Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 106 (2019)

Semialgebraic Proofs and Efficient Algorithm Design

Noah Fleming[1]
University of Toronto
noahfleming@cs.toronto.edu

Pravesh Kothari
Princeton University
kothari@cs.princeton.edu

Toniann Pitassi[1]
University of Toronto & IAS
toni@cs.toronto.edu

October 19, 2019

[1]Research supported by NSERC.

ISSN 1433-8092

Abstract

Over the last twenty years, an exciting interplay has emerged between proof systems and algorithms. Some natural families of algorithms can be viewed as a generic translation from a proof that a solution exists into an algorithm for finding the solution itself. This connection has perhaps been the most consequential in the context of semi-algebraic proof systems and basic primitives in algorithm design such as linear and semidefinite programming. The proof system perspective, in this context, has provided fundamentally new tools for both algorithm design and analysis. These news tools have helped in both designing better algorithms for well-studied problems and proving tight lower bounds on such techniques.

This monograph is aimed at expositing this interplay between proof systems and efficient algorithm design and surveying the state-of-the-art for two of the most important semi-algebraic proof systems: Sherali-Adams and Sum-of-Squares.

We rigorously develop and survey the state-of-the-art for Sherali-Adams and Sum-of-Squares both as proof systems, as well as a general family of optimization algorithms, stressing that these perspectives are formal duals to one-another. Our treatment relies on interpreting the outputs of the Sum-of-Squares and Sherali-Adams algorithms as generalized expectation functions – a viewpoint that has been essential in obtaining both algorithmic results and lower bounds. The emphasis is on illustrating the main ideas by presenting a small fraction of representative results with detailed intuition and commentary. The monograph is self-contained and includes a review of the necessary mathematical background including basic theory of linear and semi-definite programming.

Contents

1 Introduction **3**
- 1.1 Proof Complexity Primer . . . 4
- 1.2 Proof Systems for UNSAT . . . 6
- 1.3 Algebraic Proof Systems . . . 10
- 1.4 Semialgebraic Proof Systems . . . 12
- 1.5 Connection between Algorithms and Proofs. . . . 13

2 Sherali-Adams **16**
- 2.1 Linear Programming . . . 16
  - 2.1.1 Variants of Linear Programming and Relaxations . . . 18
- 2.2 Sherali-Adams . . . 19
  - 2.2.1 Sherali-Adams as Lifting Linear Programs . . . 19
  - 2.2.2 Sherali-Adams as Locally Consistent Distributions . . . 25
    - 2.2.2.1 Pseudo-Expectations . . . 26
    - 2.2.2.2 Pseudo-Distributions . . . 30
    - 2.2.2.3 Evolution of the Sherali-Adams Relaxation . . . 33
  - 2.2.3 Sherali-Adams as a Proof System . . . 38
    - 2.2.3.1 Refutations of CNF Formulae and a Simulation of Resolution 39
    - 2.2.3.2 Soundness, Completeness, and Duality . . . 44
    - 2.2.3.3 Automatizability . . . 46

3 Sum-of-Squares **48**
- 3.1 Semidefinite Programming and PSD Matrices . . . 48
  - 3.1.1 The Ellipsoid Method . . . 50
  - 3.1.2 Positive Semidefinite Matrices . . . 55
    - 3.1.2.1 A Separation Oracle for SDPs . . . 61
  - 3.1.3 Semidefinite Programming Duality . . . 62
    - 3.1.3.1 A Proof of Strong Duality . . . 64
  - 3.1.4 Sum-of-Squares Polynomials and Semidefinite Programs . . . 67
- 3.2 Sum-of-Squares . . . 72
  - 3.2.1 Sum-of-Squares as Lifting Semidefinite Programs . . . 72
  - 3.2.2 Sum-of-Squares as Locally Consistent Distributions . . . 78
    - 3.2.2.1 Pseudo-Expectations . . . 78

3.2.2.2 Pseudo-Distributions . . . 81
3.2.2.3 Evolution of the Sum-of-Squares Relaxation . . . 82
3.2.3 Sum-of-Squares as a Proof System . . . 87
3.2.3.1 Soundness and Completeness . . . 92
3.2.3.2 Comparison With Sherali-Adams and Algebraic Proof Systems 94
3.2.3.3 Automatizability . . . 95
3.3 Generalizations of Sum-of-Squares . . . 97
3.3.1 Sum-of-Squares over $\mathbb{R}$ . . . 97
3.3.1.1 Duality, Completeness, and Convergence . . . 98
3.3.2 Positivstellensatz . . . 105
3.3.2.1 Positivstellensatz Refutations . . . 105
3.3.2.2 Positivstellensatz Certificates . . . 106

4 Upper Bounds via Sum-of-Squares 109
4.1 Max-Cut . . . 110
4.2 The Unique Games Conjecture and Sum-of-Squares . . . 114
4.3 Average-Case Algorithm Design via SoS . . . 116
4.3.1 Clustering Mixtures of Gaussians . . . 117
4.3.2 Algorithm Design Via SoS: a bird's eye view . . . 118
4.3.2.1 Certifying Good Clusterings: The 1D Case . . . 120
4.3.2.2 Certifying Good Clusterings: The Higher Dimensional Case 124
4.3.3 Inefficient Algorithm from Certifiability . . . 126
4.3.4 Efficient Algorithms via SoS-ized Certifiability . . . 129

5 Lower Bounds for Sum-of-Squares 138
5.1 3XOR . . . 138
5.2 Other SoS Lower Bounds . . . 143
5.3 Applications of Lower Bounds . . . 145

Bibliography 146

Appendix 159

Index 161

Chapter 1

Introduction

Proof complexity is the study of what can be proved *efficiently* [1] in a given formal proof system. Algorithm analysis is the quest for efficient and accurate algorithms for optimization problems, that can be rigorously analyzed. Over the last twenty years, there has been an exciting interplay between proof complexity and algorithms which in a nutshell studies the proof complexity of algorithm correctness/analysis. The main focus of this monograph is on algebraic and semi-algebraic proof systems, and the story of how they became closely connected to approximation algorithms. Indeed, we will argue that proof complexity has emerged as the study of systematic techniques to obtain provably correct algorithms.

There are two high level themes underlying this connection. *The first theme is that proof system lower bounds imply lower bounds for a broad family of related algorithms.* A proof system, in a specific formal sense, corresponds to a family of efficient, provably correct algorithms. Thus, lower bounds in specific proof systems (showing hardness of proving well-definedness or other key properties of the function) rules out large classes of algorithms for solving NP-hard optimization problems.

One of the earliest appearances of this theme was in the work of Chvátal [46], which proved almost exponential lower bounds against a promising class of algorithms for independent set by studying an associated proof system. Another influential paper from 2006, aptly titled paper "Proving Integrality Gaps without Knowing the Linear Program" explicitly demonstrated the potential of this theme [5]. This paper considered broad classes of linear relaxations for NP-optimzation problems, and proved nearly tight integrality gaps for several important problems (VertexCover, MaxSAT, and MaxCut) for *any* linear relaxation from the class. The classes that they considered correspond to the algorithms that underlie the semi-algebraic proof systems SA (Sherali-Adams) and LS (Lovász-Schrijver).

Since then, there has been a huge body of work, proving integrality gaps for large families of linear programming and semidefinite programming-based algorithms for a variety of important NP-hard optimization problems. These integrality gaps are none other than proof complexity lower bounds for specific families of formulas. Most notably are the proof

[1] The emphasis on efficiency, as opposed to existence, is what distinguishes proof complexity from classical proof theory, and also what links proof complexity with complexity theory and algorithms.

systems Polynomial Calculus (PC) which is gives rise to a family of algebraic algorithms, Sherali-Adams (SA) which gives rise to a large family of linear programs, and Sum-of-Squares (SoS), which gives rise to a large family of semidefinite programs. In another exciting line of work, lower bounds for SA and SoS form the basis of exponential lower bounds on the size of extended formulations (and positive semi-definite extended formulations) for approximating MaxCut as well as for other NP-hard optimization problems.

The second theme is that (sometimes) proof system upper bounds can automatically generate efficient algorithms. More specifically, a proof system is said to be *automatizable* if there is an algorithm that can find proofs in that system efficiently, in the size of the shortest proof. (So if there is a short proof in the system, then it can be found efficiently as well.) SA is *degree* automatizable, in the sense that if there is a degree d proof, then it can be found time $n^{O(d)}$. SoS is also practically *degree* automatizable, if we assume that the coefficients have length bounded by a polynomial in n (or can be sufficiently well approximated). [2] In an automatizable proof system, an efficient proof certifying the existence of a solution automatically implies an efficient algorithm for the problem. Using this theme, several remarkable recent papers have obtained new algorithms for unsupervised learning problems via efficient SoS proofs.

In the rest of this introduction, we give a brief tour of proof complexity including an introduction to the algebraic and semi-algebraic proof systems that we will focus on in this monograph, from the proof complexity point of view.

1.1 Proof Complexity Primer

Proof complexity refers to the study of nondeterministic algorithms for solving problems in *co*NP. Abstractly, let $\mathcal{L}$ be a language in *co*NP. For example, $\mathcal{L}$ could be the set of all undirected graphs that are *not* 3-colorable. The following definition of a proof system was given by Cook and Reckhow in their seminal paper introducing the key ideas behind the field [50].

Definition 1.1 (Propositional Proof System)**.** A propositional proof system for a language $\mathcal{L} \subseteq \{0,1\}^*$ is a polynomial-time function $\mathcal{P} : \{0,1\}^* \to \{0,1\}^*$ such that the following properties hold:

(1) (Soundness) For every y, $\mathcal{P}(y) \in \mathcal{L}$;

(2) (Completeness) For every $x \in \mathcal{L}$, there exists a y such that $\mathcal{P}(y) = x$.

We think of $\mathcal{P}$ as an efficient algorithm that checks to see if y encodes a legal proof that some x is in $\mathcal{L}$. If so, then $\mathcal{P}(y)$ outputs x; otherwise (if y does not code a legal proof), then $\mathcal{P}(y)$ outputs some canonical string $x \in \mathcal{L}$. The soundness property guarantees that the proof system can only produce proofs for strings in $\mathcal{L}$ and the completeness property means that every $x \in \mathcal{L}$ has a proof.

[2] See the discussion in Section 3.2.3.3.

Definition 1.2 (Proof Size)**.** Let $\mathcal{L} \subseteq \{0,1\}^*$ and let $\mathcal{P}$ be a proof system for $\mathcal{L}$. For $x \in \mathcal{L}$ $size_{\mathcal{P}}(x)$ is the minimal natural number m such that there exists y, $|y| = m$ and $\mathcal{P}(y) = x$. In other words, $size_{\mathcal{P}}(x)$ is the length of the shortest $\mathcal{P}$-proof of x.

Definition 1.3 (Polynomially Bounded Proof System)**.** $\mathcal{P}$ is *polynomially bounded* (or p-bounded) if for sufficiently large n, for all $x \in \mathcal{L}$, $|x| \geqslant n$, $size_{\mathcal{P}}(x) \in |x|^{O(1)}$.

It is easy to see from the definitions that there exists a polynomially bounded proof system $\mathcal{P}$ for a language $\mathcal{L}$ if and only if $\mathcal{L}$ is in NP: The NP algorithm on input x simply guesses some y (of polynomial length) and accepts if and only if $\mathcal{P}(y) = x$. When $x \in \mathcal{L}$, since $\mathcal{P}$ is complete and polynomially bounded, there is some y of polynomial length such that $\mathcal{P}(y) = x$ and conversely if $x \notin \mathcal{L}$ then by soundness, for every y, $\mathcal{P}(y) \neq x$. In the other direction, any NP algorithm for $\mathcal{L}$ gives rise to a polynomially bounded proof system for $\mathcal{L}$. Since the existence of a polynomially bounded proof system for a language $\mathcal{L}$ is equivalent to saying that $\mathcal{L}$ is in NP, proving that *no* polynomially bounded proof system exists for a *co*NP-complete language (such as UNSAT) is equivalent to proving $\mathsf{NP} \neq$ *co*NP. This is a daunting task – in particular, it implies that $\mathsf{P} \neq \mathsf{NP}$.

In light of the difficulty of proving that *no* proof system for UNSAT (or for any other *co*NP-complete language) is polynomially bounded, much of the research in the field has focused on proving superpolynomial lower bounds for *standard* proof systems for the underlying *co*NP-complete language $\mathcal{L}$. For example, Resolution, Frege and Extended Frege are standard proof systems for UNSAT; the Hajos calculus is a natural proof system for proving non k-colorability of graphs; Cutting Planes, Sherali-Adams and SOS (Sum-of-Squares) are well-studied proof systems for proving that a set of linear inequalities has no integer solution; and Nullstellensatz and Polynomial Calculus are proof systems for showing that a set of polynomial equations has no common zero/one solution. In the subsequent section, we will describe many of these proof systems and give some concrete examples.

A key high-level point is that the natural proof systems are well-studied for a good reason and this is the link between proof complexity and algorithms. Namely, the best algorithms for $\mathcal{L}$, both exact and approximate, are usually associated with a natural proof system in the sense that the transcript of the algorithm on some $x \in \mathcal{L}$ *is* a proof y in the associated proof system! Moreover, we can also associate transcripts of approximation algorithms with proofs. Therefore, proving lower bounds for well-studied proof systems for $\mathcal{L}$ is tightly connected to our understanding of how well large and natural classes of algorithms can solve or approximate the optimization problem.

The next definition allows us to compare the relative strength of different proof systems for the same language $\mathcal{L}$.

Definition 1.4 (p-Simulation)**.** Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two propositional proof systems for $\mathcal{L}$. We say that $\mathcal{P}_1$ p-simulates $\mathcal{P}_2$ if there exists a polynomial q such that for sufficiently large n, for all $x \in \mathcal{L}$, $|x| \geqslant n$, $size_{\mathcal{P}_1}(x) \leqslant q(size_{\mathcal{P}_2}(x))$. In other words, $\mathcal{P}_1$ p-simulates $\mathcal{P}_2$ if for every $x \in \mathcal{L}$, the minimum proof length in $\mathcal{P}_1$ is at most polynomially larger than the minimum proof length in $\mathcal{P}_2$. $\mathcal{P}_1$ and $\mathcal{P}_2$ are p-equivalent if $\mathcal{P}_1$ p-simulates $\mathcal{P}_2$, and $\mathcal{P}_2$ also p-simulates $\mathcal{P}_1$.

While proof size is important, it is also important to be able to *find* a proof quickly. Given the likelihood that all proof systems for *co*NP-hard languages are not polynomially-bounded, we should measure the complexity of finding a proof with respect to the size of the shortest proof, which motivates the next definition.

Definition 1.5 (Polynomial Automatizibility)**.** A proof system $\mathcal{P}$ for $\mathcal{L}$ is *polynomially automatizable* if there exists an algorithm A that takes as input $x \in \mathcal{L}$ and returns a y such that $\mathcal{P}(y) = x$ and moreover, the runtime of A is polynomial in $size_P(x)$ – that is, the runtime is polynomial in the size of the shortest $\mathcal{P}$-proof of x.

Shortly, we define *algebraic* and *semialgebraic* proof systems for proving that a system of polynomial equations or inequalities has no integral solution. For these systems, proofs will consist of a sequence of polynomial equations/equalities. For these proof systems, we are interested not only in proof size (the total length of the proof), but also in the degree of the proof – the minimal degree d such that there is an algebraic proof where every polynomial in the proof has degree at most d. Thus, we define the following degree-based variant of automatizability.

Definition 1.6 (Degree Automatizibility)**.** An algebraic proof system $\mathcal{P}$ is *degree automatizable* if there is an algorithm A that returns a $\mathcal{P}$-refutation of f in time $n^{O(deg_{\mathcal{P}}(f))}$, where $deg_{\mathcal{P}}(f)$ is the minimal degree refutation of f in $\mathcal{P}$.

Derivations versus Refutations. We have defined proof systems as nondeterministic procedure for proving that $x \in \mathcal{L}$ where $\mathcal{L}$ is a language in *co*NP. What if we want to consider instead proofs of derivations, such as a proof that if $x \notin \mathcal{L}$, then $x' \notin \mathcal{L}$. Of course we can always determine if this implication is true by a reduction to our nondeterministic procedure for $\mathcal{L}$. For example, if $\mathcal{L}$ is UNSAT, and we want to prove that if x is a satisfiable Boolean formula, then x' is also satisfiable, then we can do so indirectly by obtaining a proof that $\neg(\neg x \vee x') \in UNSAT$. However, it will often be more convenient to work directly with proofs of derivations (for example, when we want to study the proof complexity of approximation algorithms). To this end, we define a proof system for derivations as a polynomial-time function $\mathcal{P}$ from strings (encodings of proofs) to strings (encodings of implications of the form $x \to x'$), with the property that the range of $\mathcal{P}$ is exactly the set of all valid implications. (An implication $x \to x'$ is valid if $x \notin \mathcal{L}$, then $x' \notin \mathcal{L}$). The special case of refutations/proofs then corresponds to implications where x' is empty.

1.2 Proof Systems for UNSAT

UNSAT is the language consisting of all unsatisfiable Boolean formulas. Thus, x is an encoding of a Boolean formula, and a proof system verifies the unsatisfiability of x, or equivalently it could verify that x is a Boolean tautology. Since any formula can be efficiently converted into an equivalent formula in conjunctive normal form (CNF), we will without loss of generality, focus our discussion on propositional proof systems for k-UNSAT – verifying the

unsatisfiability of k-CNF formulas. A k-CNF formula $\mathcal{C}$ over Boolean variables $x_1, \ldots, x_n$ is a conjunction of clauses, $C_1, \ldots, C_m$, where each clause is a disjunction of k literals. We will often view a k-CNF formula as a set of clauses or constraints. A set of clauses $\{C_1, \ldots, C_m\}$ is *satisfiable* if there exists a Boolean assignment α to the underlying variables such that every clause C_i evaluates to true under α; otherwise the set of clauses are *unsatisfiable*.

Typical propositional proof systems for UNSAT are *axiomatic*, meaning that they are described by a finite set of syntactic derivation rules, which describe how to derive new formulas from one or two previous ones. In an axiomatic system, a proof that a CNF formula $\mathcal{C}$ (over $x_1, \ldots, x_n$) is unsatisfiable will be (an encoding of) a sequence of formulas, where each formula in the sequence is either one of the initial clauses C_i, or follows from previous formulas in the sequence by one of the rules. Finally, the last formula in the sequence should be a canonical formula that is trivially unsatisfiable. (For example, the formula "0" or the formula $x \wedge \neg x$.)

Resolution. The Resolution proof system (more commonly called a refutation system since we are refuting the existence of a satisfying assignment) is one of the most well-known propositional proof systems and forms the basis for many well-known automated theorem provers and SAT solvers. There is only one rule (the Resolution rule): $(A \vee x),\ (B \vee \neg x) \to (A \vee B)$, where A and B are clauses, and by $A \vee B$ we mean the clause obtained by taking the disjunction of all literals occurring in A or B, removing duplications. For example, we can derive $(x_1 \vee x_2)$ from $(x_1 \vee x_3)$ and $(x_1 \vee x_2 \vee \neg x_3)$. A Resolution refutation of a k-CNF formula $\mathcal{C} = \{C_1, \ldots, C_m\}$ is a sequence of clauses such that every clause in the sequence is either an initial clause C_i, $i \in [m]$, or follows from two previous clauses by the Resolution rule, and such that the final clause is the empty clause. Resolution is sound and complete: a CNF formula $\mathcal{C}$ has a Resolution refutation if and only if $\mathcal{C}$ is unsatisfiable.

We will encode a Resolution refutation by encoding each clause in the sequence; since each clause has length $O(n)$, the number of clauses in the refutation is polynomially related to the bit-length encoding. Thus, for simplicity and without loss of generality we take the number of clauses to be the length of a Resolution refutation. If the initial formula is k-CNF formula for k constant, then its length is polynomial in n, the number of underlying variables. Therefore, a Resolution refutation of a k-CNF over n variables is polynomially bounded if its length is polynomial in n.

Example. Consider the family of propositional formulas, $\{\mathsf{IND}_n,\ n \geqslant 2\}$ corresponding to the induction principle. We have n variables associated with IND_n, $x_1, \ldots, x_n$, and the following clauses: (1) (x_1); (2) For all $i < n$ $(\neg x_i \vee x_{i+1})$; (3) $(\neg x_n)$. For each n, IND_n has size polynomial in n and we say that IND_n has efficient Resolution refutations if for n sufficiently large, IND_n has a Resolution refutation of polynomial size in n. It is not hard to see that there are Resolution refutations of IND_n

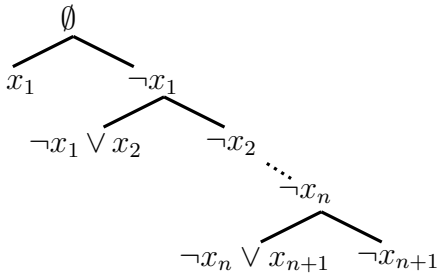


Figure 1.1: Resolution refutation of IND_n.

of linear length. More generally, any unsatisfiable Horn formula (CNF formula with at most one negated variable per clause) has efficient Resolution refutations, as does any unsatisfiable 2-CNF formula.

The well-known David-Putnam-Logemann-Loveland (DPLL) algorithms [52, 53] for satisfiability is our first example demonstrating the connection between proofs and algorithms. The DPLL algorithm is a complete, backtracking-based search algorithm for deciding the satisfiability of a CNF formula, $\mathcal{C}$. Whenever x is unsatisfiable, the transcript of the algorithm produces a decision tree over the underlying variables, where each leaf of the tree is labelled with some clause $C_i \in \mathcal{C}$ such that the partial truth assignment of the variables queried along the path to that leaf falsifies C_i. Such a decision tree, in turn, is actually a *tree-like* Resolution proof of $\mathcal{C}$. Therefore, running the DPLL algorithm on an unsatisfiable input x yields a Resolution proof that x is in Unsat. In the last decade, enormous progress has been made on practical SAT solvers, using more sophisticated backtracking algorithms, incorporating caching and restarts. In particular, the CDCL algorithm (Conflict Driven Clause Learning) routinely solves very large instances of SAT (with thousands of variables) efficiently [114]. Once again, the CDCL algorithm as well as many of its extensions are based on Resolution: running CDCL on an unsatisfiable input x yields a Resolution refutation of x. Thus superpolynomial lower bounds for Resolution proves unconditionally that DPLL, and CDCL are not in P.

We will now explain how Resolution characterizes a class of *approximation* algorithms for SAT. Any instance of 3SAT has an assignment satisfying at least 7/8th's of the clauses, and such an assignment can be found efficiently – so 3SAT has a polynomial-time 7/8-approximation algorithm. In a major result, Hastad proved that no polynomial-time algorithm can do better unless $\mathsf{P} = \mathsf{NP}$ – that is, he showed that it is NP-hard to achieve an approximation ratio of $7/8 + \varepsilon$, for $\varepsilon > 0$ [73]. Proof complexity provides a framework for proving *unconditional* lower bounds on approximation algorithms. Extensions of DPLL and CDCL have been developed for solving and for approximating MaxSAT which are again based on Resolution in the following sense. If $\mathcal{C}$ is an unsatisfiable 3CNF formula, then running a Resolution-based $(7/8+\varepsilon)$-approximation algorithm on input $\mathcal{C}$ will output a Resolution proof that C is not $(7/8 + \varepsilon)$ satisfiable. Again, known superpolynomial Resolution lower bounds on random unsatisfiable 3CNFs [47] can be invoked to prove unconditionally that *any* Resolution-based $(7/8 + \varepsilon)$ approximation algorithm is not in P.

Frege Proofs. The most well-known collection of proof systems for UNSAT, collectively referred to as Frege systems, are axiomatic systems typically presented in undergraduate logic textbooks. Lines in a Frege system are propositional formulas (usually over the standard basis $\{\wedge, \vee, \neg\}$). A Frege system is equipped with a finite set of axiom and rule schemas, and a Frege proof is a sequence of formulas, starting with axioms, and inferring new formulas from previous ones by applying these rules. Extended Frege systems are generalization of Frege systems where lines are boolean circuits (rather than formulas). Cook and Reckhow showed that standard propositional proof systems form a natural hierarchy, which mirrors the well-known circuit class hierarchy. At the bottom are Resolution proofs, where lines are clauses and thus they (roughly) correspond to depth-1 circuits; above that are bounded-depth Frege

systems where lines are bounded-depth formulas, and thus they correspond to bounded-depth AC_0 circuits. Similarly, Frege systems correspond to formulas (NC_1 circuits) and Extended Frege systems correspond to polynomial-size circuits. Thus as shown by Cook and Reckhow, bounded-depth Frege p-simulates Resolution, Frege p-simulates bounded-depth Frege, and Extended Frege p-simulates Frege.

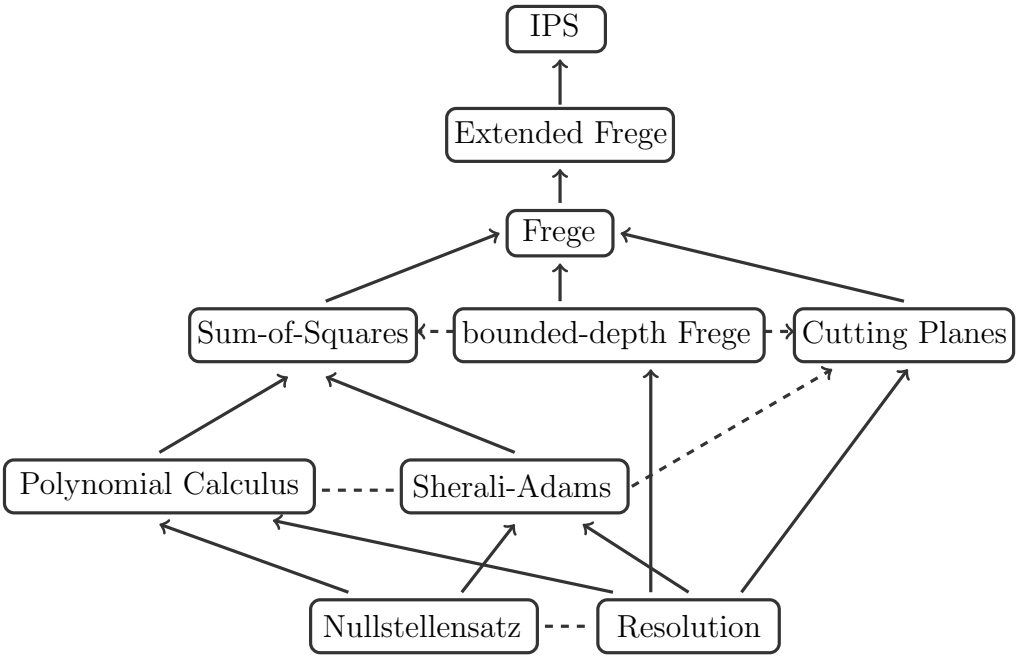


Figure 1.2: The hierarchy of standard propositional proof systems (not only those for UNSAT). An *arrow* $\mathcal{P}_1 \to \mathcal{P}_2$ indicates that $\mathcal{P}_2$ is strictly stronger than $\mathcal{P}_1$: $\mathcal{P}_2$ p-simulates $\mathcal{P}_1$ and there exists a formula which has polynomial-size proofs in $\mathcal{P}_2$ but which requires super-polynomial size to prove in $\mathcal{P}_1$. A *dashed arrow* from $\mathcal{P}_1$ to $\mathcal{P}_2$ implies that there is a formula which has short proofs in $\mathcal{P}_2$ but not in $\mathcal{P}_1$, but it is unknown whether $\mathcal{P}_2$ p-simulates $\mathcal{P}_1$. A *dashed line* between $\mathcal{P}_1$ and $\mathcal{P}_2$ indicates that $\mathcal{P}_1$ and $\mathcal{P}_2$ do not p-simulate each other.

Lower Bounds for UNSAT Proof Systems. In terms of lower bounds, Haken famously proved exponential lower bounds for Resolution (using the propositional pigeonhole principle as the hard formulas) [72]. Lower bounds for the Tseitin formulas (essentially random mod 2 equations) and for random k-CNF formulas were subsequently obtained by [149] and [47]. In [30] an even more general result was proven, showing that Resolution proof size could be reduced to Resolution width. (The width of a Resolution proof is the maximum clause size over all clauses in the proof.) Beyond Resolution, a landmark paper by Ajtai [1] proved superpolynomial lower bounds (again for the pigeonhole principle) for Frege systems of bounded depth, and in [26] this was improved to truly exponential lower bounds. It is a longstanding open problem to prove superpolynomial lower bounds for Frege systems. A comprehensive treatment of propositional proof complexity can be found in the following surveys [28, 140, 131].

Proof Search. Some non-trivial proof-search algorithms have been discovered for several weak proof systems for UNSAT. Beame and Pitassi [27] showed that any Resolution proof of size S can be found in time $n^{O(\sqrt{n \log S})}$. Ben-Sasson and Wigderson [30] noted that the same result follows from the reduction from proof size to width, by simply generating all clauses, according to the Resolution rule, of width bounded by the width of the proof. For tree-like Resolution, the size-width trade-off yields an automating algorithm which runs in quasi-polynomial time.

For stronger proof systems, a line of work has sought to rule out their automatizability under widely-believed cryptographic assumptions. This began with the work of Krajícek and Pudlák, who showed that Extended Frege is not automatizable unless RSA is not secure against polynomial size circuits [100]. Building on these ideas, Bonet et al. showed non-automatizability of Frege [35] and bounded-depth Frege systems [34] under the assumption that computing the Diffie-Hellman function cannot be computed by polynomial and sub-exponential size circuits respectively. For weaker proof systems, this approach seems less hopeful as it is not clear how to use the limited reasoning of these weak proof systems to break cryptographic assumptions leaving, in particular, the polynomial automatizability of Resolution as a tantalizing open problem. In an important paper, Alekhnovich and Razborov [2] showed that if Resolution, or even tree-like Resolution were polynomially automatizable, then the hierarchy of parameterized complexity classes would collapse; that is, $\mathsf{W[P]} = \mathsf{FPT}$. In a recent groundbreaking paper, Atserias and Müller [12] settled the question of automating Resolution proofs by showing that Resolution is not even sub-exponentially automatizable unless $\mathsf{P} = \mathsf{NP}$.

1.3 Algebraic Proof Systems

In this section we consider *algebraic* proof systems for proving that a system of polynomial equations/identities has no solution. For this language, we have a fixed ambient field or ring which is typically the integers or a finite field. An instance is now (an encoding of) a set of polynomial equations over the variables $x_1, \ldots, x_n$, and we want to prove that this set of polynomial equations has no solution over the underlying field.

Any algebraic proof system can also be viewed as a proof system for UNSAT, and more generally using polynomial-time reductions we can view a proof system for one *co*NP-complete language as a proof system for any other another *co*NP language (although it may not be a natural one). In the case of UNSAT, we translate each clause into an equivalent polynomial equation. For example $(x_1 \vee \neg x_2 \vee x_3)$ becomes the equation $(1-x_1)(x_2)(1-x_3) = 0$, and we can also add the equations $x_i^2 - x_i = 0$ in order to force only Boolean solutions.

Nullstellensatz and Polynomial Calculus. These proof systems are based on Hilbert's Nullstellensatz which states that a system of polynomial equations $p_1(x_1, \ldots, x_n), \ldots, p_m(x_1, \ldots, x_m)$ over a field F is unsatisfiable if and only if the ideal in the ring $F[x_1, \ldots, x_n]$ generated by $p_1(x), \ldots, p_m(x)$ contains 1. In other words, if and only

if there exists polynomials $q_1(x), \ldots, q_m(x)$ such that

$$p_1 q_1 + \ldots + p_m q_m = 1.$$

As mentioned above, in our standard context where the variables x_i range over the Boolean domain $\{0, 1\}$, we can enforce this by adding $x_i^2 - x_i$ to the list of polynomial equations. Alternatively we can just factor them out by working in the ring $F[x_1, \ldots, x_n]/(x_1^2 - x_1, \ldots, x_n^2 - x_n)$ of multilinear polynomials. Thus $q_1(x), \ldots, q_m(x)$ can be assumed to be multilinear and therefore of degree at most n.

A Nullstellensatz (Nsatz) refutation of $p_1(x), \ldots, p_m(x)$ is thus a set of polynomials $q_1, \ldots, q_m$ such that $\sum_i p_i q_i = 1$. It is clear that this proof satisfies the Cook-Reckhow definition: it is easy to check that $\sum_i p_i q_i = 1$, and soundness and completeness follow from Hilbert's Nullstellensatz.

The *size* of a Nullstellensatz refutation is the sum of the sizes of the $q_i(x)$'s. Another important measure for Nullstellensatz refutations is the *degree*, which is the maximal degree of the $p_i(x)q_i(x)$'s. As mentioned above, the degree is at most linear. An important property of Nullstellensatz refutations is that they are *degree automatizable*: If an initial family $p_1(x), \ldots, p_m(x)$ of polynomials has a degree d Nullstellensatz refutation, then it can be found in time $n^{O(d)}$ by simply solving a system of linear equations where the underlying variables of the equations are the coefficients of the monomials in $q_1(x), \ldots, q_m(x)$.

The Polynomial Calculus (PC) is a dynamic version of Nullstellensatz that is rule-based. The inference rules are: (i) from $f = 0$, $g = 0$ we can derive $\alpha f + \beta g = 0$, and (ii) from $f = 0$ we can derive $fg = 0$. The size of a PC proof is the sum of the sizes of all polynomials in the derivation, and the degree is the maximum degree of any line in the proof. Because this system is dynamic, it is sometimes possible (through cancellations) to obtain a much lower degree refutation than is possible using the static Nullstellensatz system. A great example is the induction principle IND_n mentioned above. It is not too hard to see that they have degree 2 PC refutations; on the other hand, it has been shown that their Nsatz degree is $\Theta(\log n)$ [38]. Clegg, Edmonds and Impagliazzo [48] proved that, like Nsatz, PC refutations are degree automatizable.

Stronger Algebraic Proof Systems. The Nullstellensatz and Polynomial Calculus proof systems witness the unsolvability of a set $\mathcal{P}$ of polynomial equations by demonstrating that 1 lies in the ideal generated by $\mathcal{P}$, where the measure of complexity of the proof is the maximal degree. More generally proofs can be viewed as directed acyclic graphs, where each line in the proof is either a polynomial from $\mathcal{P}$, or follows from two previous lines by taking a linear combination of two previous lines, or by multiplying a previous equation by a variable, and where the final line is the identically 1 polynomial. Thus, the entire proof can be viewed more compactly as an algebraic circuit, with the leaves labelled by polynomials from $\mathcal{P}$, and constants, and where internal vertices are either plus or times gates. This proof system (now called Hilbert-IPS) was introduced in [118] and is known to be quite powerful: it can efficiently simulate proofs in all standard propositional proof systems. However, it is not known to be a Cook-Reckhow proof system since proofs are not known to be verifiable in polynomial time. Determinining if a Hilbert-IPS circuit is a proof

amounts to determining if the polynomial that it computes is the identically-1 polynomial, and therefore verifying a Hilbert-IPS proof amounts to solving PIT (polynomial identity testing), a problem that admits a randomized (one-sided error) polynomial-time algorithm. A longstanding and important problem is to prove (or disprove) that PIT has a *deterministic* polynomial time algorithm.

A generalization of Hilbert-IPS called the IPS proof system (the Ideal Proof System) was introduced by Grochow and Pitassi [68]. IPS proofs have no rules – a proof of unsolvability of $\mathcal{P}$ is simply an algebraic circuit $\mathcal{C}$ with two types of inputs, $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, and subject to the following properties (which can be verified by a PIT algorithm): (i) $\mathcal{C}$ with zero substituted for each of the y_i variables evaluates to the identically zero polynomial; (ii) $\mathcal{C}$ with $p_1(x), \ldots, p_m(x)$ substituted for $y_1, \ldots, y_m$, computes the identically 1 polynomial. As for Hilbert-IPS, IPS are not known to be deterministically verifiable in polynomial time, and can simulate all standard Frege and Extended Frege systems. Grochow and Pitassi prove that superpolynomial lower bounds for IPS for any family of unsolvable polynomials $\mathcal{P}$ would resolve the longstanding problem of separating VP from VNP, thus establishing a connection between lower bounds in proof complexity and circuit lower bounds.

In [105] Tzemeret and Wang define a noncommutative version of IPS, and quite surprisingly, they prove that it is *equivalent* to standard Frege systems. In different but related work, Grigoriev and Hirsch [66] introduce an algebraic proof system with derivation rules corresponding to the ring axioms; unlike the IPS systems, proofs in their system can be verified in polynomial-time. (See [121] for a survey of algebraic proof systems.)

Lower Bounds and Proof Search. Lower bounds for Nsatz and PC are known and will be discussed in Chapter 6. For the stronger algebraic proof systems which can efficiently represent polynomials by algebraic circuits (Hilbert-IPS and IPS), there are no nontrivial lower bounds (although lower bounds have been proven for some restricted subsystems [55].) In terms of proof search, both Nsatz and PC are degree automatizable [49], using a modification of the Grobner basis algorithm.

1.4 Semialgebraic Proof Systems

In the previous section we discussed proof systems for proving that a system of polynomial identities is solvable. We can generalize to the semialgebraic setting, where now the input is a system of polynomial *inequalities*, and a semialgebraic proof should certify that the system of inequalities has no solution over the reals. This generalizes algebraic proofs (over the reals) since we can always write a polynomial equality as a set of two inequalities. And by translating clauses into a polynomial inequalities ($(x_1 \vee \neg x_2 \vee x_3)$ becomes $x_1 + x_3 \geqslant x_2$) we can also view semialgebraic proof systems as proof systems for UNSAT.

Cutting Planes. Cutting Planes is a semi-algebraic proof system over the integers, where the inequalities are *linear*. It was originally devised as a method for solving integer linear programs by relaxing them to fractional constraints (replacing $x_i \in \{0, 1\}$ by $0 \leqslant x_i \leqslant 1$), and then deriving new inequalities from previous ones via the Cutting Planes rules, as a way

to tighten the relaxed polytope, to whittle away at non-integral points. There are two rules for deriving new inequalities:

(i) Combination: From $\sum_i a_i x_i \geqslant \gamma$ and $\sum_i b_i x_i \geqslant \delta$, derive $\sum_i (\alpha a_i + \beta b_i) x_i \geqslant \alpha\gamma + \beta\delta$, for nonnegative integers α, β.

(ii) Division with rounding: From $\sum_i a_i x_i \geqslant \gamma$, derive $\sum_i \frac{a_i}{c} x_i \geqslant \lceil \frac{\gamma}{c} \rceil$, provided that c divides all a_i.

Sherali-Adams. Sherali-Adams (SA) will be the focus of Chapter 2. Like Cutting Planes, it is a semi-algebraic proof system over the integers, where lines are polynomial inequalities. Like Nsatz, it is a static proof system. Let $p_1(x), \ldots, p_m(x)$ be a set of polynomial inequalities that includes that polynomial inequalities $x_i^2 \geqslant x_i, x_i \geqslant x_i^2$ for all $i \in [n]$. A SA refutation of $p_1(x) \geqslant 0, \ldots, p_m(x) \geqslant 0$ is a set of polynomials $q_1(x), \ldots, q_m(x)$ such that each $q_i(x)$ is a non-negative combination of *non-negative juntas*, and such that $\sum_i p_i q_i = -1$. (A non-negative junta is a polynomial that corresponds to a conjunction of Boolean literals – for example, $x_1(1 - x_2)x_3$ is a non-negative junta that corresponds to the Boolean conjunction $x_1 \neg x_2 x_3$).

Sum-of-Squares. (SoS) is another static semi-algebraic proof system, and is the focus of Chapter 3. Again let $p_1(x), \ldots, p_m(x)$ be a set of polynomial inequalities that includes $x_i^2 - x_i = 0$ for all $i \in [n]$. An SoS refutation of $p_1(x) \geqslant 0, \ldots, p_m(x) \geqslant 0$ is a set of polynomials $q_1(x), \ldots, q_m(x)$ such that each q_i is a *sum-of-squares* – that is, $q_i(x)$ can be written as $\sum_j (r_{i,j}(x))^2$ for some polynomials $r_{i,j}(x)$, and such that $\sum_i p_i q_i = -1$. It is not hard to see that non-negative combinations of non-negative juntas are sum-of-squares, and thus the SoS proof system extends SA.

Lower Bounds and Proof Search. We will discuss lower bounds for semialgebraic proof systems in detail in Chapter 6. In terms of proof search, both SA and SoS are degree automatizable refutation systems (under some niceness conditions) which makes them extremely useful for designing good approximation algorithms. Indeed developing this connection is one of the main themes of this monograph.

1.5 Connection between Algorithms and Proofs.

Having described the specific proof systems that will be the highlight of the connection between proof complexity and algorithms, we will return to the two themes underlying this connection.

Suppose that you have a correct algorithm A for solving SAT (or some other NP-hard optimization problem). By correct we mean that when run on a satisfiable instance, the algorithm will always output "satisfiable" and when run on an unsatisfiable formula, the algorithm will always output "unsatisfiable". Since the algorithm is correct, on any unsatisfiable formula f, we can view the computation of the algorithm on input f as a *proof* of the unsatisfiability of f. Furthermore, if A's run on f is efficient (say polynomial-time),

then A provides us with a polynomial-length refutation that f is unsatisfiable. Moreover, the same idea applies to approximation algorithms. That is, suppose that we have a correct 2-approximation algorithm for some optimization problem, such as Independent Set. That is, for every graph G, it outputs an independent set of size at least half of the size of the largest independent set in G. If we run the algorithm on some graph G and it returns an independent set of size s, this is a proof that G does not contain an independent set of size greater than $2s$.

Now speaking somewhat informally, if A is correct, then there is a proof of A's correctness. This proof of correctness of A, when applied to each unsatisfiable f gives us a propositional proof that f is unsatisfiable, in some proof system, let's call it P_A. Therefore, superpolynomial lower bounds for this proof system show that there can be no polynomial-time algorithm whose proof of correctness is based on P_A.

As an example of this paradigm, we explained how state-of-the-art complete algorithms for SAT are based on Resolution, and thus Resolution lower bounds imply similar impossibility results for a broad class of algorithms for SAT. For stronger proof systems such as Frege and Extended Frege, what is the corresponding class of algorithms? It turns out that Extended Frege proofs capture nearly all known provably correct algorithms! Therefore, superpolynomial lower bounds for Extended Frege systems would have far-reaching consequences: it would essentially rule out almost all known algorithms and algorithmic paradigms for efficiently solving any NP-hard problem.

Unfortunately, at present we do not know how to prove superpolynomial Extended Frege lower bounds or superpolynomial Frege lower bounds. But luckily, for many of the algebraic and semi-algebraic proof systems, we *can* prove superpolynomial lower bounds, and at the same time, the corresponding algorithms capture interesting and ubiquitous families of algorithms. In particular, we will develop the proof system Sherali-Adams (SA) from an algorithmic point of view, and see that SA captures a large class of linear programming relaxations. We will see, through the theory of linear programming duality, that SA derivations (a syntactic object) are dual to points in the corresponding LP polytopes (a semantic object), thus linking the SA proof complexity of proving that a solution exists, to the algorithmic complexity of finding such a solution. In a similar manner we will develop the SoS system, and see that it captures semi-definite programming relaxations. Again, we will link SoS derivations to the points in the corresponding semi-definite cone, thus linking SoS complexity of proving the existence of a solution to the algorithmic complexity of finding such a solution.

In Chapter 5 we prove SoS and SA lower bounds for a particular family of 3XOR and instances. Using the above connection, this implies in a precise sense that *no* linear programming or semi-definite program based on low-degree SoS can approximate 3XOR or MaxSAT better than the trivial approximation. That is, we see an exact instance where a proof complexity lower bound implies lower bounds for a large class of approximation algorithms.

In the other direction, in Chapter 4 we will see how SoS *upper bounds* can lead to efficient algorithms. [3] The high level idea is as follows. Start with some optimization problem such

[3]We remark that there is a long history of related results in logic, showing strong links between proofs

as Independent Set. If we can manage to give a low degree SoS proof that for every graph G, there exists a 2-approximation, then by the degree automatizability of SoS, this implies an efficient algorithm for actually finding the solution. This idea is very powerful, and has been applied to many problems in machine learning. The general approach is to obtain low-degree SoS proofs of polynomial sample complexity bounds for the learning problem, and then by degree-automatizability of SoS, this yields an efficient learning algorithm.

As a toy example to illustrate this connection, suppose that you are given samples from an unknown Gaussian with mean μ and standard deviation one, and want to approximately recover the true mean from these samples. A necessary condition for succeeding in polynomial time are *sample complexity* bounds – that is, polynomially many samples must be enough to approximate the true mean information theoretically. Now suppose that we can formalize and prove this sample complexity bound with a low degree SoS proof. Then by degree automatizability of SoS, this automatically gives a polynomial-time algorithm for solving the learning problem! In Section 4.3, we discuss several instantiations of this approach where state-of-the-art learning algorithms are obtained for: dictionary learning, tensor decomposition, as well as learning mixtures of Gaussians.

and programs. In particular, in restricted systems of arithmetic, programs/algorithms can be extracted from proofs of existence.

Chapter 2

Sherali-Adams

2.1 Linear Programming

Linear programming describes a broad class of optimization problems in which both the objective function that we are trying to optimize, and the constraints which the solutions must satisfy are linear functions. Linear programs have the following canonical form, in which we are optimizing an objective function $c^\top x$ a set of linear constraints $\mathcal{P}$,

$$\begin{aligned} \mathcal{LP}(\mathcal{P}, c) \ &:= \ \min_{x \in \mathcal{P}} \ c^\top x, \\ \text{where } \mathcal{P} &= \{Ax \geqslant b, \quad x \geqslant 0\}, \end{aligned}$$

for $A \in \mathbb{R}^{m \times n}$ and $c \in \mathbb{R}^n$. Geometrically, this corresponds to optimizing over the convex polytope in $\mathbb{R}^n$ defined by the linear inequalities in $\mathcal{P}$. Throughout this monograph we will abuse notation and write $\mathcal{P}$ to refer to both the set of constraints, as well as the set of solutions to those constraints (i.e. the convex polytope), relying on the context to differentiate between the two interpretations. Linear programming was first shown to be solvable in polynomial time by Khachiyan [83] using the ellipsoid method[1], and it was later discovered to be complete for P. Since then, significant work has focused on designing fast and efficient algorithms for solving LPs efficiently in practice, resulting in linear programming being considered a standard tool in a wide variety of fields.

Not only does solving an LP compute a feasible solution, but it also generates a short proof of the optimality of that solution. This is a consequence of the duality theorem for linear programs, and will be the focus of the rest of this section. We will call $\mathcal{LP}(\mathcal{P}, c)$ the *primal* LP; the *dual* LP is defined as

$$\begin{aligned} \mathcal{LP}^D(\mathcal{P}, c) \ &:= \mathcal{LP}(\mathcal{P}^D, -b) = \max_{x \in \mathcal{P}^D} \ b^\top y, \\ \text{where } \mathcal{P}^D &= \{A^\top y \leqslant c, \quad y \geqslant 0\}, \end{aligned}$$

[1] We review the ellipsoid method in Section 3.1.1 for solving semi-definite programs.

Any solution to the dual LP is a lower bound on the minimum value that the primal can attain. Furthermore, the duality theorem states that if feasible solutions to the primal and dual both exist, then their optimal values coincide.

Theorem 2.1 (Linear Programming Duality)**.** *Let $\mathcal{P} = \{Ax \geqslant b, x \geqslant 0\}$ be a set of linear inequalities and $c \in \mathbb{R}^n$. Then, exactly one of the following cases holds*

1. *Neither $\mathcal{LP}(\mathcal{P}, c)$ nor $\mathcal{LP}^D(\mathcal{P}, c)$ has a feasible solution,*
2. *$\mathcal{LP}(\mathcal{P}, c)$ has solutions of arbitrarily small value, and $\mathcal{LP}^D(\mathcal{P}, c)$ is unsatisfiable.*
3. *$\mathcal{LP}(\mathcal{P}, c)$ is unsatisfiable, and $\mathcal{LP}^D(\mathcal{P}, c)$ has solutions of arbitrarily large value.*
4. *Both $\mathcal{LP}(\mathcal{P}, c)$ and $\mathcal{LP}^D(\mathcal{P}, c)$ have optimal solutions x^* and y^*, and furthermore*

$$c^\top x^* = b^\top y^*.$$

In later sections we will prove similar duality theorems between the proof system and the optimization views of Sherali-Adams and Sum-of-Squares. As a warm-up to this, we will reformulate linear programming as a proof system and illustrate how this duality theorem can be interpreted as showing a duality between the proof system and optimization views of linear programming. Let $\mathcal{P} = \{Ax \geqslant b, x \geqslant 0\}$ be a set of linear inequalities, and c_0 be some feasible solution. By linearity, any feasible solution to a set of linear inequalities must also be a solution to any non-negative linear combination of those inequalities. Therefore, if there exists a non-negative linear combination of the inequalities in $\mathcal{P} \cup \{c_0 \geqslant 0\}$ which equals $-1 \geqslant 0$ then this is a refutation of the claim that c_0 is a feasible solution to $\mathcal{P}$. In general, we define a linear programming refutation as a certificate of infeasibility of a set of linear inequalities as follows.

Definition 2.2 (Linear Programming Refutation)**.** Let $\mathcal{P}$ be a set of linear inequalities $\mathcal{P} = \{Ax \geqslant b, x \geqslant 0\}$. A linear programming refutation of $\mathcal{P}$ is a non-negative linear combination

$$\lambda^\top (Ax - b) + \mu^\top x = -1,$$

for $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^n$, with $\lambda, \mu \geqslant 0$.

Soundness and completeness of linear programming refutations follows from a well-known corollary of the duality theorem, known as Farkas' Lemma. In words, for any point lying outside of a given polytope, Farkas' Lemma guarantees the existence of a hyperplane separating that point from the polytope. We will give a slightly different, but equivalent version of Farkas' Lemma which will be particularly useful throughout this monograph.

Lemma 2.3 (Farkas' Lemma)**.** *Let $\mathcal{P}$ be a polytope described by a system of linear inequalities $\{Ax \geqslant b, x \geqslant 0\}$ for $A \in \mathbb{R}^{n \times m}$ and $b \in \mathbb{R}^n$. Then, $\mathcal{P}$ has no feasible solution over $\mathbb{R}^n$ if and only if there exists $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^n$, $\lambda, \mu \geqslant 0$ such that*

$$\lambda^\top (Ax - b) + \mu^\top x = -1,$$

for $\lambda \in \mathbb{R}^m$, $\mu \in \mathbb{R}^n$, with $\lambda, \mu \geqslant 0$.

We can also define a derivational version of linear programming.

Definition 2.4 (Linear Programming Derivation)**.** Let $\mathcal{LP}(\mathcal{P}, c)$ be a linear program over the set of constraints $\mathcal{P} = \{Ax \geqslant b, x \geqslant 0\}$. A linear programming derivation of the inequality $c^\top x \geqslant c_0$ is a non-negative vector $y^* \in \mathbb{R}^m$ such that $(y^*)^T A \leqslant c$ and $(y^*)^\top b = c_0$. This follows because $c^\top x \geqslant (y^*)^\top Ax \geqslant (y^*)^\top b = c_0$.

That is, a Linear Programming Derivation is simply a feasible solution to the dual LP.

The soundness and completeness for linear programming derivations follow from the duality theorem. It should be stressed that an LP attains a value c_0 if and only if there exists a linear programming derivation of c_0 from $\mathcal{P}$. Therefore, we have two equivalent views of linear programming: one from the perspective of optimization, and the other from the perspective of proof complexity. These dual views will be a central theme throughout this monograph.

2.1.1 Variants of Linear Programming and Relaxations

Because linear programming is in P, solving any NP-hard optimization via a polynomial-size LP is tantamount to proving $\mathsf{P} = \mathsf{NP}$. However, there has been significant success in designing LP-based algorithms that produce sufficient approximate solutions for certain NP-hard problems. One common technique for designing such approximate LPs is to first express the problem as an equivalent integer-linear program (ILP), which augments linear programming by allowing one to introduce integer-valued variables. The canonical form of an ILP is

$$\begin{aligned} \mathcal{ILP}(\mathcal{P}, c) \;&:=\; \min_{x \in \mathcal{P}} \; c^\top x \\ \text{where } \mathcal{P} &= \{Ax \geqslant b, \quad x_i \in D_i\}, \end{aligned}$$

where $D_i \subseteq \mathbb{Z}$. For example, a 0-1-linear program allows for constraints $x_i \in \{0, 1\}$, which corresponds to optimizing over the *integer hull* of $\mathcal{P}$ defined as

$$\mathsf{hull}_{\{0,1\}}(\mathcal{P}) \;:=\; \mathrm{conv}\left(\mathcal{P} \cap \{0,1\}^n\right),$$

where $\mathrm{conv}(S)$ is the convex hull of the set of points $S \subseteq \{0,1\}^n$.

It is straightforward to see that solving general ILPs, and even 0-1-LPs, is NP-complete. Therefore, one typically attempts to find a good approximation to the ILP by *relaxing* the constraints to those of an LP. This is generally done by replacing each discrete constraint $x_i \in D_i$ by the corresponding linear constraint $\min\{D_i\} \leqslant x_i \leqslant \max\{D_i\}$. For example, the 0-1-LP constraint $x_i \in \{0, 1\}$ is replaced by the linear inequality $0 \leqslant x_i \leqslant 1$. The resulting program is known as the *LP relaxation* of the ILP.

An alternative approach is to phrase the NP-hard problem as a polynomial optimization problem (POP), which augments linear-programs by allowing super-linear constraints. For example, constraining a variable x_i to take value in $\{0, 1\}$ can be done by introducing the quadratic constraint $x_i^2 - x_i = 0$. We can define a polynomial optimization problem as

$$\mathcal{POP}(\mathcal{P}, P) := \min_{x \in \mathcal{P}} \; P(x)$$

$$\text{where } \mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}.$$

where $P(x), P_1(x), \ldots, P_m(x) \in \mathbb{R}[x]$. Of course, optimizing over arbitrary polynomial constraints is NP-hard. Similar to the case of ILPs, one can hope to obtain a good approximate solution to a POP by relaxing the polynomial constraints. To do this, one typically introducing a placeholder variable y_I for every term $\prod_{i \in I} x_i$. The POP is then linearized by replacing every term in each of the original polynomials in the set $\mathcal{P}$ by their placeholder variables. These linearizations are often augmented with additional constraints to force the placeholder variables y_I to behave similarly to the terms that they represent. We will see examples of this style of relaxation when we discuss Sherali-Adams and Sum-of-Squares in the following sections.

2.2 Sherali-Adams

2.2.1 Sherali-Adams as Lifting Linear Programs

In the typical setting for Sherali-Adams we are interested in finding $\{0,1\}$-solutions that maximize some discrete optimization problem. Towards this, a natural strategy is to try to find a small LP that approximately describes the integer hull of the feasible solutions to the optimization problem. To design such an LP, a standard approach is to begin with an ILP or POP which describes the $\{0,1\}$-solutions exactly and then take its LP relaxation. The hope is that solving this relaxation will give us a good approximate solution. If this is the case, then some appropriate rounding scheme can be applied in order to obtain a $\{0,1\}$-solution that well-approximates the optimal. Unfortunately, there are cases where optimizing over the natural ILP or POP relaxation may result in extremely poor solutions. Consider for example the MaxSAT problem.

Definition 2.5 (MaxSAT)**.** Given a CNF formula $f = C_1(x) \wedge C_2(x) \wedge \ldots \wedge C_m(x)$, find an assignment to $x_1, \ldots, x_n$ that maximizes the number of satisfied clauses.

MaxSAT can be easily formulated as an ILP as follows: For each clause $C_i \in f$, where $C_i = \bigvee_{i \in I} x_i \vee \bigvee_{j \in J} \bar{x}_i$, we can write that clause as sum $\tilde{C}_i = \sum_{i \in I} x_i + \sum_{j \in J}(1 - x_j)$. The ILP becomes

$$\mathcal{ILP}\Big(\mathcal{P}, \sum_{i \in [m]} c_i\Big) \;:=\; \max_{(x,c) \in \mathcal{P}} \sum_{i \in [m]} c_i,$$
$$\text{where } \mathcal{P} = \{\tilde{C}_1 \geqslant c_1, \ldots, \tilde{C}_m \geqslant c_m,\;\; x_i, c_j \in \{0,1\}\}.$$

The LP relaxation of this ILP is obtained by replacing the constraints $x_i, c_j \in \{0,1\}$ by the linear constraints $0 \leqslant x_i \leqslant 1$ and $0 \leqslant c_i \leqslant 1$.

Example 2.6. Consider the instance of MaxSAT where $f = (x_1 \vee x_2 \vee \bar{x}_4) \wedge (x_1 \vee x_3) \vee (x_! \vee \bar{x}_2) \vee (\bar{x}_1)$. The ILP and corresponding LP relaxation for this instance are:

(ILP)

$$\begin{aligned} \max \quad & c_1 + c_2 + c_3 + c_4 \\ \text{s.t.} \quad & x_1 + x_2 + (1 - x_3) \geqslant c_1 \\ & x_1 + x_3 \geqslant c_2 \\ & x_1 + (1 - x_2) \geqslant c_3 \\ & (1 - x_1) \geqslant c_4 \\ & x_i, c_j \in \{0, 1\} \end{aligned}$$

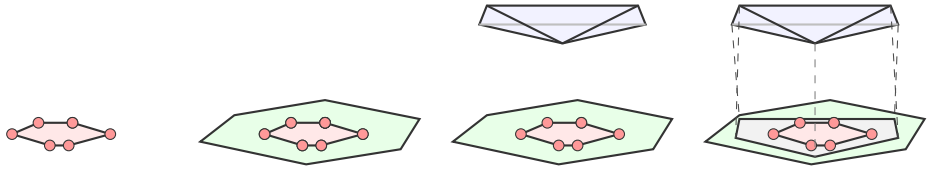
(LP Relaxation)

$$\begin{aligned} \max \quad & c_1 + c_2 + c_3 + c_4 \\ \text{s.t.} \quad & x_1 + x_2 + (1 - x_3) \geqslant c_1 \\ & x_1 + x_3 \geqslant c_2 \\ & x_1 + (1 - x_2) \geqslant c_3 \\ & (1 - x_1) \geqslant c_4 \\ & x_i, c_j \geqslant 0 \\ & x_i, c_j \leqslant 1 \end{aligned}$$

Unfortunately, maximizing the LP relaxation in the previous example results in a poor solution to the MaxSAT instance. Observe that, setting $x_i = 1/2$ for all $i \in [3]$ and $c_j = 1/2$ for all $j \in [4]$ satisfies all constraints, giving a value of 4 to the LP relaxation. However, the optimal solution to the original instance $f(x)$ is 3.

In situations where the LP relaxation is a poor approximation to the integer hull, the relaxation can be *tightened* by introducing additional constraints. This may be done either in an ad-hoc fashion, or in some systematic way. The standard systematic approaches for doing this are known as *lift-and-project* procedures, in which some family of polynomial constraints are added to the relaxation. The relaxation is then linearized by replacing every term $\prod_{i \in I} x_i$ in each of these polynomials with a new placeholder variable y_I for that term (as is standard for linearizing POPs), thus lifting the relaxation to a higher dimension (Figure 2.1c). Typically, the purpose of this additional family of constraints is to help the resulting LP retain the correlations between variables that are imposed by super-linear constraints. This is done in conjunction with exploiting the fact that $x_i^2 = x_i$ for $\{0, 1\}$-solutions.

This lifted LP relaxation can then be optimized using standard methods for solving linear programs, and the solutions to the lifted LP relaxation can be projected back to the original dimension in order to obtain solutions to the original problem (Figure 2.1d). The high-level idea is that, even if no small LP on n-variables exists that well-approximates the integer hull of the problem, there may still be some higher-dimensional polytope with much fewer facets whose projection is a good approximation to the integer hull of the problem.



(a) ILP (b) LP relaxation (c) Lift (d) Project

Figure 2.1: The generic form of a lift-and-project procedure. The circles represent the solutions to the ILP. The two-dimensional polygons represent the polytopes in the original variables, while the three-dimensional polyhedron represents the lift, whose two-dimensional *shadow* is a tightening of the LP relaxation.

A common theme among lift-and-project procedures is that they generate a hierarchy of relaxations known as the *levels* of the procedure, converging to the integer hull,

$$\mathcal{P} = \mathcal{P}^{(0)} \supseteq \mathcal{P}^{(1)} \supseteq \ldots \supseteq \mathcal{P}^{(\ell)} = \mathsf{hull}_{\{0,1\}}(\mathcal{P}),$$

where $\mathcal{P}^{(i)}$ is the i-th level of the hierarchy, with the higher levels corresponding to raising the relaxation to a higher dimension. Therefore, lift-and-project procedures are often referred to as LP and SDP *hierarchies*. These hierarchies differ in the family of constraints they introduce, as well as the type of convex program that the hierarchy produces.

Sherali-Adams [141] refers to the LP hierarchy whereby the original LP is lifted by introducing a family of non-negative juntas.

Definition 2.7 (Non-negative Junta)**.** A degree d non-negative junta is a polynomial of the form

$$J_{S,T}(x) \;:=\; \prod_{i \in S} x_i \prod_{j \in T} (1 - x_i),$$

for $S \cap T = \emptyset$ and $|S \cup T| \leqslant d$.

Before defining the Sherali-Adams (SA) hierarchy in full generality, we will sketch the process by constructing the level 2 SA relaxation for the MaxSAT instance from Example 2.6 in the following example. The level 2 SA relaxation augments the initial LP with (the linearizations of) all non-negative juntas, and products of initial constraints with non-negative juntas, of degree up to 2.

Example 2.8. Consider the MaxSAT instance from Example 2.6, whose constraints we will denote by $\mathcal{P}$. For ease of notation we will first rename the c-variables as follows: $c_1 \to x_4, c_2 \to x_5, c_3 \to x_6, c_4 \to x_7$. The level 2 SA relaxation of $\mathcal{P}$ first introduces all degree at most 2 polynomials from the family of non-negative juntas

$$x_i \geqslant 0, \qquad (1 - x_i) \geqslant 0, \qquad \forall i \in [7], \tag{2.1}$$
$$x_i x_j \geqslant 0, \qquad x_i(1 - x_j) \geqslant 0, \qquad \forall i \in [7], \tag{2.2}$$

As well, it introduces all products of constraints in $\mathcal{P}$ with non-negative juntas which have degree at most 2. For example, for the constraint $x_1 + (1 - x_2) \geqslant c_3$, we introduce

$$x_1 + (1 - x_2) \geqslant x_6, \tag{2.3}$$
$$(x_i)(x_1 + (1 - x_2) \geqslant x_6), \qquad \forall i \in [7] \tag{2.4}$$
$$(1 - x_i)(x_1 + (1 - x_2) \geqslant x_6), \qquad \forall i, j \in [7], i \neq j, \tag{2.5}$$

Next, we perform the lifting step, where the polynomial constraints are *linearized* by replacing each monomial $x_i x_j$ with a new placeholder variable $y_{\{i,j\}}$, and x_i with $y_{\{i\}}$. Here, $y_{\{i,j\}}$ is intended to represent $x_i x_j$, and $y_{\{i\}} = x_i$. The constraints (2.1) and (2.2)

become

$$\begin{aligned} y_{\{i\}} &\geqslant 0, & 1 - y_{\{i\}} &\geqslant 0, & &\forall i \in [7], \\ y_{\{i,j\}} &\geqslant 0, & y_{\{i\}} - y_{\{i,j\}} &\geqslant 0, & &\forall i, j \in [7], i \neq j, \end{aligned}$$

and the constraints (2.3) - (2.5) become

$$\begin{aligned} y_{\{1\}} + 1 - y_{\{2\}} &\geqslant y_{\{6\}}, & & \\ y_{\{1,i\}} + y_{\{i\}} - y_{\{2,i\}} &\geqslant y_{\{4,i\}}, & &\forall i \in [7], \\ y_{\{1\}} - y_{\{1,i\}} + 1 - y_{\{2\}} - y_{\{2\}} + y_{\{2,i\}} &\geqslant y_{\{6\}} - y_{\{6,i\}}, & &\forall i, j \in [7], i \neq j, \end{aligned}$$

Completing this procedure for all of the constraints in $\mathcal{P}$ gives the level 2 SA relaxation for this instance of MaxSAT.

The SA relaxation can be applied to POPs, as well as LPs. Because any LP is also a POP, we will define SA in this more general setting, where the aim is to minimize a polynomial $P(x)$ over a family of polynomial inequalities $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$. For this, it will be convenient to introduce some terminology.

Definition 2.9 (Coefficient Vector)**.** For any multilinear polynomial $P_i(x) \in \mathbb{R}[x]$, denote by $\vec{P_i}$ the coefficient vector of the representation of $P_i(x)$ as a sum of monomial. That is, the I-th entry $(\vec{P_i})_I$ is the coefficient of the monomial $\prod_{i \in I} x_i$ in $P_i(x)$.

Let $\deg(P_i)$ be the degree of the polynomial $P_i(x) \in \mathbb{R}[x]$ and for a set of polynomial inequalities $\mathcal{P}$, define the degree of $\mathcal{P}$ as the maximum degree of any polynomial in $\mathcal{P}$:

$$\deg(\mathcal{P}) \ := \max_{P_i(x) \geqslant 0 \in \mathcal{P}} \deg(P_i).$$

With these definitions in mind, we can now describe the SA relaxation of any polynomial optimization problem as follows.

Definition 2.10 (Sherali-Adams Relaxation)**.** Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be a set of polynomial inequalities. For $d \geqslant \deg(\mathcal{P})$, the d-th level SA relaxation $\mathsf{SA}_d(\mathcal{P})$ of the set of polynomial inequalities $\mathcal{P}$ is obtained as follows:

1. **Extend:** For every constraint $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$ and every $S, T \subseteq [n]$ with $|S \cup T| \leqslant d$ and $S \cap T = \emptyset$, such that $\deg(J_{S,T}) + \deg(P_i) \leqslant d$, introduce a new constraint.[2]

$$P_i(x) \prod_{i \in S} x_i \prod_{j \in T} (1 - x_j) \geqslant 0. \tag{2.6}$$

[2]Definitions of SA vary in the literature depending on whether they measure the degree of $P_i(x)$ or not. That is, whether they limit $j_{S,T}(x) \cdot P_i(x)$ to be degree at most d or degree at most $\deg(P_i)$, for $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$. When SA is discussed from the perspective of proof complexity it is most common to count both the degree of the junta and the degree of the polynomial. We compare the consequences of these two definitions further in the remark at the end of Section 2.2.2.3.

2. **Linearize:** multilinearize each of the constraints introduced in the previous step by replacing x_i^c with x_i for every $c > 1$. For every monomial $\prod_{i \in S} x_i$ occurring in (2.6), introduce a new variable y_S, and replace each occurrence of that monomial by y_S. Finally, add the constraint $y_\emptyset = 1$.

The resulting relaxation, which we denote by $\mathsf{SA}_d(\mathcal{P})$, consists of the following set of constraints

$$y_\emptyset = 1, \tag{2.7}$$

$$\sum_{T' \subseteq T} (-1)^{|T'|} y_{S \cup T'} \geqslant 0, \qquad \forall\ S \cap T = \emptyset, |S \cup T| \leqslant d \tag{2.8}$$

$$\sum_{T' \subseteq T} (-1)^{|T'|} \Bigg(\sum_{|I| \leqslant \deg(P_i)} (\vec{P_i})_I \; y_{S \cup I \cup T'} \Bigg) \geqslant 0, \qquad \begin{smallmatrix} \forall\ S \cap T = \emptyset,\ |S \cup T| \leqslant d - \deg(P_i), \\ \forall\ P_i(x) \geqslant 0 \in \mathcal{P} \end{smallmatrix}. \tag{2.9}$$

These formulas follow by inclusion/exclusion on the variables in T.

The variable $y_\emptyset$ is the placeholder for the monomial $\prod_{i \in \emptyset} x_i$, the constant 1 term. Therefore, we include Constraint (2.7) to ensure that the solutions that we obtain from optimizing over the relaxation are correctly normalized; i.e. that $y_\emptyset$ indeed corresponds to the constant 1.

Example 2.11. Consider linearizing a non-negative junta $J_{S,T}(x) \geqslant 0$ where $S = \{1, 2\}$ and $T = \{3, 4, 5\}$; that is, $J_{S,T}(x) = x_1 x_2 (1 - x_3)(1 - x_4)(1 - x_5)$. When expressed as a sum of monomials, $J_{S,T}(x)$ becomes

$$x_1 x_2 - x_1 x_2 x_3 - x_1 x_2 x_4 - x_1 x_2 x_4 + x_1 x_2 x_3 x_4 + x_1 x_2 x_3 x_5 + x_1 x_2 x_4 x_5 - x_1 x_2 x_3 x_4 x_5 \geqslant 0.$$

This is linearized by introducing the placeholder variables for each term, producing

$$y_{\{1,2\}} - y_{\{1,2,3\}} - y_{\{1,2,4\}} - y_{\{1,2,5\}} + y_{\{1,2,3,4\}} + y_{\{1,2,3,5\}} + y_{\{1,2,4,5\}} - y_{\{1,2,3,4,5\}} \geqslant 0.$$

Example 2.12. Let $\mathcal{LP}(\mathcal{P}, c)$ be an LP over a set of linear constraints $\mathcal{P} = \{a_1^\top x \geqslant b_1, \ldots, a_m^\top x \geqslant b_m\}$. The d-th level SA relaxation $\mathsf{SA}_d(\mathcal{P})$ consists of th following set of constraints:

$$y_\emptyset = 1,$$

$$\sum_{T' \subseteq T} (-1)^{|T'|} y_{S \cup T'} \geqslant 0, \qquad \forall\ S \cup T = \emptyset, |S \cup T| \leqslant d$$

$$\sum_{T' \subseteq T} (-1)^{|T'|} \Bigg(\sum_{i=1}^{n} a_i^\top \cdot y_{S \cup T' \cup \{i\}} - b_i \cdot y_{S \cup T'} \Bigg) \geqslant 0, \qquad \begin{smallmatrix} \forall\ S \cap T = \emptyset,\ |S \cup T| \leqslant d - 1, \\ \forall\ a_i^\top x - b_i \geqslant 0 \in \mathcal{P} \end{smallmatrix}$$

Solving the Sherali-Adams Relaxation After applying the SA relaxation, we want to obtain an (approximate) solution to the original problem of minimizing a polynomial $P(x)$ over $\mathcal{P}$. This can be done by linearizing the objective function $P(x)$ to $\sum_I (\vec{P})_I y_I$, and optimizing the resulting polytope $\mathsf{SA}_d(\mathcal{P})$ over the linear objective function. That is, by solving

$$\mathcal{LP}(\mathsf{SA}_d(\mathcal{P}), P(y)) \;:=\; \min_{y \in \mathsf{SA}_d(\mathcal{P})} \left(\sum_{|I| \leqslant \deg(P_i)} (\vec{P})_I y_I \right).$$

In the case of where $P(x)$ is a linear objective function $c^\top x$, this corresponds to solving $\min_{y \in \mathsf{SA}_d(\mathcal{P})} (\sum_{i \in [n]} c_i y_{\{i\}})$.

The SA relaxation $\mathsf{SA}_d(\mathcal{P})$ contains $n^{O(d)}$ variables and $m \cdot n^{O(d)}$ constraints. Therefore, $\mathcal{LP}(\mathsf{SA}_d(\mathcal{P}), P(y))$ can be solved in time $(m \cdot n)^{O(d)}$ by standard linear programming algorithms. The result of solving $\mathcal{LP}(\mathsf{SA}_d(\mathcal{P}), P(y))$ is a solution α in the lifted space over $n^{O(d)}$ variables. A solution to the original optimization problem is obtained by projecting back to the original variables over $\mathbb{R}^n$. This is achieved by taking the orthogonal projection

$$\mathrm{proj}_{[n]}(\mathcal{P}) \;:=\; \{\alpha \restriction_{\{y_{\{1\}}, \ldots, y_{\{n\}}\}} : \alpha \in \mathcal{P}\}$$

to the variables $\{y_{\{1\}}, \ldots, y_{\{n\}}\}$ corresponding to the original variables x.

Observe that the solutions produced by the SA relaxation are at least as good of an approximation as those produced by the standard LP relaxation. Indeed, the constraints of the SA relaxation are a super-set those provided in the LP relaxation. Furthermore, the solutions obtained from the d-th level SA relaxation have at least as good of an approximation guarantee as those obtained from the $(d-1)$-st level. That is, $\mathsf{SA}_d(\mathcal{P})$ corresponds to a *tightening* of $\mathsf{SA}_{d-1}(\mathcal{P})$: it preserves all integral solutions while removing some of the fractional solutions.

Lemma 2.13. *Let $\mathcal{P}$ be any set of polynomial inequalities. Then, for any $d \geqslant 0$, the following containments hold:*

1. $\mathrm{proj}_{[n]}(\mathsf{SA}_d(\mathcal{P})) \supseteq \mathrm{proj}_{[n]}(\mathsf{SA}_{d+1}(\mathcal{P}))$,
2. $\mathrm{proj}_{[n]}(\mathsf{SA}_d(\mathcal{P})) \supseteq \mathsf{hull}_{\{0,1\}}(\mathcal{P})$.

Proof. (1) is immediate from the fact that the constraints of $\mathsf{SA}_d(\mathcal{P})$ are a subset of the constraints of $\mathsf{SA}_{d+1}(\mathcal{P})$. To prove (2), we claim that any $\{0,1\}$-solution to $\mathcal{P}$ can be extended to a $\{0,1\}$-solution to $\mathsf{SA}_d(\mathcal{P})$. That is, $\mathsf{SA}_d(\mathcal{P})$ preserves the integer hull of $\mathcal{P}$. Let $\alpha \in \mathcal{P}$ such that $\alpha \in \{0,1\}^n$, we extend α to an assignment β to the variables of $\mathsf{SA}_d(\mathcal{P})$ as follows. For every $S \subseteq [n]$ such that y_S is a variable of $\mathsf{SA}_d(\mathcal{P})$, we define

$$\beta_S = \prod_{i \in S} \alpha_i.$$

Note that because $\alpha_i \in \{0,1\}$, $\beta_S \in \{0,1\}$. Now, consider applying β to one of the constraints of $\mathsf{SA}_d(\mathcal{P})$, corresponding to the linearization of $J_{S,T}(x) \cdot P(x)$, for some non-negative $(d -$

$\deg(P))$-junta $J_{S,T}(x)$ and $P(x) \geqslant 0 \in \mathcal{P}$. Because β is a $\{0,1\}$-assignment consistent with α, it behaves identically to α on $J_{S,T}(x) \cdot P(x)$. Therefore,

$$\sum_{T' \subseteq T} (-1)^{|T'|} \left(\sum_{|I| \leqslant \deg(P_i)} (\vec{P}_i)_I \beta_{S \cup I \cup T'} \right) = \begin{cases} P(\alpha) \text{ if } J_{S,T}(\beta) = 1 \\ 0 \text{ otherwise} , \end{cases}$$

where the equality follows because the outcome of any $\{0,1\}$-assignment on a non-negative junta is Boolean. Finally, because α satisfies the constraints of $\mathcal{P}$, we have that $P(\alpha) \geqslant 0$, and so $\beta \in \mathsf{SA}_d(\mathcal{P})$.

□

We will see in Section 2.2.2.3 that the degree $O(n)$ SA relaxation suffices to derive the integer hull of the original polytope. Together with Lemma 2.13, this implies that the SA relaxations form a *hierarchy* of polytopes parameterized by their level, converging to the integer hull:

$$\text{proj}_{[n]} \left(\mathsf{SA}_1(\mathcal{P}) \right) \supseteq \text{proj}_{[n]} \left(\mathsf{SA}_2(\mathcal{P}) \right) \supseteq \ldots \supseteq \text{proj}_{[n]} \left(\mathsf{SA}_{O(n)}(\mathcal{P}) \right) = \mathsf{hull}_{\{0,1\}}(\mathcal{P})$$

This is known as the SA hierarchy.

2.2.2 Sherali-Adams as Locally Consistent Distributions

The SA hierarchy corresponds a sequence of ever-tightening polytopes converging to the integer hull. For many problems, a high level of the SA hierarchy is necessary in order to converge to the integral hull. High level SA relaxations contain a prohibitively large number of constraints, and so optimizing over them is generally infeasible. Therefore, we want to understand how well of how well of an approximation the d-th level SA relaxation is to the integer hull.

In order to certify the approximation ratio provided by the d-th level, it will be useful to understand the points occurring within the d-th level polytope $\mathsf{SA}_d(\mathcal{P})$ (for some set of polynomial inequalities $\mathcal{P}$) and how they change as the level increases. A very elegant constructive characterization of the points within the SA relaxation can be obtained by taking a distributional view. As we will see, each point in the polytope of the SA relaxation can be viewed as a local expectation function, a function which behaves like a true expectation over some probability distribution when applied to polynomials of degree at most d. These local expectation functions, known as *pseudo-expectations*, are in one-to-one correspondence with points in the SA relaxation, and the necessary and sufficient conditions of pseudo-expectations will gives straightforward method to construct points which exist within the relaxation $\mathsf{SA}_d(\mathcal{P})$.

It is natural to wonder whether there is a distribution-like object over which these pseudo-expectations are being taken. After developing pseudo-expectations in Section 2.2.2.1, we introduce the notion of a *pseudo-distributions* in Section 2.2.2.2 and show that these are the distribution-analogue of pseudo-expectations. Finally, Section 2.2.2.3 we will characterize the points that survive between levels of the SA hierarchy by characterizing the points in the d-th level of the SA relaxation in terms of the points in the $(d-1)$-st level.

2.2.2.1 Pseudo-Expectations

Recall that each of the variables y_S of the SA relaxation $\mathsf{SA}_d(\mathcal{P})$ corresponds to a unique mononial $\prod_{i\in S} x_i$ in the original variables. Rather than working with these placeholder y-variables, we could instead work in the original variables x, treating each $\alpha \in \mathsf{SA}_d(\mathcal{P})$ as a map that assigns to each multilinear monomial $\prod_{i\in S} x_i$ the value α_S that α assigns to y_S.

Definition 2.14 (multilinearizing Map)**.** Denote by $\mathbb{R}[x] \setminus \{(x_i^2 - x_i)\}_{i\in[n]}$ the quotient ring of $\mathbb{R}[x]$ modulo the ideal $\{(x_i^2 - x_i)\}$. A multilinearizing map is a *linear* function $f : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$. That is, f is a function on $\mathbb{R}[x]$ associating

$$f\Big(\sum_{j\in J}\prod_{i\in I_j} x_i^{c_{i,j}}\Big) = \sum_{j\in J} f\Big(\prod_{i\in I_j} x_i\Big),$$

for all $I_j \subseteq [n]$ and $c_{i,j} \in \mathbb{N}$.

For each $\alpha \in \mathsf{SA}_d(\mathcal{P})$ we can define its corresponding multilinearizing map as

$$\tilde{\mathbb{E}}\Big[\prod_{i\in S} x_i\Big] \;:=\; \alpha_S$$

for every $S \subseteq [n]$ with $|S| \leqslant d$. We can then extend this linearly to all degree at most d polynomials in the natural way. Note that α is an assignment to the monomials, rather than the underlying variables. It treats each monomial independently regardless of whether the monomials share a subset of their underlying variables. As we will see, consistency between these related monomials is enforced by the constraints of the SA relaxation.

The constraints of the SA relaxation enforce that α, and therefore $\tilde{\mathbb{E}}$, only assigns non-negative values to non-negative juntas, as well as the product of juntas with an inequality in $\mathcal{P}$. These constraints have a natural distributional interpretation: we think of each monomial $\prod_{i\in S} x_i$ as the event, denoted by $1_{S,\emptyset}$, that $x_i = 1$ for all $i \in S$. Furthermore, we interpret the non-negative junta $J_{S,T} = \prod_{i\in S} x_i \prod_{j\in T}(1 - x_j)$ as the event $1_{S,T}$ that $x_i = 1$ for $i \in S$ and $x_j = 0$ for $j \in T$. We can view the function $\tilde{\mathbb{E}}$ applied to $J_{S,T}$ as an expectation (taken over some underlying distribution) that the event $1_{S,T}$ occurs. Because the d-th level SA relaxation can only enforce that products of at most d variables are non-negative, we only require that $\tilde{\mathbb{E}}$ behaves like a expectation on polynomials of degree at most d. Such functions $\tilde{\mathbb{E}}$ are known as *pseudo-expectations*. We define a pseudo-expectation abstractly as follows.

Definition 2.15 (Pseudo-Expectation for $\mathcal{P}$)**.** Let $\mathcal{P}$ be a set of polynomial inequalities. A multilinearizing map $\tilde{\mathbb{E}} : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ is a degree d pseudo-expectation for $\mathcal{P}$ if the following hold:

1. $\tilde{\mathbb{E}}[1] = 1$,
2. $\tilde{\mathbb{E}}[J_{S,T}(x)] \geqslant 0$ for every non-negative junta $J_{S,T}(x)$ with $\deg(J_{S,T}) \leqslant d$,
3. $\tilde{\mathbb{E}}[J_{S,T}(x)P(x)] \geqslant 0$ for every $P(x) \geqslant 0 \in \mathcal{P}$ and every non-negative junta $J_{S,T}(x)$ with $\deg(P) + \deg(J_{S,T}) \leqslant d$.

We will denote by $\mathcal{E}_d(\mathcal{P})$ the set of all degree d pseudo-expectations for $\mathcal{P}$.

Example 2.16. Consider the polynomial $P(x) = -x_1x_2x_4 + x_7 - 3x_2x_1$, and let $\alpha \in \mathsf{SA}_3(\{P(x) \geqslant 0\})$, then α defines a degree 3 pseudo-expectation $\tilde{\mathbb{E}}_\alpha$ which assigns to f the value

$$\tilde{\mathbb{E}}_\alpha [P(x)] = -\alpha_{\{1,2,4\}} + \alpha_{\{7\}} - 3\alpha_{\{1,8\}}.$$

Theorem 2.17. *Let $\mathcal{P}$ be a set of polynomial constraints, and for any $\alpha \in \mathbb{R}^{\binom{n}{\leqslant d}}$ define the multilinearizing map $\tilde{\mathbb{E}}_\alpha : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ as $\tilde{\mathbb{E}}_\alpha\big[\prod_{i\in S} x_i\big] := \alpha_S$, for every $S \subseteq [n]$, $\tilde{\mathbb{E}}_\alpha[1] = 1$. Extend $\tilde{\mathbb{E}}_\alpha$ linearly to all degree d polynomials. Then, $\alpha \in \mathsf{SA}_d(\mathcal{P})$ if and only if $\tilde{\mathbb{E}}_\alpha \in \mathcal{E}_d(\mathcal{P})$.*

Proof. Let $\alpha \in \mathsf{SA}_d(\mathcal{P})$ and let $\tilde{\mathbb{E}}_\alpha : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ be defined as in the statement of the theorem. That $\tilde{\mathbb{E}}_\alpha$ satisfies property (1) of Definition 2.17 follows immediately. To see that $\tilde{\mathbb{E}}_\alpha$ satisfies (2) and (3), let $J_{S,T}(x)$ be a non-negative junta and $P(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$ such that $\deg(J_{S,T}) + \deg(P_i) \leqslant d$, and denote by $L(y)$ its multilinearization (in the sense of the linearization step in Definition 2.10). Then,

$$\tilde{\mathbb{E}}_\alpha\big[P(x) \cdot J_{S,T}(x)\big] = L(\alpha) \geqslant 0,$$

where the final inequality follows because $L(y)$ is a constraint of $\mathsf{SA}_d(\mathcal{P})$.

For the other direction, consider some $\tilde{\mathbb{E}} \in \mathcal{E}_d(\mathcal{P})$. Define an assignment α to the variables of $\mathsf{SA}_d(\mathcal{P})$ as follows: for every $S \subseteq [n]$ such that y_S is a variable of $\mathsf{SA}_d(\mathcal{P})$, let

$$\alpha_S \;:=\; \tilde{\mathbb{E}}\left[\prod_{i\in S} x_i\right].$$

That $\alpha \in \mathsf{SA}_d(\mathcal{P})$ follows immediately from the definition of a pseudo-expectation. Again, let $L(y)$ be the multilinearization of the constraint $J_{S,T}(x) \cdot P(x) \geqslant 0 \in \mathsf{SA}_d(\mathcal{P})$ where $P(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$ and $\deg(P_i) + \deg(J_{S,T}) \leqslant d$. Then,

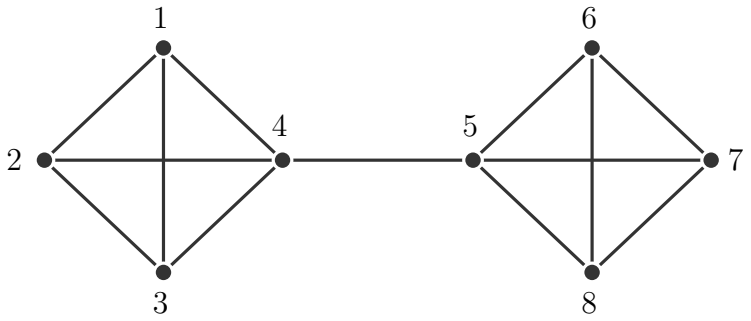
$$L(\alpha) = \tilde{\mathbb{E}}\,[J_{S,T}(x)P(x)] \geqslant 0.$$

□

We end this section by showing how to construct a pseudo-expectation for an instance of the Max Independent Set problem, and how to use this pseudo-expectation to certify that the Sherali-Adams relaxation has yet to converge to the integer hull. Recall that Max Independent Set problem is defined as follows.

Definition 2.18 (Max Independent Set)**.** Given a graph $G = (V, E)$, find the largest subset $I \subseteq V$ such that for all $u, v \in I$, $(u, v) \notin E$.

Example 2.19 (Max Independent Set)**.** Consider an instance of the Max Independent Set problem on the following graph $G = (V, E)$:



We can express this problem as an ILP with constraints $x_i \in \{0, 1\}$, and $x_i + x_j \leqslant 1$ for every $(i, j) \in E$. To obtain a solution we can solve the associated LP relaxation:

$$\max_{x \in \mathcal{P}} \sum_{i \in [8]} x_i$$

$$\text{where } \mathcal{P} = \left\{ \begin{array}{cccc} x_1 + x_2 \leqslant 1, & x_1 + x_3 \leqslant 1, & x_1 + x_4 \leqslant 1, & x_2 + x_3 \leqslant 1, \\ x_2 + x_4 \leqslant 1, & x_3 + x_4 \leqslant 1, & x_4 + x_5 \leqslant 1, & x_5 + x_6 \leqslant 1, \\ x_5 + x_7 \leqslant 1, & x_5 + x_8 \leqslant 1, & x_6 + x_7 \leqslant 1, & x_6 + x_8 \leqslant 1, \\ x_7 + x_8 \leqslant 1, & 0 \leqslant x_1 \leqslant 1, & 0 \leqslant x_2 \leqslant 1, & 0 \leqslant x_3 \leqslant 1, \\ 0 \leqslant x_4 \leqslant 1, & 0 \leqslant x_5 \leqslant 1, & 0 \leqslant x_6 \leqslant 1, & 0 \leqslant x_7 \leqslant 1, \\ 0 \leqslant x_8 \leqslant 1 & & & \end{array} \right\}$$

Because every vertex participates in one of two 4-cliques, the largest independent set has size 2. On the other hand, this LP relaxation returns a value of at least 4, because $x_i = 1/2$ for all $i \in [8]$ is a feasible solution.

Consider taking the level 2 SA relaxation $\mathsf{SA}_2(\mathcal{P})$, given by the following constraints:

$$0 \leqslant y_{\{i\}} \leqslant 1 \qquad \forall i \in [8] \tag{2.10}$$

$$0 \leqslant y_{\{i,k\}} \leqslant y_{\{k\}} \qquad \forall i, k \in [8] \tag{2.11}$$

$$0 \leqslant y_{\{i\}} - y_{\{i,k\}} \leqslant 1 - y_{\{k\}} \qquad \forall i, k \in [8] \tag{2.12}$$

$$y_{\{i\}} + y_{\{j\}} \leqslant 1 \qquad \forall (i, j) \in E \tag{2.13}$$

$$y_{\{i,k\}} + y_{\{j,k\}} \leqslant y_{\{k\}} \qquad \forall (i, j) \in E, k \in [8] \tag{2.14}$$

$$y_{\{i\}} - y_{\{i,k\}} + y_{\{j\}} - y_{\{j,k\}} \leqslant 1 - y_{\{k\}} \qquad \forall (j, k) \in E, k \in [8] \tag{2.15}$$

First, observe that the level 2 SA relaxation is a better approximation to the ILP: it does not permit the all-1/2 point. To see this, suppose that $y_{\{i\}} = 1/2$ for every $i \in [8]$. Observe that constraint (2.14) with $k = i$ is equivalent to $y_{\{i,j\}} \leqslant 0$ for every edge

$(i,j) \in E$. When combined with (2.11), this forces $y_{\{i,j\}} = 0$ for every $(i,j) \in E$. Taking constraint (2.15) on vertices $1, 2, 3$, we have

$$y_{\{1\}} - y_{\{1,3\}} + y_{\{2\}} - y_{\{2,3\}} \leqslant 1 - y_{\{3\}}.$$

Because the edges $(1,3), (2,3) \in E$, we are forced to set $y_{\{1,3\}} = y_{\{2,3\}} = 0$, and so the above inequality becomes

$$y_{\{1\}} - y_{\{1,3\}} + y_{\{2\}} - y_{\{2,3\}} \leqslant 1 - y_{\{3\}} \equiv 1/2 + 1/2 \leqslant 1 - 1/2,$$

a contradiction. Therefore, the $(1/2)^n$ point does not appear in the level 2 SA relaxation. By convexity, we can conclude that the solution $\mathcal{LP}(\mathsf{SA}_2(\mathcal{P}), \sum_{i\in[8]} x_i) < \mathcal{LP}(\mathcal{P}, \sum_{i\in[8]} x_i)$, certifying that the SA relaxation is an improvement on the LP relaxation.

Because of this, one might wonder if, for this instance of Max Independent Set, the second level of the SA hierarchy is enough to converge to the integer hull. To show that this is not the case, we will construct a degree 2 pseudo-expectation $\tilde{\mathbb{E}}$ witnessing that the solution $x_i = 1/3$ for all $i \in [8]$ belongs to $\mathsf{SA}_2(\mathcal{P})$. To construct such a pseudo-expectation we must assign values to all monomials $x_i x_j$ for $i \neq j \in [8]$ such that when $\tilde{\mathbb{E}}[x_i] = 1/3$ the following constraints are satisfied:

$$1 \geqslant \tilde{\mathbb{E}}[x_i] \geqslant 0 \qquad \forall i \in [8] \tag{2.16}$$
$$\tilde{\mathbb{E}}[x_i x_j] \geqslant 0 \qquad \forall i,j \in [8] \tag{2.17}$$
$$\tilde{\mathbb{E}}[x_k(1 - x_1)] \geqslant 0 \qquad \forall i,k \in [8] \tag{2.18}$$
$$\tilde{\mathbb{E}}[(1 - x_k)(1 - x_i)] \geqslant 0 \qquad \forall i,k \in [8] \tag{2.19}$$
$$\tilde{\mathbb{E}}[1 - x_i - x_j] \geqslant 0 \qquad \forall (i,j) \in E \tag{2.20}$$
$$\tilde{\mathbb{E}}[x_k(1 - x_i - x_j)] \geqslant 0 \qquad \forall (i,j) \in E, k \in [8] \tag{2.21}$$
$$\tilde{\mathbb{E}}[(1 - x_k)(1 - x_i - x_j)] \geqslant 0 \qquad \forall (j,k) \in E, k \in [8] \tag{2.22}$$

Note that (2.16) and (2.20) are already satisfied by our setting $\tilde{\mathbb{E}}[x_i] = 1/3$. Again observe that (2.21) with $k = i$, together with (2.17) forces us to set $\tilde{\mathbb{E}}[x_i x_j] = 0$. For the non-edges $(i,j) \notin E$ we will set $\tilde{\mathbb{E}}[x_i x_j] = 1/6$. Under this setting $\tilde{\mathbb{E}}[x_i x_k + x_j x_k] \leqslant \tilde{\mathbb{E}}[x_k]$, and therefore (2.19) and (2.21) are satisfied. Furthermore, because $\tilde{\mathbb{E}}[x_{\{i\}} + x_{\{j\}}] < 1$, this assignment satisfies constraints (2.18), (2.19), and (2.21). Therefore, $\tilde{\mathbb{E}}$ is a degree 2 pseudo-expectation witnessing that the solution $(1/3)^{[8]}$ is valid for $\mathsf{SA}_2(\mathcal{P})$, and that the value produced by $\mathcal{SDP}(\mathsf{SA}_2(\mathcal{P}), \sum_{i\in[8]} x_i)$ is at least $8/3$. On the other hand, the optimal value of this instance of Max Independent Set is 2. Together this shows that $\mathsf{SA}_2(\mathcal{P})$ has yet to converge to the integer hull, and in particular

$$\mathcal{LP}\Big(\mathsf{hull}_{\{0,1\}}(\mathcal{P}), \sum_{i\in[8]} x_i\Big) < \mathcal{LP}\Big(\mathsf{SA}_2(\mathcal{P}), \sum_{i\in[8]} x_i\Big) < \mathcal{LP}\Big(\mathcal{P}, \sum_{i\in[8]} x_i\Big).$$

2.2.2.2 Pseudo-Distributions

The term pseudo-expectation comes from the fact that $\tilde{\mathbb{E}}$ acts like an expectation on the set of polynomials of degree up to d. It is natural to ask how the corresponding distribution-like object, over which this expectation is being taken, behaves. For this, we will take a distributional view of the SA relaxation. We will begin by focusing on the points within $\mathsf{hull}_{\{0,1\}}(\mathcal{P})$. Because $\mathsf{hull}_{\{0,1\}}(\mathcal{P})$ is convex, any point $\alpha \in \mathsf{hull}_{\{0,1\}}(\mathcal{P})$ can be written as a convex combination of the $\{0,1\}$-points in $\mathsf{hull}_{\{0,1\}}(\mathcal{P})$

$$\alpha = \sum_{\beta \in \{0,1\}^n} \lambda_\beta \cdot \beta,$$

such that $\lambda_\beta = 0$ for all $\beta \notin \mathsf{hull}_{\{0,1\}}(\mathcal{P})$. Because this is a convex combination, $\lambda_\beta \geqslant 0$ and $\sum_{\beta \in \{0,1\}^n} \lambda_\beta = 1$. Thus we can view the coefficients λ_β in the convex combination as the weights in a probability distribution over $\{0,1\}$-solutions. That is, we can associate with α a probability distribution $\mu^{(\alpha)} : \{0,1\}^n \to \mathbb{R}^{\geqslant 0}$ given by

$$\mu^{(\alpha)}(\beta) = \lambda_\beta \text{ for all } \beta \in \{0,1\}^n. \tag{2.23}$$

Furthermore, the points with non-zero weight in this distribution satisfy $\mathcal{P}$. Therefore, every point in $\mathsf{hull}_{\{0,1\}}(\mathcal{P})$ defines a distribution over solutions in $\{0,1\}^n$ that satisfy $\mathcal{P}$.

The constraints of the SA relaxation can be viewed as attempting to verify that each point in $\mathsf{SA}_d(\mathcal{P})$ defines a probability distribution over $\{0,1\}$-assignments that satisfy $\mathcal{P}$. Of course, the points in $\mathsf{SA}_d(\mathcal{P}) \setminus \mathsf{hull}_{\{0,1\}}(\mathcal{P})$ cannot be expressed as a probability distribution over points in $\{0,1\}^n$ that satisfy the constraints of $\mathcal{P}$, because by definition no such convex combination exists. As we will see, the constraints enforced by low levels of the SA relaxation are unable to fully verify that the points in $\mathsf{SA}_d(\mathcal{P})$ define such a distribution, they are only able to confirm that each point defines an object that behaves *locally* like a distribution over $\{0,1\}$-solutions by observing its low-degree marginal distributions.

For any $\alpha \in \mathsf{SA}_d(\mathcal{P})$, we will think of α_S for $|S| \leqslant d$ as the probability of the event $1_{S,\emptyset}$, that $x_i = 1$ for all $i \in S$, over some underlying distribution $\mu^{(\alpha)} : \{0,1\}^n \to [0,1]$. We will denote this probability by $\mathbb{P}_{\mu^{(\alpha)}}[1_{S,\emptyset}]$. This can be written equivalently as

$$\mathbb{P}_{\mu^{(\alpha)}}[1_{S,\emptyset}] = \sum_{\beta \in \{0,1\}^n : \beta_i = 1 \ \forall i \in S} \mathbb{P}_{\mu^{(\alpha)}}[\beta] = \mu_S^{(\alpha)}[1_{S,\emptyset}].$$

That is, we are marginalizing to the variables S. If we denote by $\mu_S^{(\alpha)}$ the marginal distribution of $\mu^{(\alpha)}$ to the variables S, then we can rewrite this as

$$\sum_{\beta \in \{0,1\}^n : \beta_i = 1 \ \forall i \in S} \mathbb{P}_{\mu^{(\alpha)}}[\beta] = \mu_S^{(\alpha)}[1_{S,\emptyset}].$$

Because α assigns a value α_T for every $T \subseteq S$, the marginal distribution $\mu_S^{(\alpha)}$ is fully described by

$$\mu_S^{(\alpha)}[1_{T,K}] = \left(\sum_{K' \subseteq K} (-1)^{|J|} \alpha_{T \cup K'} \right),$$

for every $T \cup K = S$. The constraints of the SA relaxation, which are linearizations of non-negative juntas $J_{T,K}(x) \geqslant 0$, can be interpreted as enforcing that the marginal distributions of $\mu^{(\alpha)}$ are true, consistent probability distributions. In particular, the linearizations of $J_{T,K}(x) \geqslant 0$ for all $K \cup T = S$ ensure that the marginal distribution $\mu_S^{(\alpha)}$ is a probability distribution over $\{0,1\}^S$ (i.e. Boolean assignments to the variables $\{x_i : i \in S\}$). To verify this, we only need to check that the following properties of a distribution are satisfied: (i) $\mu_S^{(\alpha)}[1_{T,K}] \geqslant 0$ for every $T \cup K = S$, and (ii) $\sum_{T \cup K = S} \mu_S^{(\alpha)}[1_{T,K}]$. First, (i) follows immediately from the fact that SA enforces that the linearizations of $J_{T,K}(x)$ are non-negative $K \cup T = S$, provided $|S| \leqslant d$. For (ii), observe that

$$\sum_{T \cup K = S} \mu_S^{(\alpha)}(1_{T,K}) = \sum_{T \cup K = S} \left(\sum_{K' \subseteq K} (-1)^{|J|} \alpha_{T \cup K'} \right) = 1.$$

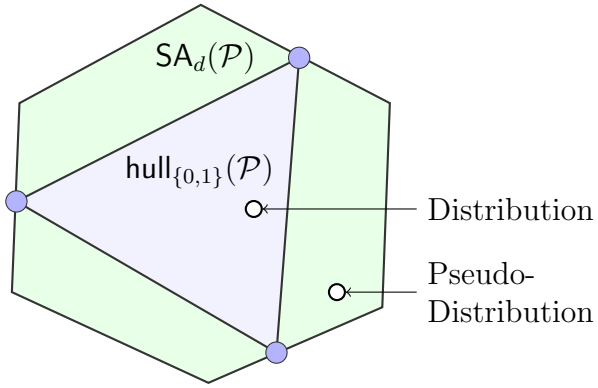


Figure 2.2: The points within $\mathsf{hull}_{\{0,1\}}(\mathsf{SA}_d(\mathcal{P}))$ (blue) form true distributions over solutions to $\mathcal{P}$, while the points in $\mathsf{SA}_d(\mathcal{P})$ for pseudo-distributions. The dark blue points indicate integer solutions.

The d-th level of the SA hierarchy is only able to reason with non-negative juntas of degree at most d. This limits the d-th level to only being able to observe the correlations between subsets of at most d variables. Therefore, the d-th level of SA is unable to differentiate between true distributions and degree d *pseudo-distributions*, objects which appear to be true probability distributions when limited to only observing marginal distributions on at most d variables. We will first state the definition of a pseudo-distribution independent of the set of constraints $\mathcal{P}$, and later specify the conditions necessary for a pseudo-distribution can give rise to a pseudo-expectation for $\mathcal{P}$.

Definition 2.20 (Pseudo-Distribution)**.** A family of distributions $\mu := \{\mu_S\}_{|S| \leqslant d}$ is a degree d pseudo-distribution if $\mu_S : \{0,1\}^S \to \mathbb{R}^{\geqslant 0}$ and for every $S \subseteq T \subseteq [n]$ with $|T| \leqslant d$, and every assignment $\alpha \in \{0,1\}^S$,

$$\mu_S(\alpha) = \sum_{\substack{\beta \in \{0,1\}^T \\ \beta \restriction_S = \alpha}} \mu_T(\beta).$$

The next lemma shows that a point lies in the degree d SA polytope if it defines a degree d *pseudo-distribution*.

Lemma 2.21. *Any degree d pseudo-expectation implies a degree d pseudo-distribution.*

Proof. Let $\tilde{\mathbb{E}}$ be a degree d pseudo-expectation. Recall that we associate with every non-negative junta $J_{S',S\setminus S'}(x)$ an event $1_{S',S\setminus S'}$ that $x_i = 1$ for all $i \in S'$ and $x_j = 0$ for $j \in S \setminus S'$. Define the distribution $\mu_S : \{0,1\}^S \to \mathbb{R}$ as follows: for every $S' \subseteq S$,

$$\mu_S(1_{S',S\setminus S'}) = \tilde{\mathbb{E}}\Big[J_{S',S\setminus S'}(x)\Big].$$

Let μ be the collection of μ_S for every $|S| \leqslant d$. We verify that μ_S is indeed a probability distribution. First, observe that $\mu_S(1_{S',S\setminus S'}) \geqslant 0$ by definition of $\tilde{\mathbb{E}}$. That $\sum_{S'\subseteq S} \mu_S(1_{S',S\setminus S'}) = 1$ follows because

$$\sum_{S'\subseteq S} \mu_S(1_{S',S\setminus S'}) = \sum_{S'\subseteq S} \tilde{\mathbb{E}}\Big[J_{S',S\setminus S'}(x)\Big] = \tilde{\mathbb{E}} \sum_{S'\subseteq S} \big[J_{S',S\setminus S'}(x)\big] = \tilde{\mathbb{E}}[1] = 1.$$

Next, we will verify that the marginal distributions of μ are consistent. Let $S \subseteq L \subseteq [n]$ with $|L| \leqslant d$, then

$$\begin{aligned}
\sum_{\substack{L'\subseteq L \\ S'\subseteq L', L\setminus L'\subseteq S\setminus S'}} \mu_L\big(1_{L',L\setminus L'}\big) &= \sum_{\substack{L'\subseteq L \\ S'\subseteq L', L\setminus L'\subseteq S\setminus S'}} \tilde{\mathbb{E}}\left[J_{L',L\setminus L'}(x)\right] \\
&= \tilde{\mathbb{E}}\Bigg[\sum_{\substack{L'\subseteq L \\ S'\subseteq L', L\setminus L'\subseteq S\setminus S'}} J_{L',L\setminus L'}(x)\Bigg] \\
&= \tilde{\mathbb{E}}\Big[J_{S',S'\setminus S}(x)\Big] = \mu_S(1_{S',S'\setminus S}).
\end{aligned}$$

□

If we enforce that the expectation taken over such a pseudo-distribution must satisfy a set of polynomial inequalities $\mathcal{P}$, as well as to products of these polynomial inequalities with non-negative juntas, then such a pseudo-distribution will *fool* SA into believing that the pseudo-distribution is the set of marginals of some true distribution over solutions in $\{0,1\}^n$ that satisfy $\mathcal{P}$. This leads us to an alternative, equivalent definition of a pseudo-expectation, obtained by replacing conditions (1) and (2) in Definition 2.15 with the conditions of a pseudo-distribution.

Definition 2.22 (Pseudo-Expectation for $\mathcal{P}$)**.** Let $\mathcal{P}$ be a set of polynomial inequalities. A multilinearizing map $\tilde{\mathbb{E}} : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ is a degree d pseudo-expectation for $\mathcal{P}$ if there exists a degree d pseudo-distribution μ such that for every non-negative junta $J_{S,T}(x)$ with $\deg(J_{S,T}) \leqslant d$,

$$\tilde{\mathbb{E}}[J_{S,T}(x)] = \mu_{S\cup T}(1_{S,T}),$$

and $\tilde{\mathbb{E}}[J_{S,T}(x) \cdot P(x)] \geqslant 0$ for every $P(x) \geqslant 0 \in \mathcal{P}$ and every non-negative junta $J_{S,T}(x)$ with $\deg(P) + \deg(J_{S,T}) \leqslant d$.

That is, a pseudo-expectation is exactly an expectation taken over a pseudo-distribution. To see that this is equivalent to the original definition of a pseudo-expectation, note that one direction follows from Lemma 2.21. For the other direction, simply observe that $\tilde{\mathbb{E}}$ defined as in Definition 2.22, $\tilde{\mathbb{E}}[J_{S,T}(x)] = \mu_{S\cup T}(1_{S,T}) \geqslant 0$ and furthermore, that $\mu_\emptyset = 1$.

2.2.2.3 Evolution of the Sherali-Adams Relaxation

Psuedo-expectations give us a straightforward way to construct points that exist within the d-th level SA relaxation. However, this does not immediately give a clean description of how the points in the d-th level of SA hierarchy relate those in earlier levels. A characterization of the points that survive from the d-th level to the $(d+1)$-st is a particularly natural question when one wants to argue about the level of the SA hierarchy required to obtain a certain approximation ratio. In what follows we show that the points $\alpha \in \mathsf{SA}_{d-1}(\mathcal{P})$ that survive to $\mathsf{SA}_d(\mathcal{P})$ are exactly the set of points that, for every $i \in [n]$, can be written as a convex combination of points in $\mathsf{SA}_{d-1}(\mathcal{P})$ that are integer-valued in coordinate $\{i\}$. A consequence of this characterization is an alternative (and more illuminating) construction of a pseudo-distribution for $\mathsf{SA}_d(\mathcal{P})$, as well as a proof that $O(n)$ levels of the SA hierarchy are always sufficient to derive the integer hull.[3]

Lemma 2.23. *Let $\mathcal{P}$ be a set of polynomial inequalities. For every $\alpha \in \mathsf{SA}_d(\mathcal{P})$ and every $i \in [n]$ such that $0 < \alpha_{\{i\}} < 1$, there exists $\beta^{(0)}, \beta^{(1)} \in \mathsf{SA}_{d-1}(\mathcal{P})$ such that $\beta^{(1)}_{\{i\}} = 1$, $\beta^{(0)}_{\{i\}} = 0$, and*

$$\alpha \in \operatorname{conv}\left(\beta^{(0)}, \beta^{(1)}\right)^{4}.$$

Proof. Let $\alpha \in \mathsf{SA}_d(\mathcal{P})$ and recall that $\alpha \in \mathbb{R}^{\binom{n}{\leqslant d}}$ is indexed by subsets $I \subseteq [n]$ with $|I| \leqslant d$. Let $i \in [n]$ be a coordinate such that $0 < \alpha_{\{i\}} < 1$. Define points $\beta^{(0)}, \beta^{(1)}$ and non-negative multipliers $\lambda_0, \lambda_1 \in \mathbb{R}$ as

$$\begin{aligned} \beta^{(0)}_S &:= \frac{\alpha_S - \alpha_{S\cup\{i\}}}{1-\alpha_{\{i\}}}, & \lambda_0 &:= 1-\alpha_{\{i\}}, \\ \beta^{(1)}_S &:= \frac{\alpha_{S\cup\{i\}}}{\alpha_{\{i\}}}, & \lambda_1 &:= \alpha_{\{i\}}, \end{aligned}$$

for every $S \subseteq [n]$ with $|S| \leqslant d-1$. Observe that the following hold:

1. $\beta^{(0)}_{\{i\}} = 0$ and $\beta^{(1)}_{\{i\}} = 1$, and
2. $\lambda_0 \cdot \beta^{(0)}_S + \lambda_1 \cdot \beta^{(1)}_S = (\alpha_S - \alpha_{S\cup\{i\}}) + \alpha_{S\cup\{i\}} = \alpha_S$, and therefore α is a convex combination of $\beta^{(0)}$ and $\beta^{(1)}$.

[3] The proofs in this section follow the ideas of the proofs in the excellent notes of Rothvoß [135] on Sum-of-Squares.

[4] To simplify our notation we are being somewhat sloppy with this statement. α is an $\binom{n}{\leqslant d}$-dimensional vector and and $\beta^{(i)}$ is $\binom{n}{\leqslant d-1}$-dimensional vector, and we mean that α, when restricted to its first $\binom{n}{\leqslant d-1}$ coordinates can be written as a convex combination of points in $\mathsf{SA}_{d-1}(\mathcal{P})$

Next, we verify that $\beta^{(0)}, \beta^{(1)} \in \mathsf{SA}_{d-1}(\mathcal{P})$. Consider the multilinearization $\sum_{J \subseteq T}(-1)^{|J|} y_{S \cup T}$ of some non-negative $(d-1)$-junta $J_{S,T}(x)$. Evaluating this function on $\beta^{(0)}$, we have

$$\begin{aligned}\sum_{J \subseteq T}(-1)^{|J|}\beta^{(0)}_{S\cup T} &= \frac{\sum_{J\subseteq T}(-1)^{|J|}\alpha_{S\cup J} - \sum_{J\subseteq T}(-1)^{|J|}\alpha_{S\cup\{i\}\cup J}}{(1-\alpha_{\{i\}})},\\ &= \frac{\sum_{J\subseteq T\cup\{i\}}(-1)^{|J|}\alpha_{S\cup J}}{(1-\alpha_{\{i\}})} \geqslant 0,\end{aligned}$$

where the final inequality follows because $\alpha \in \mathsf{SA}_d(\mathcal{P})$ and therefore must satisfy the constraint $\sum_{J\subseteq T\cup\{i\}}(-1)^{|J|}y_{S\cup J} \geqslant 0$, the linearization the non-negative d-junta $J_{S,T\cup\{i\}}(x) \geqslant 0$.

Similarly, for $\beta^{(1)}$

$$\sum_{J\subseteq T}(-1)^{|J|}\beta^{(1)}_{S\cup T} = \frac{\sum_{J\subseteq T}(-1)^{|J|}\alpha_{S\cup\{i\}\cup J}}{\alpha_{\{i\}}} \geqslant 0,$$

where the final inequality follows by considering the following two cases: either $i \notin T$, in which case $\sum_{J\subseteq T}(-1)^{|J|}y_{S\cup\{i\}\cup J}$ is a non-negative d-junta, and so α assigns satisfies $\sum_{J\subseteq T}(-1)^{|J|}y_{S\cup\{i\}\cup J}$. Otherwise, if $i \in T$, then $\sum_{J\subseteq T\cup\{i\}}(-1)^{|J|}y_{S\cup J}$ is no longer a d-junta, because $S \cap T \neq \emptyset$. In particular,

$$\begin{aligned}\sum_{J\subseteq T}(-1)^{|J|}y_{S\cup J} &= \sum_{J\subseteq T: i\notin J}(-1)^{|J|}y_{S\cup J} + \sum_{J\subseteq T: i\in J}(-1)^{|J|}y_{S\cup J}\\ &= \sum_{J\subseteq T\setminus\{i\}}(-1)^{|J|}y_{S\cup J} + \sum_{J\subseteq T\setminus\{i\}}(-1)^{|J|+1}y_{S\cup J} = 0,\end{aligned}$$

where the second equality follows because S and T contain i, and so $S \cup \{i\} = S$.

Finally, observe that the same argument goes through when we consider the multilinearization of $J_{S,T}(x)\cdot P(x)$ for $P(x) \geqslant 0 \in \mathcal{P}$ and $\deg(J_{S,T}) + \deg(P) \leqslant d-1$. This follows because α belongs to $SA_d(\mathcal{P})$ and therefore satisfies the multilinearizations of both $J_{S\cup\{i\},T}(x)\cdot P(x) \geqslant 0$ and $J_{S,T\cup\{i\}}(x)\cdot P(x) \geqslant 0$. □

The converse of this lemma holds as well. We leave the details of the proof as an exercise, but note that it follows essentially by running the proof of the previous lemma in reverse.

Corollary 2.24. *Let $\mathcal{P}$ be a set of polynomial inequalities, $d \geqslant \deg(\mathcal{P})$, and $\alpha \in \mathbb{R}^{\binom{n}{\leqslant d+1}}$. If for every $i \in [n]$ there exists $\beta^{(0)}, \beta^{(1)} \in \mathsf{SA}_d(\mathcal{P})$ with $\beta^{(0)}_{\{i\}}, \beta^{(1)}_{\{i\}} \in \{0,1\}$ such that $\alpha_S = \lambda\beta^{(1)} + (1-\lambda)\beta^{(0)}$ for $\lambda \in [0,1]$ and $\alpha_{S\cup\{i\}} = \lambda\beta^{(1)}_S$, then $\alpha \in \mathsf{SA}_{d+1}(\mathcal{P})$.*

We can iterate this construction and obtain a characterization of the points that survive k-levels of the SA hierarchy. For every subset of $S \subseteq [n]$ with $|S| \leqslant k$, these points can be written as convex combinations of points in $\mathsf{SA}_{d-k}(\mathcal{P})$ that are integer-valued within S. To do this, it will be useful to define the SA hierarchy for $d < \deg(\mathcal{P})$. The constraints of

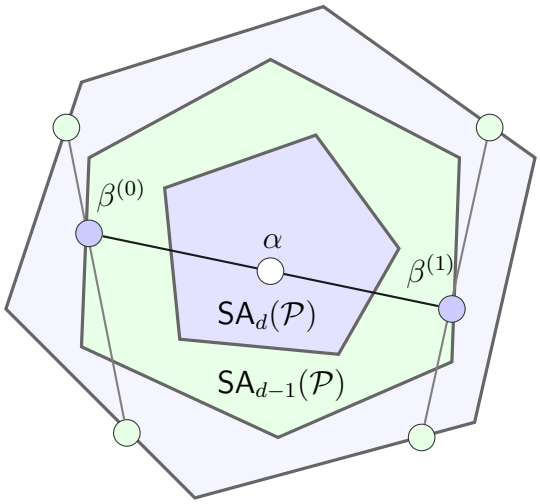


Figure 2.3: Interpolating feasible solutions in $\mathsf{SA}_d(\mathcal{P})$ from those in $\mathsf{SA}_{d-t}(\mathcal{P})$.

$\mathsf{SA}_d(\mathcal{P})$ are defined as usual, except that we will omit any constraint of degree greater than d. That is, for $d \leqslant \deg(\mathcal{P})$, $\mathsf{SA}_d(\mathcal{P})$ includes the multilinearizations of $J_{S,T}(x)$ for every non-negative junta $J_{S,T}(x)$ of degree at most d, as well as the multilinearizations of $J_{S,T}(x) \cdot P_i(x)$ for every $P_i(x) \geqslant 0 \in \mathcal{P}$ such that $\deg(J_{S,T}) + \deg(P_i) \leqslant d$.

Theorem 2.25. *Let $\mathcal{P}$ be a set of polynomial inequalities, and $0 \leqslant t \leqslant d$. For every $\alpha \in \mathsf{SA}_d(\mathcal{P})$, and every $S \subseteq [n]$ with $|S| = t$,*

$$\alpha \in \operatorname{conv}\left(\beta \in \mathsf{SA}_{d-t}(\mathcal{P}) : \beta_{\{i\}} \in \{0,1\}, \forall i \in S\right).$$

Proof. The proof is by induction on t. That it holds for $t = 1$ follows from Lemma 2.23. Now, suppose that the theorem holds for $t-1$. That is, for every $S \subseteq [n]$ with $|S| = t-1$, every $\alpha \in \mathsf{SA}_d(\mathcal{P})$ can be written as a convex combination of $\beta \in \mathsf{SA}_{d-(t-1)}(\mathcal{P})$ such that $\beta_{\{i\}} \in \{0,1\}$ for all $i \in S$. We will perform the following for every point β involved in this convex combination: Let $i \in [n] \setminus S$ be such that $0 < \beta_{\{i\}} < 1$. If no such i exists then we are done because β is a $\{0,1\}$-solution to $\mathcal{P}$. Otherwise, applying Lemma 2.23 to β and i allows us to write β as a convex combination of points $\beta^{(0)}, \beta^{(1)} \in \mathsf{SA}_{d-t}(\mathcal{P})$ such that $\beta^{(0)}_{\{i\}} = 0$, $\beta^{(1)}_{\{i\}} = 1$, and

$$\beta = \lambda_0 \beta^{(0)} + \lambda_1 \beta^{(1)},$$

where $\lambda_0 = (1 - \beta_{\{i\}})$ and $\lambda_1 = \beta_{\{i\}}$.

Finally, we need to show that applying Lemma 2.23 preserves $\{0,1\}$-coordinates; that is, $\beta^{(0)}_{\{j\}}, \beta^{(1)}_{\{j\}} \in \{0,1\}$ for all $j \in S$. To prove this, we will show that the order in which we condition on the variables is irrelevant. To show this, we will give a general formula for $\beta^{(0)}_{\{j\}}$ and $\beta^{(1)}_{\{j\}}$ in terms of the original point α. This shows that the order in which we apply Lemma 2.23 to the coordinates in $S \cup \{i\}$ does not matter. Recall that from the proof of

Lemma 2.23,

$$\begin{aligned} \beta_I^{(0)} &:= \frac{\beta_I - \beta_{I\cup\{i\}}}{1-\beta_{\{i\}}} & \lambda_0 &:= 1-\beta_{\{i\}}. \\ \beta_I^{(1)} &:= \frac{\beta_{I\cup\{i\}}}{\beta_{\{i\}}} & \lambda_1 &:= \beta_{\{i\}} \end{aligned} \tag{2.24}$$

The point β is formed by iteratively applying Lemma 2.23 to our current point γ (originally $\gamma = \alpha$) and an index $i \in [n]$, to split γ into two points. We will call the point labelled $\gamma^{(0)}$ the *negative point*, and the point labelled $\gamma^{(1)}$ the *positive point*. These points were obtained from the original point α by, at the k-th step of the recursion, applying Lemma 2.23 an index $i_k \in [n]$ and either the corresponding positive or the negative point from the previous step. Unrolling this recursion we obtain a sequence of $t-1$ points and indices from which we obtain β from α by applying Lemma 2.23,

$$\big((\alpha, i_1), (\gamma^{(p_2),2}, i_2), \ldots, (\gamma^{(p_{t-1}),t-1}, i_{t-1})\big)$$

where $i_\ell \in [n]$ is an index and $p_\ell \in \{0,1\}$ indicates whether the point $\gamma^{(p_\ell),\ell}$ was either the positive or negative point from the previous round. That is, β is obtained by an iterative process where, at the ℓ-th step, we apply Lemma 2.23 to the point $\gamma^{(p_{\ell-1}),\ell-1}$ and index $i_{\ell-1}$ and then retain the positive point if $p_\ell = 1$, and the negative point if $p_\ell = 0$.

If we let K be the set of indices $i_j \in [t-1]$ such that $p_{i_j+1} = 1$ (i.e. indices where we retain the positive point), and T be the set of indices $i_j \in [t-1]$ such that $p_{i_j+1} = 0$ (i.e. indices where retain the negative point), then we can unroll Equation 2.24 to write

$$\begin{aligned} \beta_I^{(0)} &= \frac{\sum_{J\subseteq T\cup\{i\}}(-1)^{|J|}\alpha_{I\cup K\cup J}}{\sum_{J\subseteq T\cup\{i\}}(-1)^{|J|}\alpha_{J\cup K}}, & \lambda_0 &= \sum_{J\subseteq T}(-1)^{|J|}\alpha_{J\cup K\cup\{i\}}, \\ \beta_I^{(1)} &= \frac{\sum_{J\subseteq T}(-1)^{|J|}\alpha_{I\cup K\cup\{i\}\cup J}}{\sum_{J\subseteq T}(-1)^{|J|}\alpha_{J\cup K\cup\{i\}}}, & \lambda_1 &= \sum_{J\subseteq T\cup\{i\}}(-1)^{|J|}\alpha_{J\cup K}. \end{aligned} \tag{2.25}$$

This expression is the same regardless of the order we apply Lemma 2.23, and therefore, by Lemma 2.23, implies that $\beta^{(0)}_{\{j\}}, \beta^{(1)}_{\{j\}} \in \{0,1\}$ for every $j \in S \cup \{i\}$. □

Observe that the converse of this theorem holds as well by repeated application of Corollary 2.24. In the remainder of this section we record some useful consequences of Theorem 2.25.

An Alternative Construction of Pseudo-Distributions. This theorem gives a necessary condition for points to survive t levels of the SA hierarchy. From this, we can derive an alternative construction of pseudo-distributions which give rise to pseudo-expectations for $\mathcal{P}$. To see this, let $\alpha \in \mathsf{SA}_d(\mathcal{P})$. For every $S \subseteq [n]$ with $|S| \leqslant d$, Theorem 2.25 gives a set of points $B_S \subseteq \mathsf{SA}_{d-|S|}(\mathcal{P})$ such that

$$\alpha = \sum_{\beta\in B_S} \lambda_\beta \beta,$$

where $\lambda_\beta \geqslant 0$ and $\sum_{\beta \in B_S} \lambda_\beta = 1$. Using this, we can construct a distribution $\mu_S : \{0,1\}^S \to \mathbb{R}^{\geqslant 0}$ such that for every $\kappa \in \{0,1\}^S$,

$$\mu_S(\kappa) \;:=\; \begin{cases} \lambda_\kappa & \text{if } \kappa \in B_S \restriction_S, \\ 0 & \text{otherwise,} \end{cases}$$

where $B_S \restriction_S = \{\beta \restriction_S : \beta \in B_S\}$ is the set of assignments β restricted to the coordinates S. Applying this procedure to every $S \subseteq [n]$ with $|S| \leqslant d$ gives us a family of distributions $\mu = \{\mu_S\}_{|S| \leqslant d}$. It can be checked that μ satisfies the definition of a degree d pseudo-distribution – the marginals condition follows from Equation 2.25.

Finally, observe that the pseudo-expectation defined by this pseudo-distribution is in fact a degree d pseudo-expectation for $\mathcal{P}$. Define the pseudo-expectation over μ, as $\tilde{\mathbb{E}}_\mu[\prod_{i \in S} x_i \prod_{j \in T}(1 - x_j)] = \mu_{S \cup T}[1_{S,T}]$ for every $|S| + |T| \leqslant d$, and extended linearly to polynomials. That $\tilde{\mathbb{E}}_\mu$ is a degree d pseudo-expectation for $\mathcal{P}$ follows by observing that $\tilde{\mathbb{E}}_\mu[\prod_{i \in S} x_i] = \alpha_S$.

Convergence to the Integer Hull. As a second consequence of Theorem 2.25, we argue that $n + \deg(\mathcal{P})$ levels of the SA hierarchy is always sufficient to converge to the integer hull.

Corollary 2.26. $\mathsf{SA}_{n+\deg(\mathcal{P})}(\mathcal{P}) = \mathsf{hull}_{\{0,1\}}(\mathcal{P})$.

Proof. Theorem 2.25 with $d = n + \deg(\mathcal{P})$ allows us to write any point $\alpha \in \mathsf{SA}_{n+\deg(\mathcal{P})}(\mathcal{P})$ as a convex combination of points $\beta \in \mathsf{SA}_{\deg(\mathcal{P})}$ such that for all $i \in [n]$, $\beta_{\{i\}} \in \{0,1\}$. Because $\deg(P_i) \leqslant \deg(\mathcal{P})$ for each $P_i(x) \geqslant 0 \in \mathcal{P}$, the projection of β to the original n variables, $\mathrm{proj}_{[n]}(\beta)$, must satisfy every constraint of $\mathcal{P}$. Therefore, every $\alpha \in \mathsf{SA}_{n+\deg(\mathcal{P})}$ can be written as a convex combination of $\{0,1\}$-solutions to $\mathcal{P}$. □

Remark. One might wonder why the the n-th level SA relaxation is not sufficient to derive the integer hull. This is a somewhat annoying technicality that depends on the definition of SA. Indeed, a degree n pseudo-distribution *is* a real probability distribution over $\{0,1\}^n$. The issue is that even if the pseudo-distribution gives rise to a pseudo-expectation for $\mathcal{P}$, we are not guaranteed that the points in $\{0,1\}^n$ in the support of this distribution are necessarily solutions to $\mathcal{P}$. This is because level n SA can only enforce constraints of the form $P_i(x) \cdot J_{S,T}(x)$ for non-negative juntas of degree at most $n - \deg(P_i)$ — in order to truly enforce that a solution in $\{0,1\}^n$ satisfies a constraint $P_i(x)$, we require the (linearizations of) constraints $P_i(x) \cdot J_{S,T}(x)$ for every junta of degree n. This can be seen by close examination of the proofs of Lemma 2.23 and Theorem 2.25.

Because of this, SA has been defined in several different ways in the literature. In its original form introduced by Sherali and Adams [141] the d-th level SA relaxation introduces linearizations of non-negative $J_{S,T}(x) \cdot P_i(x)$ where $J_{S,T}(x)$ is a degree d junta, ignoring the degree of $P_i(x)$. In this formulation, the n-th level of SA does indeed converge to the integer hull. The version of SA that we discuss here is standard in the proof complexity literature. From the perspective of proof complexity it is convenient to measure the degree as the maximum degree of the polynomial introduced, rather than only measuring the degrees of

the non-negative juntas. As well, this allows pseudo-distributions and pseudo-expectations to line up nicely, as they are defined only for polynomials of degree at most d. Furthermore, using polynomials of degree at most d is standard when discussing Sum-of-Squares from both the optimization and proof complexity perspectives.

2.2.3 Sherali-Adams as a Proof System

A strength of convex programming is the connection between upper and lower bounds that comes from duality. In the previous section we saw that the outcome of solving the SA relaxation of a set of polynomials $\mathcal{P}$ is an object that behaves locally like a distribution over $\{0,1\}$-solutions to the constraints of $\mathcal{P}$. However, if the SA relaxation is empty — i.e. the SA relaxation has discovered that there are no feasible $\{0,1\}$-solutions to $\mathcal{P}$ — then no such pseudo-expectation exists. In this case, solving the SA relaxation will return a certificate that the relaxation is empty; this is a consequence of LP-duality. These dual certificates have a particularly nice algebraic structure which will be the focus of this section. We will develop these dual certificates as proofs in a proof system, analogous to how the dual certificates for LPs were defined in Section 2.1.

Suppose that we want to certify that a polynomial $P(x)$ achieves a value of at least c_0 over the SA relaxation of $\mathsf{SA}_d(\mathcal{P})$. By Farka's Lemma (Lemma 2.3), it is enough to show that $P(x) - c_0$ is a non-negative linear combination of the constraints of the relaxation. Indeed, because any solution must satisfy the constraints of the SA, this implies that $P(x) \geqslant c_0$ over $\mathsf{SA}_d(\mathcal{P})$. In this setting it is standard to work over the original x-variables and polynomial constraints, along with the additional axioms $\pm(x_i^2 - x_i) \geqslant 0$ which facilitate multilinearizing.[5] Formally, a SA derivation is defined as follows.

Definition 2.27 (Sherali-Adams Derivation)**.** A SA derivation Π of a polynomial inequality $P(x) \geqslant c_0$ from a set of polynomial inequalities $\mathcal{P}$ is a formula of the form

$$\sum_{i=1}^{\ell} c_i \prod_{j \in S_i} x_j \prod_{k \in T_i} (1 - x_k) P_i(x) = P(x) - c_0,$$

where $c_i \in \mathbb{R}^{\geqslant 0}$ and each $P_i(x) \geqslant 0 \in \mathcal{P}$, or is one of the axioms of the form $x_i^2 - x_i \geqslant 0$ or $x_i - x_i^2 \geqslant 0$ for $i \in [n]$, or is $1 \geqslant 0$, and $S_i, T_i \subseteq [n]$ are multisets.

The *degree* of the refutation is the maximum degree of the polynomials $J_{S,T}(x)P_i(x)$ involved in the derivation,

$$\deg(\Pi) \;:=\; \max_{i \in [\ell]} \{|S_i| + |T_i| + \deg(P_i)\}.$$

Because the connection between the proof system and hierarchy perspectives of SA is parameterized only by the degree of the polynomials involved, the degree is the primary measure

[5] While the pre-linearized constraints along with the linearization axioms $\pm(x_i^2 - x_i) \geqslant 0$ may appear to be substantially stronger constraints than the multilinearizated SA constraints, we will see later in this section that in terms of what can be derived by each set of constraints, they are equivalent. Using the original variables and pre-linearized constraints is only a matter of convention.

of complexity studied for SoS. Even so, from the perspective of proof complexity, it is also natural to study the *size* of a derivation, size(Π), defined as the sum of the sizes of the polynomials[6] in the derivation.

In the case when there is no $\{0,1\}$-solution to a set of polynomial inequalities, their unsatisfiability can be witnessed by deriving the constant -1.

Definition 2.28 (Sherali-Adams Refutation)**.** A SA refutation of a set of polynomial inequalities $\mathcal{P}$ is a derivation of the constant -1 from $\mathcal{P}$.

Because SA derivations involve non-negative linear combinations of non-negative juntas and inequalities in $\mathcal{P}$, a derivation of -1 is only possible if $\mathcal{P}$ is infeasible.

Recall that in the definition of a SA derivation we allowed the sets of indices S_i, T_i to be multi-sets. This is because, rather than multilinearizing implicitly as we did in SA relaxations, it is standard in the setting of proof complexity to use the axioms $x_i^2 - x_i = 0$ to explicitly multilinearize the polynomials in the derivation. The following claim shows that the axioms $\pm(x_i^2 - x_i) \geqslant 0$ allow SA derivations to reproduce multilinearization without increasing the degree. Thus, any derivation done with the linearized set of constraints can be obtained as a SA derivation with the axioms $\pm(x_i^2 - x_i) \geqslant 0$.

Claim 2.29. *If there is a non-negative linear combination of the constraints of* $\mathsf{SA}_d(\mathcal{P})$ *equalling* $c_0 \in \mathbb{R}$*, then there exists a degree* d *SA derivation of* c_0 *from* $\mathcal{P}$*.*

The proof is straightforward and therefore we defer it to the Appendix.

Example 2.30. The term $-2x_1^3x_2^2$ is linearized by summing it with the following inequalities

$$\left\{2(x_1^2 - x_1)x_1x_2^2, \quad 2(x_1^2 - x_1)x_2^2, \quad 2(x_2^2 - x_2)x_1\right\}.$$

2.2.3.1 Refutations of CNF Formulae and a Simulation of Resolution

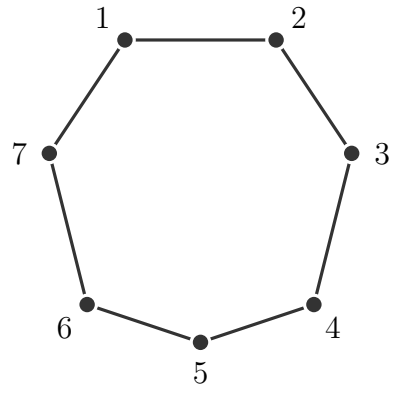
Typically, in proof complexity we are in refutations of unsatisfiable CNF formulas. Because SA operates over polynomials, we will need a translation of CNF formulas into polynomial inequalities in a way that preserves their semantics over $\{0,1\}^n$ assignments. The standard translation is as follows: Consider the CNF formula $\mathcal{C} = C_1 \vee C_2 \vee \ldots \vee C_m$. For each clause $C_i(I,J) := \bigvee_{i\in I} x_i \vee \bigvee_{j\in J} \neg x_j$, we introduce a polynomial

$$P_i(x) := \sum_{i\in I} x_i + \sum_{j\in J}(1 - x_j) - 1.$$

If we associate 1 with *true* and 0 with *false*, then it is straightforward to verify that the set of $\{0,1\}$-assignments that satisfy $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ is exactly the set of satisfying assignments to $\mathcal{C}$.

[6]The size of a polynomial is the number of bits needed to represent that polynomial.

Example 2.31 (Odd Cycle)**.** Consider an instance of the Max Independent Set problem defined on the following graph $G = (V, E)$:



We can express the constraints of this Independent Set instance as an ILP with constraints $x_i \in \{0, 1\}$, $x_i + x_j \leqslant 1$ for every $(i, j) \in E$. To obtain a solution we can solve the associated LP relaxation,

$$\max_{x \in \mathcal{P}} \sum_{i \in [7]} x_i$$

$$\text{where } \mathcal{P} = \left\{ \begin{array}{lll} x_1 + x_2 \leqslant 1, & x_2 + x_3 \leqslant 1, & x_3 + x_4 \leqslant 1, \\ x_3 + x_4 \leqslant 1, & x_4 + x_5 \leqslant 1, & x_6 + x_7 \leqslant 1, \\ x_7 + x_1 \leqslant 1, & 0 \leqslant x_1 \leqslant 1, & 0 \leqslant x_2 \leqslant 1, \\ 0 \leqslant x_3 \leqslant 1, & 0 \leqslant x_4 \leqslant 1, & 0 \leqslant x_5 \leqslant 1, \\ 0 \leqslant x_6 \leqslant 1, & 0 \leqslant x_7 \leqslant 1, & \end{array} \right\}$$

The optimum value of the of this instance of Independent Set is $\left(\frac{|cycle|-1}{2}\right) = 3$. On the other hand, the LP relaxation will return a value of at least 3.5, because $x_i = 0.5$ for $i \in [7]$ is a feasible solution for $\mathcal{P}$.

Now, consider the level 2 SA relaxation $\mathsf{SA}_2(\mathcal{P})$, consisting of the following constraints:

$$0 \leqslant y_{\{i\}} \leqslant 1 \qquad \forall i \in [7] \tag{2.26}$$

$$0 \leqslant y_{\{i,k\}} \leqslant y_{\{k\}} \qquad \forall i, k \in [7] \tag{2.27}$$

$$0 \leqslant y_{\{i\}} - y_{\{i,k\}} \leqslant 1 - y_{\{k\}} \qquad \forall i, k \in [7] \tag{2.28}$$

$$y_{\{i\}} + y_{\{j\}} \leqslant 1 \qquad \forall (i, j) \in E \tag{2.29}$$

$$y_{\{i,k\}} + y_{\{j,k\}} \leqslant y_{\{k\}} \qquad \forall (i, j) \in E, k \in [7] \tag{2.30}$$

$$y_{\{i\}} - y_{\{i,k\}} + y_{\{j\}} - y_{\{j,k\}} \leqslant 1 - y_{\{k\}} \qquad \forall (i, j) \in E, k \in [7] \tag{2.31}$$

To see how this relaxation compares to the LP relaxation, we can attempt to derive an upper bound on the value produced by this relaxation. Consider the following derivation, which we state over the y-variables, but by Claim 2.29 can be simulated in a SA derivation.

	Derive:	By:
(a)	$y_{\{1,2\}} \leqslant 0$	(2.30) with $k = 1$, and $(i, j) = (1, 2)$
(b)	$y_{\{2\}} - y_{\{1,2\}} + y_{\{3\}} - y_{\{1,3\}}$	(2.31) with $k = 1$, and $(i, j) = (2, 3)$
(c)	$y_{\{1,3\}} + y_{\{1,4\}} \leqslant y_{\{1\}}$	(2.30) with $k = 1$, and $(i, j) = (3, 4)$
(d)	$y_{\{4\}} - y_{\{1,4\}} + y_{\{5\}} - y_{\{1,5\}} \leqslant 1 - y_{\{1\}}$	(2.31) with $k = 1$, and $(i, j) = (4, 5)$
(e)	$y_{\{1,5\}} + y_{\{1,6\}} \leqslant y_{\{1\}}$	(2.30) with $k = 1$, and $(i, j) = (5, 6)$
(f)	$y_{\{6\}} - y_{\{1,6\}} + y_{\{7\}} - y_{\{1,7\}} \leqslant 1 - y_{\{1\}}$	(2.31) with $k = 1$, and $(i, j) = (6, 7)$
(g)	$y_{\{1,7\}} \leqslant 0$	(2.30) with $k = 1$, and $(i, j) = (1, 7)$

Summing up (a) – (g) derives the following upper bound,

$$y_{\{1\}} + y_{\{2\}} + y_{\{3\}} + y_{\{4\}} + y_{\{5\}} + y_{\{6\}} + y_{\{7\}} \leqslant 3. \tag{2.32}$$

Because any point that satisfies a set of linear inequalities must as well satisfy any non-negative linear combination of those constraints, every point in $\mathsf{SA}_2(\mathcal{P})$ must satisfy (2.32). This upper bound matches the true optimal solution to this instance of Max Independent Set, and therefore maximizing $\sum_{i\in[7]} y_{\{i\}}$ over the level 2 SA relaxation returns the optimal solution to this instance of Independent Set.

Now, consider adding the constraint

$$\text{(h)} \qquad \sum_{i\in[7]} x_i \geqslant 3.5,$$

which causes $\mathcal{P}$ to become unsatisfiable. We will give a SA refutation of $\mathcal{P} \cup \{\sum_{i\in[7]} x_i \geqslant 3.5\}$, this time over the original x-variables. Consider the constraints (a) - (h) before linearization, as products of constraints and 1-juntas:

	Linearization:	Pre-linearization:
(a)	$y_{\{1,2\}} \leqslant 0$	$x_1(x_1 + x_2 \leqslant 1)$
(b)	$y_{\{2\}} - y_{\{1,2\}} + y_{\{3\}} - y_{\{1,3\}}$	$(1 - x_1)(x_2 + x_3 \leqslant 1)$
(c)	$y_{\{1,3\}} + y_{\{1,4\}} \leqslant y_{\{1\}}$	$x_1(x_3 + x_4 \leqslant 1)$
(d)	$y_{\{4\}} - y_{\{1,4\}} + y_{\{5\}} - y_{\{1,5\}} \leqslant 1 - y_{\{1\}}$	$(1 - x_1)(x_4 + x_5 \leqslant 1)$
(e)	$y_{\{1,5\}} + y_{\{1,6\}} \leqslant y_{\{1\}}$	$x_1(x_5 + x_6 \leqslant 1)$
(f)	$y_{\{6\}} - y_{\{1,6\}} + y_{\{7\}} - y_{\{1,7\}} \leqslant 1 - y_{\{1\}}$	$(1 - x_1)(x_6 + x_7 \leqslant 1)$
(g)	$y_{\{1,7\}} \leqslant 0$	$x_1(x_1 + x_7 \leqslant 1)$
(h)	$\sum_{i=1}^{7} y_{\{i\}} \geqslant 3.5$	$\sum_{i=1}^{7} x_i \geqslant 3.5$

Because $\mathcal{P}$ is unsatisfiable, by Farkas' Lemma there exists a non-negative linear combi-

nation of (a) - (h) equalling $1 \leqslant 0$:

$$\begin{aligned}
&2x_1(x_1+x_2-1)+2(1-x_1)(x_2+x_3-1)+2x_1(x_3+x_4-1)\\
&+2(1-x_1)(x_4+x_5-1)+2x_1(x_5+x_6-1)+2(1-x_1)(x_6+x_7-1)\\
&+2x_1(x_1+x_7-1)+4(x_1^2-x_1)+2(3.5-x_1-x_2-x_3-x_4-x_5-x_6-x_7)\\
&=1 \leqslant 0
\end{aligned}$$

This is a degree 2 SA refutation of $\mathcal{P}$.

Unfortunately, as we saw in Example 2.19, there are instances of Independent Set where level 2 SA performs very badly. Indeed, Tulsiani [148] showed that there exist instances for which the SA, as well as the Sum-of-Squares relaxation obtains an integrality gap of $n/2^{\Theta(\sqrt{\log n \log\log n}}$ even after $2^{\Theta(\sqrt{\log n \log\log n}}$ levels.

We end this section by showing that SA is a non-trivial proof system by showing that SA can p-simulate[7] the Resolution proof system. Recall that the Resolution proof system was defined in Section 1.1. The *width* of a Resolution refutation is maximum number of literals occurring in an any clause in the refutation. We will show that any Resolution refutation of width w can be transformed into a SA refutation of degree $w+2$. Furthermore, the size of the SA refutation will be polynomial in the size of the Resolution refutation, where the size of an SA is simply measured as the number of bits needed to express the refutation. This was originally observed by Dantchev et al. [51].

Lemma 2.32 (SA p-simulates Resolution)**.** *For any width-w and size-S Resolution refutation, there exists a degree $(w+2)$ SA refutation of size at most $\mathsf{poly}(S)$ of the same formula.*

Proof. Let $\mathcal{C}$ be an unsatisfiable CNF formula and recall that the clauses of $\mathcal{C}$ are given to SA as a set of polynomial inequalities $\mathcal{P}$, where $C(S,T) := \bigvee_{i\in S} x_i \vee \bigvee_{j\in T} \neg x_j$ is encoded as the polynomial $P(x) := \sum_{i\in S} x_i \sum_{j\in T}(1-x_j) \geqslant 1$. Our proof will proceed in two steps.

(i) First, observe that an equivalent representation of each of the initial clauses $C(S,T) := \bigvee_{i\in S} x_i \vee \bigvee_{j\in T} \neg x_j$ is given by the degree d non-negative junta $-J_{T,S}(x) \geqslant 0$. We show that this non-negative junta can be derived in degree that is bounded by the width of $C(S,T)$.

(ii) Second, we show that SA can simulate the Resolution rule, deriving a clause $C(S\cup S', T\cup T')$ from clauses $C(S\cup\{i\},T)$ and $C(S',T'\cup\{i\})$ in degree at most the width of $C(S\cup S', T\cup T')$. This step requires some care to ensure that repeated simulations of the Resolution rule do not blow up the degree of the refutation.

To prove (i), consider an axiom $P_i(x) := \sum_{i\in S} x_i + \sum_{j\in T}(1-x_j) - 1 \geqslant 0$. Let $\ell \in T$ be some distinguished variable. Multiply $P_i(x)$ by the non-negative junta $J_{T\setminus\{\ell\},S}(x)$,

$$P_i(x)\cdot J_{T\setminus\{\ell\},S}(x) = \Big(\sum_{i\in S} x_i - \sum_{j\in T} x_j + |T| - 1\Big)\cdot J_{T\setminus\{\ell\},S}(x)$$

[7]Recall that p-Simulation was defined in Definition 1.4.

$$= \Big(-\sum_{j \in T} x_j + |T| - 1 \Big) \cdot J_{T\setminus\{\ell\},S}(x) + \sum_{j \in T} (x_j - x_j^2) \cdot J_{T\setminus\{\ell\},S\setminus\{j\}}(x)$$
$$= \Big(-\sum_{j \in T} x_j + |T| - 1 \Big) \cdot J_{T\setminus\{\ell\},S}(x) + 0, \tag{2.33}$$

where the final line used the fact that SA can derive $x_j^2 - x_j = 0$. Now, isolating the variable x_ℓ from the sum in (2.33), we have

$$= -x_\ell \cdot J_{T\setminus\{\ell\},S}(x) - \Big(\sum_{j \in T\setminus\{\ell\}} x_j \cdot J_{T\setminus\{\ell\},S}(x) \Big) + (|T| - 1) \cdot J_{T\setminus\{\ell\},S}(x)$$
$$= -J_{T,S}(x) - \Big((|T| - 1) \cdot J_{T\setminus\{\ell\},S}(x) \Big) + (|T| - 1) \cdot J_{T\setminus\{\ell\},S}(x) = -J_{T,S}(x) \geqslant 0,$$

where the second equality used the fact that if $j \in T \setminus \{\ell\}$, then $x_j \cdot J_{T\setminus\{\ell\},S}(x) = x_j^2 \cdot J_{T\setminus\{\ell,j\},S}(x) = J_{T\setminus\{\ell\},S}(x)$. This takes advantage of the fact that SA can derive $x_j^2 = x_j$. Observe that the degree is at most the width of the initial clause, and we have only used a linear number of additional polynomials, each with constant coefficients.

To prove (ii), suppose that we have already derived the hypotheses $-J_{T\cup\{i\},S}(x) \geqslant 0$ and $-J_{T',S'\cup\{i\}}(x)$, corresponding to the clauses $C(S \cup \{i\}, T)$ and $C'(S', T' \cup \{i\})$. We derive the resolvent $-J_{T\cup T',S\cup S'}(x) \geqslant 0$ in two steps, first weakening the inequalities, and then resolving on the complementary variable. This is done by deriving the following inequalities.

$$\textbf{Weakening Inequalities:} \begin{cases} J_{T\cup\{i\},S}(x) - J_{T\cup\{i\}\cup T',S\cup S'}(x) \geqslant 0 \\ J_{T,S\cup\{i\}}(x) - J_{T\cup T',S\cup\{i\}\cup S'}(x) \geqslant 0 \end{cases}$$
$$\textbf{Resolution Inequality:} \Big\{ J_{T\cup\{i\}\cup T',S\cup S'}(x) + J_{T\cup T',S\cup\{i\}\cup S'}(x) - J_{T\cup T',S\cup S'}(x) \geqslant 0$$

Once these inequalities have been derived, we can simply add them together with $-J_{T\cup\{i\},S}(x) \geqslant 0$ and $-J_{T',S'\cup\{i\}}(x) \geqslant 0$ to derive the resolvent. Furthermore, the derivation of weakening and resolution inequalities will not require us to use any of inequalities that we have derived thus far (that is, they are simply linear combinations of non-negative juntas), and therefore will not cause any blow-up in the degree of the proof.

The resolution inequality is straightforward to derive. First, observe that the Resolution inequality can be expanded as

$$J_{T\cup\{i\}\cup T',S\cup S'}(x) + J_{T\cup T',S\cup\{i\}\cup S'}(x) - J_{T\cup T',S\cup S'}(x)$$
$$= (x_i + (1 - x_i) - 1) \cdot J_{T\cup T',S\cup S'}(x). \tag{2.34}$$

Let $\ell \in T \cup T'$ be some arbitrary index. Consider the following sum of non-negative juntas,

$$x_i(1-x_i) \cdot J_{T\cup T',S\cup S'}(x) + x_\ell(1-x_\ell) \cdot J_{T\cup T',S\cup S'}(x) = (x_i - x_i) \cdot J_{T\cup T',S\cup S'}(x) + (1-1) \cdot J_{T\cup T',S\cup S'}(x),$$

which by the expansion in (2.34) is the resolution inequality. Here we have used the fact that SA can deduce that $x_\ell \cdot J_{T\cup T',S\cup S'}(x) = J_{T\cup T',S\cup S'}(x)$, using $x_\ell^2 = x_\ell$. Observe that the

degree of this derivation is at most $\deg(J_{T\cup T',S\cup S'}) + 2$. Because the degree of $J_{T\cup T',S\cup S'}$ is exactly the width of the corresponding clause $C(S \cup S', T \cup T')$, our bound on the width of the refutation holds. Similarly, observe that only a linear number of polynomials have been introduced, each with constant coefficients.

The derivation of the weakening inequalities follow from a similar argument. As they are easier to believe, we leave their proof as an exercise.

□

2.2.3.2 Soundness, Completeness, and Duality

SA is a proof system in the traditional sense, meaning that the proofs that it produces are efficiently verifiable and that it is both sound and complete. Both soundness and completeness follow from the duality between SA derivations and pseudo-expectations. Indeed, if the SA relaxation is infeasible, then a certificate of this fact exists in the form of an SA refutation. Conversely, if the relaxation is feasible, then a derivation of this fact can be found.

Theorem 2.33 (Soundness and Completeness of Sherali-Adams)**.** *Let $\mathcal{P}$ be a set of polynomial inequalities. There exists a degree d SA refutation of $\mathcal{P}$ if and only if $\mathsf{SA}_d(\mathcal{P})$ is infeasible.*

Proof. Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be a set of polynomial inequalities. Suppose that there is a degree d SA refutation of $\mathcal{P}$,

$$\sum_{i=1}^{\ell} c_i \cdot J_{S_i,T_i}(x) \cdot P_i(x) = -1,$$

where $P_i(x) \geqslant 0 \in \mathcal{P}$, or $P_i(x)$ is one of the axioms $x_i^2 - x_i$, or $x_i - x_i^2$, or the constant 1, $J_{S_i,T_i}(x)$ is a d-junta, and $c_i \in \mathbb{R}^{\geqslant 0}$. For contradiction, suppose that $\mathsf{SA}_d(\mathcal{P})$ is non-empty. Let $\alpha \in \mathsf{SA}_d(\mathcal{P})$, and let $\tilde{\mathbb{E}}_\alpha$ be the pseudo-expectation defined by α: $\tilde{\mathbb{E}}_\alpha[\prod_{i\in S} x_i] = \alpha_S$ for every $S \subseteq [n]$ with $|S| \leqslant d$. Applying $\tilde{\mathbb{E}}_\alpha$ to both sides of the SA refutation,

$$\tilde{\mathbb{E}}_\alpha[-1] = \tilde{\mathbb{E}}_\alpha\left[\sum_{i=1}^{\ell} c_i(J_{S_i,T_i}(x) \cdot P_i(x))\right],$$

which, by linearity of expectation, and the requirement that $\tilde{\mathbb{E}}[1] = 1$, this is equivalent to

$$-1 = \sum_{i=1}^{\ell} c_i \cdot \tilde{\mathbb{E}}_\alpha\big[J_{S_i,T_i}(x) \cdot P_i(x)\big] \geqslant 0.$$

The final inequality follows because $\alpha \in \mathsf{SA}_d(\mathcal{P})$.

For the converse, suppose that $\mathsf{SA}_d(\mathcal{P})$ is empty. We will use this to derive a degree d refutation of $\mathcal{P}$. By Farkas' Lemma (Lemma 2.3) there exists a non-negative linear combination of the inequalities that define $\mathsf{SA}_d(\mathcal{P})$ equalling -1. By Claim 2.29, this implies that

there is a SA derivation of -1,

$$\sum_{i=1}^{\ell} \lambda_i \cdot (P_i(x) \cdot J_{S_i,T_i}(x)) = -1$$

where $\lambda_i \geqslant 0$ are the coefficients in the non-negative combination, $P_i(x) \in \mathcal{P} \cup \{1 \geqslant 0, x_i^2 - x_i \geqslant 0, x_i - x_i^2 \geqslant 0\}$ and $J_{S,T}$ is a d-junta. This is a degree d SA refutation of $\mathcal{P}$. □

In fact, SA is *derivationally complete* as well, meaning that any inequality $P(x) \geqslant c_0$ that is logically implied[8] by $\mathcal{P}$ can be derived in SA. This is a consequence of the fact that degree d SA proof system and the d-th level SA relaxation are LP duals of each other. To show this, we will phrase the task of finding a degree d SA derivation of $P(x) \geqslant c_0$ as a linear program: For each product $J_{S,T}(x) \cdot P_i(x)$ of a non-negative junta with an axiom of degree at most d, introduce a variable c_{S,T,P_i} representing the coefficient of this polynomial. The task of finding a SA derivation of $P(x) \geqslant c_0$ is equivalent to solving the following LP,

$$P(x) - c_0 = \sum_{\substack{S,T \subseteq [n], \\ P_i(x) \geqslant 0 \in \mathcal{P} \cup \{\pm(x_i^2 - x_i) \geqslant 0,\ 1 \geqslant 0\} \\ \deg(J_{S,T}) + \deg(P_i) \leqslant d}} c_{S,T,P_i} \cdot P_i(x) \cdot J_{S,T}(x), \tag{2.35}$$

$$c_{S,T,P_i} \geqslant 0 \qquad \forall S, T, P_i.$$

Here the coefficients c_{S,T,P_i} are the variables of the LP, while the variables x of the polynomials are treated as constants. The following theorem proves that this LP is the dual to $\mathsf{SA}_d(\mathcal{P})$, as well as establishes derivational completeness for the SA proof system. For ease of notation, we will phrase this theorem in the language of pseudo-expectations[9].

Theorem 2.34 (Duality and Derivational Completeness of Sherali-Adams)**.** *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be a set of polynomial inequalities. For any $P(x) \in \mathbb{R}[x]$ with $\deg(P) \leqslant d$,*

$$\min\left\{\tilde{\mathbb{E}}[P(x)] | \tilde{\mathbb{E}} \in \mathcal{E}_d(\mathcal{P})\right\} = \max\left\{c_0 | \exists \text{ a degree } d \text{ SA derivation of } P(x) \geqslant c_0 \text{ from } \mathcal{P}\right\}.$$

Proof. Let c_0 be the maximum value such that there exists a degree d SA derivation of $P(x) \geqslant c_0$ from $\mathcal{P}$,

$$\sum_{i=1}^{\ell} c_i \cdot J_{S_i,T_i}(x) \cdot P_i(x) = P(x) - c_0, \tag{2.36}$$

where $P_i(x) \geqslant 0 \in \mathcal{P}$ or $P_i(x)$ is one of the axioms $\pm(x_i^2 - x_i) \geqslant 0$, or $1 \geqslant 0$, $J_{S_i,T_i}(x)$ is a $(d - \deg(P_i))$-junta, and $c_i \geqslant 0$. Let $\tilde{\mathbb{E}}$ be the pseudo-distribution in $\mathcal{E}_d(\mathcal{P})$ that assigns the

[8] We say that a polynomial $P(x) \geqslant c$ is logically implied by $\mathcal{P}$ if for every $\alpha \in \mathcal{P} \cap \{0,1\}^n$, it holds that $P(\alpha) \geqslant c$.

[9] Recall that $\mathcal{E}_d(\mathcal{P})$ is the set of all degree d pseudo-expectations for $\mathcal{P}$ and that each $\tilde{\mathbb{E}} \in \mathcal{E}_d(\mathcal{P})$ corresponds to a feasible solution in $\mathsf{SA}_d(\mathcal{P})$.

minimum value to $P(x)$. If $\tilde{\mathbb{E}}[P] = c_0$ we are done, so suppose that $\tilde{\mathbb{E}}[P] < c_0$. Applying $\tilde{\mathbb{E}}$ to both sides of the derivation, we have

$$\sum_{i=1}^{\ell} c_i \cdot \tilde{\mathbb{E}}\left[J_{S_i,T_i}(x) \cdot P_i(x)\right] = \tilde{\mathbb{E}}\left[P(x)\right] - c_0.$$

Because $\tilde{\mathbb{E}}$ is a degree d pseudo-distribution for $\mathcal{P}$, the left-hand side evaluates to $\geqslant 0$. On the other hand, $\tilde{\mathbb{E}}\left[P(x)\right] - c_0 < 0$ by assumption; a contradiction.

For the converse, suppose that the minimum $\tilde{\mathbb{E}} \in \mathcal{E}_d$ assigns $\tilde{\mathbb{E}}[P(x)] = c_0$. Then, for any $\varepsilon > 0$, $\mathsf{SA}_d(\mathcal{P}) \cup \{P(x) \leqslant c_0 - \varepsilon\}$ is unsatisfiable. Let $L(y)$ be the multilinearization of $P(x)$ over the lifted variables y. By Farkas' Lemma (Lemma 2.3) there exists a non-negative linear combination of the inequalities in $\mathsf{SA}_d(\mathcal{P}) \cup \{L(y) \leqslant c_0 - \varepsilon\}$ evaluating to $-1 \geqslant 0$. By Claim 2.29, this non-negative combination can be mimicked by a SA derivation of degree at most d,

$$\lambda(P(x) - c_0 + \varepsilon) + \sum_{i=1}^{\ell} \lambda_i \cdot P_i(x) \cdot J_{S_i,T_i}(x) = -1,$$

where $\lambda_i \geqslant 0$ are the coefficients in the convex combination, $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0,\ \pm(x_i - x_i^2) \geqslant 0\}$, and each $J_{S_i,T_i}(x)$ is a degree $(d - \deg(P_i))$ non-negative junta. Setting $\varepsilon = 1/\lambda$, this equals

$$\sum_{P_i(x) \in \mathcal{P}} \lambda_i \cdot P_i(x) = \lambda\left(P(x) - c_0\right).$$

Thus, $\sum_{P_i(x)\in\mathcal{P}} \frac{\lambda_i}{\lambda} P_i(x) = P(x) - c_0$ is a degree d SA derivation of $P(x) \geqslant c_0$. □

A consequence of Theorem 2.33, together with the proof of Theorem 2.33 is that any polynomial inequality $P(x) \geqslant c_0$ that is logically implied by a set of polynomial inequalities $\mathcal{P}$ has an SA derivation of degree at most n.

2.2.3.3 Automatizability

A key feature of SA, that results from its tight connection with linear programming, is that low-degree SA proofs can be found efficiently.

Lemma 2.35 (Degree-Automatizability of Sherali-Adams)**.** *Let $\mathcal{P}$ be any set of m polynomial inequalities over n variables. Any degree d SA derivation can be found in time $m \cdot n^{O(d)}$.*

Proof. Recall that in (2.35) in the previous section we saw how to phrase the task of finding a SA derivation as an LP. There are at most $\binom{2n}{d}$ possible d-juntas, and so at most $O(m\binom{2n}{d})$ inequalities of the form $J_{S,T}(x) \cdot P_i(x)$ with $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{\pm(x_i^2 - x_i) \geqslant 0,\ 1 \geqslant 0\}$. Therefore, this LP has at most $m \cdot n^{O(d)}$ variables and constraints, and thus can be solved by standard linear programming algorithms in time $m \cdot n^{O(d)}$. The outcome of this program is a SA derivation of $P(x) \geqslant c_0$ over $\mathcal{P}$. □

It should be stressed that this does *not* imply that SA is polynomially automatizable in the traditional sense of Definition 1.5.

Size Automatizability. Given that the connection between the SA hierarchy and the SA proof system depends only on the degree of the proof, it is not surprising that the bulk of work has focused on the degree-automatizability of SA. Even so, in the context of proof search, it is perhaps equally as natural to study the size automatizability of SA. In particular, whether high-degree SA proofs can be efficiently provided that they have small size.

It is known that both Resolution and Polynomial Calculus are not automatizable under strong complexity-theoretic assumptions [12, 2, 56]. However, for SA it remains unknown whether proofs can be found in polynomial, or even sub-exponential time in the size of the shortest proof. Indeed, the best known upper-bound for finding a SA proof is of size S is $m \cdot n^{O(\sqrt{n \log S})}$. This follows from the size-degree tradeoff for SA. Define the *monomial size* S_m of a SA proof Π to be the number of monomials in the proof Π when it is expanded as a sum of monomials before any cancellations occur. Monomial size differs from the standard notion of size as it does not take into account the size of the coefficients that occur in the proof.

Lemma 2.36 (Size-Degree Tradeoff for Sherali-Adams [120])**.** *Let $\mathcal{P}$ be a set of m polynomial inequalities. Any SA derivation of monomial size S_m from $\mathcal{P}$ implies a derivation of degree $O(\sqrt{n \log S_m} + \deg(\mathcal{P})$.*

Given that a primary reason for studying SA is that its proofs can be found efficiently in the degree of the proof, it remains an in interesting question whether the same can be said for size. That is, whether SA is polynomially automatizable.

Chapter 3

Sum-of-Squares

3.1 Semidefinite Programming and PSD Matrices

Recall that a generic linear program is defined as:

$$\mathcal{LP}(\mathcal{P},c) = \min_{x\in\mathcal{P}} \; c^\top x$$
$$\text{where } \mathcal{P} = \{Ax = b,\; x \geqslant 0\},$$

over variables $x = \{x_1, \ldots, x_n\}$. We can assume without loss of generality that the only inequalities occurring in $\mathcal{P}$ are in the non-negativity constraints $x \geqslant 0$ by introducing additional slack variables. Geometrically the linear program $\mathcal{LP}(\mathcal{P}, c)$ corresponds to optimizing over a polytope, the intersection of an affine space $Ax = b$ and the convex cone of $\{x : x \geqslant 0\}$.

Definition 3.1 (Convex Cone)**.** Let V be a vector space and $C \subseteq V$. Then C is a convex cone if for every $x, y \in C$ and $\alpha, \beta \in \mathbb{R}^{\geqslant 0}$, $(\alpha x + \beta y) \in C$.

A semidefinite program relaxes the non-negativity constraint $x \geqslant 0$ to only require that the variables, when arranged as a matrix, are symmetric positive semidefinite.

Definition 3.2 (Positive Semidefinite Matrix)**.** An $n \times n$ matrix A is *positive semidefinite* (PSD) if for every vector $u \in \mathbb{R}^n$,

$$u^\top A u \geqslant 0.$$

If this inequality is strictly positive, we say that A is *positive definite*.

We will denote by $A \succeq 0$ that the matrix A is both symmetric and positive semidefinite, and by $A \succ 0$ that A is symmetric and positive definite.

Proposition 3.3. *If $A_1, \ldots, A_\ell \in \mathbb{R}^{n\times n}$ are symmetric PSD matrices, then for any $c_1, \ldots, c_\ell \in \mathbb{R}^{\geqslant 0}$,*

$$(c_1 A_1 + \ldots + c_\ell A_\ell) \succeq 0.$$

That is, the set of positive semidefinite matrices forms a cone in $\mathbb{R}^{n\times n}$.

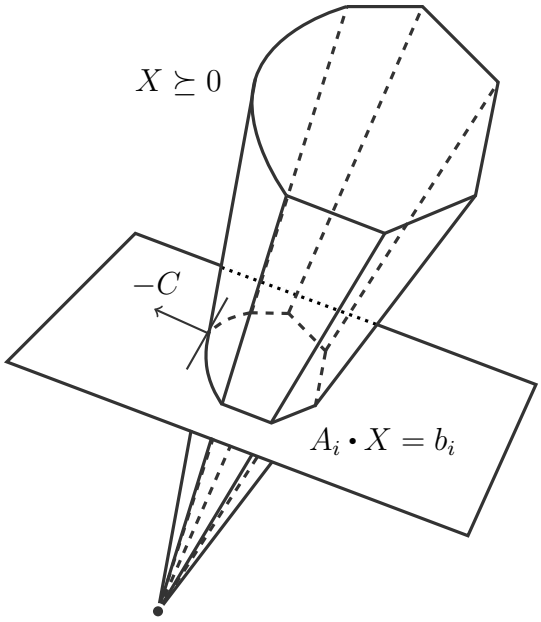


Figure 3.1: A semidefinite program minimizing $C \cdot X$ over the spectahedron formed from the intersection of the cone $X \succeq 0$ with the affine subspace $A_i \cdot X = b_i$ for $i \in [m]$.

A semidefinite program is an optimization problem over the intersection of a set of linear constraints and the convex cone of symmetric PSD matrices. Let $X = \{x_{1,1}, \ldots, x_{n,n}\}$ be the set of variables over which we are working. We will think of X as arranged as an $n \times n$ matrix where $X_{i,j} = x_{i,j}$. For two $n \times n$ matrices A and B, denote by $A \bullet B$ the inner product

$$A \bullet B = \sum_{i,j} A_{i,j} B_{i,j}.$$

Definition 3.4. (Semidefinite Program) A semidefinite program (SDP) is a mathematical program of the form

$$\begin{aligned} \mathcal{SDP}(\mathcal{S}, C) &:= \min_{X \in \mathcal{S}} C \bullet X \\ \text{where } \mathcal{S} &:= \{A_1 \bullet X = b_1, \ \ldots, \ A_m \bullet X = b_m, \ X \succeq 0\}, \end{aligned}$$

where $C, A_i \in \mathbb{R}^{n \times n}$ are symmetric, and $b_i \in \mathbb{R}$.

Geometrically, an SDP corresponds to optimizing a linear objective function over a *spectahedron* $\mathcal{S}$, the intersection of a convex cone of symmetric PSD matrices $X \succeq 0$ with an affine subspace $A_i \bullet X = 0$. This is shown in figure 3.1. It is not hard to see that a convex polytope is a spectahedron; indeed, we will show that linear programming is formally a special case of semidefinite programming. Given a linear program $\mathcal{LP}(\mathcal{P}, c)$, where (possibly after introducing slack variables)

$$\mathcal{P} = \{a_1^\top x = b_1, \ldots, a_m^\top x = b_m, x \geqslant 0\},$$

we can define a semidefinite program $\mathcal{SDP}(\mathcal{S}, C)$, where the spectahedron $\mathcal{S}$ consists of the following constraints

$$\begin{aligned} A_i \bullet X &= b_i && \text{for } i \in [m], \\ X_{i,j} &= 0 && \text{for } i \neq j \text{ and } i, j \in [n], \\ X &\succeq 0. \end{aligned}$$

The matrices A_i and C are defined as follows

$$A_i = \begin{pmatrix} a_{i,1} & 0 & \dots & 0 \\ 0 & a_{i,2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{i,n} \end{pmatrix} \qquad C = \begin{pmatrix} c_1 & 0 & \dots & 0 \\ 0 & c_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_n \end{pmatrix},$$

where $a_{i,j}$ is the j-th entry of the vector a_i. The final step is to prove that the constraints of $\mathcal{S}$ force $X \succcurlyeq 0$. For this, it suffices to show that the following claim.

Claim 3.5. *The diagonal entries of any PSD matrix are non-negative.*

Proof. Let A by a PSD matrix and suppose that the i-th diagonal entry $A_{i,i} < 0$. Let e be the vector where i-th entry is 1 and the remaining entries are 0. Then $e^\top A e = A_{i,i} < 0$, contradicting that A is PSD. □

Therefore, any linear program can be phrased as a semidefinite program with at most a quadratic blowup in the number of variables.

3.1.1 The Ellipsoid Method

Unlike LPs, no polynomial-time algorithm is known for solving general SDPs exactly. Furthermore, there are examples of small SDPs for which every solution requires an exponential number of bits to express [86, 128]. However, if we can tolerate certain structural restrictions on the spectahedron, as well as a small additive error, techniques for solving LPs, such as the ellipsoid method and interior point methods, can be adapted to efficiently solve SDPs. We will briefly describe the ellipsoid method and what is needed to adapt it to SDPs. For a more rigorous treatment, as well as proofs of the theorems in this section, we recommend the excellent book by Grötschel et al. [70].

The ellipsoid method was originally developed by Shor [144], and Iudin and Nemirovskii [80] as a method for solving non-linear non-differentiable optimization problems. Khachiyan [85] first observed that the ellipsoid method could be adapted to solve linear programs in polynomial time. In full generality the ellipsoid method can be applied to any convex set that admits an efficient *separation oracle*.

Definition 3.6 (Efficient Separation Oracle)**.** For a convex set $\mathcal{S} \subseteq \mathbb{R}^n$ an efficient separation oracle for $\mathcal{S}$ is a polynomial-time in n and size($\mathcal{S}$) procedure which, given $\alpha \in \mathbb{R}^n$, either returns that $\alpha \in \mathcal{S}$, or if $\alpha \notin \mathcal{S}$ returns a separating hyperplane $a^\top x \geqslant b$ such that for all $\beta \in \mathcal{S}$, $a^\top \beta \geqslant b$ and $a^\top \alpha < b$.

Here, size$(\cdot)$ gives the bit complexity to specify its argument.

Feasibility. Before describing how the ellipsoid method can be used to solve optimization problems, we focus on the simpler *feasiblility problem*: Given a convex set $\mathcal{S}$, find a point $\alpha \in \mathcal{S}$ or decide that $\mathcal{S}$ is empty.

At a high level, the ellipsoid method begins with some initial ellipsoid that contains $\mathcal{S}$. The volume of this ellipsoid is repeatedly shrunk until the center of the ellipsoid lands within $\mathcal{S}$, or until we have exceeded a specified bound on the number of iterations. At every iteration in which the center of the current ellipsoid falls outside of $\mathcal{S}$, the ellipsoid is refined by querying the separation oracle to obtain a hyperplane separating the center from $\mathcal{S}$. An ellipsoid with smaller volume is then constructed, such that it contains the half-ellipsoid, the intersection between an ellipsoid and the separating hyperplane. In fact, (see for example [70]) it is straightforward to explicitly calculate the ellipsoid with the smallest volume that contains a given half-ellipsoid. To formally state the ellipsoid method, let $\mathsf{Ball}(r, c)$ denote a Euclidean ball with radius r and center c.

Algorithm 3.7 (The Ellipsoid Method for Feasibility)**.**

Given: A convex set $\mathcal{S}$ with separation oracle $\mathcal{O}$. A number $num \in \mathbb{N}$ of iterations.

Output: A point $x^* \in \mathcal{S}$ if one exists.

Operation:

1. Let $\mathcal{E} = \mathsf{Ball}(0, R)$ for some $R \in \mathbb{R}$ such that $\mathcal{S} \subseteq \mathcal{E}$.
2. **Repeat** *num* times**:**
   (a) Let $x^* \in \mathbb{R}^n$ be the center of $\mathcal{E}$.
   (b) Query $\mathcal{O}$ whether $x^* \in \mathcal{S}$.
   (c) If $x^* \in \mathcal{S}$ then halt and return x^*, otherwise continue.
   (d) Let $a^\top x^* \geqslant b$ be the separating hyperplane obtained from $\mathcal{O}$ such that $a^\top x^* < b$ and $\mathcal{S} \subseteq \mathcal{E} \cap \{a^\top x \geqslant b\}$.
   (e) Set $\mathcal{E} \leftarrow \mathcal{E} \cap \{a^\top x \geqslant b\}$.
3. Return *null*.

The ellipsoid method constructs a sequence of ellipsoids $\mathcal{E}^{(0)}, \mathcal{E}^{(1)}, \mathcal{E}^{(2)}, \ldots$ with decreasing volume, where $\mathcal{E}^{(t)}$ is the ellipsoid in the t-th repetition. The key to its efficiency is that the

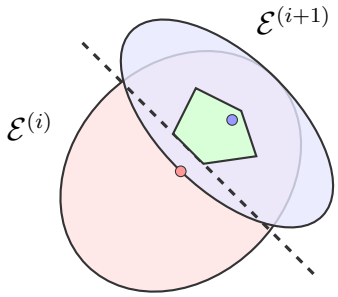


Figure 3.2: A single step of the ellipsoid method.

decrease in volume between each successive ellipsoid is sufficiently large. In particular, we can guarantee the following relationship between the volume of successive ellipsoids:

$$\frac{Vol(\mathcal{E}^{(t+1)})}{Vol(\mathcal{E}^{(t)})} \leqslant \exp\left(-\frac{1}{2(n+1)}\right).$$

Therefore, the runtime of the ellipsoid algorithm is proportional to the structure of the feasible set $\mathcal{S}$. Indeed, if the set $\mathcal{S}$ is not dense within the original ellipsoid $\mathcal{E}^{(0)} := \mathsf{Ball}(0, R)$, or if R is too large, then the algorithm may take a long time to converge. In a landmark work, Grőtschel, Lovász and Schrijver [69] showed that the ellipsoid method solves the feasibility problem in polynomial time, provided $\mathcal{S}$ is not too small within $\mathcal{E}_0$ and R is not too large, thus giving a sufficient bound on *num*.

Theorem 3.8 (Grőtschel, Lovász, Schrijver [69])**.** *Let $\mathcal{S}$ be a convex set with efficient separation oracle $\mathcal{O}$. Let $R, r > 0$ be such that $\mathsf{Ball}(c, r) \subseteq \mathcal{S} \subseteq \mathsf{Ball}(0, R) = \mathcal{E}^{(0)}$ for some $c \in \mathbb{R}^n$. Then, the ellipsoid method solves the feasibility problem in time $\mathsf{poly}(n, \mathrm{size}(\mathcal{S})) \log(R/r)$.*

Note that requiring that $\mathcal{S} \subseteq \mathsf{Ball}(0, R)$ ensures that the set of solutions in $\mathcal{S}$ have bit-length bounded by $\log(R)$.

Optimization. It is straightforward to adapt the algorithm for feasibility to one for optimization, provided that we tolerate a small additive error $\varepsilon > 0$. In particular, suppose that we are trying to optimize $c^\top x$ over a convex set $\mathcal{S}$. We will show how the ellipsoid method can be used to obtain an ε-approximate optimal solution α^*, a solution satisfying $c^\top \alpha^* \geqslant \max\{c^\top x : x \in \mathcal{S}\} - \varepsilon$. The high-level idea is to reduce optimization to feasibility by repeatedly asking for a better solution. This will be done by iterating the following subroutine: First, the ellipsoid method will be run to find a feasible point $\alpha \in \mathcal{S}$. Then, the constraint $c^\top x > c^\top \alpha$ will be added to $\mathcal{S}$, requiring that the next solution found by the ellipsoid method must be an improvement on the solutions that were previously found. The ellipsoid method will then be re-run. If the current convex set becomes empty, then the previous solution we found must have been the optimal solution. Unfortunately, this process may take prohibitively long to converge to the true optimum. However, if we allow for a small ε-error, then this method can be seen to converge in polynomial-time. This technique

is known as the *sliding objective function method* and was first introduced by Shor [144] and Iudin and Nemirovskii [80].

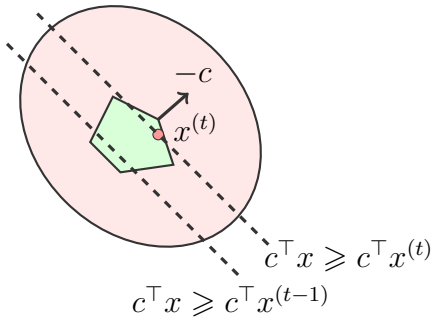


Figure 3.3: A single step of the sliding objective function method.

One issue that must be addressed is that the original separation oracle $\mathcal{O}$ for $\mathcal{S}$ may no longer be valid once we introduce the constraint $c^\top x \geqslant c^\top \alpha$. Luckily, it is straightforward to construct an oracle $\mathcal{O}'$ for $\mathcal{S} \cap \{c^\top x \geqslant c^\top \alpha\}$ from $\mathcal{O}$. For a point $\beta \in \mathbb{R}^n$, $\mathcal{O}'$ asks $\mathcal{O}$ whether $\beta \in \mathcal{S}$, then

1. If $\beta \notin \mathcal{S}$, then $\beta \notin \mathcal{S} \cap \{c^\top x \geqslant c^\top \alpha\}$ and the separating hyperplane provided by $\mathcal{O}$ is a valid separating hyperplane for $\mathcal{S} \cap \{c^\top x \geqslant c^\top \alpha\}$.
2. If $\beta \in \mathcal{S}$, if $c^\top \beta \geqslant c^\top \alpha$, return β. Otherwise, $c^\top x \geqslant c^\top \alpha$ is a separating hyperplane.

We will denote the separation oracle for the set $\mathcal{S} \cap \{c^\top x \geqslant c^\top \alpha\}$ by $\mathcal{O} \cap \{c^\top x \geqslant c^\top \alpha\}$.

Algorithm 3.9 (The Ellipsoid Method for Optimization)**.**

Given: A convex set $\mathcal{S}$ with separation oracle $\mathcal{O}$, and an objective function $c^\top x$. A precision parameter $\varepsilon > 0$. A number of iterations $num \in \mathbb{N}$.

Output: A solution $\alpha^* \in \mathcal{S}$.

Operation:

1. Let $\mathcal{E} = \mathsf{Ball}(0, R)$ for some $R \in \mathbb{R}$ such that $\mathcal{S} \subseteq \mathcal{E}$, and let $\alpha^* = null$.
2. **Repeat** num times**:**
   (a) Let $x^* \in \mathbb{R}^n$ be the center of $\mathcal{E}$.
   (b) Query $\mathcal{O}$ whether $x^* \in \mathcal{S}$.
   (c) If $x^* \in \mathcal{S}$ then
      i. Set $\alpha^* \leftarrow x^*$.
      ii. Set $\mathcal{S} \leftarrow \mathcal{S} \cap \{c^\top x \geqslant c^\top \alpha^*\}$ and $\mathcal{O} \leftarrow \mathcal{O} \cap \{c^\top x \geqslant c \top \alpha^*\}$.

(d) Otherwise

i. Let $a^\top x \geqslant b$ be the separating hyperplane obtained from $\mathcal{O}$ such that $a^\top x^* < b$ and $\mathcal{S} \subseteq \mathcal{E} \cap \{a^\top x \geqslant b\}$.

ii. Set $\mathcal{E}$ to be the smallest ellipsoid containing the half-ellipsoid $\mathcal{E} \cap \{a^\top x \geqslant b\}$.

3. Return α^*.

With these modifications the ellipsoid method can be used to obtain an ε-approximate optimal solution in time polynomial in r, R, and $\ln(1/\varepsilon)$ (see, for example, Exercise 8.4. in [33]), giving us a sufficient bound for *num*.

Theorem 3.10 (Khachiyan [85])**.** *Let $\mathcal{S}$ be a convex set with efficient separation oracle $\mathcal{O}$, objective function $c^\top x$, and precision parameter $\varepsilon > 0$. Let $R, r > 0$ be such that there exists a $c \in \mathbb{R}$ such that $\mathsf{Ball}(c, r) \subseteq \mathcal{S} \subseteq \mathsf{Ball}(0, R)$. Then the ellipsoid method will either output an ε-approximate optimal solution, or determine that $\mathcal{S} = \emptyset$ in time $\mathsf{poly}(n, \text{size}(\mathcal{S})) \log(R/r\varepsilon)$.*

Solving Semi-Definite Programs. In order to solve SDPs using the ellipsoid method we must show that SDPs admit efficient separation oracles. The canonical separation oracle for a linear program $\{a_i^\top x \geqslant b_i : i \in [m]\}$ is straightforward: simply determine whether the given point α satisfies each of the constraints $a_i^\top x \geqslant b_i$. This can be done in polynomial time in the bit complexity of a_i and b_i. If α falsifies any of these constraints then that constraint constitutes a valid separating hyperplane. To adapt this to an efficient separation oracle for SDPs, it remains to show that we can efficiently test whether $\alpha \succeq 0$, and if not, obtain from it a separating hyperplane.

Lemma 3.11 (Separation Oracle for SDPs)**.** *Given the constraints of an SDP $\mathcal{S} = \{A_1 \bullet X = b_1, \ldots, A_m \bullet X = b_m, X \succeq 0\}$, there exists a separation oracle for $\mathcal{S}$ running in time $\mathsf{poly}(m, n, \text{size}(A_i), \text{size}(b_i))$ (for $i \in [m]$).*

In order to prove this lemma we will require some additional properties of semidefinite matrices which we will develop in the following section. Because of this, we will defer the construction of the separation oracle until Section 3.1.2.1.

Provided that a separation oracle exists, the only remaining issue preventing us from solving an SDP with the ellipsoid method is that we must ensure that our convex set $\mathcal{S} = \{A_1 \bullet X = b_1, \ldots, A_m \bullet X = b_m,\ X \succeq 0\}$ contains a full-dimensional ball of some non-zero radius r within it. Observe that this is almost always false whenever $m > 0$ as the constraints $A_i \bullet X = b_i$ may restrict our feasible set to a subspace. In order to apply the ellipsoid method, we thus relax the constraints $A_i \bullet X = b_i$ to $|A_i \bullet X - b_i| < \varepsilon$ for small error parameter $\varepsilon > 0$. It is not too hard to show now that there exists a ball of radius $\Omega(\varepsilon^n / \|A\|_F^n)$ inside our convex set, where $A = A_1 \circ A_2 \circ \ldots \circ A_m$ is the concatenation of the matrices A_i, and $\|\cdot\|_F$ is the Frobenius norm.

Finally, notice that for small enough δ, δI is feasible for our relaxed SDP above. Thus, without hurting the optimum of our relaxed SDP, we can upper bound $\|X^*\|_F^2 \leqslant \delta^2 n$. for any feasible solution $X^* \in \mathbb{R}^{n \times n}$. That gives an upper bound of n on the radius of a ball enclosing our convex set.

We can thus apply ellipsoid method to obtain the following result:

Corollary 3.12 (Solving SDPs in Polynomial Time)**.** *Given $C \in \mathbb{R}^{n\times n}$, $A_1, \ldots, A_m \in \mathbb{R}^{n\times n}$, and $b \in \mathbb{R}^n$ and any $\varepsilon > 0$, there is an algorithm running in time $\mathsf{poly}(m, n, \mathrm{size}(A_i), \mathrm{size}(C), \mathrm{size}(b_i), \log(1/\varepsilon))$ (for all i) that finds X that maximizes $C \bullet X$ satisfying $X \succeq 0$ and $A_i \bullet X = b_i$ for every $i \in [m]$, up to an additive error ε.*

3.1.2 Positive Semidefinite Matrices

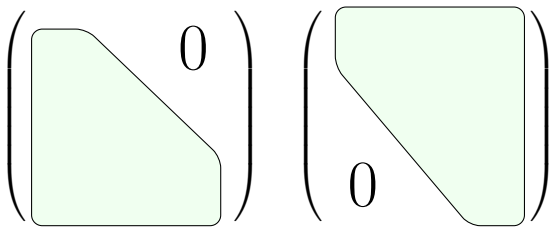
This section is dedicated to reviewing several properties and characterizations of PSD matrices which will be necessary for later sections. We will use the task of constructing a separation oracle for SDPs as a motivating example. The main technical challenge in constructing such an oracle is designing a method to efficiently test whether a matrix A is PSD; of course, it is computationally infeasible to test whether $u^\top Au \geqslant 0$ for every $u \in \mathbb{R}^n$ directly. Our separating oracle will instead rely on the following equivalent definition of a PSD matrix.

Theorem 3.13. *A matrix $A \in \mathbb{R}^{n\times n}$ is a symmetric PSD if and only if its n eigenvalues are non-negative.*

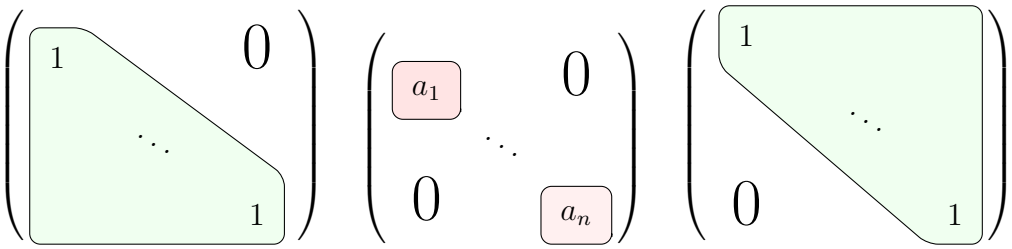
The proof of this theorem requires several decompositions of PSD matrices which will be crucial when discussing Sum-of-Squares. Therefore we defer the proof of Theorem 3.13 until these have been developed.

Theorem 3.14. *Let A be an $n \times n$ symmetric PSD matrix, then A has the following equivalent decompositions.*

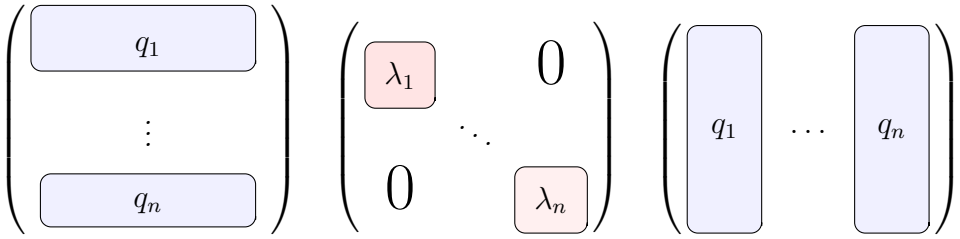
- ***Cholesky Decomposition:*** *$A = U^\top U$, where U is the* unique *upper-triangular matrix.*



- ***LDL Decomposition:*** *$A = LDL^\top$, where L is a lower-triangular matrix whose diagonal is all 1s, and D is a diagonal matrix whose entries are non-negative. Furthermore, both D and L are* unique*.*



- ***Non-Negative Spectral Value Decomposition:*** *$A = Q\Lambda Q^\top$, where Q is an orthogonal matrix (i.e. the rows q_i of Q are orthonormal vectors) and Λ is a non-negative diagonal matrix whose entries are the eigenvalues of A. This decomposition is not unique.*



Proof. We begin by proving the LDL decomposition by induction on n. Suppose that the LDL decomposition holds for all symmetric PSD matrices of dimension $(n-1) \times (n-1)$ and let $A \in \mathbb{R}^{n \times n}$ be a symmetric PSD matrix. Because A is symmetric it has the following form:

$$A = \begin{pmatrix} b & v^\top \\ v & C \end{pmatrix}$$

where $b \in \mathbb{R}$ is some constant, v is a vector of length $(n-1)$ and C is an $(n-1) \times (n-1)$ submatrix. Note that, by Claim 3.5, we know that $b \geqslant 0$. We can decompose A as follows,

$$\begin{bmatrix} 1 & \vec{0} \\ \frac{v}{b} & I \end{bmatrix} \begin{bmatrix} b & \vec{0} \\ \vec{0} & C - \frac{vv^\top}{b} \end{bmatrix} \begin{bmatrix} 1 & \frac{v}{b} \\ \vec{0} & I \end{bmatrix},$$

where I is the $(n-1) \times (n-1)$ identity matrix. Finally, we need to show that the matrix $B := C - \frac{vv^\top}{b}$ is symmetric PSD. Let $u \in \mathbb{R}^{n-1}$ be any vector, and define the vector

$$x^\top = \left[-\frac{u^\top v}{b}, u^\top \right].$$

Multiplying B by u, we have

$$u^\top B u = x^\top A x \geqslant 0$$

this follows because $-\frac{u^\top v}{b}$ has the effect of zero-ing out the first row and the first column of A.

The Cholesky decomposition is a straightforward consequence of the LDL decomposition. Consider the LDL decomposition of a symmetric PSD matrix,

$$A = L^\top D L.$$

Using that D is diagonal, we can write

$$(L^\top D^{1/2})(D^{1/2}L) = (LD^{1/2})^\top (LD^{1/2}).$$

Letting $U = (LD^{1/2})$, we have the desired decomposition $A = U^\top U$.

To see that the converse holds as well, we will show that if A has an LDL decomposition then it is PSD. Suppose that A has an LDL decomposition, then by what we proved above it has a Cholesky decomposition $A = U^\top U$ as well. Let $u \in \mathbb{R}^n$ be any vector and observe that

$$u^\top Au = u^\top U^\top Uu = (u^\top U^\top)^2 \geqslant 0.$$

Therefore A is PSD.

Finally, we argue that a matrix is PSD if and only if it has a spectral value decomposition in which the matrix Λ is non-negative. First, we will prove that every symmetric matrix has a (not necessarily non-negative) spectral value decomposition.

Lemma 3.15 (Spectral Value Decomposition)**.** *Every symmetric matrix A can be written as $A = Q\Lambda Q^\top$, where Q is an orthogonal matrix and Λ is a (not-necessarily non-negative) diagonal matrix whose entries are the eigenvalues of A.*

Proof. The spectral value decomposition follows from two properties of symmetric matrices:

Claim 3.16. *The eigenvectors corresponding to distinct eigenvalues of a symmetric matrix are orthonormal.*

Proof. To see this, let u and v be eigenvectors of a symmetric matrix A, with distinct eigenvalues λ_u and λ_v respectively. Then,

$$u^\top Av = \lambda_u u^\top v,$$

and

$$v^\top Au = \lambda_v v^\top x = \lambda_v u^\top v,$$

which follow from the symmetry of A. Therefore,

$$\lambda_u u^\top v = \lambda_v u^\top v \implies \lambda_u = \lambda_v,$$

which is a contradiction. Finally, note that each eigenvector can be made unitary by normalizing. □

Claim 3.17. *Every eigenvalue of a real symmetric matrix is real.*

Proof. Let A be a symmetric matrix, and let u be an eigenvector of A with eigenvalue λ_u. Then

$$(Au)^\top Au = u^\top A^\top Au = AuAu = \lambda_u^2 |u|^2,$$

and so

$$\lambda_u^2 = \frac{(Au)^\top Au}{|u|^2} \geqslant 0.$$

Therefore, the eigenvalues of A must be real. □

With these claims in hand, we are ready to derive the spectral value decomposition. Let A be a symmetric matrix, and let $\lambda_1, \ldots, \lambda_k$ be the eigenvalues of A. For each λ_i, let v_i be a corresponding unit eigenvector. By Claims 3.16 and 3.17, these vectors are real and orthonormal. Construct the $n \times n$ matrix Q, where the i-th column of Q is the vector v_i, and the diagonal matrix Λ, where the i-th diagonal entry is λ_i. By Theorem 3.13, we know that the eigenvalues are non-negative. If $k < n$, let $t_{k+1}, \ldots, t_n$ be a completion of $v_1, \ldots, v_k$ to an orthonormal basis for $\mathbb{R}^n$. Form the remaining $n - k$ columns of Q with these vectors, and let the remaining $n - k$ diagonal entries of Λ be 0. Because v_i is an eigenvector, $Av_i = \lambda_i v_i$, and so,

$$AQ = \Lambda Q.$$

Because Q is an orthogonal matrix, $Q^\top Q = 1$, and so $Q^{-1} = Q^\top$. Therefore,

$$A = Q\Lambda Q^\top.$$

□

Finally, the claim that a matrix is PSD if and only if has a Non-Negative spectral value decomposition follows from the spectral value decomposition, along with Theorem 3.13, asserting that a matrix is PSD iff its eigenvalues are non-negative. Therefore, to complete this proof it is enough to prove Theorem 3.13.

Proof. (of Theorem 3.13) Let $A \succeq 0$ and $v \in \mathbb{R}^n$ some vector. By the spectral value decomposition of A we have $A = Q\Lambda Q^\top$, where Λ is a diagonal matrix whose entries are the eigenvalues of A. Then,

$$v^\top A v = v^\top Q \Lambda Q^\top v = \sum_{i=1}^{n} \lambda_i (v^\top Q)_i^2 \geqslant 0,$$

where λ_i is the i-th diagonal entry of Λ, the i-th eigenvalue of A. In particular, this implies $\lambda_i \geqslant 0$ for all $i \in [n]$.

Conversely, let A be a real symmetric matrix whose n eigenvalues are non-negative. The proof proceeds in the same manner as the proof of the spectral value decomposition. By Claims 3.16 and 3.17, the eigenvectors are orthonormal, and eigenvalues are real. To complete the proof, arrange the eigenvalues and eigenvectors to form a spectral value decomposition as we did in the proof of Theorem 3.14, concluding that $A \succeq 0$. □

Theorem 3.13 together with the spectral value decomposition (Lemma 3.15) implies that a matrix is PSD if and only if it has a Non-Negative spectral value decomposition.

□

Although Theorem 3.13, along with the LDL Decomposition is enough to construct a separation oracle for SDPs, we defer its description until the end of this section in order to cover two final characterizations of PSD matrices which will be necessary later in this monograph. These characterizations are in terms of how PSD matrices interact with each

other; the first of which describes PSD matrices in terms of their inner product[1] with other PSD matrices.

Lemma 3.18. *A symmetric matrix $A \in \mathbb{R}^{n\times n}$ is PSD if and only if $A \bullet B \geqslant 0$ for all $B \succeq 0$ with $B \in \mathbb{R}^{n\times n}$.*

The proof of this lemma will rely on several properties of the matrix trace, where the *trace* of an $n \times n$ matrix A is the sum of its diagonal entries,

$$\mathsf{Tr}[A] = \sum_{i=1}^{n} A_{i,i}$$

Fact 3.19. *For $A, B \in \mathbb{R}^{n\times n}$ the following identities hold:*

1. $\mathsf{Tr}[AB] = \mathsf{Tr}[BA]$,
2. $A \bullet B = \mathsf{Tr}[A^\top B]$.

Furthermore, if A and B are symmetric, then $\mathsf{Tr}[A^\top B] = \mathsf{Tr}[AB]$.

Proof. (Of Lemma 3.18) Let $A, B \succeq 0$. Applying the spectral value decomposition to A, we can write $A = Q\Lambda Q^\top$. Then,

$$A \bullet B = \mathsf{Tr}[AB] = \mathsf{Tr}[Q\Lambda Q^\top B] = \mathsf{Tr}[\Lambda BQQ^\top],$$

which follows from applying Fact 3.19. Because Q is an orthogonal matrix, $QQ^\top = I$. Then,

$$\mathsf{Tr}[\Lambda B] = \sum_{i=1}^{n} \Lambda_{i,i} B_{i,i} \geqslant 0, \tag{3.1}$$

which follows because $\Lambda_{i,i}$ are the eigenvalues of A and therefore by Theorem 3.13, they are non-negative. As well, the diagonal entries $B_{i,i}$ are non-negative, because $B \succeq 0$.

For the other direction, suppose that $A \bullet B \geqslant 0$ for every $B \succeq 0$, but A is not PSD. That is, there exists $u \in \mathbb{R}^n$ such that $u^\top Au < 0$. Let $B = uu^\top$, and observe that $B \succeq 0$; indeed, for any $v \in \mathbb{R}^n$,

$$v^\top uu^\top v = (v^\top u)^2 \geqslant 0.$$

Then,

$$u^\top Au = \sum_{i,j\in[n]} u_i A_{i,j} u_j = \sum_{i,j\in[n]} A_{i,j}(u_i u_j) = A \bullet B < 0,$$

contradicting our assumption. □

An immediate corollary is a similar characterization for symmetric positive definite matrices which will be useful in the following section.

[1]Recall that the inner product $A \bullet B$ between matrices A and B is defined as $\sum_{i,j} A_{i,j}B_{i,j}$.

Corollary 3.20. *For any matrix $A \succ 0$, $A \bullet B > 0$ for every $B \succeq 0$, $B \neq 0$.*

This follows directly from the proof of Lemma 3.18 along with the fact that positive definite matrices admit a spectral value decomposition such that the n eigenvalues are all positive. Using this fact, Equation 3.1 holds with a strict inequality when A is a positive-definite matrix.

Our final characterization, known as *Sylvester's Criterion*, characterizes SDP matrices in terms of their submatrices. Before stating it, it will be useful to first understand under what conditions we can guarantee that a submatrix of a symmetric PSD matrix is itself symmetric PSD; this is a natural question in its own right. Immediately we can rule out non-square submatrices and submatrices formed by deleting any subset of rows and columns, as neither of these are guaranteed to be symmetric. One natural class of submatrices which preserve symmetry are *principal submatrices*, formed by deleting the same set I of rows and columns of the original matrix.

Lemma 3.21. *The principal submatrices of any symmetric PSD matrix are themselves symmetric PSD.*

Proof. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric PSD matrix. Let $I \subseteq [n]$ and let B be a principal submatrix obtained by deleting the rows and columns of A with indices in I. That B is symmetric follows from the symmetry of A. Finally, we want to show for any $u \in \mathbb{R}^{n-|I|}$, that $u^\top B u \geqslant 0$. Let $v \in \mathbb{R}^n$ be u extended to assign 0 to all coordinates in $[n] \setminus I$. Then,

$$u^\top B u = v^\top A v \geqslant 0.$$

□

Sylvester's Criterion characterization PSD matrices by the determinants of their submatrices.

Lemma 3.22 (Sylvester's Criterion)**.** *Let A be a symmetric matrix over $\mathbb{R}$. Then, $A \succeq 0$ if and only if the determinant of every principal submatrix is non-negative.*

We will only prove the forward direction of Sylvester's criterion. The converse is significantly more involved and will not be needed.

Proof. Suppose that $A \succeq 0$. By Lemma 3.21, every principal submatrix A' of A is a symmetric PSD matrix. Therefore, by the Cholesky Decomposition, we can write $A' = U^\top U$, where U is upper triangular. Then,

$$\det(A') = \det(U^\top U) = \det(U^\top)\det(U) = (\det(U))^2 \geqslant 0.$$

□

3.1.2.1 A Separation Oracle for SDPs

We end this section with the construction of an efficient separation oracle for SDPs, completing the proof of Lemma 3.11. Recall that our task is to test whether a given solution $\alpha \in \mathbb{R}^{n \times n}$ satisfies the constraints of our SDP, and if not, construct a separating hyperplane. The main challenge is to handle the constraint $X \succeq 0$.

Lemma 3.23. *There exists an efficient algorithm that tests whether $\alpha \succeq 0$, and if not, outputs $c \in \mathbb{R}^n$ such that $c^\top \alpha c < 0$.*

Proof. We can assume without loss of generality that α is symmetric by operating only over $\lceil (n+1)/2 \rceil$ variables, using the same variable for both $X_{i,j}$ and $X_{j,i}$. By the spectral value decomposition (Lemma 3.15) we can $\alpha = Q \Lambda Q^\top$ where Λ is a diagonal matrix whose entries are the eigenvalues of α, and the rows of Q are the corresponding eigenvectors. This decomposition can be computed in polynomial time using standard algorithms for obtaining eigenvalue decompositions. By Theorem 3.13, the eigenvalues of α are non-negative if and only if α is PSD. This provides an easy method for testing whether α is PSD: Let e_i be the i-th standard basis vector. For $i = 1, \ldots, n$, test whether $e_i^\top \Lambda e_i = \Lambda_{i,i} \geqslant 0$. If this test fails then Λ has a negative entry and we construct a separating hyperplane using Q as follows. Suppose that $\Lambda_{i,i} < 0$. The vector $c = Q^{-1} e_i$ witnesses that α is not PSD. Indeed,

$$c^\top \alpha c = e_i^\top (Q^{-1})^\top \alpha Q^{-1} e_i = e_i^\top \Lambda_{i,i} e_i < 0.$$

□

Lemma 3.23 gives us a polynomial-time test for PSD-ness. Furthermore, if α is not PSD then the vector c output by the algorithm provides a separating hyperplane: Let $H = (cc^\top)$, then

$$H \bullet \alpha = cc^\top \alpha = c^\top \alpha c < 0.$$

As well, for any $\beta \in \mathcal{S}$, because $\beta \succeq 0$, it holds that $c^\top \beta c \geqslant 0$. Thus $H \bullet X \geqslant 0$ is a separating hyperplane for α.

We can use this separation oracle along with Theorem 3.8 to approximately solve SDPs in polynomial time. We sketch the details and then note this consequence as a corollary here. Recall that a generic SDP minimize $C \bullet X$ subject to $\mathcal{S} = \{A_1 \bullet X = b_1, \ldots, A_m \bullet X = b_m,\ X \succeq 0\}$. Here, X, A_i, and C are $n \times n$ matrices. We assume that A_i, C, and b consist of real entries specified by polynomially many bits. To apply the ellipsoid method, we need an efficient separation oracle for the convex set consisting of $A_i \bullet X = b_i$ and $X \succeq 0$. Given any $\alpha \in \mathbb{R}^{n \times n}$, we can check if $A_i \bullet X = b_i$ by solving linear inequalities

$$A_i \bullet \alpha \leqslant b_i \quad \text{and} \quad A_i \bullet \alpha \geqslant b_i$$

in polynomial time. Notice that this requires that A be specified by polynomially many bits. If either of these fails, then the falsified halfspace can be used as a separating hyperplane. Next, we can check $X \succeq 0$ by applying the separation oracle above. Thus, altogether, we have a polynomial time separation oracle for testing feasibility in our convex set. This completes the proof of Lemma 3.11.

3.1.3 Semidefinite Programming Duality

Until now, we have been working with the *primal* form of semidefinite programs $SDP(\mathcal{P}, C)$, where $\mathcal{P} = \{A_1 \bullet X = b_1, \ldots, A_m \bullet X = b_m,\ X \succeq 0\}$ and $C, A_i \in \mathbb{R}^{n \times n}$, and $b_i \in \mathbb{R}$. Analogous to linear programming, every SDP admits a *dual*, defined as

Primal:	Dual:
$\min_X \quad C \bullet X$	$\max_y \quad b^\top y$
$A_i \bullet X = b_i \qquad \forall i \in [m]$	$C - \sum_{i \in [m]} y_i A_i \succeq 0$
$X \succeq 0$	

The dual SDP can be written in the standard form of the primal, as a single SDP constraint along with a set of linear equality constraints. To see this, introduce an $n \times n$ matrix of additional slack variables S. The dual can then be rewritten as

$$\mathcal{SDP}^D(\mathcal{S}^D, b) := \max_{y \in \mathcal{S}^D} b^\top y,$$
$$\text{where } \mathcal{S}^D = \left\{ C - \sum_{i=1}^{m} y_i A_i = S, \quad S \succeq 0 \right\}.$$

As in the case of LPs, the optimal solutions to the primal and dual are closely related. Any solution to the dual SDP is an lower bound on the minimum value that the primal can attain.

Theorem 3.24 (Weak Duality for Semidefinite Programs)**.** *If α is any feasible solution to the primal $\mathcal{SDP}(\mathcal{S}, C)$ and β is any feasible solution to the dual $\mathcal{SDP}^D(\mathcal{S}^D, b)$, then*

$$C \bullet \alpha \geqslant b^\top \beta.$$

Proof. Let α be a feasible solution to $\mathcal{SDP}(\mathcal{S}, C)$ and β be a feasible solution to the dual $\mathcal{SDP}^D(\mathcal{S}^D, b)$. Then,

$$\begin{aligned} C \bullet \alpha &= \left(\sum_{i=1}^{m} \beta_i A_i + S \right) \bullet \alpha, && (C - \textstyle\sum_{i=1}^{m} y_i A_i = S) \\ &= \sum_{i=1}^{m} \beta_i (A_i \bullet \alpha) + (S \bullet \alpha), \\ &= \sum_{i=1}^{m} \beta_i b_i + (S \bullet \alpha). && (A_i \bullet X = b_i) \end{aligned}$$

Since $S, \alpha \succeq 0$, by Lemma 3.18 we have $S \bullet \alpha \geqslant 0$. Therefore,

$$C \bullet \alpha - b^\top \beta = S \bullet \alpha \geqslant 0,$$

and so $C \bullet \alpha \geqslant b^\top \beta$. □

This duality theorem is considerably weaker than the duality theorem for linear programs (Theorem 2.1). Unlike for linear programming, we cannot ensure that if both the primal and dual SDPs are feasible then there will be no duality gap, (i.e. that their optima coincide). Such a *strong duality* theorem does not hold for semidefinite programs. The following counter example is due to Lovász [109].

Lemma 3.25. *There exists an SDP such that both the primal and dual have feasible solutions, but their optima do not coincide.*

Proof. Consider the following SDP,

$$\begin{aligned} \min \quad & y_1, \\ \text{s.t.} \quad & \begin{pmatrix} 0 & y_1 & 0 \\ y_1 & y_2 & 0 \\ 0 & 0 & y_1 + 1 \end{pmatrix} \succeq 0. \end{aligned} \tag{3.2}$$

Note that this can be transformed into an SDP in the standard primal form by introducing slack variables. Since the $(1,1)$-entry of the matrix is 0, the semidefinite constraint forces $y_1 = 0$. To see this, suppose that $y_1 \neq 0$, let M be the matrix in (3.2), and let $u \in \mathbb{R}^3$ be the vector $u = [y_2/y_1, -1, 0]$. Then,

$$u^\top M u < 0.$$

Because the diagonal must be non-negative, the feasible solutions satisfy $(y_1 = 0, y_2 \geqslant 0)$. Therefore, the optimal solution of the primal is 0.

To find the dual, we rewrite the constraint of the primal SDP (3.2) as

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + y_1 \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + y_2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \succeq 0.$$

The dual SDP becomes

$$\begin{aligned} \max \quad & -x_{3,3} \\ \text{s.t.} \quad & x_{1,2} + x_{2,1} + x_{3,3} = 1, \\ & x_{2,2} = 0, \\ & X \succeq 0. \end{aligned}$$

As was the case for the primal SDP, the PSD constraint on X along with $x_{2,2} = 0$ forces $x_{1,2} = x_{2,1} = 0$. Therefore, any feasible solution must have $x_{3,3} = 1$. In particular, this implies that the optimal value of the SDP is -1. This is an SDP with a gap of 1 between the optimal value of the primal and dual. □

Although it does not hold in general, it turns out that strong duality can be made to hold if we impose certain robustness conditions on the SDP. The standard sufficient conditions are known as *Slater's Conditions* and are conditions (1) and (2) in the following theorem.

Theorem 3.26 (Strong Duality for Semidefinite Programs)**.** *Let $\mathcal{SDP}(\mathcal{S}, C)$ be an SDP and $\mathcal{SDP}^D(\mathcal{S}^D, b)$ be its dual. Let $\mathcal{S} = \{A_1 \bullet X = b_1, \ldots, A_m \bullet X = b_m,\ X \succeq 0\}$, and so $\mathcal{S}^D = \{C - \sum_{i\in[m]} y_i A_i \succeq 0\}$. Then both $\mathcal{SDP}(\mathcal{S}, C)$, and $\mathcal{SDP}^D(\mathcal{S}^D, b)$ have optima α^* and β^* such that*

$$C \bullet \alpha^* = b^\top \beta^*$$

if either

1. *The spectahedron $\mathcal{S} \neq \emptyset$ and there exists β such that $\sum_{i\in[m]} \beta_i A_i - C \succ 0$, or*
2. *The spectahedron $\mathcal{S}^D \neq \emptyset$ and there exists $\alpha \succ 0$ such that $A \bullet \alpha = b_i$ for all $i \in [m]$.*

We will prove this theorem in the following section. For now, we state a corollary that will be useful when working over bounded domains; a proof can be found in [147, 127]. Here, we interpret $\mathcal{S}$ both as a set of constraints and as the set of all points satisfying the constraints in $\mathcal{S}$.

Corollary 3.27. *Both $\mathcal{SDP}(\mathcal{S}, C)$ and $\mathcal{SDP}^D(\mathcal{S}^D, b)$ have optima α^* and β^* such that $C \bullet \alpha^* = b^\top \beta^*$, if either*

1. *The set of optimal solutions for $\mathcal{SDP}(\mathcal{S}, C)$ is non-empty and bounded, or*
2. *The set of optimal solutions for $\mathcal{SDP}^D(\mathcal{S}^D, b)$ is non-empty and bounded.*

3.1.3.1 A Proof of Strong Duality

The key ingredient in the proof of Theorem 3.26 is an extension of Farkas' Lemma to semidefinite programs. This extension follows from the *Hyperplane Separation Theorem* due to Hermann Minkowski.

Theorem 3.28 (Hyperplane Separation Theorem)**.** *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{R}^n$ be two disjoint non-empty convex sets. There exists $v, t \in \mathbb{R}^n$, $v \neq 0$ such that for every $\alpha \in \mathcal{C}_1$ and $\beta \in \mathcal{C}_2$,*

$$v^\top \alpha \geqslant t \quad \text{and} \quad v^\top \beta \leqslant t.$$

Furthermore, if either $\mathcal{C}_1$ or $\mathcal{C}_2$ is a cone, the theorem holds for $t = 0$.

Using the hyperplane separation theorem one can prove a version of Farkas' Lemma (Lemma 2.3) for SDPs, which will simplify the proof of the strong duality theorem considerably.

Lemma 3.29 (Farkas' Lemma for SDPs)**.** *For symmetric matrices $A_1, \ldots, A_m \in \mathbb{R}^{n\times n}$, the constraint $\sum_{i=1}^m y_i A_i \succ 0$ has no feasible solution if and only if there exists $X \succeq 0$, with $X \neq 0$ such that*

$$A_i \bullet X = 0 \quad \forall i \in [m].$$

Notice that this lemma characterizes positive-definite systems that have no feasible solutions, it says nothing about positive semi-definite systems. That this lemma does not hold for PSD systems follows from the fact that strong duality does not hold for general SDPs. To prove Lemma 3.29, we follow the standard proof of Lovasz [107].

Proof. If we suppose that $\sum_{i=1}^{m} y_i A_i \succ 0$ is infeasible then, viewing each matrix as a vector in $\mathbb{R}^{n^2}$, the cone of positive-definite matrices $\{Y : Y \succ 0\}$ is disjoint from $\{\sum_{i=1}^{m} y_i A_i : y_i \in \mathbb{R}\}$. By the Hyperplane Separation Theorem there exists X such that for every symmetric PSD matrix Y, $Y \bullet X \geqslant 0$ and $\sum_{i=1}^{m} y_i A_i \bullet X \leqslant 0$ for all $y_1, \ldots, y_m \in \mathbb{R}$. Suppose that $A_i \bullet X \neq 0$ for some fixed $i \in [m]$. Consider the vectors $y^{(+)}, y^{(-)} \in \mathbb{R}^m$, defined as $y_i^{(+)} = 1$, $y_i^{(-)} = -1$ and $y_j^{(+)} = 0 = y_j^{(-)}$ for all $j \neq i$. Then,

$$\sum_{j=1}^{m} y_j^{(+)} A_j \bullet X = y_i^{(+)} A_i \bullet X = A_i \bullet X, \qquad \text{and} \qquad \sum_{j=1}^{m} y_j^{(-)} A_j \bullet X = y_i^{(-)} A_i \bullet X = -A_i \bullet X.$$

The only solution satisfying both constraints is $A_i \bullet X = 0$. All that remains is to show that $X \succeq 0$. This follows from the fact that $Y \bullet X \succeq 0$ for every $Y \succeq 0$ and Lemma 3.18.

For the other direction, suppose that there there exists $X \succeq 0$, $X \neq 0$ such that $A_i \bullet X = 0$ for all $i \in [m]$, and furthermore suppose that there is some $y \in \mathbb{R}^m$ such that $\sum_{i=1}^{m} y_i A_i \succ 0$. By Lemma 3.20, because $X \succeq 0$ and $\sum_{i=1}^{m} y_i A_i \succ 0$, we have

$$\left(\sum_{i=1}^{m} y_i A_i \right) \bullet X > 0,$$

contradicting the fact that $A_i \bullet X = 0$ for all $i \in [m]$. □

Furthermore, we can extend this to non-homogeneous systems leading to a derivational version of Farkas' Lemma for SDPs.

Lemma 3.30 (Derivational Farkas' Lemma for SDPs)**.** *For symmetric matrices $A_1, \ldots, A_m, C$, the constraint $\sum_{i=1}^{m} y_i A_i - C \succ 0$ has no feasible solution if and only if there exists $X \succeq 0$, $X \neq 0$ such that*

$$A_i \bullet X = 0 \quad \forall i \in [m], \qquad \textit{and} \qquad C \bullet X \geqslant 0.$$

Proof. Suppose that $\sum_{i=1}^{m} y_i A_i - C \succ 0$ is infeasible, and observe that this constraint is satisfiable if and only if

$$\left\{ \sum_{i=1}^{m} y_i A_i - y_{m+1} C \succ 0, \quad y_{m+1} > 0 \right\} \tag{3.3}$$

is satisfiable. In particular, for any solution $y \in \mathbb{R}^{m+1}$ to (3.3), because $y_{m+1} > 0$, the vector $y' \in \mathbb{R}^m$ defined as $y'_i = y_i / y_{m+1}$ for all $i \in [m]$ is a solution to $\sum_{i=1}^{m} y_i A_i - C \succ 0$. We

can formulate (3.3) as a single positive-definite constraint $\sum_{i=1}^{m+1} y_i A'_i \succ 0$ by defining the matrices

$$A'_1 = \begin{pmatrix} A_1 & 0 \\ 0 & 0 \end{pmatrix}, \ldots, A'_m = \begin{pmatrix} A_m & 0 \\ 0 & 0 \end{pmatrix}, A'_{m+1} = \begin{pmatrix} -C & 0 \\ 0 & 1 \end{pmatrix}.$$

Note that the bottom-right entry in the matrix A'_{m+1}, along with the positive-definiteness constraint force $y_{m+1} > 0$. Because $\sum_{i=1}^{m+1} y_i A'_i \succ 0$ is infeasible, by Lemma 3.29 there exists $X' \succeq 0$, $X' \neq 0$ such that $A'_i \bullet X' = 0$ for all $i \in [m+1]$. If we let X be the upper-left $n \times n$ submatrix of X', then $A_{m+1} \bullet X' = (-C) \bullet X + X'_{n+1,n+1} = 0$, and $A_i \bullet X = 0$ for all $i \in [m]$. We make the following observations:

i) $X \neq 0$: Suppose that $X = 0$. We claim that this implies that $X'_{n+1,n+1} > 0$, contradicting the fact that $(-C) \bullet X + X'_{n+1,n+1} = 0$. Indeed, because $X' \neq 0$, there must exists some $i \in [n+1]$ such that $X'_{i,n+1} = X'_{n+1,i} \neq 0$, where the equality holds by symmetry. Define the vector $v \in \mathbb{R}^{n+1}$ as $v_{n+1} = 1$, $v_i = -sign(X'_{n+1,i})$, and $v_j = 0$ for all $j \neq i$. Then, because $X = 0$, $v^T X' v = -2|X'_{n+1,i}| + X'_{n+1,n+1}$, and furthermore because $X' \succeq 0$, this sum must be non-negative, implying that $X'_{n+1,n+1} \geqslant 2|X'_{n+1,i}| > 0$.

ii) $X \bullet C \geqslant 0$: Since $X \bullet C = X'_{n+1,n+1}$ it suffices to show that that $X'_{n+1,n+1} \geqslant 0$, which follows by Claim 3.5 stating that the diagonal entries of any symmetric PSD matrix are non-negative.

This completes the proof of the forward direction. The proof of the reverse direction is identical to the proof in Lemma 3.29. □

With these lemmas in hand, we are prepared to prove strong duality for SDPs. We will follow the standard proof strong duality due to Lovasz [107].

Proof. (of Theorem 3.26) We will prove (1). Consider $\mathcal{SDP}(\mathcal{S}, C)$, where $\mathcal{S} = \{A_1 \bullet X = b_1, \ldots, A_m \bullet X = b_m,\ X \succeq 0\}$. Suppose that $\mathcal{S} \neq \emptyset$ and that $\sum_{i\in[n]} y_i A_i - C \succ 0$ is feasible[2]. Now, if β^* is the optimal value of the dual SDP $\mathcal{SDP}(\mathcal{S}^D, b)$, then the set of constraints

$$b^\top y > \beta^*,$$
$$C - \sum_{i\in[m]} y_i A_i \succeq 0.$$

is infeasible. We will use this to define a solution to the primal which is at least the value of $b^\top \beta^*$. We can write this as a single positive-definite constraint $C' - \sum_{i\in[m]} y_i A'_i \succeq 0$ by defining the matrices

$$A'_1 = \begin{pmatrix} A_1 & 0 \\ 0 & -b_1 \end{pmatrix}, \ldots A'_m = \begin{pmatrix} A_m & 0 \\ 0 & -b_m \end{pmatrix}, C' = \begin{pmatrix} C & 0 \\ 0 & -\beta^* \end{pmatrix},$$

[2]Recall that we interpret $\mathcal{S}$ both as the set of constraints $\mathcal{S}$ and as the set of feasible solutions to $\mathcal{S}$ using context to differentiate

The infeasibility of $\{b^\top y > \beta^*,\ C - \sum_{i\in[m]} y_i A_i \succeq 0\}$ implies that $C' - \sum_{i\in[m]} y_i A'_i \succ 0$ is infeasible as well. Therefore, applying Lemma 3.30 guarantees the existence of a matrix $X' \succeq 0$, $X' \neq 0$ such that

$$A'_i \bullet X' = 0 \quad \forall i \in [m] \qquad \text{and} \qquad C \bullet X' \geqslant 0.$$

Let X be the upper-left $n \times n$ submatrix of X', then we have

$$A'_i \bullet X' = A_i \bullet X - X_{n+1,n+1} b_i = 0 \quad \forall i \in [m], \qquad \text{and} \qquad C \bullet X - X'_{n+1,n+1}\beta^* \geqslant 0.$$

All that is left is to show that $X'_{n+1,n+1} > 0$, in which case $\frac{1}{X'_{n+1,n+1}} X$ will be a feasible solution to the primal and satisfy $\frac{1}{X_{n+1,n+1}} X \bullet C \geqslant b^\top \beta^*$. First, observe that $X'_{n+1,n+1} \geqslant 0$ because $X \succeq 0$ and the diagonal entries of a PSD matrix are non-negative. Suppose that $X'_{n+1,n+1} = 0$, then

$$A_1 \bullet X = 0, \ldots, A_m \bullet X = 0, \quad X \bullet C = 0, \quad X \succeq 0.$$

has a solution. According to Lemma 3.30, this implies that there is no feasible solution to $\sum_{i=1}^m y_i A_i - C \succ 0$, which contradicts our initial assumption. Therefore, $\frac{1}{X'_{n+1,n+1}} X \bullet C \geqslant b^\top \beta^*$. That $\frac{1}{X'_{n+1,n+1}} X \bullet C \leqslant b^\top \beta^*$ follows from weak duality (Theorem 3.24).

The proof of (2) follows by a similar argument. □

3.1.4 Sum-of-Squares Polynomials and Semidefinite Programs

As an example of the usefulness of SDPs we will discuss how they can help answer the following fundamental questions: How can one determine whether a polynomial $P(x) \in \mathbb{R}[x]$ is non-negative over $\mathbb{R}$? Perhaps the simplest way to witness non-negativity is by showing that the polynomial can be written as a sum-of-squares:

$$P(x) = \sum_{i=1}^m Q_i^2(x),$$

for $Q_1, \ldots, Q_m \in \mathbb{R}[x]$. This gives a concise certificate of the negativity of $P(x)$. Furthermore, such a certificate can be found algorithmically: there exists efficient semidefinite programs for determining whether or not a polynomial has a sum-of-squares factorization. In order to describe this SDP, it will be convenient to define the notion of a monomial vector over the variables x.

Definition 3.31 (monomial vector)**.** A monomial vector of degree d is a $\binom{n}{\leqslant d}$-dimensional vector $v_d(x)$ indexed by sets $S \subseteq n$, $|S| \leqslant d$, where the entry $v_d(x)_S$ is the monomial

$$v_d(x)_S \ := \ \prod_{i\in S} x_i.$$

The following lemma will allow us to phrase the task of finding a sum-of-squares representation as an SDP.

Lemma 3.32. *Let $P(x) \in \mathbb{R}[x]$ be a polynomial of degree at most $2d$ and let $v_d(x)$ be the degree d monomial vector. Then, $P(x)$ can be written as a sum-of-squares if and only if there exists a matrix $C \succeq 0$ such that*

$$P(x) = v_d(x)^\top C v_d(x)$$

In particular, the SDP has constraints $P(x) = v_d(x)^\top C v_d(x)$ and $C \succeq 0$. The variables of the SDP are the entries of the coefficient matrix C, while the monomials in $v_d(x)$ are treated as constants.

Proof. Let $C \succeq 0$ be such that $v_d(x)^\top C v_d(x) = P(x)$. Applying the Cholesky decomposition, we can write $C = U^\top U$, where U is an upper-triangular matrix. Therefore,

$$P(x) = v_d(x)^\top C v_d(x) = v_d(x)^\top U^\top U v_d(x) = \|U v_d(x)\|_2^2 = \sum_{i \in \binom{n}{\leqslant d}} (U v_d(x))_i^2 \geqslant 0.$$

Recalling that $v_d(x)$ is a vector of monomials, define the polynomials $g_i(x) := (U v_d(x))_i$. This gives us a representation of $P(x)$ as the polynomial $\sum_i g_i(x)^2$. Note that the number of unique $g_i(x)$'s is the equal to the rank of C.

For the other direction, suppose that we have a decomposition $P(x) = \sum_i g_i(x)^2$. Because the highest-degree terms in the $g_i(x)$'s cannot cancel, each $g_i(x)$ has degree at most $2d$. For each $g_i(x)$, let $\vec{g}_i$ be the vector formed by the coefficients of the monomials appearing in $g_i(x)$ such that $g_i(x) = \vec{g}_i \cdot v_d(x)$. Construct the matrix U by letting the i-th row of U be the coefficient vector $\vec{g}_i$. Then we have

$$v_d(x)^\top U^\top U v_d(x) = \sum_i (U v_d(x))_i^2 = \sum_i g_i^2(x) = P(x).$$

□

This tight connection between semidefinite programming and sum-of-squares polynomials will be a recurring theme in the rest of this monograph. In the following example, we will show how this SDP can be used to extract a sum-of-squares representation for a polynomial.

Example 3.33. Consider $P(x) = 2x^4 + 2x^3y - x^2y^2 + 5y^4$. Because $P(x)$ is homogeneous, if it can be written a a sum-of-squares, then the polynomials in the sum-of-squares representation must all have degree 2. Therefore, it is enough to consider (for simplicity) the monomial vector $v = [x^2, y^2, xy]$.

$$\begin{bmatrix} x^2 & y^2 & xy \end{bmatrix} \begin{bmatrix} c_{1,1} & c_{2,1} & c_{3,1} \\ c_{1,2} & c_{2,2} & c_{3,2} \\ c_{1,3} & c_{2,3} & c_{3,3} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix} = \sum_{i,j \in [3]} c_{i,j} v_i v_j,$$

where $v_1 = x^2$, $v_2 = y^2$, $v_3 = xy$. Expanding this sum, and using the fact that C is

symmetric, gives

$$\begin{aligned}\sum_{i,j\in[3]} c_{i,j}v_iv_j &= c_{1,1}v_1^2 + c_{2,2}v_2^2 + c_{3,3}v_3^2 + 2c_{1,2}v_1v_2 + 2c_{2,3}v_2v_3 + 2c_{1,3}v_1v_3 \\ &= c_{1,1}x^4 + c_{2,2}y^4 + c_{3,3}x^2y^2 + 2c_{1,2}x^2y^2 + 2c_{2,3}xy^3 + 2c_{1,3}x^3y \\ &= c_{1,1}x^4 + c_{2,2}y^4 + (c_{3,3} + 2c_{1,2})x^2y^2 + 2c_{2,3}xy^3 + 2c_{1,3}x^3y.\end{aligned}$$

Recall that the entries of C are the coefficients (treated as the unknowns of the SDP) of the terms in the sum-of-squares representation of $P(x)$, and therefore this gives us a system of constraints. The existence of such a symmetric PSD matrix C, and therefore, whether $P(x)$ can be represented as a sum-of-squares, is equivalent to the feasibility of the following SDP:

$$\begin{aligned} C &\succeq 0 \\ \text{s.t.} \quad c_{1,1} = 2,\ \ c_{2,2} = 5,\ \ 2c_{1,3} = 2,\ \ 2c_{2,3} = 0,\ \ c_{3,3} + 2c_{1,2} &= -1\end{aligned}$$

Solving this results in the following matrix:

$$C = \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix}.$$

This matrix is rank-2, and therefore $P(x)$ is a sum of two squares. In order to uncover the description of these squares, we will follow the forward direction of the proof of Lemma 3.32. Applying the Cholesky factorization,

$$C = U^\top U = \frac{1}{\sqrt{2}}\begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}^\top \frac{1}{\sqrt{2}}\begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}.$$

We can write $P(x) = \sum_i (Uv)_i^2$, where

$$Uv = \frac{1}{\sqrt{2}}\begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}}(2x^2 - 3y^2 + xy) \\ \frac{1}{\sqrt{2}}(y^2 + 3xy) \end{bmatrix},$$

and so $P(x)$ can be written as the following sum-of-squares

$$P(x) = \left(\frac{1}{\sqrt{2}}(2x^2 - 3y^2 + xy)\right)^2 + \left(\frac{1}{\sqrt{2}}(y^2 + 3xy)\right)^2.$$

Unfortunately, it is not the case that every non-negative polynomial can be written as a sum-of-squares, and the question of when a non-negative polynomial over $\mathbb{R}[x]$ has a sum-of-

squares representation has a long history dating back to Minkowski and Hilbert in 1885 [138]. At the end of this chapter (Section 3.3.2.1) we give an abridged history of this question and how it relates to Sum-of-Squares.

Hilbert [74] provided the first proof that, in general, non-negativity of a polynomial is not equivalent to having a representation as a sum-of-squares. His proof was non-constructive, and relied on tools from the theory of algebraic curves. It was not until 1967 that Theodore Motzkin [113] gave the first explicit counter example. His counter example,

$$M(x,y) \;:=\; 1 + x^4y^2 + x^2y^4 - 3x^2y^2,$$

became known as the *Motzkin polynomial*. Although we omit the proof that it cannot be represented as a sum-of-squares (see for example [134]), it will be illustrative to prove its non-negativity. To do so, we will show that it can be written as a sum-of-squares of rational functions.

Claim 3.34. *The Motzkin polynomial is non-negative.*

Proof. It can be checked that multiplying $M(x,y)$ by $(x^2+y^2)^2 > 0$ can be written as

$$(x^2+y^2)^2 M(x,y) = \\ (x^3y + xy^3 - 2xy)^2 + x^2(x^3y + xy^3 - 2xy)^2 + y^2(x^3y + xy^3 - 2xy)^2 + (x^2 - y^2)^2$$

Rearranging, we have

$$M(x,y) = \frac{(x^3y + xy^3 - 2xy)^2(1 + x^2 + y^2) + (x^2 - y^2)^2}{(x^2+y^2)^2},$$

which is a sum-of-squares of rational functions, and therefore must always be non-negative. □

Along with his proof that in general, non-negativity is not equivalent to having a representation as a sum-of-square, Hilbert characterized the subclasses of polynomials for which this equivalence does hold. They are:

1. Univariate polynomials,
2. Degree 2 polynomials,
3. Bivariate, degree 4 polynomials,
4. Functions over the Boolean hypercube $P : \{0,1\}^n \to \mathbb{R}$.

The most notable is that non-negative functions over the Boolean hypercube $P : \{0,1\}^n \to \mathbb{R}$ are equivalent to sum-of-squares polynomials. Furthermore, degree $2n$ is always sufficient.

Lemma 3.35. *Every non-negative function $P : \{0,1\}^n \to \mathbb{R}$ can be written as a degree $2n$ sum-of-squares polynomial.*

Proof. Let $g : \{0,1\}^n \to \mathbb{R}$ be the unique multilinearizing-map function[3] that agrees with $\sqrt{P}$ on the hypercube. Then, $P = g^2$ over $\{0,1\}^n$ and furthermore has degree at most $2n$. □

We can characterize sum-of-squares polynomials over $\{0,1\}^n$ in terms of symmetric PSD matrices in the same way that we did for sum-of-squares polynomials over $\mathbb{R}^n$ in Lemma 3.32. Unfortunately, for polynomials over the Boolean hypercube, the degree of the sum-of-squares representation is no-longer bounded by the degree of the original polynomial.

Theorem 3.36. *For a function $P : \{0,1\}^n \to \mathbb{R}$, the following are equivalent*

1. *$P(x)$ is a non-negative function.*
2. *$P(x)$ can be written as a sum-of-squares polynomial.*
3. *There exists a matrix $C \succeq 0$ such that $v_d(x)^\top C v_d(x) = P(x)$, where $v_d(x)$ is the monomial vector of degree n.*

Proof. The equivalence of (1) and (2) follows from Lemma 3.35. The proof of the equivalence with (3) follows by the same proof as in Lemma 3.32, except that now we are unable restrict to polynomials of degree at most the degree of $P(x)$. This is because, over $\{0,1\}^n$, we have the identity $x_i^2 = x_i$, and so we can no longer assume that the highest-degree terms do not cancel. Therefore, monomial vectors of degree n are necessary. □

This gives us an algorithm for obtaining sum-of-squares representations of non-negative polynomials over $\{0,1\}^n$: find a feasible solution to the SDP

$$\begin{aligned} P(x) &= \sum_{I,J \subseteq [n]} C_{I,J} v_n(x)_I v_n(x)_J, \\ C &\succeq 0, \end{aligned}$$

where the variables of this SDP are the $C_{I,J}$ for $I, J \subseteq [n]$. This is the same form as the SDP that we saw in Example 3.33. Unfortunately, because $v_n(x)$ is degree n monomial vector, C is $\binom{n}{\leqslant n} \times \binom{n}{\leqslant n}$-dimensional, and so this SDP has $O(n^{n^2})$ variables. This begs the question of whether degree n monomial vectors are indeed necessary in general, and if so, in what special cases are smaller degree monomial vectors sufficient. Unfortunately, it turns out that there exist constant-degree polynomials that require degree $\Omega(n)$ sum-of-squares representations. We will prove this for the 3XOR function in Section 5.1. This shows that, degree n monomial vectors are necessary in general. The latter question will be a focus of the remainder of this manuscript.

[3]Recall that a multilinearizing-map, as defined in Definition 2.14, is a linear function $f : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\}_{i \in [n]} \to \mathbb{R}$ such that $f(\prod_{i \in I} x_i^{c_i}) = f(\prod_{i \in I} x_i)$.

3.2 Sum-of-Squares

3.2.1 Sum-of-Squares as Lifting Semidefinite Programs

Sherali-Adams gave us a systematic way of creating a hierarchy of LP relaxations converging to the integer hull. Shor [143], Parillo [116, 117], and Lasserre [103, 102] showed how to extend this idea to SDPs. In this section we study the SDP hierarchy proposed by Lasserre, which has come to be known as the Sum-of-Squares hierarchy. To describe it, we will use the MaxCut problem as a running example.

Definition 3.37 (MaxCut)**.** Given a graph $G = (V, E)$ with weights $w_{i,j}$ for all $(i, j) \in E$, find a bipartition of the edges into S and $V \setminus S$ that maximizes the weight of edges the edges crossing between the two sets,

$$\mathsf{MaxCut}(G) \;:=\; \max_{S \subseteq V} \sum_{i \in S, j \in V \setminus S} w_{i,j}$$

MaxCut can be expressed naturally as a quadratic integer program,

$$\begin{aligned} \max \quad & \sum_{i<j} w_{i,j} \left(x_i - x_j\right)^2, && (3.4)\\ \text{s.t.} \quad & x_i \in \{0,1\} \qquad \forall i \in [n]. \end{aligned}$$

Of course, solving quadratic programs is NP-hard and so we will instead look for a tractable approximation, this time by a hierarchy of SDPs. A quadratic integer program can be relaxed to an SDP in much the same way as taking the LP relaxation. The difference is that instead of replacing the $\{0, 1\}$-constraints on the variables with non-negativity constraints, they are replaced with a symmetric PSD constraint.

To relax the quadratic program (3.4), first replace each product of variables $x_i x_j$ with a placeholder variable $y_{\{i,j\}}$. In fact, we will introduce variables y_I for every $I \subseteq [n]$ with $|I| \leqslant 2$. Let $\mathcal{M}_2(y)$ be the $\binom{n}{\leqslant 2} \times \binom{n}{\leqslant 2}$ matrix, obtained by arranging the variables y_I into a matrix. The rows and columns are labeled by sets $I \subseteq [n]$, with $|I| \leqslant 2$, and the entry $\mathcal{M}_2(y)_{I,J} = y_{I \cup J}$. For $n = 2$, $\mathcal{M}_2(y)$ is:

$$\begin{array}{c} \\ \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \end{array} \begin{array}{c} \begin{array}{cccc} \emptyset & \{1\} & \{2\} & \{1,2\} \end{array} \\ \begin{bmatrix} y_\emptyset & y_{\{1\}} & y_{\{2\}} & y_{\{1,2\}} \\ y_{\{1\}} & y_{\{1\}} & y_{\{1,2\}} & y_{\{1,2\}} \\ y_{\{2\}} & y_{\{1,2\}} & y_{\{2\}} & y_{\{1,2\}} \\ y_{\{1,2\}} & y_{\{1,2\}} & y_{\{1,2\}} & y_{\{1,2\}} \end{bmatrix} \end{array} \qquad (3.5)$$

Rather than constrain the entries of $\mathcal{M}_2(y)$ to be non-negative as we would have done for the LP relaxation, we instead enforce that $\mathcal{M}_2(y)$ is symmetric PSD. Finally, in order to ensure that every solution is correctly normalized, we include the constraint that $y_\emptyset = 1$, where $y_\emptyset$

represents the product $\prod_{i\in\emptyset} x_i$. Putting this together gives the following SDP relaxation of (3.4):

$$\max \quad \sum_{\substack{i<j \\ i,j\in[n]}} w_{i,j}\left(y_{\{i\}} - 2y_{\{i,j\}} + y_{\{j\}}\right), \tag{3.6}$$

$$\text{s.t.} \qquad \mathcal{M}_2(y) \succeq 0, \quad y_\emptyset = 1 \tag{3.7}$$

This relaxation is the first level Sum-of-Squares relaxation applied to the MaxCut problem. The symmetric PSD constraint effectively relaxes the feasible region from the feasible region $\{0,1\}^n$ of the quadratic program to be within the interval $[0,1]^n$. To see this, recall that by Lemma 3.22, the determinant of every principal submatrix of Y is non-negative. Let $S \subseteq [n]$ with $|S| \leqslant 2$, and compute the determinant of the principal submatrix obtained from $\mathcal{M}_2(y)$ by deleting all rows and columns except for those indexed by $\emptyset$ and S,

$$\det\left(\begin{bmatrix} y_\emptyset & y_S \\ y_S & y_S \end{bmatrix}\right) = y_S(y_\emptyset - y_S) = y_S(1 - y_S) \geqslant 0, \tag{3.8}$$

which holds if and only if $y_S \in [0,1]$.

At this point, it is worth comparing the constraints of this relaxation with the constraints of the 2nd level Sherali-Adams relaxation of (3.4), which consist of

$$\begin{aligned} y_{\{i,j\}} \geqslant 0,\ \ y_{\{i\}} - y_{\{i,j\}} \geqslant 0,\ \ 1 - y_{\{i\}} - y_{\{j\}} + y_{\{i,j\}} \geqslant 0 \qquad & \forall i \neq j \in [n], \\ y_{\{i\}} \geqslant 0,\ \ 1 - y_{\{i\}} \geqslant 0, \qquad & \forall i \in [n], \\ y_\emptyset = 1. & \end{aligned}$$

We have already seen that all of these constraints except for $1 - y_{\{i\}} - y_{\{j\}} + y_{\{i,j\}} \geqslant 0$ and $y_{\{i\}} - y_{\{i,j\}} \geqslant 0$ are enforced by $\mathcal{M}_2(y)$. To see that these final constraints are enforced as well, define the vector $u \in \mathbb{R}^{\binom{n}{\leqslant 2}}$, indexed by sets $S \subseteq [n]$ with $|S| \leqslant 2$ as $u_\emptyset = u_{\{i,j\}} = 1$, $u_{\{i\}} = u_{\{j\}} = -1$, and let $u_S = 0$ for every other coordinate. Then,

$$u^\top \mathcal{M}_2(y) u = y_\emptyset - y_{\{i\}} - y_{\{j\}} + y_{\{i,j\}} \geqslant 0$$

where the final inequality holds because $\mathcal{M}_2(y) \succeq 0$, and so $u^\top \mathcal{M}_2(y) u \geqslant 0$. Similarly, letting $u' \in \mathbb{R}^{\binom{n}{\leqslant 2}}$ be defined as $u'_{\{i\}} = u'_{\{i,j\}} = 1$ and $u'_S = 0$ for all other coordinates, we have $u'^\top \mathcal{M}_2(y) u' = y_{\{i\}} - y_{\{i,j\}} \geqslant 0$. Therefore, such relaxations are at least as expressive as the relaxations produced by the 2nd level of the SA hierarchy.

In fact, relaxations of this form are potentially much more expressive. Charikar et al. [43] showed that the Sherali-Adams relaxation of degree $\Omega(n)$ cannot achieve an approximation ratio better than $1/2 + \varepsilon$, for MaxCut. On the other hand, in Section 4.1 we will see that the semidefinite relaxation (3.6) achieves the same approximation ratio as the famous Goemans and Williamson SDP for MaxCut. Unfortunately, it is not always the case that an SDP

relaxation of the form (3.6) results in a reasonable approximation ratio. For such cases, we can develop a theory of lifting like we did for Sherali-Adams. Rather than having the matrix $\mathcal{M}_2(y)$ contain only variables representing terms that occur in the constraints of the quadratic program, we can include representations of higher-degree terms. This leads to a hierarchy of lifts, parameterized by the degree of variables that we introduce, known as the Sum-of-Squares (SoS) or Lasserre hierarchy.

To describe the SoS relaxation in full generality, suppose that we are trying to solve the following $\{0,1\}$-polynomial optimization problem:

$$\min_{x \in \mathsf{hull}_{\{0,1\}}(\mathcal{P})} P(x),$$
$$\text{where } \mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}.$$

Definition 3.38 (Sum-of-Squares Relaxation)**.** Let $d \geqslant \deg(\mathcal{P})/2$. The d-th level of the SoS hierarchy for $\mathcal{P}$, which we denote by $\mathsf{SOS}_d(\mathcal{P})$, is defined as follows:

1. **Extend:** Define matrices $(X_d)_{|I|,|J| \leqslant d)} := \prod_{i \in I, j \in J} x_i x_j$, and $((P_i, X)_d)_{|I|,|J| \leqslant d - \deg(P_i)/2} := P(x) \prod_{i \in I, j \in J} x_i x_j$, for every $P_i(x) \geqslant 0 \in \mathcal{P}$. Introduce the constraints

$$X_d \succeq 0, \tag{3.9}$$
$$(P_i, X)_d \succeq 0, \qquad \text{for every } P_i(x) \geqslant 0 \in \mathcal{P} \tag{3.10}$$

2. **Linearize:** Multilinearize each of the constraints introduced in the previous step by replacing x_i^c by x_i for every $c > 1$. Replace each monomial $\prod_{i \in S} x_i$ occurring in (3.9) and (3.10) by a variable y_S and denote the resulting matrices by

$$\mathcal{M}_d(y)_{|I|,|J| \leqslant d} := y_{I \cup J}, \qquad \text{and} \qquad \mathcal{M}_d(y, P_i)_{|I|,|J| \leqslant d - \deg(P_i)/2} := \sum_K (\vec{P_i})_K \cdot y_{I \cup J \cup K},$$

where $(\vec{P_i})_K$ is the coefficient of the term $\prod_{i \in K} x_i$ in $P_i(x)$. Finally, add the normalizing constraint $y_\emptyset = 1$.

The resulting spectahedron $\mathsf{SOS}_d(\mathcal{P})$ is defined by the set of constraints:

$$\begin{aligned} \mathcal{M}_d(y) &\succeq 0, \\ \mathcal{M}_d(y, P_i) &\succeq 0 \quad \forall i \in [m], \\ y_\emptyset &= 1. \end{aligned}$$

The resulting program is defined on $\binom{n}{\leqslant 2d}$ variables y_S with $|S| \leqslant 2d$. The variable $y_\emptyset$ is intended to represent $\prod_{i \in \emptyset} x_i$, the constant 1 term. Therefore, we include the constraint $y_\emptyset = 1$ is included to ensure that any solution we obtain from optimizing over this relaxation is correctly normalized. For clarity, the matrix $\mathcal{M}_d(y)$ can be seen in Figure 3.4.

$$\mathcal{M}_d(y) \;=\; \begin{array}{c} \\ \emptyset \\ \{1\} \\ \{2\} \\ \{1,2\} \\ \vdots \\ I \\ \vdots \end{array} \begin{array}{c} \begin{array}{ccccccc} \emptyset & \{1\} & \{2\} & \{1,2\} & \dots & J & \dots \end{array} \\ \begin{bmatrix} 1 & y_{\{1\}} & y_{\{2\}} & y_{\{1,2\}} & \cdots & y_J & \cdots \\ y_{\{1\}} & y_{\{1\}} & y_{\{1,2\}} & y_{\{1,2\}} & \cdots & y_{J\cup\{1\}} & \cdots \\ y_{\{2\}} & y_{\{1,2\}} & y_{\{2\}} & y_{\{1,2\}} & \cdots & y_{J\cup\{2\}} & \cdots \\ y_{\{1,2\}} & y_{\{1,2\}} & y_{\{1,2\}} & y_{\{1,2\}} & \cdots & y_{J\cup\{1,2\}} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & & \\ y_I & y_{I\cup\{1\}} & y_{I\cup\{2\}} & y_{I\cup\{1,2\}} & & y_{I\cup J} & \\ \vdots & \vdots & \vdots & \vdots & & & \ddots \end{bmatrix} \end{array}$$

Figure 3.4: The moment matrix $\mathcal{M}_d(y)$ of the d-th level of the SoS hierarchy.

Example 3.39. Let $\mathcal{ILP}(\mathcal{P}, c)$ be an ILP over constraints $\mathcal{P} = \{a_1^\top x \geqslant b_1, \dots, a_m^\top x \geqslant b_m\}$ and variables $x_i \in \{0,1\}$. The d-th level SoS relaxation of $\mathcal{P}$ consists of the following constraints

$$\begin{aligned} (y_{I\cup J})_{|I|,|J|\leqslant d} &\succeq 0, \\ \Big(\sum_{j=1}^{n} a_{i,j}\cdot y_{I\cup J\cup\{j\}} - b_i \cdot y_{I\cup J}\Big)_{|I|,|J|\leqslant d-\deg(P_i)/2} &\succeq 0 \qquad \forall i \in [m], \\ y_\emptyset &= 1 \end{aligned}$$

Solving the SoS Relaxation. A solution to the original optimization problem can be obtained from the SoS-relaxation by solving the SDP,

$$\mathcal{SDP}(\mathsf{SOS}_d(\mathcal{P}), P) \;:=\; \min_{\alpha\in\mathsf{SOS}_d(\mathcal{P})} \sum_{|I|\leqslant\deg(P)} \vec{P}_I \alpha_I,$$

where $\vec{P} \in \mathbb{R}^{\binom{n}{\leqslant 2d}}$ is the coefficient vector of $P(x)$ such that $\vec{P}_I$ is the coefficient of the term $\prod_{i\in I} x_i$ in $P(x)$. Using the ellipsoid method, and in particular Corollary 3.12, this SDP can be solved up to an additive ε-error in time proportional to $n^{O(d)}$, provided that the coefficients of the constraints in $\mathcal{P}$ and P are polynomial in n^d.

Corollary 3.40. *Let $\mathcal{P}$ be a set of polynomial inequalities. Then $\mathcal{SDP}(\mathsf{SOS}_d(\mathcal{P}), P)$ can be solved, up to an additive error ε, in time $\mathsf{poly}(m, n^d, \mathrm{size}(P), \mathrm{size}(\mathcal{P}))\cdot\log\varepsilon^{-1}$.*

One might object that $\mathsf{SOS}_d(\mathcal{P})$ is not in the standard form of an SDP that we saw in Definition 3.4; indeed, it contains several PSD constraints. By arranging these PSD constraints as a single block-diagonal matrix we can re-write $\mathsf{SOS}_d(\mathcal{P})$ in the standard form of a dual SDP. This transformation is described explicitly in Section 3.3.1.1.

Moment Matrices The matrices $\mathcal{M}_d(y)$ and $\mathcal{M}_d(y, P_i)$ are known as the *moment matrices* of the SoS relaxation $\mathsf{SOS}_d(\mathcal{P})$. This name comes from a distributional view of SoS, which we will explore in Section 3.2.2, where we view the entries of these matrices as the moments of some probability distribution. Intuitively, the moment matrices ensure that the linearizations of the constraints are satisfied while enforcing that the variables $y_{I,J}$ behave consistently with the products $\prod_{i \in I \cup J} x_i$ that they are intended to represent. Formally, the moment matrices satisfy the following useful properties, some of which were already discussed for $\mathcal{M}_2(y)$ in the introduction.

Lemma 3.41. *For any set of polynomial inequalities $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$, $\mathsf{SOS}_d(\mathcal{P})$ satisfies the following properties:*

1. *$0 \leqslant y_J \leqslant y_I \leqslant 1$ for every $I \subseteq J \subseteq [n]$ with $|J| \leqslant d$.*
2. *$\sum_{J \subseteq T} (-1)^{|J|} y_{S \cup J} \geqslant 0$ for every non-negative d-junta $J_{S,T}(x)$.*
3. *If $\alpha \in \{0,1\}^n$ satisfies every $P_i(x) \geqslant 0 \in \mathcal{P}$, then $\alpha \in \mathrm{proj}_{[n]}(\mathsf{SOS}_d(\mathcal{P}))$ [4] for every $d \geqslant \deg(\mathcal{P})$.*

The proof of this Lemma can be found in the Appendix, and follows by arguments similar to the proof that this held for degree 2 SoS at the beginning of this subsection. Property (2) shows that the moment matrix constraints are at least as restrictive as non-negative juntas: any solution to the level d SoS relaxation satisfies every non-negative d-junta. It follows that the SoS relaxation is a tightening of the Sherali-Adams relaxation.[5]

Corollary 3.42. *Let $\mathcal{P}$ be a set of polynomial inequalities. For every d,*

$$\mathsf{SOS}_d(\mathcal{P}) \subseteq \mathsf{SA}_d(\mathcal{P})$$

Proof. By Lemma 3.41, any solution to $\mathsf{SOS}_d(\mathcal{P})$ satisfies every d-junta. It is left to show that any solution to $\mathsf{SOS}_d(\mathcal{P})$ also satisfies the linearization of $J_{S,T}(x) \cdot P_i(x) \geqslant 0$ for every $P_i(x) \geqslant 0 \in \mathcal{P}$ and $(d - \deg(P_i)/2)$-junta. This follows by an almost identical argument to the proof of property (2) in Lemma 3.41, instead using the matrix $\mathcal{M}_d(y, P_i)$ in-place of $\mathcal{M}_d(y)$. □

Hierarchy of Relaxations Because SA forms a hierarchy of polytopes converging to the integer-hull, Corollary 3.42 suggests that the same is true for SoS. Indeed, we will show that any feasible solution to $\mathsf{SOS}_{d+1}(\mathcal{P})$ will also satisfy $\mathsf{SOS}_d(\mathcal{P})$. That is, the levels of the SoS hierarchy form a sequence of tightening spectahedron converging to the integer hull.

Lemma 3.43. *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$, then the following hold:*

[4] Recall that $\mathrm{proj}_{[n]}(\mathcal{P}) = \left\{\alpha \restriction_{y_{\{1\}}, \ldots, y_{\{n\}}} : \alpha \in \mathcal{P}\right\}$, the orthogonal projection of $\mathcal{P}$ to the first n variables.

[5] It is worth noting that the d-th level of the SoS relaxation introduces the y-variable representations of monomials of degree up to $2d$, while degree d Sherali-Adams only allows representations of degree up to d monomials.

1. $\mathsf{SOS}_d(\mathcal{P}) \supseteq \mathsf{SOS}_{d+1}(\mathcal{P})$ *for any* $d \geqslant \deg(\mathcal{P})$,

2. $\mathsf{SOS}_d(\mathcal{P}) \supseteq \mathsf{hull}_{\{0,1\}}(\mathcal{P})$,

3. $\mathrm{proj}_{[n]}\left(\mathsf{SOS}_{n+\deg(\mathcal{P})/2}\right) = \mathsf{hull}_{\{0,1\}}(\mathcal{P})$.

Proof. For (1), let $\alpha \in \mathsf{SOS}_d(\mathcal{P})$, and so $\mathcal{M}_d(\alpha) \succeq 0$ and $\mathcal{M}_d(\alpha, P_i) \succeq 0$. Observe that $\mathcal{M}_{d-1}(y)$ is the principal submatrix of $\mathcal{M}_d(y)$ obtained by removing all rows and columns indexed by I with $|I| = d$. The same is true for $\mathcal{M}_{d-1}(y, P_i)$ with respect to $\mathcal{M}_d(y, P_i)$. By Lemma 3.21, the principal submatrices of any PSD matrix are themselves PSD. Therefore $\mathcal{M}_d(\alpha), \mathcal{M}_d(\alpha, P_i) \succeq 0$.

To prove (2), because $\mathsf{SOS}_d(\mathcal{P})$ is convex, it suffices to show that every $\alpha \in \{0,1\}^n \cap \mathsf{hull}_{\{0,1\}}(\mathcal{P})$ satisfies the constraints of $\mathsf{SOS}_d(\mathcal{P})$. Define $\tilde{\alpha} \in \mathbb{R}^{\binom{n}{\leqslant d}}$ as

$$\tilde{\alpha}_I \;:=\; \prod_{i \in I} \alpha_i \qquad \forall I \subseteq [n], |I| \leqslant d.$$

Note that because $\tilde{\alpha}_I \in \{0,1\}$ for all I, $(\tilde{\alpha}_I)^2 = \tilde{\alpha}_I$. Therefore, we can write $\mathcal{M}_d(\tilde{\alpha}) = \tilde{\alpha}\tilde{\alpha}^\top$. Consider any $v \in \mathbb{R}^{\binom{n}{\leqslant d}}$, then we have

$$v^\top \mathcal{M}_d(\tilde{\alpha}) v = v^\top \tilde{\alpha}\tilde{\alpha}^\top v = (\tilde{\alpha}^\top v)^2 \geqslant 0,$$

and so $\mathcal{M}_d(\tilde{\alpha}) \succeq 0$. Similarly, consider $P_i(x) \geqslant 0 \in \mathcal{P}$, and recall that $\mathcal{M}_d(y, P_i)_{I,J}$ is the linearization of $(P_i \cdot X_{I,J})_{|I|,|J| \leqslant d}$, where $(P_i \cdot X_{I,J}) = P_i(x) \prod_{i \in I, j \in J} x_i x_j$. Because $\alpha \in \{0,1\}^n$,

$$\begin{aligned}
\mathcal{M}_d(\tilde{\alpha}, P_i) &= (P_i(\alpha) \cdot X(\alpha)_{I,J})_{|I|,|J| \leqslant d} \\
&= \Big(P_i(\alpha) \prod_{i \in I, j \in J} \alpha_i \alpha_j\Big)_{|I|,|J| \leqslant d} \\
&= P_i(\alpha)\Big(\tilde{\alpha}_I \tilde{\alpha}_J\Big)_{|I|,|J| \leqslant d} \\
&= P_i(\alpha)\tilde{\alpha}\tilde{\alpha}^\top.
\end{aligned}$$

Because α is a solution to $\mathcal{P}$, $P_i(\alpha) \geqslant 0$. Let $v \in \mathbb{R}^{\binom{n}{\leqslant d - \deg(P_i)/2}}$. Then,

$$v^\top \mathcal{M}_d(\tilde{\alpha}) v = v^\top P_i(\alpha)\tilde{\alpha}\tilde{\alpha}^\top v = P_i(\alpha) v^\top \tilde{\alpha}\tilde{\alpha}^\top v = P_i(\alpha)(\tilde{\alpha}^\top v)^2 \geqslant 0.$$

To prove (3), we will need some additional machinery which is developed in the following section. In particular, the proof of (3) is given in Corollary 3.53. For now, we note that the weaker bound, $\mathsf{SOS}_{n+\deg(\mathcal{P})} = \mathsf{hull}_{\{0,1\}}(\mathcal{P})$, follows from (2) along with Corollary 3.42 and the fact that $\mathsf{SA}_{n+\deg(\mathcal{P})}(\mathcal{P}) = \mathsf{hull}_{\{0,1\}}(\mathcal{P})$. □

By Lemma 3.43, the SoS relaxations forms a sequence of nested spectahedron, converging to the integer hull,

$$\mathrm{proj}_{[n]}\left(\mathsf{SOS}_{\deg(\mathcal{P})}(\mathcal{P})\right) \supseteq \mathrm{proj}_{[n]}\left(\mathsf{SOS}_{\deg(\mathcal{P})+1}(\mathcal{P})\right) \supseteq \ldots \supseteq \mathrm{proj}_{[n]}\left(\mathsf{SOS}_{O(n)}\right) = \mathsf{hull}_{\{0,1\}}(\mathcal{P}).$$

This is known as the *Sum-of-Squares hierarchy*.

3.2.2 Sum-of-Squares as Locally Consistent Distributions

The SoS hierarchy corresponds to an ever-tightening sequence of spectahedrons converging to the integer hull. Unfortunately, as the level of the hierarchy increases, so does the running time required to optimize over the resulting spectahedron. Therefore, we would like to understand how well of an approximation a tractable level of the hierarchy achieves. In order to do so, it is necessary to understand the points that lie within the d-th level SoS spectahedron.

3.2.2.1 Pseudo-Expectations

For Sherali-Adams we had a nice way of describing the feasible points as locally consistent expectation functions. We will construct a similar distributional representation of the point within the SoS relaxation. Recall that every point $\alpha \in \mathsf{hull}_{\{0,1\}}(\mathcal{P})$ can be represented as a probability distribution $\mu^{(\alpha)}$ over $\{0,1\}^n$ as follows. Because $\alpha \in \mathsf{hull}_{\{0,1\}}(\mathcal{P})$, it can be expressed as a convex combination of the $\{0,1\}$-solutions to $\mathcal{P}$,

$$\alpha = \sum_{\beta \in \{0,1\}^n} \lambda_\beta \cdot \beta,$$

such that $\lambda_\beta = 0$ if β does not satisfy the constraints of $\mathcal{P}$. The probability distribution $\mu^{(\alpha)} : \{0,1\}^n \to [0,1]$ is defined as $\mu^{(\alpha)}(\beta) = \lambda_\beta$.

Minimizing a polynomial $P(x)$ over $\mathsf{hull}_{\{0,1\}}(\mathcal{P})$ can be phrased in this distributional language as minimizing the expectation of $P(x)$ over the space of all such distributions,

$$\min_{\mu} \{\mathbb{E}_\mu[P(x)] : \mu \text{ is supported on } \{0,1\}\text{-solutions to } \mathcal{P}\ \} = \min_{\mathbb{E} \in \mathcal{E}(\mathcal{P})} [\mathbb{E}[P(x)]]\,,$$

where $\mathcal{E}(\mathcal{P})$ is the set of expectation functions defined on $\{0,1\}$-distributions for $\mathcal{P}$, $\mathcal{E}(\mathcal{P}) := \{\mathbb{E}_\mu : \mu \text{ is a distribution supported on } \{0,1\}\text{-solutions to } \mathcal{P}\}$. Of course, it takes an exponential amount of space to describe a distribution over $\{0,1\}^n$ in general. Therefore, this optimization problem is typically intractable. Instead, we settle for only optimizing over the relaxation $\mathsf{SOS}_d(\mathcal{P})$. This corresponds to relaxing the requirements on the expectations in the set $\mathcal{E}(\mathcal{P})$ to only require that they *look like* true expectations over $\{0,1\}$-solutions to $\mathcal{P}$ to the constraints of $\mathsf{SOS}_d(\mathcal{P})$. We will denote by $\tilde{\mathbb{E}} : \mathbb{R}[x] \to \mathbb{R}$ such a *pseudo-expectation*.

We now develop the properties that $\tilde{\mathbb{E}}$ must satisfy in order to fool the d-th level of SoS into thinking that $\tilde{\mathbb{E}}$ is a true expectation defined only on $\{0,1\}$-solutions to $\mathcal{P}$. First, because $\tilde{\mathbb{E}}$ pretends to be an expectation over a probability distribution supported only $\{0,1\}^n$, it should associate $x_i^c = x_i$ for every $i \in [n]$, and therefore be a multilinearizing map.[6] Next, because $\mathsf{SOS}_d(\mathcal{P})$ introduces only the multilinearizations of degree at most $2d$ polynomials, as we will see, this corresponds to only requiring that $\tilde{\mathbb{E}}$ be consistent with the at most $2d$-th moments of a true probability distribution over $\{0,1\}^n$. Indeed, because every point $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ satisfies $\mathcal{M}_d(\alpha) \succeq 0$, and because these pseudo-expectations are intended to

[6] In the sense of Definition 2.14.

be a distributional view of these points α, we should enforce that any pseudo-expectation $\tilde{\mathbb{E}}$ satisfies

$$\tilde{\mathbb{E}}[X_d] \succeq 0,$$

where $\tilde{\mathbb{E}}$ is applied component-wise to X_d, recalling that $(X_d)_{i\in I, j\in J} := \prod_{i\in I\cup J} x_i$. This is equivalent to requiring that for every vector $v \in \mathbb{R}^{\binom{n}{\leqslant d}}$,

$$\tilde{\mathbb{E}}[v^\top X_d v] \geqslant 0.$$

We can rephrase this as

$$\begin{aligned}\tilde{\mathbb{E}}[v^\top X_d v] &= \tilde{\mathbb{E}}\Big[\sum_{|I|,|J|\leqslant d} v_I v_J \prod_{i\in I\cup J} x_i\Big] \\ &= \tilde{\mathbb{E}}\Big[\big(\sum_{|I|\leqslant d} v_I \prod_{i\in I} x_i\big)\big(\sum_{|J|\leqslant d} v_J \prod_{j\in J} x_j\big)\Big], \\ &= \tilde{\mathbb{E}}\Big[\big(\sum_{|I|\leqslant d} v_I \prod_{i\in I} x_i\big)^2\Big].\end{aligned}$$

If we interpret v as the coefficient vector of (the multilinearization) of a polynomial $Q(x) \in \mathbb{R}[x]$ this is equivalent to the condition that for every $Q(x) \in \mathbb{R}[x]$ of degree at most d,

$$\tilde{\mathbb{E}}[Q^2(x)] \geqslant 0.$$

Similarly, the condition that $\mathcal{M}_d(y, P_i)$ enforces that $\tilde{\mathbb{E}}[P_i(x)Q^2(x)] \geqslant 0$, for every $Q(x) \in \mathbb{R}[x]$ with $\deg(Q) \leqslant d - \deg(P_i)/2$, and $P_i(x) \geqslant 0 \in \mathcal{P}$. This leads us to define a pseudo-expectation for SoS as any function that satisfies these properties.

Definition 3.44 (Pseudo-Expectation for $\mathcal{P}$)**.** Let $\mathcal{P}$ be a set of polynomial inequalities. A multilinearizing map[7] $\tilde{\mathbb{E}} : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ is a degree $2d$ (level d) pseudo-expectation for $\mathcal{P}$ if the following hold:

1. $\tilde{\mathbb{E}}[1] = 1$,
2. $\tilde{\mathbb{E}}[Q^2(x)] \geqslant 0$ for every polynomial $Q \in \mathbb{R}[x]$ with $\deg(Q) \leqslant d$,
3. $\tilde{\mathbb{E}}[Q^2(x) \cdot P_i(x)] \geqslant 0$ for every $P_i(x) \geqslant 0 \in \mathcal{P}$ and every $Q \in \mathbb{R}$ with $\deg(Q) \leqslant d - \deg(P_i)/2$.

Denote by $\mathcal{E}_{2d}(\mathcal{P})$ the set of all degree $2d$ pseudo-expectations for $\mathcal{P}$. As we will see next, the set $\mathcal{E}_{2d}(\mathcal{P})$ is equivalent to $\mathsf{SOS}_d(\mathcal{P})$.

[7]Recall that a multilinearizing map, as defined in Definition 2.14, is a linear function $f : \mathbb{R}[x]\setminus\{(x_i^2-x_i)\} \to \mathbb{R}$ such that $f\left(\sum_{j\in J}\prod_{i\in I_j} x_i^{c_{i,j}}\right) = \sum_{j\in J} f\left(\prod_{i\in I_j} x_i\right)$.

Theorem 3.45. *Let $\mathcal{P}$ be a set of polynomial constraints, and for any $\alpha \in \mathbb{R}^{\binom{n}{\leqslant 2d}}$ define the multilinearizing map $\tilde{\mathbb{E}}_\alpha : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ as*

$$\tilde{\mathbb{E}}_\alpha \left[\prod_{i \in S} x_i \right] \;:=\; \alpha_S,$$

for every $S \subseteq [n]$, $\tilde{\mathbb{E}}_\alpha[1] = 1$, and extend $\tilde{\mathbb{E}}_\alpha$ linearly. Then, $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ if and only if $\tilde{\mathbb{E}}_\alpha \in \mathcal{E}_{2d}(\mathcal{P})$.

Proof. Suppose that $\alpha \in \mathsf{SOS}_d(\mathcal{P})$, then α assigns a value α_S to every variable y_S with $|S| \leqslant 2d$. Define the multilinearizing map $\tilde{\mathbb{E}}_\alpha$ as in the statement of the theorem. For terms of higher degree, $\tilde{\mathbb{E}}$ can be defined arbitrarily. To see that $\tilde{\mathbb{E}}_\alpha$ satisfies (1), observe that $\tilde{\mathbb{E}}_\alpha[1] = \tilde{\mathbb{E}}_\alpha[\prod_{i \in \emptyset} x_i] = y_\emptyset = 1$. For (2) and (3), let $P(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$ and $Q(x) \in \mathbb{R}[x]$ with $\deg(Q) \leqslant d - \deg(P)/2$. Recalling that $\tilde{\mathbb{E}}_\alpha$ is a multilinearizing map, let $\vec{P}, \vec{Q} \in \mathbb{R}^{\binom{n}{\leqslant d}}$ be the coefficient vectors of the multilinearization of $P(x)$ and $Q(x)$ respectively (that is, applying the transformation $x_i^c = x_i$) such that $\vec{Q}_I$, $\vec{P}_I$ are the coefficients of the term $\prod_{i \in I} x_I$.

$$\begin{aligned}
\tilde{\mathbb{E}}_\alpha \left[P(x) Q^2(x) \right] &= \sum_{|I|,|J| \leqslant d - \deg(P_i)/2} \tilde{\mathbb{E}}_\alpha \left[P(x) \vec{Q}_I \vec{Q}_J \prod_{i \in I \cup J} x_i \right] \\
&= \sum_{\substack{|I|,|J| \leqslant d - \deg(P_i)/2, \\ |K| \leqslant \deg(P_i)}} \tilde{\mathbb{E}}_\alpha \left[\vec{P}_K \vec{Q}_I \vec{Q}_J \prod_{i \in I \cup J \cup K} x_i \right] \\
&= \sum_{\substack{|I|,|J| \leqslant d - \deg(P_i)/2, \\ |K| \leqslant \deg(P_i)}} \vec{Q}_I \vec{Q}_J \left(\vec{P}_K \cdot y_{I \cup J \cup K} \right) \\
&= \vec{Q}^\top \mathcal{M}(P, \alpha) \vec{Q} \geqslant 0,
\end{aligned}$$

where the final inequality follows because $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ and so $\mathcal{M}(P, \alpha) \succeq 0$.

For the other direction, suppose that $\tilde{\mathbb{E}} \in \mathcal{E}_{2d}(\mathcal{P})$. Let $P(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$, and define $\mathcal{M}(P, y)$ as

$$\mathcal{M}(P, y)_{|I|,|J| \leqslant d - \deg(P)/2} = \tilde{\mathbb{E}} \left[P(x) \prod_{i \in I \cup J} x_i \right].$$

Let $Q(x) \in \mathbb{R}[x]$ be any polynomial of degree at most $d - \deg(P)/2$, and let $\vec{P}, \vec{Q} \in \mathbb{R}^{\binom{n}{\leqslant d}}$ be the coefficient vector of the multilinearization of $P(x)$ and $Q(x)$ respectively. Then

$$\begin{aligned}
\vec{Q}^\top \mathcal{M}(P, y) \vec{Q} &= \sum_{|I|,|J| \leqslant d, K} \vec{Q}_I \vec{Q}_J \left(\vec{P}_K \cdot y_{I \cup J \cup K} \right) \\
&= \sum_{|I|,|J| \leqslant d} \vec{Q}_I \vec{Q}_J \tilde{\mathbb{E}} \left[\sum_K \vec{P}_K \prod_{i \in K} x_i \prod_{i \in I \cup J} x_i \right]
\end{aligned}$$

$$= \tilde{\mathbb{E}}\left[P(x)\sum_{|I|,|J|\leqslant d}\vec{Q}_I\vec{Q}_J\prod_{i\in I\cup J}x_i\right]$$
$$= \tilde{\mathbb{E}}[P(x)Q^2(x)] \geqslant 0.$$

□

Using Theorem 3.45 together with Corollary 3.12, we can find pseudo-expectations that approximately satisfy constraints in polynomial time. Formally, we can obtain the following corollary:

Corollary 3.46. *Let $\mathcal{P}$ be a set of polynomial constraints on $\mathbb{R}^n$. Then, for any d, there is an $n^{O(d)}$ time algorithm to find a pseudo-expectation $\tilde{\mathbb{E}} : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to \mathbb{R}$ such that for any $P(x) \geqslant 0 \in \mathcal{P}$ and polynomial $Q(x)$ of degree at most $d - \deg(P)$,*

$$\tilde{\mathbb{E}}[P(x)Q(x)] \geqslant -2^{-n^d}\Big(\sum_{S\subseteq[n]}\vec{P}_S^2\Big)\Big(\sum_{S\subseteq[n]}\vec{Q}_S^2\Big).$$

3.2.2.2 Pseudo-Distributions

Instead of working with a pseudo-expectation, we could work directly with the underlying *pseudo-distribution*. Analogous to SA, each pseudo-expectation induces a family of consistent marginal distributions. Recall from the previous section that every point $\alpha \in \mathsf{hull}_{\{0,1\}}(\mathcal{P})$ can be viewed as a distribution $\mu^{(\alpha)}$ over solutions in $\{0,1\}^n$ to $\mathcal{P}$. This distribution induces a family of marginal distributions $\{\mu_S^{(\alpha)} : S \subseteq [n]\}$, where $\mu_S^{(\alpha)}$ is the marginal distribution on $\{0,1\}^S$. [8] Like Sherali-Adams, the constraints of SoS can be viewed as attempting to verify that these marginal distributions are consistent probability distributions. Unfortunately, sub-linear levels of the SoS hierarchy are unable to fully verify each of these marginal distributions. For example, by Lemma 3.41 $\mathsf{SOS}_d(\mathcal{P})$ can only guarantees that $y_S \in [0,1]$ for $|S| \leqslant d$. This corresponds to d-th level of the SoS hierarchy only being able to verify that, for $|S| \leqslant d$, $\mu_S(\beta) \in [0,1]$ where $\beta \in \{0,1\}^S$. Therefore, for a point to lie within the d-th level of the SoS hierarchy, it only needs to be consistent with a relaxed version of a distribution. In fact, it makes sense to reuse the notion of a pseudo-distribution from Definition 2.20. The difference between a pseudo-distribution for Sherali-Adams and one from SoS will emerge in the different properties that the pseudo-expectations over these pseudo-distributions must satisfy.

Lemma 3.47. *Let $\mathcal{P}$ be a set of polynomial inequalities, then any degree $2d$ pseudo-expectation for $\mathcal{P}$ defines a degree d pseudo-distribution.* [9]

[8] Recall that $\{0,1\}^S$ is the set of Boolean assignments to the variables $\{x_i : i \in S\}$.

[9] One might wonder why a degree $2d$ pseudo-expectation defines only a degree d pseudo-distribution. One answer is that a degree n pseudo-distribution is a true distribution over $\{0,1\}^n$. A more enlightening answer is that Lemma 3.41 only guarantees that the values assigned by a degree $2d$ pseudo-expectation are within the range $[0,1]$ monomials of degree up to d. Indeed, the values assigned to terms of degree $> d$ may be negative, and therefore would not form a valid distribution.

Proof. For $\tilde{\mathbb{E}} \in \mathcal{E}_{2d}(\mathcal{P})$, let $S \subseteq [n]$ with $|S| \leqslant d$. Define the function $\mu_S : \{0,1\}^S \to \mathbb{R}$ as follows: for every $S' \subseteq S$, the event $1_{S',S\setminus S'}$ corresponding to $x_i = 1$ for all $i \in S'$ and $x_j = 0$ for all $j \in S \setminus S'$ is assigned probability

$$\mu_S(1_{S',S\setminus S'}) = \tilde{\mathbb{E}}\Big[\prod_{i\in S'} x_i \prod_{j\in S\setminus S'} (1-x_j)\Big],$$

and let $\mu = \{\mu_S\}_{|S|\leqslant d}$. By Lemma 3.41 and the equivalence between pseudo-expectations and the points $\alpha \in \mathsf{SOS}_d(\mathcal{P})$, it follows that $\mu_S(\beta) \in [0,1]$ for every $\beta \in \{0,1\}^S$. That μ_S is indeed a true probability distribution over $\{0,1\}^S$ follows from the consistency between these marginal distributions, and is identical to the proof of Lemma 2.21. □

The family of marginal distributions $\{\mu_S\}_{|S|\leqslant d}$ are consistent with each other, and so they appear to SoS to be the marginal distributions up to subsets of d variables of some true probability distribution over $\{0,1\}^n$. If we add the additional constraints that the expectation taken over this pseudo-distribution satisfies the constraints of $\mathsf{SOS}_d(\mathcal{P})$, then the pseudo-distribution *fools* SoS into believing that it is true distribution over $\{0,1\}$-solutions to $\mathcal{P}$.

Definition 3.48 (Pseudo-Expectation for $\mathcal{P}$)**.** Let $\mathcal{P}$ be a set of polynomial inequalities. A multilinearizing map $\tilde{\mathbb{E}} : \mathbb{R}[x] \setminus \{(x_i^2 - x_i)\} \to [0,1]$ is a degree $2d$ pseudo-expectation for $\mathcal{P}$ if there exists a degree $2d$ pseudo-distribution μ such that for every polynomial $S \subseteq [n]$ with $|S| \leqslant 2d$,

$$\tilde{\mathbb{E}}\Big[\prod_{i\in S} x_i\Big] = \mu(1_{S,\emptyset}),$$

and $\tilde{\mathbb{E}}[P(x)\cdot Q^2(x)] \geqslant 0$ for every $P(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$ and every $Q(x) \in \mathbb{R}[x]$ with $\deg(Q) \leqslant d - \deg(P)/2$

Remark. Occasionally in the literature an alternative definition of a pseudo-expectation is used. Here, a degree $2d$ pseudo-distribution for $\mathcal{P}$ is defined as a function $\mu : \{0,1\}^n \to \mathbb{R}$ satisfying that the expectation over by μ is a degree $2d$ pseudo-expectation. This differs from a true probability distribution by allowing the marginals of degree higher than d to assign negative probabilities. This definition of a pseudo-distribution is simply the extension of the standard definition of pseudo-distribution for $\mathcal{P}$ (Definition 3.48) to have moments up to n by assigning the marginals on $> d$ variables to arbitrary values, with the only requirement being that the marginals are consistent.

3.2.2.3 Evolution of the Sum-of-Squares Relaxation

Pseudo-expectations allow us to characterize the points that exist within each level of the SoS hierarchy. We can take a more fine-grained view and ask how the set of feasible points change between levels of the hierarchy. That is, under what conditions does a point in the d-th level survive to the $(d+1)$-st?

Lemma 3.49. *Let $\mathcal{P}$ be a set of polynomial inequalities. For every $\alpha \in SOS_d(\mathcal{P})$, and every $i \in [n]$ with $0 < \alpha_{\{i\}} < 1$, there exists $\beta^{(0)}, \beta^{(1)} \in SOS_{d-1}(\mathcal{P})$ such that $\beta^{(1)}_{\{i\}} = 1, \beta^{(0)}_{\{i\}} = 0$, and*

$$\alpha \in \operatorname{conv}\left(\beta^{(0)}, \beta^{(1)}\right).^{10}$$

Proof. This argument is due to Rothvoß [135]. Let $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ and $i \in [n]$ be an index such that $0 < \alpha_{\{i\}} < 1$. Define the points $\beta^{(0)}, \beta^{(1)} \in \mathbb{R}^{\binom{n}{\leqslant d}}$, and non-negative multipliers $\lambda_0, \lambda_1 \in \mathbb{R}$ as

$$\begin{aligned} \beta^{(0)}_S &:= \frac{\alpha_S - \alpha_{S\cup\{i\}}}{1-\alpha_{\{i\}}}, & \lambda_0 &:= 1-\alpha_{\{i\}}, \\ \beta^{(1)}_S &:= \frac{\alpha_{S\cup\{i\}}}{\alpha_{\{i\}}}, & \lambda_1 &:= \alpha_{\{i\}}, \end{aligned}$$

for every $S \subseteq [n]$ with $|S| \leqslant 2(d-1)$. Observe that the following hold:

1. $\beta^{(0)} = 0$ and $\beta^{(1)} = 1$, and
2. $\lambda_0 \cdot \beta^{(0)}_S + \lambda_1 \cdot \beta^{(1)}_S = (\alpha_S - \alpha_{S\cup\{i\}}) + \alpha_{S\cup\{i\}} = \alpha_S$. Therefore α is a convex combination of $\beta^{(0)}$ and $\beta^{(1)}$.

Next, we verify that $\mathcal{M}_{d-1}(\beta^{(0)}), \mathcal{M}_{d-1}(\beta^{(1)}) \in \mathsf{SOS}_{d-1}(\mathcal{P})$. To begin, observe that

$$\beta^{(0)}_\emptyset = \frac{\alpha_\emptyset - \alpha_{\{i\}}}{1-\alpha_{\{i\}}} = 1, \qquad \text{and,} \qquad \beta^{(1)}_\emptyset = \frac{\alpha_{\{i\}}}{\alpha_{\{i\}}} = 1.$$

Because $\mathcal{M}_d(\alpha) \succeq 0$, by the Cholesky decomposition, we can write $\mathcal{M}_d(\alpha) = U^\top U$ where U is an upper-triangular matrix. Denote the columns of U by u_I for $I \subseteq [n]$, $|I| \leqslant d$, and observe that

$$\langle u_I, u_J\rangle = \mathcal{M}_d(\alpha)_{I\cup J} = \alpha_{I\cup J}.$$

To prove that $\mathcal{M}_{d-1}(\beta^{(0)}) \succeq 0$ and $\mathcal{M}_{d-1}(\beta^{(1)}) \succeq 0$, we show that they both admit a Cholesky decomposition.

We will begin with $\beta^{(0)}$. Note that it is enough to show that $(1-\alpha_{\{i\}})\mathcal{M}_{d-1}(\beta^{(0)}) \succeq 0$ because PSD matrices are closed under conic combinations, and by Lemma 3.41 it follows that $\alpha_{\{i\}} \in [0,1]$. Define the vectors $u^{(0)}_I := (u_I - u_{I\cup\{i\}})$ for every $|I| \leqslant d-1$. Letting $I, J \subseteq [n]$ with $|I|, |J| \leqslant d-1$, observe that

$$\begin{aligned} \langle u^{(0)}_I, u^{(0)}_J\rangle &= \langle u_I, u_J\rangle - \langle u_{I\cup\{i\}}, u_J\rangle - \langle u_I, u_{J\cup\{i\}}\rangle - \langle u_{I\cup\{i\}}, u_{J\cup\{i\}}\rangle \\ &= \alpha_{I\cup J} - \alpha_{I\cup J\cup\{i\}} \\ &= (1-\alpha_{\{i\}})\beta^{(0)}_{I\cup J} \end{aligned}$$

[10] To simplify our notation we are being somewhat sloppy with this statement. α is an $\binom{n}{\leqslant d}$-dimensional vector and $\beta^{(i)}$ is $\binom{n}{\leqslant d-1}$-dimensional vector, and we mean that α, when restricted to its first $\binom{n}{\leqslant d-1}$ coordinates can be written as a convex combination of points in $\mathsf{SOS}_d(\mathcal{P})$

$$= (1 - \alpha_{\{i\}})\mathcal{M}_{d-1}(\beta^{(0)}).$$

Form the matrix $U^{(0)}$ whose columns are $u_I^{(0)}$ for every $I \leqslant d-1$, and so $(U^{(0)})^\top U^{(0)} = (1-\alpha_{\{i\}})\mathcal{M}_{d-1}(\beta^{(0)})$. Because U is upper-triangular, and $U^{(0)} = (u_I^{(0)})_{|I|\leqslant d-1}$, then $U^{(0)}$ is upper triangular as well and so $(U^{(0)})^\top U^{(0)}$ is a Cholesky decomposition of $\mathcal{M}_{d-1}(\beta^{(0)})$. By Theorem 3.14, the equivalence between the existence of a Cholesky decomposition and positive semidefiniteness, we can conclude that $\mathcal{M}_{d-1}(\beta^{(0)}) \succeq 0$.

Next, let $P_i(x) \geqslant 0$ be any polynomial inequality in $\mathcal{P}$; we aim to show that $(1-\alpha_{\{i\}})\mathcal{M}_{d-1}(\beta^{(0)}, P_i) \succeq 0$. The proof is similar to the previous argument. Because $\mathcal{M}_d(\alpha, P_i) \succeq 0$ it admits a Cholesky decomposition $U_{P_i}^\top U_{P_i}$ for some upper-triangular matrix U_{P_i}, with columns $u_I^{P_i}$ for $I \subseteq [n]$, $|I| \leqslant d$. Observe that by definition we have

$$\langle u_I^{P_i}, u_J^{P_i} \rangle = \mathcal{M}_d(\alpha, P_i)_{I,J}.$$

Define $u_I^{(0),P_i} := (u_I^{P_i} - u_{I\cup\{i\}}^{P_i})$ for all $I \subseteq [n]$, $|I| \leqslant d-1$. Then, for every $I, J \subseteq [n]$ with $|I|, |J| \leqslant d-1$,

$$\begin{aligned}
\langle u_I^{(0),P_i}, u_J^{(0),P_i} \rangle &= \langle u_I^{P_i}, u_J^{P_i} \rangle - \langle u_{I\cup\{i\}}^{P_i}, u_J^{P_i} \rangle - \langle u_I^{P_i}, u_{J\cup\{i\}}^{P_i} \rangle + \langle u_{I\cup\{i\}}^{P_i}, u_{J\cup\{i\}}^{P_i} \rangle, \\
&= \mathcal{M}_d(\alpha, P_i)_{I,J} - \mathcal{M}_d(\alpha, P_i)_{I\cup\{i\},J} - \mathcal{M}_d(\alpha, P_i)_{I,J\cup\{i\}} + \mathcal{M}_d(\alpha, P_i)_{I\cup\{i\},J\cup\{i\}}, \\
&= \mathcal{M}_d(\alpha, P_i)_{I,J} - \mathcal{M}_d(\alpha, P_i)_{I\cup\{i\},J}, \\
&= \left(1 - \alpha_{\{i\}}\right) \mathcal{M}_{d-1}(\beta^{(0)}, P_i)_{I,J},
\end{aligned}$$

where the third equality follows because $\mathcal{M}_d(\alpha, P_i)$ is symmetric and $\mathcal{M}_d(\alpha, P_i)_{I,J\cup\{i\}} = \mathcal{M}_d(\alpha, P_i)_{I\cup\{i\},J} = \mathcal{M}_d(\alpha, P_i)_{I\cup\{i\},J\cup\{i\}}$ by definition of $\mathcal{M}_d(\alpha, P_i)$. Form the matrix $U_{P_i}^{(0)} := (u_I^{(0),P_i})_{|I|\leqslant d-1}$, and observe that $(U_{P_i}^{(0)})^\top U_{P_i}^{(0)} = \mathcal{M}_{d-1}(\beta^{(0)}, P_i)$. By the same argument as above, $U_{P_i}^{(0)}$ is upper-triangular, and therefore we have given a Cholesky decomposition of $(1-\alpha_{\{i\}})\mathcal{M}_{d-1}(\beta^{(0)}, P_i) \succeq 0$, and so $\mathcal{M}_{d-1}(\beta^{(0)}, P_i) \succeq 0$.

Finally, for $\beta^{(1)}$ we prove that $\alpha_{\{i\}} \cdot \mathcal{M}_{d-1}(\beta^{(1)}) \succeq 0$. As usual, consider the Cholesky decomposition $\mathcal{M}_d(\alpha) = U^\top U$. Let $u_I^{(1)} = u_{I\cup\{i\}}$, then for every $I, J \subseteq [n]$ with $|I|, |J| \leqslant d$,

$$\langle u_I^{(1)}, u_J^{(1)} \rangle = \langle u_{I\cup\{i\}} u_{J\cup\{i\}} \rangle = \alpha_{I\cup J\cup\{i\}} = \alpha_{\{i\}} \beta_{I\cup J}^{(1)} = \alpha_{\{i\}} \mathcal{M}_{d-1}(\beta^{(1)}).$$

Define $U^{(1)} := (u_I^{(1)})_{|I|\leqslant d-1}$, then $(U^{(1)})^\top U^{(1)} = \mathcal{M}_{d-1}(\beta^{(1)})$. Because U is upper-triangular and the columns of $U^{(1)}$ are an ordered subset of the columns of U, we can conclude that $U^{(1)}$ is upper-triangular. Therefore, this is the Cholesky decomposition of $\alpha_{\{i\}}\mathcal{M}_{d-1}(\beta^{(1)})$, and so $\alpha_{\{i\}}\mathcal{M}_{d-1}(\beta^{(1)}) \succeq 0$, and $\mathcal{M}_{d-1}(\beta^{(1)}) \succeq 0$. The proof that $\mathcal{M}_{d-1}(\beta^{(1)}, P_i) \succeq 0$ for every $P_i(x) \geqslant 0 \in \mathcal{P}$ is similar, and is left as an exercise.

□

The converse of this Lemma holds as well. We will leave the proof as an exercise, noting that it follows by essentially running the proof of the previous lemma in reverse.

Corollary 3.50. *Let $\mathcal{P}$ be a set of polynomial inequalities, $d \geqslant \deg(\mathcal{P})$, and $\alpha \in \mathbb{R}^{\binom{n}{\leqslant d+1}}$. If for every $i \in [n]$ there exists $\beta^{(0)}, \beta^{(1)} \in \mathsf{SOS}_d(\mathcal{P})$ with $\beta^{(0)}, \beta^{(1)} \in \{0,1\}$ such that $\alpha_S = \lambda\beta^{(1)} + (1-\lambda)\beta^{(0)}$ for $\lambda \in [0,1]$ and $\alpha_{S\cup\{i\}} = \lambda\beta_S^{\{1\}}$, then $\alpha \in \mathsf{SOS}_{d+1}(\mathcal{P})$*

We can iterate the previous lemma to obtain a characterization of the points contained in $\mathsf{SOS}_d(\mathcal{P})$ as a convex combination of points contained in $\mathsf{SOS}_t(\mathcal{P})$, for $t < d$. For this, it will be useful to define the SoS hierarchy for $t < \deg(\mathcal{P})/2$. The constraints of $\mathsf{SOS}_t(\mathcal{P})$ are defined as usually except that we omit any constraint of degree greater than t. That is, if $\deg(P_i)/2 > t$ for $P_i(x) \geqslant 0 \in \mathcal{P}$, we do not include the constraint $\mathcal{M}_t(y, P_i) \succeq 0$.

Theorem 3.51. *Let $\mathcal{P}$ be a set of polynomial inequalities, and let $0 \leqslant t \leqslant d$. For any $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ and every $S \subseteq [n]$ with $|S| = t$,*

$$\alpha \in \operatorname{conv}\left(\{\beta \in \mathsf{SOS}_{d-t}(\mathcal{P}) : \beta_i \in \{0,1\}, \forall i \in S\}\right).$$

Proof. The proof is identical to the proof of Theorem 2.25, except that we replace every application of Lemma 2.23 with Lemma 3.49.

□

It is interesting to note that for any $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ and any subset of indices $S \subseteq [n]$ with $|S| = t$, the proof of Theorem 3.51 allows us to write $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ as the following convex combination of points in $\mathsf{SOS}_{d-t}(\mathcal{P})$,

$$\alpha = \sum_{\substack{K\cup T = S \\ K\cap T = \emptyset}} \lambda^{(K,T)}\beta^{(K,T)},$$

where $\lambda^{(K,T)} \in \mathbb{R}^{\geqslant 0}$ and $\beta^{(K,T)} \in \mathsf{SOS}_{d-t}(\mathcal{P})$ are defined as,

$$\beta_I^{(K,T)} = \frac{\sum_{J\subseteq T}(-1)^{|J|}\alpha_{I\cup K\cup J}}{\sum_{J\subseteq T}(-1)^{|J|}\alpha_{J\cup K}}, \qquad \text{and} \qquad \lambda^{(K,T)} = \sum_{J\subseteq T}(-1)^{|J|}\alpha_{J\cup K}.$$

The converse of Theorem 3.51 holds as well, giving a characterization of the evolution of points between levels in the SoS hierarchy. A consequence of this theorem is an alternative construction of a pseudo-distribution over which the expectation is a pseudo-expectation for $\mathcal{P}$.

Corollary 3.52. *Let $\mathcal{P}$ be a set of polynomial inequalities and $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ for any $d \geqslant 0$. Then α defines a degree d pseudo-distribution, over which the expectation is a degree d pseudo-expectation for $\mathcal{P}$.*

Proof. Let $\alpha \in \mathsf{SOS}_d(\mathcal{P})$. For any $S \subseteq [n]$ with $|S| \leqslant d$, applying Theorem 3.51 with $t = |S|$ gives us a set of points $\{\beta_i\} \in \mathsf{SOS}_{d-t}(\mathcal{P})$ such that

$$\alpha = \sum_{\beta\in\{\beta_i\}} \lambda_\beta \beta,$$

for some $\lambda_\beta \geqslant 0$, such that $\sum_{\beta \in \{\beta_i\}} \lambda_\beta = 1$. From this, we can construct a distribution $\mu_S : \{0,1\}^S \to \mathbb{R}$ is defined as follows. Let $\{\beta_i \restriction_S\} = \{\beta \restriction_S : \beta \in \{\beta_i\}\}$ [11], and for every $\kappa \in \{0,1\}^S$, define

$$\mu_S(\kappa) := \begin{cases} \lambda_\kappa & \text{if } \kappa \in \{\beta_i \restriction_S\}, \\ 0 & \text{otherwise.} \end{cases}$$

Performing this process for every $S \subseteq [n]$ with $|S| \leqslant d$, gives a family of distributions $\mu := \{\mu_S\}$. Define the expectation taken over μ as

$$\mathbb{E}_\mu \Big[\prod_{i \in S} x_i \prod_{j \in T} (1 - x_j) \Big] = \mu_{S \cup T}(1_{S,T}),$$

for every $|S| + |T| \leqslant d$, and extended linearly. We prove that μ is both a pseudo-distribution and that $\mathbb{E}_\mu$ is a degree d pseudo-expectation for $\mathcal{P}$ by showing that for every $I \subseteq S \subseteq [n]$ with $|S| \leqslant d$,

$$\mathbb{E}_\mu \Big[\prod_{i \in I} x_i \Big] = \alpha_I.$$

Indeed, by the definition of the multipliers λ_0, λ_1 from the proof of Theorem 3.51,

$$\mathbb{E}_\mu \Big[\prod_{i \in I} x_i \Big] = \sum_{\substack{I \subseteq J \subseteq S \\ T = S \setminus J}} \sum_{H \subseteq T} (-1)^{|H|} \alpha_{J \cup H} = \alpha_I.$$

Therefore, all $\mu_S \in \mu$ are consistent, and so μ is a pseudo-distribution. Furthermore, because $\mathbb{E}_\mu$ is the pseudo-expectation defined from α and $\alpha \in \mathsf{SOS}_d(\mathcal{P})$, $\mathbb{E}_\mu$ is a degree d pseudo-expectation for $\mathcal{P}$. □

As a consequence of Theorem 3.51, we argue that level $n + \deg(\mathcal{P})/2$ SoS is sufficient to converge to the integer hull for all $\mathcal{P}$.[12]

Corollary 3.53. *For any set of polynomial inequalities $\mathcal{P}$, $\mathsf{SOS}_{n+\deg(\mathcal{P})/2}(\mathcal{P}) = \mathsf{hull}_{\{0,1\}}(\mathcal{P})$.*

Proof. Applying theorem 3.51 with $d = n + \deg(\mathcal{P})/2$ allows us to write each point $\alpha \in \mathsf{SOS}_{n+\deg(\mathcal{P})/2}(\mathcal{P})$ as a convex combination of the points $\beta \in \mathsf{SOS}_{\deg(\mathcal{P})/2}(\mathcal{P})$ such that for all $i \in [m]$, $\beta_{\{i\}} \in \{0,1\}$. The projection of β to the original n variables must satisfy every constraint of $\mathcal{P}$, and therefore every $\alpha \in \mathsf{SOS}_{n+\deg(\mathcal{P})/2}(\mathcal{P})$ can be written as a convex combination of $\{0,1\}$-solutions to $\mathcal{P}$. □

[11] Recall that $\beta \restriction_S$ is the vector β restricted to the coordinates in the set $S \subseteq [n]$.

[12] We refer to the remark at the end of Section 2.2.2.3 for a discussion as to why level $n + \deg(\mathcal{P})/2$ is necessary, rather than level n, to converge to the integer hull.

3.2.3 Sum-of-Squares as a Proof System

The previous sections have focused on SoS as a method for producing approximate solutions to polynomial optimization problems. Suppose that we want to guarantee that the d-th level of the SoS hierarchy applied to a set of constraints $\mathcal{P}$ produces solutions that are sufficiently close to the integer hull of $\mathcal{P}$. That is, we would like to show that the SoS relaxation achieves an approximation ratio of some factor r. To do this, we will explore a dual view of SoS which produces certificates of the approximation ratio achieved by the relaxation.

To be concrete, suppose that we are trying to minimize a polynomial $P(x)$ over the $\{0,1\}$-solutions to a set of polynomial inequalities $\mathcal{P}$. Because $\mathsf{hull}_{\{0,1\}}(\mathcal{P}) \subseteq \mathsf{SOS}_d(\mathcal{P})$, showing that the relaxation achieves a r-approximation amounts to proving that

$$r \cdot \mathsf{opt} \leqslant \min_{\alpha \in \mathsf{SOS}_d(\mathcal{P})} \sum_I P_I \alpha_I,$$

where $\mathsf{opt} = \min_{x \in \mathsf{hull}_{\{0,1\}}(\mathcal{P})} P(x)$. This is equivalent to showing that d-th level of SoS can verify that the set of polynomial inequalities $\mathcal{P} \cup \{P(x) \leqslant r \cdot \mathsf{opt} + \varepsilon\}$ contains no $\{0,1\}$-solutions, for $0 < \varepsilon \ll 1$. In order to show this, it be convenient to take a dual view of SoS as a proof system for refuting unsatisfiable systems of polynomial inequalities. Recall that a point α is contained within the level d SoS spectahedron, if and only if the associated degree $2d$ pseudo-expectation $\tilde{\mathbb{E}}_\alpha$ satisfies $\tilde{\mathbb{E}}_\alpha[P_i(x)Q^2(x)] \geqslant 0$ for every $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$ and $Q(x) \in \mathbb{R}[x]$ with $\deg(Q) \leqslant d - \deg(P_i)/2$. For simplicity of notation, denote the set of degree at most $2d$ sum-of-squares polynomials as

$$\Sigma^2_{2d} := \{Q(x) \in \mathbb{R}[x] : Q(x) \text{ is a sum-of-squares polynomial}, \deg(Q) \leqslant 2d\},$$

and let $\Sigma^2 := \bigcup_{d \geqslant 0} \Sigma^2_d$ be the set of all sum-of-squares polynomials. Then, of course,

$$\tilde{\mathbb{E}}_\alpha \left[Q_0(x) + \sum_{i=1}^m P_i(x) Q_i(x) \right] \geqslant 0,$$

for $P_i(x) \geqslant 0 \in \mathcal{P}$ and every $Q_i(x) \in \Sigma^2_{2d-\deg(P_i)}$ and $Q_0(x) \in \Sigma_{2d}$. It follows that the d-th level of the SoS hierarchy can be defined as

$$\mathsf{SOS}_d(\mathcal{P}) = \left\{ \alpha \in \mathbb{R}^{\binom{n}{\leqslant 2d}} : \tilde{\mathbb{E}}_\alpha \left[Q_0(x) + \sum_{i=1}^m P_i(x) Q_i(x) \right] \geqslant 0, P_i(x) \geqslant 0 \in \mathcal{P}, Q_i \in \Sigma^2_{2d-\deg(P_i)} \right\}. \tag{3.11}$$

Because we require $\tilde{\mathbb{E}}_\alpha$ to be a multilinearizing map, this is equivalent to including the axioms $x_i^2 = x_i$ and for every $i \in [n]$ in the set of polynomials $\mathcal{P}$. Define the convex cone generated from $\mathcal{P}$ and the set of degree at most d sum-of-squares polynomials as

$$\Sigma^2_{2d}(\mathcal{P} \cup \{x_i^2 = x_i\}) :=$$
$$\left\{ Q_0(x) + \sum_{i=1}^\ell P_i(x) Q_i(x) : P_i(x) \geqslant 0 \in \mathcal{P} \cup \{\pm(x_i^2 - x_i) \geqslant 0\}, Q_i(x) \in \Sigma^2_{2d-\deg(P_i)} \right\}.$$

We can define a degree $2d$ *SoS derivation* as any polynomial that lies within this cone.

Definition 3.54 (Sum-of-Squares Derivation)**.** A degree $2d$ SoS derivation of a polynomial inequality $P(x) \geqslant c_0$ from a set of polynomial inequalities $\mathcal{P}$ is a representation of $P(x) - c_0$ as a sum-of-squares:

$$P(x) - c_0 = Q_0(x) + \sum_{i=1}^{\ell} P_i(x)Q_i(x),$$

where $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{\pm(x_i^2 - x_i) \geqslant 0\}$, and $Q_0(x) \in \Sigma^2_{2d}$ and $Q_i(x) \in \Sigma^2_{d-\deg(P_i)}$ are sum-of-squares polynomials.

A degree $2d$ SoS derivation of $P(x) \geqslant c_0$ from $\mathcal{P}$ is a proof that the minimum value achieved by $P(x)$ over all $x \in \mathrm{SA}_d(\mathcal{P})$ is at lower bounded by c_0; we will prove this fact later in Theorem 3.61. In the case when there is no $\{0,1\}$-solution to $\mathcal{P}$, the unsatisfiability of $\mathcal{P}$ over $\{0,1\}^n$ can be witnessed by a SoS derivation of any negative constant from $\mathcal{P}$. Because SoS derivations involve non-negative linear combinations of products of square polynomials, a derivation of a negative constant can only exist if $\mathcal{P}$ is infeasible.

Definition 3.55 (Sum-of-Squares Refutation)**.** A degree d SoS refutation of a set of polynomial inequalities $\mathcal{P}$ is a degree d derivation of the constant -1 from $\mathcal{P}$.

Because the connection between the proof system and hierarchy perspectives of SoS is parameterized only by the degree of the polynomials involved, the degree is the primary measure of complexity studied for SoS. Even so, from the perspective of proof complexity, it is also natural to study the *size* of a SoS derivation, defined as the sum of the sizes of the polynomials in the derivation.

Refutations of CNF Formulas. Proof complexity is typically interested in the length of refutations of CNF formulas. Therefore, we need a suitable encoding of CNF formulas as a set of linear or polynomial inequalities, and the encoding as we used for SA will suffice. Namely, we will represent a clause $C(I,J) = \bigvee_{i\in I} x_i \vee \bigvee_{j\in J} \neg x_j$ by the polynomial inequality $\sum_{i\in I} x_i + \sum_{j\in J}(1-x_j) - 1 \geqslant 0$.

SoS is a surprisingly powerful proof system, and it admits short proofs of many of the standard unsatisfiable formulas used to prove lower bounds. We will give several examples of this phenomenon, the first is by showing that SoS can prove the induction principle.

Example 3.56. Consider the induction principle IND_n defined by the following set of clauses: (1) (x_1); (2) $(\neg x_i \vee x_{i+1})$ for all $i \in [n-1]$; (3) $(\neg x_n)$. We can encode these clauses as a set of polynomials:

$$(x_1 - 1) \geqslant 0, \tag{3.12}$$

$$x_i(x_{i+1} - 1) \geqslant 0 \qquad \forall i \in [n-1], \tag{3.13}$$

$$(-x_n) \geqslant 0. \tag{3.14}$$

To see that this is unsatisfiable (over $\{0,1\}$-assignments), observe that the first inequality

forces $x_1 = 1$, the intermediate inequalities force $x_i = 1$ for $i \in [n]$, and the final inequality requires that $x_n = 0$.

SoS can refute this system in degree 4 by deriving for each $i \in [n-1]$ the constraint

$$x_{i+1}(1 - x_i) \geqslant 0, \tag{3.15}$$

which can be obtained by multilinearizing the square polynomial $(x_{i+1}(1-x_i))^2$; we claim that SoS can perform this multilinearization, and show how to complete the refutation. Adding together (3.13) with (3.15),

$$x_i(x_{i+1} - 1) + x_{i+1}(1 - x_i) = x_{i+1} - x_i, \qquad \forall i \in [n-1]. \tag{3.16}$$

Finally, summing every inequality in (3.16) gives

$$\sum_{i \in [n-1]} (x_{i+1} - x_i) = x_n - x_1, \tag{3.17}$$

which together with (3.12) and (3.14) completes the SoS refutation,

$$(x_1 - 1) + (x_n - x_1) + (-x_n) = -1 \geqslant 0. \tag{3.18}$$

Finally, we show that $(x_{i+1}^2(1 - x_i))^2$ can be multilinearized without increasing the degree. Adding the SoS inequalities $(x_{i+1} - x_{i+1}^2) \geqslant 0$, and $x_{i+1}^2(x_i - x_i^2) \geqslant 0$ to $(x_{i+1}(1 - x_i))^2 = x_{i+1}^2 - 2x_i x_{i+1}^2 + (x_i x_{i+1})^2$ gives

$$\begin{aligned} &(x_{i+1}^2 - 2x_i x_{i+1}^2 + (x_i x_{i+1})^2) + (x_{i+1} - x_{i+1}^2) + (x_{i+1}^2(x_i - x_i^2) \\ =&(x_{i+1} - 2x_i x_{i+1}^2 + (x_i x_{i+1})^2) + (x_{i+1}^2(x_i - x_i^2) = x_{i+1} - x_i x_{i+1}^2. \end{aligned} \tag{3.19}$$

Next, to multilinearize $x_i x_{i+1}^2$, sum the inequalities $\frac{1}{2}(x_{i+1} - x_{i+1}^2)(1 - x_i)^2 \geqslant 0$, $\frac{1}{2}(x_{i+1}^2 - x_{i+1}) \geqslant 0$, and $\frac{1}{2}(x_{i+1}^2 - x_{i+1})x_i^2 \geqslant 0$,

$$\begin{aligned} &\frac{1}{2}(x_{i+1} - x_{i+1}^2)(1 - x_i)^2 + \frac{1}{2}(x_{i+1}^2 - x_{i+1}) + \frac{1}{2}(x_{i+1}^2 - x_{i+1})x_i^2 \\ =&\frac{1}{2}\Big((x_{i+1} - x_{i+1}^2)(1 - 2x_i + x_i^2) - (x_{i+1} - x_{i+1}^2) - (x_{i+1} - x_{i+1}^2)x_i^2\Big) \\ =&x_i(x_{i+1}^2 - x_{i+1}). \end{aligned} \tag{3.20}$$

Adding (3.20) to (3.19) completes the linearization, $(x_{i+1} - x_i x_{i+1}^2) + (x_i(x_{i+1}^2 - x_{i+1})) = x_{i+1} - x_i x_{i+1}$.

The IND_n was the first formula shown to require super-constant degree Nullstellensatz refutations.[a] This example (along with the fact that SoS can simulate the refutations produced by Nullstellensatz) shows that SoS is strictly more expressive than the Nullstellensatz proof system.

[a] Recall that Nullstellensatz refutations were defined in Section 1.3

In the previous example SoS was able to make use of the axioms $\pm(x_i - x_i^2) \geqslant 0$ in order to multilinearize polynomials without increasing the degree. This is in fact a general phenomenon.

Claim 3.57. *Any degree d polynomial $P(x)$ of size S can be multilinearized in Sum-of-Squares in degree $2d$ and size $\mathsf{poly}(S)$.*

Proof. multilinearization can be done by repeating the following procedure for each monomial $T(x)$ in $P(x)$. Denote by $\deg(T, x_i)$ the degree of the variable x_i in $T(x)$, and let $c_T \in \mathbb{R}$ be the coefficient of $T(x)$. Repeat the following until every variable in T has degree 1:

1. Let i be such that $\deg(T, x_i) > 1$, and let E and O be the set of indices of variables with even and odd degree in $T(x)$.

2. Add the valid SoS inequalities

$$\frac{c_T}{2}(x_i^2 - x_i)\Big(x_i^{\deg(T,x_i)/2-1}\Big)^2\Big(\prod_{j\in E\cup O\setminus\{i\}}\Big(x_j^{\deg(T,x_j)/2}\Big)^2\Big(1-\prod_{j\in O}x_j\Big)^2 \geqslant 0, \tag{3.21}$$

$$\frac{c_T}{2}(x_i - x_i^2)\Big(x_i^{\deg(T,x_i)/2-1}\Big)^2\Big(\prod_{j\in E\cup O\setminus\{i\}} x_j^{\deg(T,x_j)/2}\Big)^2 \geqslant 0, \tag{3.22}$$

$$\frac{c_T}{2}(x_i - x_i^2)\Big(x_i^{\deg(T,x_i)/2-1}\Big)^2\Big(\prod_{j\in E\cup O\setminus\{i\}} x_i^{\deg(T,x_j)/2}\Big)^2\Big(\prod_{j\in O}x_j\Big)^2 \geqslant 0. \tag{3.23}$$

   Note that if c_T is negative, these still remain valid SoS inequalities because the negative sign can be absorbed into the $(x_i^2 - x_i)$ term, flipping its value to $(x_i - x_i^2)$, which is also a valid axiom of SoS. The result is to reduce the degree of x_i by 1 in $T(x)$.

3. Set $T(x)$ to $T(x)$ with the degree of x_i reduced by 1, and repeat the process.

Each term has degree at most d, and therefore this process completes after at most d iterations. Each iteration introduces a linear number of monomials. Therefore, $P(x)$ can be multilinearized in size $\mathsf{poly}(d \cdot S) = \mathsf{poly}(S)$ and degree $2d$.

□

Example 3.58. Consider multilinearizing x^3y^2. The first round of the algorithm introduces

$$\frac{1}{2}(x^2 - x)y^2(1-x)^2 + \frac{1}{2}(x - x^2)y^2 + \frac{1}{2}(x - x^2)y^2x^2 = x^2y^2 - x^3y^2. \tag{3.24}$$

The following round introduces

$$\frac{1}{2}(x^2-x)y^2+\frac{1}{2}(x-x^2)y^2+\frac{1}{2}(x-x^2)y^2=xy^2-x^2y^2, \tag{3.25}$$

and then

$$\frac{1}{2}(y^2-y)(1-x)^2+\frac{1}{2}(y-y^2)+\frac{1}{2}(y-y^2)x^2=xy^2-xy. \tag{3.26}$$

Adding Equations 3.24, 3.25, and 3.26 to x^3y^2 gives the linearization xy.

We end by showing that SoS has low-degree proofs of perhaps the most famous unsatisfiable formula in proof complexity, the pigeonhole principle. This was the original formula shown to be exponentially hard for Resolution [72] and bounded-depth Frege [26], and to require linear degree for Sherali-Adams [51] and the Polynomial Calculus [130]. The pigeonhole principle PHP^n_{n-1} is typically defined over a set of boolean variables $x_{i,j}$ for $i \in [n]$ and $j \in [n-1]$ representing whether pigeon i is mapped to hole j. The clauses of PHP^n_{n-1} enforce that this mapping is injective,

$$\textbf{Pigeon Axioms:}\quad x_{i,1}\vee\ldots\vee x_{i,n-1}, \qquad \sum_{k\in[n-1]} x_{i,k}-1\geqslant 0, \quad \forall i\in[n]$$

$$\textbf{Hole Axioms:}\quad \neg x_{i,k}\vee\neg x_{j,k}, \qquad 1-x_{i,k}-x_{j,k}\geqslant 0, \quad \forall i,j\in[n], k\in[n-1].$$

Claim 3.59. The pigeonhole principle PHP^n_{n-1} has a degree 4 SoS refutation of polynomial size.

Proof. First, from the hole axioms, we derive the inequality $1-\sum_{i\in[n]} x_{i,k}\geqslant 0$ for every $k\in[n-1]$, which says that the k-th hole has at most one pigeon mapped to it. This can be done by summing the following inequalities which are valid for SoS,

$$\sum_{i,j\in[n],i\neq j}\big(1-x_{i,k}-x_{j,k}\big)x_{i,k}^2+\big(1-\sum_{i\in[n]}x_{i,k}\big)^2=1-\sum_{i\in[n]}x_{i,j}\geqslant 0,$$

where the equality follows by multilinearizing using Claim 3.57. Summing these inequalities for all $k\in[n-1]$ together with the pigeon axioms

$$\sum_{k\in[n-1]}\Big(1-\sum_{i\in[n]}x_{i,k}\Big)+\sum_{i\in[n]}\Big(\sum_{k\in[n-1]}x_{i,k}-1\Big)=-1\geqslant 0, \tag{3.27}$$

which completes the refutation. □

The remainder of this section is organized as follows. In Section 3.2.3.1 we will discuss the relationship between the SoS as a proof system and as a method of generating a hierarchy of relaxations. In particular, how a degree $(2d+\deg(\mathcal{P}))$ SoS refutation of a set of polynomials $\mathcal{P}$ is equivalent to a proof that $\mathsf{SOS}_d(\mathcal{P})$ spectahedron is empty. In fact, we will see that

the level d SoS relaxation of a set of polynomials $\mathcal{P}$ is the formal SDP dual of a degree $2d$ SoS derivation over $\mathcal{P}$. From this will follow a proof of the soundness and completeness of SoS as a proof system. Furthermore, by the fact that level $n + \deg(\mathcal{P})/2$ is sufficient for SoS to derive the integer hull of any set of polynomial inequalities $\mathcal{P}$, it will follow that any set of polynomial inequalities $\mathcal{P}$ that are unsatisfiable over $\{0,1\}^n$ have a degree at most $2n + \deg(\mathcal{P})$ refutation in SoS. The weaker result, that degree $2n + 2\deg(\mathcal{P})$ is sufficient for any SoS refutation can be argued simply by showing that any SA proof can be converted into a proof in SoS with at most twice the degree. In Section 3.2.3.2 we will prove this conversion and discuss the relationship between SoS, SA and other prominent algebraic proof systems. Finally, in Section 3.2.3.3, we discuss under what conditions SoS proofs can be found efficiently.

3.2.3.1 Soundness and Completeness

SoS is a proof system in the traditional sense, meaning that the proofs that it produces are polynomial-time verifiable, and that it is both sound and refutationally complete. We will argue that the latter holds by showing that the existence of a pseudo-expectation implies the non-existence of any SoS refutation. Furthermore, SOS is derivationally complete as well. This is a consequence of the duality between the optimization and proof complexity views of SoS.

Theorem 3.60 (Soundness and Refutational Completeness of SoS)**.** *Let $\mathcal{P}$ be a set of polynomial inequalities. There exists a degree $2d$ SoS refutation of $\mathcal{P}$ if and only if the following equivalent conditions hold:*

1. *The level d SoS spectahedron $\mathsf{SOS}_d(\mathcal{P})$ is empty.*
2. *There is no degree $2d$ pseudo-expectation for $\mathcal{P}$.*

Proof. We will follow the argument presented in [24]. Suppose that there is a degree $2d$ SoS refutation Π of $\mathcal{P}$,

$$Q_0(x) + \sum_{i=1}^{\ell} P_i(x)Q_i(x) = -1.$$

As well, suppose that there exists a degree $2d$ pseudo-expectation $\tilde{\mathbb{E}}$ for $\mathcal{P}$. Applying $\tilde{\mathbb{E}}$ to Π, we have

$$\begin{aligned}\tilde{\mathbb{E}}[-1] &= \tilde{\mathbb{E}}\left[Q_0(x) + \sum_{i=1}^{\ell} P_i(x)Q_i(x)\right] \\ &= \tilde{\mathbb{E}}\left[Q_0(x)\right] + \sum_{i=1}^{\ell} \tilde{\mathbb{E}}\left[Q_i(x)P_i(x)\right] \geqslant 0,\end{aligned}$$

which follows by linearity of $\tilde{\mathbb{E}}$. On the other hand,

$$\tilde{\mathbb{E}}[-1] = -1 \cdot \tilde{\mathbb{E}}[1] = -1.$$

For the other direction, suppose that there is no degree $2d$ SoS refutation of $\mathcal{P}$, and so the convex cone $\Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$ does not contain the constant -1, where we interpret each polynomial $P(x)$ as its coefficient vector $\vec{P}$. By the Hyperplane Separation Theorem (Theorem 3.28), there exists a vector $\vec{E}$ such that for every $P(x) \in \Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$,

$$\langle \vec{P}, \vec{E} \rangle \geqslant 0, \qquad \text{and} \qquad \langle -\mathbf{1}, \vec{E} \rangle \leqslant 0,$$

where $\mathbf{1}$ is the all 1s vector. It remains to prove that $\langle -\mathbf{1}, \vec{E} \rangle \neq 0$ and that $\langle \mathbf{1}, \vec{E} \rangle = 1$. Suppose that $\langle -\mathbf{1}, \vec{E} \rangle = 0$, that is, $-\mathbf{1}$ lies on the boundary of $\Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$. We will show that this implies that $-\mathbf{1}$ lies in the interior of this cone as well by arguing that it is a conic combination of the elements of $\Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$.

Because $\Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$ is a convex cone, there exists a polynomial $\hat{P}(x) \in \Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$ such that

$$\lambda \hat{P}(x) - 1 \in \Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$$

for every scalar $\lambda > 0$. Furthermore, because of the axioms $\pm(x_i - x_i^2) \geqslant 0$, $\Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$ contains every inequality of the form $\prod_{i \in I} x_i \leqslant 1$. It follows that, for every polynomial $P(x) \in \Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$, we have a polynomial of the form $P(x) \leqslant r$ for some $r \in \mathbb{R}$. In particular there exists an $r' \in \mathbb{R}$ such that $r' - \hat{P}(x) \geqslant 0$ is contained within this cone. Combining this with $\lambda \hat{P}(x) - 1 \geqslant 0$, we have

$$(\lambda \hat{P}(x) - 1) + \lambda(r' - \hat{P}(x)) = -1 + \lambda r'.$$

Setting $\lambda < 1/r'$ implies that $-1 + \lambda r' < 0$, and so $-\mathbf{1}$ is a conic combination of elements of $\Sigma_d^2(\mathcal{P} \cup \{x_i^2 = x_i\})$, contradicting the assumption that $\langle -\mathbf{1}, \vec{E} \rangle = 0$.

Therefore, we can conclude that $\langle \mathbf{1}, \vec{E} \rangle > 0$. Furthermore, by scaling $\vec{E}$, we can assume that $\langle \mathbf{1}, \vec{E} \rangle = 1$. If we interpret $\vec{E}$ as a linear function $\tilde{\mathbb{E}} : \mathbb{R}[x] \to \mathbb{R}$, where $\tilde{\mathbb{E}}[P(x)] = \langle \vec{E}, \vec{P} \rangle$ for every $P(x) \in \mathbb{R}[x]$ with $\deg(P) \leqslant 2d$, then $\tilde{\mathbb{E}}$ satisfies the definition of a degree $2d$ pseudo-expectation for $\mathcal{P}$.

□

Like SA, SoS satisfies derivational completeness as well. Any polynomial inequality that is logically implied by $\mathcal{P}$ has a SoS derivation. This is a consequence of the duality between SoS as a hierarchy of SDP relaxations and as a proof system. This surprising duality is the cornerstone reason why one is able to leverage analysis about SoS proofs in order to derive algorithms and impossibility results about the corresponding hierarchy of SDPs. Formally, this duality says that a polynomial inequality is valid for the degree d SoS relaxation (i.e. that is satisfies every $\tilde{\mathbb{E}} \in \mathcal{E}_d(\mathcal{P})$) if and only if it has a degree d SoS derivation from $\mathcal{P}$. This is summarized in the following theorem.

Theorem 3.61 (Derivational Completeness and Strong Duality for SoS)**.** *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be a set of polynomial inequalities. For any $P(x) \in \mathbb{R}[x]$ with $\deg(P) \leqslant d$,*

$$\min\left\{\tilde{\mathbb{E}}[P(x)] : \tilde{\mathbb{E}} \in \mathcal{E}_d(\mathcal{P})\right\} = \max\left\{c_0 : \exists \text{ a degree } d \text{ SoS derivation of } P(x) \geqslant c_0 \text{ from } \mathcal{P}\right\}.$$

It will be convenient to prove strong duality for a generalization of SoS as a proof system over the real numbers, which we will discuss in Section 3.3.1. Therefore we will postpone the proof of Theorem 3.61 until Section 3.3.1.1, where we show that these are indeed formal SDP duals of one-another.

An immediate consequence of Theorems 3.61 along with Corollary 3.53 is an upper bound on the degree required in any SoS derivation.

Corollary 3.62. *If $P(\alpha) \geqslant c_0$ holds for every $\alpha \in \mathcal{P} \cap \{0,1\}^n$, then $P(x) \geqslant c_0$ has a degree $2n + \deg(\mathcal{P})$ SoS derivation from $\mathcal{P}$.*

3.2.3.2 Comparison With Sherali-Adams and Algebraic Proof Systems

Having argued that SoS is a formal proof system, it is natural to ask how SoS compares to other proof systems. That is, we are interested in how expressive low-degree (and also low-size) SoS refutations are compared to those of other proof systems.

To begin, it is straightforward to see that degree $2d$ SoS can p-simulate[13] degree d SA.

Lemma 3.63 (SoS p-simulates Sherali-Adams)**.** *Let $\mathcal{P}$ be a set of polynomial inequalities. If $P(x) \geqslant c_0$ has a degree d and size S Sherali-Adams derivation from $\mathcal{P}$, then it has a degree $2d$ and size $\mathsf{poly}(S)$ derivation in Sum-of-Squares.*

Proof. To prove Lemma 3.63, it is enough to show that any product of a non-negative $d - \deg(P_i)$-junta with an inequality $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{0 \geqslant 1\}$, $P_i(x) \cdot J_{S,T}(x) \geqslant 0$, has a degree $2d$ and size $\mathsf{poly}(n \cdot d)$ Sum-of-Squares proof. This follows because

$$P_i(x) \cdot (J_{S,T}(x))^2 \geqslant 0$$

has degree at most $2d$, and by Claim 3.57 can be multilinearized without increasing the degree. □

Therefore, SoS captures the reasoning power of SA. In fact, SoS proofs are strictly more expressive than the proofs produced by SA; there exists unsatisfiable systems of polynomial inequalities which have constant-degree refutations in SoS, but require degree $\Omega(n)$ to refute in SA. The standard example is the pigeonhole principle PHP^n_{n-1}, the propositional encoding that there is no injective map from $[n]$ to $[n-1]$. Dantchev et al. [51] showed that any SA refutation of PHP^n_{n-1} must have degree at least $n-1$. On the other hand, we argued in Claim 3.59 that SoS can prove PHP^n_{n-1} in constant degree and small size.

Nullstellensatz: In Example 3.56 we saw a low-degree SoS proof of the induction principle IND_n, a family of formulas that are known to require degree $\Theta(\log n)$ to refute in Nullstellensatz [38]. Buresh et al. [36] observed that a stronger separation can be obtained by looking at the pebbling contradictions, an unsatisfiable formula based on the black pebbling game. They showed that these formulas require degree $\Omega(n/\log n)$. On the other hand,

[13] Recall that p-simulation was defined in Definition 1.4.

these formulas have small degree and size refutations in SA, and therefore in SoS as well. Furthermore, it is not difficult to see that degree d SA p-simulates degree d Nullstellensatz; any derivation of degree d and size S in Nullstellensatz can be transformed into a degree $O(d)$ and size $\mathsf{poly}(S)$ derivation in SA. Taken together, this shows that Nullstellensatz is a strictly weaker system than SA.

Polynomial Calculus: Recall that the Polynomial Calculus is a dynamic (rule-based) version of Nullstellensatz.[14] Razborov [130] proved that the pigeonhole principle requires linear degree to refute PC. Together with Claim 3.59, this already separates PC from SoS in terms of degree. Furthermore, it is well known that there exists families of unsatisfiable formula that require degree $\Omega(n)$ and size $2^{\Omega(n)}$ for PC to refute, but which have constant-degree and polynomial-size SA (and therefore SoS) refutations; a simple example is the formula, $\sum_{i=1}^{n} x_i = n + 1$ [79]. Because of this, along with the fact that PC is dynamic, it was thought that SoS and PC were incomparable. In a surprising result, Berkholz [32] showed that SoS can in fact simulate PC.

Theorem 3.64 (Berkholz [32])**.** *For any unsatisfiable set of linear inequalities $\mathcal{P}$, if $\mathcal{P}$ has a degree d and size S refutation in PC, then it has a refutation of degree $2d$ and size $\mathsf{poly}(S)$ in SoS.*

This means that the cancellations of terms allowed between inferences in a PC refutation can be simulated in a one-shot SoS refutation. In the same work, Berkholz exhibits a formula that admits constant degree and polynomial size PC refutations, but for which any SA refutation requires proofs of size $2^{\Omega(n/\log n)}$ and degree $\Omega(n)$, showing that PC and SA are incomparable proof systems. This completes the picture of the relationships between these proof systems, which can be seen in Figure 3.5. For a more comprehensive discussion of the comparisons between the algebraic proof systems discussed so far, we refer the reader to the lovely exposition of Berkholz [32].

3.2.3.3 Automatizability

It is a common misconception that low degree SoS derivations can always be constructed efficiently, to high accuracy. Formally the claim is that if there exists a degree d derivation of $P(x)$ from a set of m inequalities $\mathcal{P}$, then there exists a deterministic algorithm running in time $m \cdot n^{O(d)}$ that will construct such a derivation up to some small additive error. That is, it is claimed that SoS is *degree-automatizable*. The reasoning behind this is as follows: As we will see in Section 3.3.1.1, finding a degree d SoS derivation can be phrased as an SDP of size $m \cdot n^{O(d)}$. If the coefficients of the polynomials in $\mathcal{P}$ and $P(x)$ have polynomial bit-length, then it is claimed that the ellipsoid method or interior point methods will solve this SDP in time $m \cdot n^{O(d)}$ and recover a valid derivation, up to some small additive error, if one exists.

Unfortunately this overlooks some of the subtleties of the ellipsoid method. Recall that the run-time of the ellipsoid method is polynomial so long as the spectahedron $\mathcal{S}$ is contained

[14] The Polynomial Calculus was defined in Section 1.3.

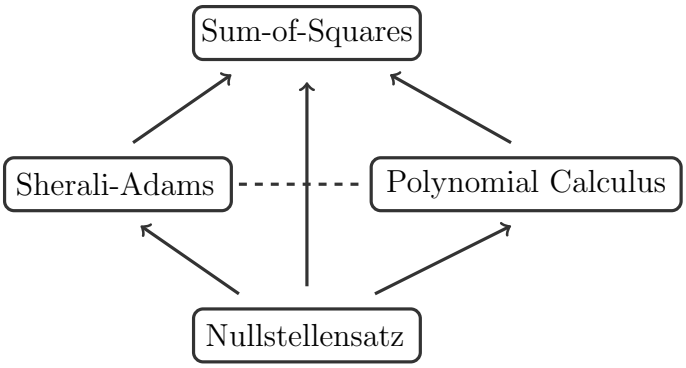


Figure 3.5: The relationship between SoS, SA, PC, Nullstellensatz. An arrow $P \to Q$ indicates that a proof in proof system P of size S and degree d can be converted into a proof of size $\mathsf{poly}(S)$ and degree $O(d)$, and that the converse does not hold. A dashed line between P and Q indicates that these proof systems are incomparable.

within some ball of radius R, where R is at most exponential in the size of the SDP. In other words, this constraint says that in order to ensure that a derivation can be found in polynomial time, it is not enough for such a derivation to exist, we need to guarantee that a derivation exists with polynomial bit-length. This is not always true; although both $P(x)$ and $\mathcal{P}$ have small coefficients, this does not guarantee any bound on the coefficients of the polynomials in a SoS derivation of $P(x)$ from $\mathcal{P}$, even for low-degree SoS derivations. O'Donnell [115] showed this in a strong sense by exhibiting a set of degree 2 polynomials $\mathcal{P}$ and $P(x)$ with constant coefficients such that any degree 2 derivation of $P(x)$ from $\mathcal{P}$ requires coefficients with exponential bit-length, even allowing for an additive error term in the conclusion. O'Donnell's proof holds even when the set of solutions to this SDP is *explicitly bounded* or *Archimedean*, a condition that implies that the SoS SDP has no duality gap[15]. Therefore, there exist well-behaved systems of polynomials which require exponential time for the ellipsoid method to even write down an approximate SoS derivation; the same issues plague interior point methods as well.

While O'Donnell's example does not admit a small degree 2 derivation, a degree 4 derivation with very small coefficients does exist. Therefore, this does not rule out the possibility that a slight increase in the degree could be sufficient to guarantee a derivation with small coefficients. Raghavendra and Weitz [126] extended O'Donnell's result, giving a set of quadratic polynomials that admit a degree 2 SoS derivation, but for which there is no derivation of degree $o(\sqrt{n})$ that has coefficients with polynomial bit-length. Like O'Donnell's example, their family of polynomials is Archimedean.

Along with their negative example, Raghavendra and Weitz [126] provide a set of sufficient conditions for guaranteeing that a set of polynomials $\mathcal{P}$ admits a degree d SoS derivation with small bit-length of a polynomial $P(x)$. Luckily, this set of conditions are satisfied by many of the applications of SoS, including, but not limited to MaxCSP, MaxClique, and

[15]The Archimedean condition and it's relationship with the duality of SoS will be discussed in Section 3.3.1.1.

BalancedSeparator. This extends an earlier result by O'Donnell [115], which shows that SoS derivations of polynomials have polynomial bit-length if the set of polynomials $\mathcal{P}$ contains only the Boolean axioms $\pm(x_i^2 - 1) \geqslant 0$.

It should be stressed that degree automatizability does not imply polynomial (size) automatizability in the traditional sense. Little is known about whether SoS proofs can be found efficiently in the *size* of the shortest proof. Indeed, the only known result is the trivial $m \cdot n^{O(n)}$-algorithm obtained by running the degree $2n$ SoS SDP.

In the case when the bit-length of a derivation is guaranteed to be bounded (for example, derivations from a set $\mathcal{P}$ that satisfies the conditions of Raghavendra and Weitz [126]), the $m \cdot n^{O(n)}$ size upper bound can be slightly improved to $m \cdot n^{O(\sqrt{n \log S_m} + \deg(\mathcal{P}))}$ by a size-degree trade-off for SoS. Define the *monomial size* S_m of a SoS proof to be the number of monomials in the proof expanded as a sum of monomials before any cancellations occur.

Lemma 3.65 (Size-Degree Trade-off for Sum-of-Squares [11])**.** *Let $\mathcal{P}$ be a set of m polynomial inequalities. Any SoS derivation of monomial size S_m from $\mathcal{P}$ implies a derivation of degree $O(\sqrt{n \log S_m} + \deg(\mathcal{P})$.*

3.3 Generalizations of Sum-of-Squares

3.3.1 Sum-of-Squares over $\mathbb{R}$

So far, we have only considered SoS as a proof system over the Boolean cube, enforced by including the axioms $x_i^2 = x_i$. It is natural (and often advantageous) to define SoS over other finite domains. This can be achieved by replacing the Boolean axioms $x_i^2 = x_i$ with axioms corresponding to this other domain (the most common being $\{\pm 1\}^n$, obtained by including $x_i^2 = 1$). However, we could consider a more general system where we omit these domain-restricting axioms altogether and instead define SoS as a proof system over the reals. That is, as a proof system for certifying that a family of polynomial inequalities share a common solution in $\mathbb{R}$. As we will see, this more general definition of SoS is closely related to the questions of Hilbert and Minkowski about the non-negativity of polynomials and their representation as a sum-of-squares polynomial that we discussed in Section 3.1.4. This section may be skipped at first reading as the latter chapters may be understood without it.

A SoS derivation over $\mathbb{R}$ of a polynomial inequality $P(x) \geqslant c_0$ from a set of polynomial inequalities $\mathcal{P}$ is a proof that $P(x) - c_0$ is non-negative over every solution $\alpha \in \mathbb{R}$ that satisfies $\mathcal{P}$. That is, it is a degree SoS derivation where we omit the axioms $x_i^2 = x_i$; it is a derivation of the form

$$P(x) - c_0 = Q_0(x) + \sum_{i=1}^{m} P_i(x) Q_i(x),$$

for $P_i(x) \geqslant 0 \in \mathcal{P}$ and $Q_0(x), Q_i(x) \in \Sigma^2$. We remark that over $\mathbb{R}$, it is no longer the case that linear degree is sufficient to refute any unsatisfiable system of polynomials. Grigoriev and Vorobjov [67] showed that the system of quadratic polynomials, known as the "telescopic

system"

$$1 - x_0 x_1 = 0, x_1^2 - x_2 = 0, x_2^2 - x_3 = 0, \ldots, x_{n-1}^2 - x_n = 0, x_n = 0 \tag{3.28}$$

requires degree 2^{n-1} to refute in SoS over $\mathbb{R}$.

Analogous to the Boolean case, we can define a pseudo-expectation for $\mathcal{P}$ as any linear operator that is non-negative over polynomials generated from sum-of-squares polynomials and the polynomials in $\mathcal{P}$. The only difference is that in this setting we no longer enforce that $x_i^2 = x_i$. Thus a pseudo-expectation over $\mathbb{R}$ for $\mathcal{P}$ is defined by relaxing the restriction that $\tilde{\mathbb{E}}$ is a multilinearizing map, to only requiring that it is a linear map in Definition 3.44. Denote by $\mathcal{E}_d^{\mathbb{R}}(\mathcal{P})$ the set of all degree d pseudo-expectations for $\mathcal{P}$ over $\mathbb{R}$.

The degree $2d$ pseudo-expectations over $\mathbb{R}$ can be re-interpreted as points belonging to an $n^d + 1$ spectahedron. Analogous to the Boolean case, the variables of this spectahedron represent terms of degree at most d in the original variables x. The difference is that we are no longer associating $x_i^c = x_i$, and therefore, we must have a variable y_I for every *multi-set* $I \subseteq [n]$ with $|I| \leqslant d$. This spectahedron is defined by $(n^d+1) \times (n^d+1)$-dimensional moment matrices, indexed by multi-sets $|I| \leqslant d$, where

$$\begin{aligned} (\mathcal{M}_d^{\mathbb{R}}(y))_{|I|,|J| \leqslant d} &:= y_{I \cup J}, \\ \mathcal{M}_d^{\mathbb{R}}(y, P_i)_{|I|,|J| \leqslant d - \deg(P_i)/2} &:= \sum_{|K| \leqslant \deg(P_i)} P_K y_{I \cup J \cup K}, \end{aligned}$$

where $I \cup J$ is now multi-set union. Optimizing a polynomial $P(x)$ over $\mathcal{P}$ can be approximated by solving the SDP with objective function $\sum_{|K| \leqslant \deg(P)} \vec{P}_K y_K$, and constraints $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$ defined by this spectahedron.

Remark 3.66. Using this general framework, we can define SoS over other domains by imposing restrictions on this general form. The most common domain, besides the $\{0,1\}$, is to work over $\{\pm 1\}$. This can be done by including the axioms $\pm(x_i^2 \geqslant 1)$ in the set of inequalities. The resulting moment matrix is of dimension $\binom{n}{\leqslant d} \times \binom{n}{\leqslant d}$, where multilinearization is done by associating x_i^2 with the constant 1, and so the (I, J)-th entry is $y_{I \triangle J}$.

3.3.1.1 Duality, Completeness, and Convergence

When discussing SoS over the Boolean cube it was fairly straightforward to show completeness for both the refutational and SDP perspectives of SoS. In particular, we showed that the SoS SDP is guaranteed to converge after taking a high enough lift. Unfortunately, for this more general system, completeness is not known to hold in general. Fortunately, under mild assumptions on the set $\mathcal{P}$ of initial inequalities, completeness can be made to hold. The proof in this general case is much more involved, and we will cover it in length over the following two sub-sections. First, we show that SoS as a proof system and SoS as a hierarchy of SDPs are in fact formal SDP dual of one another. Furthermore, under an assumption on the structure of $\mathcal{P}$, strong duality holds. Combined with the fact that the SoS hierarchy

over the Boolean cube is guaranteed converges to the integer hull, this immediately implies the derivational completeness of SoS over the Boolean cube. To prove convergence and completeness for SoS over $\mathbb{R}$ we will instead rely on a theorem from semialgebraic geometry, a variant of the *Positivstellensatz* discovered by Putinar [123].

Weak Duality The first step in our proof of completeness is to show duality between the proof system and optimization views of SoS. Throughout this sub-section, we will be working with SoS over $\mathbb{R}$. Therefore, unless stated otherwise, all sets of indices $S \subseteq [n]$ should be assumed to be multi-sets. First, we show that the task of coming up with a degree $2d$ SoS refutation of a set of polynomials $\mathcal{P}$ amounts to solving a corresponding SDP, and furthermore that this SDP is the formal dual of the level d SoS hierarchy applied to $\mathcal{P}$. This immediately establishes a weak duality between SoS proofs and the SDPs generated by the SoS hierarchy.

Theorem 3.67. *For a polynomial $P(x)$ and a set of polynomial inequalities $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$, the following programs are a formal SDP primal/dual pair:*

Dual:	Primal:
$\min_y \ \sum_I P_I y_I$	$\max \lambda$
$y_\emptyset = 1$	$P(x) - \lambda = \sum_{i=1}^m Q_i(x) P_i(x)$
$\mathcal{M}_d^{\mathbb{R}}(y, P_i) \succeq 0 \quad \forall i \in [m]$	$P_i(x) \geqslant 0 \in \mathcal{P}$
$\mathcal{M}_d^{\mathbb{R}}(y) \succeq 0$	$Q_i(x) \in \Sigma^2_{d-\deg(P_i)}$

Proof. To begin, we will rewrite the d-th level of the SoS hierarchy in the standard form of a dual SDP by rephrasing the constraints as a single matrix inequality. First, replace every constant term c occurring in an entry of $\mathcal{M}_d^{\mathbb{R}}(y)$ and $\mathcal{M}_d(y, P_i)$ by $c \cdot y_\emptyset$. For example, if one of the entries in the matrix $\mathcal{M}_d^{\mathbb{R}}(y, P_i)$ is $y_J + y_I + 5$, we replace it by $y_J + y_I + 5 \cdot y_\emptyset$. This is an equivalent reformulation, because we have enforced that $y_\emptyset = 1$. As well, we can write $y_\emptyset = 1$ as a matrix inequality. Define the 2×2 matrix $M_\emptyset$ as

$$M_\emptyset := \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} + y_\emptyset \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{3.29}$$

Constraining $y_\emptyset = 1$ is equivalent to requiring that $M_\emptyset \succeq 0$, because the diagonal entries of any PSD matrix are non-negative.

The standard form of a dual-SDP contains only a single constraint. Therefore, create the following block diagonal matrix F:

$$F := \begin{bmatrix} M_\emptyset & 0 & 0 & 0 & 0 \\ 0 & \mathcal{M}_d^{\mathbb{R}}(y) & 0 & 0 & 0 \\ 0 & 0 & \mathcal{M}_d^{\mathbb{R}}(y, P_1) & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & \mathcal{M}_d^{\mathbb{R}}(y, P_m) \end{bmatrix}$$

Observe that $F \succeq 0$ if and only if each of the diagonal blocks are symmetric PSD. Therefore, $F \succeq 0$ is equivalent to enforcing the constraints of $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$. We will now decompose the matrix F as

$$F = F_c + \sum_{|I| \leqslant 2d} y_I F_I,$$

where the y_I are variables as usual and the F_I are matrices whose entries are all constant. Here, F_c accounts for any additive constants in F. Note that F_c is only non-zero within the submatrix of F corresponding to $M_\emptyset$ because we replaced constants c with $c \cdot y_\emptyset$ in $\mathcal{M}_d^{\mathbb{R}}(y)$ and $\mathcal{M}_d^{\mathbb{R}}(y, P_i)$. With this, we can restate the SoS SDP in the standard form of an SDP dual,

$$\begin{aligned} \min_y \quad & \sum_I \vec{P}_I y_I, \\ \text{s.t.} \quad & F_c + \sum_{|I| \leqslant 2d} y_I F_I \succeq 0. \end{aligned}$$

Taking the dual of this SDP, we obtain the primal

$$\begin{aligned} \max_Z \quad & -F_c \bullet Z \\ \text{s.t.} \quad & F_I \bullet Z = \vec{P}_I \qquad \forall |I| \leqslant 2d \\ & Z \succeq 0. \end{aligned} \tag{3.30}$$

It remains to show that we can rewrite this SDP as a degree d SoS refutation. First, observe that because F is a block diagonal matrix, so are the F_I. Because of this, we can assume w.l.o.g. that Z is a block diagonal matrix of the same form. Break Z, F_I and F_c into submatrices, one corresponding to each block. Denote these submatrices $Z_{-1}, \ldots, Z_m$, $F_I^{(-1)}, \ldots, Z_I^{(m)}$, and $F_c^{(-1)}, \ldots, F_c^{(m)}$.

$$F_I := \begin{bmatrix} F_I^{(-1)} & 0 & 0 & 0 \\ 0 & F_I^{(0)} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & F_I^{(m)} \end{bmatrix} \qquad Z := \begin{bmatrix} Z_{-1} & 0 & 0 & 0 \\ 0 & Z_0 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & Z_m \end{bmatrix}$$

Using this block decomposition, we can write (3.30) equivalently as

$$\begin{aligned} \max_{Z_{-1}, \ldots, Z_m} \quad & -\sum_{i=-1}^{m} F_c^{(i)} \bullet Z_i, \\ \text{s.t.} \quad & F_I^{(-1)} \bullet Z_{-1} + \sum_{i=0}^{m} F_I^{(i)} \bullet Z_i = \vec{P}_I \qquad \forall |I| \leqslant 2d, \\ & Z_i \succeq 0 \qquad \forall i \in \{-1, 0, \ldots, m\}. \end{aligned} \tag{3.31}$$

Because $F = F_c + \sum_{|I|\leqslant 2d} y_I \cdot F_I$, we have $\mathcal{M}(y, P_i) = \sum_{|J|\leqslant 2d} y_J \cdot F_J^{(i)}$. Recall that by definition $\mathcal{M}(y, P_i)_{|S|,|T|\leqslant d-\deg(P_i)/2} = \sum_{|K|\leqslant \deg(P_i)} \vec{P}_K \cdot y_{S\cup T\cup K}$. For ease of notation, we will denote by $\mathcal{M}(y, P_0)$ the matrix $\mathcal{M}(y)$ where $P_0(x) := 1$. Therefore,

$$F_I^{(i)} \bullet Z_i = \sum_{\substack{|S|,|T|\leqslant d-\deg(P_i)/2,\ |K|\leqslant \deg(P_i) \\ S\cup T\cup K = I}} (\vec{P}_i)_K (Z_i)_{S,T}, \tag{3.32}$$

for all $i \neq -1$. For $i = -1$, because $y_\emptyset$ is the only variable occurring in $M_\emptyset$, the matrices $F_I^{(-1)} = 0$ for all $I \neq \emptyset$. Furthermore, by the definition of $M_\emptyset$ (3.29),

$$F_\emptyset^{(-1)} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \text{ and so } F_\emptyset^{(-1)} \bullet Z_{-1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} Z_{\{1,1\}} & Z_{\{2,1\}} \\ Z_{\{1,2\}} & Z_{\{2,2\}} \end{pmatrix} = Z_{\{1,1\}} - Z_{\{2,2\}} \tag{3.33}$$

where $Z_{\{i,j\}}$ is the $\{i,j\}$-th entry of the matrix Z.

Next, we rephrase the objective function $-\sum_{i=-1}^m F_c^{(i)} \bullet Z_i$. Recall that F_c is zero outside of the block $F_c^{(-1)}$ corresponding to the matrix $M_\emptyset$. Combining this with (3.33) we have

$$-\sum_{i=-1}^m F_c^{(i)} \bullet Z_i = F_c^{(-1)} \bullet Z_{-1} = -\left(-Z_{\{1,1\}} + Z_{\{2,2\}}\right), \tag{3.34}$$

Putting together (3.32), (3.33), and (3.34), we can rewrite the primal SDP (3.31) as

$$\begin{aligned} \max_{Z_0,\ldots,Z_m,Z_{\{1,1\}},Z_{\{2,2\}}} \quad & Z_{\{1,1\}} - Z_{\{2,2\}} \\ \text{s.t.} \quad & Z_{\{1,1\}} - Z_{\{2,2\}} + \sum_{i=0}^m \left((\vec{P}_i)_\emptyset (Z_i)_\emptyset\right) = \vec{P}_\emptyset \\ & \sum_{i=0}^m \left(\sum_{\substack{|S|,|T|\leqslant d-\deg(P_i)/2,\ |K|\leqslant \deg(P_i) \\ S\cup T\cup K = I}} (\vec{P}_i)_K (Z_i)_{S,T} \right) = \vec{P}_I \qquad \forall\, 0 < |I| \leqslant 2d \\ & Z_i \succeq 0 \qquad \forall\, i \in \{0,\ldots,m\} \\ & Z_{\{1,1\}}, Z_{\{2,2\}} \geqslant 0. \end{aligned} \tag{3.35}$$

Because the only constraints involving $Z_{\{1,1\}}, Z_{\{2,2\}}$ are the $\vec{P}_\emptyset$ constraint and the non-negativity constraints, we can replace $Z_{\{1,1\}} - Z_{\{2,2\}}$ by a single unconstrained variable λ. Letting e be the unit vector where $e_\emptyset = 1$ and $e_I = 0$ for all $I \neq \emptyset$ allows us to write (3.35) as

$$\begin{aligned} \max_{Z_0,\ldots,Z_m,\lambda} \quad & \lambda \\ \text{s.t.} \quad & \vec{P}_I - \lambda e_I = \sum_{i=0}^m \left(\sum_{\substack{|S|,|T|\leqslant d-\deg(P_i)/2, |K|\leqslant \deg(P_i) \\ I\cup J\cup K = S}} (\vec{P}_i)_K (Z_i)_{S,T} \right) \quad \forall |I| \leqslant 2d \end{aligned} \tag{3.36}$$

$$Z_i \succeq 0 \qquad \forall i \in \{0, \dots, m\}$$

For every $I \subseteq [n]$ with $|I| \leqslant 2d$, multiply both sides of the corresponding constraint in (3.36) by $\prod_{i \in I} x_i$. This results in

$$\vec{P}_I \prod_{i \in I} x_i - \lambda e_I \prod_{i \in I} x_i = \sum_{i=0}^{m} \left(\sum_{\substack{|S|,|T| \leqslant d - \deg(P_i)/2, |K| \leqslant \deg(P_i) \\ S \cup T \cup K = I}} (\vec{P}_i)_K (Z_i)_{S,T} \prod_{i \in I} x_i \right), \tag{3.37}$$

for all $|I| \leqslant 2d$. Recall that $e_I \lambda = 0$ unless $I = \emptyset$. Sum the equations in (3.37), and note that the constraint that results is equivalent to the set of equations in (3.36) because the variables occurring in the equation for each $I \subseteq [n]$ are distinct.

$$\begin{aligned}
P(x) - \lambda &= \sum_{i=0}^{m} \left(\sum_{\substack{|S|,|T| \leqslant d - \deg(P_i)/2, \\ |K| \leqslant \deg(P_i)}} (\vec{P}_i)_K (Z_i)_{S,T} \prod_{i \in S \cup T \cup K} x_i \right), \\
&= \sum_{i=0}^{m} \left(\sum_{|K| \leqslant \deg(P_i)} (\vec{P}_i)_K \prod_{i \in K} x_i \right) \left(\sum_{|S|,|T| \leqslant d - \deg(P_i)/2} (Z_i)_{S,T} \prod_{j \in S \cup T} x_j \right), \\
&= \sum_{i=0}^{m} P_i(x) \left(\sum_{|S|,|T| \leqslant d - \deg(\vec{P}_i)/2} (Z_i)_{S,T} \prod_{j \in S \cup T} x_j \right).
\end{aligned}$$

Let $v_k(x)$ be the degree at most k monomial vector, where $v_k(x)_I = \prod_{i \in I} x_i$ for all $|I| \leqslant d$, and denote $d - \deg(P_i)/2$ by d_i. Then, this becomes

$$P(x) - \lambda = \sum_{i=0}^{m} \vec{P}_i(x) v_{d_i}(x)^\top Z_i v_{d_i}(x).$$

Because $Z_i \succeq 0$ it admits a Cholesky decomposition, $Z_i = U_i^\top U_i$, giving

$$P(x) - \lambda = \sum_{i=0}^{m} P_i(x) v_{d_i}(x)^\top U_i^\top U_i v_{d_i}(x) = \sum_{i=1}^{m} P_i(x) (U_i v_{d_i}(x))^2.$$

Define the polynomial $Q_i(x) := (U_i v_{d_i}(x))_i$, then,

$$\begin{aligned}
P(x) - \lambda &= \sum_{i=0}^{m} P_i(x) v_{d_i}(x)^\top U_i^\top U_i v_{d_i}(x), \\
&= Q^2(x) + \sum_{i=1}^{m} P_i(x) Q_i^2(x),
\end{aligned}$$

where the final line follows because we defined $P_0(x) = 1$. Note that $\deg(Q) \leqslant d$ and $\deg(Q_i) \leqslant d - \deg(P_i)$. This completes the proof that degree $2d$ SoS derivations over $\mathcal{P}$ are the dual objects to $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$. □

The same argument can be used to show that degree $2d$ SoS derivations over $\{0,1\}$ and $\mathsf{SOS}_d(\mathcal{P})$ are dual objects. It follows immediately that weak duality holds for SOS.

Corollary 3.68 (Weak Duality for SOS)**.** *Let $\mathcal{P}$ be a set of polynomial inequalities. For any $P(x) \in \mathbb{R}[x]$,*

$$\min\left\{\tilde{\mathbb{E}}[P(x)] : \tilde{\mathbb{E}} \in \mathcal{E}_d^{\mathbb{R}}(\mathcal{P})\right\} \geqslant \max\left\{c_0 : \exists \text{ a degree } d \text{ SoS}^{\mathbb{R}} \text{ derivation of } P(x) \geqslant c_0 \text{ from } \mathcal{P}\right\}.$$

Strong Duality Recall that strong duality for SDPs holds only if certain robustness conditions, (Theorem 3.26), are satisfied by the SDP. Unfortunately, SoS is in a similar situation. Strong duality and completeness hold only when certain compactness conditions are satisfied. Luckily these conditions hold vacuously for SoS over the Boolean cube. The standard criteria for ensuring that strong duality holds for SoS is that the initial set of polynomials $\mathcal{P}$ (including any axioms such as $x_i^2 = x_i$) as is *Archimedean*.

Assumption 3.69 (Archimedean)**.** *The set of polynomials $\mathcal{P}$ contains a constraint of the form*

$$r^2 - \sum_{i=1}^{n} x_i^2 \geqslant 0.$$

In words, this condition requires that the set of feasible solutions $\{\alpha \in \mathbb{R}^n : P_i(\alpha) \geqslant 0,\ \forall P_i(x) \geqslant 0 \in \mathcal{P}\}$ is contained within a Euclidean ball of radius r. It is not hard to see that this criteria is equivalent to enforcing that the set of feasible solutions is compact. Observe that SoS over both $\{0,1\}$ and $\{-1,1\}$ trivially satisfies the Archimedean condition. Josz and Henrion [81] proved that strong duality holds for SoS under the Archimedean assumption. For completeness, we repeat their proof here.

Theorem 3.70 (Strong Duality for SoS over $\mathbb{R}$)**.** *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be an Archimedean set of polynomial inequalities. Then, for any $P(x) \in \mathbb{R}[x]$,*

$$\min\left\{\tilde{\mathbb{E}}[P(x)] | \tilde{\mathbb{E}} \in \mathcal{E}_{2d}^{\mathbb{R}}(\mathcal{P})\right\} = \max\left\{c_0 | \ \exists \text{ a degree } 2d \text{ SoS}^{\mathbb{R}} \text{ derivation of } P(x) \geqslant c_0 \text{ from } \mathcal{P}\right\}.$$

Proof. For simplicity we will assume that $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$ is non-empty, as Theorem 3.60 handles the case when it is empty over SoS over the Boolean cube. We refer the reader to [81] which handles the case when $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P}) = \emptyset$. Corollary 3.27 states that if the set of optimal solutions to $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$ is non-empty and bounded then strong duality holds. By the Archimedean assumption, the solutions to $\mathcal{P}$ are contained within a Euclidean ball of radius r^2. We claim that this implies that $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$ is contained within a Euclidean ball of radius $R = r^{O(d)}$. Assuming that this claim holds, because $\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P})$ is bounded and the objective function $\sum_{|I| \leqslant 2d} \vec{P}_I y_I$ is linear, the set of optimal solutions to $\mathsf{SOS}_d(\mathcal{P})$ is non-empty and bounded. By Corollary 3.27, strong duality holds for SoS.

To prove the claim, we show that every solution $\alpha \in \mathsf{SOS}_d(\mathcal{P})$ satisfies $\|(\alpha_I)_{|I| \leqslant d}\|_2 \leqslant R$, a Euclidean ball of radius R. Observe that

$$\|(y_I)_{|I| \leqslant 2d}\|_2 = \sqrt{\sum_{|I| \leqslant 2d} y_I} \leqslant \sum_{|I| \leqslant 2d} y_I = \mathsf{Tr}[\mathcal{M}_d(y)].$$

Therefore, it is sufficient to bound $\mathsf{Tr}[\mathcal{M}_d(y)]$, noting that because $\mathcal{M}_d(y) \succeq 0$, the diagonal entries of $\mathcal{M}_d(y)$ are non-negative by Claim3.5. Let $\ell \leqslant d$ and let $A(x) := r^2 - \sum_{i=1}^n x_i^2 \geqslant 0 \in \mathcal{P}$ be the constraint guaranteed by the Archimedean assumption. Then,

$$\mathcal{M}_\ell(y, A) = \left(\sum_{|K| \leqslant 2} (\vec{A})_K \cdot y_{I \cup J \cup K} \right)_{|I|,|J| \leqslant \ell - 1} = \left(r^2 \cdot y_{I \cup J} - \sum_{i \in [n]} y_{I \cup J \cup \{i,i\}} \right)_{|I|,|J| \leqslant \ell - 1}.$$

Now, observe that

$$\begin{aligned}
\mathsf{Tr}[\mathcal{M}_{\ell-1}(y, A)] &= \sum_{|I| \leqslant \ell - 1} \Big(r^2 \cdot y_{I \cup I} - \sum_{i \in [n]} y_{I \cup I \cup \{i,i\}} \Big), \\
&= r^2 \sum_{|I| \leqslant \ell - 1} y_{I \cup I} - \sum_{|I| \leqslant \ell - 1, |J| = 1} y_{I \cup I \cup J \cup J}, \\
&= r^2 \mathsf{Tr}\left[\mathcal{M}_{\ell-1}(y)\right] + y_\emptyset - \Big(\sum_{|I| \leqslant \ell} y_{I \cup I} + y_\emptyset \Big), \\
&= r^2 \mathsf{Tr}\left[\mathcal{M}_{\ell-1}(y)\right] + 1 - \mathsf{Tr}\left[\mathcal{M}_\ell(y)\right],
\end{aligned}$$

where the final line follows because we enforce $y_\emptyset = 1$. Rearranging the above equation, we have

$$\begin{aligned}
\mathsf{Tr}\left[\mathcal{M}_\ell(y)\right] &= r^2 \mathsf{Tr}\left[\mathcal{M}_{\ell-1}(y)\right] + 1 - \mathsf{Tr}\left[\mathcal{M}_{\ell-1}(y, A)\right], \\
&\leqslant r^2 \mathsf{Tr}\left[\mathcal{M}_{\ell-1}(y)\right] + 1.
\end{aligned}$$

We can express $\mathsf{Tr}[\mathcal{M}_\ell(y)]$ in terms of R alone by unrolling this expression,

$$\mathsf{Tr}[\mathcal{M}_1(y)] \leqslant r^2 + 1, \quad \mathsf{Tr}[\mathcal{M}_2(y)] \leqslant r^4 + r^2 + 1, \ \ \ldots \ \ , \mathsf{Tr}[\mathcal{M}_d(y)] \leqslant \sum_{i=0}^{d} r^{2d}.$$

Therefore, letting $R := \sum_{i=0}^d r^{2d}$, we have $\|(y_I)_{|I| \leqslant 2d}\|_2 \leqslant R$. □

Completeness and Convergence This strong duality theorem is already enough to complete the proof of derivational and refutational completeness for SoS over the Boolean cube (Theorem 3.61). Indeed, observe that the Archimedean condition is already satisfied by the axioms $x_i^2 = x_i$. For SoS over $\mathbb{R}$, completeness as and convergence follow from a special case of the Positivstellensatz, a fundamental theorem in semi-algebraic geometry. This special case was discovered by Putinar [123].

Theorem 3.71 (Putinar's Theorem)**.** *Let $\mathcal{P}$ be a set of polynomial inequalities that satisfy the Archimedean condition and $P(x) \in \mathbb{R}[x]$. Then $P(\alpha) > 0$ for all $\alpha \in \mathcal{P}$* [16] *if and only if there exists there exists sum-of-squares polynomials $Q_0(x), Q_1(x), \ldots, Q_m(x) \in \Sigma^2$ such that*

$$P(x) = Q_0(x) + \sum_{i=1}^{m} P_i(x) Q_i(x). \tag{3.38}$$

[16] Recall that we are abusing notation and associating $\mathcal{P}$ with the set of polynomial inequalities it contains, as well as with the set of points $\alpha \in \mathbb{R}^n$ for which $P_i(\alpha) \geqslant 0$ for every polynomial inequality $P_i(x) \geqslant 0 \in \mathcal{P}$.

Putinar's Theorem establishes refutational completeness for SoS over $\mathbb{R}$. To see this, observe that if $\mathcal{P}$ is unsatisfiable, then any polynomial, including -1, is positive over $\mathcal{P}$. Putinar's Theorem implies that -1 can be written as

$$-1 = Q_0(x) + \sum_{i=1}^{m} P_i(x)Q_i(x),$$

for sum-of-squares polynomials $Q_0(x), \ldots, Q_m(x) \in \Sigma^2$.

As observed by Lasserre [103], Putinar's Theorem immediately implies the convergence of the SoS SDP hierarchy to the true optimal solution.

Corollary 3.72 (Convergence of SoS over $\mathbb{R}$)**.** *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0$ be an Archimedean set of polynomial inequalities and let $P(x) \in \mathbb{R}[x]$ be any polynomial. Then*

$$\lim_{d\to\infty} \mathcal{SDP}(\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P}), P) = \min_{\alpha\in\mathcal{P}} P(\alpha).$$

Proof. Let $\alpha^* \in \mathcal{P}$ be such that $P(\alpha^*) := \min_{\alpha\in\mathcal{P}} P(\alpha)$. Then $P_i(\alpha^*) \geqslant 0$ for all $P(x) \geqslant 0 \in \mathcal{P}$. Let $\varepsilon > 0$, and define the polynomial $P(x) - P(\alpha^*) + \varepsilon$. Observe that $P(\alpha) - P(\alpha^*) + \varepsilon > 0$ for every $\alpha \in \mathcal{P}$. By Putinar's Theorem there exists an integer $d > 0$ and sum-of-squares polynomials $Q_0(x), \ldots, Q_m(x) \in \Sigma^2_{2d-\deg(P_i)}$ such that

$$P(x) - P(\alpha^*) + \varepsilon = Q_0(x) + \sum_{i=1}^{m} P_i(x)Q_i(x).$$

In particular this implies that $P(\alpha^*) - \varepsilon$ is a feasible solution to the SoS primal SDP

$$\begin{aligned} &\max \lambda \\ P(x) - \lambda = \sum_{i=1}^{m} Q_i(x)P_i(x) \quad &\text{s.t. } P_i(x) \geqslant 0 \in \mathcal{P}, Q_i(x) \in \Sigma^2_{2d-\deg(P_i)}. \end{aligned}$$

Denote by α_{prim} the optimal solution to this SDP; we have that $\alpha_{prim} \geqslant P(\alpha^*) - \varepsilon$. Because $\mathcal{P}$ is Archimedean, strong duality holds, and so $\mathcal{SDP}(\mathsf{SOS}_d^{\mathbb{R}}(\mathcal{P}), P) = \alpha_{prim} \geqslant P(\alpha^*) - \varepsilon$ is also a feasible solution to $\mathsf{SOS}_d(\mathcal{P})$. Taking $\varepsilon \to 0$ completes the proof. □

3.3.2 Positivstellensatz

3.3.2.1 Positivstellensatz Refutations

When the set of polynomial inequalities is non-Archimedean it is known that Putinar's Theorem no longer holds. Furthermore, Scheiderer [137] shows that there are fundamental obstructions to having such sum-of-squares representations in the non-Archimedean case. Although we are typically only concerned with sets where the Archimedean assumption is

trivially satisfied (such as the Boolean cube or $\{\pm 1\}^n$) this is still an unfortunate circumstance, and one might wonder how much the structure of these refutations must be generalized before we are able to have completeness for any set of polynomials that are unsatisfiable over $\mathbb{R}$.

The Positivstellensatz, a generalization of Hilbert's Nullstellensatz that was discovered by Krivine [101] and Stengle [145], shows that in order to obtain a proof system that is complete over $\mathbb{R}$, it is enough to allow products of the initial polynomials. Below, we state a refutational version of the Positivstellensatz.

Theorem 3.73 (Refutational Positivstellensatz)**.** *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be a set of polynomial inequalities. Then, $\mathcal{P}$ is unsatisfiable if and only if there exists sum-of-squares polynomials $Q_\ell(x)$ for $\ell \in \{0,1\}^m$ such that*

$$-1 = \sum_{\ell \in \{0,1\}^m} Q_\ell(x) P_1^{\ell_1}(x) \cdots P_m^{\ell_m}(x). \tag{3.39}$$

Therefore, if we allow products of the initial polynomials as well as square polynomials, we obtain a proof system which is complete over $\mathbb{R}$. This was proposed as a propositional proof system by Grigoriev and Vorobjov [67].

Although this proof system may appear to be significantly more expressive than SoS, lower bounds on the degree of Positivstellensatz refutations for several formulas are known. In their initial work, Grigoriev and Vorobjov [67] showed that the telescopic system (3.28) requires Positivstellensatz refutations of degree at least 2^{n-1}. Building on earlier ideas of Buss et al. [37], Grigoriev [65] showed that the degree required to refute any unsatisfiable family of $\mathbb{F}_2$ linear equations is at least the minimum width of any Resolution refutation of these equations. Applying this, he showed that several natural families of polynomial-size CNF formulas, including the Tseitin tautologies and random 3XOR, require Positivstellensatz refutations of degree $\Omega(n)$. We describe this proof in full detail in Section 5.1 for SoS.

Parillo [116, 117] and Lasserre [102, 103], building on the earlier ideas of Shor [143], observed that the task of finding bounded degree Positivstellensatz refutations can be phrased as an SDP. This is done by rephrasing

$$-1 = \sum_{\ell \in \{0,1\}^m} Q_\ell(x) P_1^{\ell_1}(x) \cdots P_m^{\ell_m}(x)$$

as an SDP in much the same way as was done for SoS in Section 3.3.1.1. As usual, the variables of this SDP are the coefficients of the polynomials. However, a drawback is that general Positivstellensatz refutations involve 2^m terms, one for each $\ell \in \{0,1\}^m$, and therefore result in unfortunately large SDPs. To overcome this issue, Lasserre proposed to instead use Putinar's variant of the Positivstellensatz, resulting in the SoS hierarchy.

3.3.2.2 Positivstellensatz Certificates

So far we have only discussed Positivstellensatz as a refutational system. In fact, the Positivstellensatz was studied from a derivational perspective, beginning with the question of

how to certify the non-negativity of a single polynomial over the reals. Recall from Section 3.1.4, that Hilbert and Motzkin disproved the conjecture that every non-negative polynomial over $\mathbb{R}$ can be written as a sum-of-squares. Indeed, Motzkin gave an explicit counter example of a polynomial that cannot be written as a sum-of-squares, but which is non-negative over $\mathbb{R}$. Motzkin's counter example already shows that, unlike Positivstellensatz refutations, if we are to have a derivational form of the Positivstellensatz it is not enough to allow square polynomials as well as products of the initial polynomial inequalities.

The proof that the Motzkin polynomial is non-negative (Claim 3.34) involved expressing it as a *sum-of-squares of rational functions*. Motivated by this observation, in 1900 Hilbert asked as his 17th problem to the Congress of Mathematicians.

Hilbert's 17th Problem: Can any non-negative polynomial $P(x) \in \mathbb{R}[x]$ be written as a sum of squares of rational functions?

That is, for every non-negative polynomial $P(x) \in \mathbb{R}[x]$, does there exist polynomials $P_1(x), \ldots, P_k(x), Q_1(x), \ldots, Q_k(x) \in \mathbb{R}[x]$ such that

$$P(x) = \sum_{i=1}^{k} \left(\frac{P_i(x)}{Q_i(x)} \right)^2.$$

In 1927, Emil Artin [9] settled the question.

Theorem 3.74 (Artin's Theorem)**.** *If $P(x) \in \mathbb{R}[x]$ is non-negative over $\mathbb{R}$ then $P(x)$ can be written as a sum-of-squares of rational functions*

This result was later extended by Krivine [101] and Stengle [145] from the non-negativity of a polynomial over $\mathbb{R}$ to non-negativity over semi-algebraic sets, resulting in what has come to be known as the *Positivstellensatz*. As usual, we will abuse notation and denote by $\mathcal{P}$ a set of polynomial inequalities, as well as the set of points $\{\alpha \in \mathbb{R}^n : P(\alpha) \geqslant 0,\ \forall P(x) \geqslant 0 \in \mathcal{P}\}$.

Theorem 3.75 (Positivstellensatz)**.** *Let $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$ be a set of polynomial inequalities and $P(x) \in \mathbb{R}[x]$, then*

1. *$P(\alpha) \geqslant 0$ for all $\alpha \in \mathcal{P}$ if and only if there exists $k \in \mathbb{N}$, and sum-of-squares polynomials $Q_\ell(x)$ and $Q'_\ell(x)$ for $(\ell_1, \ldots, \ell_m) = \ell \in \{0,1\}^m$ such that*

$$P(x) = \left(\frac{P(x)^{2k} + \sum_{\ell \in \{0,1\}^m} Q_\ell(x) P_1^{\ell_1}(x) \ldots P_m^{\ell_m}(x)}{\sum_{\ell \in \{0,1\}^m} Q'_\ell(x) P_1^{\ell_1} \ldots P_m^{\ell_m}(x)} \right). \tag{3.40}$$

2. *$\mathcal{P}$ is unsatisfiable if and only if there exists sum-of-squares polynomials $Q_\ell(x)$ for $\ell \in \{0,1\}^m$ such that*

$$-1 = \sum_{\ell \in \{0,1\}^m} Q_\ell(x) P_1^{\ell_1}(x) \cdots P_m^{\ell_m}(x). \tag{3.41}$$

Observe that setting $\mathcal{P} = \{1 \geqslant 0\}$ in Theorem 3.40 recovers Artin's Theorem. In particular, the Positivstellensatz derivation given in (3.41) is a generalization of sum-of-squares of rational functions.

The Positivstellensatz can be interpreted as follows: either a polynomial $P(x)$ is non-negative over $\mathcal{P}$, in which case there is a derivation of $P(x)$ of the form (3.41), or $\mathcal{P} \cup \{P(x) \geqslant 0\}$ has no common solution, in which case there is a Positivstellensatz refutation. The Positivstellensatz derivations are of a more general form than the refutations; the Motzkin polynomial shows that this generality is indeed necessary for completeness.

Chapter 4

Upper Bounds via Sum-of-Squares

In this chapter, we will present a glimpse of the algorithmic applications of the SoS proof system. Classical proof complexity has focused on studying proof systems that restrict NP in various ways. This opens the door to establishing lower bounds that can verify conjectured complexity class separations such as $\mathsf{P} \neq \mathsf{coNP}$ in a restricted setting. The realization that the proof complexity perspective could be useful in obtaining new algorithmic results is a much more recent and modern development. We start with a brief overview of some of the major lines of research inquiries in this direction.

Algorithmically, the SoS method simply relies on the observation that degree d pseudo-expectations in n variables can be (approximately) computed in $n^{O(d)}$ time. For a host of natural combinatorial problems, the special case of degree 2 turns out to be a well-studied and simple SDP relaxation. While such semidefinite programs were already used in clever combinatorial arguments (see for e.g. [108]), they led to an algorithmic renaissance with the celebrated work of Goemans and Williamson [60] that gave an algorithm to compute a 0.878 approximate MaxCut in any graph. In doing so, they introduced the *hyperplane rounding* technique. This technique was used and extended upon in a burgeoning body of results that derived improved algorithms for several combinatorial optimization problems [41, 3, 10, 13, 104].

The use of higher degree SoS relaxations first appeared in a breakthrough work of Arora, Rao and Vazirani [7] that gave an efficient algorithm for computing $O(\sqrt{\log n})$ approximation for least expanding sets in graphs. Their algorithm uses a semi-definite programming relaxation proposed by Goemans and Linial [59, 106] that is closely related to (and captured by) the level 2 SoS relaxation combined with a new rounding technique inspired by the theory of low-distortion metric embeddings.

Starting with the work of Khot, Kindler, Mossell and O'Donnell [90], a deep connection was discovered between algorithms based on such SDPs and Khot's Unique Games Conjecture (UGC) [87]. This sequence of works culminated in a famous result of Raghavendra [124] that showed that a natural analog of Goemans-Williamson's algorithm is optimal, in the worst-case, for all constraint satisfaction problems, assuming the UGC.

In a surprising twist, Arora, Barak and Steurer [4] discovered a sub-exponential time algorithm for the Unique Games problem itself. Soon after, in independent works, Barak,

Raghavendra and Steurer [23] and Guruswami and Sinop [71] gave a principled view of this algorithm as an instance of degree n^δ SoS. This relied on developing the *global correlation rounding* method that in turn led to new subexponential algorithms for several related problems. In the following years, Barak, Kelner and Steurer [19] and Barak, Kothari, Steurer [21] gave new rounding methods that used higher degree SoS algorithms to obtain new results for polynomial optimization problems arising in quantum information.

In the last few years, a large part of the algorithmic excitement in this area has been due to the development of a new rounding paradigm that is tailored to *average-case* problems. Crucial to this paradigm is viewing pseudo-expectations as formal relaxations of expectations (hence the terminology!) that can pass off as expectations for all polynomial inequalities that have an SoS proof.

Beginning in the work of Barak, Kelner and Steurer [20] on dictionary learning, this paradigm has been refined and used to give better algorithms for several average case problems including tensor decomposition [110], tensor completion [22, 122], outlier-robust moment estimation [99], clustering mixture models [77, 98], robust linear regression [94] and attacking cryptographic pseudorandom generators [17, 14]. The only prior usage of semidefinite programming in average-case setting was a celebrated line of work on the matrix completion problem [132, 39].

In this chapter, we will pick two vignettes to illustrate some of the main ideas in this area. We will present the famous MaxCut algorithm of Goemans and Williamson first as a introduction to the usage of the sum-of-squares method in worst-case algorithm design. We will then illustrate the applications to average-case problems by presenting a recently discovered [98] algorithm for clustering mixtures of isotropic Gaussians.

4.1 Max-Cut

MaxCut was among the first problems proved to be NP-hard [84], motivating the search for efficient approximation algorithms for this problem. Erdős [136] observed that a random bipartition of the graph cuts half of the edges in expectation. This immediately gives an $\frac{1}{2}$-approximation in expectation for MaxCut. In fact, until 1994, this algorithm was the best polynomial-time approximation for MaxCut. Subsequent work has shed light on the challenges of improving on this simple algorithm. Indeed, many standard approaches to designing approximation algorithms cannot yield better than an $\frac{1}{2}$-approximation in general for MaxCut. Charikar et al. [43] showed that no SA lift of degree $\Omega(n)$ can obtain better than an $(\frac{1}{2}+\varepsilon)$-approximation in the worst case. Kothari et al. [95], improving on the work of [40], showed that this result can be extended to any linear program which admits a linear projection to the MaxCut polytope.

In a breakthrough, Goemans and Williamson [60] showed that the approximation ratio for MaxCut could be improved to 0.878 using an SDP together with their *hyperplane rounding* technique. This was in fact the first approximation algorithm based on an SDP, catalyzing the study of SDPs as a tool for obtaining better approximation algorithms. In this section we will show that the Goemans and Williamson SDP is captured by the first level of the SoS

hierarchy.

MaxCut can be naturally phrased as the following quadratic program for a given graph G on n vertices and edge weights w:

$$\begin{aligned} \max \quad & \frac{1}{4}\sum_{i<j} w_{i,j}(x_i - x_j)^2, && (4.1) \\ \text{s.t.} \quad & x_i^2 = 1, && \forall i \in [n]. \end{aligned}$$

The constraints force x to be a ± 1-valued indicator of the partition of the vertices defining a cut in the graph. We will study how the 1st level of the SoS relaxation behaves on this quadratic program; for this it will be simpler to work with SoS over $\{\pm 1\}$ where we associate $x_i^2 = 1$ rather than $x_i^2 = x_i$, as described in Section 3.3.1. Maximizing the first level of the SoS relaxation of the quadratic program 4.1 gives a degree 2 pseudo-expectation $\tilde{\mathbb{E}}$ that maximizes $\tilde{\mathbb{E}}[\sum_{i<j} w_{i,j}(x_i - x_j)^2]$ subject to the constraints $x_i^2 = 1$ for every i.

Let (G, w) be an instance of MaxCut, and denote the value obtained from solving this SDP by $\mathsf{opt}_{\mathsf{SOS}_1}(\mathsf{MaxCut}(G,w)) = \tilde{\mathbb{E}}[\frac{1}{4}\sum_{i<j} w_{i,j}(x_i - x_j)^2]$. Similarly, define $\mathsf{opt}(\mathsf{MaxCut}(G,w))$ to be the true optimal value of the MaxCut instance. Note that because the SoS relaxation is obtained by relaxing the quadratic program it holds that $\mathsf{opt}_{\mathsf{SOS}_1}(\mathsf{MaxCut}(G,w)) \geqslant \mathsf{opt}(\mathsf{MaxCut}(G,w))$. We will show that this relaxation, along with a clever rounding procedure, achieves a 0.878-approximation in expectation.

Theorem 4.1 (Goemans and Williamson [60])**.** *Let (G, w) be an instance of* MaxCut*. There exists a randomized polynomial-time algorithm that finds a solution $\alpha^* \in \{0,1\}^n$ such that in expectation,*

$$\mathsf{opt}(\mathsf{MaxCut}(G,w)) \geqslant 0.878 \cdot \frac{1}{4}\sum_{i<j} w_{i,j}\left(\alpha_i^* - \alpha_j^*\right)^2.$$

The rounding algorithm, in retrospect, is quite natural and simple. To motivate it, it is helpful to imagine (falsely) that the pseudo-distributions corresponding to the solutions of the SoS relaxation are instead true probability distributions. In this view, we are given a probability distribution over cuts, that, in expectation achieves a large weighted cut value. If we could simply sample from this distribution, we would have found a cut that has value $\mathsf{opt}_{\mathsf{SOS}_1}(\mathsf{MaxCut}(G,w))$. There is an obvious issue with this plan – we only have access to degree 2 moments of the distribution. Thus, we can ask the following question: given degree 2 moments of a distribution supported on $\{-1,1\}^n$, can we sample from a probability distribution on $\{-1,1\}^n$ with those moments?

This turns out to be a hard question. Indeed, even verifying that a purported set of degree 2 moments correspond to a distribution on $\{-1,1\}^n$ is NP-hard. Nevertheless, it turns out that given the degree 2 moments of a pseudo-distribution, we can sample from a distribution on $\mathbb{R}^n$ with those moments.

Lemma 4.2 (Gaussian Sampling)**.** *Let $\mu \in \mathbb{R}^n$ and $\Sigma \in \mathbb{R}^{n\times n}$ be positive semidefinite. Then, there is a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with mean μ and covariance Σ. Further, given μ, Σ, we can sample from $\mathcal{N}(\mu, \Sigma)$ in polynomial time in n.*

Proof. Consider the random variable Y on $\mathbb{R}^n$ where each coordinate of Y is independently distributed as $\mathcal{N}(0,1)$ – the standard Gaussian distribution with mean 0 and variance 1. Then, $\mathbb{E}[YY^\top] = I$. Standard techniques allow efficiently sampling from the distribution of Y.

We can transform Y to have any given mean and covariance as follows. Let $\Sigma^{1/2}$ be the square root of Σ so that $\Sigma = \Sigma^{1/2}\Sigma^{1/2}$. Let $Z = \mu + \Sigma^{1/2}Y$. Then Z is a random variable with mean $\mathbb{E}[Z] = \mu$ and

$$\mathbb{E}[(Z-\mu)(Z-\mu)^\top] = \Sigma^{1/2}\mathbb{E}[YY^\top]\Sigma^{1/2} = \Sigma^{1/2}I\Sigma^{1/2} = \Sigma.$$

Further, we can sample from Z by simply drawing a sample from Y and applying the transformation $Z = \mu + \Sigma^{1/2}Y$. This completes the proof.

□

Given a degree two pseudo-expectation $\tilde{\mathbb{E}}$, we can use the Gaussian sampling lemma to sample $U \in \mathbb{R}^{n\times n}$ with mean $\tilde{\mathbb{E}}[x]$ and 2nd moment $\tilde{\mathbb{E}}[xx^\top]$. This, however, does not resolve our problem since we still need to find a cut and the coordinates of U are not Boolean in general. The key idea is to apply a simple "rounding" procedure that takes the vector u and outputs a Boolean vector that we explain and analyze in the lemma below. The analysis of our rounding will rely on the following elegant geometric fact.

Lemma 4.3 (Sheppard's Lemma)**.** *Let $g, h \sim \mathcal{N}(0,1)$ be jointly Gaussian with $\mathbb{E}[gh] = \rho$. Then,*

$$\mathbb{E}[(\mathsf{sign}(g) - \mathsf{sign}(h))^2] \geqslant \alpha_{GW}\mathbb{E}[(g-h)^2]\,,$$

where $\alpha_{GW} = \frac{4\arccos(\rho)}{\pi}$.

Proof. Let $\rho = \mathbb{E}[gh]$. Using Lemma 4.2, there are vectors $v_g, v_h \in \mathbb{R}^2$ of unit length so that for some $r = (r_1, r_2)$ with independent coordinates $r_1, r_2 \sim \mathcal{N}(0,1)$ such that g and h have the same distribution as $\langle r, v_g\rangle$ and $\langle r, v_h\rangle$ respectively.

Then, observe that for $r' = r/\|r\|_2$, $\mathsf{sign}(\langle r, v_g\rangle) = \mathsf{sign}(\langle r', v_g\rangle)$ and $\mathsf{sign}(\langle r, v_h\rangle) = \mathsf{sign}(\langle r', v_h\rangle)$. Further, $\mathsf{sign}(\langle r', v_g\rangle) \neq \mathsf{sign}(\langle r', v_h\rangle)$ if and only if v_g and v_h lie on opposite sides of r'. Now, by rotational symmetry of the standard Gaussian random variable, r' is uniformly distributed over the unit circle in $\mathbb{R}^2$. Thus, the probability that r' separates v_g and v_h is exactly $\frac{\theta}{\pi}$ where $\theta = \arccos(\rho)$ is the acute angle between the vectors v_g and v_h. Thus,

$$\mathbb{E}[(\mathsf{sign}(g) - \mathsf{sign}(h))^2] = 4\,\mathbb{P}[\mathsf{sign}(g) \neq \mathsf{sign}(h)] = 4\arccos(\rho)/\pi.$$

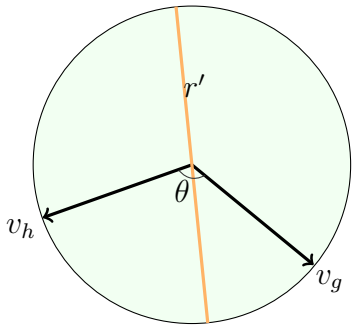


Figure 4.1: The random hyperplane r' separating v_g and v_h.

□

Lemma 4.4. *Let (G, w) be an instance of* MaxCut*. There is a randomized polynomial-time algorithm that, given a degree 2 pseudo-expectation $\tilde{\mathbb{E}}$ for the first-level SoS relaxation of (4.1), returns a solution $\alpha^* \in \{-1, 1\}^n$, such that*

$$\mathsf{opt}(\mathsf{MaxCut}(G, w)) \geqslant \mathbb{E}\left[\frac{1}{4}\sum_{i<j} w_{i,j}\left(\alpha_i^* - \alpha_j^*\right)^2\right] \geqslant 0.878 \cdot \mathsf{opt}_{\mathsf{SOS}_1}(\mathsf{MaxCut}(G, w)).$$

Proof. First, without loss of generality, we can assume that $\tilde{\mathbb{E}}$ satisfies $\tilde{\mathbb{E}}[x] = 0$. This is because if $\tilde{\mathbb{E}}$ doesn't satisfy this condition, then, we can simply define a new pseudo-expectation $\tilde{\mathbb{E}}'$ defined by $\tilde{\mathbb{E}}'[P(x)] = \frac{1}{2}(\tilde{\mathbb{E}}[P(x)] + \tilde{\mathbb{E}}[P(-x)])$. Since the objective function for our MaxCut program is invariant under negations, $\tilde{\mathbb{E}}'$ has the same value as $\tilde{\mathbb{E}}$. Thus, we will assume that $\tilde{\mathbb{E}}[x] = 0$ in the following.

Our rounding itself is extremely simple. We use the Gaussian sampling lemma to sample $g \sim \mathcal{N}(0, \tilde{\mathbb{E}}[xx^\top])$, where $\tilde{\mathbb{E}}[xx^\top]$ is the matrix $(\tilde{\mathbb{E}}[xx^\top])_{i,j} := \tilde{\mathbb{E}}[x_i x_j]$. We then output $\alpha^* \in \{\pm 1\}^n$ defined by $\alpha_i^* = \mathsf{sign}(g_i)$ for every i.

Fix any edge $\{i, j\}$ and let $\mathbb{E}[g_i g_j] = \tilde{\mathbb{E}}[x_i x_j] = \rho$. Then, $\mathbb{E}[(g_i - g_j)^2] = 2(1 - \rho)$. Using Lemma 4.3,

$$\mathbb{E}(\alpha_i^* - \alpha_j^*)^2 \geqslant \frac{\arccos(\rho)}{\pi}\mathbb{E}[(g_i - g_j)^2] = \frac{2\arccos(\rho)}{\pi(1-\rho)}\,. \tag{4.2}$$

Via elementary calculus, one can show that for every $\rho \in [-1, 1]$, $\frac{\arccos(\rho)}{2\pi(1-\rho)} \geqslant 0.878$. This completes the proof.

□

By Corollary 3.12 we can find a degree 2 pseudo-expectation $\tilde{\mathbb{E}}$ satisfying the first level SoS relaxation of (4.1) up to an additive error ε, which can be taken to be exponentially small in n. Combining this with Lemma 4.3 shows that SoS recovers Goemans and Williamson's result up to an additive error $o(1)$.

Naturally, one might ask whether Goemans and Williamson's algorithm can be improved upon. Indeed, the approximation ratio 0.878 seems somewhat arbitrary, and a-priori there is little reason to suspect that it is tight. In terms of hardness results, Bellare, Goldreich,

and Sudan [29] showed that it was NP-hard to approximate MaxCut to within a factor of 0.988. This was later improved by Håstad [73] to an approximation ratio of 0.941. This is far from the approximation achieved by the Goemans-Williamson algorithm, and yet despite significant effort, there has been no subsequent improvements on either side. Part of the reason for this could be explained by the fact that in order to improve the approximation ratio for MaxCut, one would need to refute the Unique Games Conjecture [90].

4.2 The Unique Games Conjecture and Sum-of-Squares

The advent of the PCP Theorem [8, 6] was a major breakthrough for hardness of approximation. It gave a new definition of NP-completeness that led to a multitude of results showing that for many NP-hard problems, computing an approximate solution is as difficult as solving the problem exactly. However, for several notorious NP-hard problems such as VertexCover and MaxCut, the PCP theorem was not enough to settle their approximability. In 2002, Khot [87] proposed his Unique Games Conjecture (UGC) which, if true, would settle the case for most of these outstanding optimization problems.

Informally, the Unique Games Conjecture states that finding a good approximate solution to a *Unique Game* is NP-hard.

Definition 4.5 (Unique Game)**.** A unique game consists of a graph $G = (V, E)$ as well as a set of permutations $\pi_{(u,v)} : [q] \to [q]$ for all $(u, v) \in E$ and $q \in \mathbb{N}$, defining a constraint $\pi_{(u,v)}(x_u) = x_v$. The aim is to assign a value to each vertex x_u so as to maximize the number of satisfied constraints.

One can think of a unique game as a restriction of the q-Coloring problem: given a graph, the aim is to assign one of q colors to each vertex so that no two adjacent vertices share the same color. A Unique Game is a restriction of the q-Coloring problem, enforcing that each assignment of a color to a vertex uniquely determines the colors of its neighbours. This restriction allows for Unique Games to be solved exactly in linear time: because connected components can be colored independently, we will assume without loss of generality that there is only a single component. Pick a vertex arbitrarily and assign to it a color. By the uniqueness property, this dictates the color of every other vertex in the graph. Repeating this procedure for all q possible colors will find a satisfying assignment if one exists.

On the other hand, the *Unique Games Conjecture* states that it is NP-hard to determine whether a Unique Game is *approximately* satisfiable.

Unique Games Conjecture. For every pair of constants $\varepsilon, \delta > 0$, there is a constant $q(\varepsilon, \delta)$ such that it is NP-hard to distinguish between the case when at least $1 - \varepsilon$ fraction of the constraints of a Unique Game are satisfiable, and the case when at most δ are satisfiable.

While on its own this conjecture does not look particularly interesting, it turns out that it is easy to encode many problems of interest as Unique Games. Indeed, this conjecture has

been shown to imply tight hardness of approximation results for problem such as MaxCut [89], VertexCover [92], and SparsestCut [44, 93], which have eluded hardness via the PCP theorem. Unlike conjectures such as $\mathsf{P} \neq \mathsf{NP}$, there is no consensus on the validity of the UGC. The state-of-the-art polynomial-time algorithms for Unique Games are given by Chlamtac, Makarychev, and Makarychev [45] and Charikar, Makarychev, and Makarychev [42] which, given an instance in which $(1-\varepsilon)$ constraints are satisfiable, find solutions satisfying a $1 - O(\varepsilon\sqrt{\log n \log q})$ and a $1 - O(\sqrt{\varepsilon \log q})$ fraction of the constraints respectively.

To better understand the UGC it is crucial to understand what the easy and potentially difficult instances of Unique Games are. Towards this goal, it has been particularly useful to work with an abstraction of the UGC known as the *Small Set Expansion Hypothesis* (SSEH). For a d-regular graph $G = (v, E)$, define the expansion of a set $S \subseteq V$ as $\mathsf{ex}(S) = \frac{|E(S, S\setminus V)|}{d|S|}$,[1] the fraction of edges crossing the boundary between S and $V \setminus S$.

Small Set Expansion Hypothesis. For every constant $\varepsilon > 0$ there exists $\delta > 0$ such that it is NP-hard to distinguish between the case when there exists a set $S \subseteq V$ with $|S| = \delta|V|$ where $\mathsf{ex}(S) \leqslant \varepsilon$, and the case when every set $S \subseteq V$ with $|S| := \delta|V|$ has $\mathsf{ex}(S) \geqslant 1 - \varepsilon$.

Raghavendra and Steurer [125] gave a highly non-trivial reduction showing that the SSEH implies the UGC. Furthermore, these problems have long been conjectured to be equivalent. Indeed, all known upper and lower bounds for Unique Games hold also for the Small Set Expansion problem. Therefore, this problem gives a conceptually simpler testing ground for studying the UGC, and has inspired many of the algorithmic results on the UGC. For example, using this connection, Arora, Barak, and Steurer [4] gave an $\exp(n^{o(1)})$-time algorithm for Unique Games, ruling out the existence of any exponentially-hard instance of Unique Games[2].

Regardless of the outcome, the fate of the UGC and SSEH appear to be intimately tied to Sum-of-Squares; much of the evidence for and against the UGC and SSEH is captured by the SoS hierarchy. Barak, Brandão and Harrow [15] showed that a constant level of the SoS hierarchy was sufficient to solve all of the candidate hard instance for the UGC. Furthermore, under the UGC, the approximation ratios of a large class of problems are tied to the first level of the SoS hierarchy. In a breakthrough result, Raghavendra [124] showed that the UGC implies hardness of approximation for every Constraint Satisfaction Problem (CSP), and furthermore that these approximation ratios are matched by the first level of the SoS hierarchy.

Definition 4.6 (Constraint Satisfaction Problem)**.** A CSP Δ is defined by a set of m predicates $P_1, \ldots, P_m : \Sigma^n \to \{0, 1\}$ over finite alphabet Σ. The objective is to find an assignment $\alpha \in \Sigma^n$ that satisfies the maximum number of constraints in Δ.

Recall from Section 3.2.3.3 that Raghavendra and Weitz [126] showed that CSP instances admit proofs with small coefficients, and therefore any CSP instance which has a degree d

[1] Here $E(S, V \setminus S)$ denotes the set of edges with one endpoint in S and the other in $V \setminus S$.

[2] This algorithm is also captured by the SoS hierarchy.

proof can be found in time $(m \cdot n)^{O(d)}$.

Theorem 4.7 (Raghavendra [124]). *Assuming the UGC, for every CSP Δ and every constant $\varepsilon > 0$, there is a polynomial time rounding procedure that, given as input the solution computed by $\mathsf{SOS}_1(\Delta)$, outputs a solution which is within an additive ε of the solution output by the optimal polynomial time algorithm for this problem.*

If true, the UGC would imply that for one of the most studied class of problems, there is a single algorithm achieving the best-possible approximation ratio of any polynomial time algorithm. Rather than constructing tailor-made algorithms for each CSP, we could simply appeal to this one-size-fits-all algorithm. An interesting technicality is that Raghavendra's result would not tell us what the optimal approximation ratio is for each CSP, only that SOS will match this ratio up to any small error factor. In fact, it is not even clear what the complexity of determining these approximation ratios is.

In a breakthrough result, Khot, Minzer, and Safra [91] positively resolved a weak form of the UGC known as the 2-*to*-2 *Conjecture* (with imperfect completeness). This was a major step towards positively resolving the UGC in a strong sense: it implies that the UGC is true when $\delta > 0$ and $\varepsilon > 1/2$. Even with this breakthrough, the truth of the full UGC remains a tantalizing open problem. A resolution, regardless of whether it is positive or negative, must lead to new advances in approximation algorithms and inapproximability. Additionally, due to its connection to SoS, a resolution to the UGC will likely have significant consequences for SoS. For more on the UGC we suggest the surveys by Khot [88] and Trevisan [146].

4.3 Average-Case Algorithm Design via SoS

In this section, we introduce a general approach that uses the SoS method to design algorithms for average-case *parameter-estimation* problems. These form a subclass of search problems: problems where we are interested in uncovering some hidden structure in the given data. The average-case nature of these problems shows up in modeling how the hidden structure is encoded in the observed input/data. Let us briefly make the notion of parameter estimation problems a bit more formal before proceeding.

Specifically, we imagine that the hidden structure is encoded by some vector valued hidden parameter $\Theta \in \mathbb{R}^p$. The observed input/data is modeled as some collection of points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ in some (typically high dimensional) Euclidean space. For each problem (such as clustering mixtures of Gaussians), we have an associated family of probability distributions $\{p_\Theta\}$ over $\mathbb{R}^d$ indexed by the parameter $\Theta \in \mathbb{R}^p$. Samples $x_1, x_2, \ldots, x_n$ are then *generated* by choosing n independent samples from p_Θ for some unknown Θ. We intend to design an algorithm that uses (as few as possible) samples $x_1, x_2, \ldots, x_n$ to come up with an estimate of Θ (accurate enough w.r.t. some relevant notion of distance). That is, the algorithm's goal is to *approximately invert* the above generating process.[3]

[3]Actually, the framework we present applies to slightly more general parameter-estimation problems where the data may be generated by some more complicated probabilistic process. This is not necessary to appreciate the ideas in this chapter so we omit a discussion on such "probabilistic generative models".

Such problems abound in theoretical machine learning with examples ranging from clustering mixture models, learning dictionaries to matrix completion and compressive sensing. Many interesting problems in average-case computational complexity theory and cryptography such as recovering planted cliques, recovering community structures in stochastic block models, and understanding the security properties of various notions of pseudorandom generators also conform to this general theme.

The SoS method offers a single prescription for designing efficient algorithms for such average-case parameter estimation problems. We will use the problem of clustering mixtures of isotropic Gaussians to illustrate it in this section.

4.3.1 Clustering Mixtures of Gaussians

We first describe the result that we will prove in detail in this section. Let's start by describing the algorithmic problem first. We will use the notation introduced here throughout this section. We are given n samples with an unknown partition into k equal size "clusters" $C_1, C_2, \ldots, C_k$ such that each C_r is an i.i.d. sample of size n/k from $\mathcal{N}(\mu_r, I_d)$ - Gaussian distribution with mean μ_r and covariance Identity in d dimensions. Here, the k means or centers μ_r are also unknown. We call the partition $C_1 \cup C_2 \cup \ldots \cup C_k$ the *ground-truth clustering* of the given sample and each C_r a *true cluster*.

Our goal is to recover a partition $\hat{C}_1 \cup \hat{C}_2 \cup \ldots \cup \hat{C}_k$ of the samples into k equal parts that is as close as possible to the true unknown partition. In order to understand approximation, we will say that the clustering above is δ-accurate if:

$$\min_{\pi:[k]\to[k]} \max_{r\leqslant k} \frac{k}{n} \cdot |C_r \cap \hat{C}_{\pi(r)}| \geqslant 1-\delta\,.$$

Given a δ-accurate clustering, standard methods can be used to recover good approximations to the unknown μ_rs. Observe that if for some $r \neq r'$, $\mu_r = \mu'_r$, then clearly one cannot distinguish between the corresponding clusters. In general, thus, we expect that the success of any algorithm will depend on the *separation parameter* $\Delta = \min_{r\neq r'} \|\mu_r - \mu'_r\|_2$, where $\|\cdot\|_2$ is the standard 2-norm.

We are now ready to state the main theorem we will prove.

Theorem 4.8. *For every integer $t \geqslant 2$, there is an algorithm, running in time $d^{O(t^2)}$, that takes input $X = C_1 \cup C_2 \cup \cdots C_k$ where C_r is an i.i.d. sample of size $n \gg \Theta(k)/\delta^2 \log(d) d^{2t}$ drawn according to $\mathcal{N}(\mu_r, I)$ for each r and with probability at least $1-1/d$, outputs a partition of $\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_k$ of $[n]$ into k equal pairwise disjoint subsets such that:*

$$\min_{\pi:[k]\to[k]} \max_{r\leqslant k} \frac{k}{n} \cdot |C_r \cap \hat{C}_{\pi(r)}| \geqslant 1-\delta$$

for $\delta = \Delta^{-2t}(512t)^t k^3$ and $\Delta = \min_{r\neq r'} \|\mu_r - \mu_{r'}\|_2$.

Remark (Quantitative Implications)**.** To understand the quantitative implication of the theorem, let us fix $\delta = 0.01$. Thus, we are interested in recovering a 99-percent accurate

clustering. Then, we want to choose t so that $(512\sqrt{t}/\Delta)^{2t} < 0.01/k^3$. Observe that this is only possible if $\Delta > C\sqrt{\log k}$ for some constant $C > 0$. For $\Delta = 10^4\sqrt{\log k}$ for e.g., we need to choose $t \sim \log k$ to make this happen. The theorem above thus gives a quasi-polynomial time (in k) algorithm.

As yet another parameter setting, fix some $\varepsilon > 0$, and suppose that $\Delta > 10^4 k^{\varepsilon}/\sqrt{\varepsilon}$. Then we need to choose $t = 3/\varepsilon$ to ensure that $(512\sqrt{t}/\Delta)^{2t} < 0.01/k^3$. In this case, the theorem yields a $\sim d^{O(1/\varepsilon^2)}$ time algorithm.

The above remark shows that the theorem gives no interesting guarantees when the separation parameter $\Delta \ll \sqrt{\log k}$. It turns out that this is inherent. Regev and Vijayraghavan [133] showed that when $\Delta \ll \sqrt{\log k}$, any algorithm requires an $d^{\Omega(k)}$ samples (and thus also running time) to solve the clustering problem. In the same work, they also showed an *inefficient* algorithm that succeeds in separating mixtures of k Gaussians using polynomially many samples in k and d whenever $\Delta \gg \sqrt{\log k}$. The theorem above gives a (quasi)-*efficient* algorithm to match their information theoretic result. The running time of the algorithm above at the "statistical threshold" of $\Delta \gg \sqrt{\log k}$ is quasi-polynomial. It is an open problem to give a polynomial time algorithm in this regime or prove an impossibility of such a result based on a standard average-case hardness assumption.

Remark (Generalizations)**.** The above theorem handles the problem of clustering an equi-weighted mixture of k isotropic Gaussians. The proof, however, easily extends in various ways: i) the clusters need not be equi-weighted, ii) the clusters need only have a covariance with bounded operator norm and iii) the components need only follow an arbitrary strongly log-concave distribution instead of being Gaussian. Further, the running time of the algorithm can be improved to $d^{O(t)}$ time, instead of $d^{O(t^2)}$. To keep our discussion simple, we will not discuss these corollaries or improvements in this chapter.

4.3.2 Algorithm Design Via SoS: a bird's eye view

A Proof Complexity Approach to Algorithm Design At a high-level, the algorithm-design approach we follow is rooted in proof complexity. Suppose we are interested in designing algorithms for a certain search problem. From a complexity theoretic perspective, we might want to do due diligence and first ask - given a purported solution, can we verify it? More specifically, can we verify some *certificate* of correctness of a good solution? Of course, in general, we do not expect $\mathsf{P} = \mathsf{NP}$ and thus a method to certify correctness of a good solution doesn't imply (or even help!) an efficient algorithm. The key idea in our approach is the following. If we restrict ourselves to finding certificates of correctness that can be verified in a restricted *automatizable* proof system (instead of the all-encompassing NP), then we get an efficient algorithm to find a solution along with a certificate whenever it exists! Our choice of the automatizable proof system will be low-degree SoS.

From an algorithm design perspective, this method is useful because it turns out that in several broad situations of interest, it is much easier to think of and analyze certifying algorithms than come up with search algorithms and their analyses.

Algorithms from Certifiability Let's now discuss the approach in the specific context of clustering Gaussian mixtures. As our discussion above suggests, the key idea is to consider first the task of *certifying* a good solution separately from that of designing an efficient algorithm. Eventually, the algorithm design step will become a problem-independent and almost mechanical translation from our certifiability method.

What does a certifying algorithm do? In the present situation, it gets as input the samples from unknown the mixture of Gaussian model. It also gets, in addition, a purported clustering of the samples. Our goal is to come up with a method to verify if this given clustering is close to the unknown true clustering.

Certifiability as Solution Testing We will accomplish this by developing a "certifiability test" that makes some simple checks on the given clustering. These checks must be designed so that the true clustering always passes them. This can be thought of as the "completeness" of the test. Equally importantly, these checks should be "sound": whenever some clustering passes the test, it must be close to the true clustering. In the context of average-case problem, our checks must succeed on all "typical" samples. That is, whenever any sample satisfies some fixed *deterministic* condition (somewhat unimaginatively, we will call such samples "good"), the test should be complete and sound for all possible purported clusterings. Further, the deterministic condition should hold for a (large enough) i.i.d. sample with high probability over the draw from the underlying distribution.

Identifiability from Certifiability Given a certifiability test, we immediately get an *inefficient* algorithm. Fix a good sample (i.e., a sample that satisfies the above-mentioned deterministic condition). We can then go over all possible (exponentially many) purported clusterings and run our certifiability test on each. Whenever we find one that passes our test, we are sure that we have an approximately correct clustering. This idea is referred to as information-theoretic unique *identifiability* in statistics and machine learning. Informally, identifiability means that a small finite sample (whp) uniquely determines the hidden parameter we intend to estimate from it. Identifiability is an information-theoretic idea and characterizes the *sample complexity* of the average-case parameter estimation problem but doesn't usually shed any light on its computational complexity. From what we discussed, certifiability immediately implies information-theoretic unique identifiability.

Simple Certifiability and Efficient Algorithms To obtain an efficient algorithm, we ask for a little more from our certifiability test. First, that the test be phrased as checking if for every (purported) cluster (seen as a subset of samples), its indicator vector satisfies a fixed set of polynomial inequalities. Similarly, the conclusion of the soundness property must be phrased as another polynomial inequality in the indicator vector of the subsample associated with the purported cluster. Second, the soundness property, when seen as a statement that a given system of polynomial inequalities (those checked for in the test) imply another polynomial inequality (the one in the conclusion) should have a *low-degree SoS proof.*

We can think of the above two conditions as some notion of *simplicity* of our certifiability test. The upshot is that whenever our certifiability test is simple in the above sense, we can obtain an efficient algorithm for recovering the clustering right from the certifiability statement in essentially black-box manner.

Natural certifiability tests are often simple The approach above would not be very useful if it is difficult to come up with simple certifiability test. The key reason behind the success of this approach is that fortunately, many natural proof strategies used in certifiability tests are actually already simple. This is because such arguments routinely rely on basic inequalities for e.g., triangle inequality, the Cauchy-Schwarz inequality and Hölder's inequality and their compositions. It turns out that all of these inequalities (and many more, including some deep geometric results such as isoperimetric inequalities) happen to have low-degree SoS proofs! Thus, whenever our soundness analysis of the certifiability test builds on a composition of such simple inequalities, we are automatically guaranteed a simple certifiability test in the sense above.

To summarize: algorithm design in this framework reduces to coming up with a simple certifiability test. That's exactly what we will do for designing our algorithm for separating Gaussian mixtures.

Certifying Good Clusterings

In this section, we will develop the key step of certifying clusterings in designing our algorithm for clustering Gaussian mixtures. To allow geometric intuition and minimal notation, we will first develop the test in 1D and then generalize to higher dimensions.

4.3.2.1 Certifying Good Clusterings: The 1D Case

Given a set of points $X = x_1, x_2, \ldots, x_n \in \mathbb{R}$ that is produced by a disjoint union of $C_1, C_2, \ldots C_k$ where each C_r is an i.i.d. sample of size n/k distributed as $\mathcal{N}(\mu_i, 1)$, our goal is to build a test, that takes a subset of points, with indices in $S \subseteq [n]$, and checks if it is δ-close (in Hamming distance) to some true cluster C_r. In particular, since each C_r consists of i.i.d. samples from a Gaussian with an arbitrary mean, our test must check some property that is true of (large enough) i.i.d. samples of Gaussians but not of any mixture of at least two Gaussians.

A visual inspection reveals a class of tests that seem to work, at least in a qualitative sense. A true cluster has the points in it huddled closely around the empirical mean. While a subset of points that intersects substantially with more than one cluster will have points that are more spread out around the empirical mean. This qualitative reasoning suggests that our test should be some measure of how spread out a given subset of points is, around its empirical mean. We will use a fairly natural one – the empirical centralized moments of the given subset of points.

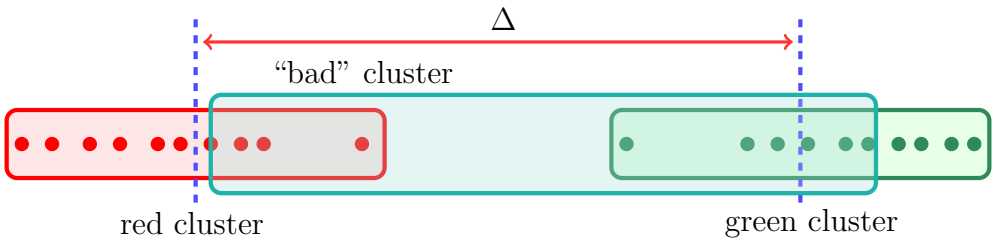


Figure 4.2: A mixture of two well-separated Gaussians in 1D with red/green points. And a blue "fake" cluster.

Definition 4.9 (Empirical Moments)**.** For a set of points $Y = y_1, y_2, \ldots, y_n \in \mathbb{R}$, we define the t-th centralized moment of Y as:

$$m_t(Y) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \mu(Y))^{2t}$$

where $\mu(Y) = 1/n \sum_{i=1}^{n} y_i$, the empirical mean of Y.

Empirical moments give a measure of the deviation from the empirical mean of the set of points Y. We now want to understand how these empirical moments behave for a true cluster in order to figure out a quantitative version of a test based on the t-th moments that can distinguish true clusters from fake ones.

Let us first define a good sample X as one that has its empirical t-th moments close to the distributions t-th moment.

Definition 4.10 (Good 1D Sample)**.** An n-sample X partitioned into clusters $C_1 \cup C_2 \cup \ldots C_k$ of equal sizes is (t, Δ)-good if $\hat{\mu}_r := \frac{k}{n}\sum_{j \in C_r} x_j$ satisfy $|\hat{\mu}_r - \hat{\mu}_{r'}| \geqslant \Delta$ for every $r \neq r'$ and $\frac{k}{n}\sum_{j \in C_r}(x_j - \hat{\mu}_r)^{2t} \leqslant (16t)^t$.

Proposition 4.11 (Convergence of Moments)**.** *Let $X = C_1 \cup C_2 \cup \ldots C_k$ where each C_r is a i.i.d. sample of size n/k from $\mathcal{N}(\mu_r, 1)$ satisfying $|\mu_r - \mu_{r'}| \geqslant \Delta$ whenever $r \neq r'$. Then, X is a $(t, \Delta - \delta)$-good sample whenever $n \geqslant \frac{k}{\delta^2}\Theta(\log(k/\eta))$ with probability at least $1 - \eta$.*

We will use the following simple inequality in the proof:

Proposition 4.12 (Almost Triangle Inequality)**.** *For any a, b, $(a + b)^{2t} \leqslant 2^{2t-1}(a^{2t} + b^{2t})$.*

Proof. Apply Jensen's inequality to obtain $(\frac{a+b}{2})^{2t} \leqslant \frac{1}{2}(a^{2t} + b^{2t})$ and rearrange. □

Proof of Proposition 4.11. We will show that each of the two properties hold with probability $1 - \eta/2$ whenever $n \geqslant \frac{k}{\delta^2}\Theta(\log(k/\eta))$. The proof then follows by a union bound.

Let $\hat{\mu}_r = \frac{k}{n}\sum_{j \in C_r} x_j$ for every r. Then, $\hat{\mu}_r$ is a Gaussian random variable with mean μ_r and variance $\frac{k}{n}$. Thus, $\mathbb{P}[|\mu_r - \hat{\mu}_r| > \delta] \leqslant \exp(-\Theta(n\delta^2/k))$. Choosing $n > \Theta(\frac{1}{\delta^2} k \log(k/\eta))$ and doing a union bound over $1 \leqslant r \leqslant k$, we thus have $|\mu_r - \hat{\mu}_r| \geqslant \Delta - \delta$ with probability at least $1 - \eta/2$.

Next, for any r, by Proposition 4.12,

$$\frac{k}{n}\sum_{j\in C_r}(x_j-\hat{\mu}_r)^{2t} = 2^{2t-1}\frac{k}{n}\sum_{j\in C_r}(x_j-\mu_r)^{2t} + 2^{2t-1}\frac{k}{n}\sum_{j\in C_r}(\hat{\mu}_r-\mu_r)^{2t}).$$

Further,

$$(\hat{\mu}_r-\mu_r)^{2t} = \left(\frac{k}{n}\sum_{j\in C_r}(x_j-\mu_r)\right)^{2t} \leqslant \left(\frac{k}{n}\sum_{j\in C_r}(x_j-\mu_r)^{2t}\right).$$

Thus, $\frac{k}{n}\sum_{j\in C_r}(x_j-\hat{\mu}_r)^{2t} \leqslant 2^{2t}\frac{k}{n}\sum_{j\in C_r}(x_j-\mu_r)^{2t}$.

Let $z_{r,j} = (x_j-\mu_r)^{2t}$. Then,

$$\mathbb{E}[z_{r,j}^{2t}] = \mathbb{E}_{g\sim\mathcal{N}(0,1)}[g^{2t}] = (2t-1)!! \leqslant (2t)^t.$$

Let $z_r = \frac{k}{n}\sum_{j\in C_r} z_{r,j}$. Thus, by Hoeffding's inequality, with probability at least $1-\eta/k$, $\mathbb{P}[z_r > q] \leqslant 2^{-Cn/kq^2/(4t)^{2t}}$ for some large enough constant C. Choosing $q = 2(2t)^t$ and $n = \Theta(\log(k/\eta))$ this ensures that $\mathbb{P}[z_r > 2(2t)^t] < \eta/2k$. By a union bound over $1 \leqslant r \leqslant k$, with probability at least $1-\eta/2$, $\frac{k}{n}\sum_{j\in C_r}(x_j-\mu_r)^{2t} \leqslant 2(2t)^t$.

Thus, with probability at least $1-\eta/2$ over the draw of X, for each $1 \leqslant r \leqslant k$, $\frac{k}{n}\sum_{j\in C_r}(x_j-\hat{\mu}_r)^{2t})] \leqslant 2^{2t+1}(2t)^t \leqslant (16t)^t$.

□

We can now design a simple test for true clusters based on checking t-th moments.

Algorithm 4.13 (1D Test for True Clusters)**.**

Given: $X = \{x_1, x_2, \ldots, x_n\} = C_1 \cup C_2 \cup \ldots \cup C_k \subseteq \mathbb{R}$ and a subset $S \subseteq [n]$

Output: *Yes* iff $|S \cap C_r| > \delta|C_r|$ for some $1 \leqslant r \leqslant k$.

Operation:

1. Let $\mu(S) = \frac{k}{n}\sum_{j\in S} x_j$.
2. Output yes iff (**i**) $|S| = n/k$, and (**ii**) $\frac{k}{n}\sum_{j\in S}(x_j-\mu(S))^{2t} \leqslant (16t)^t$.

It is important to observe that the test itself is *deterministic*. The only randomness is over the draw of the sample. The completeness of the test is immediate from the definition of a good sample.

Fact 4.14 (Completeness)**.** *For any (t,Δ)-good sample X, every C_r passes the test.*

The *soundness* of the test is more interesting. Such a result must show that *any* S that passes this test must be *close* to being a true cluster. Said differently, passing the test provides a *certificate* that the given set S is close to being a true cluster.

Lemma 4.15 (Certifying a good solution in 1D)**.** *Let $X = C_1 \cup C_2 \cup \cdots C_k$ be (t, Δ)-good sample. Suppose $w \in \mathbb{R}^n$ satisfies:*

1. ***Set Indicator**: for each $i \leqslant n$, $w_i^2 = w_i$. That is, $w_i \in \{0, 1\}$ and thus indicates a subsample of X,*
2. ***Set Size**: $\sum_{i=1}^n w_i = n/k$. That is, w indicates a subsample of size n/k, and,*
3. ***Bounded Centralized Moment**: for $\mu(w) = \frac{k}{n}\sum_i w_i x_i$, $\frac{k}{n}\sum_{i=1}^n w_i(x_i - \mu(w))^t \leqslant (16t)^t$. That is, the empirical centralized moment of the set indicated by w is as small as that of a true cluster.*

Then, for $w(C_r) = \frac{k}{n}\sum_{i\in C_r} w_i$ and $\delta = k\Delta^{-2t}256^t t^t$,

$$\sum_{r=1}^k w^2(C_r) \geqslant 1 - \delta\,.$$

In particular,

$$\max_r w(C_r) \geqslant \sum_{r=1}^k w^2(C_r) \geqslant 1 - \delta\,.$$

That is, the set indicated by w must intersect one of the true clusters in $1-\delta$ fraction of the points.

Proof. We start from the basic fact that $\sum_r w(C_r) = 1$. Squaring this we obtain that $\sum_{r,r'} w(C_r)w(C_{r'}) = 1$. Thus, it is enough to show that $\sum_{r\neq r'} w(C_r)w(C_{r'}) \leqslant \delta$ to complete the proof.

In the following, we use $\mu(w) = \frac{k}{n}\sum_i x_i w_i$. We write $\mu_{C(i)}$ to denote the "true mean of the true cluster that contains x_i". That is, $\mu_{C(i)} = \mu_r$ for r such that $i \in C_r$.

Since $|\mu_r - \mu_{r'}| > \Delta$ for every $r \neq r'$, we must have:

$$\sum_{r\neq r'} w(C_r)w(C_{r'}) \leqslant \frac{1}{\Delta^t}\sum_{r\neq r'\leqslant k} w(C_r)w(C_{r'})(\mu_r - \mu_{r'})^{2t}\,.$$

By the almost triangle inequality (Proposition 4.12),

$$\begin{aligned}\sum_{r\neq r'} w(C_r)w(C_{r'}) &\leqslant \frac{2^{2t-1}}{\Delta^{2t}}\sum_{r\neq r'\leqslant k} w(C_r)w(C_{r'})\left((\mu_r - \mu(w))^{2t} + (\mu(w) - \mu_{r'})^{2t}\right)\\ &= \frac{2^{2t}}{\Delta^t}\sum_{r\leqslant k} w(C_r)(\mu_r - \mu(w))^{2t}\,.\end{aligned}$$

Substituting $w(C_r) = \sum_{i\in C_r} w_i$ and applying the almost triangle inequality again, we have:

$$\sum_{r\neq r'} w(C_r)w(C_{r'}) \leqslant \frac{2^{2t}}{\Delta^t}\frac{k}{n}\sum_{i=1}^n w_i(\mu_{C(i)} - \mu(w))^{2t}$$

$$\leqslant \frac{2^{2t}}{\Delta^t} \cdot k\frac{2^{2t-1}}{n} \sum_{i=1}^{n} w_i \left((x_i - \mu_{C(i)})^{2t} + (x_i - \mu(w))^{2t}\right) .$$

Using $w_i \leqslant 1$, we can conclude:

$$\sum_{r \neq r'} w(C_r)w(C_{r'}) \leqslant k\frac{2^{4t-1}}{\Delta^{2t}} \left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{C(i)})^{2t} + \frac{1}{n}\sum_{i=1}^{n} w_i(x_i - \mu(w))^{2t}\right) .$$

Since X is a good sample, $\frac{1}{n}\sum_{i=1}^{n}\left((x_i - \mu_{C(i)})^{2t}\right) \leqslant (16t)^t$. And since w satisfies our test, $\frac{1}{n}\sum_{i=1}^{n} w_i\left((x_i - \mu(w))^{2t}\right) \leqslant (16t)^t$. Thus,

$$\sum_{r \neq r'} w(C_r)w(C_{r'}) \leqslant k\frac{2^{4t}}{\Delta^{2t}}(16t)^t .$$

□

This immediately gives an *inefficient* algorithm for the 1D case: for any good sample, simply enumerate over all possible clusterings and apply the 1D test until a good clustering is found. In the remainder we show how this test, together with the degree-automatizability of SoS gives rise to an efficient algorithm. This algorithm is not significantly simpler in the 1D case than the higher dimensional case, and therefore we first describe how to extend the 1D test to higher dimensions before developing this algorithm.

4.3.2.2 Certifying Good Clusterings: The Higher Dimensional Case

We can generalize the test from the previous section to higher dimensions with little to change the overall idea. As such, the main dimension-dependent part of the test is the test of low deviation via "small-centralized-moments". We will replace this check by a small-centralized-moment test for the 1D samples obtained by projecting the given sample into all possible directions.

In notation, if y is distributed as $\mathcal{N}(\mu, I)$, then, for any $v \in \mathbb{R}^d$, $\langle y - \mu, v\rangle$ is a 1 dimensional Gaussian random variable with mean 0 and variance $\|v\|_2^2$. Thus, disregarding the computational feasibility, we can simple apply the "small-centralized-moment" test to $\langle y - \mu, v\rangle$ for *every* unit vector $v \in \mathbb{R}^d$. This gives us the following test:

Algorithm 4.16 (Higher Dimensional Test for True Clusters)**.**

Given: $X = \{x_1, x_2, \ldots, x_n\} = C_1 \cup C_2 \cup \ldots \cup C_k \subseteq \mathbb{R}^d$ and a subset $S \subseteq [n]$

Output: Yes iff $|S \cap C_r| > \delta|C_r|$ for some $1 \leqslant r \leqslant k$.

Operation:

1. Let $\mu(S) = \frac{k}{n}\sum_{i \in S} x_i$.

2. Output yes iff (**i**) $|S| = n/k$, and (**ii**) $\frac{k}{n}\sum_{i\in S}\langle x_i - \mu(S), v\rangle^{2t} \leqslant (16t)^t$ for every unit vector v.

Definition 4.17 (Good Sample)**.** A k-partitioned $X = C_1 \cup C_2 \cup \ldots \cup C_k \subseteq \mathbb{R}^d$ into clusters of equal sizes is a (t, Δ)-good sample if for $\hat{\mu}_r = \frac{k}{n}\sum_{j\in C_r} x_j$ and every $v \in \mathbb{R}^d$, we have $\|\hat{\mu}_r - \hat{\mu}_{r'}\|_2 \geqslant \Delta$ whenever $r \neq r'$ and for every $r \leqslant k$,

$$\frac{k}{n}\sum_{j\in C_r}\langle x_j - \hat{\mu}_r, v\rangle^{2t} \leqslant (16t)^t \|v\|^{2t}\,.$$

As before, we by using Hoeffding's inequality, we can prove that a large enough i.i.d. sample is (t, Δ)-good with high probability.

Proposition 4.18 (Convergence of Directional Moments)**.** *Let $X = C_1 \cup C_2 \cup \ldots C_k$ where each C_r is a i.i.d. sample of size n/k from $\mathcal{N}(\mu_r, I_d)$ satisfying $\|\mu_r - \mu_{r'}\|_2 \geqslant \Delta$ whenever $r \neq r'$. Then, X is a $(t, \Delta - \delta)$-good sample whenever $n \geqslant d^{2t}\frac{k}{\delta^2}\Theta(\log(k/\eta))$ with probability at least $1 - \eta$.*

This immediately implies the completeness as in the 1D case.

Proposition 4.19 (Completeness)**.** *For any (t, Δ)-good sample X, every C_r passes the test.*

The soundness proof just needs one additional step on top of the ideas in the 1D case. Let $\mathcal{A}_w$ be the following system of quadratic constraints that captures the checks on the indicator of the given purported cluster S made in the test above.

$$\mathcal{A}_w\colon \left\{ \begin{array}{ll} & \sum_{i=1}^n w_i = \frac{n}{k} \\ \forall i \in [n]\,, & w_i^2 = w_i \\ \forall v \in \mathbb{R}^d \text{ s.t. } \|v\|_2 = 1\,, & \frac{k}{n}\sum_{i\leqslant n} w_i\langle x_i - \hat{\mu}, v\rangle^{2t} \leqslant (16t)^t \\ & \frac{k}{n}\sum_{i\leqslant n} w_i x_i = \hat{\mu} \end{array} \right\} \tag{4.3}$$

Lemma 4.20 (Certifying a good solution in high dimensions)**.** *Let $X = C_1 \cup C_2 \cup \ldots \cup C_k$ be a (t, Δ)-good sample. Suppose $w \in \mathbb{R}^n$ satisfies $\mathcal{A}_w$. Then, for $w(C_r) = \frac{k}{n}\sum_{i\in C_r} w_i$ and $\delta = \Delta^{-2t}k^3(16t)^t$,*

$$\sum_{r=1}^k w^2(C_r) \geqslant 1 - \delta\,.$$

In particular,

$$\max_r w(C_r) \geqslant \sum_{r=1}^k w^2(C_r) \geqslant 1 - \delta\,.$$

Proof. As in the proof of Lemma 4.15, we will focus on proving $\sum_{r\neq r'} w(C_r)w(C_{r'}) \leqslant \delta$. We will write $\mu_{C(i)}$ to denote the "true mean" of the cluster that contains x_i. That is, $\mu_{C(i)} = \mu_r$ for r such that $i \in C_r$.

We can repeat the argument in the proof of Lemma 4.15 to obtain for any $v \in \mathbb{R}^d$,

$$\frac{1}{\Delta^{2t}}\sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t} \leqslant \|v\|^{2t} k \frac{2^{2t}}{\Delta^{2t}}\left(\frac{1}{n}\sum_{i=1}^{n}\langle x_i - \mu_{C(i)}, v\rangle^{2t} + \frac{1}{n}\sum_{i=1}^{n} w_i \langle x_i - \mu(w), v\rangle^{2t}\right).$$

Now, since X is (t,Δ)-good, $\frac{1}{n}\sum_{i=1}^{n}\langle x_i - \mu_{C(i)}, v\rangle^{2t} \leqslant (16t)^t\|v\|^{2t}$. And since w satisfies the bounded moment constraint, $\frac{1}{n}\sum_{i=1}^{n} w_i\langle x_i - \mu(w), v\rangle^{2t} \leqslant (16t)^t\|v\|^{2t}$. Thus,

$$\frac{1}{\Delta^{2t}}\sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t} \leqslant \|v\|^{2t} k\Delta^{-2t}(256t)^t \tag{4.4}$$

To finish, we observe that for any r, r', $\sum_{i,j\leqslant k}\langle \mu_r - \mu_{r'}, \frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|}\rangle^{2t} \geqslant \Delta^{2t}$. This is in fact the contribution due to just $i = r$, and $j = r'$. Thus, letting V be the collection of $\binom{k}{2}$ unit vectors $\frac{\mu_i - \mu_j}{\|\mu_i - \mu_j\|}$ for $i \neq j$, we have:

$$\sum_{r\neq r'} w(C_r)w(C_{r'}) \leqslant \sum_{v\in V}\frac{1}{\Delta^{2t}}\sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t}.$$

Combining this with (4.4) completes the proof.

□

4.3.3 Inefficient Algorithm from Certifiability

In Section 4.3.2, we discussed that given a complete and sound certifiability test, simply going over all possible clusterings and applying our test gives an inefficient algorithm to find an approximate clustering of the input sample. In this section, we will give a different and seemingly more complicated inefficient algorithm to accomplish the same task. The upshot is that we will be able to make this into an *efficient* algorithm with one change and essentially the same *analysis*.

Rounding Maximum Entropy Distributions In this different version of the inefficient algorithm, we imagine that we are given *low-degree moments* of a probability distribution D over points $w \in \mathbb{R}^n$ that pass the checks done in Test 4.16. Since each point in the support of D passes these checks, by Lemma 4.20, D is supported on ws that indicate a subset of samples that are close to some (not necessarily the same) true cluster. Can we recover the true clusters given such a distribution D?

To begin with, D may not have information about all clusters since it may just be supported on a proper subset of them. We can force D to be informative about all clusters

by forcing it to use each sample at most $1/k$ fraction of the times. That is, if you randomly draw a $w \sim D$, then $w_i = 1$ with probability $1/k$. Equivalently, we force $\mathbb{E}_D[w_i] = 1/k$ for every i. Clearly such a distribution exists (namely, the uniform distribution on true clusters) so this is not a vacuous requirement.

Now, if (by some stroke of luck) we acquired the ability to generate samples distributed according to D, we can just take $k \log k$ samples $w \sim D$ and obtain a full (approximate) clustering of the sample. However, we don't expect to be able to sample from a distribution D given only its low-degree moments. Thus, the above problem, however artificial it might appear at this point, is at least non-trivial.

We now make a simple observation that allows us to use only low-degree moments of D and obtain a good w.

Lemma 4.21. *Let X be a (t, Δ)-good sample and let D be a distribution supported on w satisfying $\mathcal{A}_w$. Further, suppose $\mathbb{E}_{w\sim D}[w_i] = 1/k$ for each $i \in [n]$. Then $M = \mathbb{E}_{w\sim D}[ww^\top]$ satisfies:*

1. *for each i, j, $0 \leqslant M_{i,j} \leqslant \frac{1}{k}$,*
2. *For any $r \in [k]$ and for any $i \in C_r$, $\mathbb{E}_{j\sim[n]}[M_{i,j}] = \frac{1}{k^2}$, and,*
3. *For every $r \in [k]$, $\mathbb{E}_{i\sim C_r, j\notin C_r}[M_{i,j}] \leqslant k\delta$.*

Proof. Let's derive two simple consequences from the hypothesis that w satisfies the constraints generated by the checks in Test 4.16. Every such w satisfies $w_i^2 = w_i$ for each i. Thus, every $w_i \geqslant 0$ and so for every pair i, j, $w_i w_j \geqslant 0$. Taking expectations w.r.t. D yields that M is a non-negative matrix.

Let us now upper bound any entry of the matrix M. Using the Cauchy-Schwarz inequality and that $w_i^2 = w_i$ for every i, we have that for any i, j,

$$\mathbb{E}[w_i w_j] \leqslant \sqrt{\mathbb{E}[w_i^2]}\sqrt{\mathbb{E}[w_j^2]} = \sqrt{\mathbb{E}[w_i]}\sqrt{\mathbb{E}[w_j]} = 1/k.$$

Finally, for any $i \in C_r$, $(\sum_{j\in[n]} w_i w_j) = w_i \frac{n}{k}$. Taking expectations with respect to D yields that $\sum_j M_{i,j} = \mathbb{E}_{w\sim D}[w_i] = \frac{1}{k^2}$. Next, every $w \in \mathbb{R}^n$ satisfies $\frac{k}{n}\sum_{i=1}^n w_i = 1$. Squaring yields: $\frac{k^2}{n^2}\sum_{i,j} w_i w_j = 1$, and taking expectations with respect to D implies that $\mathbb{E}_{i,j\sim[n]}[M_{i,j}] = \frac{1}{k^2}$. Finally, since every w in the support of D satisfies the checks in Test 4.16, it must satisfy the conclusion of the Lemma 4.20. Thus, for every w in the support of D, we must have:

$$\sum_{r=1}^{k} (w(C_r))^2 \geqslant 1 - \delta$$

where $\delta = \Delta^{-2t} 2^{10t} k^2 (2t)^t$ and $w(C_r) = \frac{k}{n}\sum_i w_i$.

Taking expectations with respect to D, we obtain:

$$\mathbb{E}_{r\sim[k]}\mathbb{E}_{i,j\sim C_r}[M_{i,j}] \geqslant \frac{1}{k^2}(1 - \delta)$$

Equivalently,

$$\mathbb{E}_{r\in[k]}\left[\mathbb{E}_{i\in C_r, j\notin C_r}[M_{i,j}]\right] \leqslant \delta\,.$$

Thus, for every $r \in [k]$, $\mathbb{E}_{i\in C_r, j\notin C_r}[M_{i,j}] \leqslant k\delta$.

□

Why is this observation helpful? Observe that $M = \mathbb{E}_{w\sim D}[ww^\top]$ is the (scaled) 2nd (and thus, low-degree) moment matrix of the distribution D.

From the above lemma, the entries of M are non-negative, and average up to $1/k^2$. Further, the average of entries of M that correspond to pairs i, j in different clusters is at most $\frac{1}{k^2}(1-\delta)$. Thus, most of the "mass" of M comes from pairs that belong to the same cluster so it appears that we have some non-trivial information about the true clusters if we just look at the 2nd moment matrix M. We can then hope to recover the clusters approximately by looking at the large entries in M.

Let's formalize this next and obtain a "rounding" algorithm that takes distributions D as above and outputs an approximate clustering of X.

Algorithm 4.22 (Rounding Algorithm for Distributions)**.**

Given: A (t, Δ)-good sample X with true clusters $C_1, C_2, \ldots, C_k$. A distribution D satisfying $\mathcal{A}_w$ and $\mathbb{E}_D w_i = \frac{1}{k}$ for every i.

Output: A partition of X into an approximately correct clustering $\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_k$.

Operation: For $M = \mathbb{E}_{w\sim D}[ww^\top]$, repeat for $1 \leqslant r \leqslant k$:

1. Choose a uniformly random row i of M.
2. Let $\hat{C}_r$ be the largest $\frac{n}{k}$ entries in the i-th row of M.
3. Remove the rows and columns with indices in $\hat{C}_r$.

Lemma 4.23. *Let X, D satisfy the hypothesis of Lemma 4.21. Fix any $\eta > 0$. Choose $t = O(\log(k) + \log(1/\eta))$ so that $\delta < \frac{\eta^2}{k^6}$. Then, with probability at least 0.99, Algorithm finds a collection of k subsets $C'_1, C'_2, \ldots, C'_k$ so that for each $r \in [k]$, $|C_r \cap C'_r| \geqslant (1-\eta)|C_r|$.*

Proof. Fix any cluster C_r. Call an entry of M "large" if it exceeds $\frac{\eta}{k^2}$. We will analyze where the large entries in a typical row $i \in C_r$ of M come from. Using part (1) and (2) of Lemma 4.21, we obtain that for any $\eta > 0$, the fraction of entries in the i-th row that exceed η/k^2 is at least $(1-\eta)/k$. Thus, each row of M has at least $(1-\eta)/k$ fraction of its entries large.

On the other hand, using part (3) of Lemma 4.21 along with Markov's inequality, we obtain that with probability at least $1 - 1/k^2$ over the choice of a uniformly random choice of $i \in C_r$, $\mathbb{E}_{j\not\sim i}[M_{i,j}] < k^3\delta$. Call all $i \in C_r$ for which this holds "good" rows. Since δ is chosen so that $k^3\delta < \eta^2/k^3$, for each good row, again by Markov's inequality, the fraction of $j \not\sim i$ such that $M_{i,j} > \eta/k^2$ is at most η/k. Thus, for any good row in C_r, if we take the

indices j corresponding to the largest $\frac{n}{k}$ entries (i,j) in M, then, at most η fraction of such j are not from C_r. Thus, if we pick a uniformly random row in C_r and take the largest $\frac{n}{k}$ entries in that row, then, we obtain a subset of $\frac{n}{k}$ points that intersects with C_r in $(1-\eta)$ fraction of the points.

It's now easy to finish the analysis of the rounding algorithm. Our rounding algorithm chooses a random row of M, collects $\frac{n}{k}$ largest entries in that row and returns this set as a cluster. It then repeats the process on remaining rows. By the above argument, each iteration succeeds with probability $1-1/k^2$. Thus, by a union bound, all iterations succeed with probability at least $1-1/k$.

□

4.3.4 Efficient Algorithms via SoS-ized Certifiability

We are now finally ready to obtain an efficient algorithm for finding approximate clustering of Gaussian mixture data. In the previous section, we relied only on the existence of a certifiability test in order to give an inefficient algorithm. Here, we will use that the soundness of the certifiability test has a low-degree SoS proof in order to get an efficient version of the algorithm presented in the previous section.

While this part is relatively technical, it is important to note that it is not problem-specific and thus applicable more generally to average-case algorithm design. Indeed, most of the ideas appearing in this section are employed in most (if not all) known results that use the SoS method for average-case algorithm design.

Pseudo-distributions satisfying $\mathcal{A}_w$ In order to start out, let's observe the key inefficient piece in the previous section's algorithm is that it is hard to find a distribution (even its low-degree moments) supported on w satisfying $\mathcal{A}_w$. We will relax this step and instead find a low-degree *pseudo-distribution* satisfying $\mathcal{A}_w$ instead. In Section 3.2.2.2, we showed that unlike distributions, finding low-degree pseudo-distributions approximately satisfying a system of polynomially many constraints can be done in polynomial time.

Unfortunately, $\mathcal{A}_w$ actually has an infinitely large number of bounded moment constraints, of the form one for each $v \in \mathbb{R}^d$! Here, we will rely on an important and generally applicable idea that allows replacing such infinitely many constraints by polynomially many by relying on the existence of low-degree SoS proofs for the constraint system itself.

Certifiable Subgaussianity In the present context of bounded moment inequalities, this idea requires that the underlying distribution (Gaussian for us) satisfy a certain concentration property called as *certifiable subgaussianity* .

Definition 4.24 (Certifiably SubGaussian Distributions)**.** A distribution D on $\mathbb{R}^d$ is said to be k-certifiably C-subGaussian, if there is a degree k SoS proof of the following unconstrained polynomial inequality in variables $v_1, v_2, \ldots, v_d$: $\mathbb{E}_D\langle x, v\rangle^k \leqslant (Ck)^{k/2}(\mathbb{E}_D\langle x, v\rangle^2)^{k/2}$. In notation, we write: $\left|\frac{v}{k}\right. \left\{\mathbb{E}_D\langle x, v\rangle^k \leqslant (Ck)^{k/2}(\mathbb{E}_D\langle x, v\rangle^2)^{k/2}\right\}$.

Certifiable subgaussianity asserts that the empirical k-th moments of D are bounded a la subGaussian distributions and that this bound has a SoS proof of degree k. Observe that the definition is invariant under linear transformations – that is, if an $\mathbb{R}^d$-valued random variable X has a k-certifiably C-subGaussian distribution, then so does any linear transformation AX of it for any matrix $A \in \mathbb{R}^{d\times d}$.

Conveniently for us, a large enough sample of D inherits certifiable subgaussianity of D.

Lemma 4.25 (Certifiable Subgaussianity Under Sampling)**.** *Let D be k-certifiably C-subGaussian. Let Y be an i.i.d. sample from D of size $n > \Theta(d^k)$. Then, the uniform distribution on Y is k-certifiably $2C$-subGaussian.*

Multivariate Gaussian distributions are k-certifiably 1-subGaussian for all k.

Lemma 4.26 (Certifiable Subgaussianity of Gaussians)**.** *For every k and any positive semidefinite $\Sigma \in \mathbb{R}^{d\times d}$,*

$$\left|\frac{v}{k}\right. \left\{\mathbb{E}_{\mathcal{N}(0,\Sigma)}\langle x, v\rangle^k \leqslant k^{k/2}(\mathbb{E}_{\mathcal{N}(0,\Sigma)}\langle x, v\rangle^2)^{k/2}\right\} .$$

Proof. First observe that $\mathbb{E}_{\mathcal{N}(0,\Sigma)}\langle x, v\rangle^k = \alpha_k(\mathbb{E}_{\mathcal{N}(0,\Sigma)}\langle x, v\rangle^2)^{k/2}$ for $\alpha_k = k!! < k^{k/2}$ for all even k (and 0 otherwise). The proof then follows by noting that all polynomial identities of degree k have degree k SoS (or even Sherali-Adams) proofs. □

More generally, certifiable subgaussianity is satisfied by a much broader family of distributions including all discrete product distributions with subGaussian marginals and all distributions that satisfy a Poincaré inequality (see [98]). This latter class, in particular, includes all strongly log-concave distributions. We will omit this discussion in this monograph.

Succint Representation of $\mathcal{A}_w$ via Certifiable Subgaussianity Certifiable subgaussianity allows us to compress the infinitely many bounded moment constraints in (4.3). The key idea is that whenever a family of constraints are expressible as a single polynomial inequality that has a SoS proof, then we can write down an equivalent system of polynomially many (with the degree of the polynomial depending on the degree of the SoS proof) constraints at the cost of introducing polynomially many additional variables.

While we will largely stick to our current application, this technique is an important idea in algorithm design based on SoS. It can be seen as a principled way to convert "for all" quantified statements into "there exist" quantified statements and thus can be seen as a limited version of *quantifier elimination* within SoS.

We will start by describing a special case that is easily understandable before discussing the version needed for our algorithm.

Lemma 4.27 (Quantifier Elimination for Quadratic Forms)**.** *Let*

$$\mathcal{F} = \{A \in \mathbb{R}^{d\times d} | A = A^\top \text{ and } \forall v \in \mathbb{R}^d,\ v^\top A v \leqslant \|v\|_2^2\},$$

and

$$\mathcal{B} = \{A | A = A^\top \text{ and } \exists Q \in \mathbb{R}^{d\times d},\ I - A = QQ^\top\}.$$

Then, $\mathcal{B} = \mathcal{F}$.

Before giving the simple proof of this fact, let's clarify what it states. First, observe that $\mathcal{F}$ is a system of constraints with a "for all" quantifier while $\mathcal{B}$, a single constraint with a "there exists" quantifier. Next, observe that $\mathcal{B}$, as a constraint system, has an extra d^2 variables (each entry of the matrix Q). And finally, observe that $I - A = QQ^\top$ is a equality of matrices and opens up into a system of d^2 polynomial equalities, one for each entry of the matrix. Each such inequality is a quadratic equation in the variables Q. Thus, the lemma immediately gives a way to replace an infinitely many constraints in $\mathcal{F}$ by d^2 constraints in $\mathcal{B}$ by introducing d^2 extra variables.

Proof. The proof uses some elementary linear algebra. First, observe that $v^\top A v^\top + v^\top (I - A)v = \|v\|^2$. If $v^\top A v \leqslant \|v\|^2$ for all v, then, $v^\top (I - A)v \geqslant 0$ for all v and thus, $I - A$ is positive semidefinite. Thus, we can write the Cholesky decomposition $I - A = QQ^\top$ for some $d \times d$ matrix A. Rearranging yields:

$$\|v\|^2 - v^\top A v = \|Qv\|^2 = \sum_{i \leqslant d} \langle q_i, v \rangle^2 , \tag{4.5}$$

where q_i are the rows of Q.

By the above reasoning, we have: $\mathcal{F} = \{A \mid \exists Q \in \mathbb{R}^{d \times d} \; \forall v \in \mathbb{R}^d \; \|v\|^2 - v^\top A v = \|Qv\|^2\}$. A priori, the RHS above appears to have made no progress - we have introduced an extra d^2 variables Q and seem to still retain infinitely many constraints – one for each $v \in \mathbb{R}^d$. Observe however that the RHS of (4.5) states that the quadratics polynomials $\|v\|^2 - v^\top A v$ and $\|Qv\|^2$ are equal for all v. This is possible iff the two polynomials have the same coefficients. The equality of coefficients then allows us to *eliminate* v and obtain an equivalent system of $\binom{d}{2} + d$ different equality constraints.

In the present case, we can do this explicitly and obtain the matrix equality $I - A = QQ^\top$ that translates into $\binom{d}{2} + d$ equality (for each entry of the matrices up to symmetry). This yields $\mathcal{B}$ and completes the proof. □

To generalize the above idea, observe that (4.9) is an SoS proof of the non-negativity of the polynomial $\|v\|^2 - v^\top A v$. Whenever there is an SoS proof, we can simply convert this inequality constraint into a equality of two matrices which then translates into a system of equality constraints, one for each entry of the corresponding matrices. Thus, an SoS proof allows us to eliminate the vectors v and obtain a succinct representation of constraint system of the form $\forall v \in \mathbb{R}^d$, $p(v) \geqslant 0$.

In the present situation, our polynomial $p(v) = (16t)^t \|v\|^{2t} - \frac{k}{n} \sum_i w_i \langle x_i - \hat{\mu}, v \rangle^{2t}$. When w indicates a true cluster C_r, the set indicated by it is an i.i.d. sample from a Gaussian distribution. From Lemmas 4.26 and 4.25, uniform distribution on C_r is k-certifiably 2-subGaussian. Thus, when w indicates a true cluster, we know that there's a SoS proof of non-negativity of $p(v)$. Such a proof is of the form: $p(v) = \sum_{i=1}^{d^2} \langle q_i, v^{\otimes t} \rangle^2$. We can now repeat the argument above to introduce variables for q_i and eliminate v to obtain polynomially many constraints that capture the moment bounds.

We now make the above discussion concrete. First, let's introduce the following new constraint system.

$$\mathcal{B}_{w,Q}\colon \left\{ \begin{array}{rr} & \sum_{i=1}^n w_i = \frac{n}{k} \\ \forall i \in [n]\,, & w_i^2 = w_i \\ \frac{k}{n}\sum_{i\leqslant n} w_i \langle x_i - \hat{\mu}, v\rangle^{2t} + \|Qv^{\otimes t}\|^2 = (16t)^t & \\ & \frac{k}{n}\sum_{i\leqslant n} w_i x_i = \hat{\mu} \end{array} \right\} \tag{4.6}$$

Observe that even though we have infinitely many constraints in the above, we can eliminate v and replace them by $\leqslant d^{2t}$ equality constraints. We do not do this explicitly for the sake of readability. Thus, $\mathcal{B}_{w,Q}$ is a system of constraints in $d^{2t}+n$ variables and $d^{2t}+\mathsf{poly}(n)$ constraints as we desired.

Feasibility of $\mathcal{B}_{w,Q}$ Before proceeding, we must ensure that the true cluster indicators are solutions to $\mathcal{B}_{w,Q}$. For $\mathcal{A}_w$, we established this when we proved the completess of Test 4.16. Here, we will prove a similar result for $\mathcal{B}_{w,Q}$.

First, we will make our notion of (t,Δ)-good samples a bit more strict.

Definition 4.28 (Certifiably Good Sample)**.** A k-partitioned $X = C_1 \cup C_2 \cup \ldots \cup C_k \subseteq \mathbb{R}^d$ into clusters of equal sizes is a (t,Δ)-*certifiably* good sample if for $\hat{\mu}_r = \frac{k}{n}\sum_{j\in C_r} x_j$, we have $\|\hat{\mu}_r - \hat{\mu}_{r'}\|_2 \geqslant \Delta$ whenever $r \neq r'$, and for every $r \leqslant k$,

$$\left|\tfrac{v}{k}\right\} \left\{ \frac{k}{n}\sum_{j\in C_r} \langle x_j - \hat{\mu}_r, v\rangle^{2t} \leqslant (16t)^t \|v\|^{2t} \right\}.$$

Using Lemma 4.25 and 4.26, we immediately have:

Lemma 4.29 (Certifiable Convergence of Moments)**.** *Let $X = C_1 \cup C_2 \cup \ldots C_k$ where each C_r is a i.i.d. sample of size n/k from $\mathcal{N}(\mu_r, I_d)$ satisfying $\|\mu_r - \mu_{r'}\|_2 \geqslant \Delta$ whenever $r \neq r'$. Then, X is a $(t, \Delta-\delta)$-certifiably good sample whenever $n \geqslant d^{2t}\frac{k}{\delta^2}\Theta(\log(k/\eta))$ with probability at least $1-\eta$.*

The above immediately implies that $\mathcal{B}_{w,Q}$ is feasible – in that, true clusters satisfy the constraints in $\mathcal{B}_{w,Q}$. This can be thought of as SoS version of our completeness statement from before.

Sum-of-Squares Proof of Certifiability Finally, in order to construct an analog of the analysis of the inefficient algorithm 4.22, we need a SoS version of the soundness statement. Towards this, we will first phrase the certifiability as a statement about a polynomial inequality in indeterminates $w_1, w_2, \ldots, w_n$ that holds whenever the w_is satisfy the constraint system $\mathcal{A}_w$ of polynomial inequalities appearing in the checks of our certifiability test.

Lemma 4.30 (Sum-of-Squares Certifiability)**.** *Let X be a (t,Δ)-certifiably good sample. Then, for $w(C_r) = \frac{k}{n}\sum_{i\in C_r} w_i$ and $\delta = \Delta^{-2t}2^{10t}k^3(256t)^t$,*

$$\mathcal{B}_{w,Q} \left|\frac{v,w}{2t}\right. \left\{ \sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t} \leqslant k^3 \frac{(512t)^t}{\Delta^{2t}}\|v\|^{2t} \right\} . \tag{4.7}$$

for every integer t.

To prove this "SoS-version" of the soundness of certifiability test, we will show that each line of the proof of Lemma 4.20 is a polynomial inequality with a low-degree SoS proof. By using simple facts about composition of SoS proofs, we will thus have a proof of Lemma 4.30. Recall that all the inequalities in our proof of Lemma 4.20 used the "almost triangle inequality". The proof of the SoS-version of Lemma 4.20 will rely on the following "SoS-ization" of this inequality.

Lemma 4.31 (Almost Triangle Inequality)**.** *For any $k \in \mathbb{N}$,*

$$\left|\frac{a,b}{2k}\right. (a+b)^{2k} \leqslant 2^{k-1}(a^{2k}+b^{2k}) .$$

Our proof of this will rely on SoS-ization of the AM-GM inequality. The first SoS proof for the AM-GM inequality was given by Hurwitz [78] in 1891. There are elementary proofs known now (see for example, Appendix A of [20]).

Fact 4.32 (SoS AM-GM Inequality)**.** *For polynomials $w_1, w_2, \ldots, w_n$,*

$$\left|\frac{w_1,w_2,\ldots,w_n}{n}\right. \left\{ \left(\frac{w_1+w_2+\cdots+w_n}{n}\right)^n \geqslant \prod_{i\leqslant n} w_i \right\} .$$

Proof of Lemma 4.31. Using the identity, $2(a^2+b^2)-(a+b)^2 = (a-b)^2$, we have:

$$\left|\frac{a,b}{2}\right. \left\{2(a^2+b^2)-(a+b)^2 \geqslant 0\right\} .$$

Iteratively using this argument k times gives:

$$\left|\frac{a,b}{2k}\right. \left\{2^k(a^2+b^2)^k-(a+b)^{2k} \geqslant 0\right\} . \tag{4.8}$$

Now, by binomial expansion, $(a^2+b^2)^k = \sum_{i=0}^k \binom{k}{i} a^{2i}b^{2k-2i}$. By Lemma 4.32 applies to $a^{2i}b^{2k-2i}$, we have that for every $0 \leqslant i \leqslant k$,

$$\left|\frac{a,b}{2k}\right. \left\{ \left(\frac{2i}{2k}a + \frac{2k-2i}{2k}b\right)^{2k} \geqslant a^{2i}b^{2k-2i} \right\} .$$

Thus,

$$\left|\frac{a,b}{2k}\right. \left\{ \sum_{0\leqslant i\leqslant k} a^{2i}b^{2k-2i} \leqslant \sum_{0\leqslant i\leqslant k} \binom{k}{i} \left(\frac{2i}{2k}a^{2k} + \frac{2k-2i}{2k}b^{2k} \right) \right\},$$

or,

$$\left|\frac{a,b}{2k}\right. \left\{ \sum_{0\leqslant i\leqslant k} a^{2i}b^{2k-2i} \leqslant \sum_{0\leqslant i\leqslant k} \binom{k}{i} \left(\frac{1}{2}a^{2k} + \frac{1}{2}b^{2k} \right) \leqslant 2^{k-1}(a^{2k}+b^{2k}) \right\}.$$

Combining with (4.8) completes the proof.

□

We can now prove Lemma 4.30.

Proof of Lemma 4.30. Let $w(C_r)$ stand for the linear polynomial $\sum_{i\in C_r} w_i$ in w for every $r \leqslant k$. We will use the SoS-ized almost triangle inequality to show:

$$\left|\frac{v,w}{2t}\right. \left\{ \sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t} \leqslant k\frac{2^{4t}}{\Delta^{2t}} \left(\frac{1}{n}\sum_{i=1}^n \langle x_i - \mu_{C(i)}, v\rangle^{2t} + \frac{1}{n}\sum_{i=1}^n w_i \langle x_i - \mu(w), v\rangle^{2t} \right) \right\}. \tag{4.9}$$

Let us first complete the proof assuming (4.9). First, we can have a SoS proof upper bounding the two terms on the RHS above. Observe that, $\frac{1}{n}\sum_{i=1}^n \langle x_i - \mu_{C(i)}, v\rangle^{2t} = \frac{1}{k}\sum_{j\leqslant k}\frac{k}{n}\sum_{i\in C_j}\langle x_i - \mu_j, v\rangle^{2t}\|v\|^{2t}$. Since X is (t,Δ)-certifiably good, we must have:

$$\left|\frac{v}{2t}\right. \left\{ \frac{1}{n}\sum_{i=1}^n \langle x_i - \mu_{C(i)}, v\rangle^{2t} \leqslant (16t)^t \|v\|^{2t} \right\}. \tag{4.10}$$

Next, we also have:

$$\mathcal{B}_{w,Q}\left|\frac{w,v}{2t}\right. \left\{ \frac{k}{n}\sum_{i=1}^n w_i \langle x_i - \mu(w), v\rangle^{2t} \leqslant (16t)^t \|v\|^{2t} \right\}. \tag{4.11}$$

Combining the above two bounds with (4.9), we have the conclusion:

$$\mathcal{B}_{w,Q}\left|\frac{v,w}{2t}\right. \left\{ \sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t} \leqslant k^3\frac{(512t)^t}{\Delta^{2t}}\|v\|^{2t} \right\}. \tag{4.12}$$

Let us now complete the proof of (4.9). By the SoS version of the almost triangle inequality (Lemma 4.31), we have:

$$\left|\frac{w,v}{2t}\right. \left\{ \frac{1}{\Delta^{2t}}\sum_{r\neq r'} w(C_r)w(C_{r'})\langle \mu_r - \mu_{r'}, v\rangle^{2t} \leqslant \frac{2^{2t-1}}{\Delta^{2t}} \sum_{r\neq r'\leqslant k} w(C_r)w(C_{r'}) \right.$$
$$\left. \left(\langle \mu_r - \mu(w), v\rangle^{2t} + \langle \mu(w) - \mu_{r'}, v\rangle^{2t} \right) \right\}$$

$$\left|\frac{w,v}{2t}\right. \left\{ \frac{1}{\Delta^{2t}} \sum_{r \neq r'} w(C_r) w(C_{r'}) \langle \mu_r - \mu_{r'}, v \rangle^{2t} \leqslant \frac{2^{2t}}{\Delta^t} \sum_{r \leqslant k} w(C_r) (\mu_r - \mu(w))^{2t} \right\} .$$

Substituting $w(C_r) = \sum_{i \in C_r} w_i$ and applying the almost triangle inequality again, we have:

$$\left|\frac{w,v}{2t}\right. \left\{ \frac{1}{\Delta^{2t}} \sum_{r \neq r'} w(C_r) w(C_{r'}) \langle \mu_r - \mu_{r'}, v \rangle^{2t} \leqslant \frac{2^{2t}}{\Delta^t} \frac{k}{n} \sum_{i=1}^{n} w_i \langle \mu_{C(i)} - \mu(w), v \rangle^{2t} \right\}$$

$$\left|\frac{w,v}{2t}\right. \left\{ \frac{1}{\Delta^{2t}} \sum_{r \neq r'} w(C_r) w(C_{r'}) \langle \mu_r - \mu_{r'}, v \rangle^{2t} \leqslant \frac{2^{2t}}{\Delta^t} \cdot k \frac{2^{2t-1}}{n} \sum_{i=1}^{n} w_i \left(\langle x_i - \mu_{C(i)}, v \rangle^{2t} + \langle x_i - \mu(w), v \rangle^{2t} \right) \right\} .$$

Using the identity $w_i + (w_i - w_i^2) + (w_i - 1)^2 = 1$, we have:

$$\mathcal{B}_{w,Q} \left|\frac{w}{2}\right. \{ w_i \leqslant 1 \} .$$

We can thus conclude that,

$$\mathcal{B}_{w,Q} \left|\frac{w,v}{2t}\right. \left\{ \frac{1}{\Delta^{2t}} \sum_{r \neq r'} w(C_r) w(C_{r'}) \langle \mu_r - \mu_{r'}, v \rangle^{2t} \right.$$
$$\left. \leqslant k \frac{2^{2t}}{\Delta^{2t}} \left(\frac{1}{n} \sum_{i=1}^{n} \langle x_i - \mu_{C(i)}, v \rangle^{2t} + \frac{1}{n} \sum_{i=1}^{n} w_i \langle x_i - \mu(w), v \rangle^{2t} \right) \right\} .$$

This completes the proof.

□

Efficient Algorithm and Analysis We can now describe an efficient analog of Algorithm 4.22.

Algorithm 4.33 (Efficient Clustering Algorithm)**.**

Given: A (t, Δ)-good sample X with true clusters $C_1, C_2, \ldots, C_k$.

Output: A partition of X into an approximately correct clustering $\hat{C}_1, \hat{C}_2, \ldots, \hat{C}_k$.

Operation:

1. Find a degree $2t$ pseudo-distribution D satisfying $\mathcal{B}_w$ and satisfying $\tilde{\mathbb{E}}\, w_i = \frac{1}{k}$ for every $1 \leqslant i \leqslant n$.
2. For $M = \tilde{\mathbb{E}}_{w \sim D}[w w^\top]$, repeat for $1 \leqslant r \leqslant k$:
   (a) Choose a uniformly random row i of M.
   (b) Let $\hat{C}_r$ be the largest $\frac{n}{k}$ entries in the i-th row of M.

(c) Remove the rows and columns with indices in $\hat{C}_r$.

The analysis of our inefficient algorithm 4.22 relied on the simple properties of the second moment matrix of D presented in Lemma 4.21. To analyze our efficient algorithm above, we will give the analog of Lemma 4.21 and prove the *same* properties for the 2nd moment matrix of the pseudo-distribution D. This will finally finish the full analysis of our algorithm.

Lemma 4.34. *Let X be a (t, Δ)-certifiably good sample and let D be a pseudo-distribution of degree at least $2t$ on w, Q satisfying $\mathcal{B}_{w,Q}$. Further, suppose $\tilde{\mathbb{E}}_D[w_i] = 1/k$ for each $i \in [n]$. Then $M = \mathbb{E}_{w\sim D}[ww^\top]$ satisfies:*

1. *for each i, j, $0 \leqslant M_{i,j} \leqslant \frac{1}{k}$,*
2. *For any $r \in [k]$ and for any $i \in C_r$, $\mathbb{E}_{j\sim[n]}[M_{i,j}] = \frac{1}{k^2}$, and,*
3. *For every $r \in [k]$, $\mathbb{E}_{i\sim C_r, j\notin C_r}[M_{i,j}] \leqslant k\delta$.*

In our analysis, we will use the following Cauchy-Schwarz inequality for pseudo-distributions.

Proposition 4.35 (Cauchy-Schwarz Inequality)**.** *Let f, g be polynomials of degree at most k in indeterminate x. Then, for every pseudo-distribution D on x of degree at least $4k$, we have:*

$$\tilde{\mathbb{E}}_D[fg] \leqslant \sqrt{\tilde{\mathbb{E}}_D[f^2]}\sqrt{\tilde{\mathbb{E}}_D[g^2]}\,.$$

Proof. First suppose $\tilde{\mathbb{E}}[g^2] = 0$. For any $c > 0$, we have $\tilde{\mathbb{E}}[(cf - g)^2] \geqslant 0$. Expanding out and using linearity and rearranging, this gives $2c\tilde{\mathbb{E}}[fg] \leqslant c^2\tilde{\mathbb{E}}[f^2]$ or $\tilde{\mathbb{E}}[fg] \leqslant c/2\,\tilde{\mathbb{E}}[f^2]$. Taking limits as $c \to 0$ yields that $\tilde{\mathbb{E}}[fg] \leqslant 0$. A similar argument starting with $\tilde{\mathbb{E}}[(cf + g)^2] \geqslant 0$ yields that $\tilde{\mathbb{E}}[fg] \geqslant 0$. Together, we obtain that $\tilde{\mathbb{E}}[fg] = 0$. This completes the proof in this case.

Now suppose $\tilde{\mathbb{E}}[f^2], \tilde{\mathbb{E}}[g^2] > 0$. Let $f' = f/\sqrt{\tilde{\mathbb{E}}[f^2]}$ and $g' = g/\sqrt{\tilde{\mathbb{E}}[g^2]}$ and $\tilde{\mathbb{E}}[f^2] > 0$. Then, $\tilde{\mathbb{E}}[(f' - g')^2] \geqslant 0$ implies that $\tilde{\mathbb{E}}[f'g'] \leqslant \frac{1}{2}(\tilde{\mathbb{E}}[f'^2] + \tilde{\mathbb{E}}[g'^2]) = 1$. Plugging back, we obtain $\tilde{\mathbb{E}}[fg] \leqslant \sqrt{\tilde{\mathbb{E}}[f^2]}\sqrt{\tilde{\mathbb{E}}[g^2]}$ as required.

□

Proof. We have:

$$\mathcal{B}_{w,Q} \left|\tfrac{w}{4}\right. w_i w_j = w_i^2 w_j^2\,.$$

Since D is a pseudo-distribution of degree at least 4, we must have: $\tilde{\mathbb{E}}[w_i w_j] = \tilde{\mathbb{E}}[w_i^2 w_j^2] \geqslant 0$. This proves that M is a non-negative matrix.

Let us now upper bound any entry of the matrix M. Using Cauchy-Schwarz inequality for pseudo-distributions and that $w_i^2 = w_i$ for every i, we have that for any i, j,

$$\tilde{\mathbb{E}}_D[w_i w_j] \leqslant \sqrt{\tilde{\mathbb{E}}_D[w_i^2]}\sqrt{\tilde{\mathbb{E}}_D[w_j^2]} = \sqrt{\tilde{\mathbb{E}}[w_i]}\sqrt{\tilde{\mathbb{E}}[w_j]} = 1/k\,.$$

Finally, for any $i \in C_r$, we have

$$\mathcal{B}_{w,Q} \left| \frac{w}{2} \right. \left\{ \sum_{j \in [n]} w_i w_j = w_i \frac{n}{k} \right\} .$$

Taking pseudo-expectations yields that $\sum_j M_{i,j} = \tilde{\mathbb{E}}_D[w_i] = \frac{1}{k}$. Next,

$$\mathcal{B}_{w,Q} \left| \frac{w}{2} \right. \frac{k}{n} \sum_{i=1}^{n} w_i = 1 .$$

Squaring yields:

$$\mathcal{B}_{w,Q} \left| \frac{w}{2} \right. \frac{k^2}{n^2} \sum_{i,j} w_i w_j = 1 .$$

Taking pseudo-expectations w.r.t D implies that $\frac{1}{n^2} \sum_{i,j \leqslant n} [M_{i,j}] = \frac{1}{k^2}$.

Finally, using Lemma 4.30,

$$\mathcal{B}_{w,Q} \left| \frac{v,w}{2t} \right. \left\{ \sum_{r \neq r'} w(C_r) w(C_{r'}) \langle \mu_r - \mu_{r'}, v \rangle^{2t} \leqslant k^3 \frac{(512t)^t}{\Delta^{2t}} \|v\|^{2t} \right\} . \tag{4.13}$$

Taking pseudo-expectations with respect to D, we have:

$$\sum_{r=1}^{k} \tilde{\mathbb{E}}_D \, w^2(C_r) \geqslant 1 - \delta$$

where $\delta = \Delta^{-2t} k^3 (512t)^t$ and $w(C_r) = \frac{k}{n} \sum_i w_i$. Taking expectations with respect to D, we obtain:

$$\mathbb{E}_{r \sim [k]} \mathbb{E}_{i,j \sim C_r} [M_{i,j}] \geqslant \frac{1}{k^2} (1 - \delta)$$

Equivalently,

$$\mathbb{E}_{r \in [k]} \left[\mathbb{E}_{i \in C_r, j \notin C_r} [M_{i,j}] \right] \leqslant \delta .$$

Thus, for every $r \in [k]$, $\mathbb{E}_{i \in C_r, j \notin C_r} [M_{i,j}] \leqslant k\delta$.

□

Chapter 5

Lower Bounds for Sum-of-Squares

In this Chapter, our main focus will be to present the linear lower bound on the degree of SoS refutations of random 3XOR equations (Section 5.1). Then in Section 5.2, we give a brief survey of other SoS lower bounds, and applications.

5.1 3XOR

In this Section, we present the result by Grigoriev [65] and independently by Schoenbeck [139], giving linear lower bounds on the SoS degree of refuting random 3XOR equations.

Definition 5.1 (Random 3XOR)**.** A random 3XOR instance ϕ over $x_1, \ldots, x_n$ is defined as follows. Let $m = O(n)$. Choose m random mod 2 equations $x_i + x_j + x_k = b_{ijk}$, where i, j, k are chosen randomly without replacement from $[n]$, and $b_{ijk} \in \{0, 1\}$ is also chosen randomly.

The complexity of solving a system of linear equations exactly is in P. In terms of approximate solutions, a random solution is expected to satisfy at least half of the equations. Håstad proved that this is essentially optimal, unless P equals NP. In contrast, we will prove that SoS requires maximal (linear) degree even to determine whether a random 3XOR instance is satisfiable or not.

Theorem 5.2. *Let ϕ be a random* 3XOR *instance over $x_1, \ldots, x_n$ with $m = cn$ equations for $c > 0$ sufficiently large. Then with high probability, ϕ is unsatisfiable and requires SoS refutations of degree $\Omega(n)$.*

It will simplify the argument considerably to change basis from $\{0, 1\}$ to $\{-1, 1\}$. Thus, for the rest of this section, we will convert ϕ into an equivalent set of multilinear monomial equations over $\{-1, 1\}$ as follows. Given ϕ as defined above, apply the linear transformation $1 - 2x_i$, $1 - 2b_{ijk}$ to each equation to obtain multilinear monomial equations of the form:

$$x_i x_j x_k = b_{ijk} \qquad b_{ijk} \in \{\pm 1\}.$$

First, we show that a random 3XOR instance is unsatisfiable with high probability.

Lemma 5.3. *Let ϕ be an instance of* 3XOR *on n variables with $m = c_\varepsilon n$ constraints (for some constant c_ε depending only on ε) be chosen as follows: for each constraint we choose $i, j, k \sim [n], b_{ijk} \sim \{\pm 1\}$ i.i.d. Then with probability at least $1 - 2^{-n}$, every assignment $\alpha \in \{\pm 1\}^n$ satisfies at most $(\frac{1}{2} + \varepsilon)m$ constraints, where the probability is over the choice of ϕ.*

Proof. For a fixed $\alpha \in \{\pm 1\}^n$ we let Y_j^α be the event that the j-th constraint is satisfied by $x = \alpha$ and $Y^\alpha = \sum_j Y_j^\alpha$ be the number of constraints satisfied by $x = \alpha$. By construction of the constraints, for a fixed α each Y_j^α is an independent Bernoulli random variable with expectation $\frac{1}{2}$. Therefore a standard Chernoff bound implies

$$\mathop{\mathbb{P}}_{\phi}\left[Y^\alpha > \left(\frac{1}{2} + \varepsilon\right)m\right] < 2^{O(-\varepsilon^2 m)} \quad \forall \alpha \in \{\pm 1\}^n,$$

and so by a union bound on all x

$$\mathop{\mathbb{P}}_{\phi}\left[\exists \alpha \in \{\pm 1\}^n \mid Y^\alpha > \left(\frac{1}{2} + \varepsilon\right)m\right] < 2^{n - O(\varepsilon^2 m)}.$$

Choosing $m = c_\varepsilon n$ for an appropriate $c_\varepsilon = O(\frac{1}{\varepsilon^2})$ makes this probability less than 2^{-n}. □

The lower bound for SoS actually gives a reduction from SoS degree lower bounds to *Gaussian width* lower bounds, which in turn are equivalent to Resolution width lower bounds.

Definition 5.4 (Gaussian Refutation)**.** Let $x_S = \prod_{i \in S} x_i$, and Let $\mathcal{L} = \{l_1 = b_1, \ldots, l_m = b_m\}$ be a set of monomial equations over $x_1, \ldots, x_n$, where each l_i is a multilinear degree 3 monomial and $b_i \in \{\pm 1\}$. A Gaussian refutation of $\mathcal{L}$ is a sequence of multilinear monomial equations such that:

1. Each equation is either from $\mathcal{L}$, or follows from two previous equations by: $x_S = b_1, x_{S'} = b_2 \to x_{S \triangle S'} = b_1 \cdot b_2$.
2. The final equation is $1 = -1$.

The *width* of a Gaussian refutation is the maximum degree of any equation, and the *Gaussian width* of $\mathcal{L}$ is the minimum width over all Gaussian refutations of $\mathcal{L}$.

Example 5.5. Suppose that $\mathcal{L} = \{x_1 = -1, x_1 x_2 = 1, x_2 x_3 = -1, x_3 = -1\}$. Then $\mathcal{L}$ has a degree 2 Gaussian refutation:

1. $x_1 = -1$ (Initial equation)
2. $x_1 x_2 = 1$ (Initial equation)
3. $x_2 = -1$ (1 and 2)
4. $x_2 x_3 = -1$ (Initial equation)
5. $x_3 = 1$ (3 and 4)

6. $x_3 = -1$ (Initial equation)

7. $1 = -1$ (5 and 6)

The following canonical procedure determines all monomial equations, $\mathcal{L}_{\leqslant d}$ that can be derived from $\mathcal{L}$ via width d Gaussian reasoning. Initially $\mathcal{D}_{\leqslant d}$ is the set of all equations in $\mathcal{L}$. Repeat the following until no new equations can be generated: Take two equations $x_S = b_1$, $x_{S'} = b_2$ from $\mathcal{D}_{\leqslant d}$ and derive $x_{S \triangle S'} = b_1 b_2$. If this equation is not already in $\mathcal{D}_{\leqslant d}$ and has degree at most d, then add it to $\mathcal{D}_{\leqslant d}$. This procedure will converge with $\mathcal{D}_{\leqslant d}$ equal to $\mathcal{L}_{\leqslant d}$. Thus, $\mathcal{L}$ has a width-d Gaussian refutation if and only if $-1 = 1$ is in $\mathcal{L}_{\leqslant d}$.

It is not hard to see that all of the monomial equations in $\mathcal{L}_{\leqslant d}$ can be derived in degree d SoS. Remarkably we will show that, for 3XOR, linear combinations of such equations is essentially all that can be derived in degree d SoS. The following two lemmas imply Theorem 5.2.

Lemma 5.6. *Let ϕ be an instance of* 3XOR*. Then any SoS refutation of ϕ has degree equal to the Gaussian width of ϕ.*

Lemma 5.7. *Let ϕ be a random instance of* 3XOR *over $x_1, \ldots, x_n$, with m equations. For $m = O(n)$ sufficiently large, the Gaussian width of ϕ is $\Omega(n)$.*

We first prove Lemma 5.6. Lemma 5.7 is fairly standard, so we will leave its proof until the end.

Proof. (of Lemma 5.6) Let $\mathcal{L}$ be the monomial equations corresponding to ϕ. We leave the easy direction (showing that if $\mathcal{L}$ has Gaussian width d, then $\mathcal{L}$ has an SoS refutation of degree d) as an exercise. For the harder direction, assume that $\mathcal{L}$ does not have a refutation of degree d, so $-1 = 1$ is not in $\mathcal{L}_{\leqslant d}$. Let $x_S = \prod_{i \in S} x_i$. To prove that there is no SoS refutation it suffices to define a pseudo-expectation operator $\tilde{\mathbb{E}}$ satisfying the following conditions:

1. $\tilde{\mathbb{E}}[x_S]$ is defined for all $|S| \leqslant d$, and extends linearly to all $\tilde{\mathbb{E}}[P(x)]$, for $P(x) \in \mathbb{R}[x]$ of degree at most d,
2. $\tilde{\mathbb{E}}[x^2 P(x)] = \tilde{\mathbb{E}}[P(x)]$,
3. $\tilde{\mathbb{E}}[1] = 1$,
4. $\tilde{\mathbb{E}}[x_{ijk}] = b_{ijk}$ for all equations $x_i x_j x_k = b_{ijk} \in \mathcal{L}$,
5. $\mathcal{M} \succeq 0$ where $\mathcal{M}_{S,T} = \tilde{\mathbb{E}}[x_S]\tilde{\mathbb{E}}[x_T]$.

Our pseudo-expectation operator is based on the canonical degree d Gaussian procedure defined above. Applying the procedure, each (multilinear) monomial x_S of degree $\leqslant d$ is labelled as *determined* if $x_S = -1$ or $x_S = 1$ is in $\mathcal{D}_{\leqslant d}$. (Note that since we are assuming $-1 = 1 \notin \mathcal{L}$, at most one of $x_S = 1$, $x_S = -1$ is in $\mathcal{L}_{\leqslant d}$.) Otherwise x_S is labelled as *undetermined*.

Our pseudo-expectation operates on degree $\leqslant d$ monomials as follows: First, we multilinearize by applying $x_i^2 = 1$. This corresponds to forcing all variables to have domain $\{\pm 1\}$, and thus we will have $\tilde{\mathbb{E}}[x_i^2 x_S] = \tilde{\mathbb{E}}[x_S]$. Secondly, if the monomial is determined, then we set its pseudo-expectation to this value, and otherwise (the monomial is undetermined), then we set its pseudo-expectation to 0, corresponding to it being set to -1 and 1 each with probability 1/2. We extend $\tilde{\mathbb{E}}$ to all degree $\leqslant d$ polynomials by linearity.

Conditions (1)-(4) are satisfied by construction, and thus we only need to show (5), that the moment matrix M defined by $\tilde{\mathbb{E}}$ is positive-semidefinite. To prove this, we exhibit vectors $\{v_S \mid S \subseteq [n],\ |S| \leqslant d/2\}$ such that for all S, T $|S|, |T| \leqslant d/2$, $\mathcal{M}_{S,T} = v_S^\top v_T$. That is, the vectors v_S give a Cholesky decomposition of $\mathcal{M}$.

It is important to note that a monomial equation can be derived in more than one way. For example, the monomial $x_1x_2x_3x_4 = -1$ could also be written as $x_1x_2 = -x_3x_4$. For this reason, our vectors have to be defined so that for any determined monomial x_T, and any pair S, S' such that $S \triangle S' = T$, $v_S^\top v_{S'}$ equals $\tilde{\mathbb{E}}[x_T]$. To this end, we define an equivalence relation $\sim$ on sets of size at most $d/2$ as follows: $S \sim T$ iff $x_{S\triangle T}$ is determined. It is not hard to check that $\sim$ is an equivalence relation. In particular, if $S \sim T$ and $T \sim U$, then $x_{S\triangle T} = b_1$ and $x_{T\triangle U} = b_2$ for some $b_1, b_2 \in \{\pm 1\}$, and thus $x_{S\triangle T\triangle T\triangle U} = x_{S\triangle U}$ is also defined with value b_1b_2.

The vectors will have dimension q, where q is the number of equivalence classes induced by $\sim$, and $v_S \in \{0, -1, 1\}^q$. Consider $S \subseteq [n]$ with $|S| \leqslant d/2$. For equivalence class j with representative I_j, if $S \sim I_j$ (so S is in equivalence class j) then $S \triangle I_j$ is defined to have some value $b \in \{-1, 1\}$, so we define $v_{S,j} = b$. For all other equivalence classes $j' \neq j$, let $v_{S,j} = 0$.

We want to show that for every pair $S, T \subseteq [n]$, with $|S|, |T| \leqslant d/2$, $v_S^\top v_T = \tilde{\mathbb{E}}[x_{S\triangle T}]$. First consider the case when $S \sim T$. Then S, T are both in some equivalence class $j \in [q]$ with representative element I_j. Thus $x_{S\triangle I_j}$ and $x_{T\triangle I_j}$ are both determined. Say that $x_{S\triangle I_j} = b_1$ and $x_{T\triangle I_j} = b_2$. Then $x_{S,T}$ is also determined and has value b_1b_2. So by definition, $v_{S,j} = b_1$ and $v_{T,j} = b_2$, and for all other $j' \neq j$, $v_{S,j'} = v_{T,j'} = 0$. Therefore, $v_S^\top v_T^\top = b_1b_2 = \tilde{\mathbb{E}}[x_{S\triangle T}]$.

Next, consider the case where $S \not\sim T$. Then $S \triangle T$ is not defined, so by definition $\tilde{\mathbb{E}}[x_{S\triangle T}] = 0$. Since S and T are in different equivalence classes, for every equivalence class j, at least one of $v_{S,j}$ and $v_{T,j}$ is 0. Thus $v_S^\top v_T = 0$ as desired.

This gives a Cholesky decomposition of $\mathcal{M}$ and thus we have proven that $\mathcal{M}$ is positive-semidefinite, to complete the proof. □

Proof. (of Lemma 5.7) We first prove using the probabilistic method that the clause-variable graph corresponding to a random 3XOR instance is highly expanding and thus has large boundary expansion. Secondly, we show that large boundary expansion implies large width.

A random instance ϕ gives rise to a clause-to-variable bipartite graph $G_\phi = (L, R, E)$: L consists of m vertices, one for each constraint, R consists of n vertices one for each variable, and edge (i, j) is present if and only if constraint C_i contains the variable x_j. An example is given in Figure 5.1. Let the neighborhood of constraint j be $\Gamma(\{j\})$, and for a subset of constraints $T \subseteq [m]$, let $\Gamma(T)$ be the neighborhood of T in G_ϕ.

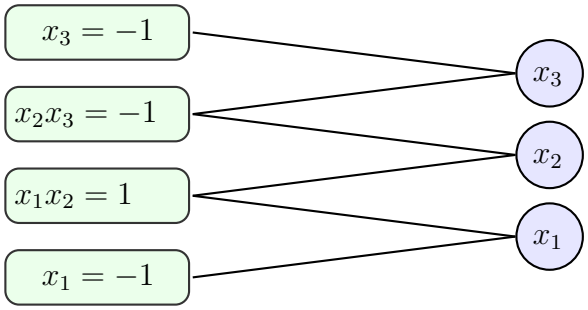


Figure 5.1: Constraint-variable graph $G_{\mathcal{L}}$ for the CSP $\mathcal{L}$ from Example 5.5.

Definition 5.8 ((t,β)-Expander Graph)**.** G is a (t,β)-expander if for all subsets $T \subseteq L$, $|T| \leqslant t$, $|\Gamma(T)| \geqslant \beta|T|$.

The following lemma shows that random 3XOR instance have good expansion with high probability.

Lemma 5.9. *For $m = c_\varepsilon n$ and any constant $\delta > 0$ there exists a constant $\eta > 0$ depending on δ and c_ε such that with probability 0.99, the constraint graph G_ϕ is $(\eta n, 2-\delta)$-expanding.*

Proof. Let Y_S be the event that the set $S \subseteq [m]$ of size $s \leqslant \eta n$ has expansion less than $2-\delta$. There are at most $\binom{n}{(2-\delta)s}$ possible neighborhoods, and each vertex has $\binom{(2-\delta)s}{3}$ possible individual neighborhoods from this total neighborhood, each occurring with probability $\frac{1}{n^3}$. By extension there are $\binom{\binom{(2-\delta)s}{3}}{s}$ possible settings of all the edges on S, each of which occurs with probability $(\frac{1}{n^3})^s$. Thus, we get that

$$\begin{aligned}
\mathbb{P}[Y_S > 1] &\leqslant \tbinom{n}{(2-\delta)s}\tbinom{\binom{(2-\delta)s}{3}}{s}\left(\tfrac{1}{n^3}\right)^s \\
&\leqslant \left(\tfrac{ne}{(2-\delta)s}\right)^{(2-\delta)s}\left(\left(\tfrac{(2-\delta)se}{3}\right)^3\tfrac{1}{s}\right)^s\tfrac{1}{n^{3s}} \\
&\leqslant C\left(\tfrac{n^{2-\delta}}{s^{2-\delta}}\cdot s^2\cdot\tfrac{1}{n^3}\right)^s \\
&\leqslant C\left(\tfrac{s^\delta}{n^{1+\delta}}\right)^s \\
&\leqslant C\left(\tfrac{s}{n}\right)^{\delta s} n^{-s}
\end{aligned}$$

. Taking the sum over all $\binom{m}{s}$ possible S gives us

$$\begin{aligned}
\mathbb{P}[\exists S, |S| \leqslant s \mid Y_S > 1] &\leqslant m^s \cdot C(\tfrac{s}{n})^{\delta s} n^{-s} \\
&\leqslant C(\tfrac{s}{n})^{\delta s}(\tfrac{c_\varepsilon n}{n})^s \\
&= \left(c_{\varepsilon,\delta}\tfrac{s}{n}\right)^{\delta s}
\end{aligned}$$

which is at most 0.01 for $s \leqslant \eta n$ as long as $\eta \leqslant \frac{1}{2c_{\varepsilon,\delta}}$. □

Definition 5.10 (Boundary Expander)**.** G_ϕ is a (t,γ)-*boundary expander* if for all subsets $T \subseteq L$, $|T| \leqslant t$, $|\mathcal{B}(T)| \geqslant \beta|T|$, where $\mathcal{B}(T)$ is the *boundary* of $\Gamma(T)$ – that is, the set of all vertices from R with exactly one neighbor in T.

It is not hard to see that if G 3-regular (t,β)-expander graph, then G is a $(t, 2\beta-3)$-boundary expander. Fix any set $S \subseteq L$ of size $s \leqslant t$. Letting the number of edges incident with S be $E(S)$, we have:

$$\begin{aligned} 3|S| &= E(S) \\ &\geqslant |\mathcal{B}(S)| + 2|\Gamma(S)/\mathcal{B}(S)| \\ &= 2|\Gamma(S)| - |\mathcal{B}(S)| \\ &\geqslant 2\beta|S| - |\mathcal{B}(S)| \end{aligned}$$

Rearranging, $|\mathcal{B}(S))| \geqslant (2\beta-3)|S|$ as desired.

If we choose δ in our expansion lemma to be strictly less than 0.5, say $\delta = 0.3$, then our graph will have constant boundary expansion for sets up to the same size. When $\delta = 0.3$ the boundary expansion will be $2(1.7) - 3 = 0.4$, and so from here on out we assume that G_ϕ is a $(\eta n, 0.4)$-boundary expander.

Let $\mathcal{L}$ refer to the monomial equations corresponding to ϕ. We want to show that any Gaussian refutation of $\mathcal{L}$ has width $\Omega(n)$. Let $\mathcal{S} = \{l_1, l_2, \ldots, l_q\}$ be a Gaussian refutation of $\mathcal{L}$. Label each equation $l_i \in \mathcal{S}$ with the set $Ax(l_i) \subseteq \mathcal{L}$ of initial equations that were used to derive l_i. The size of $Ax(l_i)$ is a subadditive complexity measure. Viewing the derivation $\mathcal{S}$ as a directed acyclic graph, the initial equations are in $\mathcal{L}$, and thus $|Ax(l_i)| = 1$ for each l_i that is a leaf/axiom of $\mathcal{S}$. If l_i and l_j derive l_k, then $|Ax(l_k)| \leqslant |Ax(l_i)| + |Ax(l_j)|$. Finally, since we assumed that G_ϕ is a $(\eta n, .4)$-boundary expander, this implies that $|Ax(-1=1)| \geqslant \eta n)$ since all variables have cancelled out and for any subset of at most ηn initial equations, there are a lot of boundary variables, so they cannot derive $-1 = 1$.

Thus by subadditivity there must exist an equation l in the proof such that $\eta n/3 \leqslant |Ax(l)| \leqslant 2\eta n/3$. By boundary expansion, there must be at least $0.4|Ax(l)|$ variables in l (since boundary variables cannot have cancelled out), and therefore the width of l is $\Omega(n)$. □

This completes the SoS lower bound for 3XOR, which implies the same lower bound for SA as well.

5.2 Other SoS Lower Bounds

There is a fairly long history of degree lower bounds preceding and following this result. Lower bounds for Nullstellensatz refutations were obtained in many papers. (For example, see [25, 64] and references therein.) Lower bounds for the Polynomial Calculus (a stronger subsystem of SoS) were then obtained by many works. (i.e., [48, 37, 130].) Grigoriev's proof [65] presented in Section 5.1 builds on the earlier lower bounds for random 3XOR equations that were proven for the Nullstellensatz and Polynomial Calculus [64, 37]. The same lower bound for 3XOR was also obtained independently by Schoenebeck [139]. There is also a fairly long history of degree lower bounds known for SA including lower bounds for

unsatisfiable systems of equations, and in addition a long series of papers proving integrality gaps for SA. For example, see [58, 31, 4, 43, 54, 111] and references therein.

Building on many on these techniques, a line of work sought to prove lower bounds on SoS for various NP-hard optimization problems. Schoenebeck [139] proved unconditional lower bounds for high-degree SoS proofs of several NP-hard combinatorial optimization problems such as MAX 3SAT and MAX Independent Set. These results follow by fairly straightforward reductions to the 3XOR lower bound for SoS. Tulsiani [148] gave a general method to do reductions within the SoS framework and extended Schonebeck's lower bounds to a large class of constraint satisfaction problems with *pairwise uniform* and *algebraically linear* predicates. Barak, Chan and Kothari [16] finally extended these lower bounds to all *pairwise uniform* predicates. Here we give a simple example, showing how SoS degree lower bounds for 3SAT can be obtained by a reduction to the 3XOR lower bound.

Corollary 5.11. *Let ϕ be an instance of* 3SAT *on n variables with $m = c_\varepsilon n$ constraints (for c_ε a constant only dependent on ε) chosen as follows: for each constraint we choose $i, j, k \sim [n]$, $e_i, e_j, e_k \sim \{0,1\}$ i.i.d. and take our clause to be $(x_i^{e_i} \vee x_j^{e_j} \vee x_k^{e_k})$, where $x^0 = x$, $x^1 = \overline{x}$. Then with probability at least 0.99,*

- *(Soundness) every assignment $\alpha \in \{\pm 1\}^n$, satisfies at most $\left(\frac{7}{8} + \varepsilon\right) m$ clauses*
- *(Completeness) there exists a pseudo-distribution of degree $\Omega(n)$ such that in expectation all clauses of ϕ are satisfied,*

where the probability is over the choice of ϕ.

Proof. Let ϕ be our random instance. For the soundness, we leave it as an exercise to the reader to use the same argument as in the 3XOR case. For completeness, we define $\phi_\oplus$ to be a 3XOR instance as follows: for each clause $C : (x_i^{e_i} \vee x_j^{e_j} \vee x_k^{e_k})$, we have a constraint $C' : x_i x_j x_k = a_{ijk}$, where $a_{ijk} = (-1)^{e_i + e_j + e_k}$. This is a random instance of 3XOR, as i, j, k were chosen i.i.d. and $(-1)^{e_i+e_j+e_k}$ is uniformly distributed over ± 1. Thus, with probability 0.99 there exists a pseudo-distribution satisifying all constraints in $\phi_\oplus$. The result follows by noting that if we transform this pseudo-distribution back to $\{0,1\}$ valued variables via $x \to \frac{1-x}{2}$, any assignment in the support of the pseudo-distribution satisfies C', and any assignment satisfying C' also satisfies C. □

Recently there has been a surge of works for showing SoS lower bounds for *average-case* settings. [97] proved a sharp SoS lower bound to precisely characterize the number of clauses required for refuting a constraint satisfaction problem with a given predicate. Following a sequence of work [112, 75], Barak et. al. [18] proved an optimal lower bound for the planted clique problem via the new technique of *pseudocalibration*. This technique was later used in [76] to prove strong lower bounds for optimizing random degree 3 polynomials over the unit sphere and Sparse principal component analysis (PCA).

5.3 Applications of Lower Bounds

So far we have focused on using SoS degree bounds and integrality gaps is to rule out LP and SDP relaxations of NP-hard optimization problems. Quite surprisingly, SA and SoS lower bounds can also be used to rule out other very general classes of algorithms. Again these proofs are reductions, albeit much more sophisticated ones. The reductions we discuss next are examples of *hardness escalation* whereby (SA or SoS) lower bounds for computing (or approximating) a function in a weaker model (LP or SDP) can be lifted via function composition to obtain lower bounds in a stronger model of computation. Some of the early examples of lifting include Sherstov's pattern matrix method [142], and Raz and McKenzie's separation of the monotone NC hierarchy [129]. In recent years, many lifting theorems have been discovered, and have in turn resolved a large number of open problems in circuit complexity, game theory and proof complexity (i.e., [62, 61, 63, 57]). Unfortunately, the ideas and even the setup for these results are beyond the scope of this manuscript, so we will settle with at least mentioning some of the main lifting theorems that use SA or SoS lower bounds as their starting point.

First, SoS degree lower bounds, and more specifically Nullstellensatz degree bounds, have been used to prove exponential size lower bounds for monotone *circuit models*. Monotone span programs capture the power of reasoning using linear algebra in order to compute monotone functions [82]. [119] prove a lifting theorem between Nullstellensatz degree and monotone span program size that implies exponential lower bounds on the size of monotone span programs for several functions (and over all fields). By the known equivalence between monotone span programs and linear secret sharing schemes, this also implies exponential lower bounds for the latter.

Secondly, SoS degree lower bounds have been used to prove exponential lower bounds for extended formulations [150]. More specifically, SA degree bounds were shown to imply lower bounds for LP extended formulations [40, 96] and similarly, SoS lower bounds were shown to imply lower bounds for SDP extended formulations [95].

As mentioned above, all of these applications of SoS lower bounds are instantiations of *lifting theorems* whereby query complexity lower bounds for a particular search problem are *lifted* via composition with an inner gadget, in order to obtain stronger communication complexity lower bounds in the corresponding communication measure. For example, [129, 62] lift tree-like Resolution height (here the query measure is decision tree height) to deterministic communication complexity, which in turn is known to be equivalent to monotone formula size. [119] lift Nullstellensatz degree (here the query measure is polynomial degree) to its corresponding communication measure, which in turn is known to be equivalent to monotone span program size. And [96] lift SA degree (where the query measure is junta degree) to the nonnegative rank of the communication matrix, which in turn is known to be equivalent to LP extension complexity [150]. Finally, [95] proved via a lifting theorem, that SoS lower bounds imply SDP extension complexity lower bounds.

Acknowledgements

The authors are grateful to Aleksandar Nikolov, Robert Robere, and Morgan Shirley for their comments and suggestions on an earlier version on this monograph. The authors would like to thank Ian Mertz for his suggestions on the presentation of the 3XOR lower bound in Section 5.1.

Bibliography

[1] Miklós Ajtai. The complexity of the pigeonhole principle. *Combinatorica*, 14(4):417–433, 1994.

[2] Michael Alekhnovich and Alexander A. Razborov. Resolution is not automatizable unless W[P] is tractable. *SIAM J. Comput.*, 38(4):1347–1363, 2008.

[3] Noga Alon and Assaf Naor. Approximating the cut-norm via grothendieck's inequality. *SIAM J. Comput.*, 35(4):787–803, 2006.

[4] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 563–572, 2010.

[5] Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(2):19–51, 2006.

[6] 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.

[7] Sanjeev Arora, Satish Rao, and Umesh V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, 2009.

[8] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, 1998.

[9] Emil Artin. Über die zerlegung definiter funktionen in quadrate. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 5(1):100–115, Dec 1927.

[10] Takao Asano and David P. Williamson. Improved approximation algorithms for MAX SAT. *J. Algorithms*, 42(1):173–202, 2002.

[11] Albert Atserias and Tuomas Hakoniemi. Size-degree trade-offs for sums-of-squares and positivstellensatz proofs. *CoRR*, abs/1811.01351, 2018.

[12] Albert Atserias and Moritz Müller. Automating resolution is np-hard. *CoRR*, abs/1904.02991, 2019.

[13] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for max bisection. *ACM Trans. Algorithms*, 13(1):2:1–2:27, 2016.

[14] Boaz Barak, Zvika Brakerski, Ilan Komargodski, and Pravesh K. Kothari. Limits on low-degree pseudorandom generators (or: Sum-of-squares meets program obfuscation). In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, pages 649–679, 2018.

[15] Boaz Barak, Fernando G. S. L. Brandão, Aram Wettroth Harrow, Jonathan A. Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 307–326, 2012.

[16] Boaz Barak, Siu On Chan, and Pravesh Kothari. Sum of squares lower bounds from pairwise independence. *CoRR*, abs/1501.00734, 2015.

[17] Boaz Barak, Samuel B. Hopkins, Aayush Jain, Pravesh Kothari, and Amit Sahai. Sum-of-squares meets program obfuscation, revisited. In *Advances in Cryptology - EUROCRYPT 2019 - 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19-23, 2019, Proceedings, Part I*, pages 226–250, 2019.

[18] Boaz Barak, Samuel B. Hopkins, Jonathan A. Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 428–437, 2016.

[19] Boaz Barak, Jonathan A. Kelner, and David Steurer. Rounding sum-of-squares relaxations. *CoRR*, abs/1312.6652, 2013.

[20] Boaz Barak, Jonathan A. Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 143–151, 2015.

[21] Boaz Barak, Pravesh K. Kothari, and David Steurer. Quantum entanglement, sum of squares, and the log rank conjecture. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 975–988, 2017.

[22] Boaz Barak and Ankur Moitra. Noisy tensor completion via the sum-of-squares hierarchy. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 417–445, 2016.

[23] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 472–481, 2011.

[24] Boaz Barak and David Steurer. Sum-of-squares proofs and the quest toward optimal algorithms. *CoRR*, abs/1404.5236, 2014.

[25] Paul Beame, Russell Impagliazzo, Jan Krajícek, Toniann Pitassi, and Pavel Pudlák. Lower bound on hilbert's nullstellensatz and propositional proofs. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 794–806, 1994.

[26] Paul Beame, Russell Impagliazzo, Jan Krajícek, Toniann Pitassi, Pavel Pudlák, and Alan R. Woods. Exponential lower bounds for the pigeonhole principle. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 200–220, 1992.

[27] Paul Beame and Toniann Pitassi. Simplified and improved resolution lower bounds. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 274–282, 1996.

[28] Paul Beame and Toniann Pitassi. Current trends in theoretical computer science, 2001.

[29] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free bits, pcps, and nonapproximability-towards tight results. *SIAM J. Comput.*, 27(3):804–915, 1998.

[30] Eli Ben-Sasson and Avi Wigderson. Short proofs are narrow - resolution made simple. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 517–526, 1999.

[31] Siavosh Benabbas, Konstantinos Georgiou, Avner Magen, and Madhur Tulsiani. SDP gaps from pairwise independence. *Theory of Computing*, 8(1):269–289, 2012.

[32] Christoph Berkholz. The relation between polynomial calculus, sherali-adams, and sum-of-squares proofs. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:154, 2017.

[33] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997.

[34] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. Non-automatizability of bounded-depth frege proofs. *Computational Complexity*, 13(1-2):47–68, 2004.

[35] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. On interpolation and automatization for frege systems. *SIAM J. Comput.*, 29(6):1939–1967, 2000.

[36] Josh Buresh-Oppenheim, Matthew Clegg, Russell Impagliazzo, and Toniann Pitassi. Homogenization and the polynomial calculus. *Computational Complexity*, 11(3-4):91–108, 2002.

[37] Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. Syst. Sci.*, 62(2):267–289, 2001.

[38] Samuel R Buss and Toniann Pitassi. Good degree bounds on nullstellensatz refutations of the induction principle. *Journal of Computer and System Sciences*, 57(2):162 – 171, 1998.

[39] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

[40] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. *J. ACM*, 63(4):34:1–34:22, 2016.

[41] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Encyclopedia of Machine Learning and Data Mining*, page 231. Springer US, 2017.

[42] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 205–214, 2006.

[43] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for sherali-adams relaxations. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 283–292, 2009.

[44] 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.

[45] Eden Chlamtac, Konstantin Makarychev, and Yury Makarychev. How to play unique games using embeddings. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 687–696, 2006.

[46] Vasek Chvátal. Determining the stability number of a graph. *SIAM J. Comput.*, 6(4):643–662, 1977.

[47] Vasek Chvátal and Endre Szemerédi. Many hard examples for resolution. *J. ACM*, 35(4):759–768, 1988.

[48] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 174–183, 1996.

[49] Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 174–183, 1996.

[50] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979.

[51] Stefan S. Dantchev, Barnaby Martin, and Mark Nicholas Charles Rhodes. Tight rank lower bounds for the sherali-adams proof system. *Theor. Comput. Sci.*, 410(21-23):2054–2063, 2009.

[52] Martin Davis, George Logemann, and Donald W. Loveland. A machine program for theorem-proving. *Commun. ACM*, 5(7):394–397, 1962.

[53] Martin Davis and Hilary Putnam. A computing procedure for quantification theory. *J. ACM*, 7(3):201–215, 1960.

[54] Wenceslas Fernandez de la Vega and Claire Kenyon-Mathieu. Linear programming relaxations of maxcut. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 53–61, 2007.

[55] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 32:1–32:17, 2016.

[56] Nicola Galesi and Massimo Lauria. On the automatizability of polynomial calculus. *Theory Comput. Syst.*, 47(2):491–506, 2010.

[57] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 902–911, 2018.

[58] Konstantinos Georgiou, Avner Magen, Toniann Pitassi, and Iannis Tourlakis. Integrality gaps of 2-o(1) for vertex cover sdps in the lov[a-acute]sz–schrijver hierarchy. *SIAM J. Comput.*, 39(8):3553–3570, 2010.

[59] Michel X. Goemans. Semidefinite programming in combinatorial optimization. *Math. Program.*, 79:143–161, 1997.

[60] Michel X. Goemans and David P. Williamson. .879approximationn algorithms for max cut and max 2sat. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC '94, pages 422–431, New York, NY, USA, 1994. ACM.

[61] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. *SIAM J. Comput.*, 45(5):1835–1869, 2016.

[62] Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic communication vs. partition number. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 1077–1088, 2015.

[63] Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 132–143, 2017.

[64] Dima Grigoriev. Tseitin's tautologies and lower bounds for nullstellensatz proofs. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 648–652, 1998.

[65] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theor. Comput. Sci.*, 259(1-2):613–622, 2001.

[66] Dima Grigoriev and Edward A. Hirsch. Algebraic proof systems over formulas. *Theor. Comput. Sci.*, 303(1):83–102, 2003.

[67] Dima Grigoriev and Nicolai Vorobjov. Complexity of null-and positivstellensatz proofs. *Ann. Pure Appl. Logic*, 113:153–160, 2001.

[68] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018.

[69] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[70] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[71] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with PSD objectives. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 482–491, 2011.

[72] Armin Haken. The intractability of resolution. *Theor. Comput. Sci.*, 39:297–308, 1985.

[73] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[74] David Hilbert. Ueber die darstellung definiter formen als summe von formenquadraten. *Mathematische Annalen*, 32(3):342–350, Sep 1888.

[75] Samuel B. Hopkins, Pravesh Kothari, Aaron Henry Potechin, Prasad Raghavendra, and Tselil Schramm. On the integrality gap of degree-4 sum of squares for planted clique. *ACM Trans. Algorithms*, 14(3):28:1–28:31, 2018.

[76] Samuel B. Hopkins, Pravesh K. Kothari, Aaron Potechin, Prasad Raghavendra, Tselil Schramm, and David Steurer. The power of sum-of-squares for detecting hidden structures. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 720–731, 2017.

[77] Samuel B. Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1021–1034, 2018.

[78] A. Hurwitz. Ueber den vergleich des arithmetischen und des geometrischen mit-tels. *Journal fur die reine und angewandte Mathematik*, 1891.

[79] Russell Impagliazzo, Pavel Pudlák, and Jirí Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999.

[80] DB Iudin and AS Nemirovskii. Informational complexity and effective methods of solution for convex extremal problems. matekon: Translations of russian and east european math. *Economics*, 13:3–25, 1976.

[81] Cédric Josz and Didier Henrion. Strong duality in lasserre's hierarchy for polynomial optimization. *Optimization Letters*, 10:3–10, 2016.

[82] Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of the Eigth Annual Structure in Complexity Theory Conference, San Diego, CA, USA, May 18-21, 1993*, pages 102–111, 1993.

[83] Narendra Karmarkar. A polynomial-time algorithm for solving linear programs. *Math. Oper. Res.*, 5(1):iv–iv, February 1980.

[84] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[85] Leonid G Khachiyan. A polynomial algorithm in linear programming. In *Doklady Academii Nauk SSSR*, volume 244, pages 1093–1096, 1979.

[86] L.G. Khachiyan. Polynomial algorithms in linear programming. *USSR Computational Mathematics and Mathematical Physics*, 20(1):53 – 72, 1980.

[87] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity, Montréal, Québec, Canada, May 21-24, 2002*, page 25, 2002.

[88] Subhash Khot. On the unique games conjecture (invited survey). In *Proceedings of the 2010 IEEE 25th Annual Conference on Computational Complexity*, CCC '10, pages 99–121, Washington, DC, USA, 2010. IEEE Computer Society.

[89] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 146–154, Washington, DC, USA, 2004. IEEE Computer Society.

[90] 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.

[91] Subhash Khot, Dor Minzer, and Muli Safra. Pseudorandom sets in grassmann graph have near-perfect expansion. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:6, 2018.

[92] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008.

[93] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative-type metrics into l1. *J. ACM*, 62(1):8:1–8:39, 2015.

[94] Adam R. Klivans, Pravesh K. Kothari, and Raghu Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018.*, pages 1420–1430, 2018.

[95] Pravesh Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. *CoRR*, abs/1610.02704, 2016.

[96] Pravesh K. Kothari, Raghu Meka, and Prasad Raghavendra. Approximating rectangles by juntas and weakly-exponential lower bounds for LP relaxations of csps. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 590–603, 2017.

[97] Pravesh K. Kothari, Ryuhei Mori, Ryan O'Donnell, and David Witmer. Sum of squares lower bounds for refuting any CSP. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 132–145, 2017.

[98] Pravesh K. Kothari and Jacob Steinhardt. Better agnostic clustering via relaxed tensor norms. *CoRR*, abs/1711.07465, 2017.

[99] Pravesh K. Kothari and David Steurer. Outlier-robust moment-estimation via sum-of-squares. *CoRR*, abs/1711.11581, 2017.

[100] Jan Krajícek and Pavel Pudlák. Some consequences of cryptographical conjectures for s^1_2 and EF. *Inf. Comput.*, 140(1):82–94, 1998.

[101] Jean-Louis Krivine. Anneaux préordonnés. *Journal d'Analyse Mathématique*, 12(1):307–326, 1964.

[102] Jean B. Lasserre. An explicit exact sdp relaxation for nonlinear 0-1 programs. In Karen Aardal and Bert Gerards, editors, *Integer Programming and Combinatorial Optimization*, pages 293–303, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[103] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[104] Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the max 2-sat and max di-cut problems. In *International Conference on Integer Programming and Combinatorial Optimization*, pages 67–82. Springer, 2002.

[105] Fu Li, Iddo Tzameret, and Zhengyu Wang. Non-commutative formulas and frege lower bounds: a new characterization of propositional proofs. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 412–432, 2015.

[106] Nathan Linial. Finite metric spaces: combinatorics, geometry and algorithms. In *Proceedings of the 18th Annual Symposium on Computational Geometry, Barcelona, Spain, June 5-7, 2002*, page 63, 2002.

[107] L. Lovász. *Semidefinite Programs and Combinatorial Optimization*, pages 137–194. Springer New York, New York, NY, 2003.

[108] László Lovász. On the shannon capacity of a graph. *IEEE Trans. Information Theory*, 25(1):1–7, 1979.

[109] László Lovász. Semidefinite programs and combinatorial optimization, 1995.

[110] Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 438–446, 2016.

[111] Claire Mathieu and Alistair Sinclair. Sherali-adams relaxations of the matching polytope. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 293–302, 2009.

[112] Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 87–96, 2015.

[113] T.S. Motzkin. The arithmetic-geometric inequality. *Inequalities (Proc. Symposium on Inequalities, edited by O. Shisha, Academic Press*, pages 205–224, 1967.

[114] Jakob Nordström. On the interplay between proof complexity and SAT solving. *SIGLOG News*, 2(3):19–44, 2015.

[115] Ryan O'Donnell. SOS is not obviously automatizable, even approximately. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 59:1–59:10, 2017.

[116] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.

[117] Pablo A Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical programming*, 96(2):293–320, 2003.

[118] Toniann Pitassi. Algebraic propositional proof systems. In *Descriptive Complexity and Finite Models, Proceedings of a DIMACS Workshop 1996, Princeton, New Jersey, USA, January 14-17, 1996*, pages 215–244, 1996.

[119] Toniann Pitassi and Robert Robere. Lifting nullstellensatz to monotone span programs over any field. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1207–1219, 2018.

[120] Toniann Pitassi and Nathan Segerlind. Exponential lower bounds and integrality gaps for tree-like lovász-schrijver procedures. *SIAM J. Comput.*, 41(1):128–159, 2012.

[121] Toniann Pitassi and Iddo Tzameret. Algebraic proof complexity: progress, frontiers and challenges. *SIGLOG News*, 3(3):21–43, 2016.

[122] Aaron Potechin and David Steurer. Exact tensor completion with sum-of-squares. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 1619–1673, 2017.

[123] Mihai Putinar. Positive polynomials on compact semi-algebraic sets. *Indiana University Mathematics Journal*, 42(3):969–984, 1993.

[124] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 245–254, 2008.

[125] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 755–764, 2010.

[126] Prasad Raghavendra and Benjamin Weitz. On the bit complexity of sum-of-squares proofs. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, pages 80:1–80:13, 2017.

[127] M. Ramana, L. Tunçel, and H. Wolkowicz. Strong duality for semidefinite programming. *SIAM Journal on Optimization*, 7(3):641–662, 1997.

[128] Motakuri V. Ramana. An exact duality theory for semidefinite programming and its complexity implications. *Mathematical Programming*, 77(1):129–162, Apr 1997.

[129] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Combinatorica*, 19(3):403–435, 1999.

[130] Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, 1998.

[131] Alexander A. Razborov. Guest column: Proof complexity and beyond. *SIGACT News*, 47(2):66–86, 2016.

[132] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

[133] Oded Regev and Aravindan Vijayaraghavan. On learning mixtures of well-separated gaussians. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 85–96, 2017.

[134] Bruce Reznick. Some concrete aspects of hilbert's 17th problem. In *In Contemporary Mathematics*, pages 251–272. American Mathematical Society, 1996.

[135] Thomas Rothvoß. The lasserre hierarchy in approximation algorithms. *Lecture Notes for the MAPSP*, pages 1–25, 2013.

[136] Sartaj Sahni and Teofilo Gonzalez. P-complete approximation problems. *Journal of the ACM (JACM)*, 23(3):555–565, 1976.

[137] Claus Scheiderer. Sums of squares on real algebraic curves. *Mathematical journal*, 245(4):725–760, 2003.

[138] Konrad schmidt. Around hilbert's 17th problem. *Documenta Mathematica, Optimization stories, extra volume ISMP*, pages 433–438, 2012.

[139] Grant Schoenebeck. Linear level lasserre lower bounds for certain k-csps. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 593–602, 2008.

[140] Nathan Segerlind. The complexity of propositional proofs. *Bulletin of Symbolic Logic*, 13(4):417–481, 2007.

[141] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations and convex hull characterizations for mixed-integer zero-one programming problems. *Discrete Applied Mathematics*, 52(1):83–106, 1994.

[142] Alexander A. Sherstov. The pattern matrix method. *SIAM J. Comput.*, 40(6):1969–2000, 2011.

[143] N. Z. Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics*, 23(5):695–700, Sep 1987.

[144] Naum Z Shor. Cut-off method with space extension in convex programming problems. *Cybernetics and systems analysis*, 13(1):94–96, 1977.

[145] G. Stengle. A nullstellensatz and positivstellensatz in semialgebraic geometry. *Math. Ann.*, 207:87–97, 1994.

[146] Luca Trevisan. On khot's unique games conjecture, 2011.

[147] Mária Trnovská. Strong duality conditions in semidefinite programming. *Journal of Electrical Engineering*, 56(12):1–5, 2005.

[148] Madhur Tulsiani. CSP gaps and reductions in the lasserre hierarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 303–312, 2009.

[149] Alasdair Urquhart. Hard examples for resolution. *J. ACM*, 34(1):209–219, 1987.

[150] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs (extended abstract). In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 223–228, 1988.

Appendix

Section 2.2.3 Missing Proofs

Claim 2.29. *If there is a non-negative linear combination of the constraints of* $\mathsf{SA}_d(\mathcal{P})$ *equalling* $c_0 \in \mathbb{R}$*, then there exists a degree* d *SA derivation of* c_0 *from* $\mathcal{P}$*.*

Proof. Denote by $L(y)$ the SA constraint corresponding to the multilinearization of $P_i(x)J_{S,T}(x)$ over the placeholder variables y, for $P_i(x) \geqslant 0 \in \mathcal{P} \cup \{1 \geqslant 0\}$, and $J_{S,T}(x)$ a degree at most $(d - \deg(P_i))$ non-negative junta. Suppose that there exists a non-negative linear combination

$$c_i \sum_{i=1}^{\ell} L(y) = c_0. \tag{5.1}$$

We can translate this into a sum over the x-variables by replacing each linearized $L(y)$ by its corresponding term in the x variables, $J_{S_i,T_i}(x) \cdot P_i(x)$

$$c_i \sum_{i=1}^{\ell} P_i(x) \cdot J_{S_i,T_i}(x).$$

It may no longer be the case that this evaluates to c because terms which previously cancelled in the linearized sum may no longer cancel in this non-linearized sum. The axioms $\pm(x_i - x_i^2) \geqslant 0$ can be used to mimic the linearization. Each term $c\prod_{i\in[k]} x_i^{a_i}$ can be linearized by introducing the following telescoping sum

$$\sum_{i=1}^{k} \left(\sum_{\ell=0}^{a_i-2} c(x_i - x_i^2) x_i^{\ell} \prod_{j>i} x_j^{a_j} \prod_{j<i} x_j \right).$$

Each term in this sum is of the form $(x_i^2 - x_i) \cdot J_{S,T}(x)$ for some S, T with $|S| + |T| \leqslant d - 2$, and therefore is a valid inequality for SA. The degree of this proof is the maximum degree of among the constraints being linearized, and therefore is bounded above by d. □

Section 3.1 Missing Proofs

Lemma 3.41. *For any set of polynomial inequalities* $\mathcal{P} = \{P_1(x) \geqslant 0, \ldots, P_m(x) \geqslant 0\}$*,* $\mathsf{SOS}_d(\mathcal{P})$ *satisfies the following properties:*

1. $0 \leqslant y_J \leqslant y_I \leqslant 1$ *for every* $I \subseteq J \subseteq [n]$ *with* $|J| \leqslant d$.
2. $\sum_{J \subseteq T} (-1)^{|J|} y_{S \cup J} \geqslant 0$ *for every non-negative* d*-junta* $J_{S,T}(x)$.
3. *If* $\alpha \in \{0,1\}^n$ *satisfies every* $P_i(x) \geqslant 0 \in \mathcal{P}$*, then* $\alpha \in \operatorname{proj}_{[n]}(\mathsf{SOS}_d(\mathcal{P}))$ *for every* $d \geqslant \deg(\mathcal{P})$.[1]

Proof. To prove (1), we will use the fact that the diagonal entries of a symmetric PSD matrix are non-negative (Claim 3.5), and that $\mathcal{M}_d(y)$ is symmetric PSD. First, let $I \subseteq [n]$ with $|I| \leqslant d$. Observe that y_I occurs on the diagonal of $\mathcal{M}_d(y)$ and therefore $y_I \geqslant 0$. To prove that $y_I \leqslant 1$, define $u \in \mathbb{R}^{\binom{n}{\leqslant d}}$ as

$$u_K = \begin{cases} 1 \text{ if } K = \emptyset, \\ -1 \text{ if } K = I, \\ 0 \text{ otherwise.} \end{cases}$$

Then,

$$u^\top \mathcal{M}_d(y) u = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_\emptyset & y_I \\ y_I & y_I \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = y_\emptyset - y_I \geqslant 0,$$

where the final inequality follows because $\mathcal{M}_d(y) \succeq 0$. Finally, because $y_\emptyset = 1$, we have $1 - y_I \geqslant 0$. Now, for any $I \subseteq J \subseteq [n]$ with $|J| \leqslant 2d$, define $u' \in \mathbb{R}^{n^{2d}+1}$ as

$$u'_K = \begin{cases} 1 \text{ if } K = I, \\ -1 \text{ if } K = J, \\ 0 \text{ otherwise.} \end{cases}$$

Then, as before,

$$u'^\top \mathcal{M}_d(y) u' = \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_I & y_J \\ y_J & y_J \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = y_I - y_J \geqslant 0.$$

Therefore, we have $0 \leqslant y_J \leqslant y_I \leqslant 1$.

For (2), let $\sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T}$ be the y-variable representation of some non-negative d-junta $J_{S,T}(x)$. Define the vector $v \in \mathbb{R}^{\binom{n}{\leqslant d}}$ as $v_I = 1$ if y_I occurs positively in the junta, $v_I = -1$ if y_I occurs negatively, and $v_I = 0$ if y_I is absent from the junta. We claim that

$$v^\top \mathcal{M}_d(y) v = \sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T}.$$

Indeed, multiplying by v is equivalent to multiplying the principal submatrix M of $\mathcal{M}_d(y)$ corresponding to rows and columns indexed by $\{I : v_I \neq 0\}$ by the vector $v' := v \restriction_{v_I \neq 0}$. First, let's look at the vector $v'^\top M$. The first entry of this vector, corresponding to multiplying $v'^\top$ by the column indexed by S, is $\sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T}$. We claim that the rest of

[1] Recall that $\operatorname{proj}_{[n]}(\mathcal{P}) = \{\alpha \restriction_{y_{\{1\}}, \ldots, y_{\{n\}}} : \alpha \in \mathcal{P}\}$, the orthogonal projection of $\mathcal{P}$ to the first n variables.

the entries in this vector are 0. To prove this, we will use the following property of juntas $J_{S,T}(x)$: for any $i \in T$, because T is disjoint from S, we can write

$$\sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T} = \sum_{J \subseteq T : i \in J} (-1)^{|J|} y_{S \cup J} - \sum_{J \subseteq T : i \notin J} (-1)^{|J|} y_{S \cup J},$$

as well as the observation that the only non-zero entries v_L are such that $S \subseteq L \subseteq S \cup T$. Now, consider an entry, other than the first, in the vector $v'^{\top} M$. For this entry to be included in $v'^{\top} M$, it must be non-zero in $v^{\top} \mathcal{M}_d(y)$, and so by the definition of v it must correspond to a column of $\mathcal{M}_d(y)$ indexed by $S \cup K$ for some $K \subseteq T$ with $K \neq \emptyset$. Denote this entry by $(v'^{\top} M)_{S \cup K}$, and let $i \in K \cap T$. Then,

$$\begin{aligned}(v'^{\top} M)_{S \cup K} &= \sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T \cup K}, \\ &= \sum_{J \subseteq T : i \in J} (-1)^{|J|} y_{S \cup J \cup K} - \sum_{J \subseteq T : i \notin J} (-1)^{|J|} y_{S \cup J \cup K} = 0.\end{aligned}$$

Therefore,

$$v^{\top} \mathcal{M}_d(y) v = v'^{\top} M v' = \left(\sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T},\ 0,\ 0,\ \ldots,\ 0 \right) v' = \sum_{J \subseteq T} (-1)^{|J|} y_{S \cup T} \geqslant 0,$$

where the last equality follows because the first entry of v' is the entry $v_S = 1$.

For (3), let $\alpha \in \{0,1\}^n$ such that $P_i(\alpha) \geqslant 0$ for every $P_i(x) \geqslant 0 \in \mathcal{P}$. The moment matrix corresponding to α is defined as $\mathcal{M}(\alpha)_{I,J} = \prod_{i \in I} \alpha_i \prod_{j \in J} \alpha_j$ for every $|I|, |J| \leqslant d$ with $I \cap J = \emptyset$, and $\mathcal{M}(\alpha, P_i)$ is defined analogously. Extend α to an $\binom{n}{\leqslant d}$-dimensional vector $\tilde{\alpha}$ by defining $\tilde{\alpha}_I = \prod_{i \in I} \alpha_I$. Then, for any $v \in \mathbb{R}^{\binom{n}{\leqslant d}}$,

$$v^{\top} \mathcal{M}(\tilde{\alpha}) v = v^{\top} \tilde{\alpha} \tilde{\alpha}^{\top} v = (v^{\top} \tilde{\alpha})^2 \geqslant 0,$$

and so $\mathcal{M}(\tilde{\alpha}) \succeq 0$. To see that $\mathcal{M}(\tilde{\alpha}, P_i) \succeq 0$, define the vector p where $p_I = P_i(\alpha) \prod_{j \in I} \alpha_j$ for $|I| \leqslant d - \deg(P_i)/2$, and observe that $pp^{\top} = \mathcal{M}(\tilde{\alpha}, P_i)$. Therefore $v^{\top} \mathcal{M}(\tilde{\alpha}, P_i) v = (v^{\top} p)^2 \geqslant 0$. □

Index

$\mathcal{LP}(\mathcal{P}, c)$: Linear program with constraint set $\mathcal{P}$ and objective function c 16

$\mathcal{LP}^D(\mathcal{P}, c)$: Dual linear program with constraint set $\mathcal{P}$ and objective function c 16

$\mathcal{ILP}(\mathcal{P}, c)$: Integer linear program with constraint set $\mathcal{P}$ and objective function c 18

$\mathsf{hull}_{\{0,1\}}(\mathcal{P})$: Integer hull of a set $\mathcal{P}$ 18

$\mathrm{conv}(S)$: Convex hull of a set S 18

$\mathcal{POP}(\mathcal{P}, P)$: Polynomial optimization problem with constraints $\mathcal{P}$ and objective $P(x)$ 18

$J_{S,T}(x)$: Non-negative junta 21

$\vec{P_i}$: Coefficient vector of polynomial $P_i(x)$ 22

$\deg(P_i)$: Degree of a polynomial $P_i(x)$ 22

$\mathsf{SA}_d(\mathcal{P})$: Degree d Sherali-Adams relaxation of a set of polynomials $\mathcal{P}$ 22

$\mathrm{proj}_{[n]}(\mathcal{P})$: Orthogonal projection of the set of points $\mathcal{P}$ to the variables x_i for $i \in [n]$ .. 24

$\tilde{\mathbb{E}}$: Pseudo-expectation 26

$\mathcal{E}_d(\mathcal{P})$: Set of degree d pseudo-expectations for $\mathcal{P}$ 26

$\mathcal{SDP}(\mathcal{S}, C)$: Semi-Definite program with constraint set S and objective function C ... 49

$\mathrm{size}(\cdot)$: Returns the bit complexity to specify the argument 51

$\mathsf{Ball}(r, c)$: Euclidean ball with radius r and center c 51

$\|\cdot\|_F$: Frobenius norm of a matrix 54

$\mathsf{Tr}[A]$: Matrix trace 59

$\mathcal{SDP}^D(\mathcal{S}^D, b)$: Dual semidefinite program with constraints $\mathcal{S}^D$ and objective function b 62

$\mathsf{SOS}_d(\mathcal{P})$: Degree d Sum-of-Squares relaxation 74

$\mathcal{M}_d(y)$: The degree d moment matrix for the SoS relaxation 74

Σ^2_{2d} : The set of all sum-of-squares polynomials of degree at most $2d$ 87

Σ^2 : The set of all sum-of-squares polynomials 87

$\Sigma^2_{2d}(\mathcal{P} \cup \{x_i^2 = x_i\})$: The convex cone of degree at most $2d$ polynomials generated from $\mathcal{P}$ and Σ^2_{2d} 87

$\mathcal{E}^{\mathbb{R}}_d(\mathcal{P})$: Set of degree d pseudo-expectations over $\mathbb{R}$ for $\mathcal{P}$ 98

$\mathsf{SOS}^{\mathbb{R}}_d(\mathcal{P})$: Degree d Sum-of-Squares relaxation over $\mathbb{R}$ 98

$\mathcal{N}(\mu, \Sigma)$: Gaussian distribution with mean μ and covariance Σ 111

$\|\cdot\|_2$: 2-norm 117

$\left|\frac{v}{k}\right.$: Degree k SoS derivation in variables v 129

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il