \Pi$.*

Given vector $y_0 \in \mathbb{R}^{[n]}$, we will use $y_0 \in \Pi$ if y_0 is in the span of Π, i.e. $\Pi y_0 = y_0$ and we will use $\Pi^{-1}(y_0)$ to denote the following set of vectors:

$$\Pi^{-1}(y_0) \stackrel{\text{def}}{=} \left\{ y \in \mathbb{R}^{[n]} \middle| \Pi^\perp(y - y_0) = 0 \right\}.$$

Notation 5.5 (Polytopes). *Given matrix $A \in \mathbb{R}^{m \times n}$ and vector $b \in \mathbb{R}^m$, let*

$$\mathrm{poly}(A, b) \stackrel{\text{def}}{=} \left\{ x \in \mathbb{R}^{[n]} | Ax \leq b \right\}.$$

Lemma 5.6. *Given a polytope $P = \mathrm{poly}(A, b)$, for any positive real $\varepsilon > 0$,*

$$\mathbb{B}(P, -\varepsilon) = \mathrm{poly}(A, b - \varepsilon\sqrt{\mathrm{diag}(AA^T)}).$$

Here $\sqrt{\mathrm{diag}(AA^T)}$ denotes the vector whose i^{th} coordinate is equal to Euclidean norm of i^{th} row of A.

Proof. [See Grötschel et al., 1993, Lemma 3.2.35] □

5.3.2 Ellipsoid Method

Definition 5.7 (Separation Oracle). *Given a convex body $K \subseteq \mathbb{R}^{[n]}$, $\textbf{\textit{SEP}}_\delta(y)$ is a separation oracle for K if the following holds. On inputs a rational vector $y \in \mathbb{Q}^{[n]}$ and rational number $\delta > 0$, $\textbf{\textit{SEP}}_\delta(y)$ asserts feasible if $y \in K$. Otherwise, if $y \notin K$, it returns c such that $\|c\|_\infty = 1$ and*

$$\forall x \in K : \langle c, x \rangle \leq \langle c, y \rangle + \delta.$$

We will use $T(\textbf{\textit{SEP}}_\delta)$ to denote the worst case running time of $\textbf{\textit{SEP}}_\delta$.

Theorem 5.8 ([Grötschel et al., 1993, Central-Cut Ellipsoid Method]). *There exists an algorithm, called the central-cut ellipsoid method,*

$$\textbf{\textit{CCUT-E}}(\textbf{\textit{SEP}}_\delta, \Pi, y_0, \varepsilon_0)$$

that solves the following problem. Given a projection matrix $\Pi \in \mathbb{S}_+^{[n]}$ of rank m, vector $y^0 \in \Pi$, a convex body $K \subseteq [-\Delta, \Delta]^{[n]}$ for some positive Δ with $\textbf{\textit{SEP}}_\delta$ (see Definition 5.7) and rational number $\varepsilon_0 > 0$, it runs in time

$$|\log \Delta| N \left[\text{poly}(n) + T(\textbf{\textit{SEP}}_{2^{-N}})\right] \text{ where } N \leq 6(n-m)(|\log \varepsilon_0| + (n-m));$$

after which it outputs:

1. *Either a vector $a \in \mathbb{Q}^{[n]}$ such that $a \in K \cap \Pi^{-1}(y^0)$;*
2. *Or a polytope of the form $P = \text{poly}(C, d)$, where $C \in \mathbb{Q}^{[N] \times [n]}, d \in \mathbb{Q}^{[N]}$ with $K \subseteq P$ and $\text{vol}_{n-m}(P \cap \Pi^{-1}(y^0)) < \varepsilon_0$.*

Proof. Such algorithm can be obtained by trivial modifications to the central-cut ellipsoid algorithm of Grötschel et al. [1993], which we outline here. Handling the constraint $\Pi a = y^0$ can be done by projecting the covariance matrix of ellipsoid onto $\Pi^{-1}(y^0)$. At k^{th} iteration, for all k, we add hyperplanes returned by $\textbf{SEP}_\delta$ to P.

Algorithm terminates with a feasible $a \in \mathbb{Q}^{[n]}$ with $\Pi a = y^0$ only if $\textbf{SEP}_\delta(a)$ asserts feasible for some $\delta \leq \varepsilon_0$, in which case $a \in K$. Otherwise, when the maximum number of iterations is reached we simply return. □

5.4 Finding Separating Hyperplanes on a Subspace

We now describe our main technical contribution: An ellipsoid algorithm which can output a certificate of infeasibility **on a restricted subspace** using only the separation oracle $\textbf{SEP}_\delta$ as in Definition 5.7. The procedure uses the central-cut ellipsoid method Grötschel et al. [1993] as a sub-routine. The main technical ingredient of our algorithm is Theorem 5.12, which is stated and proven in Section 5.4.1: It allows us to express this as another convex programming problem in terms of the "history" of constraints returned by separation oracle. Finally in Section 5.4.2 we present our ellipsoid algorithm, bound its running time and prove its correctness.

5.4.1 An Equivalent Convex Problem

We first state some useful propositions. Recall our goal: Given convex body K, a subspace Π and $y_0 \in \Pi$, we have a polytope P separating various points $\{y\} \subset \Pi^{-1}(y_0)$ from K. We want to compute a separating hyperplane on Π. Our approach is formulated in Lemma 5.11, see also Figure 5.1.

Proposition 5.9. *Given $K \subseteq \mathbb{R}^{[n]}$, a projection matrix $\Pi \in \mathbb{S}_+^{[n]}$ with $\operatorname{rank}(\Pi) = m$, and $y \in \mathbb{R}^{[n]}$ the following holds: For any $\delta > \operatorname{vol}_{n-m}^{-1}(\Pi^{-1}(y) \cap K)$,*

$$\Pi^{-1}(y) \cap \mathbb{B}(K, -\delta) = \emptyset.$$

Proof. If $\exists x \in \Pi^{-1}(y) \cap \mathbb{B}(K, -\delta)$, then Observation 2.31 implies $\mathbb{B}(x, \delta) \subseteq K$. In particular,

$$\mathbb{B}(x, \delta) \cap \Pi^{-1}(y) \subseteq \Pi^{-1}(y) \cap K \implies \operatorname{vol}_m(\Pi^{-1}(y) \cap K) \geq \operatorname{vol}_m(\Pi^{-1}(y) \cap \mathbb{B}(x, \delta)).$$

Finally since $x \in \Pi^{-1}(y)$, $\mathbb{B}(x, \delta) \cap \Pi^{-1}(y)$ is an m-dimensional ball of radius δ, whose volume is $\operatorname{vol}_m(\delta) > \operatorname{vol}_m(\Pi^{-1}(y) \cap K)$. Hence

$$\operatorname{vol}_m(\Pi^{-1}(y) \cap K) \geq \operatorname{vol}_m\left(\mathbb{B}(x, \delta) \cap \Pi^{-1}(y)\right) > \operatorname{vol}_m(\Pi^{-1}(y) \cap K),$$

which is a contradiction. □

The following is a quantitative version of above, showing that points further interior in K have far off projections from Πy_0.

Lemma 5.10. *Given convex body $K \subseteq \mathbb{R}^{[n]}$, a projection matrix $\Pi \in \mathbb{S}_+^{[n]}$ with $\operatorname{rank}(\Pi) = m$, vector $y_0 \in \mathbb{R}^{[n]}$ and positive real $\delta > \operatorname{vol}_{n-m}^{-1}(\Pi^{-1}(y_0) \cap K)$,*

$$\text{for all } y \in \mathbb{B}(K, -2\delta),\ \|\Pi(y - y_0)\| \geq \delta.$$

Proof. For the sake of contradiction, assume there exists $y \in \mathbb{B}(K, -2\delta)$ such that $\|\Pi(y - y_0)\| < \delta$. Consequently $\Pi\mathbb{B}(y, \delta)$, which is a sphere of radius δ on span of Π, contains Πy_0. In other words,

$$\emptyset \neq \Pi^{-1}(y_0) \cap \mathbb{B}(y, \delta). \tag{5.3}$$

Since $y \in \mathbb{B}(K, -2\delta)$, by convexity of K, we can repeatedly apply Observation 2.31 to show that

$$\begin{aligned} y \in &\mathbb{B}(K, -2\delta) = \mathbb{B}(\mathbb{B}(K, -\delta), -\delta), \\ \mathbb{B}(y, \delta) \subseteq &\mathbb{B}(\mathbb{B}(\mathbb{B}(K, -\delta), -\delta), \delta) \subseteq \mathbb{B}(K, -\delta). \end{aligned}$$

Substituting this into eq. (5.3), we have $\emptyset \neq \Pi^{-1}(y_0) \cap \mathbb{B}(y, \delta) \subseteq \Pi^{-1}(y_0) \cap \mathbb{B}(K, -\delta)$ which contradicts Proposition 5.9 for our choice of δ. □

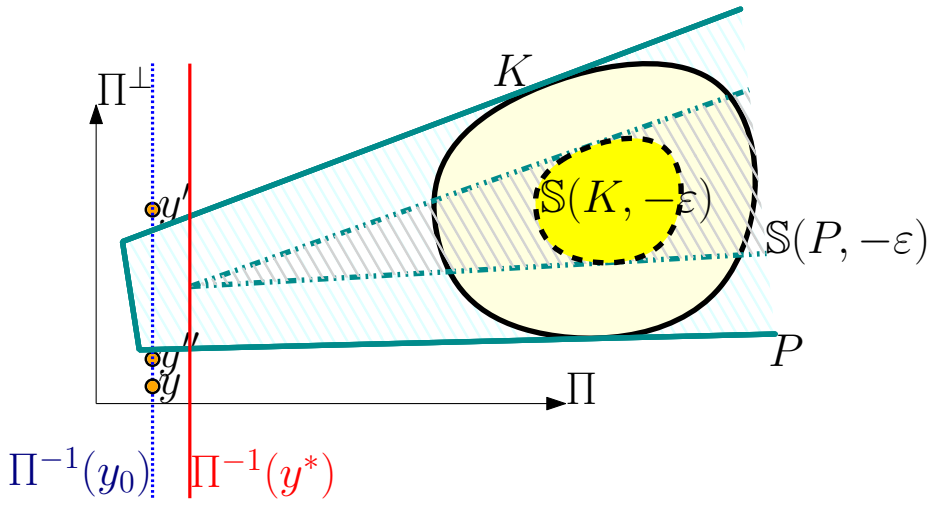

Figure 5.1: We want to find a hyperplane parallel to $\Pi^{\perp}$ separating $\Pi^{-1}(y_0)$ and K, using only the inequalities returned by separation oracle, polytope P. The optimal solution of Lemma 5.11 is given by y^* with corresponding hyperplane $\Pi^{-1}(y^*)$.

Having shown that there is a δ-neighborhood of Πy_0 disjoint from interior of K whenever their intersection has small volume, we can immediately use Minkowski's Separating Hyperplane Theorem to infer the existence of such hyperplane. In fact, any hyperplane perpendicular to the line from y_0 to the closest point in K has this property. We formalize this below.

Lemma 5.11. *Given convex body $K \subseteq \mathbb{R}^{[n]}$, a projection matrix $\Pi \in \mathbb{S}_+^{[n]}$ with $\operatorname{rank}(\Pi) = m$, vector $y_0 \in \mathbb{R}^{[n]}$ and positive real $\delta > \operatorname{vol}_{n-m}^{-1}(\Pi^{-1}(y_0) \cap K)$, the hyperplane perpendicular to the projection of direction from y to closest point in the interior of ΠK separates y_0 and interior of ΠK:*
Formally any optimal solution y^* to the following eq. (5.4):

$$\begin{array}{ll} \textit{Minimize} & \|\Pi(y - y_0)\|^2 \\ \textit{subject to} & y \in \mathbb{B}(K, -2\delta), \end{array} \tag{5.4}$$

satisfies eq. (5.5):

$$\min_{x \in \mathbb{B}(K,-2\delta)} \langle \Pi(y^* - y_0), x - y_0 \rangle \geq \|\Pi(y^* - y_0)\|^2. \tag{5.5}$$

Proof. By contradiction. Assume for optimal solution y^*, there exists $x \in \mathbb{B}(K, -2\delta)$ such that

$$\langle \Pi(y^* - y_0), x - y_0 \rangle < \|\Pi(y^* - y_0)\|^2 = \langle \Pi(y^* - y_0), \Pi(y^* - y_0) \rangle = \langle \Pi(y^* - y_0), y^* - y_0 \rangle .$$

Therefore

$$\langle \Pi(y^* - y_0), x - y^* \rangle < 0 \tag{5.6}$$

For some $\theta \in (0, 1]$ to be chosen later, consider $y(\theta) \leftarrow (1-\theta) \cdot y^* + \theta \cdot x$, which is always feasible for eq. (5.4) as $x \in \mathbb{B}(K, -2\delta)$ and $\mathbb{B}(K, -2\delta)$ is convex. Then:

$$\begin{aligned}
\frac{1}{2} \frac{\partial \|\Pi(y(\theta) - y_0)\|^2}{\partial \theta}\bigg|_{\theta \to 0^+} &= \left\langle \Pi(y(\theta) - y_0), \Pi \frac{\partial (y(\theta) - y_0)}{\partial \theta} \right\rangle \bigg|_{\theta \to 0^+} \\
&= \left\langle \Pi(y(\theta) - y_0), \frac{\partial (y(\theta) - y_0)}{\partial \theta} \right\rangle \bigg|_{\theta \to 0^+} \\
&= \langle \Pi(y(\theta) - y_0), x - y^* \rangle|_{\theta \to 0^+} = \langle \Pi(y(0) - y_0), x - y^* \rangle \\
&= \langle \Pi(y^* - y_0), x - y^* \rangle < 0
\end{aligned}$$

where we used eq. (5.6) at the last step. We arrive at a contradiction by noting that above inequality implies existence of $\theta^* \in (0, 1]$ such that

$$\|\Pi(y(\theta^*) - y_0)\|^2 < \|\Pi(y^* - y_0)\|^2, \; y(\theta^*) \in \mathbb{B}(K, -2\delta). \qquad \square$$

Given Lemma 5.11, we can choose our separating hyperplane c as $c = -\frac{\Pi(y^* - y_0)}{\|\Pi(y^* - y_0)\|_\infty}$. But this does not quite work for two reasons:

1. Hyperplane c only separates "strict interior" of K as it is, whereas we need to separate K itself.
2. Depending on K, it might not be possible to represent optimal c using polynomially many bits, thus we need to account for near optimal solutions.

We now show how to overcome these problems.

Theorem 5.12. *Given convex body $K \subseteq [-\Delta, \Delta]^n$ for some $\Delta > 0$, a projection matrix $\Pi \in \mathbb{S}_+^{[n]}$ with $\operatorname{rank}(\Pi) = m$, a vector $y_0 \in [-\Delta, \Delta]^n$, for any positive real $\delta > 0$ with $\delta > \operatorname{vol}_{n-m}^{-1}(\Pi^{-1}(y_0) \cap K)$, the following holds: If y' is an $\frac{\delta^2}{2\Delta\sqrt{m}}$- approximate solution to eq. (5.7)*

$$\begin{aligned}
&\textit{Minimize} && \|\Pi(y - y_0)\|^2 \\
&\textit{subject to} && y \in \mathbb{B}\left(K, -\left(2 + \tfrac{\delta}{2\sqrt{m}\Delta}\right) \cdot \delta\right)
\end{aligned} \tag{5.7}$$

then $\Pi(y' - y_0) \neq 0$ and for c being

$$c \stackrel{\text{def}}{=} -\frac{\Pi(y' - y_0)}{\|\Pi(y' - y_0)\|_\infty} \implies \forall x \in K : \langle c, x \rangle \leq \langle c, y_0 \rangle + 2\delta\sqrt{m}. \tag{5.8}$$

Proof. Before we begin, we set $\varepsilon \stackrel{\text{def}}{=} \frac{\delta}{2\sqrt{m}\Delta}$. Let y^* be an optimal solution of eq. (5.7) with $\|y^* - y'\| \le \varepsilon\delta$. Since $y^* \in \mathbb{B}(K, -(2+\varepsilon)\delta)$, we have $\mathbb{B}(K, -2\delta) \supseteq \mathbb{B}(y^*, \varepsilon\delta) \ni y'$. By Lemma 5.10, this implies $\|\Pi(y' - y_0)\| \ge \delta$, proving $\Pi(y' - y_0) \neq 0$. For any $x \in K$, we can decompose x as $x = x' + z$ for some $x' \in \mathbb{B}(K, -2\delta)$ and $\|z\|_2 \le 2\delta$. Then:

$$\begin{aligned}
\langle \Pi(y' - y_0), x - y_0 \rangle =& \langle \Pi(y' - y_0), x' - y_0 \rangle + \langle \Pi(y' - y_0), z \rangle \\
\ge& \langle \Pi(y' - y^*), x' - y_0 \rangle + \langle \Pi(y^* - y_0), x' - y_0 \rangle - \|z\| \cdot \|\Pi(y' - y_0)\| \\
\ge& -\underbrace{\|\Pi(y' - y^*)\|}_{\le \varepsilon\delta} \underbrace{\|\Pi(x' - y_0)\|}_{\le 2\Delta\sqrt{m}} + \underbrace{\delta^2}_{\text{by Lemma 5.11}} - 2\delta\|\Pi(y' - y_0)\| \\
\ge& -2\delta\|\Pi(y' - y_0)\| \text{ (by the choice of } \varepsilon = \frac{\delta}{2\Delta\sqrt{m}}) \\
\ge& -2\delta\sqrt{m}\|\Pi(y' - y_0)\|_\infty.
\end{aligned}$$

Since $c = -\frac{\Pi(y' - y_0)}{\|\Pi(y' - y_0)\|_\infty}$, we have $\langle c, x \rangle \le \langle c, y_0 \rangle + 2\delta\sqrt{m}$ for any $x \in K$. □

5.4.2 Ellipsoid Algorithm with Certificate of Infeasibility

Our solver is given in Algorithm 5.1. The proof of the following theorem follows by combining various ingredients we have so far, especially Theorem 5.8 and Theorem 5.12.

Theorem 5.13 (Main technical tool)**.** *Algorithm 5.1 runs in time* $N \cdot T(\textbf{\textit{SEP}}_{2^{-N}}) + \operatorname{poly}(n) \log^2 \frac{1}{\varepsilon_0}$ *where* $N = O\left((\#\textit{ of free variables})^2 \log \frac{\#\textit{ of fixed variables}}{\varepsilon_0}\right) = O\left((n-m)^2 \log \frac{m}{\varepsilon_0}\right)$, *and provides the following guarantee: If* $\operatorname{vol}_{n-m}^{-1}(K \cap \Pi^{-1}(y_0)) > \frac{\varepsilon_0}{2\sqrt{m}}$ *then it outputs* $y \in K \cap \Pi^{-1}(y_0)$.

Proof of Running Time. By Theorem 5.8, step 1 finishes in time $N(\operatorname{poly}(n) + T(\textbf{SEP}_{2^{-N}}))$ with $N = O((n-m)^2 + (n-m)\log 1/\varepsilon)$, where $\log 1/\varepsilon = \log 1/\operatorname{vol}_{n-m}(\varepsilon_0/2\sqrt{m}) \le O\left((n-m)\log \frac{m}{\varepsilon_0}\right)$ so $N = O\left((n-m)^2 \log \frac{m}{\varepsilon_0}\right)$.

For step 4, we can implement a simple separation oracle which runs in time $O(Nn)$. The regular ellipsoid method requires $O(n^2 + n\log(1/\varepsilon'\delta))$ iterations to reach an accuracy of $\varepsilon'\delta$. Each iteration takes time $\operatorname{poly}(n)$ in addition to separation oracle, therefore the total running time of fourth step is bounded by $N \operatorname{poly}(n) \log(1/\varepsilon_0) = \operatorname{poly}(n) \log^2 \frac{1}{\varepsilon_0}$. Hence the claim follows. □

Algorithm 5.1 CERTIFY-E($\mathbf{SEP}_\delta, \Pi, y_0, \varepsilon_0$): Ellipsoid method with certificate of infeasibility.

Input: • Convex body $K \subseteq [0,1]^n$ and separation oracle $\mathbf{SEP}_\delta$ as in Definition 5.7,
• Projection matrix $\Pi \in \mathbb{S}_+^{[n]}$ with $\operatorname{rank}(\Pi) = m$, $y_0 \in \mathbb{R}^{[n]}$ and positive real $0 < \varepsilon_0 < 1$.

Output: • Either a vector $y \in \mathbb{Q}^n$ st $y \in K \cap \Pi^{-1}(y_0)$.
• Or $c \in \Pi$ with $\|c\|_\infty = 1$ and $\forall x \in K : \langle c, x\rangle \leq \varepsilon_0 + \langle c, y_0\rangle$.

Procedure: 1. Run **CCUT-E**($\mathbf{SEP}_\delta, \Pi, y_0, \varepsilon$) where $\varepsilon \leftarrow \operatorname{vol}_{n-m}\left(\frac{\varepsilon_0}{2\sqrt{m}}\right)$.
2. If it returns $y \in K \cap \Pi^{-1}(y_0)$, then return y.
3. Else let $P = \operatorname{poly}(C, d)$ be the polytope it returns. Set $\delta \leftarrow \frac{\varepsilon_0}{2\sqrt{m}}$. $\varepsilon' \leftarrow \frac{\delta}{2\Delta\sqrt{m}}$ and
4. Solve eq. (5.9) using regular ellipsoid method to find an $\varepsilon'\delta$-approximate solution, $y^* \in \mathbb{Q}^n$:

$$\text{Minimize } \|\Pi(y - y_0)\|^2 \text{ subject to } Cy \leq d - (2 + \varepsilon')\delta\sqrt{\operatorname{diag}(C^T C)}. \tag{5.9}$$

5. Return $c \leftarrow -\frac{\Pi(y^* - y_0)}{\|\Pi(y^* - y_0)\|_\infty}$.

Proof of Correctness. There are two cases. If algorithm outputs y at step 2, by Theorem 5.8 $y \in \mathbb{Q}^n \cap \Pi^{-1}(y_0) \cap K$.

Now consider the other case. Then step 1 will output a polytope $P = \operatorname{poly}(C, d)$ such that $K \subseteq P$, whose volume is bounded by

$$\operatorname{vol}_{n-m}(P \cap \Pi^{-1}(y_0)) < \operatorname{vol}_{n-m}\left(\frac{\varepsilon_0}{2\sqrt{m}}\right).$$

The set of feasible solutions for eq. (5.9) is $\mathbb{B}(P, -(2 + \varepsilon')\delta)$ by Lemma 5.6. Since $\operatorname{vol}_{n-m}^{-1}(P \cap \Pi^{-1}(y_0)) < \delta := \frac{\varepsilon_0}{2\sqrt{m}}$, we can apply Theorem 5.12 and conclude that c as constructed in step 5 will have the following properties:

- $c \in \Pi$,
- $\|c\|_\infty = 1$,
- For all $x \in P$, $\langle c, x\rangle \leq \langle c, y_0\rangle + 2\sqrt{m}\delta$. To see this, note $K \subseteq P$, for all $x \in K$

means:

$$\langle c, x\rangle \leq \max_{x\in K}\langle c, x\rangle \leq \max_{x'\in P}\langle c, x'\rangle \leq \langle c, y_0\rangle + 2\sqrt{m}\delta = \langle c, y_0\rangle + \varepsilon_0.$$

This finishes the proof of correctness. □

5.5 Faster Solver for Local Rounding Algorithms

We return back to our motivating example. Assume we have n variables, and we want to find a discrete labeling $\mathbf{x} \in L^{[n]}$ of those from a set of labels L, under various constraints and objective. Suppose we "lifted" this problem into a higher dimension $\mathbb{R}^N$ where $|N| \gg n$, and obtained a family of increasingly tight convex relaxations defined over various subspaces of $\mathbb{R}^N$.

Formally, we have a set of subspaces $\{\Pi_S\}_{S\subseteq[N]}$, represented by their projection matrices and associated with subsets of $[N]$, and with each subspace Π_S, we have an associated convex body, $K_S \subseteq \Pi_S[0,1]^N$ with such that

$$\Pi_S \subseteq \Pi_T \implies \Pi_S K_T \subseteq K_S.$$

We are given functions **FEASIBLE**, **ROUND** and **SEED**, along with positive integers n, s such that:

- **FEASIBLE**$_S$: $\Pi_S\mathbb{Q}^N \to \{\text{feasible}, \Pi_S\mathbb{Q}^N\}$. On input $S \subseteq N$,$y \in \Pi_S Q^N$, it asserts feasible if $y \in K_S$ or returns $c \in \Pi_S\mathbb{Q}^N : \|c\|_\infty = 1$ such that[3] $\forall x \in K_S : \langle c, x\rangle < \langle c, y\rangle$ in time $\text{poly}(\text{rank}(\Pi_S))$.

- **SEED**$_S$: $K_S \to 2^N$. Given $S \subseteq [N]$ and $y \in \Pi_S\mathbb{Q}^N$, it returns subset $S' \supseteq S$ such that $\frac{\text{rank}(\Pi_{S'})}{\text{rank}(\Pi_S)} \leq s$ when $S \neq \emptyset$, and $\text{rank}(\Pi_{S'}) \leq n$ when $S = \emptyset$. Its worst case running time is bounded by $\text{poly}(\text{rank}(\Pi_S))$ (or $\text{poly}(n)$ in the case of $S = \emptyset$).

- **ROUND**$_S$: $K_S \to L^{[n]}$. On inputs $S \subseteq N$ and $y \in K_S$, returns an approximation to the original problem in time $\text{poly}(\text{rank}(\Pi_S))$.

We now describe our main solver. Note that once the algorithm outputs $y^* \in K_{S(\ell)}$, the final output labeling will simply be **ROUND**$_{S(\ell)}(y^*)$.

[3]We can handle **FEASIBLE**$_S$ that only returns a weak separation oracle, but since in our application to SDPs we have access to a strong separation oracle, we assume this for simplicity.

Algorithm 5.2 Fast Solver (to fool local rounding algorithm)

Input: • Maximum number of iterations ℓ and positive real $\varepsilon_0 > 0$,
• n, r, $(K_S)_{S\subseteq[N]}$ with separation oracle **FEASIBLE**, **SEED**, $\Pi_\emptyset$ and $y(0) \in K_\emptyset\mathbb{Q}^N$ all as described in Section 5.5.

Output: • Either asserts $\operatorname{vol} K \le \varepsilon_0$,
• Or outputs $y^* \in K_{S(\ell)}$ and $S(0), \dots, S(\ell)$ st for all i: 1. $\Pi_{S(i)}y^* \in K_{S(i)}$; 2. $S(i+1) = \mathbf{SEED}_{S(i)}(y^*)$.

Procedure: 1. Initialize global variables $S(1), \dots, S(\ell)$ representing seed sets and global sparse vector $y^* \in \mathbb{Q}^{[N]}$ representing the final solution (it will be in span of $\Pi_{S(\ell)}$).
2. Set $S(0) \leftarrow \{\emptyset\}$.
3. Run **CCUT-E**($\mathbf{SEP}_{S(0),\delta}, 0, 0, \varepsilon_0$) (see Theorem 5.8) where **SEP** is given in Algorithm 5.3.
4. If it asserts feasible, output $S(0), \dots, S(\ell)$ and y^*.
5. Else assert $\operatorname{vol} K \le \varepsilon_0$.

Theorem 5.14. *Algorithm 5.2 runs in time* $\left[s^\ell n \log(1/\varepsilon_0)\right]^{O(\ell)}$ *(compare this with the straightforward algorithm which runs in time* $N^{O(1)} \log(1/\varepsilon_0)$*). Furthermore there is no algorithm which runs in time* $n^{o(\ell)}$ *assuming Exponential Time Hypothesis.*
Provided that $\operatorname{vol} K > \varepsilon_0$*, it outputs* $y^* \in K_{S(\ell)}$ *and* $S(0), \dots, S(\ell)$ *st for all* i*:*

$$\Pi_{S(i)}y^* \in K_{S(i)}, \tag{5.10}$$
$$S(i+1) = \boldsymbol{SEED}_{S(i)}(y^*). \tag{5.11}$$

Otherwise it asserts $\operatorname{vol} K \le \varepsilon_0$.

Proof of Correctness. First we assume correctness of Algorithm 5.3 and prove correctness of Algorithm 5.2. There are only two cases:

1. Algorithm 5.2 returns $y^* \in K_{S(\ell)}$ only if **CCUT-E**$_{S(1)}$ returns a feasible solution. For such y^*, by Theorem 5.8, $\mathbf{SEP}_{S(1),\varepsilon_0}(y^*)$ asserts feasible. By correctness of $\mathbf{SEP}_{S(1),\varepsilon_0}(y^*)$, y^* satisfies all claims.

2. Else algorithm asserts $\operatorname{vol}_{[N]} K \le \varepsilon_0$ which means **CCUT-E** asserted $\varepsilon_0 \ge \operatorname{vol}_{S(0)} \mathbb{B}(K_{S(1)}, \varepsilon_0) \ge \operatorname{vol}_{S(0)} K_{S(0)}$. We know that $\Pi_{S(0)}K \subseteq K_{S(0)}$ and $K \subseteq [0,1]^{[N]}$ therefore $\operatorname{vol}_{[N]}(K) \le \operatorname{vol} \Pi_{S(0)}K \le \operatorname{vol}_{S(0)} K_{S(0)} \le \varepsilon_0$.

Algorithm 5.3 $\mathbf{SEP}_{S(i),\varepsilon_0}(y)$:Separation Oracle.

Input: • Positive real $\varepsilon_0 > 0$, current iteration i, current seeds $S(i)$, vector $y \in \mathbb{Q}^{S(i)}$.

Output: • Either asserts feasible, and sets values of global variables $S(i+1), \ldots, S(\ell)$ along with y^* so that:

1. $\Pi_{S(j)} y^* \in K_{S(j)}$ for all $j : i \leq j \leq \ell$,
2. $S(j+1) = \mathbf{SEED}_{S(j)}(y^*)$ for all $j : i \leq j \leq \ell - 1$.

• Or returns $c \in \Pi_{S(i)}$ with $\|c\|_\infty = 1$ such that $\forall x \in K_{S(i)} : \langle c, x - y \rangle < \varepsilon_0$.

Procedure: 1. If $\mathbf{FEASIBLE}_{S(i)}(y)$ returns $c \in \Pi_{S(i)}$, return c.

2. Else if $i \geq \ell$, set $y^* \leftarrow y$. Assert feasible and return.
3. $S(i+1) \leftarrow \mathbf{SEED}_{\emptyset}(y)$.
4. Run $\mathbf{CERTIFY\text{-}E}(\mathbf{SEP}_{\Pi_{S(i+1)},\delta}, \Pi_{S(i)}, y, \varepsilon_0)$ (see Algorithm 5.1).
5. If it returns c, return c.
6. Else assert feasible.

Now we will prove correctness of Algorithm 5.3 inductively starting from $i = \ell$. For each i, in order to prove inductive step, we need to consider in which one of the following steps $\mathbf{SEP}_{S(i),\varepsilon_0}$ returned:

1. Step 1: Follows from definition of **FEASIBLE**.
2. Steps 2 and 6: By construction of $S(j)$'s and inductive hypothesis, $S(j+1) = \mathbf{SEED}_{S(j)}(y^*)$ holds for all $j \geq i$.

   We will prove that $\mathbf{FEASIBLE}_{S(j)}(y^*)$ asserts feasible for all $j \geq i$. For $j > i$, this immediately follows from inductive hypothesis. For $j = i$, at Step 1 $\mathbf{FEASIBLE}_{S(i)}(y^*)$ asserted feasible. Thus $\Pi_{S(j)} y^* \in K_{S(j)}$ for all j.
3. Step 5: It returns c, only if **CERTIFY-E** at step 4 outputs c, correctness of which follows from Theorem 5.13. □

Proof of Running Time. If we let $n_i \leftarrow n s^{i-1}$ for $i \geq 1$, we can see that $\operatorname{rank}(\Pi_{S_i}) \leq n_i$ at i^{th} iteration. Hence

$$T(\mathbf{FEASIBLE}_{S(i)}) + T(\mathbf{SEED}_{S(i)}) = (s n_i)^{O(1)} = n_{i+1}^{O(1)}.$$

If we use T_i to denote the maximum of $T(\mathbf{SEP}_{S(i),\varepsilon_0}(y))$ over all possible $S(i)$ and y's, with $T_0 = T(\text{main})$; then $T_\ell = n_\ell^{O(1)} = (r^\ell n)^{O(1)}$ and for any $i < \ell$:

$$\begin{aligned} T_i \leq & O(n_{i+1}^2 \log n_i/\varepsilon_0) T_{i+1} + n_{i+1}^{O(1)} = (n_{i+1})^{O(1)} \log(1/\varepsilon_0) T_{i+1} = s^{O(i)} n^{O(1)} \log 1/\varepsilon_0 \cdot T_{i+1} \\ T_0 = & s^{O(\ell^2)} n^{O(\ell)} \log^\ell(1/\varepsilon_0). \end{aligned}$$

□

Proof of ETH Hardness. Consider k-clique problem, which can not be computed in time $f(k)n^{o(k)}$ under ETH hypothesis [Lokshtanov et al., 2011]. Moreover it is easy to see that k rounds of Lasserre Hierarchy relaxation captures this problem [Laurent, 2003], which can be found using a seed selection procedure with $\ell = k$ levels. □

5.6 Separation Oracle for Lasserre Hierarchy

In this section, we present a separation oracle for Lasserre Hierarchy relaxation as introduced in Chapter 3 for solving the integer programming problem given in eq. (1.1), which is re-produced below in terms of k-labeling for convenience. Recall that $\mathbf{x}_{(u,i)}$ is the indicator variable for labeling variable $u \in [n]$ with label $i \in [k]$:

$$\begin{array}{lll} \text{Minimize} & \mathsf{q}(\mathbf{x}) & \\ \text{subject to} & \mathsf{p}(\mathbf{x}) \geq 0 & \text{for all } \mathsf{p} \in \mathcal{P}, \\ & \sum_i \mathbf{x}_{(u,i)} = 1 & \text{for all } u \in [n], \\ & \mathbf{x} \in \{0,1\}^{V \times [k]}. & \end{array} \tag{5.12}$$

Given such k-labeling problem, let's cast its r'^{th} round Lasserre relaxation in our framework:

- The set of labels is $L = \{0,1\}^{[k]}$, corresponding to the indicator vector over all k labels per each variable,
- Lifted space N is $\binom{V \times [k]}{\leq r'}$, the subsets of $V \times [k]$ of size at most $r' + d$,
- For any subset $S \subseteq V$, we define Π_S as the projection matrix onto $\mathbb{R}^{\text{ex}_k(S,d) \uplus \text{ex}_k(S,d)}$ where

  $$\text{ex}_k(S,d) \stackrel{\text{def}}{=} 2^{S \times [k]} \uplus (V \times [k])_{\leq d}$$

  so that

  $$[\Pi_S x]_T = \begin{cases} 1 & \text{if } T \in \text{ex}_k(S,d), \\ 0 & \text{else.} \end{cases}$$

The associated convex body, K_S, is defined as

$$K_S = \left\{ x \in \mathbb{R}^{\mathrm{ex}_k(S,d)} : \begin{array}{c} x_\emptyset = 1,\ \mathbf{M}_{\mathrm{ex}_k(S,d)}(x) \succeq 0, \\ \sum_i x_{u(i)} = x_\emptyset, \\ \mathbf{M}_{\mathrm{ex}_k(S,0)}(P * x) \succeq 0 \end{array} \right\}.$$

Before stating the **FEASIBLE** procedure, we need the following well known result:

Proposition 5.15. *Given a symmetric matrix $A \in \mathbb{S}^B$, there exists an algorithm which asserts if $A \succeq 0$ or returns $x \in \mathbb{Q}^B$ such that $y^T A y < 0$ in time at most polynomial in size of A.*

Then our **FEASIBLE**$_S(x)$ procedure is trivial given a problem of the form eq. (5.12): It asserts feasible if $\mathbf{M}_{\mathrm{ex}(S,d)}(x) \succeq 0$ and $\mathbf{M}_{\mathrm{ex}(S,0)}(\mathsf{p} * x) \succeq 0$ for all $\mathsf{p} \in \mathcal{P}$. Else it returns $y \in \mathbb{Q}^{\mathrm{ex}(S,2)}$ for which $y^T\, \mathbf{M}_{\mathrm{ex}(S,d)}(x) y < 0$ (or $y^T\, \mathbf{M}_{\mathrm{ex}(S,0)}(\mathsf{p} * x) y < 0$).

Chapter 6

Our Results

In this chapter, we list all our approximation algorithms given in this thesis including their approximation factor and running time along with pointers to the appropriate sections. For each problem, we will give the formal definition as well as a review of related work in detail at the respective chapter.

All the algorithms here are obtained by applying our main algorithm as given in Algorithm 5.2 to various rounding algorithms for Lasserre Hierarchy relaxations. For all these problems, our separation oracle is the same procedure as described in Section 5.6. The running times we obtained as well as approximation factors and other guarantees are summarized in Figure 6.1. The last two columns list the value of s (the factor by which $\mathrm{rank}(\Pi_S)$ increases in each step of seed selection) and ℓ (the number of iterations of seed selection) used by the rounding algorithm in each case. The parameter r refers to the index of the eigenvalue governing the approximation guarantee, and ε is a positive parameter.

The claimed running times follow from the $\approx s^{O(\ell^2)} n^{O(\ell)}$ runtime guaranteed by Theorem 5.14 for our solver (Algorithm 5.2). The rounding algorithm in each case runs within the same time. For problems marked with *, check the caption for required conditions.

The works Raghavendra and Tan [2012] and Arora and Ge [2011] use greedy seed selection, but these can be replaced by the above column selection procedure as well.

We conclude the chapter by listing two notable Lasserre based approximation algorithms for which we are not able to get a runtime improvement: the algorithm for independent sets in 3-uniform hyper-graphs Chlamtac and Singh [2008], and the algorithm for directed Steiner tree Rothvoß [2011]. This is because these algorithms are adaptive with a large number of stages ℓ in the rounding procedure.

Problem Name	Running Time	Rounding	Imbalance	s	ℓ
Minimum Uncut (Section 8.3)	$2^{O(r/\varepsilon^3)}n^{O(1/\varepsilon)}$	$\frac{1+\varepsilon}{\lambda_{n-r}}\eta$		$2^{O(r/\varepsilon)}$	$O(1/\varepsilon)$
k-Unique Games (Section 8.4)	$k^{O(r/\varepsilon)}n^{O(1)}$	$\frac{2+\varepsilon}{\lambda_r}\eta$		$k^{O(r/\varepsilon)}$	1
Minimum (Section 7.4.1)	$2^{O(r/\varepsilon^3)}n^{O(1/\varepsilon)}$	$\frac{1+\varepsilon}{\lambda_r}\eta$	$o(1)$	$2^{O(r/\varepsilon)}$	$O(1/\varepsilon)$
Bisection [1]	$2^{O(1/\varepsilon)}n^{O(1)}$	$O(\sqrt{\eta})$	$O(\varepsilon)$	$2^{O(1/\varepsilon)}$	1
Maximum (Section 8.3)	$2^{O(r/\varepsilon^3)}n^{O(1/\varepsilon)}$	$\frac{1+\varepsilon}{\lambda_{n-r}}\eta$	$o(1)$	$2^{O(r/\varepsilon)}$	$O(1/\varepsilon)$
Bisection Complement [1]	$2^{\mathrm{poly}(1/\varepsilon)}n^{O(1)}$	$O(\sqrt{\eta})$	$O(\varepsilon)$	$2^{\mathrm{poly}(1/\varepsilon)}$	1
Sparsest Cut (Section 9.4)*	$2^{O(r/\varepsilon)}n^{O(1)}$	$\frac{\eta}{\varepsilon}$		$2^{O(r/\varepsilon)}$	1
Independent (Section 7.5) *	$2^{O(r)}n^{O(1)}$	$\Omega(\eta)$		$2^{O(r)}$	$O(1)$
Set [2]	$2^{O(r)}n^{O(1)}$	$\frac{\eta}{16}$		$2^{O(r)}$	1

Figure 6.1: Running times and approximation guarantees for various Lasserre Hierarchy relaxation rounding algorithms using our faster solver (η is optimal value). [1]=Raghavendra and Tan [2012], [2]=Arora and Ge [2011]. For sparsest cut, the spectral assumption is $\lambda_r \geq \eta/(1-\varepsilon)$. For first independent set, the spectral assumption is $\lambda_{n-r} \leq 1 + O(1/d_{\max})$.For second independent set [Arora and Ge, 2011], the assumptions are that G is 3-colorable and $\lambda_{n-r} \leq 17/16$.

Chapter 7

Graph Partitioning with Linear Constraints

In this chapter, we will generalize the rounding algorithm from Chapter 4 to quadratic integer programming problems with non-negative cost functions under simple global constraints. Basic rounding algorithms are given in Algorithms 7.1 and 7.2. We show how to implement this under our framework as described in Chapter 5, and present the running time using the faster solver.

Next we demonstrate how we can use this approximation algorithm to approximate minimum bisection, small set expansion and their k-way generalizations immediately. Our final application is to the problem of finding large independent sets in a graph. We end the chapter by presenting a different perspective for our rounding, in terms of variance reduction.

Our analysis yields bounds in terms of the underlying cost graph's spectrum: Faster the spectrum grows, the better our solutions become. Unfortunately for a problem of main interest, Unique Games, this means our performance is related to the actual cost graph's spectrum (also known as the lifted graph), which might be much smaller than original constraint graph's spectrum. We will show how to get around this issue later in Chapter 8.

7.1 Seed Based Rounding

We first describe how to perform the rounding *after* a good choice of the seed set $\mathcal{S}^*$ has been made, followed by an analysis of its properties. The algorithm is given in Algorithm 7.1. This part is quite simple; the crux of our rounding is how

Algorithm 7.1 Seed based labeling algorithm with running time $O\left(k^{r'}+n\right)$.

Input: • Positive integers n, k, r' representing number of variables, labels and rounds respectively; seed set $\mathcal{S}^* \in \binom{V}{\leq r'}$ and k-label moment sequence x.

Output: • Indicator vector for a valid k labeling of V, $\mathbf{x} \in \{0,1\}^{V\times[k]}$.

Procedure: 1. Let $[\vec{x}_{S(f)}]_{S,f}$ be labeling vectors for moment sequence x as described in Definition 3.46.

2. Choose $f \in [k]^{\mathcal{S}^*}$ with probability $\|\vec{x}_{\mathcal{S}^*(f)}\|^2$.

3. Label every node $u \in V$ by choosing a label $j \in [k]$ with probability $\frac{\langle \vec{x}_{S(f)}, \vec{x}_{u(j)}\rangle}{\|\vec{x}_{S(f)}\|^2}$:

$$\mathbf{x}_{(u,i)} \leftarrow \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{else.} \end{cases}$$

to choose the best $\mathcal{S}^*$ and bound the performance when it is used as the seed set. This will be described in Section 7.2.

Definition 7.1 (Rounding distribution)**.** *Given seed set $\mathcal{S}^* \in \binom{V}{\leq r'}$ and moment sequence x, let $[\vec{x}_{S(f)}]_{S,f}$ be labeling vectors for x.*

We will use $f \sim \|\vec{x}_{\mathcal{S}^*(f)}\|^2$ to denote the distribution on $[k]^{\mathcal{S}^*}$ where each $f \in [k]^{\mathcal{S}^*}$ is chosen with probability $\|\vec{x}_{\mathcal{S}^*(f)}\|^2$. In other words:

$$\mathrm{Prob}_{f'\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[f' = f\right] = \|\vec{x}_{\mathcal{S}^*(f)}\|^2.$$

For any $f \in [k]^{\mathcal{S}^*}$, we use $\|\vec{x}_{\circ|\mathcal{S}^*(f)}\|^2$ as the distribution on binary vectors corresponding to indicator vectors of labelings of V, $\{0,1\}^{V\times[k]}$, in which each node $u \in V$ receives, independently at random, a label $j \in [k]$ with probability:

$$\mathrm{Prob}_{\mathbf{x}\sim\|\vec{x}_{\circ|\mathcal{S}^*(f)}\|^2}\left[\mathbf{x}_{(u,j)} = 1\right] = \|\vec{x}_{u(j)|\mathcal{S}^*(f)}\|^2 = \frac{\langle \vec{x}_{\mathcal{S}^*(f)}, \vec{x}_{u(j)}\rangle}{\|\vec{x}_{\mathcal{S}^*(f)}\|^2}.$$

We will abuse the notation and use $\mathbf{x} \sim \|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2$ for sampling a binary labeling vector by first choosing $f \sim \|\vec{x}_{\mathcal{S}^*(f)}\|^2$ and then choosing $\mathbf{x} \sim \|\vec{x}_{\circ|\mathcal{S}^*(f)}\|^2$.

We now prove some simple properties of this rounding. All claims below hold for every fixed choice of $\mathcal{S}^*$.

Claim 7.2. *For any $u \in V$ and $j \in [k]$, we have*

$$\mathrm{Prob}_{\mathbf{x}\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[\mathbf{x}_{(u,j)} = 1\right] = \|\vec{x}_{u(j)}\|^2.$$

Proof. Indeed, by definition of the rounding scheme, $\mathrm{Prob}_{\mathbf{x}\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[\mathbf{x}_{(u,j)} = 1\right]$ equals

$$\sum_f \|\vec{x}_{\mathcal{S}^*(f)}\|^2 \frac{\langle \vec{x}_{\mathcal{S}^*(f)}, \vec{x}_{u(j)}\rangle}{\|\vec{x}_{\mathcal{S}^*(f)}\|^2} = \sum_f \langle \vec{x}_{\mathcal{S}^*(f)}, \vec{x}_{u(j)}\rangle = \langle \vec{x}_\emptyset, \vec{x}_{u(j)}\rangle = \|\vec{x}_{u(j)}\|^2 . \qquad \square$$

Before stating the next claim, let us again recall the definition of the projection operator used in the analysis of the rounding.

Definition 7.3. *Given k-labeling vectors $[\vec{x}_{S(f)}]_{S,f}$ for a moment sequence x we define $\Pi_S \in \mathbb{R}^{\Upsilon\times\Upsilon}$ as the projection matrix onto the span of $\{\vec{x}_{S(f)}\}_{f\in[k]^S}$ for given S:*

$$\Pi_S \stackrel{\mathrm{def}}{=} \sum_{f\in[k]^S} \overline{\vec{x}_{S(f)}} \cdot \overline{\vec{x}_{S(f)}}^T.$$

Define $\Pi_S^\perp = I - \Pi_S$ to be the projection matrix onto the orthogonal complement of the span of $\{\vec{x}_{S(f)}\}_{f\in[k]^S}$, where I denotes the identity matrix of appropriate dimension.

Claim 7.4. *For any $u \neq v \in V$ and $i, j \in [k]$:*

$$\mathrm{Prob}_{\mathbf{x}\sim\|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2}\left[\mathbf{x}_{(u,i)} = 1 \wedge \mathbf{x}_{(v,j)} = 1\right] = \langle \Pi_{\mathcal{S}^*}\vec{x}_{u(i)}, \Pi_{\mathcal{S}^*}\vec{x}_{v(j)}\rangle.$$

Proof.

$$\begin{aligned}
\mathrm{Prob}_{\mathbf{x}\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[\mathbf{x}_{u(i)} = 1 \wedge \mathbf{x}_{v(j)} = 1\right] &= \sum_f \|\vec{x}_{\mathcal{S}^*(f)}\|^2 \frac{\langle \vec{x}_{\mathcal{S}^*(f)}, \vec{x}_{u(i)}\rangle\langle \vec{x}_{\mathcal{S}^*(f)}, \vec{x}_{v(j)}\rangle}{\|\vec{x}_{\mathcal{S}^*(f)}\|^4} \\
&= \sum_f \langle \overline{\vec{x}_{\mathcal{S}^*(f)}}, \vec{x}_{u(i)}\rangle\langle \overline{\vec{x}_{\mathcal{S}^*(f)}}, \vec{x}_{v(j)}\rangle \\
&= \sum_f \vec{x}_{u(i)}^T \overline{\vec{x}_{\mathcal{S}^*(f)}} \cdot \overline{\vec{x}_{\mathcal{S}^*(f)}}^T \vec{x}_{v(j)} \\
&= \vec{x}_{u(i)}^T \Pi_{\mathcal{S}^*}\vec{x}_{v(j)} = \langle \Pi_{\mathcal{S}^*}\vec{x}_{u(i)}, \Pi_{\mathcal{S}^*}\vec{x}_{v(j)}\rangle. \qquad \square
\end{aligned}$$

Claim 7.5. *Given any $\mathcal{S}^*$ with $\mathbf{x}$ sampled from $\|\vec{x}_{\mathcal{S}^*(f)}\|^2$ as described, the following identity holds: For any matrix $L \in \mathbb{S}^{V\times[k]}$, if we let $\vec{X} = [\vec{x}_{u(i)}]_{u\in V, i\in[k]} \in \mathbb{R}^{\Upsilon\times(V\times[k])}$ be the matrix whose columns correspond to vectors $\vec{x}_{u(i)}$:*

$$\mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2}\left[\mathbf{x}^T L\mathbf{x}\right] = \mathrm{Tr}(\vec{X}^T\Pi^{\perp}_{\mathcal{S}^*}\vec{X}\,\mathrm{diag}(L)) + \mathrm{Tr}(\vec{X}^T\Pi_{\mathcal{S}^*}\vec{X}L)$$

Proof. Consider $L = \mathrm{diag}(A) + L^o$:

$$\mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[\mathbf{x}^T L\mathbf{x}\right] = \mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[\mathbf{x}^T\,\mathrm{diag}(L)\mathbf{x} + \mathbf{x}^T L^o\mathbf{x}\right]$$

Using Claims 7.2 and 7.4:

$$\begin{aligned}
&= \mathrm{Tr}(\vec{X}^T\vec{X}\,\mathrm{diag}(L)) + \mathrm{Tr}(\vec{X}^T\Pi_{\mathcal{S}^*}\vec{X}L^o)\\
&= \mathrm{Tr}(\vec{X}^T\vec{X}\,\mathrm{diag}(L)) + \mathrm{Tr}(\vec{X}^T\Pi_{\mathcal{S}^*}\vec{X}(L - \mathrm{diag}(L)))\\
&= \mathrm{Tr}(\vec{X}^T\Pi^{\perp}_{\mathcal{S}^*}\vec{X}\,\mathrm{diag}(L)) + \mathrm{Tr}(\vec{X}^T\Pi_{\mathcal{S}^*}\vec{X}L).
\end{aligned}$$

□

7.2 Choosing Good Seeds

In this section we show how to pick a good $\mathcal{S}^*$ and prove our main result, Theorem 7.7, which lets us relate the performance of our rounding algorithm to the objective value of relaxation. The final seed selection algorithm is given in Algorithm 7.2.

We begin with a lemma relating the best bound achieved by column-selection for a matrix $\vec{X}$ (as in Theorem 10.1) to the objective function $\mathrm{Tr}(\vec{X}^T\vec{X}L)$ with respect to an arbitrary matrix $L \in \mathbb{S}^{V\times[k]}_+$.

Lemma 7.6. *Given $X \in \mathbb{R}^{R\times C}$ and a PSD matrix $L \in \mathbb{S}^C_+$, for any positive integer r and positive constant $\varepsilon > 0$, there exists r/ε columns, $S \in \binom{C}{\leq r}$ of X such that*

$$\mathrm{Tr}(X^T X^{\perp}_S X\,\mathrm{diag}(L)) \leq \frac{\mathrm{Tr}(X^T XL)}{(1-\varepsilon)\lambda_{r+1}(\mathcal{L})}$$

where $\lambda_{r+1}(\mathcal{L})$ is $(r+1)^{th}$ smallest normalized eigenvalue of L as defined in Definition 2.43. Furthermore such S can be found in deterministic $O(rn^4)$ time.

Algorithm 7.2 Deterministic seed selection algorithm with running time $O(n^5)$.

Input: • Positive integers $n, k, r, r' = r/\varepsilon^2$ representing number of variables, labels, rounds per iteration and total rounds respectively; k-label moment sequence x.

• Positive semi-definite cost matrix $L \in \mathbb{S}_+^{V\times[k]}$.

Output: • Seed set $\mathcal{S}^* \in \binom{V}{\leq r'}$ satisfying eq. (7.1).

Procedure: 1. Let $\mathcal{S}^* \leftarrow \emptyset$.

2. Let $[\vec{x}_{S(f)}]_{S,f}$ be labeling vectors for x as described in Definition 3.46.

3. Repeat for $1/\varepsilon$ times:

   (a) Let $\Pi_{\mathcal{S}^*} \leftarrow \sum_{f\in[k]^{\mathcal{S}^*}} \overline{\vec{x}_{S(f)}} \cdot \overline{\vec{x}_{S(f)}}^T$ be the projection matrix of $\operatorname{span}\{\vec{x}_{\mathcal{S}^*(f)}\}_f$.

   (b) Find new $\frac{r}{\varepsilon}$-many seeds $\widetilde{T} \in \binom{V\times[k]}{\leq r/\varepsilon}$ by choosing columns from matrix $\Pi^{\perp}_{\mathcal{S}^*}[\vec{x}_{u(i)}]_{u\in V, i\in[k]} \operatorname{diag}(L)^{1/2}$ using Algorithm 10.2 so as to minimize reconstruction error in Frobenius norm.

   (c) $T \leftarrow \left\{ u \middle| \exists j \in [k] : \ (u,j) \in \widetilde{T} \right\}$.

   (d) $\mathcal{S}^* \leftarrow \mathcal{S}^* \bigcup T$.

Proof. Let $\widetilde{X} \leftarrow X \operatorname{diag}(L)^{1/2}$ and $\mathcal{L} \leftarrow \operatorname{diag}(L)^{-1/2} L \operatorname{diag}(L)^{-1/2}$ with convention $0/0 = \infty$ and $0 \cdot \infty = 0$. i^{th} smallest eigenvalue of $\mathcal{L}$, $\lambda_i(\mathcal{L})$, corresponds to the i^{th} smallest generalized eigenvalue $\lambda_i(L; \operatorname{diag}(L))$ by Theorem 2.47. If we let σ_i be i^{th} largest eigenvalue of $\widetilde{X}^T\widetilde{X}$, then using Theorem 10.1 on vectors $\widetilde{X}$, we can find $S \in \binom{C}{\leq r'}$ in time $O(r|C|^4)$ such that

$$\operatorname{Tr}(\widetilde{X}^T \widetilde{X}_S^{\perp} \widetilde{X}) \leq \frac{1}{1-\varepsilon} \sum_{i\geq r+1} \sigma_i.$$

By von Neumann-Birkhoff theorem, $\operatorname{Tr}(\widetilde{X}^T\widetilde{X}\mathcal{L})$ is minimized when the i^{th} largest eigenvector of $\widetilde{X}^T\widetilde{X}$ corresponds to the i^{th} smallest eigenvector of $\mathcal{L}$:

$$\operatorname{Tr}(\widetilde{X}^T\widetilde{X}\mathcal{L}) \geq \sum_i \sigma_i\lambda_i \geq \sum_{i\geq r+1} \sigma_i\lambda_i \geq \lambda_{r+1} \sum_{i\geq r+1} \sigma_i \geq (1-\varepsilon)\lambda_{r+1} \operatorname{Tr}(\widetilde{X}^T \widetilde{X}_S^{\perp} \widetilde{X}).$$

The span of $\{\widetilde{X}_u\}_{u\in S}$ is the same with $\{X_u\}_{u\in S}$ since $\widetilde{X}_u$ differs from X_u only by a scaling factor which leaves the span unchanged. In particular, $\widetilde{X}_S^\perp = X_S^\perp$:

$$\mathrm{Tr}(\widetilde{X}^T\widetilde{X}_S^\perp\widetilde{X}) = \mathrm{Tr}(\widetilde{X}^T X_S^\perp\widetilde{X}) = \mathrm{Tr}(X^T X_S^\perp X\,\mathrm{diag}(L)).$$

The proof is complete by noting that $\mathrm{Tr}(\widetilde{X}^T\widetilde{X}\mathcal{L}) = \mathrm{Tr}(X^T X L)$. □

Theorem 7.7 (Main technical theorem)**.** *Given positive integer r and $\varepsilon \in (0,1)$, let x be a moment sequence satisfying $r' = O\left(\frac{r}{\varepsilon^2}\right)$ rounds of Lasserre hierarchy constraints, k-label moment sequence x,*

Given $L \in \mathbb{S}_+^{V\times[k]}$, we can find a seed set $\mathcal{S}^*$ of size at most r' in deterministic time $O(n^5)$ with the following properties. For $\mathbf{x}$ randomly sampled from the distribution $\|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2$, $\mathbf{x} \sim \|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2$, as described in Definition 7.1:

1. *$\mathbf{x}$ is a binary vector, $\mathbf{x} \in \{0,1\}^{V\times[k]}$.*
2. *$\mathbf{x}$ is an indicator function of a proper labeling of V. In particular for any $u \in V$,*

$$\sum_{i\in[k]} \mathbf{x}_{(u,i)} = 1.$$

3. $\mathbb{E}\left[\mathbf{x}_{(u,i)}\right] = x_{u(i)}$.
4. *The expected correlation of $\mathbf{x}$ with L is bounded by the correlation of x with L as follows.*

   If we let $[\vec{x}_{S(f)}]_{S,f}$ be labeling vectors for this moment sequence and $\vec{X} \overset{\text{def}}{=} [\vec{x}_{u(i)}]_{u\in V, i\in[k]}$ be the matrix with columns being vectors $\vec{x}_{u(i)}$ for all $u \in V, i \in [k]$:

$$\mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2}\left[\mathbf{x}^T L\mathbf{x}\right] \le \frac{1+\varepsilon}{1-\varepsilon}\frac{\mathrm{Tr}(\vec{X}^T\vec{X}L)}{\min\{\lambda_{r+1}(L,\mathrm{diag}(L)),1\}}.$$

Furthermore this set $\mathcal{S}^*$ satisfies the following bound

$$\mathrm{Tr}(\vec{X}^T\Pi_{\mathcal{S}^*}^\perp\vec{X}\,\mathrm{diag}(L)) + \mathrm{Tr}(\vec{X}^T\Pi_{\mathcal{S}^*}\vec{X}L) \le \frac{1+\varepsilon}{1-\varepsilon}\frac{\mathrm{Tr}(\vec{X}^T\vec{X}L)}{\min\{\lambda_{r+1}(L,\mathrm{diag}(L)),1\}} \tag{7.1}$$

where $\Pi_{\mathcal{S}^*}$ is defined in Definition 7.3.

Proof. Note that the first three properties follow by construction of $\|\vec{x}_{\mathcal{S}^*(f)}\|^2$. Using Claim 7.5, it can be seen that the bound in eq. (7.1) is equivalent to item 4. Therefore it suffices to prove item 4.

Define $\lambda_{r+1} \stackrel{\text{def}}{=} \lambda_{r+1}(L, \operatorname{diag}(L))$ and let $r_0 \leftarrow r/\varepsilon$. Consider picking our "seed" nodes in the following iterative way as described in Algorithm 7.2. Starting with $\Delta S_0 \leftarrow \emptyset$ and $\vec{X}(0) \leftarrow \vec{x}_{\emptyset}^{\perp}\vec{X}$, for each $i \in \{1, 2, \ldots\}$, set $\Delta\widetilde{S}_i$ as

$$\Delta\widetilde{S}_i \leftarrow \operatorname*{argmin}_{S \in \binom{V\times[k]}{\le r_0}} \operatorname{Tr}(\vec{X}(i-1)^T \vec{X}(i-1)_S^{\perp} \vec{X}(i-1)\operatorname{diag}(L)),$$

and ΔS_i as the set of nodes whose at least one label appears in $\Delta\widetilde{S}_i$ so that

$$\Delta S_i \leftarrow \left\{u \mid \exists g \in [k] \text{ such that } (u,g) \in \Delta\widetilde{S}_i\right\};\ S_i \leftarrow \bigcup_{j\le i} \Delta S_j;$$

followed by $\vec{X}(i) \leftarrow \Pi_{S_i}^{\perp}\vec{X}$. At each step we set $\mathcal{S}^* \leftarrow S_i$ and repeat this until $\mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2}\left[\mathbf{x}^T L\mathbf{x}\right]$ is at most $\frac{1+\varepsilon}{1-\varepsilon}\frac{\eta}{\min(1,\lambda_{r+1})}$, where $\eta \stackrel{\text{def}}{=} \operatorname{Tr}(\vec{X}^T\vec{X}L)$.

Note that, by Lasserre constraints, all vectors in $\{\vec{x}_{u(i)}\}_{u\in S, i\in[k]}$ are linear combinations of vectors in $\{\vec{x}_{\Delta S_f}\}_{f\in[k]^S}$. Hence for any subset of nodes $T \subseteq V$ of size at most r', $\vec{X}_{T\times[k]}^{\perp} \succeq \Pi_T^{\perp}$.

For any i, using $\|\vec{x}_{\circ|S_i(\circ)}\|^2$ to denote the distribution at iteration i with seed set chosen as $\mathcal{S}^* \leftarrow S_i$, by Claim 7.5:

$$\mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\circ|S_i(\circ)}\|^2}\left[\mathbf{x}^T L\mathbf{x}\right] = \operatorname{Tr}(\vec{X}^T\Pi_{S_i}^{\perp}\vec{X}\operatorname{diag}(L)) + \operatorname{Tr}(\vec{X}^T\Pi_{S_i}\vec{X}L) \tag{7.2}$$

Let ξ_i be defined as $\xi_i \stackrel{\text{def}}{=} \mathbb{E}_{\mathbf{x}\sim\|\vec{x}_{\circ|S_i(\circ)}\|^2}\left[\mathbf{x}^T L\mathbf{x}\right]$ so that:

$$\xi_i = \underbrace{\operatorname{Tr}(\vec{X}^T\Pi_{S_i}^{\perp}\vec{X}\operatorname{diag}(L))}_{\delta_i} + \underbrace{\operatorname{Tr}(\vec{X}^T\Pi_{S_i}\vec{X}L)}_{\eta_i}$$

Finally for convenience we define λ'_{r+1} as the following:

$$\lambda'_{r+1} \stackrel{\text{def}}{=} (1-\varepsilon)\min\left\{\lambda_{r+1}(L, \operatorname{diag}(L)), 1\right\}. \tag{7.3}$$

We will show that this procedure will stop for some i with $i \le \lceil\frac{1}{\varepsilon}\rceil$ in Claim 7.12. Note that each iteration takes time at most $O(r_0 n^4)$. If this procedure takes K iterations, we have $r_0 K \le n$, hence running time is $O(K r_0 n^4) = O(n^5)$.

Observation 7.8.

$$\delta_{i+1} = \mathrm{Tr}(\vec{X}^T \Pi^{\perp}_{S_{i+1}} \vec{X} \,\mathrm{diag}(L)) \leq \mathrm{Tr}(\vec{X}(i)^T \Pi^{\perp}_{\Delta S_{i+1}} \vec{X}(i) \,\mathrm{diag}(L))$$

Proof. Note that $\Pi^{\perp}_{S_i} \Pi^{\perp}_{\Delta S_{i+1}} \Pi^{\perp}_{S_i} \succeq \Pi^{\perp}_{S_{i+1}}$ since all vectors of the form $\vec{x}_{S_i(f)}$ and $\vec{x}_{\Delta S_i(f')}$ are linear combinations of vectors $\vec{x}_{S_{i+1}(g)}$. Using the definition of $\vec{X}(i)$, $\vec{X}(i) = \Pi^{\perp}_{S_i} \vec{X}$, the proof is complete. □

Observation 7.9. *For any $i \geq 0$, we have $\eta_i \leq \eta$.*

Proof. Note $\Pi_{S_i} \preceq I$. Since $L \succeq 0$, $\eta_i = \mathrm{Tr}(\vec{X}^T \Pi_{S_i} \vec{X} L) \leq \mathrm{Tr}(\vec{X}^T \vec{X} L) = \eta$. □

Claim 7.10. *For any $i \geq 0$,*

$$\delta_{i+1} \leq \frac{\eta - \eta_i}{\lambda'_{r+1}}.$$

where λ'_{r+1} is defined in eq. (7.3).

Proof. Using Observation 7.8,

$$\begin{aligned}
\delta_{i+1} &\leq \mathrm{Tr}(\vec{X}(i)^T \Pi^{\perp}_{\Delta S_{i+1}} \vec{X}(i) \,\mathrm{diag}(L)) \\
&\leq \mathrm{Tr}(\vec{X}(i)^T X^{\perp}_{\Delta S_{i+1} \times [k]} \vec{X}(i) \,\mathrm{diag}(L)) \\
&\leq \mathrm{Tr}(\vec{X}(i)^T X^{\perp}_{\widetilde{S}(i+1)} \vec{X}(i) \,\mathrm{diag}(L)) \\
&\leq \frac{1}{(1-\varepsilon)\lambda_{r+1}} \mathrm{Tr}(\vec{X}(i)^T \vec{X}(i) L) \,,
\end{aligned}$$

where the first inequality follows from $\Pi^{\perp}_{\Delta S_{i+1}} \preceq \vec{X}^{\perp}_{\Delta S_{i+1} \times [k]}$, and the second inequality from $\Delta \widetilde{S}_{i+1} \subseteq \Delta S_{i+1} \times [k]$. For the last inequality, we can immediately apply the bound from Lemma 7.6. Using $(1-\varepsilon)\lambda_{r+1}(L, \mathrm{diag}(L)) \geq \lambda'_{r+1}$, where λ'_{r+1} is as defined in eq. (7.3), and the identity

$$\mathrm{Tr}(\vec{X}(i)^T \vec{X}(i) L) = \mathrm{Tr}(\vec{X}^T \Pi^{\perp}_{S_i} \vec{X} L) = \mathrm{Tr}(\vec{X}^T \vec{X} L) - \mathrm{Tr}(\vec{X}^T \Pi_{S_i} \vec{X} L) = \eta - \eta_i$$

we conclude the proof. □

Claim 7.11. *If $\xi_{i+1} > \eta \frac{1+\varepsilon}{\lambda'_{r+1}}$, then*

$$\frac{\varepsilon + \eta_i}{\lambda'_{r+1}} < \eta_{i+1}.$$

Proof. Using Claim 7.10,

$$\eta\frac{1+\varepsilon}{\lambda'_{r+1}} < \xi_{i+1} = \delta_{i+1} + \eta_{i+1} \leq \frac{\eta - \eta_i}{\lambda'_{r+1}} + \eta_{i+1}.$$

Hence

$$\frac{\varepsilon + \eta_i}{\lambda'_{r+1}} < \eta_{i+1}\,. \qquad \square$$

Claim 7.12. *There exists $i \leq \lceil \frac{1}{\varepsilon} \rceil$ for which $\xi_i \leq \eta\frac{1+\varepsilon}{\lambda'_{r+1}}$.*

Proof. By contradiction. Let $K = \lceil \frac{1}{\varepsilon} \rceil$ and assume for all $i \leq K$, $\xi_i > \eta\frac{1+\varepsilon}{\lambda'_{r+1}}$. By Claim 7.11,

$$\begin{aligned}
\eta_1 &> \eta\frac{\varepsilon}{\lambda'_{r+1}} \geq \eta\varepsilon \\
\eta_2 &> \eta\frac{\varepsilon}{\lambda'_{r+1}} + \frac{\eta_1}{\lambda'_{r+1}} > \eta\frac{\varepsilon}{\lambda'_{r+1}}(1+1) \geq \eta \cdot 2\varepsilon \\
&\vdots \\
\eta_K &> \eta\frac{\varepsilon}{\lambda'_{r+1}} + \frac{\eta_{K-1}}{\lambda'_{r+1}} > \eta K\frac{\varepsilon}{\lambda'_{r+1}} \geq \eta \cdot K\varepsilon \implies \eta_K > \eta.
\end{aligned}$$

which contradicts Observation 7.9. $\square$

This completes the proof of Theorem 7.7. $\square$

7.3 Combining with Our Faster Solver

In this section, we will show how to cast Algorithm 7.2 in our local rounding framework from Chapter 5 with the ultimate goal being reducing the running time. Observe that our labeling procedure, Algorithm 7.1, needs no modifications and will work fine. Only seed selection Algorithm 7.2 needs some trivial tweaks, all related only to changing it from being iterative to recursive. New seed selection algorithm is given in Algorithm 7.3. We put everything together in Theorem 7.13 and obtain an approximation algorithm for generic quadratic integer programming (QIP) problems with positive semi-definite objectives functions:

Algorithm 7.3 $T = \textbf{FAST-SEED-QIP}^A_S(x)$: Seed selection procedure for approximation algorithms given in Chapter 7 under local rounding framework.

Input: • Cost matrix $A \in \mathbb{S}_+^{V\times[k]}$,

• Subset $S \subseteq V$ representing seeds chosen so far,

• Moment sequence x with $x_\emptyset = 1$,

• Number of seeds to choose as a positive integer r'.

Output: • Enlarged seed set, $T \supseteq S$, with $|T| \leq k^{r'} \cdot |S|$.

Procedure: 1. Let $[\vec{x}_{A(g)}]_{A\in \mathrm{ex}(S,2),g\in[k]^A}$ be labeling vectors for moment sequence x as described in Definition 3.46.

2. Let $\Pi_S^\perp \leftarrow \sum_{f\in[k]^S} \frac{1}{\|\vec{x}_{S(f)}\|^2}\vec{x}_{S(f)}\vec{x}_{S(f)}^T$ and $\Pi_S^\perp \leftarrow I - \Pi_S$.

3. Let $\vec{X} \leftarrow [\vec{X}_{(u,j)}]_{u\in V,j\in[k]} \in \mathbb{R}^{\Upsilon,V\times[k]}$ be the matrix whose columns $\vec{X}_{(u,j)} \in \mathbb{R}^\Upsilon$ are given by
$$\vec{X}_{(u,j)} \leftarrow \Pi_S^\perp \vec{x}_{u(j)} \sqrt{A_{u(j),u(j)}}.$$

4. Use deterministic column selection procedure, Algorithm 10.2, on $\vec{X}$ to choose r' columns, $S' \in \binom{V\times[k]}{r'}$.

5. Return $S \cup \left\{u \in V \middle| \exists j \in [k] : (u,j) \in S'\right\}$.

Theorem 7.13. *Consider a quadratic integer programming problem*

$$\begin{array}{ll} \min & \mathbf{x}^T A \mathbf{x} \\ \text{st} & B\mathbf{x} \geq c \\ & \sum_{i\in[k]} \mathbf{x}_{(u,i)} = 1 \quad \textit{for all } u \in V, \\ & \mathbf{x} \in \{0,1\}^{V\times[k]} \end{array}$$

where $A \in \mathbb{S}_+^{V\times[k]}$ represents a quadratic objective function and $B \in \mathbb{R}^{N,V\times[k]}$, $c \in \mathbb{R}^N$ represent linear constraints.

Given such A and a separation oracle for linear constraints $B\mathbf{x} \geq c$, for any $0 < \varepsilon < 1$ and positive integer r, there exists an algorithm that runs in time $2^{O(r/\varepsilon^2)} n^{O(1/\varepsilon)} \cdot T(\mathbf{SEP})$ which outputs a seed set $\mathcal{S}^* \in \binom{V}{\leq r/\varepsilon}$ and k-label moment sequence x satisfying

all guarantees from Theorem 7.7.

Proof. Run Algorithm 5.2 with following input:

- $\varepsilon_0 \leftarrow \varepsilon^{-n}$.
- $\ell \leftarrow 1/\varepsilon$.
- **FEASIBLE** is the separation oracle for Lasserre Hierarchy as outlined in Section 5.6.
- **SEED** is Algorithm 7.3.
- (Not necessary, only for completeness) **ROUND** is Algorithm 7.1. □

7.4 Applications

In this section, we show how we can use the algorithm from Theorem 7.13 for various combinatorial optimization problems.

7.4.1 Minimum Bisection

We will express Minimum Bisection problem in a slightly different way than Chapter 4. This presentation will be useful later for generalization to k-way partitioning.

$$\begin{aligned} \min_{\mathbf{x}} \sum_{u<v} w_{u,v}(\mathbf{x}_u - \mathbf{x}_v)^2 \\ \text{st} \sum_{u \in V} \mathbf{x}_u = \mu,\ \mathbf{x} \in \{0,1\}^V. \end{aligned}$$

Given this formulation, we can immediately use Theorem 7.13 and obtain the following:

Corollary 7.14 (Minimum Bisection)**.** *Given $0 < \varepsilon < 1$, positive integer r, graph G, a target size $\mu \leq \frac{n}{2}$, there exists an algorithm which runs in time $2^{O(r/\varepsilon^3)} n^{O(1/\varepsilon)}$ and outputs a set whose indicator vector $\mathbf{x} \in \{0,1\}^V$ satisfies the following with high probability:*

$$\mu - O\left(\sqrt{\mu \log(1/\varepsilon)}\right) \leq \|\mathbf{x}\|_1 \leq \mu + O\left(\sqrt{\mu \log(1/\varepsilon)}\right),$$

$$\mathbf{x}^T L \mathbf{x} \leq \frac{1+\varepsilon}{\min\{\lambda_{r+1}(\mathcal{L}), 1\}} \mathsf{OPT}\,.$$

Proof. Note that the objective matrix takes the form $L' = \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix}$. The corresponding $(r+1)^{th}$ smallest generalized eigenvalue $\lambda_{r+1}(L'; \mathrm{diag}(L'))$ is equal to that of the normalized graph Laplacian matrix, $\lambda_{r+1}(\mathcal{L})$. Thus we can use Theorem 7.13 with $\varepsilon/10$ and get

$$\mathbb{E}\left[\mathbf{x}^T L \mathbf{x}\right] \leq (1+\varepsilon/10)\frac{\mathsf{OPT}}{\lambda_{r+1}}, \ \mathbb{E}\left[\|\mathbf{x}\|_1\right] = \mu.$$

Using Markov inequality (Theorem 2.56) on the first expectation and Chernoff bound (Theorem 2.58) on the second one together with each x being independent:

$$\mathrm{Prob}\left[\mathbf{x}^T L \mathbf{x} \geq (1+\varepsilon/2)\frac{\mathsf{OPT}}{\lambda_{r+1}}\right] \leq 1-\varepsilon/3, \ \mathrm{Prob}\left[\left|\|\mathbf{x}\|_1 - \mu\right| \geq O(\sqrt{\mu \log(1/\varepsilon)})\right] \leq \varepsilon/3.$$

Taking a union bound:

$$\mathrm{Prob}\left[\mathbf{x}^T L \mathbf{x} \leq (1+\varepsilon/2)\frac{\mathsf{OPT}}{\lambda_{r+1}} \wedge \left|\|\mathbf{x}\|_1 - \mu\right| \leq O(\sqrt{\mu \log(1/\varepsilon)})\right] \geq \varepsilon/3.$$

Repeating this $O(1/\varepsilon)$ times finishes the proof. □

7.4.2 Small Set Expansion

Our next result is on the small set expansion problem. A naïve application of Theorem 7.13 will yield good bounds only when the graph does not have high degree nodes (compared to the average degree). However our guarantee is irrespective of the degree distribution on graph G such that we are always able to find a set of volume $\mu(1 \pm \varepsilon)$. In order to achieve this while still making sure that the running time depends on $2^{O(r)}$, it becomes crucial that our rounding permits arbitrary unary constraints.

We use the following standard integer programming formulation of SSE:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{e=\{u,v\} \in E} (\mathbf{x}_u - \mathbf{x}_v)^2, \\ \text{st} \quad & \sum_{u \in V} d_u \mathbf{x}_u = \mu, \\ & \mathbf{x} \in \{0,1\}^V. \end{aligned}$$

Theorem 7.15 (Small Set Expansion). *Given $0 < \varepsilon < 1$, positive integer r, a target volume μ, there exists an algorithm which runs in time $n^{O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)} 2^{O(r/\varepsilon^2)}$ and outputs a set whose indicator vector $\mathbf{x} \in \{0,1\}^V$ satisfies the following:*

$$\mathbf{x}^T L \mathbf{x} \leq \frac{1+\varepsilon}{\min\{\lambda_{r+1}(\mathcal{L}), 1\}} \mathsf{OPT}$$

and

1. *If maximum degree satisfies $d_{\max} \leq O\left(\frac{\mu}{\log \frac{1}{\varepsilon}}\right)$, then*

$$\mu\left(1 - O\left(\sqrt{\frac{d_{\max}}{\mu} \log \frac{1}{\varepsilon}}\right)\right) \leq \mathbf{x}^T \operatorname{diag}(L)\mathbf{x} \leq \mu\left(1 + O\left(\sqrt{\frac{d_{\max}}{\mu} \log \frac{1}{\varepsilon}}\right)\right)$$

2. *Else*

$$\mu\,(1-\varepsilon) \leq \mathbf{x}^T \operatorname{diag}(L)\mathbf{x} \leq \mu\,(1+\varepsilon)\ .$$

Proof of item 1. Proof is the same with that of Corollary 7.14 with the only difference being the usage of Hoeffding bound Theorem 2.57 instead of Chernoff bound. □

Proof of item 2. At a high level, our algorithm proceeds in the following way: We enumerate all subsets U_0 of volume at most μ from the set of high degree nodes $\mathcal{H}$, which is defined by $\mathcal{H} \stackrel{\text{def}}{=} \left\{u \mid d_u \geq \frac{\varepsilon^2}{\log(1/\varepsilon)}\mu\right\}$. For each such subset U_0, we solve the corresponding Lasserre SDP relaxation of Small Set Expansion problem on this graph with constraints $\mathbf{x}_u = 1$ for any $u \in U_0$ and $\mathbf{x}_{(v,2)} = 1$ for any $v \in \mathcal{H} \setminus U_0$. Objective matrix for this problem is $L_{V\setminus\mathcal{H}, V\setminus\mathcal{H}}$ whose normalized eigenvalues interlace that of original graph Laplacian matrix L. Moreover our volume constraint takes the form

$$\sum_{v \notin \mathcal{H}} d_v \mathbf{x}_v \leq \mu' \stackrel{\text{def}}{=} \mu - \sum_{u \in U_0} d_u.$$

Observe that the maximum degree, say $d'_{\max}$, in the induced graph is at most $\frac{\varepsilon^2}{\log(1/\varepsilon)}\mu$. Now there are two possible cases:

1. If $d'_{\max} < \frac{1}{\log(1/\varepsilon)}\mu'$ then this reduces to item 1, which finds an indicator vector $\mathbf{x}$ that satisfies

$$\left|\mathbf{x}^T \operatorname{diag}(L)\mathbf{x} - \mu\right| \leq O(\sqrt{\mu' d'_{\max} \log(1/\varepsilon)}) \leq \sqrt{\mu' \varepsilon^2 \mu} \leq \sqrt{\mu \varepsilon^2 \mu} = \mu\varepsilon.$$

2. Else, we have $d'_{\max} \geq \frac{1}{\log(1/\varepsilon)}\mu' \implies \frac{1}{\log(1/\varepsilon)}\mu' \leq \frac{\varepsilon^2}{\log(1/\varepsilon)}\mu \implies \mu' \leq \varepsilon^2\mu$. Then, instead of Chernoff bound, we can use simple Markov bound and conclude that

$$\text{Prob}\left[\mathbf{x}^T \operatorname{diag}(L)\mathbf{x} \geq \frac{\mu'}{\varepsilon/2}\right] \leq \varepsilon/2.$$

Combining this $\text{Prob}\left[\mathbf{x}^T L\mathbf{x} \geq (1+\varepsilon)\mathbb{E}\left[\mathbf{x}^T L\mathbf{x}\right]\right] \leq 1-\varepsilon$, with probability $\Omega(\varepsilon)$, we will find $\mathbf{x}$ with

$$\left|\mathbf{x}^T \operatorname{diag}(L)\mathbf{x} - \mu\right| \leq \mu'/(\varepsilon/2) \leq 2\varepsilon\mu \text{ and } \mathbf{x}^T L\mathbf{x} \leq (1+\varepsilon)\mathbb{E}\left[\mathbf{x}^T L\mathbf{x}\right].$$

After enumerating all such sets, we return the one with smallest cut. Correctness of this algorithm is obvious.

For running time, note that number of nodes we can choose from $\mathcal{H}$ is at most $\frac{\log(1/\varepsilon)}{\varepsilon^2}$. Hence we invoke the algorithm from theorem 7.13 at most

$$\binom{|\mathcal{H}|}{\leq \frac{\log(1/\varepsilon)}{\varepsilon^2}} \leq n^{O\left(\frac{\log(1/\varepsilon)}{\varepsilon^2}\right)}$$

times, from which the running time bound follows. □

7.4.3 k-Way Partitioning Problems

Note that all these results can be generalized to their respective k-way partitioning versions (wherever it makes sense). The only difference is that, in each case, the objective matrix will be a block diagonal matrix consisting of k copies of graph Laplacian matrix. It is easy to see that such a matrix has exactly k copies of the original eigenvalues, so instead of r rounds of Lasserre hierarchy, we will use $k \cdot r$ rounds instead.

Corollary 7.16 (Minimum k-way Section)**.** *Given $0 < \varepsilon < 1$, positive integer r and a target set sizes $(\mu_i)_{i\in[k]}$ with $\sum_i \mu_i = n$, there exists an algorithm which runs in time $2^{O\left(\frac{kr}{\varepsilon^2}\right)} n^{O(1/\varepsilon)}$ to find a k-way partitioning whose indicator vector, $\mathbf{x} \in \{0,1\}^{V\times[k]}$ satisfies the following:*

$$\forall i: \ \mu_i - O\left(\sqrt{\mu_i \log(k/\varepsilon)}\right) \leq \sum_{u\in V} \mathbf{x}_{(u,i)} \leq \mu_i + O\left(\sqrt{\mu_i \log(k/\varepsilon)}\right),$$

$$\sum_i \mathbf{x}_{V\times\{i\}}^T L \mathbf{x}_{V\times\{i\}} \leq \frac{1+\varepsilon}{\min\{\lambda_{r+1}(\mathcal{L}), 1\}}\mathsf{OPT}$$

provided that such sets exist.

Proof. Proof follows by applying Theorem 7.13 to:

$$\begin{aligned}
\min_{\mathbf{x}} \sum_i \sum_{e=\{u,V\}\in E} & w_e(\mathbf{x}_{(u,i)} - \mathbf{x}_{(v,i)})^2, \\
\text{st} \sum_{u\in V} \mathbf{x}_{(u,i)} &= \mu_i \quad \text{for all } i \in [k], \\
\sum_{i\in[k]} \mathbf{x}_{(u,i)} &= 1 \quad \forall u \in V, \\
\mathbf{x} &\in \{0,1\}^{V\times[k]}.
\end{aligned}$$

Let $\widetilde{L}$ be the matrix in the objective. As remarked at the beginning of this section, the corresponding normalized matrix has k copies of each eigenvalue of the normalized graph Laplacian, so $\mathbf{x}$ will satisfy

$$\mathbf{x}^T \widehat{L} \mathbf{x} \leq \frac{1+O(\varepsilon)}{\min\{\lambda_{r/k}(\mathcal{L}), 1\}}\mathsf{OPT},$$

with probability $\geq 1-\varepsilon$. Our proof is complete by combining this with a Chernoff bound on the size of each partition and taking a union bound so as to prove that with probability $\Omega(\varepsilon/k)$, $\mathbf{x}$ will satisfy all conditions given. □

In this section we state and prove the main results concerning our rounding algorithm for Lasserre SDP solutions, and in particular prove Theorem 7.7 which we used to analyze our algorithm for quadratic integer programming and its applications to graph partitioning. Some of this discussion already appeared in the simpler setting of Minimum Bisection in Chapter 4. All our rounding algorithms are based on choosing labels of a carefully chosen "seed" set $\mathcal{S}^*$ of appropriate size r', which is then propagated to other nodes conditioned on the particular labeling of $\mathcal{S}^*$.

For easy reference, we describe the rounding procedure in Algorithm 7.1 and the seed selection procedure in Algorithm 7.2.

7.5 Independent Set

Our final algorithmic result is on finding independent sets in a graph. For simplicity, we focus on unweighted graphs though the extension for graphs with non-negative vertex weights is straightforward. We denote by $\alpha(G)$ the size of the largest independent set in G. Finally we assume $G = (V, E)$ is not a disjoint union of cycle graphs, i.e. there exists a node $u \in V$ with $d_u \geq 3$ (otherwise the problem is trivial).

Theorem 7.17. *Given $0 < \varepsilon < 1$, positive integer r, for any graph $G = (V, E)$ with maximum degree d, there exists an algorithm to find an independent set $I \subseteq V$ such that:*

$$|I| \geq \alpha(G) \cdot \min\left\{\frac{1}{2d}\left(\frac{1}{(1-\varepsilon)\min\{2-\lambda_{n-r-1}(\mathcal{L}), 1\}} - 1\right)^{-1}, 1\right\} \tag{7.4}$$

in time $n^{O\left(\frac{1}{\varepsilon}\right)} 2^{O(r/\varepsilon^3)}$.

Remark 7.18. *The above bound eq. (7.4) implies that if λ_{n-r-1}, which is the $(r+1)^{\text{st}}$ largest eigenvalue of the normalized graph Laplacian matrix is very close to 1, then we can find large independent sets in $2^{O(r/\varepsilon^2)} n^{O(1/\varepsilon)}$ time. In particular, if it is at most $1 + \frac{1}{4d}$ where d is the maximum degree, then taking $\varepsilon = O(1/d)$, we can find an optimal independent set. The best approximation ratio for independent set in terms of d is about $O\left(\frac{d \cdot \log\log d}{\log d}\right)$ by Halldórsson [1998] and Halperin [2002]. The bound eq. (7.4) gives a better approximation ratio as soon as $\lambda_{n-r-1} \leq 1 + O\left(\frac{1}{\log d}\right)$.* □

Proof. (of Theorem 7.17) Note that it is not possible to use Theorem 7.13 as a black box, while making sure that we find a proper independent set. Instead we will directly use Theorem 7.13. Consider the following integer program for finding largest independent set in G:

$$\begin{aligned} \max \quad & \sum_u \mathbf{x}_u \\ \text{st} \quad & \mathbf{x}_u \mathbf{x}_v = 0 \qquad \text{for any edge } e = (u, v) \in E\,, \\ & \mathbf{x} \in \{0, 1\}^V. \end{aligned}$$

Note that we can easily enforce the constraints $\mathbf{x}_u \mathbf{x}_v = 0$ by substituting $x_{\{u,v\}} \leftarrow 0$ in the Lasserre hierarchy relaxation.

Let $[\vec{x}_{S(f)}]$ be labeling vectors for moment sequence x (see Definition 3.46) and $\vec{X} = [\vec{x}_{\{u\}}]_{u \in V} \in \mathbb{R}^{\Upsilon, V}$ be the matrix whose columns are vectors $\vec{x}_{\{u\}}$ over all $u \in V$.

Observe that for A and D being adjacency and degree matrix for G respectively, the objective value of above relaxation is:

$$\mathrm{Tr}(\vec{X}^T \vec{X} D^{-1/2}(D+A)D^{-1/2}) = \mathrm{Tr}(\vec{X}^T \vec{X}).$$

We will use $\mathcal{A} \stackrel{\text{def}}{=} D^{-1/2}AD^{-1/2}$ to denote the normalized adjacency matrix so that $\mathrm{Tr}(\vec{X}^T \vec{X}\mathcal{A}) = 0$

At this point, we sample $\mathbf{x} \sim \|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2$ as in Definition 7.1. Then we convert $\mathbf{x}$ into an independent set as follows.

1. For each u, if $\mathbf{x}_u = 1$ then let $I \leftarrow I \cup \{u\}$ with probability p_u which we will specify later.
2. After the first step, for each edge $e = \{u, v\}$, if $\{u, v\} \subseteq I$, we choose one end point randomly, say u, and set $I \leftarrow I \setminus \{u\}$.

Finally we output I. It is easy to see that I is an independent set by construction. For any u, the probability that it will be included in the final independent set I is at least:

$$\begin{aligned}\mathrm{Prob}\Big[u \in I\Big] &\geq \mathbb{E}\Big[p_u \mathbf{x}_u\Big] - \frac{1}{2}\mathbb{E}\Big[\sum_{v \in N(u)} p_u p_v \mathbf{x}_u \mathbf{x}_v\Big] \\ &= p_u \|\vec{x}_u\|^2 - \frac{1}{2}\sum_{v \in N(u)} p_u p_v \langle \Pi_S \vec{x}_u, \Pi_S \vec{x}_v\rangle. \end{aligned} \tag{7.5}$$

By section 7.5, the expected size of the independent set found by the algorithm satisfies

$$\mathbb{E}\Big[|I|\Big] \geq \sum_u p_u \|\vec{x}_u\|^2 - \sum_{\{u,v\} \in E} p_u p_v \langle \Pi_S \vec{x}_u, \Pi_S \vec{x}_v\rangle. \tag{7.6}$$

Note that for every edge $\{u, v\} \in E$,

$$\langle \Pi_S \vec{x}_u, \Pi_S \vec{x}_v\rangle = \sum_{f \in \{0,1\}^S} \frac{\langle \vec{x}_{S(f)}, \vec{x}_u\rangle \langle \vec{x}_{S(f)}, \vec{x}_v\rangle}{\|\vec{x}_{S(f)}\|^2} \geq 0\,. \tag{7.7}$$

We now consider two cases.

Case 1: $\langle \Pi_S \vec{x}_u, \Pi_S \vec{x}_v\rangle = 0$ for all edges $\{u, v\} \in E$. In this case, we take $p_u = 1$ for all $u \in V$, and by eq. (7.6), we find an independent set of expected size at least $\mu \geq \alpha(G)$.

Case 2: In this case, we have

$$\sum_{\{u,v\}\in E} \langle \Pi_S \vec{x}_u, \Pi_S \vec{x}_v \rangle = \frac{1}{2} \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} A) > 0. \tag{7.8}$$

We define

$$\xi \stackrel{\text{def}}{=} \frac{\operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} \mathcal{A})}{\operatorname{Tr}(\vec{X}^T \vec{X})}. \tag{7.9}$$

By eqs. (7.7) and (7.8), we have $\xi > 0$.

We now pick $p_u = \frac{\alpha}{\sqrt{d_u}}$ for all $u \in V$, where we will optimize the choice of α shortly. For this choice, we have

$$\begin{aligned}
\mathbb{E}\Big[|I|\Big] &\geq \alpha \sum_u \frac{1}{\sqrt{d_u}} \|\vec{x}_u\|^2 - \frac{1}{2}\alpha^2 \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} \mathcal{A}) \\
&\geq \frac{\alpha}{\sqrt{d}} \sum_u \|\vec{x}_u\|^2 - \frac{1}{2}\alpha^2 \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} \mathcal{A}) \\
&= \mu\left(\frac{\alpha}{\sqrt{d}} - \frac{1}{2}\alpha^2 \underbrace{\frac{\operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} \mathcal{A})}{\operatorname{Tr}(\vec{X}^T \vec{X})}}_{\xi}\right)
\end{aligned}$$

This expression is maximized when $\alpha = \frac{1}{\xi \cdot \sqrt{d}}$, for which it becomes:

$$\mathbb{E}\Big[|I|\Big] \geq \frac{\mu}{2d}\frac{1}{\xi}. \tag{7.10}$$

We know that for the seed set $\mathcal{S}^*$ chosen by Theorem 7.13 satisfies:

$$\begin{aligned}
\operatorname{Tr}(\vec{X}^T \Pi_S^{\perp} \vec{X}) + \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X}(I + \mathcal{A})) &\leq \frac{\operatorname{Tr}(\vec{X}^T \vec{X}(I + \mathcal{A}))}{\lambda'} \\
&= \frac{1}{\lambda'} \operatorname{Tr}(\vec{X}^T \vec{X}) = \mu
\end{aligned}$$

where $\lambda' = (1 - \varepsilon) \min\{\lambda_{r+1}(I + \mathcal{A}), 1\} = (1 - \varepsilon) \min\{2 - \lambda_{n-r-1}(\mathcal{L}), 1\}$.
On the other hand,

$$\begin{aligned}
\frac{\operatorname{Tr}(\vec{X}^T \Pi_S^{\perp} \vec{X}) + \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X}(I + \mathcal{A}))}{\operatorname{Tr}(\vec{X}^T \vec{X})} &= \frac{\operatorname{Tr}(\vec{X}^T \Pi_S^{\perp} \vec{X}) + \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X}) + \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} \mathcal{A})}{\operatorname{Tr}(\vec{X}^T \vec{X})} \\
&= \frac{\operatorname{Tr}(\vec{X}^T \vec{X}) + \operatorname{Tr}(\vec{X}^T \Pi_S \vec{X} \mathcal{A})}{\operatorname{Tr}(\vec{X}^T \vec{X})} = 1 + \xi.
\end{aligned}$$

Thus, for such $\mathcal{S}^*$ we have $\xi \leq \frac{1}{\lambda'} - 1$. Substituting this back into eq. (7.10):

$$\mathbb{E}\Big[|I|\Big] \geq \frac{\mu}{2d} \frac{1}{1/\lambda' - 1}. \qquad \square$$

7.6 Variance Reduction Perspective

In this section, we will offer an alternative perspective to interpret our seed based rounding procedure given in this chapter. Let's focus on $0/1$ case for sake of simplicity. For any set S and any of its labeling, $f \in \{0,1\}^S$, the distribution corresponding to our rounding is the following: For each $u \in V$, there is an associated independent random variable $\mathbf{X}_u$ with

$$\mathbb{E}\big[\mathbf{X}_u\big] = \text{Prob}\big[\mathbf{X}_u = 1\big] = \|\vec{x}_{u|S(f)}\|^2, \ \text{Prob}\big[\mathbf{X}_u = 0\big] = \|\vec{x}_{u(0)|S(f)}\|^2.$$

Here

$$\vec{x}_{u|S(f)} \stackrel{\text{def}}{=} \frac{\langle \vec{x}_u, \vec{x}_{S(f)} \rangle}{\|\vec{x}_{S(f)}\|^2}.$$

Then we can express its squared variance as:

$$\begin{aligned}\text{Var}\big[\mathbf{X}_u\big]^2 =& \text{Prob}\big[\mathbf{X}_u = 1\big](1 - \mathbb{E}\big[\mathbf{X}_u\big])^2 + \text{Prob}\big[\mathbf{X}_u = 0\big](0 - \mathbb{E}\big[\mathbf{X}_u\big])^2 \\ =& \mathbb{E}\big[\mathbf{X}_u\big](1 - \mathbb{E}\big[\mathbf{X}_u\big]) = \|\vec{x}_{u|S(f)}\|^2(1 - \|\vec{x}_{u|S(f)}\|^2).\end{aligned}$$

Roughly this measures how close the quantity $\|\vec{x}_{u|S(f)}\|^2$ is to $\{0,1\}$. So we need to show that for some choice of f, it is small. Let's try sampling sampling f with probability $\|\vec{x}_{S(f)}\|^2$, which leads us to a familiar expression:

$$\mathbb{E}_f\Big[\text{Var}\big[\mathbf{X}_u\big]^2\Big] = \|\Pi_S^{\perp}\vec{x}_u\|^2.$$

In other words, we just proved that for some $f \in \{0,1\}^S$, the variance of $\mathbf{X}_u$ is bounded by the projection distance $\|\Pi_S^{\perp}\vec{x}_u\|^2$:

$$\text{Var}\big[\mathbf{X}_u\big]^2 \leq \|\Pi_S^{\perp}\vec{x}_u\|^2.$$

Chapter 8

Maximum Cut, Unique Games and Similar Problems

In this chapter, we obtain approximation algorithms for Unique Games type problems in terms of constraint graph spectrum. This chapter is intended to be the second part of Chapter 7 and we will heavily rely on it, therefore we assume reader is familiar with it.

8.1 Introduction

Let us quickly recall the definition of the Unique Games problem. An instance of Unique Games consists of a graph $G = (V_0, E, W)$, $n = |V_0|$, with *non-negative edge weights* w_e for each edge $e \in E$, a label set $[k]$, and bijection constraints $\pi_e : [k] \to [k]$ for each edge $e = \{u, v\}$. The goal is to find a labeling $f : V_0 \to [k]$ that minimizes the number of unsatisfied constraints, where $e = \{u, v\}$ is unsatisfied if $\pi_{e(f(u))} \neq f(v)$ (we assume the label of the lexicographically smaller vertex u is projected by π_e). Maximum cut is a special case of Unique Games in which there are two labels, $k = 2$, and all constraints consist of inequalities, $\pi_e(1) = 2, \pi_e(2) = 1$: In other words, we want to find a partition which cuts *as many edges* as possible.

Remark 8.1. *Unique Games can also be captured in the quadratic integer programming framework of Chapter 7, where the matrix A defining the objective function corresponds to the Laplacian of the "lifted graph" $\widehat{G}$ with vertex set $V_0 \times [k]$ obtained by replacing each edge in G by a matching corresponding to its permutation constraint. However, except for the problem of maximum cut, we are unable to apply the results from that section directly because there is no known way to relate the r^{th} eigenvalue of the constraint graph to say*

the $\mathrm{poly}(r)^{th}$ eigenvalue of the lifted graph. Hence we use the "projection distance" type bound based on column selection (similar to Section 4.4), after constructing an appropriate embedding to relate the problem to the original graph. □

Remark 8.2. *Although we do not explicitly mention in the theorem statements, we can provide similar guarantees in the presence of constraints similar to graph partitioning problems such as*

- *constraining labels available to each node,*
- *constraining fraction of labels used among different subsets of nodes.*

For example, the guarantee for maximum cut algorithm immediately carries over to maximum bisection with guarantees on partition sizes similar to minimum bisection. □

8.2 Related Work

The Lasserre SDPs seem very powerful, and as mentioned earlier, for problems shown to be hard assuming the UGC (such as beating Goemans-Williamson for Max Cut), integrality gaps are not known even for a small constant number of rounds. A gap instance for Unique Games is known if the Lasserre constraints are only *approximately* satisfied Khot et al. [2010]. It is interesting to contrast this with our positive result. The error needed in the constraints for the construction in Khot et al. [2010] is $r/(\log\log n)^c$ for some $c < 1$, where n is the number of vertices and r the number of rounds. Our analysis requires the Lasserre consistency constraints are met exactly. In fact, even our solver from Chapter 5 can produce valid Lasserre SDP solutions in time $(k)^{O(r)} n^{O(1)} \log^{O(1)}(1/\varepsilon_0)$ with an objective value at most ε_0 more than optimal.

There are *mixed* hierarchies, which are weaker than Lasserre and based on combining an LP characterized by local distributions (from the Sherali-Adams hierarchy) with a simple SDP, that have been used for several approximation algorithms. Raghavendra [2008] proved that for every constraint satisfaction problem, assuming the Unique Games conjecture, the best approximation ratio is achieved by a small number of levels from the mixed hierarchy.

In an independent work, Barak et al. [2011] consider the above-mentioned mixed hierarchy, and extend the local propagation rounding of Arora et al. [2008a] to these SDPs in a manner similar to our work. Their analysis methods are rather different from ours. Instead of column-based low-rank matrix approximation, they use the graph spectrum to infer global correlation amongst the SDP vectors

from local correlation, and use it to iteratively to argue that a random seed set works well in the rounding. Their main result is an *additive approximation for Max 2-CSPs*. Translating to the terminology used in this paper, given a 2-CSP instance over domain size k with optimal value (fraction of satisfied constraints) equal to v, they give an algorithm to find an assignment with value $v - O\left(k\sqrt{1-\lambda_r}\right)$ based on $r' \gg kr$ rounds of the mixed hierarchy. (Here λ_r is the r'th smallest eigenvalue of the normalized Laplacian of the *constraint* graph; note though that λ_r needs to be fairly close to 1 for the bound to kick in.) For the special case of Unique Games, they get the better performance of $v - O\left(\sqrt[4]{1-\lambda_r}\right)$ which doesn't degrade with k, and also a factor $O(1/\lambda_r)$ approximation for minimizing the number of unsatisfied constraints in time *exponential* in k.

For 2CSPs, our results only apply to a restricted class (corresponding to PSD quadratic forms), but we get approximation-scheme style *multiplicative* guarantees for the harder *minimization objective*, and can handle *global linear constraints*. (Also, for Unique Games, our algorithm has running time *polynomial* in the number of labels k and $2^{O(r)}$, whereas runtime of [Barak et al., 2011] has exponential dependence on k, $2^{O(k)}$.) Our approach enables us to get approximation-scheme style guarantees for several notorious graph partitioning problems that have eluded even APX-hardness.

Using techniques similar to [Barak et al., 2011], Raghavendra and Tan [2012] gave rounding algorithms achieving Goemans-Williamson style approximation factors with global cardinality constraints up to error $1 \pm \varepsilon$.

8.3 Maximum Cut

We first start with the simplest problem fitting in the framework for unique games — finding a maximum cut in a graph. Our algorithm also works for the case of maximum bisection with guarantee on the partition size similar to minimum bisection. We use the following standard integer programming formulation. Note that this formulation is for the complementary objective of finding minimum uncut:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{e=\{u,v\}\in E} w_e \cdot \frac{1}{2}\left[(\mathbf{x}_{(u,1)} - \mathbf{x}_{(v,2)})^2 + (\mathbf{x}_{(u,2)} - \mathbf{x}_{(v,1)})^2\right], \\ \text{st } & \mathbf{x}_{(u,1)} + \mathbf{x}_{(u,2)} = 1 \quad \forall u \in V_0, \\ & \mathbf{x} \in \{0,1\}^{V_0 \times [2]}, \end{aligned}$$

and the corresponding r' rounds of Lasserre Hierarchy relaxation (where we directed each edge by adding $(u,v),(v,u)$ for each $\{u,v\} \in E$):

$$\min_x \frac{1}{2} \sum_{e=(u,v)\in E} w_e \left(x_{\{(u,1)\}} + x_{\{(v,2)\}} - 2x_{\{(u,1),(v,2)\}}\right)$$
$$\text{st } x \text{ is a pseudo moment sequence.}$$

Theorem 8.3 (Maximum Cut / Minimum Uncut)**.** *Given a non-negative weighted undirected graph $G = (V, E, W)$ with $V = V_0$, for all $\varepsilon \in (0,1)$ and a positive integer r, let* OPT *be the minimum total weight of uncut edges over all subsets of V. There exists an algorithm which, in time $n^{O\left(\frac{1}{\varepsilon}\right)} 2^{O\left(\frac{r}{\varepsilon^2}\right)}$, finds a subset of V whose total weight of uncut edges is at most*

$$\leq \mathsf{OPT} \min\left\{1 + \frac{2+\varepsilon}{\lambda_{r+1}(\mathcal{L})}, \frac{1+\varepsilon}{\min\{2 - \lambda_{n-r-1}(\mathcal{L}), 1\}}\right\}.$$

Proof. Our algorithm is the following: We run the algorithms from both Theorems 7.13 and 8.4, and output the better solution. Running time is obvious. Moreover first bound on weight of uncut edges follows from the more general result for Unique Games given in Theorem 8.4, so we focus on the second bound claiming an approximation ratio of $(1+\varepsilon)/\min\{2 - \lambda_{n-r-1}, 1\}$.

The Laplacian matrix, $\widehat{L}$ corresponding to the lifted graph, $\widehat{G}$, for minimum uncut can be expressed as:

$$\widehat{L} = \begin{pmatrix} D & -A \\ -A^T & D \end{pmatrix} = \begin{pmatrix} D & -A \\ -A & D \end{pmatrix},$$

with normalized Laplacian matrix being:

$$\widehat{\mathcal{L}} = \begin{pmatrix} I & -\mathcal{A} \\ -\mathcal{A} & I \end{pmatrix}.$$

Let $\vec{X} \stackrel{\text{def}}{=} [\vec{x}_{u(i)}]_{u\in V_0, i\in[2]}$ be the matrix with columns $\vec{x}_{u(i)}$ and $\vec{X}(i)$ be the matrix with $\vec{X}(i) = [\vec{x}_{u(i)}]_{u\in V_0}$ for fixed $i \in [2]$.

By direct substitution, it is easy to see that, for every eigenvector q_i of constraint graph's normalized Laplacian matrix, $\mathcal{L}$, there are two corresponding eigenvectors for $\widehat{\mathcal{L}}$, $\begin{pmatrix} \frac{1}{\sqrt{2}} q_i \\ \frac{1}{\sqrt{2}} q_i \end{pmatrix}$ and $\begin{pmatrix} \frac{1}{\sqrt{2}} q_i \\ -\frac{1}{\sqrt{2}} q_i \end{pmatrix}$ with corresponding eigenvalues given by λ_i and

$2 - \lambda_i$ respectively. As a convention, we will refer to the first type of eigenvectors as even eigenvectors and the latter type as odd eigenvectors.

For any node $u \in V_0$, we can express $\vec{x}_{u(i)}$ for $i \in [2]$ as

$$\vec{x}_{u(i)} = \|\vec{x}_{u(i)}\|^2 \vec{x}_\emptyset + (-1)^i \|\vec{x}_{u(1)}\| \|\vec{x}_{u(2)}\| y_u \ ,$$

where y_u is a unit vector orthogonal to $\vec{x}_\emptyset$, $\langle \vec{x}_\emptyset, y_u \rangle = 0$. For any set S, $\Pi_S^\perp \vec{x}_{u(1)} = \Pi_S^\perp(\vec{x}_\emptyset^\perp \vec{x}_{u(1)}) = \Pi_S^\perp y_u = -\Pi_S^\perp \vec{x}_{u(2)}$. Consequently for the matrix $\vec{X} = [\vec{x}_{u(1)}, \vec{x}_{u(2)}]_{u \in V} \in \mathbb{R}^{\Upsilon, V}$, $\vec{X}^T \Pi_S^\perp \vec{X}$ has *zero* correlation with even eigenvectors of $\widehat{L}$. Therefore we have the following identity:

$$\mathrm{Tr}(\vec{X}^T \Pi_S^\perp \vec{x}_\emptyset^\perp \Pi_S^\perp \vec{X} \widehat{L}) = \mathrm{Tr}(\vec{X}(1)^T \Pi_S^\perp \vec{x}_\emptyset^\perp \Pi_S^\perp \vec{X}(1)(D + A)).$$

In particular, we can slightly modify Theorem 7.7 to take into account only the eigenvectors of $\widehat{\mathcal{L}}$ with which $\vec{x}_\emptyset^\perp \vec{X}$ has non-zero correlation. Using the bound on total weight of edges cut from Theorem 7.13, we see that the fraction of "uncut" edges is bounded by $(1+\varepsilon)\frac{\mathsf{OPT}}{\min(\lambda_{r+1}(I+\mathcal{A}),1)}$. The proof is now complete by noting that $\lambda_{r+1}(I + \mathcal{A}) = 2 - \lambda_{n-r-1}(\mathcal{L})$. □

8.4 Unique Games

In this section, we prove our main result for approximating Unique Games. We consider the following IP formulation:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \sum_{e=\{u,v\} \in E} w_e \cdot \frac{1}{2} \sum_{i \in [k]} (\mathbf{x}_{u(i)} - \mathbf{x}_{v(\pi_e(i))})^2, \\ \text{subject to} \quad & \sum_{i \in [k]} \mathbf{x}_{u(i)} = 1 \quad \forall u \in V_0, \\ & \mathbf{x} \in \{0,1\}^{V_0 \times [k]}. \end{aligned}$$

Theorem 8.4 (Unique Games)**.** *Let $G = (V_0, E, W)$, $n = |V_0|$, be an instance of Unique Games on label set $[k]$ with permutation constraints π_e for each $e \in E$. Suppose* OPT *is the total weight of unsatisfied constraints in the optimal labeling.*

For any $\varepsilon \in (0,1)$ and positive integer r, there exists an algorithm that finds a labeling $f : V_0 \to [k]$ in time $n^{O(1)} k^{O\left(\frac{r}{\varepsilon}\right)}$ whose total weight of unsatisfied constraints is at most:

$$\leq \left(1 + \frac{2+\varepsilon}{\lambda_{r+1}(\mathcal{L})}\right) \mathsf{OPT}$$

Proof. Our algorithm is very similar to Theorem 7.13 with only one iteration of seed selection. The crucial difference lies in how we choose our seed set: Instead of choosing columns from matrix with columns $\vec{x}_{u(i)}$, we embed each vector "bundle" $\{\vec{x}_{u(i)}\}_{i\in[k]}$, over all u, to a single vector X_u using Theorem 8.5 and we choose columns from the matrix $X = [X_u]_{u\in V}$. After choosing seed set from this embedding, the rounding algorithm proceeds as usual.

As usual, we will start by bounding the total weight of unsatisfied constraints for fixed seed set S. For $[\vec{x}_{T(g)}]$ being labeling vectors for moment sequence with objective value $\eta \leq \mathsf{OPT}$ we have:

$$\eta = \frac{1}{4}\sum_{e=(u,v)\in E} w_e \sum_f \|\vec{x}_{u(f)} - \vec{x}_{v(\pi_e(f))}\|^2.$$

where for notational convenience we treat each undirected edge $\{u, v\}$ as two directed edges of half the weight.

The indicator vector of labeling chosen randomly from distribution $\|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2$ as described in Definition 7.1, $\mathbf{x} \sim \|\vec{x}_{\circ|\mathcal{S}^*(\circ)}\|^2$, we can bound the expected weight of unsatisfied constraints, η', using Claim 7.4 as:

$$\begin{aligned}
\eta' &= \frac{1}{2}\sum_{e=(u,v)\in E} w_e \mathrm{Prob}_{\mathbf{x}\sim\|\vec{x}_{\mathcal{S}^*(f)}\|^2}\left[\exists i \in [k] : \mathbf{x}_{u(i)} \neq \mathbf{x}_{v(\pi_{e(i)})}\right] \\
&= \frac{1}{2}\sum_{e=(u,v)\in E} w_e \sum_f \langle \Pi_{\mathcal{S}^*}\vec{x}_{u(f)}, \Pi_{\mathcal{S}^*(\vec{x}_\emptyset - \vec{x}_{v(\pi_e(f))})}\rangle \\
&= \frac{1}{2}\sum_{e=(u,v)\in E} w_e \sum_f \|\vec{x}_{u(f)}\|^2 - \langle \vec{x}_{u(f)}, \vec{x}_{v(\pi_e(f))}\rangle + \langle \Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}, \Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{v(\pi_e(f))}\rangle \\
&= \eta + \frac{1}{2}\sum_{e=(u,v)\in E} w_e \sum_f \langle \Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}, \Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{v(\pi_e(f))}\rangle \\
&\leq \eta + \frac{1}{2}\sum_{e=(u,v)\in E} w_e \sum_f \frac{\|\Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}\|^2 + \|\Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{v(f)}\|^2}{2} = \eta + \frac{1}{2}\sum_u d_u \sum_f \|\Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}\|^2.
\end{aligned}$$

Recall that for $P_{\mathcal{S}^*}$ being the projection matrix onto $\mathrm{span}\{\vec{x}_{v(f)}\}_{v\in\mathcal{S}^*, f\in[k]}$, $\|\Pi^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}\|^2 \leq \|P^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}\|^2$. Substituting this back into above bound, we obtain:

$$\leq \eta + \frac{1}{2}\sum_u d_u \sum_f \|P^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}\|^2 = \eta\left(1 + \frac{\frac{1}{2}\sum_u d_u \sum_f \|P^{\perp}_{\mathcal{S}^*}\vec{x}_{u(f)}\|^2}{\frac{1}{4}\sum_{e=(u,v)\in E}\sum_f w_e \|\vec{x}_{u(f)} - \vec{x}_{v(\pi_e(f))}\|^2}\right).$$

Since we chose our seed set on matrix $X = [X_u]$ whose columns, X_u, were obtained by embedding $\{\vec{x}_{u(f)}\}_{f\in[k]} \mapsto X_u$ as given in Theorem 8.5:

$$\leq \eta\left(1 + \frac{\frac{1}{2}\sum_u d_u \|X_{\mathcal{S}^*}^{\perp} X_u\|^2}{\frac{1}{8}\sum_{e=(u,v)\in E} w_e \|X_u - X_v\|^2}\right) = \eta\left(1 + 4\frac{\mathrm{Tr}(X^T X_{\mathcal{S}^*}^{\perp} X D)}{2\,\mathrm{Tr}(X^T X L)}\right)$$

If we further scale X by $D^{1/2}$ so that $X' = D^{1/2}X$, we can rewrite final bound as:

$$= \eta\left(1 + 2\frac{\mathrm{Tr}(X'^T X'^{\perp}_{\mathcal{S}^*} X')}{\mathrm{Tr}(X'^T X' \mathcal{L})}\right)$$

where $\mathcal{L}$ is the normalized Laplacian matrix. Since $\mathcal{S}^*$ was chosen using column selection, we apply the bound from Lemma 7.6 and finish the proof. □

Theorem 8.5 (A useful embedding)**.** *Given vectors $[\vec{x}_{u(i)}]_{u\in V_0, i\in[k]}$ with the property that, for any $u \in V_0$, whenever $f, g \in [k]^u$ are two different labelings of u, $f \neq g$,*

$$\langle \vec{x}_{u(f)}, \vec{x}_{u(g)} \rangle = 0;$$

there exists an embedding $\{\vec{x}_{u(f)}\}_{f\in[k]^u} \mapsto X_u$ satisfying the following:

1. *For any $u \in V_0$, $\|X_u\|^2 = \sum_f \|\vec{x}_{u(f)}\|^2$.*
2. *For any $u, v \in V_0$ and any permutation $\pi : [k] \to [k]$,*

$$\sum_{i\in[k]} \|\vec{x}_{u(i)} - \vec{x}_{v(\pi(i))}\|^2 \geq \frac{1}{2}\|X_u - X_v\|^2.$$

3. *For any set $S \subseteq V_0$ and any node $u \in V_0$, if we let P_S be the projection matrix onto the span of $\{\vec{x}_{s(f)}\}_{s\in S, f\in[k]}$:*

$$\|X_S^{\perp} X_u\|^2 \geq \sum_{f\in[k]} \|P_S^{\perp} \vec{x}_{u(f)}\|^2.$$

Our embedding is as follows. Assume that the vectors $\vec{x}_{u(f)}$ belong to $\mathbb{R}^m$. Let $e_1, e_2, \ldots, e_m \in \mathbb{R}^m$ be the standard basis vectors. Define $X_u \in \mathbb{R}^m \otimes \mathbb{R}^m$ as

$$X_u = \sum_{i=1}^{m} \sum_{f\in[k]} \langle \overline{\vec{x}_{u(f)}}, e_i \rangle \vec{x}_{u(f)} \otimes e_i \ .$$

Observation 8.6. *For vectors $x, y \in \mathbb{R}^m$, $\sum_{i=1}^{m} \langle x, e_i\rangle\langle y, e_i\rangle = \langle x, y\rangle$.*

Proof of Theorem 8.5. The first property of the vectors X_u follows from above observation easily:

$$\begin{aligned}
\|X_u\|^2 &= \sum_i \sum_{f,g} \langle \overline{\vec{x}_{u(f)}}, e_i\rangle \langle \overline{\vec{x}_{u(g)}}, e_i\rangle \langle \vec{x}_{u(f)}, \vec{x}_{u(g)}\rangle \\
&= \sum_{f,g} \langle \vec{x}_{u(f)}, \vec{x}_{u(g)}\rangle \sum_i \langle \overline{\vec{x}_{u(f)}}, e_i\rangle \langle \overline{\vec{x}_{u(g)}}, e_i\rangle \\
&= \sum_{f,g} \langle \vec{x}_{u(f)}, \vec{x}_{u(g)}\rangle \langle \overline{\vec{x}_{u(f)}}, \overline{\vec{x}_{u(g)}}\rangle \\
&= \sum_f \|\vec{x}_{u(f)}\|^2.
\end{aligned}$$

We prove second and third properties in Claims 8.7 and 8.8 respectively. □

Claim 8.7. *For any permutation $\pi : [k] \to [k]$,*

$$\frac{1}{2}\|X_u - X_v\|^2 \le \sum_{i\in[k]} \|\vec{x}_{u(i)} - \vec{x}_{v(\pi(i))}\|^2.$$

Proof. Without loss of generality, we assume π is the identity permutation. We have

$$\begin{aligned}
\frac{1}{2}\|X_u - X_v\|^2 &= \frac{\|X_u\|^2 + \|X_v\|^2}{2} - \langle X_u, X_v\rangle \\
&= \frac{\|X_u\|^2 + \|X_v\|^2}{2} - \sum_{f,g} \langle \vec{x}_{u(f)}, \vec{x}_{v(g)}\rangle \sum_i \langle \overline{\vec{x}_{u(f)}}, e_i\rangle \langle \overline{\vec{x}_{v(g)}}, e_i\rangle \\
&= \sum_f \frac{\|\vec{x}_{u(f)}\|^2 + \|\vec{x}_{v(f)}\|^2}{2} - \sum_{f,g} \langle \overline{\vec{x}_{u(f)}}, \overline{\vec{x}_{v(g)}}\rangle^2 \|\vec{x}_{u(f)}\| \|\vec{x}_{v(g)}\|
\end{aligned}$$

The sum over all pairs is lower bounded by summing only the corresponding pairs:

$$\begin{aligned}
&\le \frac{1}{2}\sum_f \left(\|\vec{x}_{u(f)}\|^2 + \|\vec{x}_{v(f)}\|^2 - 2\langle \vec{x}_{u(f)}, \vec{x}_{v(f)}\rangle \langle \overline{\vec{x}_{u(f)}}, \overline{\vec{x}_{v(f)}}\rangle\right) \\
&= \frac{1}{2}\sum_f \|\vec{x}_{u(f)} - \vec{x}_{v(f)}\|^2 + \sum_f \langle \vec{x}_{u(f)}, \vec{x}_{v(f)}\rangle \underbrace{\left(1 - \langle \overline{\vec{x}_{u(f)}}, \overline{\vec{x}_{v(f)}}\rangle\right)}_{\ge 0} \qquad (8.1)
\end{aligned}$$

Since the coefficient of $\langle \vec{x}_{u(f)}, \vec{x}_{v(f)} \rangle$ is positive, we can use Cauchy-Schwarz inequality to replace $\langle \vec{x}_{u(f)}, \vec{x}_{v(f)} \rangle$ with $\|\vec{x}_{u(f)}\| \cdot \|\vec{x}_{v(f)}\|$ in eq. (8.1) to obtain:

$$\leq \frac{1}{2} \sum_f \|\vec{x}_{u(f)} - \vec{x}_{v(f)}\|^2 + \sum_f \left(\|\vec{x}_{u(f)}\| \cdot \|\vec{x}_{v(f)}\| - \langle \vec{x}_{u(f)}, \vec{x}_{v(f)} \rangle \right) \quad (8.2)$$

Using inequality $\|\vec{x}_{u(f)}\| \cdot \|\vec{x}_{v(f)}\| \leq \frac{1}{2} \left(\|\vec{x}_{u(f)}\|^2 + \|\vec{x}_{v(f)}\|^2 \right)$ on eq. (8.2):

$$\leq \frac{1}{2} \sum_f \left(\|\vec{x}_{u(f)} - \vec{x}_{v(f)}\|^2 + \|\vec{x}_{u(f)}\|^2 + \|\vec{x}_{v(f)}\|^2 - 2\langle \vec{x}_{u(f)}, \vec{x}_{v(f)} \rangle \right)$$
$$= \sum_f \|\vec{x}_{u(f)} - \vec{x}_{v(f)}\|^2. \qquad \square$$

Claim 8.8.

$$\|X_S^\perp X_u\|^2 \geq \sum_f \|P_S^\perp \vec{x}_{u(f)}\|^2.$$

Proof. For any $\theta \in \mathbb{R}^S$:

$$\|X_u - \sum_v \theta_v X_v\|^2 = \sum_{i=1}^m \left\| \sum_f \langle \overline{\vec{x}_{u(f)}}, e_i \rangle \vec{x}_{u(f)} - \underbrace{\sum_{v \in S, g} \theta_v \langle \overline{\vec{x}_{v(g)}}, e_i \rangle \vec{x}_{v(g)}}_{P_S \Theta_i} \right\|^2. \quad (8.3)$$

Substituting $\alpha_f = P_S^\perp \vec{x}_{u(f)}$ and $\beta_f = P_S \vec{x}_{u(f)}$, eq. (8.3) is equal to:

$$= \sum_{i=1}^m \left\| \sum_f \langle \overline{\vec{x}_{u(f)}}, e_i \rangle (\alpha_f + \beta_f) - P_S \Theta_i \right\|^2$$
$$= \sum_{i=1}^m \left\| \sum_f \langle \overline{\vec{x}_{u(f)}}, e_i \rangle \alpha_f \right\|^2 + \left\| \sum_f \langle \overline{\vec{x}_{u(f)}}, e_i \rangle \beta_f - P_S \Theta_i \right\|^2$$
$$\geq \sum_{i=1}^m \left\| \sum_f \langle \overline{\vec{x}_{u(f)}}, e_i \rangle \alpha_f \right\|^2$$
$$= \sum_{f,g} \langle \alpha_f, \alpha_g \rangle \sum_{i=1}^m \langle \overline{\vec{x}_{u(f)}}, e_i \rangle \langle \overline{\vec{x}_{u(g)}}, e_i \rangle$$
$$= \sum_{f,g} \langle \alpha_f, \alpha_g \rangle \langle \overline{\vec{x}_{u(f)}}, \overline{\vec{x}_{u(g)}} \rangle = \sum_f \|\alpha_f\|^2 = \sum_f \|P_S^\perp \vec{x}_{u(f)}\|^2. \qquad \square$$

This concludes the proof of Theorem 8.5, therefore also the proof of Theorem 8.4.

Chapter 9

Sparsest Cut and Other Expansion Problems

In this chapter, we give another approximation algorithm for non-uniform sparsest cut. We obtain the following guarantee: For any $\varepsilon, \delta \in (0,1)$, given input graph G and demand graph H on vertex set V, we can find a cut $U \subseteq V$ with $\frac{w_G(U,V\setminus U)}{w_H(U,V\setminus U)}$ at most $\frac{1+\varepsilon}{\delta}$ times the optimal non-uniform sparsest cut value, in time $2^{r/(\delta\varepsilon)}\,\mathrm{poly}(n)$ provided $\lambda_r(G;H) \geq \Phi^*(G;H)/(1-\delta)$, where $\Phi^*(G;H)$ is the sparsity of the optimal cut. As mentioned in Chapter 2, we use $\lambda_r(G;H)$ to denote the r'th smallest generalized eigenvalue of the Laplacian of G w.r.t that of H; $w_G(U, V\setminus U)$ (resp. $w_H(U, V\setminus U)$) as the weight of edges crossing the $(U, V\setminus U)$ cut in G (resp. H). In other words, we show that non-uniform sparsest is easy when the generalized spectrum grows moderately fast. To the best of our knowledge, there were no results based on higher order spectra for non-uniform sparsest cut prior to this work.

Even for uniform sparsest cut, the quantitative aspects of our result are somewhat stronger than previous methods. Similar results hold for other expansion measures like edge expansion, normalized cut, and conductance, with the r'th smallest eigenvalue of the *normalized* Laplacian playing the role of $\lambda_r(G)$ in the latter two cases.

Our proof is based on an ℓ_1-embedding of vectors from a semi-definite program from the Lasserre hierarchy. The embedded vectors are then rounded to a cut using standard threshold rounding. Another aspect of the analysis is the adaptation of the column selection paradigm from Chapter 7 to pick a set of *edges* rather than nodes. This feature is important in order to extend the algorithms to non-uniform sparsest cut.

9.1 Related Work

As the sparsest cut problem (and closely related variants such as edge expansion and conductance) are all NP-hard in general, theoretically much effort has gone into the design of good approximation algorithm for the problem.

The hard direction of Cheeger's inequality shows that in d-regular graphs one can "round" the eigenvector corresponding to $\lambda_2(G)$ to a cut U satisfying $\Phi_U(G) \leq \sqrt{8d\lambda_2(G)}$. This gives $O(\sqrt{d/\Phi^*(G)}) \leq O(\sqrt{d/\lambda_2(G)})$ approximation to Uniform Sparsest Cut which is good for moderate values of $\Phi^*(G)$.

For smaller values of $\Phi^*(G)$, the best approximation for sparsest cut is based on solving a convex relaxation of the problem, and then rounding the solution to a cut. Using linear programming (LP), in a seminal work Leighton and Rao [1988] gave a factor $O(\log n)$ approximation for Non-Uniform Sparsest Cut (here n denotes the number of vertices). Beautiful connections of approximation sparsest cut to embeddings of metric spaces into the ℓ_1-metric were later discovered by Linial et al. [1995] and Aumann and Rabani [1998]. Using a semi-definite programming (SDP) relaxation, the approximation ratio was improved to $O(\sqrt{\log n})$ for Uniform Sparsest Cut in the breakthrough work of Arora et al. [2009]. For Non-Uniform Sparsest Cut, using ℓ_1 embeddings of negative type metrics, an approximation factor of $O(\log^{3/4} n)$ was obtained by Chawla et al. [2008] and a factor $\tilde{O}(\sqrt{\log n})$, nearly matching the Uniform Sparsest Cut case, was obtained by Arora et al. [2008b].

Recently, higher order eigenvalues were used to approximate many graph partitioning problems. In Chapter 7, we already saw an algorithm based on SDPs from the Lasserre hierarchy achieving an approximation factor of the form $(1 + \varepsilon)/\min\{1, \tilde{\lambda}_r\}$ for problems such as minimum bisection, small set expansion, etc. On a similar front, for Uniform Sparsest Cut problem, if r^{th} eigenvalue is large relative to expansion, we show how to combine the eigen-space enumeration of Arora et al. [2010] with cut improvement of Andersen and Lang [2008][1] to obtain a constant factor approximation for Uniform Sparsest Cut in time $n^{O(1)}2^{O(r)}$. For details of such algorithm, see Section 9.8. We will revisit this approach in Section 9.2 to show why it does not work for Non-Uniform Sparsest Cut.

Our result in this chapter follows the common theme of our thesis: One can obtain a constant factor approximation with running time being a function of how fast the spectrum grows. Put it differently, one can identify a generic condition which highlights what kind of graphs are easy.

[1]We thank anonymous reviewers for bringing this paper to our attention.

Best to our knowledge, in case of Non-Uniform Sparsest Cut with arbitrary demand graph H, no such result was known. In fact not even the analogue of harder direction for Cheeger inequality existed, let alone spectrum based approximation schemes. In this chapter, we present such an approximation scheme.

9.2 Our Contribution

Let us now describe the main steps of our algorithm at a high level. We will focus on unweighted cost graph $G = (V, E_G)$ and demand graph $H = (V, E_H)$ for concreteness. We solve the standard r rounds of Lasserre SDP relaxation (see Section 9.4 for details). This relaxation yields one vector for any possible event on up to r variables and they have the same properties with corresponding random variables. To be more precise, for each subset $S \in \binom{V}{\leq r}$, for each 0-1 labeling of S, $f \in \{0,1\}^S$, there is a vector $\vec{x}_{S(f)} \in \mathbb{R}^\Upsilon$ (think of $\|\vec{x}_{S(f)}\|^2$ as the "probability" of observing f on S) with following properties:

1. (Probability Measure) $\|\vec{x}_\emptyset\|^2 = 1$;
2. (Disjoint Events) $\langle \vec{x}_{S(f)}, \vec{x}_{T(g)} \rangle = 0$ if f and g are inconsistent, i.e. if $f(S \cap T) \neq g(S \cap T)$;
3. (Joint Events) $\langle \vec{x}_{S(f)}, \vec{x}_{T(g)} \rangle = \|\vec{x}_{S \cup T}(h)\|^2$ where $h \in \{0,1\}^{S \cup T}$ is such that $h(S) = f$ and $h(T) = g$ if $f(S \cap T) = g(S \cap T)$;
4. (Proper Distribution) $\sum_f \vec{x}_{S(f)} = \vec{x}_\emptyset$.

Given this point of view, a tempting approach for rounding would be to find an event, conditioning upon which would force the resulting vectors to look more binary, say by measuring the average variation on atomic events. Viewing it in yet another perspective, we try to find a small set of "seed" variables, S, such that any local assignment to them can be propagated to other variables globally in a rather unique way.

Surprisingly it turns out the structure of these vectors allow us to relate this idea into a nice and intuitive geometric quantity. In Chapter 7, we saw that for any r-subset S, if one chooses $f \in \{0,1\}^S$ at random with probability $\|\vec{x}_{S(f)}\|^2$ and conditions on $S(f)$, then the resulting variation is bounded by how closely the span of vectors $\{\vec{x}_{S(f)}\}_{f \in \{0,1\}^S}$ approximates atomic events, $\{\vec{x}_{u(1)}\}$. This observation allowed us to reduce the problem to that of choosing r column subset of a

matrix best approximating it under Frobenius norm for which we proved optimal upper and lower bounds in Chapter 10.

Let's go back to Non-Uniform Sparsest Cut problem with cost graph G and demand graph H. If we adopt the above strategy, we are left with the problem of simultaneously relating the variations along edges of G and H with nodes V. On a high level, this means we need an orthonormal matrix which diagonalizes Laplacian matrices of G and H at the same time, which means they had the same eigen-space to begin with – definitely not the case for arbitrary G and H.

This reason also rules out any subspace enumeration based strategy, which at least worked for uniform sparsest cut case (see Section 9.8): Basically, we need to identify an r-dimensional subspace for which there is some close optimal solution in ℓ_2 norm. Then, again we run into the previous problem of finding an orthonormal matrix diagonalizing Laplacians of G and H simultaneously.

Our main contribution in this paper is the following. Instead of naïvely conditioning on labelings of individual nodes, which forces our atomic events to be the nodes, we instead condition on cutting edges from demand graph H, and choosing an appropriate partitioning such that it will never cut more edges of G in expectation than SDP solution. In turn, we relate the variation along edges of H to edges of G using generalized eigenvectors between their graph Laplacians, and observe that in the space of our atomic events, edges of H, the generalized eigenvectors are indeed orthonormal. Finally we prove how to choose best set of r edges from H by adopting Algorithm 10.2.

9.3 A Useful Lower Bound on Trace

The following is crucial for bounding the approximation ratio by graph spectra. Recall that $\sigma_j(\vec{X})$ denotes the j'th *largest* eigenvalue of $\vec{X}$.

Proposition 9.1. *Given two symmetric positive semi-definite matrices $X, Y \in \mathbb{S}_+ V$, assume the intersection of their null space, $U \subset \mathbb{R}^V$, has $\geq m$ dimensions. Then the following holds:*

$$\mathrm{Tr}(XY) \geq \sum_{j=1}^{n-m} \sigma_j(X)\lambda_{j+m}(Y).$$

Proof. Note that if $z \in U$, then z is an eigenvector of both X and Y with eigenvalue 0. Without loss of generality, we assume 0 eigenvectors of X and Y are rotated and ordered such that

$$z_1(X) = z_1(Y), z_2(X) = z_2(Y), \ldots, z_m(X) = z_m(Y) \in U.$$

If $i \leq m$, then $\langle z_i(X), z_i(Y)\rangle = 1$ and $\langle z_i(X), z_j(Y)\rangle = 0$ for all $j \neq i$. Consider expressing $\mathrm{Tr}(XY)$ in the following way:

$$\mathrm{Tr}(XY) = \sum_{i,j} \langle z_i(X), z_j(Y)\rangle^2 \lambda_i(X)\lambda_j(Y) \tag{9.1}$$

$$= \sum_{i \leq m} \langle z_i(X), z_i(Y)\rangle^2 \cdot 0 + \sum_{i>m, j>m} \langle z_i(X), z_j(Y)\rangle^2 \lambda_i(X)\lambda_j(Y) \tag{9.2}$$

Since $\sum_{j>m} \langle z_i(X), z_j(Y)\rangle^2 = 1$ for all $i > m$, we can use the Birkhoff-von Neumann theorem to express eq. (9.2) as a convex combination of all permutations $\{m+1, \ldots, n\}$, $\mathrm{sym}(V \setminus [m])$:

$$= \sum_{\pi \in \mathrm{sym}(V \setminus [m])} \theta_\pi \sum_{i>m} \lambda_i(X)\lambda_{\pi(i)}(Y) \tag{9.3}$$

Note that $\sum_{i>m} \lambda_i(X)\lambda_{\pi(i)}(Y)$ is minimized when π reverses the ordering between $\lambda_i(X)$'s and $\lambda_j(Y)$'s (i.e. it matches $\sigma_{i-m}(X)$ with $\lambda_i(X)$ for all $i > m$). Thus we can lower bound eq. (9.3) as:

$$\geq \sum_{\pi \in \mathrm{sym}(V \setminus [m])} \theta_\pi \sum_{i>m} \sigma_{i-m}(X)\lambda_i(Y)$$

$$= \sum_{i>m} \sigma_{i-m}(X)\lambda_i(Y) \sum_{\pi \in \mathrm{sym}(V \setminus [m])} \theta_\pi = \sum_{i>m} \sigma_{i-m}(X)\lambda_i(Y). \qquad \square$$

9.4 Sparsest Cut Problem and Its Relaxation

We first give the formal definition of non-uniform sparsest cut problem.

Definition 9.2 (Non-Uniform Sparsest Cut)**.** *Given two graphs G (cost graph) and H (demand graph) on node set V, for any proper binary partitioning whose indicator vector is given $\mathbf{x} \in \{0,1\}^V$, the sparsity of this partition is defined as:*

$$\Phi_{\mathbf{x}}(G, H) \stackrel{\text{def}}{=} \frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}},$$

with $\Phi^*(G, H) \stackrel{\text{def}}{=} \min_{\mathbf{x}} \Phi_{\mathbf{x}}(G, H)$ being minimum over all such U. We will refer to $\Phi^*(G, H)$ as sparsity of G with respect to H.

Uniform sparsest cut corresponds to demand graph being the normalized clique K.

Definition 9.3 (Uniform Sparsest Cut and Edge Expansion)**.** *For a given graph G, uniform sparsest cut (*Uniform Sparsest Cut*) corresponds to the sparsest cut problem with demand graph $H = K$ (the normalized clique from Notation 2.53). We denote uniform sparsity of $\mathbf{x}$ with $\Phi_{\mathbf{x}}(G)$ and minimum $\Phi_{\mathbf{x}}(G)$ over all $\mathbf{x}$ with $\Phi^*(G)$.*

The edge expansion of $\mathbf{x}$ in G is defined as

$$\phi_{\mathbf{x}}(G) \stackrel{\text{def}}{=} \frac{\mathbf{x}^T L_G \mathbf{x}}{\min\left(\|\mathbf{x}\|_1, n - \|\mathbf{x}\|_1\right)}$$

with $\phi^*(G)$ being the minimum over all $\mathbf{x}$. We will refer $\phi^*(G)$ as the expansion of graph G.

Claim 9.4. *The following SDP is an r-rounds of basic Lasserre Hierarchy relaxation for* Non-Uniform Sparsest Cut *problem:*

$$\begin{aligned} &\textit{Minimize} && \textstyle\sum_{u<v} w^G_{u,v} \|\vec{x}_u - \vec{x}_v\|^2, \\ &\textit{subject to} && \textstyle\sum_{u<v} w^H_{u,v} \|\vec{x}_u - \vec{x}_v\|^2 = 1, \\ & && \langle \vec{x}_S, \vec{x}_T \rangle = x_{S \cup T}, \\ & && x_\emptyset \geq \frac{1}{\sum_{u,v} w^H_{u,v}}, \\ & && x \in \mathbb{R}^{V_{\leq 2r}}. \end{aligned} \tag{9.4}$$

We will use $\vec{x}_{S(f)}$ to denote the vectors as in Definition 3.42.

Proof. Given $\mathbf{x} \in \{0,1\}^V$, for each S, f set $\vec{x}_{S(f)} \leftarrow \frac{1}{\sqrt{\mathbf{x}^T L_H \mathbf{x}}} \begin{cases} 1 & \text{if } \mathbf{x}_S = f, \\ 0 & \text{else.} \end{cases}$ It is easy to verify that this is indeed a feasible solution with objective value $\mathbf{x}^T L_G \mathbf{x} / \mathbf{x}^T L_H \mathbf{x}$. □

9.5 Partitioning Algorithm and Its Analysis

In this section, we will show how to cast Algorithms 7.1 and 7.2 in the framework we introduced in Chapter 5 so as to improve the running time. First observe that our labeling procedure, Algorithm 7.1, will work without any modification in our framework. We only need to make minor modifications for seed selection procedure, Algorithm 7.2. The new seed selection algorithm is given in Algorithm 7.3.

In this section, we present our basic algorithm in Algorithm 9.1. Running time bound and sparsity of output are proven in Theorem 9.12. In the following section, we will show how to combine this with our faster solver.

Notation 9.5. *Throughout the whole section, we will use*

$$\Pi_S \stackrel{\text{def}}{=} \sum_{f\in\{0,1\}^S} \overline{\vec{x}_{S(f)}} \cdot \overline{\vec{x}_{S(f)}}^T$$

to denote the projection matrix onto span of vectors $\{\vec{x}_{S(f)}\}_{f\in\{0,1\}^S}$ *where* S *is the set chosen by Algorithm 9.1 in Item b.*

Proposition 9.6 (Seed Based ℓ_1 Embedding)**.** *For any fixed* S*, the vectors* $(y_u)_{u\in V}$ *constructed in Item a of Algorithm 9.1 have the following property. For any pair* $u, v \in V$*,*

$$\|\Pi_S(\vec{x}_{u(1)} - \vec{x}_{v(1)})\|^2 \le \|y_u - y_v\|_1 \le \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|^2.$$

Proof. We will first prove the upper bound. Using identity $\vec{x}_{a(1)} = \vec{x}_{a(1),b(1)} + \vec{x}_{a(1),b(0)}$, we have $\vec{x}_{u(1)} - \vec{x}_{v(1)} = \vec{x}_{u(1),v(0)} - \vec{x}_{u(0),v(1)}$. Therefore

$$\begin{aligned}
\|y_u - y_v\|_1 =& \sum_f \left|\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle\right| = \sum_f \left| \underbrace{\langle \vec{x}_{S(f)}, \vec{x}_{u(1),v(0)}\rangle}_{\ge 0} - \underbrace{\langle \vec{x}_{S(f)}, \vec{x}_{u(0),v(1)}\rangle}_{\ge 0}\right| \\
\le& \sum_f \langle \vec{x}_{S(f)}, \vec{x}_{u(1),v(0)}\rangle + \langle \vec{x}_{S(f)}, \vec{x}_{u(0),v(1)}\rangle \\
=& \langle \vec{x}_{\emptyset}, \vec{x}_{u(1),v(0)}\rangle + \langle \vec{x}_{\emptyset}, \vec{x}_{u(0),v(1)}\rangle = \|\vec{x}_{u(1),v(0)}\|_2^2 + \|\vec{x}_{u(0),v(1)}\|_2^2.
\end{aligned}$$

Since $\langle \vec{x}_{u(1),v(0)}, \vec{x}_{u(0),v(1)}\rangle = 0$, we have $\|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|^2 = \|\vec{x}_{u(1),v(0)} - \vec{x}_{u(0),v(1)}\|^2 = \|\vec{x}_{u(1),v(0)}\|^2 + \|\vec{x}_{u(0),v(1)}\|^2$ which proves the upper bound.

In order to prove the lower bound, note that for any f

$$\begin{aligned}
\left|\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle\right| \le& \langle \vec{x}_{S(f)}, \vec{x}_{u(1),v(0)}\rangle + \langle \vec{x}_{S(f)}, \vec{x}_{u(0),v(1)}\rangle \\
=& \|\vec{x}_{S(f),u(1),v(0)}\|^2 + \|\vec{x}_{S(f),u(0),v(1)}\|^2 \\
\le& \|\vec{x}_{S(f),u(1),v(0)}\|^2 + \|\vec{x}_{S(f),u(0),v(1)}\|^2 \\
&+ \|\vec{x}_{S(f),u(1),v(1)}\|^2 + \|\vec{x}_{S(f),u(0),v(0)}\|^2 = \|\vec{x}_{S(f)}\|^2.
\end{aligned}$$

Therefore, for any $\vec{x}_{S(f)} \neq 0$:

$$\|\vec{x}_{S(f)}\|^2 \langle \overline{\vec{x}_{S(f)}}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle^2 = \left|\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle\right|^2 \le \|\vec{x}_{S(f)}\|^2 \left|\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle\right|.$$

Summing over f,

$$\|y_u - y_v\|_1 = \sum_f \left|\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle\right| \geq \sum_f \overline{\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} - \vec{x}_{v(1)}\rangle}^2 = \left\|\Pi_S(\vec{x}_{u(1)} - \vec{x}_{v(1)})\right\|^2.$$

□

Notation 9.7. *Let $\vec{X} = [\vec{x}_{u(1)}]_{u\in V}$ be the matrix whose columns are vectors $\vec{x}_{u(1)}$ for each $u \in V$ so that*

$$\sum_{u<v} w^G_{u,v}\|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|^2 = \mathrm{Tr}(\vec{X}^T\vec{X}L_G) \quad \textit{and} \quad \sum_{u<v} w^H_{u,v}\|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|^2 = \mathrm{Tr}(\vec{X}^T\vec{X}L_H).$$

Let $B_H \stackrel{\text{def}}{=} \sqrt{W_H}B$ so that $L_H = {B_H}^T B_H$. Observe that $\sqrt{W_H}$ is a well defined since W_H is diagonal and non-negative.

Observation 9.8. *The matrices $\vec{X}$ given in Notation 9.7 and $\vec{X}$ as constructed in Algorithm 9.1 satisfies the following: $\vec{X} = \vec{X}B_H^T$. Also*

$$\|\vec{X}\|_F^2 = \mathrm{Tr}(\vec{X}^T\vec{X}L_H) = \sum_{u<v} w^H_{u,v}\|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|^2 .$$

Lemma 9.9 (Converting Edges to Seeds)**.** *For any $\widehat{S}$, the set S constructed in Item a of Algorithm 9.1 has size $|S| \leq 2r'$ and it satisfies*

$$\sum_{u<v} w^H_{u,v}\left\|\Pi_S(\vec{x}_{u(1)} - \vec{x}_{v(1)})\right\|_2^2 \geq \left\|\vec{X}^{\Pi}_{\widehat{S}}\vec{X}\right\|_F^2.$$

Proof. By construction of S, $|S| \leq 2|\widehat{S}| \leq 2r'$. The matrix $\vec{X}$ satisfies $\|\vec{X}\|_F^2 = \mathrm{Tr}(\vec{X}^T\vec{X}) = \mathrm{Tr}(\vec{X}^T\vec{X}L_H)$ using Notation 9.7 and Observation 9.8. We know that each column of $\vec{X}_{\widehat{S}}$ is a linear combination of columns in $\vec{X}_S = [\vec{x}_{u(1)}]_{u\in S}$, $\vec{X}^{\Pi}_{\widehat{S}} \preceq \vec{X}^{\Pi}_S$. Furthermore for each $u \in S$, the corresponding column of $\vec{X}_S$, $\vec{x}_{u(1)}$, can be expressed as a linear combination of matrix Π_S, as in Chapter 7, which implies $\vec{X}^{\Pi}_{\widehat{S}} \preceq \vec{X}^{\Pi}_S \preceq \Pi_S$. Consequently

$$\left\|\vec{X}^{\Pi}_{\widehat{S}}\vec{X}\right\|_F^2 \leq \left\|\Pi_S\vec{X}\right\|_F^2 = \sum_{u<v} w^H_{u,v}\|\Pi_S(\vec{x}_{u(1)} - \vec{x}_{v(1)})\|^2.$$

□

Lemma 9.10 (Edge Selection). *For any positive real ε and positive integer r with $r' \geq \frac{r}{\varepsilon} + r - 1$, the set $\widehat{S}$ chosen in Item a satisfies*

$$\left(\lambda_{r+1}(G,H) - \lambda_2(G,H)\right) \left\| \vec{X}^{\perp}_{\widehat{S}} \vec{X} \right\|_F^2 \leq (1+\varepsilon)\Big(\sum_{u<v} w^G_{u,v} \left\| \vec{x}_{u(1)} - \vec{x}_{v(1)} \right\|^2 - \lambda_2(G,H) \sum_{u<v} w^H_{u,v} \left\| \vec{x}_{u(1)} - \vec{x}_{v(1)} \right\|^2 \Big) . \tag{9.5}$$

Proof. In Theorem 10.1, it was shown that using algorithm Algorithm 10.2 to choose $\frac{r}{\varepsilon} + r - 1$ columns from matrix $\vec{X}$, the columns we obtain $\widehat{S} \subseteq \binom{V}{2}$ will have the following property:

$$\left\| \vec{X}^{\perp}_{\widehat{S}} \vec{X} \right\|_F^2 \leq (1+\varepsilon) \sum_{j \geq r+1} \sigma_j(\vec{X}^T \vec{X}), \tag{9.6}$$

where σ_j is the j^{th} largest eigenvalue of matrix $\vec{X}^T \vec{X}$. In order to bound $\sum_{j \geq r+1} \sigma_j(\vec{X}^T \vec{X})$, consider the matrix $C \stackrel{\text{def}}{=} L_G^{1/2} L_H{}^{\dagger} B_H^T \in \mathbb{R}^{V, \binom{V}{2}}$:

$$\begin{aligned} \mathrm{Tr}\left(\vec{X}^T \vec{X} C^T C \right) &= \mathrm{Tr}\left(\vec{X} C^T C \vec{X}^T \right) \\ &= \mathrm{Tr}\left(\vec{X} B_H L_H{}^{\dagger} L_G L_H{}^{\dagger} B_H^T \vec{X}^T \right) \\ &= \mathrm{Tr}(\vec{X} B_H^T B_H L_H{}^{\dagger} L_G L_H{}^{\dagger} B_H^T B_H \vec{X}^T) \end{aligned} \tag{9.7}$$

Using the identity $B_H^T B_H = L_H$ and the fact that $L_H{}^{\dagger} L_H = L_H L_H^{\dagger}$ is a projection matrix, we can upper bound the middle matrix in eq. (9.7) as $B_H^T B_H L_H{}^{\dagger} L_G L_H{}^{\dagger} B_H^T B_H \preceq L_G$.

$$\leq \mathrm{Tr}(\vec{X} L_G \vec{X}^T) = \mathrm{Tr}(\vec{X}^T \vec{X} L_G) . \tag{9.8}$$

Our next task is to lower bound $\mathrm{Tr}(\vec{X}^T \vec{X} C^T C)$ in terms of the spectrum of $\vec{X}^T \vec{X}$ and the generalized eigenvalues $\lambda_j(G,H)$. Observe that right null space of matrix B^T has $M \stackrel{\text{def}}{=} \binom{n}{2} - n + 1$ dimensions since the matrix $B^T B = I - \frac{1}{n} \mathbb{1}\mathbb{1}^T$ has $n-1$ non-zero eigenvalues. Furthermore right null space of B^T is common to both $\vec{X}$ and C as they are multiplied by B^T on the right, which means the intersection of null spaces of $\vec{X}^T \vec{X}$ and $C^T C$ have at least M dimensions. Using Proposition 9.1, we conclude that

$$\mathrm{Tr}\left(\vec{X}^T \vec{X} C^T C \right) \geq \sum_{j=1}^{n-1} \sigma_j(\vec{X}^T \vec{X}) \lambda_{j+M}(C^T C).$$

Finally consider the n-by-n matrix $CC^T = L_G^{1/2} L_H^\dagger L_H L_H^\dagger L_G^{1/2} = L_G^{1/2} L_H^\dagger L_H^{1/2}$, whose eigenvalues are same with largest n eigenvalues of $C^T C$. Therefore

$$\lambda_{j+M}(C^T C) = \lambda_{j+1}(CC^T) = \lambda_{j+1}(L_G^{1/2} L_H^\dagger L_G^{1/2})$$

Both $L_G^{1/2}$ and $L_H^{\dagger/2}$ are symmetric, so $\lambda_{j+1}(L_G^{1/2} L_H^\dagger L_G^{1/2}) = \lambda_{j+1}(L_H^{\dagger/2} L_G^{1/2} L_G^{1/2} L_H^{\dagger/2})$:

$$= \lambda_{j+1}(L_H^{\dagger/2} L_G L_H^{\dagger/2}) = \lambda_{j+1}(G, H).$$

Putting it all together:

$$\begin{aligned}
\mathrm{Tr}\left(\vec{X}^T \vec{X} C^T C\right) &\geq \sum_{j=1}^{n-1} \sigma_j(\vec{X}^T \vec{X}) \lambda_{j+1}(G, H) \\
&= \lambda_2(G, H) \sum_{j=1}^{n-1} \sigma_j(\vec{X}^T \vec{X}) + \sum_{j=1}^{n-1} \left(\lambda_{j+1}(G, H) - \lambda_2(G, H)\right) \sigma_j(\vec{X}^T \vec{X}) \\
&\geq \lambda_2(G, H) \sum_{j=1}^{n-1} \sigma_j(\vec{X}^T \vec{X}) + \left(\lambda_{r+1}(G, H) - \lambda_2(G, H)\right) \sum_{j \geq r+1} \sigma_j(\vec{X}^T \vec{X})
\end{aligned} \tag{9.9}$$

Using $\sigma_j(\vec{X}^T \vec{X}) = 0$ for $j \geq n$, we have $\sum_{j=1}^{n-1} \sigma_j(\vec{X}^T \vec{X}) = \|\vec{X}\|_F^2$. So we can lower bound eq. (9.9) as:

$$\begin{aligned}
&\geq \lambda_2(G, H) \|\vec{X}\|_F^2 + \left(\lambda_{r+1}(G, H) - \lambda_2(G, H)\right) \sum_{j \geq r+1} \sigma_j(\vec{X}^T \vec{X}) \\
&= \lambda_2(G, H)\, \mathrm{Tr}(\vec{X}^T \vec{X} L_H) + \left(\lambda_{r+1}(G, H) - \lambda_2(G, H)\right) \sum_{j \geq r+1} \sigma_j(\vec{X}^T \vec{X})
\end{aligned} \tag{9.10}$$

using Observation 9.8. Substituting the bound $\mathrm{Tr}\left(\vec{X}^T \vec{X} C^T C\right) \leq \mathrm{Tr}(\vec{X}^T \vec{X} L_G)$ from eq. (9.8) and the lower bound on $\sum_{j \geq r+1} \sigma_j(\vec{X}^T \vec{X})$ from eq. (9.6) finishes the proof of the bound eq. (9.5). □

We produce the final partitioning by choosing the best threshold cut. The following gives sufficient conditions for such thresholding to work. We skip the proof of this well known lemma.

Lemma 9.11 (ℓ_1 Embeddings to Partitions [Linial et al., 1995]). *The indicator vector* $\mathbf{x}$ *produced by Algorithm 9.1 in Item b satisfies* $\mathbf{x}^T L_H \mathbf{x} > 0$ *and*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}} \leq \frac{\sum_{u<v} w^G_{u,v} \|y_u - y_v\|_1}{\sum_{u<v} w^H_{u,v} \|y_u - y_v\|_1}.$$

Theorem 9.12 (Main result: Approximating Non-Uniform Sparsest Cut). *For any positive integer* r *and positive real* $\varepsilon > 0$*, for any pair of simple, undirected and connected graphs* G, H *with non-negative edge weights such that* $\lambda_r(G,H) - \lambda_2(G,H) > (1+\varepsilon)(\Phi^*(G,H) - \lambda_2(G,H))$*, Algorithm 9.1 for the choice* $r' = O\left(r + \frac{r}{\varepsilon}\right)$ *and feasible solution to eq. (9.4) outputs* $\mathbf{x} \in \{0,1\}^V$ *in time* $n^{O(1)} 2^{O\left(r+\frac{r}{\varepsilon}\right)}$ *such that* $\mathbf{x}^T L_H \mathbf{x} > 0$ *and*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}} \leq \frac{\Phi^*(G,H)}{1 - (1+\varepsilon)\frac{\Phi^*(G,H) - \lambda_2(G,H)}{\lambda_r(G,H) - \lambda_2(G,H)}}.$$

Proof of Running Time. Let $r' = O\left(r + \frac{r}{\varepsilon}\right)$. Algorithm 10.2 can be performed in $\mathrm{poly}(n)$. The subsequent steps all take time at most $2^{O(r')}\mathrm{poly}(n)$, and so does the running time. □

Proof of Sparsity. Let $\Phi_{SDP} = \frac{\sum_{u<v} w^G_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2}{\sum_{u<v} w^H_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2}$. We know that $\Phi_{SDP} \leq \Phi^*$. Consider the set S chosen and vectors y constructed. Using Proposition 9.6:

$$\frac{\sum_{u<v} w^G_{u,v} \|y_u - y_v\|_1}{\sum_{u<v} w^H_{u,v} \|y_u - y_v\|_1} \leq \frac{\sum_{u<v} w^G_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2}{\sum_{u<v} w^H_{u,v} \|\Pi_S(\vec{x}_{u(1)} - \vec{x}_{v(1)})\|_2^2},$$

Applying the bounds from Lemma 9.9 and Lemma 9.10 successively:

$$\sum_{u<v} w^H_{u,v} \|\Pi_S^{\perp}(\vec{x}_{u(1)} - \vec{x}_{v(1)})\|_2^2 \leq (1+\varepsilon)\frac{\sum_{u<v} w^G_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2 - \lambda_2 \sum_{u<v} w^H_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2}{\lambda_r - \lambda_2}.$$

where we use $\lambda_i = \lambda_i(G;H)$ for notational simplicity. Finally applying Lemma 9.11, we obtain the following:

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}} \leq \frac{\sum_{u<v} w^G_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2}{\sum_{u<v} w^H_{u,v} \|\vec{x}_{u(1)} - \vec{x}_{v(1)}\|_2^2 \left(1 - (1+\varepsilon)\frac{\Phi_{SDP} - \lambda_2}{\lambda_r - \lambda_2}\right)} = \frac{\Phi_{SDP}}{1 - (1+\varepsilon)\frac{\Phi_{SDP} - \lambda_2}{\lambda_r - \lambda_2}}.$$

Substituting $\Phi_{SDP} \leq \Phi^*(G,H)$ completes our proof. □

Corollary 9.13 (Certifying Non-Uniform Sparsest Cut). *For any positive integer r and positive real $\varepsilon > 0$, for any pair of simple, undirected graphs G, H with non-negative edge weights, one of the following holds. For every $\delta \in (0,1)$, in time $n^{O\left(r+\frac{r}{\delta\varepsilon}\right)}$:*

- *Either one can find a partitioning $\mathbf{x} \in \{0,1\}^V$ such that $\mathbf{x}^T L_H \mathbf{x} > 0$ and*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}} \leq \frac{(1+\varepsilon)}{\delta} \Phi^*(G,H),$$

- *Or $\Phi^*(G,H) > (1-\delta)\lambda_r(G,H)$.*

Proof. We can invoke Theorem 9.12 with parameters r and $O(\delta\varepsilon)$, which in time $n^{O(r+r/(\delta\varepsilon))}$ will output vectors (y_u).

If $\frac{\sum_{u<v} w^G_{u,v}\|y_u - y_v\|_1}{\sum_{u<v} w^H_{u,v}\|y_u - y_v\|_1} \leq \frac{1+\varepsilon}{\delta}\Phi^*(G,H)$, then by Lemma 9.11 using thresholding on each coordinate of vectors $[y_u]$, we can find an indicator vector $\mathbf{x} \in \{0,1\}^V$ such that $\mathbf{x}^T L_H \mathbf{x} > 0$ and $\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}} \leq \frac{\sum_{u<v} w^G_{u,v}\|y_u - y_v\|_1}{\sum_{u<v} w^H_{u,v}\|y_u - y_v\|_1} \leq \frac{1+\varepsilon}{\delta}\Phi^*(G,H)$. Else we have

$$\frac{1+\varepsilon}{\delta}\Phi^*(G,H) < \frac{\sum_{u<v} w^G_{u,v}\|y_u - y_v\|_1}{\sum_{u<v} w^H_{u,v}\|y_u - y_v\|_1} \leq \frac{\Phi^*(G,H)}{1-(1+O(\delta\varepsilon))\frac{\Phi^*(G,H)-\lambda_2}{\lambda_r-\lambda_2}},$$

so that

$$\delta > \frac{1+\varepsilon}{1+o(1)}\left[1-(1+O(\delta\varepsilon))\frac{\Phi^*-\lambda_2}{\lambda_r-\lambda_2}\right] \implies \frac{\Phi^*-\lambda_2}{\lambda_r-\lambda_2} > \frac{1-\delta\frac{1+o(1)}{1+\varepsilon}}{1+O(\delta\varepsilon)} \geq 1-\delta. \qquad \square$$

For $H = K$, we can can see that Theorem 9.12 and Corollary 9.13 hold for uniform sparsest cut problem with eigenvalues $\lambda_j(G,K)$ corresponding to the usual eigenvalues of graph G's Laplacian matrix, $\lambda_j(G,K) = \lambda_j(L_G)$.

We can also adapt the same algorithm to certify graph expansion in terms of $\lambda_j(L_G)$'s. The algorithm is the same with Algorithm 9.1 with the demand graph being a clique, $H = K$.

Corollary 9.14 (Certifying Graph Expansion). *For any positive integer r and $\varepsilon > 0$, for any simple, undirected graph G with non-negative edge weights, one of the following holds. For every $\delta \in (0,1)$, in time $n^{O\left(r+\frac{r}{\delta\varepsilon}\right)}$,*

- *Either one can find a non-trivial partitioning, given by $\mathbf{x} \in \{0,1\}^V$ such that*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\min(\|\mathbf{x}\|_1, n - \|\mathbf{x}\|_1)} \leq \frac{(1+\varepsilon)}{\delta}\phi^*(G),$$

- *Or* $\phi^*(G) > (1-\delta)\lambda_r(G)$.

Proof. We only highlight the differences. In eq. (9.4), we replace the first constraint with $\sum_u w^H_{u,v}\vec{x}_{u(1)} = \nu\vec{x}_\emptyset$ and constraint ν to be in $(0, n/2]$. We run the rest of Algorithm 9.1 assuming demand graph $H = K$. Observing that $\min(\|\mathbf{x}\|_1, n - \|\mathbf{x}\|_1) = \min(\|\mathbf{x}\|_2^2, n - \|\mathbf{x}\|_2^2) \geq \|\mathbf{x}\|_2^2(1 - \|\mathbf{x}\|_2^2/n) = \mathbf{x}^T L_K \mathbf{x}$ and $\lambda_j(G, K) = \lambda_j(G)$ completes our proof. □

9.6 Combining with Fast Solver

Corollary 9.15 (Final Algorithm for Non-Uniform Sparsest Cut)**.** *For any positive integer r and positive real $\varepsilon > 0$, for any pair of simple, undirected and connected graphs G, H with non-negative edge weights such that $\lambda_r(G, H) - \lambda_2(G, H) > (1+\varepsilon)(\Phi^*(G, H) - \lambda_2(G, H))$, there exists an algorithm that runs in time $2^{O(r/\varepsilon)}n^{O(1)}$ which outputs $\mathbf{x} \in \{0, 1\}^V$ such that $\mathbf{x}^T L_H \mathbf{x} > 0$ and*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_H \mathbf{x}} \leq \frac{\Phi^*(G, H)}{1 - (1+\varepsilon)\frac{\Phi^*(G,H) - \lambda_2(G,H)}{\lambda_r(G,H) - \lambda_2(G,H)}}.$$

Proof. Run Algorithm 5.2 with following input:

- $\varepsilon_0 \leftarrow \varepsilon^{-n}$.
- $\ell \leftarrow 1$.
- **FEASIBLE** is the separation oracle for Lasserre Hierarchy as outlined in Section 5.6.
- **SEED** is Algorithm 9.2.
- **ROUND** is Algorithm 9.3. □

9.7 Bounds for Normalized Cut and Conductance

Two closely related problems to uniform sparsest cut and edge expansion are normalized cut and conductance respectively, where each node in the denominator is weighted proportional to its degree.

Definition 9.16 (Normalized Cut and Conductance)**.** *For a given graph G, normalized cut of G is defined as*

$$\mathrm{ncut}(G) \stackrel{\text{def}}{=} \min_{\emptyset \neq U \subsetneq V} \frac{\sum_{u \in U, v \notin U} w^G_{u,v}}{\left(\sum_{u \in U} d^G_u\right)\left(\sum_{v \notin U} d^G_v\right)/m}, \text{ where } m \stackrel{\text{def}}{=} \sum_{u \in V} d^G_v.$$

Similarly the conductance of G is defined as

$$\mathrm{conductance}(G) \stackrel{\text{def}}{=} \min_{\emptyset \neq U \subsetneq V} \frac{\sum_{u \in U, v \notin U} w^G_{u,v}}{\min\left\{\sum_{u \in U} d^G_u, \sum_{v \notin U} d^G_v\right\}}.$$

By running the algorithm Algorithm 9.1 on demand graph $H = K(G)$, we can obtain similar guarantees for normalized cut and conductance in terms of eigenvalues of the *normalized graph Laplacian*. We first introduce the necessary definitions.

Notation 9.17 (Degree Weighted Clique Graph)**.** *Given graph G, we will use $K(G)$ to denote the degree weighted clique graph of G, where $w^{K(G)}_{u,v} = \frac{d^G_u d^G_v}{m}$ for $u \neq v$. Note that $d^{K(G)}_u = d^G_u\left(1 - \frac{d^G_u}{m}\right)$.*

Proposition 9.18. *For any graph G, the following hold:*

1. $\mathrm{ncut}(G) = \Phi^*(G, K(G))$.
2. *Given* $\mathbf{x} \in \{0,1\}^V$, $\min\left\{\sum_u \mathbf{x}_u d^G_u, \sum_u (1 - \mathbf{x}_u) d^G_u\right\} \geq \mathbf{x}^T L_{K(G)} \mathbf{x}$.
3. $\lambda_j(\mathcal{L}_G) = \lambda_j(G, K(G))$.

Proof. The first two properties are trivial, we will only prove the last one. Since both $\lambda(G, K(G))$'s and $\lambda(\mathcal{L}_G)$ are scale invariant with respect to the edge weights, we assume $m = \sum_u d^G_u = 1$. Let d and D be the vector and diagonal matrix of degrees in G respectively. We will use $\sqrt{d}$ to denote the vector which is the element-wise square root of d. Observe that $\|\sqrt{d}\| = \sqrt{m} = 1$ and $L_{K(G)} = D - dd^T = D^{1/2}(I - \sqrt{d}\sqrt{d}^T)D^{1/2} = D^{1/2}\sqrt{d}^{\perp}D^{1/2}$. Without loss of generality, we can assume $z_1(G, K(G)) \propto \mathbb{1}$. It is easy to verify $\lambda_1(G, K(G)) = \lambda_1(\mathcal{L}(G)) = 0$. Given $j \geq 2$ let $z'_j \leftarrow \left(L^{\dagger}_{K(G)}\right)^{1/2} z_j(G, K(G))$ and $\lambda = \lambda_j(G, K(G))$. Since $z_j \perp \mathbb{1}$ we have $L^{1/2}_{K(G)}(L^{\dagger}_{K(G)})^{1/2} z_j = z_j$ implying $z'_j \perp \mathbb{1}$ and:

$$L_G z'_j = \lambda L_{K(G)} z'_j. \tag{9.11}$$

Now consider $y_j \leftarrow D^{1/2} z'_j$: $D^{1/2} y_j \perp \mathbb{1} \implies y_j \perp z_1(\mathcal{L}_G) \implies \sqrt{d}^{\perp} y_j = y_j$. Substituting this into eq. (9.11) and multiplying on the left with $D^{-1/2}$:

$$\begin{aligned} D^{-1/2} L_G D^{-1/2} y_j =& \lambda D^{-1/2} L_{K(G)} D^{-1/2} y_j \\ \implies \mathcal{L}_G y_j =& \lambda \sqrt{d}^{\perp} y_j = \lambda y_j. \end{aligned}$$

In other words, the **set** of eigenvalues are the same between $\mathcal{L}_G$ and $(L^{\dagger}_{K(G)})^{1/2} L_G (L^{\dagger}_{K(G)})^{1/2}$. To show that multiplicities are the same, we note that the vectors $\{z'_j\}$ are linearly independent since z_j's are orthogonal and all lie in the span of $L_{K(G)}$. Similarly, vectors $\{y_j\}$ are linearly independent as $D^{1/2}$ is invertible. This means the multiplicities are the same in both $\mathcal{L}_G$ and $(L^{\dagger}_{K(G)})^{1/2} L_G (L^{\dagger}_{K(G)})^{1/2}$ as well. □

The following corollaries follow from Theorem 9.12 and Proposition 9.18 in a manner identical to Corollary 9.13 and Corollary 9.14.

Corollary 9.19 (Certificate for Normalized Cut)**.** *For any positive integer r and positive real $\varepsilon > 0$, for any simple, undirected graph G with non-negative edge weights, one of the following holds. For every δ, $0 < \delta < 1$, in time $n^{O\left(r + \frac{r}{\delta\varepsilon}\right)}$,*

- *Either one can find a non-trivial partitioning, given by $\mathbf{x} \in \{0,1\}^V$ such that*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\mathbf{x}^T L_{K(G)} \mathbf{x}} \leq \frac{(1+\varepsilon)}{\delta} \mathrm{ncut}(G)$$

- *Or $\mathrm{ncut}(G) > (1-\delta)\lambda_r(\mathcal{L}_G)$.*

Corollary 9.20 (Certificate for Conductance)**.** *For any positive integer r and positive real $\varepsilon > 0$, for any simple, undirected graph G with non-negative edge weights, one of the following holds. For every δ, $0 < \delta < 1$, in time $n^{O\left(r + \frac{r}{\delta\varepsilon}\right)}$,*

- *Either one can find a non-trivial partitioning, given by $\mathbf{x} \in \{0,1\}^V$ such that*

$$\frac{\mathbf{x}^T L_G \mathbf{x}}{\min\left\{\sum_u \mathbf{x}_u d^G_u, \sum_u (1-\mathbf{x}_u) d^G_u\right\}} \leq \frac{(1+\varepsilon)}{\delta} \mathrm{conductance}(G)$$

- *Or $\mathrm{conductance}(G) > (1-\delta)\lambda_r(\mathcal{L}_G)$.*

9.8 Using Subspace Enumeration for Uniform Sparsest Cut

Throughout this section, we will assume that $G = (V, W)$ is a 1-regular graph for simplicity with nodes $V = V$ and edge weights $W = (w_G(u, v))$. Since G is regular, definitions of uniform sparsest cut / normalized cut and edge expansion / conductance coincide. Thus we will use focus on Uniform Sparsest Cut and edge expansion, which we denote by $\phi_x(G)$ and $\Phi_x(G)$ for given $x \in \{0,1\}^V$.

The following theorem is adapted from Andersen and Lang [2008] for our setting:

Theorem 9.21 (Cut Improvement, see Andersen and Lang [2008])**.** *Given 1-regular graph G with $V(G) = V$ and indicator vector $x^* \in \{0,1\}^V \setminus \{0^V, 1^V\}$, if $x \in \{0,1\}^V \setminus \{0^V, 1^V\}$ is an indicator vector satisfying*

$$0 < \|x\|_1 \leq \frac{n}{2} \text{ and } \frac{\langle x, x^* \rangle}{\|x^*\|_1} > \frac{\|x\|_1}{n}$$

then in polynomial time one can find $y \in \{0,1\}^V \setminus \{0^V, 1^V\}$ whose edge expansion is bounded by:

$$\frac{\phi_y(G)}{\phi_{x^*}(G)} \leq \frac{1 - \|x\|_1/n}{\langle x, x^* \rangle / \|x^*\|_1 - \|x\|_1/n}.$$

The following lemma is adapted from Arora et al. [2010]:

Theorem 9.22 (Eigenspace Enumeration, see Arora et al. [2010])**.** *Given 1-regular graph G, in time $2^{O(r)}n^{O(1)}$, there exists an algorithm which outputs a set $\vec{X} \subset \{0,1\}^V \setminus \{0^V, 1^V\}$ in time $2^{O(r)}n^{O(1)}$ containing some $x \in \vec{X}$ for which there exists $x^* \in \{0,1\}^V$ such that:*

$$\frac{\|x - x^*\|_1}{\|x^*\|_1} \leq \frac{8}{\lambda_r(G)} \frac{x^{*T} L_G x^*}{x^{*T} L_K x^*}.$$

Combining these two, we obtain the following:

Corollary 9.23. *Given 1-regular graph G with $V(G) = V$, for any positive integer r, if r^{th} eigenvalue satisfies $\lambda_r > 8\phi^*$ where Φ^* is the uniform sparsest cut value of G, then in time $n^{O(1)}2^{O(r)}$ one can find a partitioning $y \in \{0,1\}^V \setminus \{0^V, 1^V\}$ such that:*

$$\Phi_y(G) \leq \frac{2\Phi^*}{1 - 8\frac{\Phi^*}{\lambda_r}}.$$

Proof. Let x^* be a uniform sparsest cut with sparsity ϕ having $\|x^*\|_1 \leq \frac{n}{2}$. Then we can find a vector $x \in \{0,1\}^V$ using eigen-space enumeration such that:

$$\frac{\|x^* - x\|_1}{\|x^*\|_1} \leq \delta,$$

where $\delta = \frac{8{x^*}^T L_G x^*}{\lambda_r {x^*}^T L_K x^*}$.

Similarly, complement of x, $1 - x$, satisfies:

$$\frac{\|(1 - x^*) - (1 - x)\|_1}{\|1 - x^*\|_1} = \frac{\|x^* - x\|_1}{n - \|x^*\|_1} \leq \frac{\|x^* - x\|_1}{\|x^*\|_1} \leq \delta.$$

Therefore

$$\frac{\langle x, x^* \rangle}{\|x^*\|_1}, \frac{\langle 1 - x, 1 - x^* \rangle}{\|1 - x^*\|_1} \geq 1 - \delta.$$

We apply cut improvement algorithm to x if $\|x\|_1 \leq \frac{n}{2}$, and to $1 - x$ otherwise. Without loss of generality, let's assume $\|x\|_1 \leq \frac{n}{2}$. Then cut improvement finds y such that:

$$\Phi_y(G) \leq \phi_y(G) \leq \frac{\phi_{x^*}(G)}{1 - \delta} \leq \frac{2\Phi^*}{1 - \delta}.$$

Substituting the expression for δ completes the proof. □

Algorithm 9.1 Given graphs G, H and positive integer r', in time $n^{O(r')}$ this algorithm outputs a partitioning $\mathbf{x} \in \{0,1\}^V$ whose sparsity is bounded in Theorem 9.12.

Procedure: 1. Solve the SDP formulation given in eq. (9.4) for $2r'+2$ rounds. Let $(\vec{x}_T/\|\vec{x}_\emptyset\| \in \mathbb{R}^m)$ be the optimal solution normalized so that $\|\vec{x}_\emptyset\| = 1$.

2. Let $\vec{X} = [\vec{X}_{u,v}]$ be the matrix whose columns $\vec{X}_{u,v}$ are given by $\vec{X}_{u,v} \leftarrow \sqrt{w^H_{u,v}}(\vec{x}_{u(1)} - \vec{x}_{v(1)})$ where $u < v$ for each $\{u,v\} \in \binom{V}{2}$.

3. (Seed selection)

   (a) Use Algorithm 10.2 to choose r' columns, $\widehat{S} \subseteq \binom{V}{2}$, of matrix $\vec{X}$ (corresponding to r' edges of H) so as to (approximately) minimize the matrix reconstruction error $\left\|\vec{X}^{\perp}_{\widehat{S}}\vec{X}\right\|_F$ in Frobenius norm.

   (b) Let $S \leftarrow \left\{u : \exists v, \{u,v\} \in \widehat{S}\right\}$ be the set of both endpoints of each edge in $\widehat{S}$.

4. (Rounding)

   (a) (ℓ_1-embedding) For each $u \in V$, let $y_u \in \mathbb{R}^{\{0,1\}^S}$ be $y_u \leftarrow \left[\langle \vec{x}_{S(f)}, \vec{x}_{u(1)}\rangle\right]_{f\in\{0,1\}^S}$.

   (b) (Threshold cuts) For each $f \in \{0,1\}^S$ and $u \in V$, let $\mathbf{x}_{u,f} = [\mathbf{x}_{u,f}(v)]_{v\in V} \in \{0,1\}^V$ be the indicator vector with $\mathbf{x}_{u,f}(v) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } y_v(f) \geq y_u(f), \\ 0 & \text{else.} \end{cases}$.

   (c) Return $\mathbf{x}$ where $\mathbf{x} \leftarrow \operatorname{argmin}_{u,f} \frac{\mathbf{x}^T_{u,f} L_G \mathbf{x}_{u,f}}{\mathbf{x}^T_{u,f} L_H \mathbf{x}_{u,f}}$.

Algorithm 9.2 Seed selection for sparsest cut problem on graphs G and H.

Input: • Graphs G, H on nodes V with edge weights given by $w_{u,v}^{G,H}$ respectively,

• Moment sequence $x \neq 0$ and positive integer r'.

Output: • Seed set $S \in \binom{V}{\leq r'}$.

Procedure: 1. Let $[\vec{x}_{S(f)}]_{S,f}$ be labeling vectors for moment sequence x as in Definition 3.42.

2. Let $X_{u,v} \leftarrow \sqrt{w_{u,v}^{H}}(\vec{x}_{u(1)} - \vec{x}_{v(1)})$. Use Algorithm 10.2 to choose r' columns, $\widehat{S} \subseteq \binom{V}{2}$, of matrix $\vec{X}$ (corresponding to r' edges of H) so as to (approximately) minimize the matrix reconstruction error $\left\| \vec{X}_{\widehat{S}}^{\perp} \vec{X} \right\|_F$ in Frobenius norm.

3. Return S.

Algorithm 9.3 Seed based threshold rounding for sparsest cut on G and H.

Input: • Graphs G, H on nodes V with edge weights given by $w_{u,v}^{G,H}$ respectively,

• Seed set S and moment sequence $x \neq 0$.

Output: • Seed set $S \in \binom{V}{\leq r'}$.

Procedure: 1. (ℓ_1-embedding) For each $u \in V$, let $y_u \in \mathbb{R}^{\{0,1\}^S}$ be $y_u \leftarrow \left[\langle \vec{x}_{S(f)}, \vec{x}_{u(1)} \rangle\right]_{f \in \{0,1\}^S}$.

2. (Threshold cuts) For each $f \in \{0,1\}^S$ and $u \in V$, let $\mathbf{x}_{u,f} = [\mathbf{x}_{u,f}(v)]_{v \in V} \in \{0,1\}^V$ be the indicator vector with $\mathbf{x}_{u,f}(v) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } y_v(f) \geq y_u(f), \\ 0 & \text{else.} \end{cases}$.

Return $\mathbf{x}$ where $\mathbf{x} \leftarrow \operatorname{argmin}_{u,f} \frac{\mathbf{x}_{u,f}^T L_G \mathbf{x}_{u,f}}{\mathbf{x}_{u,f}^T L_H \mathbf{x}_{u,f}}$.

Chapter 10

Column Based Matrix Reconstruction

Observe that one major component, common to the analysis of all our approximation algorithms presented in this thesis, is that their running is always exponential in r with r being the number of columns one has to choose from an m-by-n matrix so as to approximate it as good as the best rank-k approximation in Frobenius norm. Thus finding the optimal dependence between r and k is a question of natural significance for all our approximation algorithms. Furthermore in order to achieve a running time of the form $2^{O(r)}\,\mathrm{poly}(n)$, the brute force search to find such columns is not an option and we need efficient ways of finding in time polynomial in n.

In this chapter, we prove upper bounds for r linear in k and construct matrices where the dependence between r and k is optimal up to lower order terms. Finally we complement our upper bounds with both a deterministic algorithm with running time $O(rnm^3)$ and a randomized algorithm with running time $O(rnm^2)$.

10.1 Introduction

Given a matrix $X \in \mathbb{R}^{[m]\times[n]}$ and a positive integer $k < n$, the best rank-k approximation to X is given by top k singular vectors of X:

$$X_{(k)} = \sum_{i=1}^{k} \sqrt{\sigma_i} u_i v_i^T$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ are the eigenvalues of $X^T X$, and u_i (resp. v_i) are the associated left (resp. right) singular vectors for each singular value $\sqrt{\sigma_i}$.

Furthermore $X_{(k)}$ can be computed in time $O(\min(n,m)mn)$-time using Singular Value Decomposition (SVD) [Golub and Loan, 1996].

One related question that has received considerable attention in recent years is choosing r *columns* of X, for some input parameter $r \geq k$, whose span approximates X as nearly as well as $X_{(k)}$. In other words, we would like to relate

$$\min_{C \in \binom{[n]}{r}} \|X - X_C^{\Pi} X\|_\xi = \min_{C \in \binom{[n]}{r}} \|X_C^{\perp} X\|_\xi$$

to $\|X - X_{(k)}\|_\xi$ for some norm ξ, and efficiently find a subset C of r columns coming close to this bound. Here X_C denotes matrix formed by columns of X corresponding to C and X_C^{Π} (resp. $X_C^{\perp}$) is the projection matrix onto X_C (resp. onto null space of X_C).

Aside from our theoretical interest, This basic problem seems well-motivated in various application settings. For example, this problem has applications in data sets arising from document classification problems, face recognition tasks, and so on, where it is important to pick a subset of features that are dominant (and it is not appropriate to work with linear combinations of features output by usual dimension reduction techniques like random projection or singular value decomposition). We refer the reader to [Mahoney and Drineas, 2009] for comparisons of SVD and column selection on experimental data.

Our main results in this chapter are the following two theorems. We are able to get the best known dependence between r and k, show its optimality up to lower order terms, and achieve this with an efficient deterministic algorithm (Theorem 10.1). This answers one of the open questions mentioned by Boutsidis et al. [2011]. We are also able to give a more efficient randomized algorithm, via a faster implementation of exact volume sampling (Theorem 10.2). The deterministic algorithm of Theorem 10.1 is a derandomization of the volume sampling algorithm via conditional expectations method of Deshpande and Rademacher [2010].

Theorem 10.1. *Given $X \in \mathbb{R}^{[m]\times[n]}$, and positive integers $k \leq r$, one can find a set C of r columns, deterministically using at most $O(rnm^\omega \log m)$ many arithmetic operations (where ω is the exponent of matrix multiplication), such that*

$$\|X - X_C^{\Pi} X\|_F^2 \leq \frac{r+1}{r+1-k} \|X - X_{(k)}\|_F^2 \,. \tag{10.1}$$

Furthermore, for any $r = o(n)$, this bound is tight up to lower order terms.

Theorem 10.2. *Given a matrix $X \in \mathbb{R}^{[m]\times[n]}$, $m \leq n$, and $r \geq 1$, there is an algorithm **Vol-Sample** that samples a subset of r columns of X, $C \in \binom{[n]}{r}$, with probability*

Paper	r	Ratio	Running Time	Deterministic
This work	$k + \frac{k}{\varepsilon} - 1$	$1+\varepsilon$	$O(rnm^{\omega} \log m)$	Yes
This work	$k + \frac{k}{\varepsilon} - 1$	$1+\varepsilon$	$O(rnm^2)$	No
[1]	$\frac{2k}{\varepsilon}$	$1+\varepsilon$	$O(k\varepsilon^{-1}nm + k^3\varepsilon^{-2/3}n)$	No
[2]	k	$k+1$	$O(knm^{\omega} \log m)$	Yes
[3]	$O(\eta^2(A)k \log k/\varepsilon^2)$	$1+\varepsilon$	$O(k^2mn \log k)$	Yes
[4]	$O(k \log k + k\varepsilon^{-1})$	$1+\varepsilon$	$O((k \log k + k\varepsilon^{-1})mn+$ $(k \log k + k\varepsilon^{-1})^2)(m+n)$	No
[5]	$O(k^2 \log k + k\varepsilon^{-1})$	$1+\varepsilon$	$O(k^2mn \log k)$	No

Table 10.1: Performance and running time of various column selection algorithms. In the table, the papers are referred by [1] = Boutsidis et al. [2011], [2]=Deshpande and Rademacher [2010], [3]=Çivril and Magdon-Ismail [2008], [4]=Sarlós [2006], [5] = Deshpande and Vempala [2006].

$\frac{|X_C^T X_C|}{\sum_{T \in \binom{[n]}{r}} |X_T^T X_T|}$ *using at most* $O\left(rnm^2\right)$ *arithmetic operations. For every* $k \leq r$*, the subset* C *returned by* ***Vol-Sample*** *satisfies*

$$\mathbb{E}_C \left[\|X - X_C^{\Pi}\|_F^2 \right] \leq \frac{r+1}{r+1-k} \|X - X_{(k)}\|_F^2 \, .$$

Note that $\|X - X_C^{\Pi} X\|_F^2 = \|X_C^{\perp} X\|_F^2 = \mathrm{Tr}(X^T X_C^{\perp} X)$. Henceforth in this paper, we will use the Trace notation.

10.2 Related Work

The first algorithm for k-column matrix reconstruction was given in a seminal paper of Frieze et al. [2004], where they presented a randomized algorithm to find $\mathrm{poly}(k, 1/\varepsilon, 1/\delta)$ columns that achieve an *additive* error of $\varepsilon \|X\|_F$. Subsequent works concentrated on removing the additive factor and getting multiplicative (or relative error) guarantees, and improving the dependence between r and k to get a desired relative error. Some of these works are mentioned in Table 10.1. In the table, r is the number of columns needed so as to obtain the given approximation ratio, defined as $\mathrm{Tr}(X^T X_C^{\perp} X)/\|X - X_{(k)}\|_F^2$.

To briefly place our result in context, let us mention the known existential bounds on the relation between r, k, and the ratio achieved. Deshpande et al.

[2006] proved the existence of k columns achieving a ratio $k+1$, and also show that this is best possible up to lower order terms. Deshpande and Vempala [2006] proved that for small $\varepsilon > 0$, there exists a matrix M for which the best error achieved by a rank-k matrix, whose columns are restricted to belong to the span of $r \geq k/\varepsilon$ columns of M, is at least $1+\varepsilon-o(1)$ times the best rank-k approximation.[1]

Until recently, even the best existential bound to achieve $(1+\varepsilon)$ approximation was super-linear in k. In an independent and concurrent work, Boutsidis et al. [2011] proved a bound of $r \approx k + \frac{2k}{\varepsilon}$ along with a randomized algorithm to find such a subset of columns.[2] Our main result proves that $k/\varepsilon + k - 1$ columns are sufficient, and further those columns can be found in *deterministic* polynomial time.

The $(1+\varepsilon)$ approximation achieved by Boutsidis et al. [2011] holds in the restricted model (in which the above-mentioned k/ε lower bound due to Deshpande and Vempala [2006] applies) where one must find a *rank-k approximation matrix* contained in the span of the chosen r columns, whereas our approximating matrix uses the full span of the chosen columns. So our results and [Boutsidis et al., 2011] are incomparable in this respect. We stress though that even allowing for full column span, no bounds on r which were linear in k were known till recently, for achieving say a factor 2 approximation. Further, we extend the lower bound of Deshpande and Vempala [2006] to show that even allowing for full column span, $r = k/\varepsilon$ columns are needed for a factor $(1+\varepsilon-o(1))$ approximation.

Note that our result gives the optimal $(k+1)$ factor approximation (taking $\varepsilon = k$) for $r = k$, and for $\varepsilon \to 0$, the near-optimal $(1+\varepsilon)$ factor for $r \approx k/\varepsilon$, in a uniform way. As for the algorithmic claim, recently Deshpande and Rademacher [2010] gave an efficient implementation of volume sampling and a deterministic algorithm to find a set k columns with approximation ratio $k+1$, thus matching the bound of Deshpande et al. [2006] algorithmically. We simply bound the ratio achieved by this algorithm when it is allowed to pick $r > k$ columns. In other words, the algorithmic part of Theorem 10.1 follows from [Deshpande and Rademacher, 2010], given our combinatorial bound.

Prior to our work, the fastest algorithm known for *exact* volume sampling was given by Deshpande and Rademacher [2010] using $O\left(rnm^{\omega}\log m\right)$ arithmetic operations. We give an asymptotically faster sampling algorithm, by using binary search to pick the lowest index column in the sampled set with the correct

[1] Although the lower bound of Deshpande and Vempala [2006] is stated as $1+\frac{\varepsilon}{2}-o(1)$, the actual lower bound they prove is stronger and equals $1+\varepsilon-o(1)$.

[2] The theorem statement in [Boutsidis et al., 2011] mentions the weaker bound $r \leq 10k/\varepsilon$, but the sharper bound is given at the end of Section 4 of the paper.

marginal probability, and then recursing to sample the remaining $r-1$ columns.

10.3 Our Techniques

Our proof is based on the following bound:

$$\min_{C \in \binom{[n]}{r}} \mathrm{Tr}(X^T X_C^{\perp} X) \leq (r+1)\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)} = \mathbb{E}_{\mathbf{C}\sim\mathcal{C}_r(X)}\left[\mathrm{Tr}(X^T X_{\mathbf{C}}^{\perp} X)\right] \tag{10.2}$$

where $\mathbf{C} \sim \mathcal{C}_r(X)$ denotes sampling C with probability proportional to determinant of $X_C^T X_C$, $\left|X_C^T X_C\right|$, and $\mathbf{S}_r(\sigma)$ is the r'th symmetric function of $\sigma_1, \sigma_2, \ldots, \sigma_n$. The bound eq. (10.2) already appears in the work of Deshpande et al. [2006] where sampling from $\mathcal{C}_r(X)$ is called "volume sampling."

Our main technical contribution is to use the *Schur-concavity* of $\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)}$ and theory of majorization [see Marshall et al., 2009] to bound $\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)}$ in terms of $\sum_{i\geq k+1}\sigma_i$. At an intuitive level, the ratio $\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)}$ should be larger when $\{\sigma_i\}_{i=1}^n$ is more "uniform." Majorization and Schur-concavity allow us to turn this intuition into a precise and formal statement. This leads us to the inequality

$$(r+1)\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)} \leq \frac{r+1}{r+1-k}\sum_{i>k}\sigma_i\,, \tag{10.3}$$

which together with $\|X - X_{(k)}\|_F^2 = \sum_{i>k}\sigma_i$ and eq. (10.2) yields the claimed bound (Equation (10.1)). For the nearly matching lower bound, we prove that for the construction given by Deshpande and Vempala [2006], the lower bound on approximation ratio holds even in the unrestricted model where the full column span of the r columns is allowed; this analysis appears in Section 10.9.

As for the algorithm, Deshpande and Rademacher [2010] used the method of conditional expectations to find $C \in \binom{[n]}{r}$ satisfying $\mathrm{Tr}(X^T X_C^{\perp} X) \leq \mathbb{E}_{\mathbf{C}\sim\mathcal{C}_r(X)}\left[\mathrm{Tr}(X^T X_{\mathbf{C}}^{\perp} X)\right]$ deterministically using $O(rnm^{\omega}\log m)$ operations. Together with our bound eq. (10.3), this implies a deterministic algorithm, given in Algorithm 10.2, achieving a $\frac{r+1}{r+1-k}$ ratio. In light of this, we do not discuss the deterministic part any further in this paper, and focus on proving eqs. (10.2) and (10.3), which we do in Sections 10.5 and 10.6 respectively. Our more efficient volume sampling algorithm is described in Section 10.7. The proof of our lower bound is presented in Section 10.9.

10.4 Preliminaries

In addition to the mathematical background given in Chapter 2, we will also make extensive use of theory of majorization and elementary symmetric polynomials. Since these two are only specific to this chapter, we chose to introduce the relevant background in this section.

Given real vector $a = [a_i]_{i=1}^n \in \mathbb{R}^{[n]}$, we will use $a{\uparrow}_i$ (resp. $a{\downarrow}_i$) to denote the i^{th} smallest (resp. largest) element of $\{a_i\}_i$.

Notation 10.3 (Determinants)**.** *For any symmetric matrix $A \in \mathbb{S}^{[n]}$, we will use $|A|$ to denote A's determinant.*

Notation 10.4 (Majorization)**.** *We say $a = [a_i]_{i=1}^n \in \mathbb{R}^{[n]}$* majorizes *$b = [b_i]_{i=1}^n \in \mathbb{R}^{[n]}$ if for all $j \in [n]$, $\sum_{j' \le j} a{\downarrow}_{j'} \geq \sum_{j' \le j} b{\downarrow}_{j'}$ and $\sum_j a_j = \sum_j b_j$. We denote this relation by $a \succ_m b$.*

Observation 10.5. *For any non-negative vector $a \in \mathbb{R}_+^{[n]}$, the following holds:*

$$(1, 0, \ldots, 0) \succ_m \frac{1}{\sum_i a_i} a \succ_m \left(\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n}\right)$$

Definition 10.6 (Schur Concavity)**.** *A function $F : \mathbb{R}^{[n]} \to \mathbb{R}$ is called Schur-concave if whenever $a \in \mathbb{R}^{[n]}$ majorizes $b \in \mathbb{R}^{[n]}$, i.e. $a \succ_m b$, then $F(a) \le F(b)$.*

Definition 10.7 (Symmetric polynomials)**.** *For a given $\sigma = [\sigma_i]_{i=1}^n \in \mathbb{R}^{[n]}$, let $\mathbf{S}_r(\sigma)$ denote the r^{th} elementary symmetric polynomial:*

$$\mathbf{S}_r(\sigma) \stackrel{\text{def}}{=} \sum_{S \in \binom{[n]}{r}} \prod_{i \in S} \sigma_i.$$

Likewise, for a given symmetric matrix $A \in \mathbb{S}^m$, $\mathbf{S}_r(A)$ is defined as

$$\mathbf{S}_r(A) = \sum_{U \in \binom{[m]}{r}} |A_{U,U}| \ ,$$

where $A_{U,U}$ is the minor of A corresponding to columns and rows in U.

Lemma 10.8. *If $A \in \mathbb{S}^{[m]}$ has eigenvalues $\{\sigma_i\}$, then $\mathbf{S}_r(A) = \mathbf{S}_r(\sigma)$.*

Proof. The coefficient of x^{m-r} in $\prod_i(\sigma_i - x)$ equals $(-1)^{m-r}\mathbf{S}_r(\sigma)$. Similarly $(-1)^{m-r}\mathbf{S}_r(A)$ is the coefficient of x^{m-r} in $|-xI + A|$. Now, note that $|-xI + A| = \prod_i(\sigma_i - x)$. $\square$

Given a matrix $X \in \mathbb{R}^{[m]\times[n]}$ and $i \in [n]$, we use X_i to denote i^{th} column of X. Similarly given a subset of columns, $C \subseteq [n]$, we use X_C to denote the matrix formed by columns from C, $X_C = (X_i)_{i\in C}$. Also we will let X^{Π} and $X^{\perp}$ be the projection matrix onto range and null space of X respectively.

For any symmetric matrix $A \in \mathbb{S}^{[m]}$, we will use $|A|$ to denote the determinant of A, $\mathrm{Tr}(A)$ to denote trace of A and $\sigma_i(A)$ to denote the i^{th} largest eigenvalue of A.

Lemma 10.9. *For any $A \in \mathbb{R}^{[m]\times[r]}$, if all r columns of A are linearly independent, then the distance of $x \in \mathbb{R}^{[m]}$ to span of A is given by $\|A^{\perp}x\|^2 = \dfrac{\begin{vmatrix} A^TA & A^Tx \\ x^TA & x^Tx \end{vmatrix}}{|A^TA|}$.*

Proof. Note that by elementary row operations,

$$\begin{vmatrix} A^TA & A^Tx \\ x^TA & x^Tx \end{vmatrix} = \begin{vmatrix} A^TA & \vdots \\ 0 & x^Tx - x^TA(A^TA)^{-1}A^Tx \end{vmatrix} = \left|A^TA\right| \left|x^TA^{\perp}x\right| = \left|A^TA\right| \|A^{\perp}x\|^2$$

where we used the fact that $A(A^TA)^{-1}A^T = A^{\Pi}$ and $I - A^{\Pi} = A^{\perp}$. □

10.5 Bound on Ratio of Symmetric Functions

The following theorem was first proved in the classic paper of Schur [1923]. [See also Marshall et al., 2009, Section 3]. We present a different proof below.

Theorem 10.10. *For any $\sigma \in \mathbb{R}_+^{[n]}$, the ratio $\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)}$ is Schur-concave.*

Proof. By Schur's criterion to establish Schur-concavity of symmetric functions, it suffices to show that

$$\underbrace{\left(\frac{\partial \mathbf{S}_{r+1}(\sigma)/\mathbf{S}_r(\sigma)}{\partial \sigma_i} - \frac{\partial \mathbf{S}_{r+1}(\sigma)/\mathbf{S}_r(\sigma)}{\partial \sigma_j}\right)}_{(*)} (\sigma_i - \sigma_j) \le 0$$

for all i, j. Using the identities

$$\begin{aligned} \frac{\partial \mathbf{S}_{r+1}(\sigma)/\mathbf{S}_r(\sigma)}{\partial \sigma_i} =& \frac{\mathbf{S}_r(\sigma)\mathbf{S}_r(\sigma \setminus \sigma_i) - \mathbf{S}_{r+1}(\sigma)\mathbf{S}_{r-1}(\sigma \setminus \sigma_i)}{\mathbf{S}_r^2(\sigma)} \\ \mathbf{S}_k(\sigma \setminus \sigma_i) =& \sigma_j \mathbf{S}_{k-1}(\sigma \setminus \{\sigma_i, \sigma_j\}) + \mathbf{S}_k(\sigma \setminus \{\sigma_i, \sigma_j\}) \end{aligned}$$

we have that

$$\begin{aligned}(*)\mathbf{S}_r^2(\sigma) =& \mathbf{S}_r(\sigma)\left[\mathbf{S}_r(\sigma \setminus \sigma_i) - \mathbf{S}_r(\sigma \setminus \sigma_j)\right] - \mathbf{S}_{r+1}(\sigma)\left[\mathbf{S}_{r-1}(\sigma \setminus \sigma_i) - \mathbf{S}_{r-1}(\sigma \setminus \sigma_j)\right] \\ =& \mathbf{S}_r(\sigma)\left(\sigma_j - \sigma_i\right)\mathbf{S}_{r-1}\left(\sigma \setminus \{\sigma_i, \sigma_j\}\right) - \mathbf{S}_{r+1}(\sigma)\left(\sigma_j - \sigma_i\right)\mathbf{S}_{r-2}\left(\sigma \setminus \{\sigma_i, \sigma_j\}\right) \\ =& \left(\sigma_j - \sigma_i\right)\left(\mathbf{S}_r(\sigma)\mathbf{S}_{r-1}\left(\sigma \setminus \{\sigma_i, \sigma_j\}\right) - \mathbf{S}_{r+1}(\sigma)\mathbf{S}_{r-2}\left(\sigma \setminus \{\sigma_i, \sigma_j\}\right)\right)\end{aligned}$$

Note that if we can show that the expression

$$\mathbf{S}_r(\sigma)\mathbf{S}_{r-1}\left(\sigma \setminus \{\sigma_i, \sigma_j\}\right) - \mathbf{S}_{r+1}(\sigma)\mathbf{S}_{r-2}\left(\sigma \setminus \{\sigma_i, \sigma_j\}\right)$$

is non-negative, we are done. For $r = 2$, $\mathbf{S}_{r-2} = 0$ hence we will consider the case when $r \geq 3$.

We will do so by exhibiting a flow f on a bipartite graph with left nodes labeled with $L = \binom{[n]}{r+1} \times \binom{[n]\setminus\{i,j\}}{r-2}$ and right nodes labeled with $R = \binom{[n]}{r} \times \binom{[n]\setminus\{i,j\}}{r-1}$ with the property that if there is a non-zero flow from $(S,T) \in L$ to $(S',T') \in R$ then $\prod_{i\in S}\sigma_i \prod_{j\in T}\sigma_j \leq \prod_{i\in S'}\sigma_i \prod_{j\in T'}\sigma_j$ and total flow leaving any node on left is 1 whereas total flow entering any node on right is at most 1.

Given $(S,T) \in \binom{[n]}{r+1} \times \binom{[n]\setminus\{i,j\}}{r-2}$, consider $U = S \setminus (T \cup \{i,j\}) \neq \emptyset$. For each $k \in U$, we set

$$f_{(S,T),(S\setminus\{k\},T\cup\{k\})} = \frac{1}{|U|}.$$

By construction, this satisfies the following:

1. $\sum_{(S',T')\in R} f_{(S,T),(S',T')} = 1$.
2. $f_{(S,T),(S',T')}\left(\prod_{i\in S}\sigma_i \prod_{j\in T}\sigma_j - \prod_{i\in S'}\sigma_i \prod_{j\in T'}\sigma_j\right) = 0$.

In order to prove that $\sum_{(S,T)\in L} f_{(S,T),(S',T')} \leq 1$, if $f_{(S,T),(S',T')} \neq 0$, then there exists k for some $k \in T' \setminus S'$ such that $T = T' \setminus \{k\}$, $S = S' \cup \{k\}$. Hence $|S' \setminus (T' \cup \{i,j\})| = |S \setminus (T \cup \{i,j\})| - 1$. Therefore

$$\begin{aligned}\sum_{(S,T)\in L} f_{(S,T),(S',T')} =& \sum_{k\in T'\setminus S'} \frac{1}{|S' \setminus (T' \cup \{i,j\})| + 1} \\ =& \frac{|T' \setminus S'|}{|S' \setminus (T' \cup \{i,j\})| + 1} \qquad (10.4)\end{aligned}$$

We have $|S'| = |T'| + 1 \geq 3$, $|S' \setminus (T' \cup \{i,j\})| + 1 \geq |S' \setminus T'| - 2 + 1$. Therefore eq. (10.4) can be upper bounded by:

$$\leq \frac{|T' \setminus S'|}{|S' \setminus T'| - 1} = 1 \qquad (10.5)$$

where eq. (10.5) follows from $|S'| = |T'| + 1 \implies |S' \setminus T'| = |T' \setminus S'| + 1$. □

We now use the Schur-concavity to prove our upper bound on $\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)}$.

Lemma 10.11. *For any non-negative vector $\rho \in \mathbb{R}_+^{[n]}$, positive integers k, r such that $r \geq k$:*

$$\frac{\mathbf{S}_{r+1}(\rho)}{\mathbf{S}_r(\rho)} \leq \frac{1}{r+1-k}\left(\sum_{i \geq k+1} \rho_{\downarrow i}\right)$$

Proof. Note that, for any β:

$$\frac{\mathbf{S}_{r+1}(\beta\rho)}{\mathbf{S}_r(\beta\rho)} = \frac{\beta^{r+1}}{\beta^r}\frac{\mathbf{S}_{r+1}(\rho)}{\mathbf{S}_r(\rho)} = \beta\frac{\mathbf{S}_{r+1}(\rho)}{\mathbf{S}_r(\rho)}.$$

Thus without loss of generality, we may assume that $\sum_i \rho_i = 1$. Further, we can assume that ρ is sorted in non-increasing order. Let $\alpha \stackrel{\text{def}}{=} \sum_{i \leq k} \rho_i$. Consider the following series ρ'.

$$\rho'_i = \begin{cases} \frac{1-\alpha}{n-k} & \text{if } i \geq k+1, \\ \frac{\alpha}{k} & \text{else.} \end{cases}$$

Since ρ is sorted in non-increasing order, it is easy to see that, for all i we have $\rho'_i \geq \rho'_{i+1}$. We have $(\rho'_1, \ldots, \rho'_k) = (\frac{\alpha}{k}, \ldots, \frac{\alpha}{k}) \prec (\rho_1, \ldots, \rho_k)$ and $(\rho'_{k+1}, \ldots, \rho'_n) = (\frac{1-\alpha}{n-k}, \ldots, \frac{1-\alpha}{n-k}) \prec (\rho_{k+1}, \ldots, \rho_n)$. Therefore $\rho' \prec \rho$ which implies:

$$\begin{aligned}
\frac{\mathbf{S}_{r+1}(\rho)}{\mathbf{S}_r(\rho)} \leq \frac{\mathbf{S}_{r+1}(\rho')}{\mathbf{S}_r(\rho')} &= \frac{\sum_{0 \leq \ell \leq k} \binom{k}{\ell}\binom{n-k}{r-\ell+1}\left(\frac{1-\alpha}{n-k}\right)^{r-\ell+1}\left(\frac{\alpha}{k}\right)^{\ell}}{\sum_{0 \leq \ell \leq k} \binom{k}{\ell}\binom{n-k}{r-\ell}\left(\frac{1-\alpha}{n-k}\right)^{r-\ell}\left(\frac{\alpha}{k}\right)^{\ell}} \\
&= \frac{1-\alpha}{n-k} \cdot \frac{\sum_{0 \leq \ell \leq k} \binom{k}{\ell}\frac{n-k-r+\ell}{r-\ell+1}\binom{n-k}{r-\ell}\left(\frac{1-\alpha}{n-k}\right)^{r-\ell}\left(\frac{\alpha}{k}\right)^{\ell}}{\sum_{0 \leq \ell \leq k} \binom{k}{\ell}\binom{n-k}{r-\ell}\left(\frac{1-\alpha}{n-k}\right)^{r-\ell}\left(\frac{\alpha}{k}\right)^{\ell}} \\
&\leq \frac{n-r}{n-k}\frac{1-\alpha}{r-k+1} \leq \frac{1}{r-k+1}(1-\alpha)\,.
\end{aligned}$$

□

10.6 Bounds on Column Reconstruction

We now present the upper bound relating the best r-column reconstruction of a matrix X to the error $\|X - X_{(k)}\|_F^2$ of the best rank-k approximation in the Frobenius norm.

Theorem 10.12. *For any $X \in \mathbb{R}^{[m]\times[n]}$ and positive integers $r \geq k \geq 1$,*

$$\min_{S \in \binom{[n]}{r}} \operatorname{Tr}(X^T X_S^{\perp} X) \leq \mathbb{E}_{\mathbf{C} \sim \mathcal{C}_r(X)}\left[\operatorname{Tr}(X^T X_{\mathbf{C}}^{\perp} X)\right] \leq \frac{r+1}{r+1-k}\|X - X_{(k)}\|^2\,.$$

where $\mathbf{C} \sim \mathcal{C}_r(X)$ *denotes sampling* C *with probability proportional to determinant of* $X_C^T X_C$, $\left|X_C^T X_C\right|$. *In other words, for any positive real* $\varepsilon > 0$,

$$\min_{S \in \binom{[n]}{k/\varepsilon + k - 1}} \operatorname{Tr}(X^T X_S^{\perp} X) \leq (1+\varepsilon)\|X - X_{(k)}\|^2.$$

Furthermore, for any $r = o(n)$, *this bound is tight up to lower order terms in the number of columns chosen: There exists a matrix* $\widetilde{X} \in \mathbb{R}^{[n] \times [n]}$ *such that*

$$(1 + \varepsilon - o(1))\,\|\widetilde{X} - \widetilde{X}_{(k)}\|^2 \leq \min_{S \in \binom{[n]}{k/\varepsilon}} \operatorname{Tr}(\widetilde{X}^T \widetilde{X}_S^{\perp} \widetilde{X}).$$

Proof. The first bound is obvious since the minimum is upper bounded by the average. For the second bound, note that

$$\begin{aligned}
\mathbb{E}_{\mathbf{C} \sim \mathcal{C}_r(X)}\left[\operatorname{Tr}(X^T X_{\mathbf{C}}^{\perp} X)\right] &= \frac{\sum_{S \in \binom{[n]}{r}} \left|X_S^T X_S\right| \operatorname{Tr}(X^T X_S^{\perp} X)}{\sum_{S \in \binom{[n]}{r}} |X_S^T X_S|} \\
&= \frac{\sum_{S \in \binom{[n]}{r}} \sum_u \left|X_S^T X_S\right| \|X_S^{\perp} X_u\|^2}{\sum_{S \in \binom{[n]}{r}} |X_S^T X_S|} \\
&= \frac{\sum_{S \in \binom{[n]}{r}} \sum_u \left|X_{S,u}^T X_{S,u}\right|}{\sum_{S \in \binom{[n]}{r}} |X_S^T X_S|} \quad \text{(using Lemma 10.9)} \\
&= \frac{(r+1)\sum_{T \in \binom{[n]}{r+1}} \left|X_T^T X_T\right|}{\sum_{S \in \binom{[n]}{r}} |X_S^T X_S|} \\
&= (r+1)\frac{\mathbf{S}_{r+1}(\sigma)}{\mathbf{S}_r(\sigma)} \quad \text{(using Lemma 10.8)}
\end{aligned}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ are the eigenvalues of $X^T X$. The claimed upper bound now follows by applying the bound from Lemma 10.11 and recalling $\|X - X_{(k)}\|_F^2 = \sum_{i \geq k+1} \sigma_i$.

Existence of $\widetilde{X}$ follows from Lemma 10.18 given in Section 10.9. □

10.7 Fast Volume Sampling Algorithm

In this section, we describe and analyze our volume sampling algorithm, which leads to the proof of Theorem 10.2.

Algorithm 10.1 $C =$ **Vol-Sample**(X, r): Algorithm for volume sampling subsets of columns.

Input: • $X \in \mathbb{R}^{[m]\times[n]}$ and positive integer r.

Output: • $C \in \binom{[n]}{r}$ chosen probability $\propto \left|X_C^T X_C\right|$, i.e. $C \sim \mathcal{C}_r(X)$.

Procedure: 1. Let $C \leftarrow \emptyset$. Initialize the table $\mathcal{T}$ of the n outer products $X_{[\ell,n]}X_{[\ell,n]}^T$, for all ℓ in $[n]$.

2. Choose τ uniformly at random from $[0,1]$, $\tau \in_u [0,1]$.

3. $t \leftarrow \tau \cdot \mathbf{S}_r(X^T X)$.

4. For $i \leftarrow 1$ to r:

   1. $\ell \leftarrow 1$, $u \leftarrow n$.
   2. While $\ell \neq u$
      (a) $m \leftarrow \lfloor \frac{\ell+u}{2} \rfloor$.
      (b) $h \leftarrow \mathbf{S}_r\left(X_{[\ell,n]}^T X_{[\ell,n]}\right) - \mathbf{S}_r\left(X_{[m+1,n]}^T X_{[m+1,n]}\right)$ which is equal to $\mathbf{S}_r\left(X_{[\ell,n]}{X_{[\ell,n]}}^T\right) - \mathbf{S}_r\left(X_{[m+1,n]}{X_{[m+1,n]}}^T\right)$ using $\mathcal{T}$.
      (c) If $t > h$, then $t \leftarrow t - h$ and $\ell \leftarrow m+1$.
      (d) Else $u \leftarrow m$.
   3. $C \leftarrow C \cup \{\ell\}$, $X \leftarrow X_\ell^\perp X$ and update the table $\mathcal{T}$ of outer products.

5. Return C.

Theorem 10.13. *Given a matrix $X \in \mathbb{R}^{m\times n}$, $m \leq n$, and an integer r, Algorithm* ***Vol-Sample****(X, r) returns $C \in \binom{[n]}{r}$ with probability $\frac{\left|X_C^T X_C\right|}{\sum_{T\in\binom{[n]}{r}}\left|X_T^T X_T\right|}$.*

Furthermore it can be implemented using at most $O\left(rm^2 n\right)$ arithmetic operations.

Proof of Correctness. For correctness, notice that for C sampled with probability $\left|X_C^T X_C\right|$, if we let $C = \{i_1 < i_2 < \ldots < i_r\}$:

$$\mathrm{Prob}_{i_1,\ldots,i_r}\left[i_1 = j\right] = \|X_j\|^2 \frac{\mathbf{S}_{r-1}(X_{[j+1,n]}^T X_j^\perp X_{[j+1,n]})}{\mathbf{S}_r(X^T X)}.$$

Notice that the algorithm, when it exists out of the while loop for the first time, chooses each ℓ with probability

$$\frac{\mathbf{S}_r(X_{[\ell,n]}^T X_{[\ell,n]}) - \mathbf{S}_r(X_{[\ell+1,n]}^T X_{[\ell+1,n]})}{\mathbf{S}_r(X^T X)} = \|X_\ell\|^2 \frac{\mathbf{S}_{r-1}(X_{[\ell+1,n]}^T X_\ell^\perp X_{[\ell+1,n]})}{\mathbf{S}_r(X^T X)}$$

which completes the proof. □

Proof of the Running Time. We assume each elementary arithmetic operation takes unit time.

Using the algorithm given in [Bürgisser et al., 2010, Section 16.6], we can compute $\mathbf{S}_r(X_{[\ell,n]}^T X_{[\ell,n]}) = \mathbf{S}_r(X_{[\ell,n]} X_{[\ell,n]}^T)$ in time $O(m^\omega \log m)$ *given* the outer product $X_{[\ell,n]} X_{[\ell,n]}^T$. Since $X_{A\cup B} X_{A\cup B}{}^T = X_A X_A^T + X_B X_B{}^T$, we can compute the table $\mathcal{T}$ all the n outer products $X_{[\ell,n]} X_{[\ell,n]}^T$, for $\ell \in [n]$, in time $O(m^2 n)$. Also, given X_ℓ, if we let $z = \frac{X_\ell}{\|X_\ell\|}$:

$$(X_\ell^\perp X_S)(X_\ell^\perp X_S)^T = X_S X_S{}^T + zz^T(z^T X_S X_S^T z) - zz^T X_S X_S^T - X_S X_S^T zz^T.$$

Hence, after choosing some column ℓ, we can update each outer product matrix in the table $\mathcal{T}$ in $O(m^2)$ time. Since there are at most n matrices in this table, each update step takes $O(m^2 n)$ time.

For each column we choose, we evaluate at most $O(\log n)$ many symmetric functions $\mathbf{S}_r$. Thus choosing one column takes time $O(m^\omega \log m \log n)$ given the table $\mathcal{T}$. Since we choose r columns, the total amount of time, including the time to initialize and update $\mathcal{T}$ in each iteration, is bounded by

$$O\left(rm^\omega \log m \log n + rm^2 n\right) = O\left(rm^2(m^{\omega-2} \log m \log n + n)\right).$$

Since $m^{\omega-2} \log m \log n \le \sqrt{n} \log^2 n = o(n)$, this bound becomes $O(rm^2 n)$. □

The claim in Theorem 10.2 about the performance of Algorithm **Vol-Sample** as a column-selection algorithm follows from the upper bound on $\mathbb{E}_{\mathbf{C}\sim\mathcal{C}_r(X)}\left[\operatorname{Tr}(X^T X_{\mathbf{C}}^\perp X)\right]$ in Theorem 10.12.

10.8 Deterministic Column Selection Algorithm

Recall that for simplicity we restricted the seed selection procedure of our faster solver framework Chapter 5 to be deterministic. Combined with the fact that the

exact running time is not important as long as it is polynomial in both matrix dimensions and number of columns to choose, we describe a simpler deterministic column selection algorithm. It is quite similar to algorithm of Deshpande and Rademacher [2010], and is based on the method of conditional expectations.

However there is another more sinister issue for integrating column selection into our rounding algorithms: In finite precision, it is not possible to exactly compute the Cholesky decomposition from a given Gram matrix. Although it is possible to get around of this problem by taking into account the rounding errors, there is a much simpler way: To run the column selection procedure directly on the Gram matrix itself.

The final deterministic algorithm for Gram matrices is given in Algorithm 10.2.

Algorithm 10.2 $S =$ **Find-Cols-Det**$(X^T X, r)$: Deterministic column selection for Gram matrices.

Input: • Gram matrix $X^T X \in \mathbb{S}_+^{[n]}$ for some $X \in \mathbb{R}^{[m],[n]}$, positive integer r.

Output: • $S \in \binom{[n]}{r}$ satisfying Theorem 10.12.

Procedure: 1. For $i \leftarrow 1$ to n:

(a) Compute $Y(i) = [Y(i)_{a,b}]_{a,b\in[n]}$ as the following symmetric matrix in exact arithmetic:

$$Y(i)_{a,b} \leftarrow \langle X_a, X_b\rangle - \frac{\langle X_a, X_i\rangle\langle X_b, X_i\rangle}{\|X_i\|^2}.$$

(b) Let $\tau_i \leftarrow \frac{\mathbf{S}_{r-1}(Y(i))}{\mathbf{S}_{r-2}(Y(i))}$.

2. Let $i^* \leftarrow \operatorname{argmin}_i \tau_i$.

3. Return $\{i^*\} \cup$ **Find-Cols-Det**$(Y(i^*), r-1)$.

We want to remark that Algorithm 10.2 is not the most efficient implementation, as the one given by Deshpande and Rademacher [2010] is faster: But for our rounding algorithms, this is sufficient. Moreover it can be implemented exactly as it does not rely on Cholesky decomposition.

Theorem 10.14. *Algorithm 10.2 runs in time* $O\left(rn^{3+\omega}\log n\right)$ *and it returns* $S \in \binom{[n]}{r}$ *satisfying* $\operatorname{Tr}(X^T X_S^{\perp} X) \leq \frac{r+1}{r+1-k}\|X - X_{(k)}\|^2$ *for any* $k \leq r$.

Proof. Running time bound is trivial. Observe that the matrix $Y(i)$ satisfies $Y(i) = X^T X_i^{\perp} X$ for any i. The rest of correctness proof follows from arguments of Deshpande and Rademacher [2010]. □

10.9 Lower Bound on Number of Columns Needed

In this section, we construct matrices for given k and r for which the upper bound stated in Theorem 10.12 is nearly tight. Our construction is in fact the same as the one given by Deshpande and Vempala [2006]. Our analysis is different and shows a lower bound on the quantity $\mathrm{Tr}(X^T X_S^{\perp} X)$ where the full column span of the chosen r columns is allowed for approximating X.

Definition 10.15. *Given $\delta > 0$ and m, we define $M^{(m,\delta)} \in \mathbb{R}^{[m]\times[m]}$ as*

$$M^{(m,\delta)} \stackrel{\text{def}}{=} \delta I + J,$$

where I is the identity matrix of dimension m, and J the all 1's $m \times m$ matrix.

Observation 10.16. *Given any $\delta > 0$ and positive integer m, the followings hold for the matrix $M^{(m,\delta)}$:*

1. $\mathrm{Tr}(M^{(m,\delta)}) = m(1+\delta)$.
2. *Its largest eigenvector is the all 1's vector, with corresponding eigenvalue $\sigma_1 = \sigma_1\left(M^{(m,\delta)}\right)$ given by $\sigma_1 = \delta + m$. Rest of the eigenvalues are all equal with value $\sigma_2 = \sigma_3 = \ldots = \sigma_m = \delta$.*
3. $\left|M^{(m,\delta)}\right| = \prod_{i=1}^{m} \sigma_i = \delta^m + m\delta^{m-1}$.

Lemma 10.17. *Given any $\delta > 0$ and positive integer r, for $n \geq r$, if we let $X^T X = M^{(n,\delta)}$, then*

$$\min_{S \in \binom{[n]}{r}} \frac{\mathrm{Tr}\left(X^T X_S{}^{\perp} X\right)}{\|X - X_{(1)}\|_F^2} \geq 1 + \frac{k}{r} - o(1).$$

Proof. Note that $\|X - X_{(1)}\|_F^2 = \sum_{i\geq 2} \sigma_i = (n-1)\delta$. For any subset $C \subseteq [n]$ of size $|C| = r$, the corresponding minor of $X^T X$ is given by

$$X_C^T X_C = M^{(|C|,\delta)} \implies \left|X_C^T X_C\right| = \delta^r + r\delta^{r-1}.$$

Consequently for $i \notin C$,

$$\left\|X_C^{\perp} X_i\right\|^2 = \frac{\left|X_{C\cup\{i\}}^T X_{C\cup\{i\}}\right|}{|X_C^T X_C|} = \frac{\delta^r\left(\delta+(r+1)\right)}{\delta^{r-1}\left(\delta+r\right)} = \delta\left(1+\frac{1}{r+\delta}\right).$$

In particular,

$$\mathrm{Tr}(X^T X_C^{\perp} X) = (n-r)\delta\left(1+\frac{1}{r+\delta}\right).$$

Therefore

$$\frac{\mathrm{Tr}\left(X^T {X_S}^{\perp} X\right)}{\|X - X_{(1)}\|_F^2} = \frac{n-r}{n-1}\left(1+\frac{1}{r+\delta}\right). \qquad \square$$

Lemma 10.18. *For any positive integer n and positive integers k and r, $r \geq k$, such that $r = o(n)$, there exists an n-by-n matrix $X \in \mathbb{R}^{[n]\times[n]}$ for which the following holds:*

$$\min_{S\in\binom{[n]}{r}} \frac{\mathrm{Tr}(X^T X_{\overline{S}}^{\perp} X)}{\left\|X - X_{(k)}\right\|^2} \geq \frac{n-r}{n-k}\left(1+\frac{k}{r}-o(1)\right).$$

Proof. We will fix δ to be an infinitesimally small number, $\delta = o(1)$.

For $n = n_0 \cdot k$ with $n_0 \geq r+1$, let X be chosen so that $X^T X$ is *block diagonal* matrix of size $n \times n = n_0 k \times n_0 k$ with k copies of $M^{(n_0,\delta)}$ on its diagonals:

$$X^T X = \begin{pmatrix} M^{(n_0,\delta)} & 0^{(n_0)} & \cdots & 0^{(n_0)} \\ 0^{(n_0)} & M^{(n_0,\delta)} & & \vdots \\ \vdots & & \ddots & \\ 0^{(n_0)} & \cdots & & M^{(n_0,\delta)} \end{pmatrix} = I^{(k)} \otimes M^{(n_0,\delta)}$$

where we used $0^{(m)}$ and $I^{(m)}$ to denote matrices of size $m \times m$ consisting of all zeroes and identity respectively. Here $\otimes$ denotes tensor (Kronecker) product. By property of tensoring [see Horn and Johnson, 1991], $X^T X$ has k copies of each eigenvalue of $M^{(n_0,\delta)}$. In particular,

$$\left\|X - X_{(k)}\right\|^2 = n(1+\delta) - n - k\delta = (n-k)\delta. \tag{10.6}$$

We will use $[k] \times [n_0]$ to index the columns of matrix X, so that for any $i \in [k]$, if we let $X^{(i)} \stackrel{\text{def}}{=} X_{\{i\}\times[n_0]}$, we have ${X^{(i)}}^T X^{(i)} = M^{(n_0,\delta)}$, and for any $i \neq j \in [k]$, ${X^{(i)}}^T X^{(j)} = 0^{(n_0)}$.

Proceeding as in [Deshpande and Vempala, 2006], given S, let S_i be the set of columns chosen from i^{th} block, so that $S_i \stackrel{\text{def}}{=} \{j \in [n_0] \mid (i,j) \in S\}$. It is easy to see that,

$$\operatorname{Tr}\left(X^{(i)^T} X_S^{\perp} X^{(i)}\right) = \operatorname{Tr}\left(X^{(i)^T} X_{S_i}^{(i)^{\perp}} X^{(i)}\right) \geq \delta(n_0 - |S_i|)\left(1 + \frac{1}{\delta + |S_i|}\right).$$

where we used Lemma 10.17. Therefore

$$\operatorname{Tr}\left(X^T {X_S}^{\perp} X\right) = \sum_i \operatorname{Tr}\left(X^{(i)^T} X_{S_i}^{(i)^{\perp}} X^{(i)}\right) = \sum_i \delta(n_0 - |S_i|)\left(1 + \frac{1}{\delta + |S_i|}\right). \tag{10.7}$$

Note that $(n-x)(1+1/(\delta+x))$ is convex as long as $x + \delta \geq 0$. Therefore we can use Jensen's inequality and lower bound the expression in eq. (10.7) by

$$\begin{aligned}\delta k\left(n_0 - \frac{1}{k}\sum_i |S_i|\right)\left(1 + \frac{1}{\delta + \frac{1}{k}\sum_i |S_i|}\right) &= \delta k\left(n_0 - \frac{r}{k}\right)\left(1 + \frac{1}{\delta + \frac{r}{k}}\right) \\ &= \delta\,(n-r)\left(1 + \frac{1}{\delta + \frac{r}{k}}\right)\end{aligned}$$

Recalling the bound (Equation (10.6)) for the best rank-k approximation, we see that for any S with $|S| = r = o(n)$ and $\delta = o(1)$:

$$\frac{\operatorname{Tr}\left(X^T {X_S}^{\perp} X\right)}{\left\|X - X_{(k)}\right\|^2} \geq \frac{n-r}{n-k}\left(1 + \frac{k}{r}(1 - o(1))\right) \geq 1 + \frac{k}{r} - o(1). \qquad \square$$

Chapter 11

Existence of Primal and Dual Optimal Solutions

In this chapter, we analyze our relaxations from a dual perspective and we show the existence of primal and dual optimal solutions.

11.1 Preliminaries

First we review the basic notions of open and closed sets for Euclidean spaces from geometric topology.

Definition 11.1 (Closed and Open Sets)**.** *Given $X \subseteq \mathbb{R}^A$, X is a **closed set** if, for any $y \notin X$, there exists a ball of radius $\varepsilon > 0$ around y, $\mathbb{B}_d(y, \varepsilon)$, disjoint from X: $\mathbb{B}_d(y, \varepsilon) \cap X = \emptyset$. Similarly X is a **open set** if its complement, $\mathbb{R}^A \setminus X$, is closed.*

11.1.1 Linear Conic Programming

Recall the definition of (convex) cones from Section 2.5 and consider the problem of linear optimization over such sets.

Definition 11.2 (Linear Conic Programming)**.** *Given:*

- *Two linear spaces E_1, E_2 with a bi-linear form $\langle\langle \cdot, \cdot \rangle\rangle : E_1 \times E_2 \to \mathbb{R}$;*
- *A linear transform $T : E_1 \to E_1$ with adjoint $\widehat{T} : E_2 \to E_2$ such that:*

$$\text{for any } p \in E_1, q \in E_2 \ \langle\langle T(p), q \rangle\rangle = \langle\langle p, \widehat{T}(q) \rangle\rangle;$$

- *A convex cone $K \subseteq E_1$ with dual $K^* \subseteq E_2$ such that:*

$$K^* \stackrel{\text{def}}{=} \left\{ q \in E_2 \;\middle|\; \langle\!\langle p, q \rangle\!\rangle \geq 0 \text{ for all } p \in K \right\};$$

- *Two points $b \in E_1, c \in E_2$;*

eq. (11.1) *is a linear conic programming (LCP) instance with dual eq.* (11.2)*:*

$$\inf \langle\!\langle x, c \rangle\!\rangle \text{ st } T(x) = b,\ x \geq_K 0,\ x \in E_1. \qquad \text{(PRIMAL)} \qquad (11.1)$$

$$\sup \langle\!\langle b, y \rangle\!\rangle \text{ st } \widehat{T}(y) \leq_{K^*} c,\ y \in E_2. \qquad \text{(DUAL)} \qquad (11.2)$$

*Given $x \geq_K 0$ (resp. y), we say x is **primal feasible** if $T(x) = b$ (resp. **dual feasible** if $\widehat{T}(y) \leq_{K^*} c$). We will denote the optimum value of eqs.* (11.1) *and* (11.2) *with OPT_P and OPT_D respectively. We say x (resp. y) is a primal (resp. dual) **optimal solution** if it is feasible and $\langle\!\langle x, c \rangle\!\rangle = \mathsf{OPT}_p$ (resp. $\langle\!\langle b, y \rangle\!\rangle = \mathsf{OPT}_d$).*

The associated primal and dual cones are defined as:

$$\left\{ (T(x), \langle\!\langle x, c \rangle\!\rangle) \middle| x \in K \right\}, \qquad \text{(PRIMAL CONE)} \qquad (11.3)$$

$$\left\{ (\widehat{T}(y), \langle\!\langle b, y \rangle\!\rangle) \middle| y \in K^* \right\}. \qquad \text{(DUAL CONE)} \qquad (11.4)$$

Lemma 11.3. *Given a linear conic programming instance as in Definition 11.2:*

1. **(Weak Duality)** *For any pair of feasible primal and dual solutions (x, y), $\langle\!\langle x, c \rangle\!\rangle \geq \langle\!\langle b, y \rangle\!\rangle$.*
2. **(Optimality Condition)** *Provided that $\mathsf{OPT}_p = \mathsf{OPT}_d$ x, y are optimal primal and dual solutions iff $\langle\!\langle x, c - \widehat{T}(y) \rangle\!\rangle = 0$.*

Proof. [see Barvinok, 2002, Borwein and Lewis, 2000] □

Theorem 11.4 (Strong Duality)**.** *Given a linear conic programming instance as in Definition 11.2 if (1) There exists a primal feasible solution, (2) Primal is bounded, (3) Primal cone is closed; Then $\mathsf{OPT}_P = \mathsf{OPT}_D$ and there exists optimal primal solution.*

Proof. [see Borwein and Lewis, 2000, Barvinok, 2002]. □

Corollary 11.5 (Cone Programming Duality)**.** *Given a linear conic programming instance as in Definition 11.2 if*

(1) There exist primal and dual feasible solutions,

(2) Primal is bounded,

(3) Primal and dual cones are closed;

then $\mathsf{OPT}_P = \mathsf{OPT}_D$ *and there exists optimal primal and dual solutions satisfying optimality condition.*

Proof. $\mathsf{OPT}_P = \mathsf{OPT}_D$ and existence of primal solution follows from Theorem 11.4. By weak duality from Lemma 11.3, dual is bounded. Therefore we can use Theorem 11.4 on dual to infer the existence of dual optimal solution. Our proof is complete by using optimality condition from Lemma 11.3. □

Corollary 11.5 allows us to characterize when a primal or dual optimal solution exists in terms of closedness of primal and dual cones as in Definition 11.2.

11.1.2 Closed Convex Cones

First we list some standard examples for closed cones.

Lemma 11.6. *For any* A, $\mathbb{R}^A$, $\mathbb{R}^A_+$, $\mathbb{S}^A$, $\mathbb{S}^A_+$ *are closed, convex cones.*

Claim 11.7.

1. K^* *is a closed, convex cone.*
2. $(K^*)^* \supseteq K$.
3. *If* $K, L \subseteq \mathbb{R}^A$ *are two convex cones then*

$$(K \cap L)^* = \text{convex}(K^* + L^*).$$

4. *If* $K, L \subseteq \mathbb{R}^A$ *are two closed convex with* $K \perp L$, $K + L$ *is a closed convex cone.*
5. *If* $K \subseteq \mathbb{R}^A$, $L \subseteq \mathbb{R}^B$ *are two closed convex cones with* $A \cap B = \emptyset$, *then* $K \oplus L$ *is a closed convex cone.*

Proof of Item 1. Note $H_+(x) \stackrel{\text{def}}{=} \left\{h \middle| \langle\langle x, h\rangle\rangle \geq 0\right\}$ is a closed, convex set for any x. Since

$$K^* = \bigcap_{x \in K} H_+(x),$$

K^* is also closed and convex. For any $y \in K^*$ and $t \in \mathbb{R}_+$, $\langle\langle y, x\rangle\rangle \geq 0 \implies \langle\langle ty, x\rangle\rangle \geq 0$ hence $y \in K^* \implies ty \in K^*$ so K^* is a cone. □

Proof of Item 2. For any $h \in K^*$,

$$\forall x \in K : \langle\langle h, x \rangle\rangle \geq 0 \implies K \subseteq H_+(h).$$

Consequently

$$K^{**} = \bigcap_{h \in K^*} H_+(h) \supseteq K.$$

□

Proof of Item 3. **(⊇)** Given $h \in \text{convex}(K^* + L^*)$ of the form $h = \alpha h' + \beta h''$ with $h' \in K^*, h'' \in L^*$ and $\alpha, \beta \geq 0$; observe that

$$\langle\langle h', x \rangle\rangle \geq 0 \text{ for all } x \in K \supseteq K \cap L,$$

hence $h' \in (K \cap L)^*$ (similarly $h'' \in (K \cap L)^*$ as well).

(⊆) By contradiction. Given $h \in (K \cap L)^*$, suppose $h \notin \text{convex}(K^* + L^*)$. Then there exists a hyperplane separating h and $\text{convex}(K^* + L^*)$, y, such that

$$\langle\langle h, y \rangle\rangle < 0 \text{ and } \langle\langle \text{convex}(K^* + L^*), y \rangle\rangle \geq 0 \implies \langle\langle K^* \cup L^*, y \rangle\rangle \geq 0.$$

Hence $y \in (K^*)^* \subseteq K$ as well as $y \in (L^*)^* \subseteq L$ so $y \in K \cap L$. But then $\langle\langle h, y \rangle\rangle < 0$ implies $h \notin (K \cap L)^*$ which is a contradiction. □

Proofs of Items 4 and 5. $K + L$ and $K \oplus L$ are convex cones by construction. Closedness directly follows from properties of direct sum topology. □

Theorem 11.8. *If K is a closed convex cone, then*

$$K^{**} = K.$$

Proof. Observe that K^{**} is closed by Claim 11.7 and contains K. Consider

$$\inf_{y \in K^{**}} \sup_{x \in K : \|x\| \leq 1} \|x - y\|_2.$$

Since both $K \cap \mathbb{B}(0, 1)$ and K^{**} are closed, convex sets, there exists an optimal value $\delta \geq 0$ and corresponding optimal solutions x' and y' such that:

$$\max_{x \in K^{**} : \|x\| \leq 1} \min_{y \in K : \|y\| \leq 1} \|x - y\|_2 = \|x' - y'\|_2 = \delta.$$

Moreover $\delta = 0$ iff $K = K^{**}$. Now suppose $\delta > 0$. Consider the hyperplane h that goes through y' and origin in the direction from x' to y'. Then $\langle\langle h, K \rangle\rangle \geq 0$ for otherwise y' will not be optimal so $h \in K^*$. Moreover $\langle\langle h, x \rangle\rangle < 0$. But $x \in K^{**}$ which means $\langle\langle x, h \rangle\rangle \geq 0$ a contradiction. □

Our primal and dual cones are defined as linear transformations of some other cones whose closedness usually follows from Lemma 11.6 and Claim 11.7. Then if we can characterize what kind of transformations preserve closedness we can easily show that associated primal and dual cones are closed as well.

Proposition 11.9 ([See Barvinok, 2002, Borwein and Lewis, 2000]). *Given linear subspaces E_1, E_2, a closed convex cone $K \subseteq E_1$ and a linear mapping $T : E_1 \mapsto E_2$, if*

$$\ker(T) \cap K = \{0\}$$

then $T(K)$ is closed.

Lemma 11.10 ([See Borwein and Moors, 2009]). *Given linear subspaces E_1, E_2, a closed convex cone $K \subseteq E_1$ and a linear mapping $T : E_1 \mapsto E_2$, if*

$$\ker(T) \cap K$$

is a linear subspace then $T(K)$ is closed.

Proof. Let U be the linear space corresponding to $\ker(T) \cap K$. Then we can express K as the direct sum of two orthogonal sets, $K = U^\perp K \oplus U$. Furthermore both $U^\perp K$ and U are convex and closed (by closedness of K) cones. Observe that $\ker(T) \cap U^\perp K = \{0\}$ hence by Proposition 11.9, $T(U^\perp K)$ is a closed convex cone. Finally $T(K) = T(U^\perp K) \oplus T(U) = T(U^\perp K)$ therefore $T(K)$ is closed also. □

As an immediate application, we can prove that the convex cone of SoS polynomials, which we introduced back in Definition 3.32, is also closed:

Theorem 11.11. *(i) $\Sigma_{\mathcal{F}}$ is closed.*

(ii) $\Sigma_{\mathcal{F}}^*$ is the closed dual cone of $\Sigma_{\mathcal{F}}$.

Proof. (i) By first property from Lemma 3.33, we have $\widehat{\mathbb{S}_+^{\mathcal{F}}} = \Sigma_{\mathcal{F}}$. We know $\mathbb{S}_+^{\mathcal{F}}$ is closed by Lemma 11.6. Since $\widehat{\cdot}$ is a linear map, let $\ker = \{G \in \mathbb{R}^{\mathcal{F},\mathcal{F}} \mid \widehat{G} = 0\}$ be its kernel and consider $G \in \mathbb{S}_+^{\mathcal{F}} \cap \ker$. Then $\widehat{G} = \sum_i \mathsf{g}_i^2$ but $\sum_i \mathsf{g}_i^2 \equiv 0 \pmod{\mathcal{B}_V}$ which means $\mathsf{g}_i \equiv 0$ for all i. But $\mathsf{g}_i \in \mathbb{ML}_{\mathcal{F}}[\mathbf{X}]$ thus $\mathsf{g}_i = 0 \implies G = 0$. By Proposition 11.9 our proof is complete.

(ii) Duality follows from Theorem 3.34. Since it is dual cone, it is closed by Claim 11.7. □

Corollary 11.12. $\mathsf{f} \in \Sigma_{\mathcal{F}} \iff \langle\!\langle \mathsf{f}, x \rangle\!\rangle \geq 0$ *for all* $x \in \Sigma_{\mathcal{F}}^*$.

Proof. Since $\Sigma_{\mathcal{F}}$ is a closed and convex cone, we see that dual of its dual cone is equal to itself, $(\Sigma_{\mathcal{F}})^{**} = \Sigma_{\mathcal{F}}$, by Theorem 11.8. □

11.2 Existence of Primal and Dual Optimal Solutions for Select Problems

In this section, we present the duals for some of our relaxations and prove existence of primal and dual optimal solutions. As opposed to the overall theme of our thesis, we will work on a problem-by-problem basis in this section.

As mentioned in the beginning of this chapter, our focus will be on Lasserre relaxations for binary partitioning problems but it is trivial to translate all our results to k-labeling problems as well. For such relaxations, we can define our linear subspaces E_1, E_2 as:

$$E_1 = \mathbb{R}^{\mathcal{F}\oplus\mathcal{F}},\ E_2 = \{\mathsf{f} \in \mathbb{R}[\mathbf{X}] : [\mathsf{f}] \in \mathbb{ML}_{\mathcal{F}\uplus\mathcal{F}}[\mathbf{X}]\}\,.$$

We defined an inner product between these two spaces, $\langle\!\langle \cdot, \cdot \rangle\!\rangle$, in terms of the pseudo-evaluation operator so that given $x \in E_1$ and $\mathsf{f} \in E_2$:

$$\langle\!\langle \mathsf{f}, x \rangle\!\rangle = \sum_{S\in\mathcal{F}\uplus\mathcal{F}} x_S[\mathsf{f}]_S.$$

11.2.1 Minimum Bisection

Recall the integer programming formulation for minimum bisection problem:

$$\begin{array}{ll} \min & \sum_{u<v} w^G_{u,v}(\mathbf{x}_u - \mathbf{x}_v)^2 \\ \text{st} & \sum_u \mathbf{x}_u = \mu, \\ & \mathbf{x} \in \{0,1\}^V. \end{array}$$

In Table 11.1, we give primal and dual formulations for Minimum Bisection problem.

Theorem 11.13. *Given graph $G = (V, E, W)$, down family $\mathcal{F} \supseteq V_{\leq 1}$, positive integer μ, the moment relaxation for* Minimum Bisection *problem on $\mathcal{F}$ along with its dual as given in Table 11.1 both have optimal solutions and any pair of such optimal solutions always satisfy optimality condition.*

Proof. Note the usual relaxation for Minimum Bisection in terms of moment sequences:

$$\begin{array}{ll} \min & \langle\!\langle \widehat{L_G}, x \rangle\!\rangle \\ \text{st} & \langle\!\langle (\sum_u \mathbf{X}_u - \mu)^2, x \rangle\!\rangle = 0, \\ & x_\emptyset = 1, \quad x \succeq_{\Sigma^*_{\mathcal{F}}} 0, \quad x \in \mathbb{R}^{\mathcal{F}\uplus\mathcal{F}}. \end{array}$$

Minimum Bisection	
Primal	Dual
$\begin{array}{ll}\min & \sum_{u<v} w^G_{u,v}\|\vec{x}_u - \vec{x}_v\|^2 \\ \text{st} & \sum_u \vec{x}_u = \mu\vec{x}_\emptyset, \\ & \|\vec{x}_\emptyset\|^2 = 1, \\ & \langle\!\langle \vec{x}_S, \vec{x}_T \rangle\!\rangle = x_{S\cup T} \text{ for all } S,T \in \mathcal{F}.\end{array}$	$\begin{array}{ll}\max & \eta \\ \text{st} & y\left(\sum_u \mathbf{X}_u - \mu\right)^2 + \eta \leq_{\Sigma_\mathcal{F}} \widehat{L_G}, \\ & \eta \in \mathbb{R}, y \in \mathbb{R}.\end{array}$
Optimality Condition	
$\sum_{u<v} w^G_{u,v} \|\vec{x}_u - \vec{x}_v\|^2 = \eta.$	

Table 11.1: Primal and dual SDP formulations corresponding to moment relaxations of Minimum Bisection on down family $\mathcal{F}$ over V. Theorem 11.13 proves existence of optimal primal and dual solutions satisfying optimality condition. Note that here, for any positive integer r, $\mathcal{F} = \binom{V}{\leq r}$ corresponds to r-rounds of Lasserre relaxation.

We can see that the dual of this formulation indeed corresponds to the dual from Table 11.1 using Definition 11.2 and Corollary 11.12. In order to proceed, we verify the conditions from Corollary 11.5 one by one:

(1) Let $\mathbf{x} \in \{0,1\}^V$ be such that $\sum_u \mathbf{x}_u = \mu$. Then the vectors $[\vec{x}_S \leftarrow \mathbf{X}^S(\mathbf{x})]_S$ form a primal feasible solution. Similarly $y \leftarrow 0$ and $\eta \leftarrow 0$ forms a dual feasible solution since $L_G \succeq 0$.

(2) $L_G \succeq 0$, thus primal is bounded from below by 0.

(3) $\Sigma^*_\mathcal{F}$ is a closed convex cone. Primal cone is given by:

$$\left\{(\langle\!\langle \mathsf{p}, x\rangle\!\rangle, \langle\!\langle \mathsf{q}, x\rangle\!\rangle, x_\emptyset) \,\middle|\, x \in \Sigma^*_\mathcal{F}\right\},$$

where $\mathsf{p} \stackrel{\text{def}}{=} \widehat{L_G}$ and $\mathsf{q} \stackrel{\text{def}}{=} \left(\sum_u \mathbf{X}_u - \mu\right)^2$. For any $x \in \Sigma^*_\mathcal{F}$ in its kernel, we have $x_\emptyset = 0$. By Claim 3.22, this implies $x = 0$. Using Proposition 11.9 we see that the primal cone is *closed.*

Dual cone is

$$\left\{(\eta, y\mathsf{q}+\eta)\,\middle|\, y\mathsf{q}+\eta \succeq_{\Sigma_{\mathcal{F}}} 0,\ y\in\mathbb{R}, \eta\in\mathbb{R}\right\}.$$

It is easy to see that the cone $\left\{(y,\eta)\middle| y\mathsf{q}+\eta \succeq_{\Sigma_{\mathcal{F}}} 0,\ y\in\mathbb{R},\eta\in\mathbb{R}\right\}$ is closed. For any (η, y) in the kernel, $\eta = 0$ and $y\mathsf{q} \equiv 0 \iff y = 0$. Using items 4 and 5 from Claim 11.7 we see that the dual cone is closed. □

11.2.2 Sparsest Cut

Now we consider the problem of sparsest cut:

$$\begin{array}{ll} \min & \dfrac{\sum_{u<v} w^G_{u,v}(\mathbf{x}_u-\mathbf{x}_v)^2}{\sum_{u<v} w^H_{u,v}(\mathbf{x}_u-\mathbf{x}_v)^2} \\ \text{st} & \sum_{u<v} w^H_{u,v}(\mathbf{x}_u-\mathbf{x}_v)^2 \le 2\underbrace{\sum_{u,v} w^H_{u,v}}_{\stackrel{\text{def}}{=} m^H}, \\ & \mathbf{x}\in\{0,1\}^V. \end{array}$$

This formulation might seem odd: The first inequality constraint is always satisfied as $(\mathbf{x}_u-\mathbf{x}_v)^2 \le 1$ always for any $u, v \in V$ rendering it redundant. However we chose to explicitly state this constraint as it will ensure that the primal cone in our relaxation will be closed.

For this problem, due to the objective function being non-linear, we were not able to use the standard relaxation. But we can normalize the denominator instead of $x_\emptyset$ to express this as an SDP formulation as in eq. (9.4) (where the equivalence was also proven.) In Table 11.2, we give this formulation, along with its dual and state the optimality condition:

Theorem 11.14. *Given graphs G and H on node set V, down family $\mathcal{F} \supseteq V_{\le 1}$, positive integer μ, the moment relaxation for* Non-Uniform Sparsest Cut *problem on $\mathcal{F}$ along with its dual as given in Table 11.2 both have optimal solutions and any pair of such optimal solutions always satisfy optimality condition.*

Proof. Let $\varepsilon \stackrel{\text{def}}{=} \frac{1}{2m^H}$. Our relaxation in terms of moment sequences is:

$$\begin{array}{ll} \min & \langle\!\langle \widehat{L_G}, y\rangle\!\rangle \\ \text{st} & \langle\!\langle \widehat{L_H}, y\rangle\!\rangle = 1, \\ & y_\emptyset \ge \varepsilon, \quad y \succeq_{\Sigma^*_{\mathcal{F}}} 0, \\ & y\in\mathbb{R}^{\binom{V}{\le 2r}}. \end{array} \quad \begin{array}{c} \xrightarrow{y:=\frac{x}{\langle\!\langle \widehat{L_H},x\rangle\!\rangle}} \\ \xleftarrow[x:=\frac{y}{y_\emptyset}]{} \end{array} \quad \begin{array}{ll} \min & \frac{\langle\!\langle \widehat{L_G},x\rangle\!\rangle}{\langle\!\langle \widehat{L_H},x\rangle\!\rangle} \\ \text{st} & x_\emptyset = 1, \quad \langle\!\langle \widehat{L_H}, x\rangle\!\rangle \le \frac{1}{\varepsilon}, \\ & x\in\mathbb{R}^{\binom{V}{\le 2r}}. \end{array}$$

Non-Uniform Sparsest Cut	
Primal	Dual
$\min \quad \frac{\sum_{u<v} w^G_{u,v}\|\vec{x}_u-\vec{x}_v\|^2}{\sum_{u<v} w^H_{u,v}\|\vec{x}_u-\vec{x}_v\|^2}$ $\text{st} \quad \sum_{u<v} w^H_{u,v}\|\vec{x}_u-\vec{x}_v\|^2 \le 2m^H,$ $\langle\!\langle \vec{x}_S, \vec{x}_T \rangle\!\rangle = x_{S\cup T}$ for all $S,T \in \mathcal{F}$.	$\max \quad \Phi + 2m^H\gamma$ $\text{st} \quad \Phi \cdot \widehat{L_H} + \gamma \le_{\mathcal{F}} \widehat{L_G},$ $\Phi \in \mathbb{R}, \gamma \in \mathbb{R}_+.$
Optimality Condition	
$\gamma = 0$ $\sum_{u<v} w^G_{u,v}\|\vec{x}_u-\vec{x}_v\|^2 = \Phi \sum_{u<v} w^H_{u,v}\|\vec{x}_u-\vec{x}_v\|^2.$	

Table 11.2: Primal and dual SDP formulations corresponding to moment relaxations of Non-Uniform Sparsest Cut on down family $\mathcal{F}$ over V. Theorem 11.14 proves existence of optimal primal and dual solutions satisfying optimality condition. Note that here, for any positive integer r, $\mathcal{F} = \binom{V}{\le r}$ corresponds to r-rounds of Lasserre relaxation.

Dual of left hand side is:

$$\begin{aligned} \max \quad & \Phi + \tfrac{\gamma}{\varepsilon} \\ \text{st} \quad & \Phi \cdot \widehat{L_H} + \gamma \le_{\Sigma_{\mathcal{F}}} \widehat{L_G}, \\ & \Phi \in \mathbb{R}, \gamma \in \mathbb{R}_+. \end{aligned} \tag{11.5}$$

Let's verify the conditions from Corollary 11.5.

(1) Let $\mathbf{x} \in \{0,1\}^V$ be $\mathbf{x}_u = 1$ and $\mathbf{x}_{V\setminus\{u\}} = 0$. By assumption, H has no isolated node so $\mathbf{x}^T L_H \mathbf{x} > 0$. Furthermore $\mathbf{x}^T L_H \mathbf{x} \le \mathrm{Tr}(L_H) < \frac{1}{\varepsilon}$. Hence the moment sequence $y = [y_S]$ defined as $y_S \leftarrow \frac{\mathbf{x}^S(\mathbf{x})}{\mathbf{x}^T L_H \mathbf{x}}$ is feasible.

Similarly $\Phi \leftarrow 0$ and $\gamma \leftarrow 0$ forms a dual feasible solution since $L_G \succeq 0$.

(2) $L_G \succeq 0$ thus primal is bounded from below by 0.

(3) $\Sigma^*_{\mathcal{F}}$ is a closed. Primal cone is given by:

$$\left\{ \left(\langle\!\langle \widehat{L_G}, y \rangle\!\rangle, \langle\!\langle \widehat{L_H}, y \rangle\!\rangle, y_\emptyset \right) \middle| y \in \Sigma^*_{\mathcal{F}} \right\}.$$

For any $y \in \Sigma_{\mathcal{F}}^*$ in its kernel, we have $y_\emptyset = 0$. By Claim 3.22, this implies $y = 0$. Using Lemma 11.10 we see that primal cone is *closed.* Dual cone is:

$$\left\{ \left(\Phi + \frac{\gamma}{\varepsilon}, \Phi \cdot \widehat{L_H} + \gamma \right) \middle| \Phi \cdot \widehat{L_H} + \gamma \succeq_{\Sigma_{\mathcal{F}}} 0,\ \Phi \in \mathbb{R}, \gamma \in \mathbb{R}_+ \right\}.$$

It is easy to see that the cone $\left\{ (\Phi, \gamma) \middle| \Phi \cdot \widehat{L_H} + \gamma \succeq_{\Sigma_{\mathcal{F}}} 0, \Phi \in \mathbb{R}, \gamma \in \mathbb{R}_+ \right\}$ is closed. For any (Φ, γ) in the kernel:

$$\begin{aligned} 0 =& \Phi + \frac{\gamma}{\varepsilon} \implies \Phi = -\frac{\gamma}{\varepsilon} \leq 0. \\ 0 \equiv& \Phi \cdot \widehat{L_H} + \gamma \equiv -\frac{\gamma}{\varepsilon} \widehat{L_H} + \gamma \\ \implies & \gamma \left(\frac{\widehat{L_H}}{\varepsilon} - 1 \right) \equiv 0 \implies \gamma = 0 \implies \Phi = 0. \end{aligned}$$

By Proposition 11.9, we see that dual cone is closed.

Optimality conditions are (stated in terms of normalized moment sequence $x_\emptyset$):

$$\begin{aligned} 0 =& \langle\!\langle \widehat{L_G} - \Phi \widehat{L_H} - \gamma, x \rangle\!\rangle, \\ 0 =& \gamma \left(\frac{1}{\varepsilon} - \langle\!\langle \widehat{L_H}, x \rangle\!\rangle \right). \end{aligned}$$

For sake of contradiction, assume $\gamma \neq 0$ in an optimal solution. Then $\frac{1}{\varepsilon} = \langle\!\langle \widehat{L_H}, x \rangle\!\rangle \leq \mathrm{Tr}(L_H) < \frac{1}{\varepsilon}$, a contradiction, therefore $\gamma = 0$. $\square$

Chapter 12

Conclusion

In this thesis, we developed new approaches for rounding solutions of relaxations based on Lasserre Hierachy for many fundamental graph partitioning problems such as Non-Uniform Sparsest Cut, Minimum Bisection, k-Unique Games, etc... We related the quality of solutions constructed by the rounding to column based matrix reconstruction problem for which we proved optimal bounds on the number of columns necessary as well as gave efficient deterministic and randomized algorithms. By exploiting the way our rounding algorithms work, we also gave a recursive ellipsoid based algorithm only constructs the relevant portion of solution read by the rounding, effectively reducing the running times from $n^{O(r)}$ to $2^{O(r)}n^{O(1)}$.

For many problems we studied, there were no known way to obtain constant factor approximation even for restricted classes of graphs. Our algorithms are the first in this sense: Provided that graph spectrum increases relatively fast, we proved that all our algorithms achieve constant factor approximation.

We believe that the research presented in this thesis opens up a lot of interesting directions to pursue. However our main question still remains open:

> *"Is there a constant factor approximation algorithm for any of the problems we studied running in quasi-polynomial time?"*

Now we survey some future research directions, all based on trying to understand the above question better.

Approximation Guarantees Independent of Spectrum. One intermediate question one can ask is whether our approximation algorithms achieve factors such as $O(\log n)$ or $O(\sqrt{\log n})$.

Other Possibilities for Conditioning. The rounding framework we proposed is based on conditioning partial labelings on seeds. But are these the only events we can condition on? For example, Karlin et al. [2010] showed how to "derive" variables corresponding to larger subsets for knapsack problem. Is there such variables we can use?

Handling Graphs with Bounded Tree-width or Genus. These are two natural graph classes whose spectrum is worst possible for us: Number of eigenvalues smaller than, say expansion, is on polynomial in number of nodes! On the other hand, for such graphs there are efficient algorithms which achieve constant factor (or better) approximation. Moreover Chlamtac et al. [2010] showed that a weaker hierarchy allows decent approximation for sparsest cut on bounded tree-width.

Exact Solver for Relaxations. Unlike LP, no optimal (or exact) solver for SDP is known. However our SDP relaxations are nicely structured, some of which we showed in Chapter 11. Therefore it might be possible to devise exact solvers for such relaxations, while avoiding the difficulties faced in exactly solving generic SDPs.

Handling Small Set Expander Graphs. Despite recent progress on lower bounding graph spectrum in terms of small set expansion [Arora et al., 2010, Lee et al., 2012, Louis et al., 2012, O'Donnell and Witmer, 2012], the bounds still fall too short to be useful for us in disproving small set expansion conjecture. But such a detour might not be necessary at all: Can we bound the projection distance assuming underlying graph is small set expander?

Bibliography

Christoph Ambühl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM J. Comput.*, 40(2):567–596, 2011. 6

Reid Andersen and Kevin J. Lang. An algorithm for improving graph partitions. In *SODA*, pages 651–660, 2008. 106, 120

Sanjeev Arora and Rong Ge. New tools for graph coloring. In *APPROX-RANDOM*, pages 1–12, 2011. 57, 73, 74

Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998. 5

Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998. 5

Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy. In *STOC*, pages 21–28, 2008a. 47, 96

Sanjeev Arora, James Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. *J. American Mathematical Society*, 21(1):1–21, 2008b. 106

Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2), 2009. 4, 106

Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for Unique Games and related problems. In *FOCS*, pages 563–572, 2010. 106, 120, 152

Yonatan Aumann and Yuval Rabani. An $o(\log k)$ approximate Min-Cut Max-Flow theorem and approximation algorithm. *SIAM J. Comput.*, 27(1):291–301, 1998. 106

Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(1):27–43, 2011. 5

Baruch Awerbuch and David Peleg. Sparse partitions (extended abstract). In *FOCS*, pages 503–513, 1990. 1

Francisco Barahona, Martin Grötschel, Michael Jünger, and Gerhard Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36(3):493–513, 1988. 2

Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *FOCS*, April 2011. 56, 57, 96, 97

Boaz Barak, Aram Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, Sum-of-Squares proofs, and their applications. In *STOC*, 2012. To appear. 7

Alexander Barvinok. *A Course in Convexity*, volume 54 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002. 142, 145

Aditya Bhaskara, Moses Charikar, Aravindan Vijayaraghavan, Venkatesan Guruswami, and Yuan Zhou. Polynomial integrality gaps for strong sdp relaxations of densest *k*-subgraph. In *SODA*, pages 388–405, 2012. 4

Sandeep N. Bhatt and Frank Thomson Leighton. A framework for solving VLSI graph layout problems. *J. Comput. Syst. Sci.*, 28(2):300–343, 1984. 1

Jonathan Borwein and Adrian S. Lewis. *Convex Analysis and Nonlinear Optimization: Theory and Examples*, volume 3 of *CMS Books in Mathematics*. Springer, 2000. 142, 145

Jonathan M. Borwein and Warren B. Moors. Stability of closedness of convex cones under linear mappings. *J. Convex Analysis*, 16(3):699–705, 2009. 145

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal column-based matrix reconstruction. In *FOCS*, 2011. 50, 126, 127, 128

Peter Bürgisser, Michael Clausen, and Mohammad Shokrollahi. *Algebraic Complexity Theory*. Springer, 2010. 136

Ali Çivril and Malik Magdon-Ismail. Deterministic sparse column based matrix reconstruction via greedy approximation of SVD. In *ISAAC*, pages 414–423, 2008. 127

Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating Multicut and Sparsest-Cut. *Computational Complexity*, 15(2):94–114, 2006. 5

Shuchi Chawla, Anupam Gupta, and Harald Räcke. Embeddings of negative-type metrics and an improved approximation to generalized sparsest cut. *ACM T. Algorithms*, 4(2), 2008. 106

Eden Chlamtac. Approximation algorithms using hierarchies of semidefinite programming relaxations. In *FOCS*, pages 691–701, 2007. 4

Eden Chlamtac and Gyanit Singh. Improved approximation guarantees through higher levels of sdp hierarchies. In *APPROX-RANDOM*, pages 49–62, 2008. 4, 57, 73

Eden Chlamtac and Madhur Tulsiani. Convex relaxations and integrality gaps. In *Handbook on Semidefinite, Cone and Polynomial Optimization*, 2011. URL http://www.cs.princeton.edu/~chlamtac/sdpchapter.pdf. 4, 7

Eden Chlamtac, Robert Krauthgamer, and Prasad Raghavendra. Approximating sparsest cut in graphs of bounded treewidth. In *APPROX-RANDOM*, pages 124–137, 2010. 56, 152

Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *FOCS*, pages 329–338, 2010. 126, 127, 128, 129, 137, 138

Amit Deshpande and Santosh Vempala. Adaptive sampling and fast low-rank matrix approximation. In *APPROX-RANDOM*, pages 292–303, 2006. 127, 128, 129, 138, 140

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. *Theory of Computing*, 2(1):225–247, 2006. URL http://www.theoryofcomputing.org/articles/v002a012. 127, 128, 129

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast Monte-Carlo algorithms for finding low-rank approximations. *J. ACM*, 51:1025–1041, November 2004. 127

Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. 3, 18

Gene H. Golub and Charles F. Van Loan. *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)*. The Johns Hopkins University Press, 3rd edition, October 1996. ISBN 0801854148. 126

Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1993. 2, 18, 60, 61, 62

Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *FOCS*, 2011. 11

Venkatesan Guruswami and Ali Kemal Sinop. Lasserre SDPs, ℓ_1-embeddings, and approximating non-uniform sparsest cut via generalized spectra. Submitted, 2012a. 12

Venkatesan Guruswami and Ali Kemal Sinop. Optimal column-based low-rank matrix reconstruction. In *SODA*, 2012b. 12

Venkatesan Guruswami and Ali Kemal Sinop. Faster SDP hierarchy solvers for local rounding algorithms. Submitted, 2012c. 11

Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the random ordering is hard: Every ordering CSP is approximation resistant. *SIAM J. Comput.*, 40(3):878–914, 2011. 5

Venkatesan Guruswami, Ali Kemal Sinop, and Yuan Zhou. Constant factor lasserre integrality gaps for graph partitioning problems. *CoRR*, abs/1202.6071, 2012. 4

Magnús M. Halldórsson. Approximations of independent sets in graphs. In *APPROX*, pages 1–13, 1998. 90

Eran Halperin. Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM J. Comput.*, 31(5):1608–1623, 2002. 90

Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. 5

Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991. 139

Anna R. Karlin, Claire Mathieu, and C. Thach Nguyen. Integrality gaps of linear and semi-definite programming relaxations for knapsack. *CoRR*, abs/1007.1283, 2010. 4, 40, 57, 152

Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775, 2002. 5

Subhash Khot. Ruling out PTAS for Graph Min-Bisection, Dense k-Subgraph, and Bipartite Clique. *SIAM J. Comput.*, 36(4):1025–1071, 2006. 6

Subhash Khot and Rishi Saket. SDP integrality gaps with local ℓ_1-embeddability. In *FOCS*, pages 565–574, 2009. 7

Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. 5

Subhash Khot, Preyas Popat, and Rishi Saket. Approximate Lasserre integrality gap for unique games. In *APPROX-RANDOM*, pages 298–311, 2010. 96

Jean B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM J. Optimization*, 12(3):756–769, 2002. 4, 7, 9, 37

Monique Laurent. A comparison of the Sherali-Adams, Lovász-Schrijver, and Lasserre relaxations for 0-1 programming. *Math. Oper. Res.*, 28(3):470–496, 2003. 7, 71

James R. Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. In *STOC*, 2012. to appear. 152

Frank Thomson Leighton and Satish Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *FOCS*, pages 422–431, 1988. 106

Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995. 106, 115

Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bull. EATCS*, 105:41–72, 2011. 71

Anand Louis, Prasad Raghavendra, Prasad Tetali, and Santosh Vempala. Many sparse cuts via higher eigenvalues. In *STOC*, 2012. to appear. 152

László Lovász and Alexander Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. Optimization*, 1:166–190, 1991. 4, 7

Michael Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proc. of the National Academy of Sciences USA*, 106:697–702, 2009. 126

Rajsekar Manokaran, Joseph Naor, Prasad Raghavendra, and Roy Schwartz. SDP gaps and UGC hardness for Multiway Cut, 0-extension, and Metric Labeling. In *STOC*, pages 11–20, 2008. 5

Albert W. Marshall, Ingram Olkin, and Barry C. Arnold. *Inequalities: Theory of Majorization and its Applications*. Springer, 2009. 129, 131

Ryan O'Donnell and David Witmer. Improved small-set expansion from higher eigenvalues. *CoRR*, abs/1204.4688, 2012. 152

Lorant Porkolab and Leonid Khachiyan. On the complexity of semidefinite programs. *J. Global Optimization*, 10:351–365, 1997. 60

Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *STOC*, pages 245–254, 2008. 5, 96

Prasad Raghavendra and David Steurer. Integrality gaps for strong SDP relaxations of Unique Games. In *FOCS*, pages 575–585, 2009. 7

Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *STOC*, pages 755–764, 2010. 6

Prasad Raghavendra and Ning Tan. Approximating CSPs with global cardinality constraints using SDP hierarchies. In *SODA*, 2012. 55, 57, 73, 74, 97

Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions between expansion problems. In *CCC*, 2012. 6

Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. 5

Thomas Rothvoß. Directed Steiner Tree and the Lasserre hierarchy. *CoRR*, abs/1111.5473, 2011. 73

Tamás Sarlós. Improved approximation algorithms for large matrices via random projections. In *FOCS*, pages 143–152, 2006. 127

Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *FOCS*, pages 593–602, 2008. 4

Issai Schur. Uber eine klasse von mittelbindungen mit anwendungen in der determinantentheorie. *Sitzungsber Math Gesellschaft*, 22:9–20, 1923. 131

Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Mathematics*, 3:411–430, 1990. 4, 7

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000. 1

Ali Kemal Sinop and Leo Grady. Uninitialized, globally optimal, graph-based rectilinear shape segmentation: The Opposing Metrics method. In *ICCV*, pages 1–8, 2007. 1

Luca Trevisan, Gregory B. Sorkin, Madhu Sudan, and David P. Williamson. Gadgets, approximation, and linear programming. *SIAM J. Comput.*, 29(6):2074–2097, 2000. 5

Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *STOC*, pages 303–312, 2009. 4

Vijay V. Vazirani. *Approximation algorithms*. Springer, 2001. ISBN 978-3-540-65367-7. 2