Sum of Squares Upper Bounds, Lower Bounds, and Open Questions

Boaz Barak

December 10, 2014

Preface

These are lecture notes for a seminar series I gave at MIT in the fall of 2014. The notes are very rough, undoubtedly containing plenty of bugs and tpyos, but I hope people would find them useful nonetheless. (One particular area in which the notes are rough is that in most cases I mention papers and work in the text without providing a formal references.) The notes were made before the lectures, and in some cases I deviated quite a bit in class from the prepared presentation. Thanks to all the students that participated in this series, and in particular to Adrian Vladu and Henry Yuen who scribed the notes of the second lecture, as well as to Akash Kumar, Samuel Hopkins, Jerry Li and Tal Wagner, who scribed notes from my summer 2014 course on this topic in the Swedish Summer School on Computer Science, which I used as a basis for these notes. Thanks to Jon Kelner and Ankur Moitra for giving guest lectures in this course.

Contents

Chapter 1

Introduction

Reading Sections 1 and 2 of my survey with Steurer "Sum of Squares proofs and the quest toward optimal algorithms"

Additional reading Introduction of Ryan O'Donnell and Yuan Zhou's paper "Approximability and Proof Complexity"

Lecture notes of Monique Laurent (`https://sites.google.com/site/mastermathsdp/lectures`) and Pablo Parrilo (`http://stellar.mit.edu/S/course/6/sp14/6.256/materials.html`)

Disclaimer I haven't tried to solve most of the exercises myself, and it could be that you'll run into various inaccuracies or issues while trying to solve them, let me know if you do. Also, all historical discussions and references are from memory or second-hand sources, and not based on the original texts, and so may be inaccurate.

Prelude

Consider the following questions:

1. Do we need a different algorithm to solve every computational problem, or can a single algorithm give the best performance for a large class of problems?

2. In statistical physics and other areas, many people believe in the existence of a *computational threshold* effect, where a small change in the parameters of a computational problems seems to lead to a huge change in its computational complexity. Can we give rigorous evidence for this intuition?

3. In machine learning there often seem to be tradeoffs between sample complexity, error probability, and computation time. Is there a way to map the curve of this tradeoff?

4. Suppose you are given a 3SAT formula φ with a unique (but unknown) satisfying assignment x. Is there a way to make sense of statements such as "The probability that $x_{17} = 1$ is 0.6" or "The entropy of x is 1000"? (even though of course information theoretically x is completely determined by φ, and hence that probability is either 0 or 1 and x has zero entropy).

5. Is Khot's Unique Games Conjecture true?

If you learn the answers to these questions by the end of this seminar series, then I hope you'll explain them to me, since I definitely don't know them. However we will see that, despite these questions a-priori having nothing to do with Sums of Squares, that the SOS algorithm can yield a powerful lens to shed light on some of those questions, and perhaps be a step towards providing some of their answers.

1.1 Introduction

Theoretical computer science studies many computational models for different goals. There are some models, such as bounded-depth (i.e. AC_0) circuits, that we can prove unconditional lower bounds on, but do not aim to capture all relevant algorithmic techniques for a given problem. (For example, we don't view the results of Furst-Sax-Sipser and Håstad as evidence that computing the parity of n bits is a hard problem.) Other models, such as bilinear circuits for matrix multiplication, are believed to be strong enough to capture all known algorithmic techniques for some problems, but then we often can't prove lower bounds on them.

The *Sum of Squares (SOS)* algorithm (discovered independently by researchers from different communities including Shor, Parrilo, Nesterov and Lasserre) can be thought of as another example of a concrete computational model. On one hand, it is sufficiently weak for us to know at least some unconditional lower bounds for it. In fact, there is a sense that it is weaker than AC_0, since for a given problem and input length, SOS is a *single algorithm* (as opposed to an exponential-sized family of circuits). Despite this fact, proving lower bounds for SOS is by no means trivial, even for a single distribution or instances (such as a random graph) or even a single instance. On the other hand, while this deserves more investigation, it does seem that for many interesting problems, SOS does encapsulate all the algorithmic techniques we are aware of, and that there is some hope that SOS is an *optimal algorithm* for some interesting family of problems, in the sense that no other algorithm with similar efficiency can beat SOS's performance on these problems.

The possibility of the existence of such an *optimal algorithm* is very exciting. Even if at the moment we can't hope to prove its optimality unconditionally, this means that we can (modulo some conjectures) reduce analyzing the difficulty of a problem to analyzing a single algorithm, and this has several important implications. For starters, it reduces the need for creativity in designing the algorithm, making it required only for the algorithm's *analysis*. In some sense, much of the progress in science can be described as attempting to automate and make routine and even boring what was once challenging. Just as we today view as commonplace calculations that past geniuses such as Euler and Gauss spent much of their time on, it is possible that in the future much of algorithm design, which now requires an amazing amount of creativity, would be systematized and made routine. Another application of optimality is automating *hardness results*— if we prove the optimal algorithm *can't* solve a problem X then that means that X can't be solved by any efficient algorithm.

But beyond just systematizing what we already can do, optimal algorithms could yield qualitative new insights on algorithms and complexity. For example, in many problems arising in statistical physics and machine learning, researchers believe that there exist *computational phase transitions*— where a small change in the parameter of a problems causes a huge jump in its computational complexity. Understanding this phase transitions is of great interest for both researchers in these areas and theoretical computer scientists. The problem is that these problems involve *random inputs* (i.e., *average case complexity*) and so, based on current state of art, we have no way of proving the existence of such phase transitions based on assumptions such as $\mathbf{P} \neq \mathbf{NP}$. In some cases, such as the planted clique problem, the problem has been so well studied that the existence of a computational phase transition had been proposed as a conjecture in its own right, but we don't know of good ways to reduce such reductions to one another, and we clearly don't want to have as many conjectures as there are problems. If we assume that an algorithm is optimal for a class of problems, then we can prove a computational phase transition by analyzing the running time of this algorithm as a function of the parameters. While by no means trivial, this is a tractable approach to understanding this question and getting very precise estimates as to the location of the threshold where the phase transition occurs. (Note that in some sense the existence of a computational phase transition implies the existence of an optimal algorithm, since in particular it means that there is a single algorithm A such that beating A's performance by a little bit requires an algorithm taking much more resources.)

Beyond all that, an optimal algorithm gives us a new understanding of just what is it about a problem that makes it easy or hard, and a new way to look at efficient computation. I don't find explanations such as "Problem A is easy because it has an algorithm" or "Problem B is hard because it has a reduction from SAT" very satisfying. I'd rather get an explanation such as "Problem A is easy because it has property P" and "Problem B is hard because it doesn't have P" where P is some meaningful property (e.g., being convex, supporting some polymorphisms, etc..) such that every problem (in some domain) with P is easy and every problem without it is hard. For that, we would want an algorithm that will solve all problems in P and a proof (or other evidence) that it is optimal. Such understanding of computation could bear other fruits as well. For example, as we will see in this seminar series, if the SOS algorithm is optimal in a certain domain, then we can use this to build a theory of "computational Bayesian reasoning" that can capture the "computational beliefs" of a bounded-time agent about a certain quantity, just as traditional Bayesian reasoning captures the beliefs of an unbounded-time agent about quantities on which it is given partial information.

I should note that while much of this course is very specific to the SOS algorithms, not all of it is, and it is possible that even if the SOS algorithm is superseded by another one, some of the ideas and tools we develop will still be useful. Also, note that I have deliberately ignored the question of *what* family of problems would the SOS be optimal for. This is clearly a crucial issue— every computational model (even AC_0) is optimal for *some* problems, and every model falling short of general polynomial-time Turing machines would not be optimal for *all* problems. It definitely seems that some algebraic problems, such as integer factoring, have very special structure that makes it hard to conjecture that any generic algorithm (and definitely not the SOS algorithm) would be optimal for them. (See also my blog post `http://wp.me/p2bJCi-Gp` on this topic.) The reason I don't discuss this issue is that we still don't have a good answer for it, and one of the research goals in this area is to understand what should be the right *conjecture* about optimality of SOS. However, we do have some partial evidence and intuition, including those arising from the SOS algorithm's complex (and not yet fully determined) relation to Khot's Unique Games Conjecture, that leads us to believe that SOS could be an optimal algorithm for a non-trivial and interesting class of problems.

In this course we will see:

1. A description of the SOS algorithm from different viewpoints— the traditional semidefinite programming/convex optimization view, as well as the proof system view, and the "pseudo-distribution" view.

2. Discussion of positive results (aka "upper bounds") using the SOS algorithms to solve graph problems such as sparsest cut, and problems in machine learning.

3. Discussion of known negative results (aka "lower bounds" / "integrality gaps") for this algorithm.

4. Discussion of the interesting (and not yet fully understood) relation of the SOS algorithm to Khot's *Unique Games Conjecture* (UGC). On one hand, implies that the SOS algorithm is *optimal* for a large class of problems. On the other hand, the SOS algorithm is currently the main candidate to refute the UGC.

1.2 Polynomial optimization

The SOS algorithm is an algorithm for solving a computational problem. Let us now define what this problem is:

Definition 1.1. A *polynomial equation* is an equation of the form $\{P(x) \geq 0\}$ (in which case it is called an *inequality*) or an equation of the form $\{P(x) = 0\}$ (in which case it is called an *equality*) where P is a multivariate polynomial mapping $x \in \mathbb{R}^n$ to $\mathbb{R}$. The equation $\{P(x) \geq 0\}$ (resp. $\{P(x) = 0\}$) is *satisfied* by $x \in \mathbb{R}^n$ if $P(x) \geq 0$ (resp. $P(x) = 0$).

A set $\mathcal{E}$ of polynomial equations is *satisfiable* if there exists an x that satisfies all equations in $\mathcal{E}$.

The *polynomial optimization* problem is to output, given a set $\mathcal{E}$ of polynomial equations as input, either an x satisfying all equations in $\mathcal{E}$ or a proof that $\mathcal{E}$ is unsatisfiable.

(Note: throughout this seminar we will ignore all issues of numerical accuracy— assume the polynomials always have rational coefficients with bounded numerator and denominator, and all equalities/inequalities can be satisfied up to some small error $\epsilon > 0$.)

Here are some examples for polynomial optimization problems:

Linear programming If all the polynomials are *linear* then this is of course linear programming that can be done in polynomial time.

Least squares If the equations consist of a single quadratic then this is the least squares algorithm. Similarly, one can capture computing eigenvalues by two quadratics.

3SAT Can encode 3SAT formula as degree 3 polynomial equations: the equation $x_i^2 = x_i$ is equivalent to $x_i \in \{0,1\}$. The equation $x_i x_j (1 - x_k) = 0$ is equivalent to $\overline{x_i \wedge x_j \wedge \overline{x_k}} = \overline{x_i} \vee \overline{x_j} \vee x_k$.

Clique Given a graph $G = (V, E)$ the following equations encode that x is a $0/1$ indicator vector of a k-clique: $x_i^2 = x_i$, $\sum x_i = k$, $x_i x_j = 0$ for all $(i,j) \notin E$.

Other examples Learning, etc...

The SOS algorithm is designed to solve the polynomial optimization problem. As we can see from these examples, the full polynomial optimization problem is NP hard, and hence we can't expect SOS (or any other algorithm) to efficiently solve it on every instance. (**Exercise 1.1:** prove that this is the case even if all polynomials are quadratic, i.e. of degree at most 2.) Understanding how close the SOS algorithm gets in particular cases is the main technical challenge we will be dealing with.

These examples also show that polynomial optimziation is an extremely versatile formalism, and many other computational problems (including SAT and CLIQUE) can be directly and easily phrased as instances of it. Henceforth we will ignore the question of *how* to formalize a problem as a polynomial optimization, and either assume the problem is already given in this form, or use the simplest most straightforward translation if it isn't. While there are examples where choosing between different natural formulations could make a difference in complexity, this is not the case (to my knowledge) in the questions we will look at.

Note: We can always assume without loss of generality that all our equations are *equalities*, since we can always replace the equation $P(x) \geq 0$ by $P(x) - y^2 = 0$ where y is some new auxiliary variable. Also, we sometimes will ask the question of minimizing (or maximizing) a polynomial $P(x)$ subject to x satisfying equations $\mathcal{E}$, which can be captured by looking for the largest μ such that $\mathcal{E} \cup \{P \geq \mu\}$ is satisfiable.

1.3 The SOS algorithm

The Sum of Squares algorithm is an algorithm to solve the polynomial optimization problem. Given that it is NP hard, the SOS algorithm cannot run in polynomial time on all instances. The main focus of this course is trying to understand in which cases the SOS algorithm takes a small (say polynomial or quasipolynomial) amount of time, in which cases it takes a large (say exponential) amount. An equivalent form of this question (which is the one we'll mostly use) is that, for some small ℓ (e.g. a constant or logarithmic) we want to understand in which cases the "n^ℓ-capped" version of SOS succeeds to solve the problem and in which cases it doesn't, where the "$T(n)$-capped" version of the SOS algorithm halts in time $T(n)$ regardless of whether or not it solved the problem.

In fact, we will see that for every value of d, the SOS of squares always returns some type of a meaningful output. The main technical challenge is to understand whether that output can be transformed to an exact or approximate solution for the polynomial optimization problem.

Definition 1.2 (Sum of Squares - informal definition)**.** The SOS algorithm gets a parameter ℓ and a set of equations $\mathcal{E}$, runs in time $n^{O(\ell)}$ and outputs either:

- An object we will call a "degree-ℓ pseudo solution" (or more accurately a degree-ℓ *pseudo-distribution* over solutions).

  or

- A proof that a solution doesn't exist.

We will later make this more precise: what is exactly a degree-ℓ pseudo solution, what is exactly the form of the proof, and how does the algorithm work.

History. The SOS algorithm has its roots in questions raised in the late 19*th* century by Minkowski and Hilbert of whether any non-negative polynomial can be represented as a sum of squares of other polynomials. Hilbert realized that except for some special cases (most notably univariate polynomials and quadratic polynomials), the answer is negative and that there is an example (which he constructed by non constructive means) of non-negative polynomial that cannot be represented in this way. It was only in the 1960's that Motzkin gave a very concrete example of such a polynomial

$$1 + x^4y^2 + x^2y^4 3x^2x^2 \tag{1.1}$$

In his famous 1900 address, Hilbert asked as his 17th problem whether any polynomial can be represented as a sum of squares of *rational* functions. (For example, Motzkin's polynomial (1.1) can be shown to be the sum of squares of (I think) four rational functions of denominator and

Figure 1.1: SOS was used to analyze the “falling leaf” mode of the U.S. Navy F/A-18 ”Hornet”, see A. Chakraborty, P. Seiler, and G. J. Balas, Journal of guidance, control, and dynamics, 34(1):7385, 2011

numerator degree at most 6). This was answer positively by Artin in 1927. His approach can be summarized as, given a hypothetical polynomial P that cannot be represented in this form, to use the fact that the rational functions are a field to extend the reals into a ”pseudo-real” field $\tilde{\mathbb{R}}$ on which there would actually be an element $\tilde{x} \in \tilde{R}$ such that $P(\tilde{x}) < 0$, and then use a ”transfer principle” to show that there is an actual real $x \in \mathbb{R}$ such that $P(x) < 0$. (This description is not meant to be understandable but to make you curious enough to look it up..) Later in the 60’s and 70’s Krivine and Stengle extended this result to show that any unsatisfiable system of polynomial equations can be certified to be unsatisfiable via a Sum of Squares proof, a result known as the Positivstallensatz.

In the late 90’s / early 2000’s, there were two separate efforts on getting quantitative or algorithmic versions of this result. On one hand Grigoriev and Vorobjov asked the question of *how large* the degree of an SOS proof needs to be, and in particular Grigoriev proved several lower bounds on this degree for some interesting polynomials. On the other hand Parrilo and Lasserre (independently) came up with hierarchies of algorithms for polynomial optimization based on the Positivstallensatz using semidefinite programming. (Something along those lines was also described by Naum Shor in a 1987 Russian paper, and mentioned by Nesterov as well.)

It took some time for people to realize the connection between all these works, and in particular the relation between Grigoriev-Vorbjov’s work and the works from the optimization literature took some time to be discovered, and even 10 years after, it was still the case that some results of Grigoriev were rediscovered and reproven in the Lasserre language.

Applications of SOS SOS has applications to: equilibrium analysis of dynamics and control (robotics, flight controls, ...), robust and stochastic optimization, statistics and machine learning, continuous games, software verification, filter design, quantum computation and information, automated theorem proving, packing problems, etc... (For two very different examples, see Figures 1.1, 1.2.)

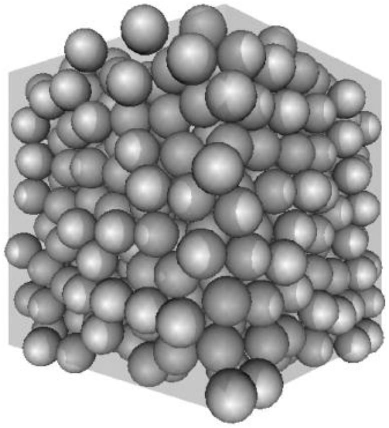


Figure 1.2: SOS was used to get the best known bounds on the classical "sphere packing" problem. See D. de Laat, F.M. de Oliveira Filho, F. Vallentin, Forum of Mathematics, Sigma, 2 (2014), e23

The TCS vs Mathematical Programming view of SOS

While the SOS algorithm is intensively studied in several communities, there are some differences in emphasizes between the different aspects. While I am not an expert on all SOS works, my impression that the main characteristics of the TCS viewpoint, as opposed to others are:

1. In the TCS world, we typically think of the number of variables n as large and tending to infinity (as it corresponds to our input size), and the degree d of the SOS algorithm as being relatively small— a constant or logarithmic. In contrast, in the optimization and control world, the number of variables can often be very small (e.g. around ten or so, maybe even smaller) and hance d may be large compared to it.

   Note that since both time and space complexity of the general SOS algorithm scale roughly like n^d, even $d = 6$ and $n = 100$ would take something like a petabyte of memory (in practice, though we didn't try to optimize too much, David Steurer and I had a hard time executing a program with $n = 16$ and $d = 4$ on a Cornell cluster). This may justify the optimization/control view of keeping n small, although if we show that SOS yields a polynomial-time algorithm for a particular problem, then we can hope that we would be able to then optimize further and obtain an algorithm that doesn't require a full-fledged SOS solver.

2. Typically in TCS our inputs are discrete and the polynomials are simple, with integer coefficients etc. Often we have constraints such as $x_i^2 = x_i$ that restrict attention to the Boolean cube, and so we are less concerned with issues of numerical accuracy, boundedness, etc..

3. Traditionally people have been concerned with *exact convergence* of the SOS algorithm—- when does it yield an exact solution to the optimization problem. This often precludes d from being much smaller than n. In contrast as TCS'ers we would often want to understand *approximate convergence*— when does the algorithm yield an "approximate" solution (in some problem-dependent sense).

   Since the output of the algorithm in this case is not actually in the form of a solution to the equations, this raises the question of a obtaining *rounding* algorithms, which are procedures to translate the output of the algorithm to an approximate solution.

1.4 Several views of the SOS algorithm

We now describe the SOS algorithm more formally. For simplicity, we consider the case that the set $\mathcal{E}$ only consists of equalities (which is without loss of generality as we mentioned before). When convenient we will assume all equalities are homogenous polynomials of degree d. (This can be always be arranged by multiplying the constraints.) You can restrict attention to $d = 4$— this will capture all of the main issues of the general case.

1.4.1 SOS Algorithm: convex optimization view

We start by presenting one view of the SOS algorithm, which technically might be the simplest, though perhaps at first not conceptually insightful.

Definition 1.3. Let $\mathbb{R}_d^n$ denote the set of n-variate polynomials of degree at most d. Note that this is a linear subspace of dimension roughly n^d.

We will sometimes also write this as $\mathbb{R}[x]_d$ where we want to emphasize that these polynomials take the formal input $x = x_1 \ldots x_n$.

Definition 1.4. Let $\mathcal{E} = \{p_1 = \cdots p_m = 0\}$ be a set of polynomial equations where $p_i \in \mathbb{R}_d^n$ for all i. Let $\ell \in \mathbb{N}$ be some integer multiple of $2d$. The *degree-ℓ SOS algorithm* either outputs `'fail'` or a bilinear operator $M : \mathbb{R}_{\ell/2}^n \times \mathbb{R}_{\ell/2}^n \to \mathbb{R}$ satisfying:

- Normalization: $M(1, 1) = 1$ (where 1 is simply the polynomial $p(x) = 1$).
- Symmetry: If $p, q, r, s \in \mathbb{R}_{\ell/2}^n$ satisfy $pq = rs$ then $M(p, q) = M(r, s)$.
- Non-nonnegativity (positive semi definiteness): For every p, $M(p, p) \geq 0$.
- Feasibility: For every $i \in [m]$, $p \in \mathbb{R}_{\ell/2-d}^n$, $q \in \mathbb{R}_{\ell/2}^n$, $M(p_i p, q) = 0$.

Exercise 1.2: Show that if the symmetry and feasibility constraints hold for monomials they hold for all polynomials as well.

Exercise 1.3: Show that the set of M's satisfying the conditions above is convex and has an efficient separation oracle.

Indeed, such an M can be represented as an $n^{\ell/2} \times n^{\ell/2}$ PSD matrix satisfying some linear constraints. (Can you see why?) Thus by semidefinite programming finding such an M if it exists can be done in $n^{O(\ell)}$ time (throughout this seminar we ignore issues of precision etc..). The question is why does this have anything to do with solving our equations, and one answer is given by the following lemma:

Lemma 1.5. *Suppose that $\mathcal{E}$ is satisfiable. Then there exists an operator M satisfying the conditions above.*

Proof. Let x^0 be a solution for the equations and let $M(p, q) = p(x^0)q(x^0)$. Note that M clearly satisfies all the conditions. □

Since the set of such operators M is convex, for every distribution μ over solutions of $\mathcal{E}$, the operator $M(p, q) = \mathbb{E}_{x\sim\mu}\, p(x)q(x)$ also satisfies the conditions. As ℓ grows, eventually the only operators that satisfy the condition will be of this form.

For this reason we will call M a *degree-ℓ pseudo-expectation operator.* For a polynomial p of degree at most ℓ, we define $M(p)$ as follows: we write $p = \sum \alpha_i p_i$ where each p_i is a *monomial* of degree at most ℓ, and then decompose $p_i = p'_i p''_i$ where the degree of p'_i and p''_i is at most $\ell/2$ and then define $M(p) = \sum \alpha_i M(p'_i, p''_i)$. We will often use the suggestive notation $\tilde{\mathbb{E}}\, p$ for $M(p)$.

Exercise 1.4: Show that $M(p)$ is well defined and does not depend on the decomposition.

1.4.2 Intuition— the Boolean cube

To get some intuition, we now focus attention about the special case that our goal is to maximize some polynomial $p_0(x)$ over over Boolean cube $\{\pm 1\}^n$ (i.e., the set of x's satisfying $x_i^2 = 1$.) This case is not so special in the sense that (a) it captures much of what we want to do in TCS and (b) the intuition it yields largely applies to more general settings.

Recall that we said that for every distribution μ over x's satisfying the constraints, we can get an operator M as above by looking at $\mathbb{E}_{x \sim \mu}\, p(x)q(x)$. We now show that in some sense *every* operator has this form, if, in a manner related to and very reminiscent of quantum information theory, we allow the probabilities to go negative.

Definition 1.6. A function $\mu : \{\pm 1\}^n \to \mathbb{R}$ is a *degree-ℓ pseudo-distribution* if it satisfies:

- Normalization: $\sum_{x \in \{\pm 1\}^n} \mu(x) = 1$.

- Restricted non-negativity: For every polynomial p of degree at most $\ell/2$, $\tilde{\mathbb{E}}_{x \sim \mu}\, p(x)^2 \geq 0$, where we define $\tilde{\mathbb{E}}_{x \sim \mu}\, f(x)$ as $\sum_{x \in \{\pm 1\}^n} \mu(x) f(x)$.

Note that if μ was actually pointwise non-negative then it would be an actual distribution on the cube. Thus an actual distribution over the cube is always a pseudo distribution.

Exercise 1.5: Show that a degree $2n$ pseudo-distribution is an actual distribution.

Exercise 1.6: Show that if μ is a degree ℓ pseudo-distribution, then there exists a degree-ℓ pseudo-distribution μ' such that $\tilde{\mathbb{E}}_{x \sim \mu}\, p(x) = \tilde{\mathbb{E}}_{x \sim \mu'}\, p(x)$ for every polynomial p and that $\mu'(x)$ is a degree ℓ polynomial in the variables of x. (Hence for our purposes we can always represent such pseudo-distributions with $n^{O(\ell)}$ numbers.)

Exercise 1.7: Show that for every polynomial p_0 of degree at most $\ell/2$, there exists a degree ℓ pseudo-distribution μ on the cube satisfying $\tilde{\mathbb{E}}_{x \sim \mu}\, p_0(x) \geq \lambda$ if and only if there exists a degree ℓ pseudo-expectation operator M as above satisfying $\{x_i^2 = 1 : i = 1..n\}$ such that $M(p_0) \geq \lambda$.

Therefore, we can say that the degree-ℓ SOS algorithm outputs either a degree-ℓ pseudo-distribution over the solutions to $\mathcal{E}$ or `'fail'` and only outputs the latter if the former doesn't exist. In particular if it outputs `'fail'` then there isn't any *actual* distribution over the solutions, and so the fact that the algorithm outputs `'fail'` is a *proof* that the original equations are unsatisfiable. We will see that by convex duality, the algorithm actually outputs an explicit proof of this fact that has a natural interpretation.

Exercise 1.8: (optional— for people who have heard about the Sherali-Adams linear programming hierarchy) Show that the variant of pseudo-distributions where we replace the condition that expectation is non-negative on all squares of degree $\ell/2$ polynomials with the condition that it should be non-negative on all non-negative functions that depend on at most ℓ variables can be optimized over using linear programming and is equivalent to ℓ rounds of the Sherali-Adams LP.

Are all pseudo-distributions distributions? For starters, we can always find a distribution matching all the quadratic moments.

Lemma 1.7 (Gaussian Sampling Lemma)**.** *Let M be a degree-ℓ pseudo-expectation operator for $\ell \geq 2$. Then there exists a distribution $(y_1, \ldots, y_n)$ over $\mathbb{R}^n$ such that for every polynomial p of degree at most 2, $M(p) = \mathbb{E}\, p(y)$. Moreover, y is a (correlated) Gaussian distribution.*

Note that even if M comes from a pseudo-distribution μ over the cube, the output of y will be real numbers that although satisfying $\mathbb{E}\, y_i^2 = 1$, will be in $\{\pm 1\}$.

Unfortunately, we don't have an analogous result for higher moments:

Exercise 1.9: Prove that if there was an analog of the Gaussian Sampling Lemma for every polynomial p of degree at most 6 then P=NP. (Hint: show that you could solve 3SAT, can you improve the degree to 4? maybe 3?)

Unfortunately, this will not be our way to get fame and fortune:

Exercise 1.10: Prove that there exists a degree 4 pseudo-distribution μ over the cube such that there does not exist any actual distribution ν that matches its expectation on all polynomials of degree at most 4. (Can you improve this to 3?)

1.5 Sum of Square Proofs

As we said, when the SOS algorithm outputs `'fail'` this can be interpreted as a proof that the system of equations is unsatisfiable. However, it turns out this proof actually has a special form that is known as an SOS proof or *positivstenelsatz*. An SOS proof uses the following rules of inference

$$\begin{aligned} p \geq 0, q \geq 0 &\models p + q \geq 0 \\ p \geq 0, q \geq 0 &\models pq \geq 0 \\ &\models p^2 \geq 0 \end{aligned}$$

They should be interpreted as follows. If you know that a set of conditions $\mathcal{E} = \{p_1 \geq 0, \ldots, p_m \geq 0\}$ is satisfied on some set S, then any conditions derived by the rules above would on that set as well. (Note that we only mentioned inequalities above, but of course $\{p = 0\}$ is equivalent to the conditions $\{p \geq 0, -p \geq 0\}$.)

Definition 1.8. Let $\mathcal{E}$ be a set of equations. We say that $\mathcal{E}$ implies $p \geq 0$ via a degree-ℓ SOS proof, denoted $\mathcal{E} \models_\ell p \geq 0$, if $p \geq 0$ can be inferred from the constraints in $\mathcal{E}$ via a sequence of applications of the rules above where all intermediate polynomials are of syntactic degree $\leq \ell$.

The *syntactic degree* of the polynomials in $\mathcal{E}$ is their degree, while the syntactic degree of $p+q$ (resp. pq) is equal to the maximum (resp. the sum) of the syntactic degrees of p, q. That is, the syntactic degree tracks the degrees of the intermediate polynomials without accounting for cancellations.

(Note: If we kept track of the actual degree instead of the syntactic degree we get a much stronger proof system for which we don't have a static equivalent form, and can prove some things that the static system cannot. See the paper of Grigoriev, Hirsch and Pasechnik `http://eccc.hpi-web.de/report/2001/103/` for discussion of this other system.**)**

Definition 1.9. Let $\mathcal{E}$ be a set $\{p_1 = \cdots = p_m = 0\}$ of polynomial equalities. We say that $\mathcal{E}$ has a *degree-ℓ SOS refutation* if $\mathcal{E} \models_\ell 0 \geq 1$.

It turns out that a degree-ℓ refutation can always be put in a particular compact *static* form.

Exercise 1.11: For every $d < \ell$, prove that $\mathcal{E} = \{p_1 = \cdots = p_m = 0\}$ (where all p_i's are of degree d) has a *degree-ℓ SOS refutation* if and only if there exists $q_1, \ldots, q_m$ of degree at most $\ell' = O(\ell)$ and $r_1, \ldots, r_{m'}$ of degree at most $\ell'/2$ such that

$$\sum q_i p_i = 1 + s \tag{1.2}$$

where $s = \sum_{i=1}^{m'} r_i^2$, i.e. it is a *sum of squares*. (It's OK if you lose a bit in each direction, i.e., in the if direction it could be that $\ell' = 2\ell$ while in the only if direction it could be that $\ell' = \ell/2$.)

Exercise 1.12: Show that we can take m' to be at most $n^{2\ell}$.

Exercise 1.13: Show that the set $(p_1, \ldots, p_m, s)$ satisfying (1.2) is a convex set with an efficient separation oracle.

Positivstellensatz (Stengle 64, Krivine 74) For every unsatisfiable system $\mathcal{E}$ of equalities there exists a finite ℓ s.t. $\mathcal{E}$ has a degree ℓ proof of unsatisfiability. **Exercise 1.14:** Prove P-satz for systems that include the constraint $x_i^2 = x_i$ for all i. In this case, show that ℓ needs to be at most $2n$ (where n is the number of variables). As a corollary, we get that the SOS algorithm does not need more than $n^{O(n)}$ time to solve polynomial equations on n Boolean variables. (Not very impressive bound, but good to know. In all TCS applications I am aware of, it's easy to show that the SOS algorithm will solve the problem in exponential time.)

Exercise 1.15: Show that if there exists a degree-ℓ SOS proof that $\mathcal{E}$ is unsatisfiable then there is no degree-ℓ pseudo-distribution consistent with $\mathcal{E}$.

SOS Theorem (Shor, Nesterov, Parrilo, Lasserre) Under some mild conditions (see Theorem 2.7 in survey), there is an $n^{O(\ell)}$ time algorithm that given a set $\mathcal{E}$ of polynomial equalities either outputs:

- A degree-ℓ pseudo-distribution μ consistent with $\mathcal{E}$

  or

- A degree-ℓ SOS proof that $\mathcal{E}$ is unsatisfiable.

1.6 Discussion

The different views of pseudo distributions The notion of pseudo-distribution is somewhat counter-intuitive and takes a bit of time to get used to. It can be viewed from the following perspectives:

- Pseudo-distributions is simply a fancy name for a PSD matrix satisfying some linear constraints, which is the dual object to SOS proofs.

- SOS proofs of unbounded degree is a sound and complete proof system in the sense that they can prove any true fact (phrased as polynomial equations) about actual distributions over $\mathbb{R}^n$.

  SOS proofs of degree d is a sound and not complete proof system for actual distributions, but it is a (sound and) complete system for degree d pseudo-distributions, in the sense that any true fact that holds not merely for actual distributions but also for degree d pseudo-distributions has a degree d SOS proof.

- In statistical learning problems (and economics) we often capture our knowledge (or lack thereof) by a distribution. If an unknown quantity X is selected and we are given the observations y about it, we often describe our knowledge of by a the distribution $X|y$. In computational problems, often the observations y completely determine the value X, but pseudo-distribution can still capture our "computational knowledge".

- The proof system view can also be considered as a way to capture our limited computational abilities. In the example above, a computationally unbounded observer can deduce from the observations y all the true facts it implies and hence completely determine X. One way to capture the limits of a computationally bounded observer is that it can only deduce facts using a more limited, sound but not complete, proof system.

Lessons from History It took about 80 years from the time Hilbert showed that polynomials that are not SOS exist non-constructively until Motzkin came up with an explicit example, and even that example has a low degree SOS proof of positivity. One lesson from that is the following:
"Theorem": If a polynomial P is non-negative and "natural" (i.e., constructed by methods known to Hilbert— not including probabilistic method), then there should be a low degree SOS proof for this fact.
Corollary (Marley, 1980): If you analyze the performance of an SOS based algorithm pretending pseudo-distributions are actual distributions, then unless you used Chernoff+union bound type arguments, then every little thing gonna be alright.

We will use Marley's corollary extensively in analyzing SOS algorithms. There is a recurring theme in mathematics of "power from weakness". For example, we can often derandomize certain algorithms by observing that they fall in some restricted complexity classes and hence can be fooled by certain pseudorandom generator. Another example, perhaps closer to ours, is that even though the original way people defined calculus with "infitesimal" amounts were based on false permises, still much of the results they deduced were correct. One way to explain this is that they used a weak proof system that cannot prove all true facts about the real numbers, and in particular cannot detect if the real numbers are replaced with an object that does have such an "infitesimal" quantity added to it. In a similar way, if you analyze an algorithm using a weak proof system (e.g. one that is captured by a small degree SOS proof), then the analysis will still hold even if we replaced actual distributions with a pseudo-distribution of sufficiently large degree.

Chapter 2

Max-Cut, Sparsest Cut, Small Set Expansion and some relations of Isoperimetry and Hypercontractivity

Lecture notes by Adrian Vladu and Henry Yuen

Suggested reading

- As I mentioned in the email, Spielman's Spectral Graph Theory lectures 1,2 and 6 [1] are good reading for the background to this lecture. Spielman's disclaimer (and in particular the warning that you should "Be skeptical of all statements in these notes that can be made mathematically rigorous") also applies to the lecture notes in this course.
- See Section 6.2 (pages 143-147 in electronic version) of [WS11] for an overview of the Goemans-Williamson Max-Cut algorithm.
- The Cheeger-Alon-Milman Inequality is covered in many places. One good source is Trevisan's CS359G Lecture Notes [2].
- The Feige-Schechtman graph is from their paper [FS02]. It is also described in several lecture notes. One good source is the lecture of Ryan O'Donnell from the CMU "Advanced Approximation Algorithms" course [3].
- The result that degree 4 SOS (or more accurately degree 2 + the squared triangle inequalities) solve max-cut on random geometric (i.e. Feige-Schechtman) graphs is from [BHHS11].
- Chapter 9 ("basic hypercontractivity") in Ryan O'Donnell's book is highly recommended reading. In particular what we show here is that what he calls "The Bonami Lemma" has a degree 4 SOS proof. This result, as well as other applications, and an equivalence between small set expansion and hypercontractive norm bounds is from [BBH$^+$12a]. (The direction of the equivalence we described was known before, and in particular appears in O'Donnell's book, but you may be interested in looking at the proof of the other direction.)

[1] See http://www.cs.yale.edu/homes/spielman/561/

[2] See http://theory.stanford.edu/~trevisan/cs359g/, Lectures 3 and 4

[3] Available on http://www.cs.cmu.edu/~anupamg/adv-approx/lecture16.pdf.

Both papers mentioned above are available on my home page.

Recap and some musings

The Sum of Squares algorithm is parameterized by a number ℓ, known as its *degree*, and its running time is $n^{O(\ell)}$. When $\ell = 1$ this corresponds to *linear programming*, when $\ell = 2$, it corresponds to *semi-definite programming*, and when $\ell = n$ it corresponds to the *brute force/exhaustive search* algorithm.

In this course, we are most interested in the range $2 < \ell \ll n$. To borrow an analogy from Avi Wigderson, this regime is a bit like the "dark matter" of the SOS algorithm. We know it exists, but we have surprisingly few examples of problems that cannot be solved by the degree 2 case, and can provably be solved by SOS of non-trivially small degree. This is related to the well known phenomenon that, while we know by Ladner's theorem that there are infinitely many "NP intermediate"' problems, most natural computational problems are either in **P** or **NP**-hard. In fact, most natural problems either have a low-exponent polynomial-time algorithm (e.g., n^2 or n^3) or are exponentially hard (e.g., no $\exp(n^{0.99})$ algorithm is known). There are of course problems, such as k-SUM or k-CLIQUE, where the natural exhaustive search algorithm takes time $n^{O(k)}$ where k is some parameter of the problem; I would consider those as "exponential" in the sense that the best algorithm is still exhaustive search even if it runs in polynomial time.

One could play devil's advocate and suggest that maybe the only problems in this "dark matter" are artificial problems such as those constructed by Ladner, and so perhaps studying SOS for degree larger than 2 is a waste of time. In this course, starting with this lecture, we will see several of the few known examples where degree > 2 proofs help. I will leave to your judgment how natural they are, and, most importantly, I hope you manage to find some new ones!

(I should remark that, even if we restrict attention to domains where SOS is optimal, the "dark matter" region I am describing here is not identical to the set of problems that are not in **P** but not **NP** complete. There are very few examples, perhaps the most notable ones arising from Lattice-based cryptography, of natural problems in **NP** that are believed to be exponentially hard but not **NP**-complete.)

Here are some exercises to make sure that you are comfortable with pseudo-distributions:

Exercise 2.1: Prove the pseudo-distribution Cauchy-Schwarz condition: If μ is a pseudo-distribution of degree at most $2d$, and P, Q are polynomials of degree d then

$$\tilde{\mathbb{E}}_\mu PQ \le \sqrt{\tilde{\mathbb{E}}_\mu P^2}\sqrt{\tilde{\mathbb{E}}_\mu Q^2}$$

Exercise 2.2: Recall that we say that a degree-d pseudo-distribution μ satisfies the constraint $\{p = 0\}$ if $\tilde{\mathbb{E}}_\mu pq = 0$ for every q for which this expectation makes sense (i.e. of degree at most $d - \deg p$). Give an example of a pseudo-distribution μ that satisfies $\tilde{\mathbb{E}}\, p(x) = 0$, but does not satisfy the constraint $\{p = 0\}$. How high can you make the degree of μ?

Exercise 2.3: Show that if $\tilde{\mathbb{E}}_\mu p^2 = 0$ then $\tilde{\mathbb{E}}_\mu pq = 0$ for every q of degree at most $d/2 - \deg P$.

Exercise 2.4: (Hölder's inequality) Prove that if μ is a degree 4 distribution over variables $u_1, \ldots, u_n, w_1, \ldots, w_n \in \mathbb{R}^n$ then $\tilde{\mathbb{E}}_\mu \sum_i u(i)^3 w(i) \le \left(\tilde{\mathbb{E}}_\mu \|u\|_4^4\right)^{3/4} \left(\|w\|_4^4\right)^{1/4}$.

A tale of two problems

Let $G = (V, E)$ be a d-regular graph on n vertices with normalized adjacency matrix A ($A_{i,j} = 1/d$ if $(i, j) \in E$ and $A_{i,j} = 0$ otherwise). Let $L = I - A$ be its normalized *Laplacian matrix.* For a set $S \subseteq V$, we consider the following quantities:

1. The *expansion* or *conductance* of S, denoted by $\phi(S)$, is defined as

   $$\phi(S) = \frac{E(S, \overline{S})}{d \cdot \min\{|S|, n - |S|\}}$$

   Note that this number is always in $[0, 1]$; sometimes this number is defined as

   $$\phi(S) = \frac{nE(S, \overline{S})}{d|S||\overline{S}|}$$

   (one can see that this is equivalent up to a factor of 2). The conductance of the graph G is defined as $\phi(G) = \min_S \phi(S)$.

2. The (fractional) *cut size* of S is the number

   $$\mathsf{cut}(S) = \frac{E(S, \overline{S})}{|E|}$$

   Note that for $S = \Theta(n)$, $\mathsf{cut}(S) = \Theta(\phi(S))$. The *maximum cut value* of G, denoted by $\mathsf{maxcut}(G)$ is $\max_S \mathsf{cut}(S)$.

3. The (uniform) *sparsest cut* problem is the task of computing $\phi(G)$ (or possibly also finding the set S such that $\phi(G) = \phi(S)$ — we will not distinguish between these two problems). The *max-cut* problem is the task of computing $\mathsf{maxcut}(G)$ (or finding the set S such that $\mathsf{cut}(S) = \mathsf{maxcut}(G)$).

Known results. We now survey what is known about those two problems, which turn out to be extremely similar in their computational status. In both cases, finding the exact solution is NP hard, and so we are looking for some form of approximation.

A random subset of measure $1/2$ will cut half the edges, and in particular this gives an algorithm achieving a cut of value at least $\mathsf{cut}(G)/2$ for the max-cut problem. In fact, this algorithm for max-cut was suggested by Erdös in 1967, and is one of the first analyses of any approximation algorithm.

A priori, it is not so clear how to beat this. Let us consider the case of max-cut. In a random d-regular graph (which is an excellent expander), one cannot cut more than a $1/2 + \epsilon$ fraction of the edges (where ϵ goes to zero as n goes to infinity). But locally, it is hard to distinguish a random d-regular graph from a random d-regular "almost bipartite" graph, where we split the graph into two equal parts L and R and each edge is with probability ϵ inside one of those parts and with probability $1 - \epsilon$ between them. Such a graph G obviously has $\mathsf{maxcut}(G) \geq 1 - \epsilon$ but every neighborhood of it looks like a d-regular tree, just as in the case of a random d-regular graph. For this reason, "combinatorial" (or even linear programming) algorithms have a hard time getting an approximation factor better than $1/2$ for max-cut. To support this claim, [CLRS13] shows that no polynomial-time extension of the max-cut linear program can beat the $1/2$ factor. For a similar reason, a priori it is not clear how to find any set with $\phi(S) \ll 1/2$, even if $\phi(G) = o(1)$. However, in both those cases it turns out one can beat the "combinatorial" (or linear programming) algorithms.

The famous *Cheeger's Inequality* (or more accurately, its discrete variant by Alon, Alon-Milman, Dodziuk) implies that there is a polynomial time algorithm to find S with $\phi(S) = O(\sqrt{\phi(G)})$. Cheeger's Inequality can be viewed as the degree-2 SOS algorithm. The analogous algorithm for max-cut took more time, but was found eventually by Goemans and Williamson [GW95] who gave an algorithm, based on rounding the degree-2 SOS algorithm; on an input graph G with $\mathsf{maxcut}(G) \geq 1-\epsilon$, it find a set S with $\mathsf{cut}(S) \geq 1-f(\epsilon)$ for some $f(\epsilon) = O(\sqrt{\epsilon})$. This algorithm is often described in terms of its approximation ratio $\mathsf{cut}(S)/\mathsf{maxcut}(G)$, which is $\min_{\epsilon>0} \frac{1-f(\epsilon)}{1-\epsilon} \approx 0.878$. Leighton and Rao gave a polynomial-time algorithm to find S with $\phi(S) = O(\log n)\phi(G)$ [LR99] and in a breakthrough work, Arora, Rao and Vazirani improved this to an algorithm that outputs a set S with $O(\sqrt{\log n})\phi(G)$ [ARV09]. Their algorithm uses the degree 4 SOS algorithm, and we will see it in this course. Shortly thereafter, Agarwal, Charikar, Makarychev and Makarychev gave the analogous result for max-cut, namely an algorithm that given G with $\mathsf{maxcut}(G) = 1-\epsilon$, outputs S with $\mathsf{cut}(S) \geq 1 - O(\sqrt{\log n})\epsilon$ [ACMM05].

Assuming the *Unique Games Conjecture* (or its close variant the *Small-Set Expansion Conjecture*), the algorithms of Cheeger and Goemans-Williamson are optimal. Namely, there is no poly-time algorithm to find S with $\phi(S) = o(\sqrt{\phi(G)})$, and no poly-time algorithm that given G with $\mathsf{maxcut}(G) = 1-\epsilon$, finds a set S with $\mathsf{cut}(S) = 1 - o(\sqrt{\epsilon})$. One way to think of the Unique Games Conjecture is that it is saying that *degree* 2 *SOS is special* in the sense that for great many problems, improving on it requires to going to degree $n^{\Omega(1)}$. (A priori you could perhaps think that degree 2 is very special, in the sense that improving on it would require degree $\Omega(n)$, but that has been refuted by the sub-exponential algorithm for unique games [ABS10] and its encapsulation within the SOS framework [BRS11, GS11]. Degree 2 SOS *is* special in the sense that it seems much easier to for us to analyze, but whether degree 4 offers no improvements, or is just more challenging for us to prove that it does, remains to be seen. We will see today and in the next lectures some examples where higher (but not super high) degree does help, but as of now these still fall short of disproving the UGC.

Remark: Isoperimetry, local testing, and extremal questions

The max-cut and sparsest cut problems are all special case of the task of determining isoperimetric properties of graphs.

The classical isoperimteric inequality states that the circle is the body with the most area for a given perimeter. More generally, the isoperimetric question in a particular space, is to determine the set in that space with the smallest surface area given some prescribed volume. (With the circle being the answer in two dimensional Euclidean space, and spheres in higher dimensions.) Such questions have a great many applications in a variety of areas, including geometry, graph theory, probability theory and mathematical physics. In particular isoperimetric inequalities are often used to analyze the convergence times of random walks. For a graph, $\phi(S)$ is a natural proxy for the surface area of a set S, and so one variant of the isoperimetric question is to find, given some number $\delta > 0$, the set S of size δn that minimizes $\phi(S)$. This is known as the "small set expansion" question and is intimately related to the unique games conjecture. We are often able to prove isoperimetric inequalities for particular families of graphs, and understanding whether or not these proofs can be "SOS'ed" with low degree is key to understanding the power of this algorithm.

Isoperimetric inequalities are a special case of a more general paradigm of extremal questions in mathematics. Such questions again arise in great many areas, and in particular not just in geometry but also in coding theory and additive combinatorics. In many cases we have some collection of objects Ω (e.g. all subsets of some space of a certain size, or maybe all strings of some length, or all subsets of some group) and some parameter or "test" $T : \Omega \to \mathbb{R}$ (e.g., $T(S)$ can be the surface area of some set S, or $T(S)$ might measure the probability that a certain local test fails on S, or the size of the set $S+S$ where $+$ is the group operation). The generalization of an isoperimetric inequality would be to show that there is some set $\mathcal{C}$ of "special" objects (e.g., the spheres, or codewords of some code, or subgroups) that minimize $T(\cdot)$. The next natural question is the "unique decoding" question, showing that if $T(S)$ is close to the minimum, then S itself is close to some member of $\mathcal{C}$. Another natural question is the "list decoding" question, showing that if $T(S)$ is much smaller than the average value of $T(\cdot)$, then S is at least somewhat correlated with an element of $\mathcal{C}$.

Linear algebra view

Let x be the $\{\pm 1\}$ characteristic vector of S (i.e., $x_i = +1$ if $i \in S$, and $x_i = -1$ otherwise).

Then

$$\langle x, Lx\rangle = \langle x, \left(I - \tfrac{1}{d}A\right) x\rangle = \sum_i x_i^2 - \tfrac{1}{2d}\sum_{i\sim j} x_i x_j = \tfrac{1}{2d}\sum_{i\sim j}(x_i - x_j)^2 = 4E(S,\overline{S})/d = 2n \cdot \mathsf{cut}(S)$$

Observe that each edge is counted twice in the sum over $i \sim j$, and contributes 4 if it belongs to the cut.

Similarly, if x is the $\{0,1\}$ characteristic vector instead (i.e., $x_i = 1$ if $i \in S$ and $x_i = 0$ otherwise) and $|S| \le n/2$ then

$$\langle x, Lx\rangle = \tfrac{1}{d}E(S,\overline{S}) = \phi(S)\|x\|_2^2$$

(since $\|x\|_2^2 = |S|$).

Exercise 2.5: Show that if x is the mean 0 characteristic vector of S (i.e., $x_i = n - |S|$ if $i \in S$ and $x_i = -|S|$ if $i \notin S$) then $\langle x, Lx\rangle/\|x\|^2 = \frac{nE(S,\overline{S})}{d|S||\overline{S}|}$. Show that this also equals $\phi(S)$ up to a multiplicative factor of two.

2.1 The Goemans Williamson Algorithm

The Goemans-Williamson algorithm takes as input a graph G with $\mathsf{maxcut}(G) \geq 1-\epsilon$ and outputs a set S such that $\mathsf{cut}(S) \geq 1 - O(\sqrt{\epsilon})$. It follows from the theorem below:

Theorem 2.1 (Goemans-Williamson)**.** *There is a polynomial-time algorithm R which, given*

- *an n-vertex d-regular graph $G = (V, E)$*
- *a degree 2 pseudo-distribution μ, such that $\tilde{\mathbb{E}}_{x\sim\mu} x_i^2 = 1$ and $\tilde{\mathbb{E}}\langle x, Lx\rangle \geq 2n(1-\epsilon)$*

outputs a vector $z \in \{\pm 1\}^n$, such that $\langle z, Lz\rangle \geq 2n(1 - f_{GW}(\epsilon))$ where $f_{GW}(\epsilon) \leq 10\sqrt{\epsilon}$.

This immediately implies the algorithm, because given a graph with $\mathsf{maxcut}(G) \geq 1-\epsilon$, running the degree-2 SOS algorithm on $\{x_i^2 = 1\ \forall i, \langle x, Lx\rangle = 2n(1-\epsilon)\}$ yields a valid pseudo-distribution satisfying the requirements for Theorem 2.1. Running algorithm R, then gives an integral cut of value at least $1 - f_{GW}(\epsilon) = 1 - O(\sqrt{\epsilon})$.

At the heart of the proof of the Goemans-Williamson algorithm is the following very useful lemma:

Lemma 2.2 (Quadratic Sampling Lemma)**.** *Let μ be a degree-2 pseudo-distribution over $\mathbb{R}^n$. Then there is a poly-time that algorithm can sample from a Gaussian distribution y over $\mathbb{R}^n$ such that $\mathbb{E}\, p(y) = \tilde{\mathbb{E}}_{x\sim\mu}\, p(x)$ for every polynomial p of degree at most 2.*

Note on notation: In the rest of this course, in cases where there is little chance of confusion, we will often denote a pseudo-distribution as $\{x\}$ rather than μ and then use notation such as $\tilde{\mathbb{E}}\, p(x)$ instead of $\tilde{\mathbb{E}}_{x\sim\mu}\, p(x)$. So, we could also write this lemma as:

Lemma 2 (Quadratic Sampling Lemma)**.** Let $\{x\}$ be a degree-2 pseudo-distribution over $\mathbb{R}^n$. Then there is a poly-time that algorithm can sample from a Gaussian distribution y over $\mathbb{R}^n$ such that $\mathbb{E}\, p(y) = \tilde{\mathbb{E}}\, p(x)$, for every polynomial p of degree at most 2.

Why does Lemma 2.2 imply Theorem 2.1? The theorem follows by simply letting $z_i = \mathsf{sign}(y_i)$. Note that under our assumptions

$$\tfrac{1}{2n}\,\tilde{\mathbb{E}}\langle x, Lx\rangle = \tfrac{1}{2n}\,\tilde{\mathbb{E}}\left(\tfrac{1}{2d}\sum_{i\sim j}(x_i - x_j)^2\right) = \tfrac{1}{4dn}\sum_{i\sim j}\tilde{\mathbb{E}}(x_i - x_j)^2 \geq 1-\epsilon$$

and hence

$$\tfrac{1}{4dn}\sum_{i\sim j}\mathbb{E}(y_i - y_j)^2 \geq 1-\epsilon$$

We need to show that $\frac{1}{4dn}\sum_{i\sim j}\mathbb{E}(z_i - z_j)^2 \geq 1 - O(\sqrt{\epsilon})$. Using convexity, this follows from the following claim:

CLAIM: Let y, y' be Gaussian variables with $\mathbb{E}\, y^2 = \mathbb{E}\, y'^2 = 1$ such that $\mathbb{E}(y - y')^2 \geq 1-\delta$. Then $\Pr[\mathsf{sign}(y) = \mathsf{sign}(y')] \leq 10\sqrt{\delta}$.

This can be proven via a standard calculation (**Exercise 2.6:** do this), and essentially follows from the fact that two unit vectors of inner product $1-\delta$ have distance roughly $\sqrt{\delta}$. Thus all that remains is to prove the quadratic sampling lemma:

Proof of the quadratic sampling lemma. By shifting, it suffices to consider the case that $\tilde{\mathbb{E}}\, x_i = 0$ for all i (**Exercise 2.7:** show this). Let M be the $n \times n$ matrix such that $M_{i,j} = \tilde{\mathbb{E}}\, x_i x_j$. Since $\tilde{\mathbb{E}}\, f(x)^2 \geq 0$ for every linear function f, it must holds that M is positive semidefinite and hence $M = VV^\top$ for some matrix V. Another way to say this is that $M_{i,j} = \langle v^i, v^j \rangle$ for some N, and for some vectors $v^1, \ldots, v^n \in \mathbb{R}^N$. Let g be a standard Gaussian vector in $\mathbb{R}^N$ and define $y_i = \mathbb{E}\langle v^i, g\rangle$. (Note that, as desired, $\mathbb{E}\langle v^i, g\rangle = 0$ for all i.) For every i, j we have that

$$\mathbb{E}\, y_i y_j = \mathbb{E}\langle v^i, g\rangle\langle v^j, g\rangle = \sum_{k,\ell} \mathbb{E}\, v^i_k g_k v^j_\ell g_\ell$$

but since we have that $\mathbb{E}\, g_k g_\ell = \begin{cases} 1, & \text{if } k = \ell \\ 0, & \text{if } k \neq \ell \end{cases}$, this equals

$$\sum_k v^i_k v^j_k = \langle v^i, v^j \rangle = M_{i,j}$$

Hence y agrees with x on all quadratic monomials and hence on all quadratic polynomials as well. □

Exercise 2.8: Prove that if x is an actual distribution over $\mathbb{R}^n$ with mean 0^n, taking the value $x^{(\alpha)} \in \mathbb{R}^n$ with probability p_α, where α ranges over some finite index set I, we would get an equivalent distribution to the one above by letting $y = \sum_{\alpha \in I} g_\alpha x^{(\alpha)}$, where g_α is a Gaussian with mean 0 and variance p_α.

Note that the algorithm resulting from the proof of Theorem 2.1 performs the following: given the vectors $v_1, \ldots, v_n$ that arise from the pseudo-distribution $\{x\}$, we obtain a cut z (identified with a vector in $\{\pm 1\}^n$) by choosing a random gaussian vector g and outputting $z_i = \mathsf{sign}\langle v^i, g\rangle$. That is, we cut the vertices based on which side of the hyperplane defined by g they fall on. For this reason, this rounding technique is often known as “random hyperplane rounding”.

Also note that in analyzing this algorithm we didn’t use ”Marley’s Corollary”, since the quadratic sampling lemma is easy enough to prove even without assuming that $\{x\}$ was an actual distribution, but we will find that assumption useful as we analyze algorithms using higher degrees, including the Arora-Rao-Vazirani algorithm.

Remark: Vector view of SOS

In the proof above, we used the convenient fact that an $n \times n$ matrix M is psd if and only if there are vectors $v_1, \ldots, v_n \in \mathbb{R}^{\Omega}$ (for some Ω) such that $M_{i,j} = \langle v_i, v_j \rangle$. Thus a degree 2 pseudo-distribution can be completely characterized by such vectors (along with another vector which would correspond to the first moments / averages). This can be generalized to higher degrees as follows:

Exercise 2.9: Let $\mathcal{P}$ be some basis for the degree $\leq d/2$ polynomials (you can think of the monomial basis). Prove that a bilinear operator $M : \mathbb{R}^n_{d/2} \times \mathbb{R}^n_{d/2} \to \mathbb{R}$ is a degree d pseudo-expectation operator if and only if there exists vectors $\{v_p\}_{p \in \mathcal{P}}$ in $\mathbb{R}^N$ for some N such that $M(p,q) = \langle v_p, v_q \rangle$ and $\langle v_p, v_q \rangle = \langle v_r, v_s \rangle$ whenever $pq = rs$.

Note that if M corresponded to the expectation of an actual random variable X, which we can think of as a function from some probability space Ω to $\mathbb{R}^N$, then we could choose the vector v_p to have as its ω^{th} coordinate the value $p(X(\omega))$, where we will use the *expectation* inner product, i.e. $\langle\!\langle\!\langle u, v \rangle\!\rangle\!\rangle = \mathbb{E}_{\omega \in \Omega} u(\omega)v(\omega) = \sum_{\omega \in \Omega} p_\omega v_i(\omega) v_j(\omega)$ (p_ω is the probability of the element ω; the space Ω can even be infinite in which case the sum is replaced with an integral).

2.2 Can we do better with degree 2 SOS?

It is perhaps surprising that we can do better than the random cut algorithm, but knowing that we can bypass it whets our appetite for more. So far, we don't know if we can beat the Goemans-Williamson algorithm, but we do know that won't be possible with the degree-2 SOS program.

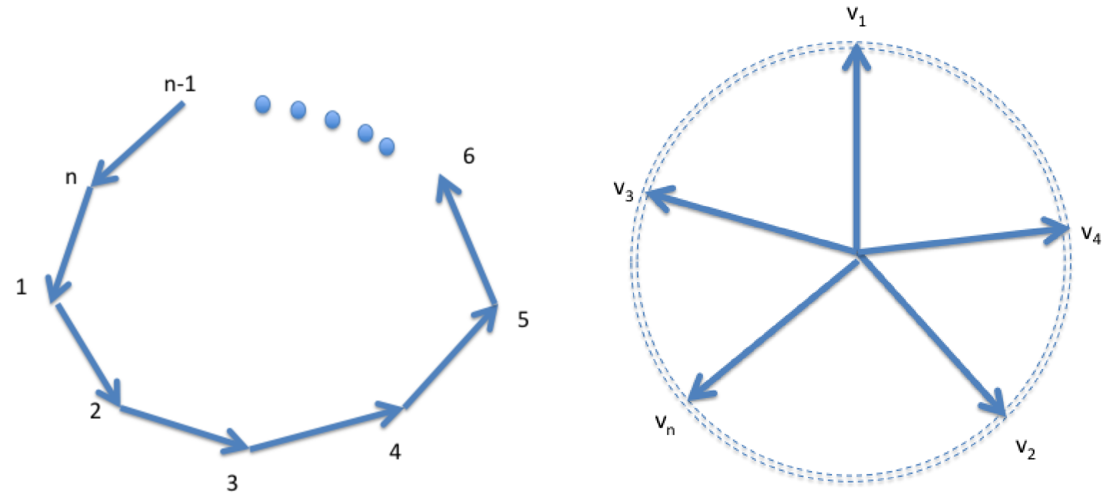
Theorem 2.3. *There is a graph G and $\epsilon > 0$ such that $\mathsf{maxcut}(G) \leq 1 - \sqrt{\epsilon}/10$ but there is a degree-2 pseudo-distribution $\{x\}$ such that $\tilde{\mathbb{E}}\, x_i^2 = 1$ for all i and $\tilde{\mathbb{E}}\langle x, Lx \rangle \geq 2n(1-\epsilon)$.*

Proof. The graph is simply the odd cycle on $n = 1/\sqrt{\epsilon}$ vertices. Since the graph is not bipartite, every cut must cut at least one edge, so $\mathsf{maxcut}(G) \leq 1 - 1/n = 1 - \sqrt{\epsilon}$. For the pseudo distribution, we arrange unit vectors $v_1, \ldots, v_n$ along the two dimensional circle such that $\langle v_i, v_j \rangle = -1 + \epsilon$ if $j = i + 1 \pmod{n}$. Let $\tilde{\mathbb{E}}\, x_i x_j = \langle v_i, v_j \rangle$. Then, for all i, $\tilde{\mathbb{E}}\, x_i^2 = \langle v_i, v_i \rangle = 1$, and $\tilde{\mathbb{E}}\langle x, Lx \rangle = \tilde{\mathbb{E}}(\sum_i x_i^2 - \frac{1}{2}\sum_{i \sim j} x_i x_j) = n(2-\epsilon) = 2n(1-\epsilon/2) \geq 2n(1-\epsilon)$. □

One issue with this result, apart from the fact that the odd cycle doesn't seem like a very hard instance, is that the value of ϵ here is $1/n^2$, which means that even finding a cut of $1 - 1/\sqrt{\epsilon}$ is pretty good (in fact, the best one can do, given that the graph isn't bipartite). However, this of course can be easily fixed by simply considering the disjoint union of many odd $1/\sqrt{\epsilon}$ cycles, hence yielding an instance where ϵ is independent of n. Another issue is to determine the right constant, and more generally, for every $\epsilon > 0$, come up with a graph that has a degree-2 pseudo-distribution pretending to range over cuts with value $1 - \epsilon$, but where the true max cut is at most $1 - f_{GW}(\epsilon) + o(1)$ where $f_{GW}(\cdot)$ is the function obtained by the proof of the Goemans-Williamson theorem (Theorem 2.1). This was achieved by Feige and Schechtman [FS02], who defined the following graph:

For $\epsilon > 0$, $\ell, n \in \mathbb{N}$, the Feige-Schechtman graph $FS(\epsilon, \ell, n)$ is obtained by sampling n random unit vectors $v_1, \ldots, v_n$ in $\mathbb{R}^\ell$ and letting $i \sim j$ if $\langle v_i, v_j \rangle \leq -1 + \epsilon$. We will typically think of the case that n is exponentially larger than ℓ and so this graph closely approximates the infinite graph where vertices are all vectors in the unit ℓ-dimensional sphere. In [FS02], Feige and Schechtman proved the following two results about this graph:

Figure 2.1: The odd-cycle graph, along with a depiction of a vector solution for the degree-2 SoS program.



Lemma 2.4. *There is a degree-2 pseudo-distribution $\{x\}$ such that $\tilde{\mathbb{E}}\, x_i^2 = 1$ for all i and $\tilde{\mathbb{E}}(x_i - x_j)^2 \geq 2(1-\epsilon)$ for all $i \sim j$.*

Proof. This is essentially by construction. Define $\tilde{\mathbb{E}}\, x_i x_j = \langle v_i, v_j \rangle$ where $v_1, \ldots, v_n$ are the vectors used to define the graph. Note that this is PSD and hence is a valid degree-2 pseudo-expectation operator, and that it satisfies the conditions of the Lemma. □

Lemma 2.5. *For every $\delta > 0$, if n is large enough, with high probability* $\mathsf{maxcut}(FS(\epsilon, \ell, n)) \leq 1 - f_{GW}(\epsilon) + \delta$

This is the heart of their proof. To show this, one needs to show that the maximum cut in the Feige-Schechtman graph is obtained by a *hyperplane cut*, namely by a set S of the form $\{i : \langle v_i, a \rangle > 0\}$ for some vector a. This turns out to be related to classical isoperimetric results of Borell. We will prove a slightly weaker result. Namely, that the Feige-Schechtman graph is at least not worse than the cycle:

Lemma 2.6. *There is some constant $c > 0$ such that for every $\delta > 0$, if n is large enough,* $\mathsf{maxcut}(FS(\epsilon, \ell, n)) \leq 1 - c\sqrt{\epsilon}$.

Proof. We will assume that $1/\sqrt{2\epsilon - \epsilon^2} = k$ for some odd integer k (this assumption can be waived with a bit more work, at the cost of having a suboptimal value for the constant c). For sufficiently large n, we can imagine that the FS graph is simply on the continuous sphere. Pick a random edge $i \sim j$ of the graph, and consider the intersection of the sphere with the plane spanned by v_i and v_j. Inside that plane we can find a k-cycle subgraph of the graph that contains the edge $i \sim j$. By following this approach we obtain a set $C_1, \ldots, C_N$ of k-cycles that uniformly covers the edges of the Feige-Schechtman graph (i.e., each edge of the Feige-Schechtman graph is contained in about the same numbe of cycles). Now, for every possible cut S, it must miss at least one edge from every one of those C_i's. But by the uniformity condition, if a cut misses a α fraction of the edges in the FS graph, on average it should miss an α fraction (or $\alpha \pm o(1)$, to account for our finite approximations) of the edges in the cycles. □

This is a weaker result because the constant c will not match that in f_{GW}.

Similar results hold for *sparsest cut*— the cycle (here it doesn't matter if it's even or odd) yields an example of a graph with eigenvalue gap of ϵ but where the best cut has conductance $\sqrt{\epsilon}$, and the analog of the Feige-Schechtman graph can give better constant dependence.

2.3 Can we do better with higher degree?

This is a question many people are interested in, and we don't know the answer. As mentioned above, Khot's *Unique Games Conjecture* implies that no polynomial-time algorithm (or even an $\exp(n^{o(1)})$-time one) can beat the Goemans-Williamson algorithm. So, in particular it should mean that using SOS algorithm with degree $n^{o(1)}$ will not yield improved performance over degree 2. However, we don't even know if degree 4 SOS doesn't do better than degree 2. Indeed, the known hard instances for degree 2, including the odd cycle and the Feige-Schechtman graph, can in fact be solved via degree 4 SOS (for some other instances the best known bound is 16 or so, but we have no evidence that degree 4 doesn't work as well).

Lemma 2.7. *Let n be odd. There is no degree-4 pseudo-distribution $\{x\}$ over $\mathbb{R}^n$ consistent with the constraints $\{x_i^2 = 1\}_{i=1..n}$ such that $\tilde{\mathbb{E}} \sum_{i=1}^n (x_i - x_{i+1})^2 > 4(n-1)$ (identifying x_{n+1} with x_1).*

The proof of this lemma follows from the following exercise

Exercise 2.10: (Squared Triangle Inequality) Let $\{x\}$ be a degree-4 pseudo-distribution over $\mathbb{R}^n$ consistent with the constraints $\{x_i^2 = 1\}$. Then for all $i, j, k \in [n]$, $\tilde{\mathbb{E}}(x_i - x_k)^2 \leq \tilde{\mathbb{E}}(x_i - x_j)^2 + \tilde{\mathbb{E}}(x_j - x_k)^2$.

Note that if $\{x\}$ is an *actual* distribution consistent with these constraints, then it means that it is supported on $\{\pm 1\}^n$, and hence $\mathbb{E}(x_i - x_j)^2 = 4\Pr[x_i \neq x_j]$ which immediately implies the inequality. Therefore, the 3-variate polynomial $(x_i - x_j)^2 + (x_j - x_k)^2 - (x_i - x_k)^2$ is non-negative on $\{\pm 1\}^3$ which immediately implies the result for degree-6 pseudo-distributions. (Can you see why?) This is just as good for our purposes, but working out the degree 4 case should be a nice exercise.

We can now prove the lemma:

Proof. Using the equation $(a+b)^2 = 2a^2 + 2b^2 - (a-b)^2$ we get that

$$\tilde{\mathbb{E}} \sum_{i=1}^n (x_i + x_{i+1})^2 < 4n - 4(n-1) = 4$$

By the triangle inequality applied to the variables x_i, x_{i+2} and $-x_{i+1}$, we get that

$$\tilde{\mathbb{E}}(x_i - x_{i+2})^2 \leq \tilde{\mathbb{E}}(x_i + x_{i+1})^2 + (x_{i+1} + x_{i+2})^2$$

which repeating $(n-1)/2$ times we get that

$$\tilde{\mathbb{E}}(x_i - x_{i+1})^2 \leq \sum_{j \in (i+1,..,n,1,..,i-1)} \tilde{\mathbb{E}}(x_j + x_{j+1})^2$$

If we sum this over all i's, then on the RHS every term $\tilde{\mathbb{E}}(x_j + x_{j+1})^2$ gets counted $n-1$ times and so we get

$$\sum_i \tilde{\mathbb{E}}(x_i - x_{i+1})^2 \leq (n-1) \sum_j \tilde{\mathbb{E}}(x_j + x_{j+1})^2 < (n-1)4$$

contradicting our assumptions. □

This results immediately implies that the degree 4 SOS algorithm can also certify that the max-cut of the $FS(\epsilon, \ell, n)$ graph is $1-\Omega(\sqrt{\epsilon})$ if n is large enough (can you see why?). Interestingly, it can be shown that if we choose a significantly smaller n (though still exponential in ℓ) so that almost all short cycles (and in particular the odd ones) disapper, then as long as the average degree remains large enough, the value of the maximum cut remains $1-\Omega(\sqrt{\epsilon})$. However, the proof that the degree 4 SOS algorithm certifies this breaks down. Nonetheless, it turns out that the degree-4 SOS algorithm still gives a value of $1-\Omega(\sqrt{\epsilon})$ even in this regime (see Barak, Hardt, Holenstein, Steurer [BHHS11]).

2.4 SOS'ing proofs of isoperimetric inequalities

Moving beyond max cut, another important problem is the *small set expansion* problem. This is the task, given some graph $G = (V, E)$ and δ, of computing $\min_{|S| \leq \delta |V|} \phi(S)$. Once again, the question is about approximating it, and the *small set expansion conjecture* posits that it is NP hard to determine if this quantity is at most ϵ or at least $1-\epsilon$, where $\epsilon = 1/O(\log(1/\delta))$. (The conjecture was stated by Raghavendra and Steurer [RS10] in a slightly different form, and its equivalence to this form was shown by Raghavendra, Steurer and Tulsiani [RST10].)

The *Boolean cube* (i.e., the graph on 2^ℓ vertices identified with $\{\pm 1\}^\ell$ such that $x \sim y$ if $\sum |x_i - y_i| = 2$) is a canonical example of a small set expander. That is, even though the graph is not a great expander, since for example $1 - 1/\ell$ fraction of the edges touching the set $S = \{(+1, x) : x \in \{\pm 1\}^{\ell-1}\}$ stay inside it, for smaller set a much larger fraction of the edges go out. In fact, one can show that for every k, the sets S of measure 2^{-k} that minimize $\phi(S)$ have the form $S = \{(\alpha, x) : x \in \{\pm 1\}^{\ell-k}\}$ for some $\alpha \in \{\pm 1\}^k$. (Note that these sets have $1 - k/\ell$ of their edges staying inside them.)

How do you prove such a thing (at least approximately)? The key here is again linear algebra. Recall that for every set S of measure less than 1/2, $\langle x, Lx \rangle / \|x\|^2 = \phi(S)$ where x is the $\{0, 1\}$ characteristic vector of S. Therefore, to prove that if $|S|$ is small then $\phi(S)$ is large, it is enough to show that sparse vectors are not close to the low eigenspace of the operator L. Specifically, we have the following result:

Lemma 2.8. *Let $G = (V, E)$ be regular graph, $\lambda \in (0, 1)$ and W be the span of eigenvectors of $L(G)$ corresponding to eigenvalue at most λ. If every $w \in W$ satisfies:*

$$\mathbb{E}_i\, w_i^4 \leq C \left(\mathbb{E}_i\, w_i^2\right)^2 \tag{2.1}$$

then for every set S of measure δ set,

$$\phi(S) \geq \lambda(1 - \sqrt{C\delta})$$

To understand the lemma, let us try to parse what (6.2) means in the case that w is the 0/1 characteristic vector of some set S of measure μ. In this case the LHS equals μ and the RHS equals $C\mu^2$, and so we get that $\mu \geq 1/C$. Thus in particular (6.2) implies that the space W does not contain the characteristic vector of any set of measure $< 1/C$, and the conclusion of the Lemma is that if S has measure $\ll 1/C$ then it has expansion at least $\lambda - o(1)$, which means that its characteristic vector has almost all of its mass outside W.

Proof. Throughout this proof, it will be convenient for us to use the *expectation* norms and inner product, and so we denote $\|\!|x|\!\|_p = (\mathbb{E}_i\, |x_i|^p)^{1/p}$ and $\langle\!\langle\!\langle x, y \rangle\!\rangle\!\rangle = \mathbb{E}\, x_i y_i$. Thus (6.2) translates to

$\|w\|_4^4 \leq C\|w\|_2^4$ for every $w \in W$. Let S be the set and x its 0/1 characteristic vector. Note that we still have

$$\phi(S) = \langle\langle\langle x, Lx\rangle\rangle\rangle / \|x\|^2 \ .$$

Write $x = x' + x''$ where $x' \in W$ and $x'' \in W^{\perp}$. Our main claim is

Claim: $\|x'\|_2 \leq C^{1/4}\|x\|_{4/3}$

Proof of Claim:

$$\|x'\|_2^2 = \langle\langle\langle x', x'\rangle\rangle\rangle = \langle\langle\langle x', x\rangle\rangle\rangle \leq \|x'\|_4\|x\|_{4/3}$$

where the second equality is because x' is a projection of x, and the last inequality is an application of Holder's inequality. The proof then follows using $\|x'\|_4 \leq C^{1/4}\|x'\|_2$.

Given this, since $\|x\|_{4/3} = \delta^{3/4}$ we can use the eigenvector decomposition $(v_1, \ldots, v_n)$ and $(\lambda_1, \ldots, \lambda_n)$ of L to write

$$\langle\langle\langle x, Lx\rangle\rangle\rangle = \sum \lambda_i \langle\langle\langle x, v_i\rangle\rangle\rangle^2$$

bunching together all the vectors with eigenvalue smaller than λ (whose contribution to the sum is non-negative), and all the vectors with eigenvalues larger than λ (who contribute at least $\lambda\|x''\|_2^2$) we get

$$\langle\langle\langle x, Lx\rangle\rangle\rangle \geq \lambda\|x''\|_2^2 = \lambda(\|x\|_2^2 - \|x'\|_2^2) \geq \lambda(\delta - C^{1/2}\delta^{3/2})$$

and thus (using $\|x\|_2^2 = \delta$) we get

$$\phi(S) = \tfrac{\langle\langle\langle x, Lx\rangle\rangle\rangle}{\|x\|_2^2} \geq \lambda(1 - \sqrt{C\delta})$$

□

Let us now see how we apply this result to the Boolean cube. First, we need the following characterization of the Boolean cube

Exercise 2.11: Let G be the Boolean cube on $\{\pm 1\}^{\ell}$. Prove that the eigenvectors of G are $\{\chi_S\}_{S\subseteq[\ell]}$ where for every $x \in \{\pm 1\}^{\ell}$, $\chi_S(x) = \prod_{i\in S} x_i$ and the eigenvalue corresponding to χ_S is $|S|/\ell$.

Thus, for every λ, the subspace spanned by eigenvectors of eigenvalue at most λ is the set of $f : \{\pm 1\}^{\ell} \to \mathbb{R}$ that are spanned by the functions χ_S with $|S| \leq \lambda\ell$. Thus the following result shows that sufficiently small sets in the hypercube expand a lot

Theorem 2.9 ((2, 4) hypercontractivity). *Let $f = \sum f_{\alpha}\chi_{\alpha}$ with $|\alpha| \leq d$. Then*

$$\mathbb{E}_{x\in\{\pm 1\}^{\ell}} f(x)^4 \leq 9^d \left(\mathbb{E}_{x\in\{\pm 1\}^{\ell}} f(x)^2\right)^2 \tag{2.2}$$

Theorem 2.9 has a simple proof but underlies many results used in hardness of approximation, social choice theory, and more (see Ryan's book [O'D14]). In particular, as we mentioned, by combining it with Lemma 6.2 it implies some isoperimetric results on the Boolean cube.

Proof. We prove the result by induction on d and ℓ. (The case $\ell = 0$ or $d = 0$ is trivial.) Separate f to the parts that do and don't depend on ℓ and write

$$f(x) = f_0(x_1, \ldots, x_{\ell-1}) + x_{\ell} f_1(x_1, \ldots, x_{\ell-1})$$

note that the degree of f_1 is at most $d-1$. Now let us expand $\mathbb{E}\, f(x)^4$ and note that the expectation of odd powers of x_ℓ vanish (since it is independent from the other variables) and so we get that

$$\mathbb{E}\, f^4 = \mathbb{E}\, f_0^4 + f_1^4 + 6 f_0^2 f_1^2 \tag{2.3}$$

By Cauchy Schwarz we can bound the last term by $6\sqrt{\left(\mathbb{E}\, f_0^4\right)\left(\mathbb{E}\, f_1^4\right)}$. By induction we can assume $\mathbb{E}\, f_b^4 \le 9^d \left(\mathbb{E}\, f_b^2\right)^2$ for $b = 0, 1$ and so plugging this into (2.3) we get

$$\mathbb{E}\, f^4 \le 9^d \left(\mathbb{E}\, f_0^2\right)^2 + 9^{d-1}\left(\mathbb{E}\, f_1^2\right)^2 + 6 \cdot 9^{d-1/2}\, \mathbb{E}\, f_0^2\, \mathbb{E}\, f_1^2 \le$$

$$9^d \left(\left(\mathbb{E}\, f_0^2\right)^2 + \left(\mathbb{E}\, f_1^2\right)^2 + 2\, \mathbb{E}\, f_0^2\, \mathbb{E}\, f_1^2 \right) =$$

$$9^d \left(\mathbb{E}\, f_0^2 + \mathbb{E}\, f_1^2\right)^2$$

but this equals $9^d \left(\mathbb{E}\, f^2\right)^2$, since $f = f_0 + x_\ell f_1$ and $\mathbb{E}\, x_\ell f_0 f_1 = 0$. □

Remark: Max-cut, sparsest-cut and small-set expansion as finding non-Gaussian vectors in subspaces

The sparsest-cut, max-cut, and small-set expansion can all be thought of as the problem of finding

$$\min_x p(x)$$

where $p : \mathbb{R}^n \to \mathbb{R}$ is a quadratic polynomial (e.g., $p(x) = \langle x, Lx \rangle$ or $p(x) = -\langle x, Lx \rangle$) and subject to x satisfying certain constraints (e.g. $x \in \{0,1\}^n$ (or sometimes $\{\pm 1\}^n$) in the case of max-cut/sparsest cut, or x is the characteristic vector of a sparse set in the case of small set expansion). By scaling appropriately, we can also assume x is restricted to have unit norm.

Since a vector x of unit norm minimizes a quadratic form $p(\cdot)$ if and only if it resides in the linear subspace correspondings to the small eigenvalues of $p(\cdot)$ (thought of as a linear operator), the problem essentially reduces to finding a vector in (or close to) W that satisfies these constraints. If W was a "generic" subspace of not too high a dimension, then all the vectors w inside it would be rather "smooth" and in particular every $w \in W$ will satisfy that the distribution $X(w)$ which is obtained by sampling a random coordinate i and outputting w_i is close to the Gaussian distribution. So, in some sense all of these problems are about finding non-Gaussian vectors in a subspace. Note that for every w there is some Gaussian distribution that matches the first two moments of $X(w)$. Thus being "non Gaussian" inherently implies looking at higher moments. For example, one can verify that if w is a sparse vector shifted and scaled to have $\mathbb{E}\, X(w) = 0$ and $\mathbb{E}\, X(w)^2 = 1$ then $\mathbb{E}\, X(w)^4$ is much larger than $\mathbb{E}\, N(0,1)^4$ (indeed this is the underlying reasoning behind Lemma 6.2, and as I mentioned, some kind of a reverse direction holds as well). So, one can hope that degree > 2 SOS would help with that.

Making this into an SOS proof

Lemma 6.2 reduces the question of certifying whether a graph is a small set expander to certifying a polynomial equation, and so to understand if the degree 4 SOS algorithm can certify the expansion properties of the Boolean cube, we need to come up with an SOS proof for the $(2, 4)$ hypercontractivity theorem (Theorem 2.9). Eyeballing the proof, we see that it doesn't use the probabilistic method, and so by Marley's Corollary it should have an SOS proof. However, as far as we know, Marley didn't publish his proof in a peer-reviewed journal, and so we'd better doublecheck the case. (I should note that joking aside, it is definitely *not* a universal statement that all known interesting low degree polynomial inequalities that have a non-probabilistic method proof are known to have a low degree SOS proof, in fact Ryan O'Donnell has several interesting open questions along these lines. However so far in my experience, though it took some work, we typically were always able to find such proofs for the statements that arise in analyzing SOS algorithms (or some close-enough approximation of them), and so the main hurdle was to actually phrase the statement as low degree polynomial inequality in the first place.)

Indeed, this turns out to be the case as shown by the following lemma giving an SOS proof for Theorem 2.9. For two polynomials p, q we write $p \preceq q$ if $p = q - \sum_{i=1}^{m} r_i^2$ for some polynomials $r_1, .., r_m$.

Lemma 2.10. *Let $d, \ell \in \mathbb{N}$. For every $x \in \{\pm 1\}^n$, let $L_x(\cdot)$ be the linear function in the variables $\{f_\alpha\}_{|\alpha| \le d}$ such that*

$$L_x(f) = \sum_{\alpha} \left(\prod_{i \in \alpha} x_i \right) f_\alpha$$

then

$$\mathbb{E}_{x \in \{\pm 1\}^\ell} L_x(f)^4 \preceq 9^d \left(\mathbb{E}_{x \in \{\pm 1\}^\ell} L_x(f)^2 \right)^2 \tag{2.4}$$

The notation in this lemma are a bit subtle, and so it's worth taking the time and make sure we parse it. First, note that the lemma immediately implies Theorem 2.9. Indeed, since $L_x(f) = f(x)$, plugging this in (2.5) yields (2.2). However, we use the second notation to emphasize that x is *not* a variable of L_x, nor of the polynomials implicitly defined in (2.5). These are polynomials in the coefficients of f. In particular, L_x is linear, no matter what the number d is, and both the LHS and RHS of (2.2) are degree 4 polynomials. The lemma also implies that if $\{f\}$ is a degree-4 pseudo-distribution then

$$\tilde{\mathbb{E}}_f \, \mathbb{E}_{x \in \{\pm 1\}^\ell} f(x)^4 \le 9^d \, \tilde{\mathbb{E}}_f \left(\mathbb{E}_{x \in \{\pm 1\}^\ell} f(x)^2 \right)^2$$

We now turn to prove the lemma. The proof is a close variant of the proof of Theorem 2.9 but we have to be a bit more careful and make a stronger induction hypothesis. In particular, we will prove the following stronger result

Lemma 2.11. *Let $d, e, \ell \in \mathbb{N}$. For every $x \in \{\pm 1\}^n$, let $L_x(\cdot)$ be the linear function in the variables $\{f_\alpha\}_{|\alpha| \le d}$ such that*

$$L_x(f) = \sum_{\alpha} \left(\prod_{i \in \alpha} x_i \right) f_\alpha$$

and let $L'_x(\cdot)$ be the linear function in the variables $\{g_\alpha\}_{|\alpha| \le e}$ such that

$$L'_x(f) = \sum_{\alpha} \left(\prod_{i \in \alpha} x_i \right) g_\alpha$$

then

$$\mathbb{E}_{x\in\{\pm1\}^\ell} L_x(f)^2 L'_x(g)^2 \preceq 9^{(d+e)/2} \left(\mathbb{E}_{x\in\{\pm1\}^\ell} L_x(f)^2\right) \left(\mathbb{E}_{x\in\{\pm1\}^\ell} L'_x(f)^2\right) \tag{2.5}$$

Proof. We prove the lemma by induction on ℓ, d, e, again, if any of those is zero then the result is trivial. Below we use the notation $f(x)$ for $L_x(f)$ and $g(x)$ for $L'_x(g)$, but you should remember that $f(x)$ is a linear polynomial in the variables $\{f_\alpha\}$ (rather than being a degree d polynomial in x).

Let f_0, f_1, g_0, g_1 be such that $f(x) = f_0(x) + x_\ell f_1(x)$ and $g(x) = g_0(x) + x_\ell g_1(x)$. Note that the coefficients of f_0, f_1, g_0, g_1 are a linear function of the coefficients of f, g (because $f_0(x) = \frac{1}{2}f(x_1,\ldots,x_{n-1},1) + \frac{1}{2}f(x_1,\ldots,x_{n-1},-1)$ and $f_1(x) = \frac{1}{2}f(x_1,\ldots,x_{n-1},1) - \frac{1}{2}f(x_1,\ldots,x_{n-1},-1)$). Moreover, the monomial-size of f_0, f_1, g_0, and g_1 are at most d, $d-1$, e, and $e-1$, respectively.

Since $\mathbb{E}\, x_\ell = \mathbb{E}\, x_\ell^3 = 0$, if we expand $\mathbb{E}\, f^2g^2 = \mathbb{E}(f_0 + x_\ell f_1)^2(g_0 + x_\ell g_1)^2$ then the terms where x_ℓ appears in an odd power vanish, and we obtain

$$\mathbb{E}\, f^2g^2 = \mathbb{E}\, f_0^2g_0^2 + f_1^2g_1^2 + f_0^2g_1^2 + f_1^2g_0^2 + 4f_0f_1g_0g_1$$

By expanding the square expression $2\,\mathbb{E}(f_0f_1 - g_0g_1)^2$, we get $4\,\mathbb{E}\, f_0f_1g_0g_1 \preceq 2\,\mathbb{E}\, f_0^2g_1^2 + f_1^2g_0^2$ and thus

$$\mathbb{E}\, f^2g^2 \preceq \mathbb{E}\, f_0^2g_0^2 + \mathbb{E}\, f_1^2g_1^2 + 3\,\mathbb{E}\, f_0^2g_1^2 + 3\,\mathbb{E}\, f_1^2g_0^2\,. \tag{2.6}$$

Applying the induction hypothesis to all four terms on the right-hand side of 2.6 (using for the last two terms that the monomial-size of f_1 and g_1 is at most $d-1$ and $e-1$),

$$\begin{aligned}\mathbb{E}\, f^2g^2 &\preceq 9^{\frac{d+e}{2}}\left(\mathbb{E}\, f_0^2\right)\left(\mathbb{E}\, g_0^2\right) + 9^{\frac{d+e}{2}}\left(\mathbb{E}\, f_1^2\right)\left(\mathbb{E}\, g_1^2\right)\\ &\quad + 3\cdot 9^{\frac{d+e}{2}-1/2}\left(\mathbb{E}\, f_0^2\right)\left(\mathbb{E}\, g_1^2\right) + 3\cdot 9^{\frac{d+e}{2}-1/2}\left(\mathbb{E}\, f_1^2\right)\left(\mathbb{E}\, g_0^2\right)\\ &= 9^{\frac{d+e}{2}}\left(\mathbb{E}\, f_0^2 + \mathbb{E}\, f_1^2\right)\left(\mathbb{E}\, g_0^2 + \mathbb{E}\, g_1^2\right)\,.\end{aligned}$$

Since $\mathbb{E}\, f_0^2 + \mathbb{E}\, f_1^2 = \mathbb{E}(f_0 + x_\ell f_1)^2 = \mathbb{E}\, f^2$ (using $\mathbb{E}\, x_\ell = 0$) and similarly $\mathbb{E}\, g_0^2 + \mathbb{E}\, g_1^2 = \mathbb{E}\, g^2$, we derive the desired relation $\mathbb{E}\, f^2g^2 \preceq 9^{\frac{d+e}{2}}\left(\mathbb{E}\, f^2\right)\left(\mathbb{E}\, g^2\right)$. □

Chapter 3

SOS Lower bounds: 3XOR/3SAT and planted clique

These notes are an expanded version of the notes for my summer course scribed by Akash Kumar

In this lecture we will see some *lower bounds* (or more accurately, negative results) for the Sum of Squares algorithm. Namely, we will see computational problems which the SOS algorithms *fails* to solve with a small degree. Another way to say this is that we will see cases that demonstrate the difference between pseudo-distributions and actual distributions. If we take the point of view that pseudo-distributions capture the knowledge of a computationally-bounded observer, then these are examples where being computationally bounded has very a significant effect on this knowledge. We have jokingly referred to "Marley's Corollary" as roughly saying that as long as you don't use the probabilistic method then "every little thing gonna be alright" and pseudo-distributions can be counted on to be similar to actual distributions. Thus it's not surprising that we will in fact use the probabilistic method in constructing these examples.

3.1 Lower bounds for random 3SAT/3XOR

In the Max-3XOR problem, we are given a set of linear equations $\pmod 2$ in n Boolean variables $x_1, \ldots, x_n$ such that each equation only involves three variables (i..e., has the form $x_i \oplus x_j \oplus x_k = a_{i,j,k}$), and we need to find the assignment x that satisfies the largest number of equations. (For simplicity of notation, we will actually think of x a string in $\{\pm 1\}^n$, meaning that the equations have the form $x_i x_j x_k = a_{i,j,k}$ for some $a_{i,j,k} \in \{\pm 1\}$.) Finding whether or not there exists an assignment that satisfies *all* equations can of course be done via Gaussian elimination, but Håstad proved in 1995 that for every $\epsilon > 0$ it is NP-hard to distinguish between the case that there is an assignment satisfying $1 - \epsilon$ fraction of the equations, and the case that every assignment satisfies $1/2 + \epsilon$ fraction of the equations. (There is always an assignment satisfying $1/2$ of the equations— can you see why?.) In fact, since Håstad's reduction only had linear blowup from the underlying PCP system (known as "Label Cover"), and thanks to the work Moshkovitz and Raz we now know of such PCP systems which themselves have a quasilinear blowup from 3SAT, if we assume that there is no $2^{n^{0.999}}$ algorithm for 3SAT (an extremely reasonable assumption which is in fact weaker than what's known as the "Exponential Time Hypothesis") then the SOS algorithm would require degree at least $n^{0.999}$ to do so. However, it is always good to verify these predictions by proving them unconditionally, which is what is achieved (in a very strong form) by the following theorem:

Theorem 3.1 (Grigoriev, 1999). *For every constant $\epsilon > 0$ and large enough n, there exists an instance ψ of Max-3XOR over n variables such that:*

- *Every assignment $x \in \{\pm 1\}^n$ satisfies at most $1/2 + \epsilon$ fraction of the equations of ψ.*
- *There exists a degree $\Omega(n)$ pseudo distribution $\{x\}$ that is consistent with the constraints $\{x_i^2 = 1\}$ for all $i \in [n]$ and the constraint $\{x_i x_j x_k = a_{i,j,k}\}$ for every i, j, k such that ψ contains the equation $a_{i,j,k} x_i x_j x_k = 1$.*

Moreover, there is $m = O(n)$ such that with constant positive probability, a random ψ with m equations will satisfy the properties above. (I believe this constant probability can actually be upgraded to $1 - o(1)$ at the expense of a slight complication in the proof; ***Exercise 3.1:*** *verify this, see footnote for hint*[1]*)*

Note that this theorem is stronger than what is predicted by the NP-hardness results, as it says that SOS cannot even distinguish between the case that the equations are completely satisfiable and the case that one can satisfy at most a $1/2 + \epsilon$ fraction, which as we mentioned can in fact be done in polynomial time via Gaussian elimination. So how come this powerful algorithm does not solve this easy problem? One answer is that while the SOS algorithm may be optimal in some domains, it does not mean it's optimal for all problems.[2] In particular, it does not seem able to take advantage of algebraic structure such as the one present in linear equations. Another related observation is that, unlike some other algorithms, the SOS algorithm (at least when applied to natural simple systems of equations such as those arising from constraint satisfaction problems) doesn't seem to do "half measures" in the sense that it is *inherently robust to noise.* That is, because of its continuous nature, the SOS algorithm does not really distinguish between the case that x satisfies all the equations (i.e. $\sum_{i,j,k} a_{i,j,k} x_i x_j x_k = m$) and the case that it satisfies almost all of them (i.e. $\sum_{i,j,k} a_{i,j,k} x_i x_j x_k \geq (1 - \epsilon)m$) . This is in stark contrast to algebraic algorithms such as Gaussian elimination that are very *brittle*, and completely break down even in the presence of very small amounts of noise. Therefore, if the SOS algorithm would have solved the "1 vs. $1/2 + \epsilon$" Max-3XOR problem, then it would also have been able to solve the "$1 - \epsilon$ vs. $1/2 + \epsilon$" variant of the problem, but this latter variant is NP-hard.

Some bibliographical remarks: Theorem 3.1 was proven by Grigoriev in 2001, and later rediscovered by Schoenebeck in 2008. Schoenebeck also observed that it immediately implies a lower bound for 3SAT, since $a \oplus b \oplus c = 1$ implies that $a \vee b \vee c = 1$, and a random 3SAT instance is unsatisfiable. Tusliani extended this further by first showing that the same ideas can be used to show a lower bound for a very particular type of the "Label Cover" problem, which can be thought of as an "SOS PCP theorem". He then showed that SOS lower bounds are in general closed under "gadget reductions" and so managed to transform many of the NP-hardness results obtained from the PCP theorem into matching unconditional SOS lower bounds. Siu On Chan (2013) provided a very interesting result in the other direction, giving an actual PCP Theorem that exactly matches the parameters of the "SOS PCP Theorem". An excellent question asked

[1] **Hint:** The reason that a random instance is not an expander with high probability is that there may be a few pairs of 3-variable equations that have two common variables. The effect of this should be the same as adding a small number of equations of the form $x_i \oplus x_j = \sigma$ (or $x_i x_j = \sigma$ in $\{\pm 1\}$ notation) to the instance, and one should be able to show that the pseudo-distribution we construct can be modified to satisfy these constraints as well.

[2] Throughout this course, when saying that the SOS algorithm solves a problem X, I always assume that the representation of X as polynomial equations is fixed to some canonical form. If we allowed arbitrary polynomial-time computable representations then we could simulate any algorithm using the SOS algorithm as even linear programming is **P**-complete.

in class is whether there is an algorithm that combines both SOS and Gaussian elimination in a natural way. I don't know of an algorithm for this, but there is a proof system, which simply uses the same rules of derivation $\{P \geq 0, Q \geq 0\} \models \{P + Q \geq 0, PQ \geq 0\}$ as the SOS system but tracks the *actual* degree as opposed to the *syntactic* degree, see this paper of Grigoriev, Hirsch, and Pasechnik `http://eccc.hpi-web.de/report/2001/103/`. A proof system is still very interesting since it can demonstrate that a problem lies in $NP \cap coNP$ and (as we discussed) there are very few examples for problems in this class that are not also known to be in P. We know very few lower bounds for this system, though the NP completeness results imply that there should be a 3XOR instance where one can satisfy at most, say, 0.51 fraction of the equations, but the best upper bound proven by this system with degree $\ll n$ would not be better than 0.99.

3.1.1 Proving Theorem 3.1

To prove Theorem 3.1 we need to (1) give a construction of some highly unsatisfiable 3XOR instance ψ and (2) construct a degree $O(n)$ pseudo-distribution $\{x\}$ that pretends to satisfy all the constraints of ψ . As mentioned, the construction of ψ is simple - we simply choose it as a random 3XOR instance with $m = cn$ constraints for some constant c (depending on ϵ).[3] Let us think of the choice of ψ as choosing a bipartite graph on G on $m + n$ vertices with left-degree 3 and a random $a \in \{\pm 1\}^m$ such that the ℓ^{th} equation is that $a_\ell x_i x_j x_k = 1$ where $\{i, j, k\}$ are the neighbors of ℓ. The theorem follows from the following lemmas:

Lemma 3.2. *For every G, with $1 - \exp(-\Omega(n))$ probability over the choice of $a \in \{\pm 1\}^{cn}$ it will hold that every assignment x satisfies at most $1/2 + \epsilon$ fraction of the equations where ϵ tends to zero as c grows.*

Lemma 3.3. *There is some $\delta, p > 0$ such that with probability at least $p > 0$ the graph G is a $(\delta n, 1.7)$-expander, where we say that a bipartite graph $G = (L, R, E)$ is a (s, α) expander if $|\Gamma(S)| \geq \alpha |S|$ for every $S \subseteq L$ with $|S| \leq s$, where $\Gamma(S)$ denotes the set of neighbors of S.*

Lemma 3.4. *For every $a \in \{\pm 1\}^m$, if G is a (k, α) expander for $\alpha > 1.5$ then there exists a degree $k/100$ pseudo-distribution $\{x\}$ consistent with the constraints $\tilde{\mathbb{E}}\, x_i x_j x_k = a_\ell$ for every ℓ and $\{i, j, k\} = \Gamma(\ell)$.*

We defer the proofs of Lemmas 3.2 and 3.3. The proofs use, as promised, the probabilistic method, but are not complicated and follow by the usual Chernoff+union bound argument. Despite being simple, the fact that they use the probabilistic method is in some sense the reason that they do not carry over to the SOS setting. This serves as a caution that we shouldn't equate a proof being "SOS'able" with it being "simple"— there can be highly complex proof in the SOS setting, and every simple proof that is not "SOS'able" with low degree. I believe that Lemma 3.3 actually can be derandomized using the paper of Capalbo, Reingold, Vadhan, Wigderson. One way to demonstrate that Marley's corollary is not a universal rule is to do **Exercise 3.2:** Prove that there is no SOS proof that the CRVW graph is a (k, α) expander for $\alpha > 1.5$. (I actually don't know if this exercise is true, and if it is I think it would actually be an interesting research result, showing a "robust" SOS lower bound with a deterministic construction.) Alekhnovich (2001) had a fascinating conjecture that as long as G is a sufficiently good expander, the instance (G, a) with a random a would be hard.

Note that despite having a very simple proof, I don't know of any derandomization for Lemma 3.2.

[3] We will use the words equations, constraints and clauses interchangebly.

3.1.2 Proof of Lemma 3.4

To prove Lemma 3.4 we need to show the existence of a degree $\Omega(n)$ pseudo-distribution $\{x\}$ that pretends to range over $x \in \{\pm 1\}^n$ that satisfies the constraints of the system (G, a). Our philosophy is that a pseudo-distribution captures the knowledge of *computationally bounded* observer. Note that (actual) distributions are the standard way to model knowledge of a *computationally unbounded* observers that only have *partial information*— this is known as Bayesian reasoning. For example, suppose that Mickey is a computationally all-powerful observer. If Mickey was given no information at all then we model its knowledge by the uniform distribution over $\{\pm 1\}^n$. Note that this distribution satisfies that $\mathbb{E}\, x_S := \mathbb{E} \prod_{i \in S} x_i = 0$ for every non empty set S. (Note that since the distribution is over x's that satisfy $x_i^2 = 1$ then knowing $\mathbb{E} \prod_{i \in S} x_i$ for every set S suffices to deduce $\mathbb{E}\, p(x)$ for every polynomial $p(\cdot)$.)

If Mickey later learns that $x_7 x_{13} x_{22} = -1$ and $x_{22} x_{44} x_{60} = +1$ then we model his knowledge by the distribution that is uniform over the the strings that satisfy these conditions— namely $\mathbb{E}\, x_S = 0$ unless S is either empty or one of $\{7, 13, 22\}, \{22, 44, 60\}$ and $\{7, 13, 44, 60\}$.

Now if Mickey was given all the equations, then, being computationally unbounded, then if the equations come from Lemma 3.2 he would be to figure out that there exists no x that satisfies all the equations (or even a 0.6 fraction of them), and hence that there simply exists no such distribution. However, if these equations are given to the Donald that can only do n^s-time computation, then he might not be able to do that. Specifically, Donald could deduce from $x_7 x_{13} x_{22} = -1$ and $x_{22} x_{44} x_{60} = +1$ that $x_7 x_{13} x_{44} x_{60} = -1$ etc.. but not able to draw all possible logical inferences and hence figure out that there is no solution x to these equations. Thus, just in the case of Mickey, we design the pseudo-distribution to be as random as possible subject to being consistent with the deductions Donald is able to make, in other words we follow Einstein's maxim that

Pseudo-distributions should be as random as possible but not randomer

More concretely we will assume that Donald's knowledge only applies to terms of the form $\prod_{i \in S} x_i$ for $|S| \leq s$, and that given subsets S, T such that $|S|, |T| \leq s$, if $U = S \oplus T$ has size at most s, then he can deduce that $x_U = x_S x_T$, but these are all the deductions he can make. Specifically we define the pseudo-distribution $\{x\}$ by the following iterative process:

- Input to the process: graph $G = ([m] \cup [n], E)$ and string $a \in \{\pm 1\}^m$.
- For every $\ell \in [m]$, define $\tilde{\mathbb{E}}\, x_{\Gamma(\ell)} = a_\ell$.
- Apply the following rule until we can't apply it any further: for every subsets S, T of size at most s such that $\tilde{\mathbb{E}}\, x_S$ and $\tilde{\mathbb{E}}\, x_T$ are defined, $|S \oplus T| \leq s$, define $\tilde{\mathbb{E}}\, x_{S \oplus T} = \tilde{\mathbb{E}}\, x_S \tilde{\mathbb{E}}\, x_T$. (If $\tilde{\mathbb{E}}\, x_{S \otimes T}$ was already defined before with a different value, the process fails and halts.)
- When done, for every nonempty $|S| \leq s$ such that $\tilde{\mathbb{E}}\, x_S$ is undefined, define $\tilde{\mathbb{E}}\, x_S = 0$. For every monomoial $m(x)$ of degree at most s that contains square terms define $\tilde{\mathbb{E}}\, m(x) = \tilde{\mathbb{E}}\, x_S$ where S is the set of i's such that x_i appears with an odd degree in $m(x)$.

We need to prove that $\{x\}$ defined above is a valid pseudo-distribution. For starters, we need to verify that it never halts:

Lemma 3.5. *If the G is a $(10s, 1.7)$ expander then the process above never halts.*

Before proving the lemma, lets see why it implies that $\{x\}$ is a valid pseudo-distribution for degree at most $s/2$. First note (**Exercise 3.3:** check this) that by definition, we get that for every polynomial p of degree at most $s-3$ and every $S=\Gamma(\ell)$, $\tilde{\mathbb{E}}\, p(x)x_S = a_\ell \tilde{\mathbb{E}}\, p(x)$. Thus we need to prove that for every polynomial q of degree at most $s/2$, $\tilde{\mathbb{E}}\, q^2 \geq 0$. **Exercise 3.4:** Prove that it suffices to do so for the case that q is *multilinear* namely $q(x) = \sum_{|S|\leq s/2} \alpha_S x_S$ for some real numbers $\{\alpha_S\}_{|S|\leq s/2}$.

For two subsets S,T of size at most $s/2$, we say that $S \equiv T$ if $\tilde{\mathbb{E}}\, x_{S\oplus T} \neq 0$. Note that this is indeed an equivalence relation— it is symmetric, reflexive (since $\tilde{\mathbb{E}}\, x_\emptyset = \tilde{\mathbb{E}}\, 1 = 1$), and transitive, since if $S\equiv T$ and $T \equiv U$ then $\tilde{\mathbb{E}}\, x_{S\oplus U} = \tilde{\mathbb{E}}\, x_{(S\oplus T)\oplus(T\oplus U} = \tilde{\mathbb{E}}\, x_{S\oplus T}\, \tilde{\mathbb{E}}\, x_{T\oplus U}$. Separate the monomials of q into the equivalence classes and so write $q = \sum_{i=1}^m q_i$ where all monomials in q_i belong to a particular equivalence class. Note that if S and T are not equivalent then $\tilde{\mathbb{E}}\, x_S x_T = \tilde{\mathbb{E}}\, x_{S\oplus T}$ equals zero, and hence

$$\tilde{\mathbb{E}}\, q^2 = \tilde{\mathbb{E}}(\sum_i q_i)^2 = \sum \tilde{\mathbb{E}}\, q_i^2$$

Thus it suffices to show that $\tilde{\mathbb{E}}\, q^2 \geq 0$ where $q(x) = \sum_{S\in\mathcal{C}} \alpha_S x_S$ where $\mathcal{C}$ is some fixed equivalence class. Let S_0 be a member of that class. Thus for every $S,T\in\mathcal{C}$, $\tilde{\mathbb{E}}\, x_{S\oplus S_0} \neq 0$ and $\tilde{\mathbb{E}}\, x_{T\oplus S_0} \neq 0$ and so (by our rule) $\tilde{\mathbb{E}}\, x_{S\oplus T} = \tilde{\mathbb{E}}\, x_{S\oplus S_0}\, \tilde{\mathbb{E}}\, x_{T\oplus S_0}$. Therefore,

$$\tilde{\mathbb{E}}\, q^2 = \sum_{S,T\in\mathcal{C}} \alpha_S\alpha_T\, \tilde{\mathbb{E}}\, x_S x_T = \sum_{S,T\in\mathcal{C}} \alpha_S\alpha_T (\tilde{\mathbb{E}}\, x_{S\oplus S_0})(\tilde{\mathbb{E}}\, x_{T\oplus S_0}) = \Big(\sum \alpha_S\, \tilde{\mathbb{E}}\, x_{S\oplus S_0}\Big)^2 \geq 0$$

□

3.1.3 Proof of Lemma 3.5

[BOAZ: this proof is copy pasted from previous lecture notes and so uses somewhat inconsistent notation. On a very high level, the proof goes as follows: for a set E of equations, let $\underline{\Gamma}(E)$ denote $\oplus_{\ell\in E}\Gamma(\ell)$, i.e. $\overline{\Gamma}(E)$ is the set of variables that appear an odd number of times in the equations in E. If we have a derivation $U_1,\ldots,U_t$ of sets of size at most d such that either U_i is the variables of an equation (i.e., $U_i = \Gamma(\ell)$) or the value $\tilde{\mathbb{E}}\, U_i$ is always derived from prior values $\tilde{\mathbb{E}}\, U_j$, $\tilde{\mathbb{E}}\, U_k$ for $j,k<i$, then we can keep track of the sets of equations E_i that correspond to every U_i (e.g., if $U_i = \Gamma(\ell)$ then $E_i = \{\ell\}$ and in the other case $E_i = E_j \oplus E_k$. Note that $U_i = \underline{\Gamma}(E_i)$ and the value derived to $\tilde{\mathbb{E}}\, U_i$ is equal to $\prod_{\ell\in E_i} a_{\Gamma(\ell)}$. We then use the expansion property to argue that for every i, $|E_i| \leq 10s$, and $|\underline{\Gamma}(E)| \geq |E_i|/10$ for all E with $|E|\leq 100s$. But this implies that U_i uniquely determines E_i, since if we had $U_i = U_j$ but $E_i \neq E_j$ then we would get that $\emptyset = U_i \oplus U_j = \underline{\Gamma}(E_i) \oplus \underline{\Gamma}(E_j) = \underline{\Gamma}(E_i\oplus E_j)$. But since $|\underline{\Gamma}(E_i\oplus E_j)| \geq |E_i \oplus E_j|/10$, if $E_i \neq E_j$ then the set $\underline{E_i \oplus E_j}$ can't be empty. Thus the value for a set U that can be derived in some way is always uniquely defined as $\oplus_{\ell\in E} x_{\Gamma(\ell)}$ no matter how we derived it.]

Observe that if there is a derivation that gives different values to some monomial U of degree at most d, then there is also another derivation that shows $\tilde{\mathbb{E}}\big[x_\phi\big] = -1$. We will show that this leads to a contradiction. Let the derivation of $\tilde{\mathbb{E}}\big[x_\phi\big] = -1$ be described by a sequence of set $U_1, U_2, \cdots U_t$ with $U_t = \phi$. Notice that each U_i in this derivation is either some constraint in $\mathcal{E}$ or a product of some of the constraints from $\mathcal{E}$. Let $E_1, E_2 \cdots E_m$ be the sets corresponding to the equations in $\mathcal{E}$ and $\sigma_1, \sigma_2, \cdots \sigma_m$ be the corresponding values. For every U_i that is derived, we see that there is some $S_i \subseteq L$ for which $U_i = \cup_{\ell in \mathbb{S}_i} E_\ell$.

Now, we will see that the assumption of there being two different derivations for $\tilde{\mathbb{E}}\big[x_\phi\big]$ with opposite values leads to the conclusion that the above algorithmic process must assign some value to $\tilde{\mathbb{E}}\left[x_U\right]$ for some $|U| > d$ which is not possible. This conflicts with the assumption that there are 2 different derivations for the same set with opposite values. The plan is roughly the following. First we observe that every derived monomial with short derivation spans many variables (by expansion property) (has large degree). The idea is to show that two derivations with opposite values means that there is some monomial with degree greater than d that the algorithm defines.

Claim 3.6. *For every $S \subseteq L$ with $|S| \leq 100d$, $| \oplus_{\ell \in S} E_\ell| \geq |S|/10$.*

Proof. Suppose not. Let $T = \oplus_{\ell \in S} E_\ell$. Observe that the monomial in T may not all of the variables from $\Gamma(S)$ (that is, those which appear in $\Gamma(S) \setminus T$) . So, these variable nodes must have at least 2 neighbors in S as they need to cancel out. This gives (by expansion property)

$$\begin{aligned} \#\text{edges leaving } |S| &\geq \#\text{ edges entering } S \text{ from the ``omitted'' variables in } \Gamma(S) \setminus T \\ \implies 3|S| &\geq 2(1.7|S| - |T|) \\ \implies |T| &\geq 0.2|S| \end{aligned}$$

□

As observed already the above claim asserts that whatever you derive by XORing a small subset of clauses on the left gives rise to a monomial with decent degree.

Corollary 3.7. *Notice that this means that every set S_i in the derivation must have size no larger than $10d$.*

Proof. A quick proof. Suppose not – then the first such violating set $S_i = S_j \oplus S_k$ where S_j and S_k both describe short derivations and S_i is a bigger derivation with length in the interval $(10d, 20d]$. And by the above observation, this would mean $\oplus_{\ell \in S_i} E_\ell > d$. □

Thus, the existence of 2 sets with different values implies that there is a set $S = S_t$ with size at most $10d$ such that $\oplus_{\ell \in S} E_\ell = \phi$ contradicting the observation above.

For $P : \{\pm 1\}^k \to \pm$, we define a Max-P instance ψ to be a collection of equations of the form $P(a_{i_1} x_{i_1}, \ldots, a_{i_k} x_{i_k}) = a$, and the goal is again to find an assignment x satisfying as many of the equations as possible. We say that P is a *nice subspace predicate* if there is some subspace $V \subseteq GF(2)^k$ such that $P = 1_V$ (using the identification $GF(2) \leftrightarrow \{\pm 1\}$ using the map $n \leftrightarrow (-1)^b$) and such that every $u \in V^\perp$ satisfies $|u| \geq 3$.

As was noted by Tulsiani, Chan, the proof of Theorem 3.1 generalizes to the following statement: **Exercise 3.5:** Prove the following theorem.

Theorem 3.8. *For every nice subspace predicate $P = 1_V$, constant $\epsilon > 0$ and large enough n, there exists an instance ψ of Max-P over n variables such that:*

- *Every assignment $x \in \{\pm 1\}^n$ satisfies at most $|V|/2^k + \epsilon$ fraction of the equations of ψ.*
- *There exists a degree $\Omega(n)$ pseudo distribution $\{x\}$ that is consistent with the constraints $\{x_i^2 = 1\}$ for all $i \in [n]$ and the constraint corresponding to every equation in ψ.*

Moreover, there is $m = O(n)$ such that with constant positive probability, a random ψ with m equations will satisfy the properties above.

3.1.4 Proof of Lemmas 3.3 and 3.2

[BOAZ — these two proofs are also at the moment copy-pasted from previous notes and use inconsistent notation.]

Proof. We will prove the result for the graphs G_ϕ obtained from a random k-xor instance with n variables and γn constraints for some large constant Γ. In fact, the same proof holds for a random instance of any *Max-K-CSP* over any alphabet. Let us begin by trying to understand what is the probability that a set fails to have large expansion. Let us say that in fact, there is a small set of variables $T \subseteq R$ that is the vertex boundary of some set S of size $s = \epsilon n$ where ϵ is some constant to be determined later. Let $|T| \le cs = (k-1-\delta)s$ denote the size of this small set of variables that appear in this set S of the random k-xor instance (where $\delta \in (0, \frac{1}{2})$). Observe that the probability p that this indeed happens – that a set S of size ϵn happens to have small vertex boundary can be upper bounded by

$$p \le \binom{n}{cs} \cdot \binom{\binom{cs}{k}}{s} \cdot s!\binom{\gamma n}{s} \cdot \binom{n}{k}^{-s}$$

Here,

- $\binom{n}{cs}$ denotes which cs variables to use.
- $\binom{\binom{cs}{k}}{s}$ denotes the number of ways cs variables can be put together to get s clauses each of arity k.
- $s!\binom{\gamma n}{s}$ counts the choices for where to put such clauses in our ordered sequence of γn clauses.
- And the last term, $\binom{n}{k}^{-s}$ denotes the probability that the clauses were generated using the described method.

Now, using Stirling's which says $\frac{a}{b}^b \le \frac{a}{b} \le \frac{ae}{b}^b$ and $s! \le s^s$, we obtain by collecting terms we get

$$\begin{aligned} p &\le \left(\frac{s}{n}\right)^{2\delta \cdot s/2} \left(e^{2k+1-\delta} k^{1+\delta} \gamma\right)^s \\ &\le \left(\frac{s}{n}\right)^{\delta s} \cdot \left(\gamma^5\right)^s \\ &= \left(\frac{s\gamma^{5/\delta}}{n}\right)^{\delta s} \end{aligned}$$

We need to show that the probability that any set of at most $s \le \epsilon n$ constraints contains less than cs variables is $o(1)$. To do this, consider the following.

$$\begin{aligned} \sum_{s=1}^{\epsilon n} \left(\frac{s\gamma^{5/\delta}}{n}\right)^{\delta s} &= \sum_{s=1}^{\ln^2 n} \left(\frac{s\gamma^{5/\delta}}{n}\right)^{\delta s} + \sum_{s=\ln^2 n+1}^{\epsilon n} \left(\frac{s\gamma^{5/\delta}}{n}\right)^{\delta s} \\ &\le O\left(\frac{\gamma^5}{n^\delta} \cdot \ln^2 n\right) + O\left(\epsilon \gamma^{5/\delta}\right)^{\delta \ln^2 n} \end{aligned}$$

Now, we are almost done. Observe that the first term is clearly $o(1)$ and that the second term is $o(1)$ for $\epsilon = O(\frac{1}{100\gamma^{5/\delta}})$. This gives us that indeed small sets of size ϵn fail to have a large boundary with large probability which is what we wanted. □

3.2 SOS Lower bounds for planted clique

The planted clique problem is one of the most classical computational problems, whose roots come from a 1976 question of Karp of whether we can find the largest clique in a random graph. In the early 1990's, Jerrum and Kucera suggested the easier *planted* model, whereby the goal is to find a ω-clique that has been added to a random graph. Note that if $\omega \gg \log n$ then this would be the unique maximum ω-clique in the graph. (**Exercise 3.6:** prove that with probability at least 0.99 a random $G(n, 1/2)$ graph has the maximum clique size of at most $c \log n$ for some constant c; how small can you make c? .) The larger ω is, the easier this problem. Another variant which seems to have equivalent difficulty is to distinguish between a random graph and a graph to which a random ω-clique was added. One easy bound on ω for this problem arises from the following exercises:

Exercise 3.7: Let A be the adjacency matrix of a random $G(n, 1/2)$ graph and $B = 2A - J$ where J is the all 1's matrix. **(1)** Prove that for every t, $\mathbb{E}\,\mathrm{Tr}(B^t) \leq 2^{O(t)} n^{t/2}$. **(2)** Conclude that with probability at least 0.99, $\|B\| \leq O(\sqrt{n})$.

Exercise 3.8: Let A be the adjacency matrix of a n-vertex graph with average degree $n/2$ that contains a ω clique and $B = 2A - J$. Prove that $\|B\| \geq \Omega(\omega)$.

Thus we can easily distinguish between a random $G(n, 1/2)$ graph and an n vertex graph containing a $\omega \gg \sqrt{n}$ clique, and in fact these ideas can be used to actually find the clique in the latter case (**Exercise 3.9:** Show this; see footnote for hint[4]). Many people have thought of improving this $\sqrt{n}$ bound but with no success, often proving that certain methods *won't* work. In fact by now the difficulty of this question has been conjectured in several works that have connected this problem to questions in machine learning, compressed sensing, computing equilibrium and more.

The algorithm that works for $\omega \sim \sqrt{n}$ can be thought of as an instantiation of the degree 2 SOS algorithm and thus we come again to the question of whether degree $d > 2$ SOS can do better than degree 2. As in the case of the Unique Games, Small-Set Expansion, Max-Cut, Cheeger etc.., the answer is that we don't know. But, given that this is an average case problem (such as the random 3XOR problem discussed above), one could perhaps hope that we will be able to prove some SOS lower bounds in this case. Indeed last year Meka and Wigderson claimed that for every constant d there is some $\epsilon > 0$ such that the degree d SOS algorithm cannot certify that a random graph doesn't contain an $\epsilon\sqrt{n}$ clique. However (as we will see) their proof was flawed. Nevertheless in a very new result, Meka, Potechin and Wigderson were able to prove a weaker result. Namely that the degree d SOS cannot certify that a random graph doesn't contain an $\tilde{\Omega}(n^{1/d})$ clique. We will see a (slight weakening of a) special case of their result, namely

Theorem 3.9. *Let $G = G(n, 1/2)$ be a random graph. With probability at least* 0.9*, there exists a degree* 4 *pseudo-distribution $\{x\}$ over $\mathbb{R}^n$ satisfying $\{x_i^2 = x_i\}$ for all i, $\{x_i x_j = 0\}$ for all i and j that are not neighbors in G, and*

$$\tilde{\mathbb{E}} \sum x_i \geq \Omega(n^{1/8})$$

3.2.1 Proof of Theorem 3.9

Once again, to construct a pseudo distribution, we think of a computationally bounded observer that is told that there is a planted clique in the graph, and needs to form beliefs about what is the probability that some set S with $|S| \leq 4$ is contained in the clique (or equivalently, what should be the expectation $\tilde{\mathbb{E}}\, x_S$). Clearly if S is not itself a clique, then this probability is zero.

[4] **Hint:** Show that we can get a set S with large correlation with the clique by looking at the largest eigenvector of B, and then show that we can use to actually find the clique by looking at the set of vertices that have very large degree into S.

Otherwise, since both clique and surrounding graph are random, we would expect the probability to be the roughly the same for every clique. This motivates defining our pseudo-distribution: for every set S of size at most 4, if S is not a clique then $\tilde{\mathbb{E}}\, x_S = 0$, and otherwise $\tilde{\mathbb{E}}\, x_S = 2^{\binom{|S|}{2}}(\omega/n)^{|S|}$ (where $\binom{1}{2} = 0$).[5] Analogous to what we did for 3XOR, we use the constraints that $x_i^2 = x_i$ to reduce every monomial $m(x)$ to a multilinear monomial of the form x_S. Note that we get that $\tilde{\mathbb{E}} \sum x_i = n(\omega/n) = \omega$.

It turns out that this simplistic pseudo-distribution is already sufficient to prove a weak lower bound on the clique size

Lemma 3.10. *If $\omega \ll n^{1/8}$ then the pseudo-distribution above is a valid degree 4 pseudo-distribution satisfying the conditions of Theorem 3.9*

On the other hand, we (and Meka-Wigderson) have violated Einstein's maxim and made this pseudo-distribution "randomer than possible" if we want to reach the $\omega \sim \sqrt{n}$ bound.

Lemma 3.11. *If $\omega \gg n^{1/3}$ then there exists a quadratic polynomial Q such that $\tilde{\mathbb{E}}\, Q^2 < 0$*

3.2.2 Proof of Lemma 3.10

Let $M_{a,b,c,d} = \tilde{\mathbb{E}}\, x_a x_b x_c x_d$. We need to prove that the matrix M is positive semidefinite. Let us focus our attention to the rows $\{a,b\}$ of M where $a \neq b$ and columns $\{c,d\}$ where $c \neq d$ (this turns out to be the crux of the proof). Since the entire row corresponding to $\{a,b\}$ will be zero if (a,b) is not an edge of$G$, we can think of M as an $E \times E$ matrix where E is the edge set of G. Recall that for every a,b,c,d $M_{a,b,c,d}$ depends solely on $|\{a,b,c,d\}$. Thus we can write $M = M^2 + M^3 + M^4$ where $M^s_{a,b,c,d} = M_{a,b,c,d}$ if $|\{a,b,c,d\}| = s$ and equals zero otherwise. Let us ignore M^3 for now (this is not the main issue) and so focus on proving that $M^2 + M^4$ is positive semidefinite. Note that M^2 simply contains all diagonal elements and each has magnitude $\tilde{\mathbb{E}}\, x_a x_b = 2\omega^2/n^2$. Therefore, we can scale by this number and reduce showing that $M^2 + M^4$ is psd to showing that $I + M'$ is p.s.d where I is the $E \times E$ identity, and M' is defined as follows:

$$M'_{a,b,c,d} = \begin{cases} 0 & |\{a,b,c,d\}| \neq 4 \\ 3\omega^2/n^2(1-1/16) & \{a,b,c,d\} \text{ is 4-clique} \\ -(3/16)\omega^2/n^2 & \{a,b,c,d\} \text{ is not a 4-clique} \end{cases}$$

Note that M' is simply M^4, scaled by $n^2/(2\omega^2)$ and subtracting from each entry $\{a,b,c,d\}$ with $|\{a,b,c,d\}|$ a constant so that the expected entry of M^4 is zero. The reason that this suffices follows from the following exercise:

Exercise 3.10: Let E be the $n^2 \times n^2$ *expectation* matrix of M. Namely, for every a,b,c,d, $E_{a,b,c,d}$ is the expected value of $M_{a,b,c,d}$ which is $2^{-\binom{s}{2}} 2^{\binom{s}{2}}(\omega/n)^s$ where $s = |\{a,b,c,d\}|$. Prove that E is p.s.d and its smallest nonzero eigenvalue is $\Omega(\omega^2/n^2)$.

This exercise is actually a special case follows from the theory of *Johnson Association Schemes* that is used in proving more general bounds. In particular the following is true (and may be useful for the previous exercise):

[5] This is because conditioned on S being a clique, the probability that it is contained in the random ω-clique would be roughly $\binom{w}{|S|}$ divided by the the number of $|S|$-clique in the graph which is $\binom{n}{|S|}2^{-\binom{|S|}{2}}$; we get the above probability using the approximation $\binom{n}{k} \sim \left(\frac{en}{k}\right)^k$.

Exercise 3.11: Let $\ell \in \mathbb{N}$ and J be a matrix indexed by all subsets $S \subseteq [n]$ of size at most s such that $J_{S,T} = \binom{|S \cap T|}{\ell}$. Then J is psd. (If you get stuck, take a look at the Meka-Wigderson paper.)

The Frobenius norm squared of M', namely $\sum_{a,b,c,d} (M'_{a,b,c,d})^2$ equals $O(n^4\omega^4/n^4) = O(\omega^4)$. Note that the Frobenius norm squared is the sum of the eigenvalues squared, and thus if M' was "generic" or "pseudorandom" in the sense that it would have $\Theta(n^2)$ eigenvalues with roughly the same magnitude λ, then λ will satisfy $n^2\lambda^2 = O(\omega^4)$ or $\lambda \leq O(\omega^2/n)$. Thus in this case, as long as $\omega \ll \sqrt{n}$ the matrix $I + M'$ (and hence M) will be positive semidefinite. A priori you might hope that, sicne M' arises from a random graph then it will in fact be sufficiently "psuedorandom" to satisfy this, but as we will see in Lemma 3.11, this turns out to be false. Nevertheless we are able to prove the following claim:

Claim: w.h.p. $\text{Tr}(M'^4) \leq O(\omega^8/n)$

The theorem follows from the claim since we get that $\|M'\| \leq \text{Tr}(M'^4)^{1/4} \leq \omega^2/n^{1/4}$. Hence if $\omega^2 \ll n^{1/4}$ (or $\omega^8 \ll n$) then $I + M'$ will be psd.

Proof of Claim: By Markov we simply need to show that $\mathbb{E}\,\text{Tr}(M'^4) \leq O(\omega^8/n)$. (Note that this is an actual expectation, taken over the random choice of the graph G; since the set E of edges depends on this randomness, it might be more convenient to think of M' as an $n^2 \times n^2$ matrix for this argument.) The expectation of the trace is the sum over all 4-tuples of edges e_1, e_2, e_3, e_4 of

$$\mathbb{E}\, M'_{e_1,e_2} M'_{e_2,e_3} M'_{e_3,e_4} M'_{e_4,e_1} \tag{3.1}$$

Now if e_1, e_2, e_3, e_4 are all disjoint, then, conditioned on e_1, e_2, e_3, e_4 being edges, the events "$e_1 \cup e_2$ is a 4-clique", "$e_2 \cup e_4$ is a 4-clique", etc.. are independent. Thus we get that (3.1) equals

$$\mathbb{E}\, M'_{e_1,e_2}\, \mathbb{E}\, M'_{e_2,e_3}\, \mathbb{E}\, M'_{e_3,e_4}\, \mathbb{E}\, M'_{e_4,e_1}$$

but $\mathbb{E}\, M'_{e,f} = 0$ for every pair of edges e, f by the construction of M'. Therefore, the contribution to the trace must come from 4-tuples of edges that are not all disjoint. Since each such 4-tuple involves at most 7 vertices, there are $O(n^7)$ such tuples, and since every entry as magnitude $O(\omega^2/n^2)$, the expectation of the trace is at most

$$O(n^7) \cdot O(\omega^2/n^2)^4 = O(\omega^8/n)$$

Exercise 3.12: Prove that in fact $\mathbb{E}\,\text{Tr}(M'^4) \leq O(\omega^8/n^2)$, hence concluding that the pseudo-distribution is psd as long as $\omega \ll n^{1/4}$. See footnote for hint[6]

3.2.3 Proof of Lemma 3.11

Intuitively, one may hope that the pseudo-distribution above remains valid even for a larger value of ω, as long as $\omega \ll \sqrt{n}$. However, Lemma 3.11 shows this is not the case. To understand the reason, let's see that there is a simple observation available to the computationally-bounded Donald that would yield different results in this pseudo-distribution than it would have if it was truly a distribution over planted cliques. Specifically, for a n-vertex graph G, let $r_1, \ldots, r_n \in \mathbb{R}^n$ be the vectors such that

$$r_i(j) = \begin{cases} +1 & (i,j) \text{ is an edge} \\ 0 & i = j \\ -1 & \text{otherwise} \end{cases}$$

[6] **Hint:**Show that in fact the only nonzero contribution to the trace come from 4-tuples of edges e_1, e_2, e_3, e_4 such that $|e_1 \cup e_2 \cup e_3 \cup e_4| \leq 6$.

Let $P(x)$ be the polynomial $\sum \langle r_i, x\rangle^4$. Note the following facts about $P(x)$: (**Exercise 3.13:** verify those)

1. For every fixed $x \in \mathbb{R}^n$, if we choose the graph at random then $\mathbb{E}\, P(x) = O(n\|x\|^4)$.
2. If x is the 0/1 characteristic vector of an ω-clique (and hence $\|x\|^2 = \omega$) then $P(x) \geq \omega(\omega-1)^4 = \Omega(\omega^5)$.

Hence when $\omega^5 \gg n\omega^2$ (or $\omega \gg n^{1/3}$), $P(x)$ will distinguish between a "typical" vector x and a vector x that is the characteristic vector of an ω-clique. We claim that in the distribution $\{x\}$ above, $\tilde{\mathbb{E}}\, P(x)$ actually behaves as if x was "typical" and thus gives it too low a value:

Claim: With high probability $\tilde{\mathbb{E}}\, P(x) \leq O(n\omega^2)$.

Proof: Using Markov, it suffices to prove that $\mathbb{E}_G \tilde{\mathbb{E}}\, P(x) \leq O(n\omega^2)$ where the expectation is taken over the random choices in making the graph and the pseudo-expectation is as defined above. Lets open up the definition of $P(x)$ and write

$$\mathbb{E}\, \tilde{\mathbb{E}}\, P(x) = \sum_i \sum_{a,b,c,d \in [n]\setminus\{i\}} \mathbb{E}\, r_i(a)r_i(b)r_i(c)r_i(d)\, \tilde{\mathbb{E}}\, x_a x_b x_c x_d$$

(using the fact that $r_i(i) = 0$ for all i). Fix some $i \in [n]$, and now suppose that we fix the random choices of all neighbors in the graph except the neighbors of i. This means that $\tilde{\mathbb{E}}\, x_a x_b x_c x_d$ is determined for every $\{a,b,c,d\} \subseteq [n] \setminus \{i\}$ and the random $\{\pm 1\}$ variables $r_i(a), r_i(b), r_i(c), r_i(d)$ are independent of this choice. Thus we can write for every i and $a,b,c,d \in [n] \setminus \{i\}$, Therefore

$$\mathbb{E}\, r_i(a)r_i(b)r_i(c)r_i(d)\, \tilde{\mathbb{E}}\, x_a x_b x_c x_d = \tilde{\mathbb{E}}\, x_a x_b x_c x_d\, \mathbb{E}\, r_i(a)r_i(b)r_i(c)r_i(d)$$

Note that $\mathbb{E}\, r_i(a)r_i(b)r_i(c)r_i(d) = 0$ unless $a = b = c = d$ or $|\{a,b,c,d\}| = 2$. In the first case $\tilde{\mathbb{E}}\, x_a x_b x_c x_d = \omega/n$ and in the second case it equals $O(\omega^2/n^2)$. Hence for every i

$$\sum_{a,b,c,d \in [n]\setminus\{i\}} \mathbb{E}\, r_i(a)r_i(b)r_i(c)r_i(d)\, \tilde{\mathbb{E}}\, x_a x_b x_c x_d \leq n\omega/n + O(n^2\omega^2/n^2) = O(\omega^2)$$

which summing over all i implies that

$$\mathbb{E}\, \tilde{\mathbb{E}}\, P(x) = O(n\omega^2)$$

The above shows that $\{x\}$ is already very fishy as a pseudo-distribution, since (in the $\omega \gg n^{1/3}$ range) it gives $P(x)$ a value that is far too low to be consistent with being a distribution over ω-cliques. But we still haven't shown that it does in fact violate the constraints of being a valid pseudo-distribution. We now show this

Lemma 3.12 (Lemma 3.11,restated). *If $\omega \gg n^{1/3}$ then there exists a quadratic polynomial Q such that $\tilde{\mathbb{E}}\, Q^2 < 0$*

Proof. We let

$$Q(x) = (c(n/\omega)x_1 - n\langle r_1, x\rangle^2)^2$$

for some large enough constant c to be determined later. Now

$$\tilde{\mathbb{E}}\, Q^2 = \tfrac{c^2n^2}{\omega^2}\, \tilde{\mathbb{E}}\, x_1^2 + \tilde{\mathbb{E}}\langle r_1, x\rangle^4 - \tfrac{2cn}{\omega}\, \tilde{\mathbb{E}}\langle r_1, x\rangle^2 x_1^2 \tag{3.2}$$

Let us compute each term of (3.2). First, clearly

$$\tilde{\mathbb{E}}\, x_1^2 = \tilde{\mathbb{E}}\, x_1 = \omega/n$$

and hence the first term equals $c^2 n/\omega$. We just computed the second term above as $O(\omega^2)$. To compute the third term, note that

$$\tilde{\mathbb{E}}\langle r_1, x\rangle^2 x_1^2 = \sum_{a,b\in[2..n]} r_1(a) r_1(b)\, \tilde{\mathbb{E}}\, x_a x_b x_1^2$$

which simply counts the number of triangles in the graph of the form $\{1, a, b\}$ (which is $\Omega(n^2)$) multiplied by $\Omega(\omega^3/n^3)$. (Indeed, note that if $\{1, a, b\}$ is a triangle then $r_1(a) = r_1(b) = +1$, and otherwise $\tilde{\mathbb{E}}\, x_a x_b x_1^2 = \tilde{\mathbb{E}}\, x_a x_b x_1 = 0$.) Thus the third term is $-\Omega(c(n/\omega)n^2(\omega^3/n^3) = -\Omega(c\omega^2)$. We see that if we want this expression to be negative then we need the third term to dominate the other two, and hence we need to satisfy $c \gg 1$ and $c\omega^2 \gg c^2 n/\omega$, or $\omega^3 \gg cn$. Thus if $\omega \gg n^{1/3}$ then we can find a constant c that would make $\tilde{\mathbb{E}}\, Q^2 < 0$. □

3.3 Knapsack lower bound

One very nice SOS lower bound we did not show is the following theorem of Grigoriev

Theorem 3.13 (Grigoriev). *For every n there is degree $\Omega(n)$ pseudo-distribution $\{x\}$ satisfying the constraints $\{x_i^2 = x_i\}$ and $\{\sum x_i = n/2\}$.*

Note that if n is odd, then there cannot be an actual distribution satisfying these constraints. The arxiv paper of Meka and Wigderson, despite having a fatal flaw, is still very much worth reading, and in particular is a good source for understanding the proof of this result.

Chapter 4

SOS Upper Bounds: Planted sparse vector and dictionary learning

Based on scribed (and greatly expanded) notes by Samuel Hopkins and Jerry Li

Part I: Finding Sparse Planted Vector

4.1 Introduction

In this lecture we will see how the SOS algorithm can be used to solve the following problem: Suppose that $V \subseteq R^n$ is a random k-dimensional linear subspace in which someone "planted" a sparse vector v_0. *Sparse* here means that v_0 has few nonzero coordinates in the standard basis—perhaps ϵn. The goal is to recover v_0 given an arbitrary basis of V. We give a more formal description below.

The problem itself is somewhat natural, and can be thought of as an average-case real (as opposed to finite field) version of the "shortest codeword" or "lattice shortest vector" problem. This also turns out to be related (at least in terms of techniques) to problems in unsupervised learning such as dictionary learning / sparse coding.

There is a related problem, often called "compressed sensing" or "sparse recovery" in which we are given an *affine* subspace A of the form $v_0 + V$, where v_0 is again sparse and V is an (essentially) random linear subspace, and the goal is again to recover v_0. Note that typically this problem is described somewhat differently: we have an $m \times n$ matrix A, often chosen at random, and we get the value $y = Av_0$. This determines the $k = n - m$ dimensional affine subspace $v_0 + \mathrm{Ker}(A)$, and we need to recover v_0.

One difference between the problems is parameters (we will think of $k \ll n$, while in sparse recovery typically $k \sim n - o(n)$), but another more fundamental difference is that a linear subspace always has the all-zeroes vector in it, and hence, in contrast to the affine case, v_0 is *not* the sparsest vector in the subspace (only the sparsest nonzero one).

This complicates matters, as the algorithm of choice for sparse recovery is L1 minimization:

find $v \in A$ that minimizes $\|v\|_1 = \sum_{i=1}^n |v_i|$. This can be done by solving the linear program:

$$\begin{aligned} & \min \sum_{i=1}^n x_i \\ \text{subject to} \quad & x_i \geq v_i \\ & x_i \geq -v_i \\ & v \in A \end{aligned}$$

But of course if A were a linear subspace but not affine, then this would return the all-zero vector. (Though see below on variants that do make sense for the planted vector problem.)

4.1.1 Formal description of problem

We assume that $v_1, \ldots, v_k \in \mathbb{R}^n$ are chosen randomly as standard Gaussian vectors (i.e. with i.i.d. entries drawn from $N(0,1)$), and v_0 is some arbitrary unit vector with at most ϵn nonzero coordinates. We are given an arbitrary basis B for $\mathrm{Span}\{v_0, v_1, \ldots, v_k\}$. The goal is to recover v_0.

For this lecture, this means recovering a unit vector v such that $\langle v, v_0\rangle^2 \geq 0.99$ (though see the paper [BKS14] for recovery with arbitrary accuracy). For simplicity let's also assume that v_0 is orthogonal to $v_1, \ldots, v_k$. (This is not really needed but helps simplify some minor calculations.)

4.1.2 Ratios of Norms

Rather than trying to directly trying to find a sparse vector, we will define some smoother *proxy* for sparsity, that is some polynomial $P(\cdot)$ so that $P(v)$ is larger for sparse vectors than for small ones. Then we will look for a vector v in the subspace that maximizes $P(v)$ (subject to some normalization) and hope that (a) we can efficiently do this and (b) that the answer is v_0. This makes the problem more amenable for the SOS algorithm and also makes for a more robust notion, allowing for some noise in v_0 (and lets us not worry about issues of numerical accuracy).

So, we want some function that will favor vectors that are "spikier" as opposed to "smoother". We use the observation that taking high powers amplifies "spikes". Specifically, we note that if $q > p$ a sparse/spiky vector v would have a larger ratio of $\|v\|_q / \|v\|_p$ than a dense/smooth one. Indeed, compare the all 1's vector $\vec{1}$ with the vector 1_S for a set S of size ϵn. $\|\vec{1}\|_q / \|\vec{1}\|_p = n^{1/q - 1/p}$ while $\|1_S\|_q / \|1_S\|_p = (\epsilon n)^{1/q-1/p}$ which means that if $q > p$, the latter ratio is larger than the former by some power of $1/\epsilon$. Moreover, an application of Hölder's inequality reveals that if v is ϵn-sparse then its q vs p norm ratio can only be higher than this.

Claim 4.1. *If $v \in \mathbb{R}^n$ has at most ϵn nonzero coordinates, then*

$$(\mathbb{E}_i\big[v(i)^q\big])^{1/q} \geq \epsilon^{1/q-1/p} (\mathbb{E}_i\big[v(i)^p\big])^{1/p}.$$

Proof. Let $1_{|v|>0}$ be the vector which is 1 if $|v(i)| > 0$ and 0 otherwise. Let $w \in \mathbb{R}^n$ be given by $w = 1_{|v|>0}/n^{1-q/p}$. Then by Hölder's inequality,

$$\begin{aligned} (\mathbb{E}_i\big[v(i)^p\big]) &= \sum_i w(i) \frac{v(i)^p}{n^{q/p}} \\ &\leq (\sum_i w(i)^{1/(1-p/q)})^{1-p/q} (\sum_i v(i)^q/n)^{p/q} \\ &= \epsilon^{1-p/q} (\mathbb{E}_i\big[v(i)^q\big])^{p/q}. \end{aligned}$$

Rearranging gives the result. □

How good a proxy for sparsity is this? We know that vectors which are actually sparse “look sparse” in the ratio-of-norms sense, but what about the other way around—could the ratio of norms be fooled by vectors which are not actually sparse? The answer is yes. For example, if $q = \infty$ and $p = 1$, the vector which has a 1 in one coordinate and ϵ in the other coordinates looks like an ϵ-sparse (or more accurately $\epsilon - 1/n$-sparse) vector as far as the ∞ versus 1 norm ratio is concerned, but in the strict ℓ_0-sense is actually maximally non-sparse.

However, as the gap between p and q shrinks, a random subspace becomes less and less likely to contain these kind of “cheating vectors” that are not sparse but look sparse when comparing ℓ_q versus ℓ_p norms. Alternatively phrased, the closer we can take p and q, the higher dimension random subspace we can tolerate before the subspace becomes likely to contain a vector which confuses the ℓ_q versus ℓ_p sparsity proxy. Unfortunately, there are very values $q > p$ for which we know how to compute $\max_{v \in V} \|v\|_q / \|v\|_p$ (e.g. $q = \infty, p \in \{1, 2\}$; not sure if there are any other examples, see also Bhaskara and Vijayaraghavan (SODA 2011) for a discussion of related questions, though note that they are talking of a slightly different question, $\max_{\|v\|_p = 1} \|Av\|_q$ for a linear operator A (which for $p = 2$ can encapsulate our question by taking A to be a generator or projector operator to the subspace V, and also, somewhat confusingly, their roles for p and q are switched, and so they mostly deal with the case $q \leq p$ which is not our focus.)

Demanet and Hand [HD13] and Spielman, Wang, and Wright [SWW13] use the ℓ_∞ versus ℓ_1 proxy for sparsity to attack this problem. This can be efficiently computable by running the n linear programs

$$\begin{aligned} &\max v_i && \text{subject to} \\ &x_i \geq v_i \\ &x_i \geq -v_i \\ &\sum_i x_i = 1 \\ &v \in \operatorname{Span}\{v_0, v_1, \ldots, v_k\} \end{aligned}$$

and picking the best optimum.

However, if $k \gg 1$, this will not detect a vector v that is 0.01-sparse.

Exercise 4.1: Prove that for every subspace V of dimension k, there exists a vector $v \in V$ with $\max_i v_i = 1$ and $\sum |v_i| \leq \sqrt{k}/(10n)$

Some works have suggested to use the ℓ_2 vs ℓ_1 proxy. Which actually works pretty well in the sense that if V is a random subspace of dimension at most ηn, then there is no vector $v \in V$ whose ℓ_2 vs ℓ_1 ratio pretends to be a δ-sparse vector where δ is some function of η.

Exercise 4.2:

1. Prove that for every $\eta < 1$ there exists some $\delta = \delta(\eta)$ such that if $v_1, \ldots, v_{\eta n}$ are random Standard Gaussian vectors (each coordinate is distributed according to $N(0, 1)$) then with probability at least 0.9 for every $x \in \mathbb{R}^{\epsilon n}$ with $\|x\|_2^2 = 1$

$$\sum_{i=1}^{\epsilon n} |\langle v_i, x \rangle| \geq \delta n$$

See footnote for hint[1]

[1] **Hint:** This uses concentration of measure. See the papers of Guruswami, Lee and Razoborov and Guruswami, Lee and Wigderson for discussion of this result, its proof, and derandomization.

2. Conclude that for every $\eta < 1$, there is some $\delta = \delta(\eta)$ such that a random subspace (in our model above) does not contain a δ-sparse vector.

However, the ℓ_2 vs ℓ_1 problem has one caveat - we don't know how to compute it, even for a random subspace. In fact, this problem seems quite related to the question of certifying the *restricted isometry property* of a matrix— this is the goal of certifying the a random $m \times n$ matrix A (for $n > m$) satisfies that $\|Ax\|_2 \in (C, 1/C)\|x\|_2$ for every *sparse* vector x. In particular this would be false if there was a sparse vector in the *Kernel* of A, which is a subspace of $\mathbb{R}^n$ of dimension $m - n$. Known methods to certify this property require that the sparse vector x has at most $\sqrt{m}$ nonzero coordinates. See also this blog post of Tao `http://terrytao.wordpress.com/2007/07/02/open-question-deterministic-uup-matrices/` and a paper of Koiran and Ziyzuas connecting this problem to the planted clique problem. (Although note that, unlike the planted clique problem, even a quasipolynomial time algorithm for this problem would be very interesting.)

In the following, we will use ℓ_4 versus ℓ_2 as our proxy for sparsity. A priori this is the "worst of both worlds". On one hand, though it is better than the ℓ_∞ vs ℓ_1 proxy, the ℓ_4/ℓ_2 ratio is a worse proxy than the ℓ_2 vs ℓ_1 ratio, and to detect 1/100-sparse vectors we will need to require the dimension k of the subspace to be at most $\epsilon\sqrt{n}$ for some $\epsilon > 0$ (which is much better than $k = O(1)$ needed in the ℓ_∞/ℓ_1 case but $k = \Omega(n)$ achieved in the ℓ_2/ℓ_1 case). On the other hand, we don't know how to compute this ratio either. In fact, [BBH+12b] showed (via connections with the quantum separability problem) that computing this ratio cannot be done in $n^{O(\log n)}$ time unless SAT has a subexponential time algorithm, and that even achieving weaker approximations would break the Small-Set Expansion (and hence probably also the Unique Games) conjecture. Nevertheless, we will show that we can in fact compute this ratio in the random case, using the degree 4 SOS system. However, as mentioned above, this cannot detect 1/100-sparse vectors if the subspace as dimension $\gg \sqrt{n}$:

Exercise 4.3: Prove that if $V \subseteq R^n$ has dimension $k > \sqrt{n}$ then there is a vector $v \in V$ such that $\mathbb{E}\, v_i^4 \geq \frac{k^2}{10n}\left(\mathbb{E}\, v_i^2\right)^2$.

4.2 Using SoS to Do Better

4.2.1 Description of the Algorithm

We need to phrase our problem as one of polynomial optimization. We have already mentioned that we will optimize the ratio $\|v\|_4/\|v\|_2$. To be more specific: on input a basis B for the subspace V we have real variables $v_1, \ldots, v_n$. Our program is

$$\begin{aligned} &\max \|v\|_4^4 \qquad \text{subject to} \\ &\|v\|_2^2 = 1 \\ &v \in V \end{aligned}$$

(Note that the condition $v \in V$ can be expressed as $n - k$ linear equations.)

We run the level-4 SoS algorithm on this program to obtain a pseudodistribution $\{v\}$ with attending pseudoexpectation operator $\tilde{\mathbb{E}}$. We then run the Quadratic Sampling Lemma to obtain a random vector $w \in V$ that matches the second moments of $\{v\}$. The result will then follow from the following result

Lemma 4.2 (Sparse vector recovery— main lemma)**.** *If the subspace $V = \mathrm{Span}\{v_1, \ldots, v_k\}$ is chosen at random and v_0 is ϵ-sparse for $\epsilon \leq k^2/(100000n)$ then $\tilde{\mathbb{E}}\,\|Pw\|_2^2 \leq 0.01$ where P is the projector to $\mathrm{Span}\{v_1, \ldots, v_k\}$.*

This result means that if $w \in V$ is a vector such that both $\|w\|^2$ and $\|Pw\|_2^2$ are close to their expectations (which are 1 and at most 0.01 respectively) then, writing $w = \langle w, v_0 \rangle v_0 + w'$ where w' is in the span of $\{v_1, \dots, v_k\}$, we see that $\|w'\|^2 \leq 0.01$ and hence $\langle w, v_0 \rangle^2 \geq 0.99$. Somewhat cumbersome but not too hard calculations spelled out below will show that we can get sufficiently close concentration (especially since we can repeat the process and output the sparsest vector w we can find).

Remark Note that the algorithm only looks at the first two moments of the distribution $\{v\}$. So, why did we need $\{u\}$ to be a degree 4 (as opposed to degree 2) pseudo distribution? This is only for the proof, though note that the ℓ_4/ℓ_2 SOS program doesn't even make for degree < 4 pseudo-distributions.

4.2.2 Proof of Main Lemma

The SoS algorithm gives us a pseudodistribution satisfying the constraints

$$\mathcal{E} = \left\{ \|v\|_4^4 = C^4/n, \|v\|_2^2 = 1, v \in \operatorname{Span}\{v_0, \dots, v_k\} \right\}$$

where C is some number so that C^4/n is the value of the solution returned by the level-4 SoS relaxation.

We first prove the main lemma for actual distributions and then demonstrate an instance of "Marley's Hypothesis" [Mar77]: if you proved it for real distributions and didn't use anything too fancy, then every little thing gonna be all right (when you try to prove it for pseudodistributions).

The main result we will take at the moment as a given is the following:

Lemma 4.3 (random subspaces don't contain ℓ_4 versus ℓ_2 sparse vectors—actual distributions)**.** *If $k \ll \sqrt{n}$, with high probability*

$$\|Pv\|_4^4 \leq 10\|Pv\|_2^4/n \tag{4.1}$$

for every v.

We will show that Lemma 4.3 implies our Main Lemma for actual distributions. Namely,

Lemma 4.4 (an ℓ_4 versus ℓ_2 sparse vector must be correlated with v_0—actual distributions)**.** *If P satisfies (4.1) then for every unit vector $w \in V$ with $\|w\|_4 \geq \|v_0\|_4/100 = C/100n^{1/4}$, the square correlation of w with v_0 satisfies $\langle w, v_0 \rangle^2 \geq 1 - O(1/C)$.*

(Note that this is indeed equivalent to the main lemma since $\|w\|_2^2 = \langle w, v_0 \rangle^2 + \|Pw\|_2^2$.)

Proof of Lemma 4.4. Let $w \in V$ be a unit vector. We can write $w = \alpha v_0 + Pw$. Hence, using the triangle inequality for the ℓ_4-norm,

$$\|w\|_4 \leq \alpha\|v_0\|_4 + \|Pw\|_4$$

which can be rearranged to

$$\alpha \geq 1 - \frac{\|Pw\|_4}{\|v_0\|_4}$$

But since $\|v_0\|_4 = C/n^{1/4}$, and Lemma 4.3 $\|Pw\|_4 \leq 2/n^{1/4}$, the RHS is at least $1 - 2/C$. □

4.3 Pseudo-distribution version and proofs

We now state the pseudo-distribution versions of our lemmas and prove them:

Lemma 4.5 (random subspaces don't contain ℓ_4 versus ℓ_2 sparse vectors—pseudodistributions)**.** *With high probability*

$$\|Pv\|_4^4 \preceq 10\|Pv\|_2^4/n \tag{4.2}$$

where we now think of $\|Pv\|_4^4$ *and* $\|Pv\|_2^4$ *as polynomials in indeterminates* v *and with coefficients determined by* P*, and* $\preceq$ *denoting that the polynomial* $10\|Pv\|_2^4 - \|Pv\|_4^4$ *is a sum of squares.*

Lemma 4.6 (an ℓ_4 versus ℓ_2 sparse vector must be correlated with v_0—pseudodistributions)**.** *If* P *satisfies (4.2) then for every degree* 4 *pseudo-distribution* $\{x\}$ *satisfying* $\{\|x\|_2^2 = 1, \|x\|_4^4 = \|v_0\|_4^4 = C^4/n\}$ *it holds that* $\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^2\right] \geq 1 - O(1/C)$*.*

Now we test "Marley's Hypothesis" by lifting the proof of Lemma 4.4 to a proof of Lemma 4.6, using Lemma 4.5 rather than Lemma 4.3 to do the heavy lifting. We need to be able to mimic all the steps we used when everything is wrapped in pseudoexpectations. The main interesting step was a use of the triangle inequality.

Lemma 4.7 (Triangle Inequality for Pseudodistributions)**.** *Let* $\{x, y\}$ *be a degree-4 pseudodistribution. Then*

$$\tilde{\mathbb{E}}\left[\|x+y\|_4^4\right]^{1/4} \leq \tilde{\mathbb{E}}\left[\|x\|_4^4\right]^{1/4} + \tilde{\mathbb{E}}\left[\|y\|_4^4\right]^{1/4}.$$

Exercise 4.4: Prove Lemma 4.7

We note that the following easier bound would be fine for us (and follows from past exercises): if the distribution satisfies the constraint $\|x\|_4^4 \geq \|y\|_4^4$ then

$$\tilde{\mathbb{E}}\left[\|x+y\|_4^4\right] \leq \tilde{\mathbb{E}}\left[\|x\|_4^4\right] + 15\left(\tilde{\mathbb{E}}\left[\|x\|_4^4\right]^{1/4}\right)^{3/4}\left(\tilde{\mathbb{E}}\left[\|y\|_4^4\right]\right)^{1/4}.$$

Proof of Lemma 4.6 from Lemma 4.5. The proof is almost identical to the proof of Lemma 4.4. Let P satisfy

$$\|Px\|_4^4 \preceq \frac{10\|Px\|_2^4}{n}$$

where we interpret both sides as polynomials in x. Let $\{x\}$ be a degree-4 pseudodistribution satisfying $\{\|x\|_2^2 = 1, \|x\|_4^4 = \|v_0\|_4^4 = C^4/n\}$. Using the pseudodistribution triangle inequality,

$$\tilde{\mathbb{E}}\left[\|x\|_4^4\right]^{1/4} \leq \tilde{\mathbb{E}}\left[\|\langle x, v_0\rangle v_0\|_4^4\right]^{1/4} + \tilde{\mathbb{E}}\left[\|Px\|_4^4\right] = \frac{C}{n^{1/4}}\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^4\right]^{1/4} + \tilde{\mathbb{E}}\left[\|Px\|_4^4\right]^{1/4}.$$

Rearranging and using our assumptions on $\{x\}$,

$$\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^4\right]^{1/4} \geq \frac{n^{1/4}}{C}(\tilde{\mathbb{E}}\left[\|x\|_4^4\right]^{1/4} - \tilde{\mathbb{E}}\left[\|Px\|_4^4\right]^{1/4}) = 1 - \frac{n^{1/4}}{C}\tilde{\mathbb{E}}\left[\|Px\|_4^4\right]^{1/4}.$$

Now we use our assumption on P to get

$$\tilde{\mathbb{E}}\left[\|Px\|_4^4\right]^{1/4} \leq 2\frac{\tilde{\mathbb{E}}\left[\|Px\|_2^4\right]^{1/4}}{n^{1/4}}.$$

Moreover, note that $\|Px\|_2^4 \preceq \|x\|_2^4$, since both are homogeneous degree-4 polynomials all of whose monomials are squares and the coefficient of every monomial on the left-hand side is smaller than the corresponding coefficient on the right. This gives

$$\tilde{\mathbb{E}}\left[\|Px\|_2^4\right] \leq \tilde{\mathbb{E}}\left[\|x\|_2^4\right].$$

Putting it together, we get

$$\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^4\right]^{1/4} \geq 1 - \frac{2}{C}\tilde{\mathbb{E}}\left[\|x\|_2^4\right]^{1/4}.$$

Since $\{x\}$ satisfies $\tilde{\mathbb{E}}\left[\|x\|_2^2\right] = 1$, we have

$$\tilde{\mathbb{E}}\left[\|x\|_2^2\left(\|x\|_2^2 - 1\right)\right] = 0$$

and therefore $\tilde{\mathbb{E}}\left[\|x\|_2^4\right] = 1$. Plugging this in to the above,

$$\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^4\right]^{1/4} \geq 1 - \frac{2}{C}.$$

The last step is to relate $\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^4\right]$ and $\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^2\right]$. Again using that $\{x\}$ satisfies $\tilde{\mathbb{E}}\left[\|x\|_2^2\right] = 1$, we have

$$\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^2\|x\|_2^2\right] = \tilde{\mathbb{E}}\left[\langle x, v_0\rangle^2\right].$$

Moreover, since $\langle x, v_0\rangle^2 \preceq \|x\|_2^2$ we must have $\langle x, v_0\rangle^4 \preceq \langle x, v_0\rangle^2\|x\|_2^2$ (the difference of the two sides in the former is a sum of squares; multiplying that SoS polynomial by the square polynomial $\|x, v_0\|^2$ yields another SoS polynomial which is the difference between the two sides in the latter case).

All together, we get

$$\tilde{\mathbb{E}}\left[\langle x, v_0\rangle^2\right] \geq \tilde{\mathbb{E}}\left[\langle x, v_0\rangle^4\right] \geq \left(1 - \frac{2}{C}\tilde{\mathbb{E}}\left[\|x\|_2^4\right]^{1/4}\right)^4 \geq 1 - \frac{8}{C}$$

and we are done. □

4.4 Proof of Lemma 4.5

True to form, we would like to start by proving Lemma 4.3 and then lift the proof to the SoS setting. Lets start with a heuristic argument on why would Lemma 4.3 be true. Think of the case that we fix a unit vector $x \in \mathbb{R}^k$ and pick $v_1, \ldots, v_k$ as random Gaussian vectors of unit norm in $\mathbb{R}^n$, i.e., each entry is distributed as $N(0, 1/\sqrt{n})$. Then, the vector $w = \sum x_i v_i$ would have each coordinate be a Gaussian random variable distributed as $N(0, 1/\sqrt{n})$ (since $\sum x_i^2 = 1$). Now the probability $\|w\|_4^4 \geq C^4/n$ is the probability that $\sum_{i=1}^n g_i^4 \geq nC^4$ where the g_i's are independent standard Gaussians. The dominant term in this probability is the probability that one of those g_i's is at least $Cn^{1/4}$ which happens with $\exp(-C^2\sqrt{n})$ probability. So, if $C^2\sqrt{n} \gg k$, we would be able to do a union bound over a sufficiently fine net of $\mathbb{R}^k$ and rule this out.

This argument can be turned into a proof, but note that we have used a concentration and union bound type of argument, i.e. the dreaded *probabilistic method*, and hence cannot appeal to Marley's Corollary for help. So, we will want to try to present a different argument, that still uses concentration but somehow will work out fine.

4.4.1 Intuition and Heuristic Argument

A formulation that will work just as well for the proof of the main theorem is: given an orthonormal basis matrix B for $\mathrm{Span}\{v_1, \dots, v_k\}$,

$$\|Bv\|_4^4 \leq 10\|v\|_2^4/n \tag{4.3}$$

Now, the matrix B whose columns are $v_1/\sqrt{n}, \dots, v_k/\sqrt{n}$ is almost such a matrix (since these vectors are random, they are nearly orthogonal), and so let's just assume it is the basis matrix. So, we need to show that if B has i.i.d. $N(0, 1/\sqrt{n})$ coordinates and $n \gg k^2$ then with high probability (4.3) is satisfied.

Let $w_1, \dots, w_n$ be the rows of B.

$$n\|Bv\|_4^4 = \sum_{i=1}^{n} \langle w_i, v\rangle^4 = \tfrac{1}{n}\sum_{i=1}^{n} n^2 \langle w_i, v\rangle^4$$

That means that we can think of the polynomial $Q(v) = \|Bv\|_4^4 n$ as the average of n random polynomials each chosen as $\langle g, v\rangle^4$, where $g = \sqrt{n}w$ has i.i.d $N(0, 1)$ entries. Since in expectation $\langle g, v\rangle^4 \leq 5\|v\|_2^4$ (**Exercise 4.5:** verify this), we can see that if n is sufficiently large then $Q(v)$ will with high probability be very close to its expectation and so have $Q(v) \leq 10\|v\|_2^4$.

It turns out that "sufficiently large" in this case means as long as $n \gg k^2$.

We now give some high level arguments on how to make this into a proper proof. We first recall the following exercise:

Exercise 4.6: Let P, Q be two homogenous n-variate degree 4 polynomials, then $P \preceq Q$ if and only if there exist matrices M_P, M_Q such that for every $x \in \mathbb{R}^n$, $P(x) = \langle M_P, x^{\otimes 4}\rangle$ and $Q(x_= \langle M_Q, x^{\otimes 4}\rangle$ such that $M_P \preceq M_Q$ in the spectral sense. (i.e., where we say that a matrix A satisfies $0 \preceq A$ if $w^\top A w \geq 0$ for all w.)

As a corollary, such a polynomial P satisfies $P \preceq \lambda\|x\|_2^4$ if there exists such a matrix M_P with $\|M_P\| \leq \lambda$ where $\|M_P\|$ denotes the spectral norm. (Can you see why?)

This connection suggest using the *Matrix Chernoff Bound* and specifically the following theorem

Theorem 4.8 (Matrix Chernoff Bound, Ahlswede and Winter)**.** *Let $X_1, \dots, X_n$ be i.i.d. $m \times m$ matrix valued random variables with expectation M and with $M - cI \preceq X_i \preceq M + cI$, then*

$$\Pr[\tfrac{1}{n}\sum X_i \notin M \pm \epsilon I] \leq m \exp(-\epsilon^2 n/c^2)$$

(One intuition for this bound is that it turns out that diagonal matrices are the hardest ones, and if the distribution was on diagonal matrices, then we need to use the usual Chernoff bound m times and so lose a factor of m in the probability bound.)

In our case, the distribution of X_i's is the distribution of the matrix corresponding to the polynomial $\langle g, x\rangle^4$ whose largest eigenvalue is $\|g\|^4 = k^2$, and so the RHS becomes $k^2 \exp(-\epsilon^2 n/k^4)$ and so if $n \gg k^4 \log k$ this will suffice (I think this argument can be made tighter to replace k^4 with k^2). It turns out that (at considerable pain) one can avoid the $\log k$ factor and get the condition $n \gg k^2$.

4.5 Full proof of Lemma 4.5

These next sections contain a great exposition of the full proof with the right, $k = O(\sqrt{n})$ bound, as heroically written by Samuel Hopkings.

4.5.1 Lemma 4.5 Holds in Expectation

As in the heuristic argument, the first step in both proofs is to show that the SoS relation we need holds in expectation. For convenience we now change notation and let x be a typical vector in $\mathrm{Span}\{v_1, \ldots, v_k\}$. That is, for some indeterminates $\alpha_1, \ldots, \alpha_k$, we have $x = \sum \alpha_i v_i$. We want to show

$$\|x\|_4^4 \preceq \frac{10\|x\|_2^4}{n}. \tag{4.4}$$

where both sides are now polynomials in $\alpha_1, \ldots, \alpha_k$. We mechanically expand both sides to the equivalent formulation

$$\sum_s \sum_{i,j,k,l} \alpha_i \alpha_j \alpha_k \alpha_l v_i(s) v_j(s) v_k(s) v_l(s) \preceq \frac{10}{n} \sum_{s,t,i,j,k,l} \alpha_i \alpha_j \alpha_k \alpha_l v_i(s) v_j(s) v_k(t) v_l(t)$$

Our task in this section is to hit both sides with $\mathbb{E}_{v_1,\ldots,v_k}\left[\cdot\right]$ and show

$$\mathbb{E}_{v_1,\ldots,v_k}\left[\sum_s \sum_{i,j,k,l} \alpha_i \alpha_j \alpha_k \alpha_l v_i(s) v_j(s) v_k(s) v_l(s)\right] \preceq \mathbb{E}_{v_1,\ldots,v_k}\left[\frac{10}{n} \sum_{s,t,i,j,k,l} \alpha_i \alpha_j \alpha_k \alpha_l v_i(s) v_j(s) v_k(t) v_l(t)\right]. \tag{4.5}$$

We need to calculate the expected coefficient of every monomial $\alpha_i \alpha_j \alpha_k \alpha_l$ on both the right-and left-hand sides of (4.5). This is an unpleasant but not terribly difficult case analyis.

Notational Conventions We need to distinguish between ordered multisets of indices i, j, k, l, which we will denote just like that, and sets of indices which do not have repeated elements (even though in our notation some elements may be listed multiple times), which we denote $\{i, j, k, l\}$. When we want to sum over all pairs i, j we write $\sum_{i,j}$, and if we don't want to double-count, we use $\sum_{i \le j}$.

Left-Hand Side of (4.5) For each $\{i, j, k, l\}$ we calculate

$$\sum_{\pi \in S_4(i,j,k,l)} \mathbb{E}\left[\sum_s v_{\pi(i)}(s) v_{\pi(j)} j(s) v_{\pi(k)}(s) v_{\pi(l)}(s)\right]$$

which is the coefficient of $\alpha_i \alpha_j \alpha_k \alpha_l$, where S_4 is the symmetric group on the set $\{i, j, k, l\}$.

First note that each term in the sum is identical, so we may equivalently calculate

$$n |S_4(i,j,k,l)| \, \mathbb{E}\left[v_i(1) v_j(1) v_k(1) v_l(1)\right].$$

If one of $\{i, j, k, l\}$ is unique then this is 0. If the $\{i, j, k, l\}$ has exactly two unique elements, then $\mathbb{E}\left[v_i(1) v_j(1) v_k(1) v_l(1)\right] = \mathbb{E}\left[\gamma^2 \gamma_2'\right] = 1$, where $\gamma, \gamma' \sim N(0,1)$, and $|S_4(i,j,k,l)| = 3$. If $\{i, j, k, l\}$ has just one unique element, then $\mathbb{E}\left[v_i(1) v_j(1) v_k(1) v_l(1)\right] = \mathbb{E}\left[\gamma^4\right] = 3$ and $|S_4(i,j,k,l)| = 1$. In sum, the left-hand side is equal to

$$3n \sum_{i \le j} \alpha_i^2 \alpha_j^2. \tag{4.6}$$

Right-Hand Side of (4.5) For each i, j, k, l we calculate

$$\sum_{\pi \in S_4(i,j,k,l)} \sum_{s,t} \mathbb{E}\left[v_{\pi(i)}(s)v_{\pi(j)}(s)v_{\pi(k)}(t)v_{\pi(l)}(t)\right].$$

We split it into two sums:

$$\sum_{\pi \in S_4(i,j,k,l)} \sum_{s=t} \mathbb{E}\left[v_{\pi(i)}(s)v_{\pi(j)}(s)v_{\pi(k)}(t)v_{\pi(l)}(t)\right] + \sum_{\pi \in S_4(i,j,k,l)} \sum_{s \neq t} \mathbb{E}\left[v_{\pi(i)}(s)v_{\pi(j)}(s)v_{\pi(k)}(t)v_{\pi(l)}(t)\right].$$

In the first, we have recovered exactly the left-hand side of (4.5). In the second:

- If $\{i, j, k, l\}$ has some element appearing just once, then the corresponding terms are all 0, as before.
- If $\{i, j, k, l\}$ contains exactly two unique elements then there are four elements π of $S_4(i, j, k, l)$ which will have $\pi(i) = \pi(j)$ and $\pi(k) = \pi(l)$ and $n^2 - n$ terms in the inner sum, so we get $4n^2 - 4n$
- If $\{i, j, k, l\}$ has just one unique element, we have

  $$\mathbb{E}\left[v_{\pi(i)}(s)v_{\pi(j)}(s)v_{\pi(k)}(t)v_{\pi(l)}(t)\right] = \mathbb{E}\left[\gamma^2\gamma'^2\right] = 1.$$

  and so the corresponding sum over $s \neq t$ contributes $n^2 - n$.

All in all, the right-hand side is equal to

$$\frac{10}{n}\left[3n\sum_{i \le j} \alpha_i^2\alpha_j^2 + (n^2 - n)\left(\sum_{i<j} 4\alpha_i^2\alpha_j^2 + \sum_i \alpha_i^4\right)\right]. \tag{4.7}$$

It's now a straightforward exercise to check that (4.6) $\preceq$ (4.7).

4.5.2 First Proof of Lemma 4.5, $k = O(n^{1/4})$

We now know that (4.4) holds in expectation, by which we mean that some polynomial $R(\alpha)$ is a sum of squares. Conceptually, what remains to do is show that R is close to its expectation with high probability. What is the right sense of closeness? We will see in the next section that, to achieve the optimum bound of $k = O(n^{1/2})$, we need to interpret "close" to mean that some matrix derived R's coefficient matrix is close to its expectation in the spectral norm.

However, we can achieve $k = O(n^{1/4})$ with a somewhat cruder argument. Our first observation is that

$$\alpha_i\alpha_j\alpha_k\alpha_l \preceq \alpha_i^2\alpha_j^2 + \alpha_k^2\alpha_l^2 \tag{4.8}$$

$$-\alpha_i\alpha_j\alpha_k\alpha_l \preceq \alpha_i^2\alpha_j^2 + \alpha_k^2\alpha_l^2. \tag{4.9}$$

At a high level, the idea is that so as long as the coefficients of the terms $\alpha_i\alpha_j\alpha_k\alpha_l$ which are not squares do not get too big (they are 0 in expectation) and the coefficients of the dominating terms $\alpha_i^2\alpha_j^2$ and $\alpha_k^2\alpha_l^2$ do not get too small, we can use this relation to preserve SoS-ness.

Now we get a little more formal. Let

$$R(\alpha) = \frac{10}{n}\sum_{s,t,i,j,k,l} \alpha_i\alpha_j\alpha_k\alpha_l v_i(s)v_j(s)v_k(t)v_l(t) - \sum_s \sum_{i,j,k,l} \alpha_i\alpha_j\alpha_k\alpha_l v_i(s)v_j(s)v_k(s)v_l(s).$$

We will be a little fast-and-lose with the constants for the sake of readability. In particular, we don't lose too much if we ignore the permutations π and just treat each permutation individually.

We will charge to the coefficient of $\alpha_i^2\alpha_j^2$ the coefficients of $\alpha_i\alpha_j\alpha_k\alpha_l$ for all indices k, l. As long as for all i, j the coefficient of $\alpha_i^2\alpha_j^2$ stays positive when we subtract off the absolute values of the coefficients we are charging to it, R is SoS by (4.8) and (4.9).

Unfortunately, even for this cruder version of the argument we need a concentration inequality whose proof is outside scope of these notes. The following statement is a special case of Theorem 1.10 in [SS12], who refer the reader to [Jan97] for a proof.

Theorem 4.9. *Consider a degree-q polynomial $f(Y) = f(Y_1, \ldots, Y_m)$ of independent centered Gaussian random variables $Y_1, \ldots, Y_m$. Then*

$$\Pr\left[|f(Y) - \mathbb{E}\left[f(Y)\right]| \geq \lambda\right] \leq e^2 e^{-\left(\frac{\lambda^2}{AVar[f(Y)]}\right)^{1/q}}$$

where A is a universal constant.

We will apply Theorem 4.9 with the degree-4 polynomials which are the coefficients in $R(\alpha)$. To apply the theorem we must estimate the variances of the coefficients. Let

$$f_{ijkl} = \frac{10}{n}\sum_{s,t} v_i(s)v_j(s)v_k(t)v_l(t) - \sum_s v_i(s)v_j(s)v_k(s)v_l(s).$$

By independence considerations, $Var[f_{ijkl}] \leq Var[f_{iiii}]$. It is a somewhat involved and unenlightening calculation to check that $Var[f_{iiii}] = O(n)$. [2] We assume it here, and note that in the preceding section we showed that the coefficients of $\alpha_i^2\alpha_j^2$ in R are at least $5n$ in expectation.

We are nearly there—the rest of the analysis is a standard combination of the concentration inequality and a union bound, so we will be hand-wavy and ignore the log factors needed to make things precise.

The probability that the coefficient of $\alpha_i^2\alpha_j^2$ is less than $4n$ is $O(e^{-n^{1/2}})$ by application of Theorem 4.9. On the other hand, the probability that any of the k^2 coefficients (of $\alpha_i\alpha_j\alpha_k\alpha_l$) being charged to $\alpha_i^2\alpha_j^2$ is greater than $3n/k^2$ in absolute value is, again by application of the theorem, at most e^{-3n/k^4}. As long as $k^4 < n$, we can pick constants and *polylog* factors to complete the proof.

4.5.3 Second Proof of Lemma 4.5, $k = O(n^{1/2})$

The arguments in the following section first appear in section 7 of [BBH+12b], and are fleshed out in [DS].

The first proof loses something in requiring a particular SoS decomposition of R. The following more delicate argument avoids this by using a single application of a concentration inequality to the entire polynomial at once rather than bounding each coefficient separately.

Matrix Concentration Setup Matrix concentration inequalities are the analogue of Chernoff/Bernstein/Azuma/Hoeffding/etc. bounds when the terms being summed are independent or weakly-dependent random matrices rather than scalars. They bound the distance of the resulting random matrix from its expectation in the spectral norm. For a readable treatment of "elementary"

[2] To check this, expand the variance as $\mathbb{E}\left[\cdot^2\right] - \mathbb{E}\left[\cdot\right]^2$ and count how many terms in each resulting sum cancel between the square-expectation and the expectation-squared. Terms always cancel unless indices match up, violating independence.

matrix concentration inequalities, see [Tro12] or [Tao12]. Most of these inequalities rely on variables which are bounded; our variables instead have Gaussian tails. This could be dealt with by some truncation business, but instead we will use a higher-tech result which simultaneously handles the Gaussian-ness of the underlying distribution and saves a log factor over more elementary methods.

The following presentation follows [BBH$^+$12b], with notation somewhat modified to fit these notes. For background on the ψ_p norm (in particular the ψ_2 case) see [Ver10], especially section 5.2.3 on sub-Gaussian random variables.

The ψ_p norm of a distribution $\{a\}$ on $\mathbb{R}^k$[3] is the least $C > 0$ so that

$$\max_{w \in \mathbb{R}^k, \|w\|_2 = 1} \mathbb{E}\left[e^{\frac{|\langle w, a\rangle|^p}{k^{p/2} C^p}} \right] \leq 2.$$

(We let it be ∞ if no such C exists.) Observe that for $p = 2$ this is quantifying the "Gaussian-ness" of the one-dimensional marginals of the distribution $\{a\}$. The scale factor of $k^{p/2}$ is not in the usual definition but we want to match the theorem statement in [ALPTJ11].

We also require a bounded-ness condition: that there is a constant $K \geq 1$ so that for independent samples $a_1, \ldots, a_n \sim \{a\}$,

$$\Pr\left[\max_{i \leq n} \|a_i\|_2 \geq K(nk)^{1/4} \right] \leq e^{-\sqrt{k}}.$$

Now we can state the main theorem of [ALPTJ11, ALP$^+$10], as stated in [BBH$^+$12b] (modulo a minor adjustment of scale factors).

Theorem 4.10. *Let $\{a\}$ be a distribution on $\mathbb{R}^k$ so that $\mathbb{E}\left[aa^T\right] = I$, the ψ_1 norm of $\{a\}$ is at most $\psi \geq 0$, and the boundedness condition holds for $\{a\}$ with constant K. Let $a_1, \ldots, a_n$ be independent samples from $\{a\}$. Then for some universal constants $c, C > 0$, with probability at least $1 - 2e^{-c\sqrt{k}}$,*

$$(1 - \epsilon) I \preceq \frac{1}{n} \sum_{i=1}^{n} a_i a_i^T \preceq (1 + \epsilon) I$$

where I is the identity matrix, $\preceq$ is the PSD ordering, and $\epsilon = C(\psi + K)^2 \sqrt{k/n}$.

From Matrix Concentration to Lemma 4.5: Plan of Attack We recall the correspondence between polynomials and coefficient matrices that makes the SoS algorithm tick in the first place. If we can find matrices M_A and M_B for A, B polynomials so that $x^T M_A x = A(x)$ and $x^T M_B x = B(x)$, and if $M_A \preceq M_B$, then $A \preceq B$.

We will use this method to show that two inequalities each hold with high probability:

$$\frac{1}{3} \mathbb{E}\left[\|x\|_2^4\right] \preceq \|x\|_2^4 \tag{4.10}$$

$$\|x\|_4^4 \preceq 3\,\mathbb{E}\left[\|x\|_4^4\right]. \tag{4.11}$$

Since we already know

$$\mathbb{E}\left[\|x\|_4^4\right] \preceq \frac{10}{n} \mathbb{E}\left[\|x\|_2^4\right]$$

this gives us Lemma 4.5 (modulo a minor adjustment of the constants).

[3] We have to use k and n here on purpose—when we apply the theorem, each sample will correspond to a dimension of our ambient space and there will have dimension the same as the dimension of our subspace.

Warm Up: Concentration for $\|x\|_2^2$ We turn now to showing that $\|x\|_2^2$ is rarely too much smaller than its expectation. We will be able to leverage this to show that (4.10) holds with high probability. We expand $\|x\|_2^2$ as a polynomial in $\alpha_1, \dots, \alpha_k$:

$$\|x\|_2^2 = \sum_{s,i,j} \alpha_i \alpha_j v_i(s) v_j(s).$$

Let $a_1, \dots, a_n \in \mathbb{R}^k$ be the rows of the matrix whose columns are $v_1, \dots, v_k$. Let $\{a\}$ be the distribution of the a_i's. Observe that we can rewrite the previous equation as

$$\|x\|_2^2 = \sum_{s,i,j} \alpha_i \alpha_j a_s(i) a_s(j)$$

which has coefficient matrix $\sum_s a_s a_s^T$. Furthermore, $\mathbb{E}\left[aa^T\right] = I$. Now we need to calculate the ψ_1 norm. The distribution $\{a\}$ is rotationally invariant, so we may take $w = e_1$ to be the first standard basis vector, in which case we need to find C so that

$$\mathbb{E}\left[e^{|a(1)|/k^{1/2}C}\right] \leq 2$$

Mathematica says that 2 is an upper bound.

The last thing to check before applying the matrix concentration theorem is the boundedness condition. The event $\max \|a_i\|_2 \geq K(nk)^{1/4}$ is equivalent to the event $\max \|a_i\|_2^2 \geq K^2(nk)^{1/2}$. Since $\|a_i\|_2^2$ is a degree-2 polynomial of independent Gaussians, we can use Theorem 4.9 to show that $K = 1$ suffices if $k \approx \sqrt{n}$.[4]

Now we can apply the concentration theorem with $k = \sqrt{n}$ to get that with probability at least $1 - e^{-c\sqrt{k}}$,

$$(1 - 9C^2 n^{-1/4})I \preceq \frac{1}{n} \sum_{i=1}^{n} a_i a_i^T \preceq (1 + 9C^2 n^{-1/4})I.$$

As soon as n gets big enough, this yields

$$0.99\, \mathbb{E}\left[\|x\|_2^2\right] \preceq \|x\|_2^2.$$

Concentration for $\|x\|_2^4$ We now make some observations:

1. If A, B are SoS polynomials with $A \preceq B$, then $A^2 \preceq B^2$. To see this, write $B^2 - A^2 = (B + A)(B - A)$ and note that the multiplicands in the latter are both SoS by hypothesis, so their product is as well.

2. $\mathbb{E}\left[\|x\|_2^4\right]$ and $\mathbb{E}\left[\|x\|_2^2\right]^2$ differ only by a constant factor. The proof is mechanical. In particular,

$$\mathbb{E}\left[\|x\|_2^4\right] \preceq 3\, \mathbb{E}\left[\|x\|_2\right]^2.$$

3. $\|x\|_2^2 \succeq \mathbb{E}\left[\|x\|_2^2\right] \succeq 0$.

Taken all together, we get $0.1\, \mathbb{E}\left[\|x\|_2^4\right] \preceq \|x\|_2^4$ which, modulo some adjustments to the constants, is (4.10).

[4] Actually we could take K to be $o(1)$ in this case. The analysis where we have to worry about these things is for $\|x\|_4^4$.

Concentration for $\|x\|_4^4$ It remains only to dispatch (4.11). This we also do by appeal to our matrix concentration theorem, but we will have to be somewhat more careful in using it, for a couple reasons:

1. We will not initially end up with a distribution $\{a\}$ with $\mathbb{E}\left[aa^T\right] = I$.
2. The calculations to show that the ψ_1 and boundedness conditions hold will be somewhat more arduous.

To handle the first of these, we will start in the same way as before by finding the distribution whose empirical covariance matrix is the coefficient matrix of $\|x\|_4^4$, but then we will have to hit these vectors with the pseudo-inverse of $\mathbb{E}\left[\|x\|_4^4\right]$ to get a distribution which has true covariance matrix I. Then we use the lemma below to get the result we want. To handle the second issue, we shut up and calculate. Be prepared for some Taylor series.

Lemma 4.11. *Let Σ be a symmetric real PSD matrix, $\Sigma^{1/2}$ its square root, Σ^{-1} its pseudo-inverse, and $\Sigma^{-1/2}$ the square root of its pseudo-inverse. Then*

- $\Sigma^{-1/2}\Sigma\Sigma^{-1/2} = I$.
- $\Sigma^{1/2}\Sigma^{-1/2} = I$.

Proof. The proofs of both facts are straightforward applications of the characterization of the pseudoinverse and square roots of diagonal PSD matrices (respectively, take the inverses and square roots of the diagonal entries) plus the fact that a symmetric real matrix can be diagonalized. □

We recall that $\|x\|_4^4$ expands as

$$\|x\|_4^4 = \sum_s \sum_{i,j,k,l} \alpha_i\alpha_j\alpha_k\alpha_l v_i(s)v_j(s)v_k(s)v_l(s).$$

Once again, let the vectors $a_1, \ldots, a_n$ be the rows of the matrix whose columns are the v_i's. Then we can rewrite:

$$\|x\|_4^4 = \sum_s \sum_{i,j,k,l} \alpha_i\alpha_j\alpha_k\alpha_l a_s(i)a_s(j)a_s(k)a_s(l).$$

The coefficient matrix of $\|x\|_4^4$ therefore $\sum_s (a_s \otimes a_s)(a_s \otimes a_s)^T$ where $\otimes$ is the tensor product. Let $\{a\}$ again be the distribution of the a_s's, and let $\Sigma = \mathbb{E}\left[(a_s \otimes a_s)(a_s \otimes a_s)^T\right]$ be the true covariance matrix of $(a_s \otimes a_s)$. Since it is a covariance matrix, Σ is symmetric and PSD, so the results of Lemma 4.11 apply.

Consider now the distribution $\{b\}$ given by $\Sigma^{-1/2}(a \otimes a)$. By Lemma 4.11, the true covariance matrix satisfies $\mathbb{E}\left[bb^T\right] = I$. Let $b_1, \ldots, b_s$ be independent samples from $\{b\}$. If we could prove

$$\sum_s b_s b_s^T \preceq 3I$$

with high probability, then by hitting both sides with $\Sigma^{1/2}$ on right and left and again applying Lemma 4.11 we would have (4.11). Hence, all that remains to do is calculate ψ_1 norm and the boundedness constant for $\{b\}$ in order to apply the matrix concentration theorem.

We begin with the ψ_1-norm calculation. We need an upper bound on C so that

$$\max_{w \in \mathbb{R}^{k^2}, \|w\|_2 = 1} \mathbb{E}\left[e^{\frac{|\langle w, b\rangle|}{kC}}\right] \le 2.$$

Making the substitution $u = \Sigma^{-1/2}w$, we need to find C so that for all u with $u^T \Sigma u \leq 1$,

$$\mathbb{E}\left[e^{\frac{|\langle \Sigma^{1/2}u, a \otimes a\rangle|}{kC}}\right] \leq 2.$$

The entires of Σ are all either zeroth, first, or second moments of a standard Gaussian, depending on how many repeated indices are at a particular entry. In particular, $\Sigma_{iiii} = 3$ and $\Sigma_{ijij} = \Sigma_{iijj} = \ldots = 1$, and all other entires are 0. (See our analysis of $\mathbb{E}\left[\|x\|_4^4\right]$.)

The condition $u^T \Sigma u \leq 1$, if we interpret u as a $k \times k$ matrix M, implies that

$$\sum_{ij} M_{ij}^2 + \sum_{ij} M_{ij}M_{ji} + \sum_{ij} M_{ii}M_{jj} \leq 1.$$

(where we have dropped some 3's to 1's, which can only reduce the left-hand side). Because $a \otimes a$, considered as a $k \times k$ matrix, is symmetric, we may assume that M is also symmetric (otherwise take $M'_{ij} = (M_{ij} + M_{ji})/2$ and note that the inner product with $a \otimes a$ is preserved). With the symmetry assumption we get

$$2\sum_{ij} M_{ij}^2 + \left(\sum_i M_{ii}\right)^2 \leq 1$$

which we wastefully use to get

$$\sum_{ij} M_{ij}^2 + \left(\sum_i M_{ii}\right)^2 \leq 1.$$

Since the trace of a matrix is the sum of its eigenvalues, we get

$$\left(\sum_i \lambda_i\right)^2 + \sum_i \lambda_i^2 \leq 1$$

which gives $\sum_i \lambda_i \leq 1$ and $\sum_i \lambda_i^2 \leq 1$. All this shows that it will now suffice to prove that there is $C = O(1)$ so that for every symmetric $k \times k$ matrix M with $TrM \leq 1$ and $TrM^2 \leq 1$,

$$\mathbb{E}\left[e^{\frac{|a^T M a|}{kC}}\right] \leq 2.$$

By rotational invariance of a, we may actually assume that M is diagonal. Then

$$\frac{1}{kC}|a^T Ma| = \frac{1}{kC}|\sum_i \lambda_i a_i^2| \leq \frac{1}{kC}\left(\left|\sum_i \lambda_i\right| + \left|\sum_i \lambda_i(a_i^2 - 1)\right|\right) \leq \frac{1}{kC} + \frac{1}{kC}\left|\sum_i \lambda_i(a_i^2 - 1)\right|$$

where the second-to-last step is the triangle inequality and the last step is since $TrM \leq 1$.

Conditioned on $|\sum_i \lambda_i(a_i^2 - 1)| \leq 0$, the expectation we're bounding becomes at most $e^{1/kC} + 1$, so clearly we can take $C = O(1)$ in this case. So we may assume that at least half of the total expectation comes from the case when $|\sum_i \lambda_i(a_i^2 - 1)| \geq 0$. By independence and the preceding analysis, we get

$$\mathbb{E}\left[e^{\frac{|a^T Ma|}{kC}}\right] \leq 2e^{1/kC} \prod_i \mathbb{E}\left[e^{\lambda_i(a_i^2 - 1)/kC}\right]. \tag{4.12}$$

Now we Taylor expand:

$$\mathbb{E}\left[e^{\lambda_i(a_i^2-1)/kC}\right] = \sum_p \frac{1}{p!}\mathbb{E}\left[\frac{1}{(kC)^p}\lambda_i^p(a_i^2 - 1)^p\right].$$

We want to bound each term in the Taylor expansion with something involving λ_i^2 so we can use the condition $TrM^2 \le 1$. Recall that the moments $\mathbb{E}\left[a_i^{2p}\right]$ grow like $\prod_{q \text{ odd}, \; q \le p} q \approx p!2^p$. So as long as $1/(kC) \le 1/16$ or so, recalling that $\sum_i \lambda_i^2 \le 1$ and therefore $|\lambda_i| \le 1$ for all i, we can estimate

$$\frac{1}{(kC)^p}\lambda_i^p a_i^{2p} \le p!(1/kC)^2\lambda_i^2 2^{-p}.$$

Plugging this into the Taylor expansion, using Jensen's inequality on $(a_i^2-1)^p$, and splitting off the terms with $p \le 2$ gives

$$\mathbb{E}\left[e^{\lambda_i(a_i^2-1)/kC}\right] \le 2 + (kC)^{-2}\lambda_i^2\sum_{k\ge 2}2^{-k} \le e^{O((kC)^{-2}\lambda_i^2)}.$$

Finally, plugging this back into (4.12) we get

$$\mathbb{E}\left[e^{\frac{|a^T Ma|}{kC}}\right] \le 2e^{1/kC}e^{O((kC)^{-2})\sum\lambda_i^2} \le 2$$

for all k and sufficiently small C. So the ψ_1 norm of $\{b\}$ is $O(1)$.

The very last thing we need to do is check the boundedness condition for $\{b\}$. Note that it will suffice to show that $(a \otimes a)$ satisfies the boundedness condition rather than $\{b\}$ if we can show that the largest nonzero eigenvalue of $\Sigma^{-1/2}$ is $O(1)$, which would follow if we could show that the smallest nonzero eigenvalue of Σ is $\Omega(1)$.

The proof of the boundedness condition for $a \otimes a$ is similar to that for $\{a\}$.

4.6 Analyzing success probability, proof of the qaudratic sampling lemma

We restate the QSL here:

Lemma 4.12 (Quadratic Sampling Lemma)**.** *If $\{x\}$ is a degree $d \ge 2$ pseudo distribution, then there exists a Gaussian distribution $\{u\}$ such that $\tilde{\mathbb{E}}\left[P(x)\right] = \mathbb{E}\left[P(u)\right]$ for every polynomial P of degree at most 2. This distribution can be efficiently computed from input $\{x\}$.*

Proof. By shifting we can assume that $\tilde{\mathbb{E}}\left[x_i\right] = 0$ for all i. Since $\{x\}$ is a degree 2 pseudo-distribution, its second moment matrix $M = \tilde{\mathbb{E}}\left[x^{\otimes 2}\right] = \tilde{\mathbb{E}}\left[xx^\top\right]$ is psd. Hence, we can write $M = B^\top B$ where B is a $d \times n$ matrix with columns $b_1, \ldots, b_n$ and so $M_{i,j} = \langle b_i, b_j\rangle$. Choose a random standard Gaussian vector $g = (g_1, \ldots, g_n)$ and let $z_i = \langle b_i, g\rangle$.

Then, for every i, j, we get that

$$\mathbb{E}\left[z_iz_j\right] = \mathbb{E}\left[\langle b_i, g\rangle\langle b_j, g\rangle\right] = \sum_{a,b} b_i(a)g_a b_i(b)g_b = \sum a_i(a)b_j(a) = \langle b_i, b_j\rangle = M_{i,j}$$

using the fact that the Gaussians are independent and so $\mathbb{E}\left[g_a g_b\right]$ equals 0 if $a \ne b$ and equals 1 otherwise. □

Proof of Main Theorem from Main Lemma and Quadratic Sampling Lemma. Let $\{u\}$ be the Gaussian distribution obtained from $\{x\}$ which satisfies $\mathcal{E}$. The Main Lemma says that $\tilde{\mathbb{E}}_x\left[\|Px\|_2^2\right] \le 0.001$ with high probability, and since $\|Px\|_2^2$ is a degree-2 polynomial, the Quadratic Sampling Lemma then implies that $\mathbb{E}_u\left[\|Pu\|_2^2\right] \le 0.001$. By the same argument, $\mathbb{E}_u\left[\|u\|_2^2\right] = 1$.

We can argue using standard techniques to transfer the expectation statements to probability bounds (the proof comes at the end of the proof of the main theorem).

1. $\Pr_u\left[\|u\|_2^2 \leq \frac{1}{2}\right] \leq \frac{5}{6}$
2. $\Pr_u\left[\|Pu\|_2^2 \geq 0.01\right] \leq 1/10$.

Hence, with probability at least $1/15$ the algorithm samples u with $\|u\|_2^2 \geq 1/2$ and $\|Pu\|_2^2 \leq 0.01$. In this case, $\|Pu\|_2^2 \leq 0.02\|u\|_2^2$. We assumed $v_0 \perp v_1, \dots v_k$, which means we can write

$$\|u\|_2^2 = \langle u, v_0\rangle^2 \|v_0\|_2^2 + \|Pu\|_2^2 = \langle u, v_0\rangle^2 + \|Pu\|_2^2.$$

Since $\|Pu\|_2^2$ makes up only a 0.02 fraction of this mass, $\langle u, v_0\rangle$ must make up the rest, and we get $\langle u, v_0\rangle \geq 0.98\|u\|_2^2$. Scaling u to be unit, we recover a unit vector $u/\|u\|$ with very high correlation with v_0.

By the first part of the main lemma, to test whether it has succeeded, the algorithm simply checks the ℓ_4-versus-ℓ_2 sparsity of the vector u. To succeed with probability $1 - 1/poly(n)$ it will need to sample about $\log n$ times.

Proof of (1). We start with a standard second-moment concentration inequality, which we prove here for completeness. Let X be a nonnegative random variable and let $\theta > 0$. Then

$$\mathbb{E}[X] \leq \theta + \Pr[X \geq \theta]\,\mathbb{E}[X \mid X \geq \theta]$$

$$\mathbb{E}\left[X^2\right] \geq \Pr[X \geq \theta]\,\mathbb{E}\left[X^2 \,\middle|\, X \geq \theta\right] \overset{\text{Jensen}}{\geq} \Pr[X \geq \theta]\,\mathbb{E}\left[X^2 \,\middle|\, X \geq \theta\right]^2.$$

Combining the equations by eliminating $\mathbb{E}\left[X \middle| X \geq 0\right]$ and rearranging gives

$$\Pr[X \geq \theta] \geq \frac{\mathbb{E}[X - \theta]^2}{\mathbb{E}[X^2]}.$$

We apply this to the random variable $\|u\|_2^2$ for some θ to be chosen later to get

$$\Pr\left[\|u\|_2^2 \geq \theta\right] \geq \frac{\mathbb{E}\left[\|u\|_2^2 - \theta\right]^2}{\mathbb{E}\left[\|u\|_2^4\right]}. = \frac{(1-\theta)}{\mathbb{E}\left[\|u\|_2^4\right]}.$$

We need to upper-bound $\mathbb{E}\left[\|u\|_2^4\right]$. We expand

$$\mathbb{E}\left[\|u\|_2^4\right] = \sum_{i,j} \mathbb{E}\left[u(i)^2 u(j)^2\right] \overset{\text{Cauchy-Schwarz}}{\leq} \sum_{i,j} \sqrt{\mathbb{E}[u(i)^4]}\sqrt{\mathbb{E}[u(j)^4]} = \left(\sum_i \sqrt{\mathbb{E}[u(i)^4]}\right)^2.$$

For fixed i, let μ_i, σ_i be such that $u(i) \sim N(\mu_i, \sigma_i)$. It is a Wikipedia-able fact that

$$\mathbb{E}\left[u(i)^2\right] = \mu_i^2 + \sigma_i^2$$
$$\mathbb{E}\left[u(i)^4\right] = \mu_i^4 + 6\mu_i^2\sigma_i^2 + 3\sigma_i^4.$$

Hence,

$$\mathbb{E}\left[u(i)^4\right] = \mathbb{E}\left[u(i)^2\right]^2 + 4\mu_i^2\sigma_i^2 + 2\sigma^4 \leq 3\,\mathbb{E}\left[u(i)^2\right]^2$$

which yields

$$\left(\sum_i \sqrt{\mathbb{E}[u(i)^4]}\right)^2 \leq 3\left(\sum_i \mathbb{E}\left[u(i)^2\right]\right)^2 = 3.$$

So if we pick $\theta = \frac{1}{2}$ we get $\Pr\left[\|u\|_2^2 \geq \frac{1}{2}\right] \geq \frac{1}{6}$. □

Proof of (2). This is straight Markov's inequality. □

□

Part II: Dictionary Learning

4.7 Introduction

The *dictionary learning / sparse coding* problem is defined as follows: there is an unknown $n \times m$ matrix $A = (a_1|\cdots|a_m)$ (think of $m = 10n$). We are given access to many examples of the form

$$y = Ax + e \tag{4.13}$$

for some distribution $\{x\}$ over sparse vectors and distribution $\{e\}$ over noise vectors with low magnitude.

Our goal is to learn the matrix A, which is called a *dictionary.*

The intuition behind this problem is that natural data elements are sparse when represented in the “right” basis, in which every coordinate corresponds to some meaningful features. For example while natural images are always dense in the pixel basis, they are sparse in other bases such as wavelet bases, where coordinates corresponds to edges etc.. and for this reason these bases are actually much better to work with for image recognition and manipulation. (And the coordinates of such bases are sometimes in a non-linear way to get even more meaningful features that eventually correspond to things such as being a picture of a cat or a picture of my grandmother etc. or at least that’s the theory behind deep neural networks.) While we can simply guess some basis such as the Fourier or Wavelet to work with, it is best to learn the right basis directly from the data. Moreover, it seems that in many cases it is actually better to learn an *overcomplete* basis: a set of $m > n$ vectors $a_1, \ldots, a_m \in \mathbb{R}^n$ so that every example from our data is a sparse linear combination the a_k’s. (Sometimes just considering the case that the a_m’s are a union of two bases, such as the standard and Fourier one, already gives rise to many of the representational advantages and computational challenges.)

Olshausen and Field were the first to define this problem - they used a heuristic to learn such a basis for some natural images, and argued that representing images via such an dictionary is somewhat similar to what is done in the human visual cortex. Since then this problem has been used in a great many applications in computational neuroscience, machine learning, computer vision and image processing. Most of the time people use heuristics without rigorous analysis of running time or correctness. There has been some rigorous work using a method known as ”Independent Component Analysis”, but that method makes quite strong assumptions on the distribution $\{x\}$ (namely independence). Lately, starting with the Spielman-Wang-Wright paper mentioned earlier, there was a different type of rigorously analyzed algorithms, but they all required the vector x to be *very sparse*— less than $\sqrt{n}$ nonzero coordinates. The SOS method allows recovery in the much denser case where x has up to ϵn nonzero coordinates for some $\epsilon > 0$.

Once again this problem has a similar flavor to the ”sparse recovery” problem. In the sparse recovery problem, we know the dictionary A (which is also often assumed to have some nice properties such as being random or satisfying ”restricted isometry property”) and from a single value $y = Ax$ we need to recover x. In the dictionary learning problem we get many examples but, crucially, we know neither A nor x, which makes it a more challenging problem.

4.7.1 Model

First, we will ignore the vector e in (4.13). Morally, the SOS algorithm is naturally robust to noise, and thus these small perturbations change little in the analysis, so we will omit them for simplicity. The simplified problem is already quite interesting.

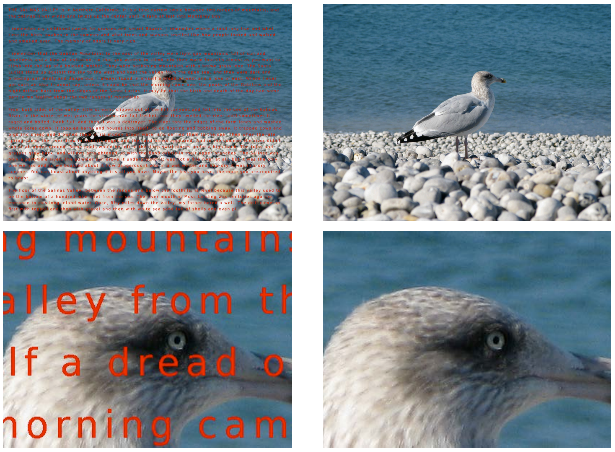


Figure 4.1: Using dictionary learning to remove overlaid text from images. The authors learned a dictionary A from many natural images, and then removed the text from an image y by (roughly) first representing y as $\sum x_k a^k$ and then zeroing out all the x_i's that are below some threshold. Photos taken from: J. Mairal, F. Bach, J. Ponce, and G. Sapiro. *Online Dictionary Learning for Sparse Coding.* In ICML 2009 (See also Mairal, Julien, Michael Elad, and Guillermo Sapiro. "Sparse representation for color image restoration." , IEEE Transactions on Image Processing 17.1 (2008): 53-69 for a clearer description of the method as well as some nice images of how the dictionary looks like that should be added to the scribe notes...)

To allow recovery of A, even in the statistical sense, we need to make some assumptions on the distribution $\{x\}$. These assumptions should capture "sparsity". Most rigorous work assumed a hard sparsity constraint, but we will assume a much softer one (as mentioned above). We also make some additional assumptions that are still strictly weaker than those used by most other works (and incomparable to the others). Nevertheless, trying to find the minimal assumptions needed is a great open problem.

Second, we need to make some assumptions on the distribution $\{x\}$ to allow recovery. It will be convenient for us to assume that d is a power of 2. We also will make the following assumption: for some large constant d, we normalize so that $\mathbb{E}\left[x_i^d\right] = 1$ for every i, and then require that for some parameter $\tau = o(1)$

$$\mathbb{E}\left[x_i^{d/2} x_j^{d/2}\right] \leq \tau \tag{4.14}$$

for every $i \neq j$. We will also make the additional condition that x_i is somewhat symmetric around zero, in the sense that for every non-square monomial x^α of degree at most d (i.e., $\sum \alpha_i \leq d$ and there is some i for which α_i is odd)

$$\mathbb{E}\left[x^\alpha\right] = 0 \ . \tag{4.15}$$

Condition (4.14) is essentially minimal, and roughly corresponds to x having at most τn nonzero (or significant) coordinates.

Example 4.13. For example, note that if the distribution $\{x\}$ is obtain by setting τn random coordinates to equal $\pm\tau^{-(1/d)}$ and the rest zero, then indeed $\mathbb{E}\left[x_i^d\right] = 1$ for all i, and if $i \neq j$

$$\mathbb{E}\left[x_i^{d/2} x_j^{d/2}\right] = \left(\tau\tau^{-(1/2)}\right)^2 = \tau.$$

By appealing to the Arithmetic-Mean-Geometric-Mean inequality, one can show that if assume condition (4.14) holds with the RHS equalling τ^{4d} (which tends to zero if τ does) then we get the stronger condition

$$\mathbb{E}\left[x^{\alpha}\right] \leq \tau \tag{4.16}$$

for every degree d monomial x^{α} that is not of the form x_i^d. Thus, we call a distribution $\{x\}$ satisfying (4.16) and (4.15) (d, τ)*-nice.*

Condition (4.15) is morally stronger, and it is not clear that it is essential, but it is still fairly natural. In particular for this problem it is without loss of generality to assume that $\mathbb{E}\left[x_i^k\right] = 0$ for every odd k, and so this can be considered a mild generalization of this condition.

We will also assume that every column of A has unit norm, and the spectral norm σ of $AA^{\top}$ is at most $O(1)$. These are fairly reasonable assumptions as well.

Example 4.14. For example, consider the case when A is the union of 10 orthonormal bases, so that $m = 10n$. Then for any unit vector $v \in \mathbb{R}^n$, we have that $v^T AA^T v = \|A^T v\|_2 = 10$.

Another minor assumption we make is that $\mathbb{E}\left[x_i^{2d}\right] \leq n^{O(1)}$— this is an extremely mild condition and in some sense necessary for recovery, and so we will not speak much of it except in the one place we use it.

Theorem 4.15 (Main Result (quasipoly version))**.** *There are some constants $d \in \mathbb{N}, \tau > 0$ and an quasipoly time algorithm R that given poly(n) samples from the distribution $y = Ax$ outputs unit vectors $\{\tilde{a}_1, \ldots, \tilde{a}_m\}$ that are 0.99 close to $\{a_1, \ldots, a_m\}$ in the sense that for every i there is a j such that $\langle a_k, \tilde{a}_j \rangle^2 \geq 0.99$ and vice versa.*

We should note that the paper has a version that runs in polynomial time while requiring sparsity $\tau = n^{-\delta}$ for arbitrarily small $\delta > 0$.

(See the paper for a version that runs in polynomial time while requiring sparsity $\tau = n^{-\delta}$ for arbitrarily small $\delta > 0$.)

Notes on constants:

- In the more general statement the constants d, τ depend on the accuracy (e.g., 0.99) and on the top eigenvalue of $AA^{\top}$.
- We will think of d as chosen first and then $\tau > 0$ being an extremely small constant depending on d. So for the rest of the analysis we will think of d as some large constant and $\tau = o(1)$.

4.8 Outline of algorithm

The algorithm is very simple: given examples $y_1, \ldots, y_S$ do the following:

1. Construct the polynomial $\tilde{P}(u) = \frac{1}{S}\sum_{i=1}^{S} \langle y_i, u \rangle^d$
2. Run the SOS algorithm to obtain a degree k pseudo-distribution $\{u\}$ satisfying the constraints $\{\|u\|^2 = 1\}$ that maximizes $\tilde{\mathbb{E}}_u\left[\tilde{P}(u)\right]$. The parameter $k = O(\log n)$ would be specified later.
3. Pick $t = O(\log n)$ random (e.g. Gaussian) vectors $w^1, \ldots, w^t$.
4. Compute the matrix M such that $M_{i,j} = \tilde{\mathbb{E}}\left[\prod_{\ell=1}^{t} \langle w_\ell, u \rangle^2 u_i u_j\right]$.
5. Output a random Gaussian vector v such that $\mathbb{E}\left[v_i v_j\right] = M_{i,j}$.

We will prove the following:

Lemma 4.16 (Main Lemma)**.** *Suppose the algorithm outputs v. With probability $n^{-O(1)}$, there exists some i such that $\langle v, a_i\rangle^2 \geq 0.99\|v\|^2$.*

The main lemma says that we can get one vector with inverse polynomial probability. We will also show that we can verify when we are successful and so amplify this probability to as close to 1 as we wish. It is unclear how to use a black box reduction to get from this statement recovery of all vectors, but it is possible to do so by a simple extension of the main ideas of this lemma, see the paper for details. Intuitively, all we do at the next step is add a constraint to the SOS algorithm which enforces that the next output we get is far away from the one we've found.

4.8.1 Proof Outline—Main Ideas

We first give some intuition as to what this algorithm is doing and give an overview of the proof. The first lemma we will need is that $\tilde{P}$ behaves in an interesting way:

Lemma 4.17. *Let $P(u) = \|A^T u\|_d^d$. For $S \geq ???$ then with probability $\geq ???$*

$$P(u) - \tau\|u\|_d^d \preceq \tilde{P}(u) \preceq P(u) + \tau\|u\|_d^d.$$

Recall that $f \preceq g$ simply means that $g - f$ is a sum of squares.

We make a few straightforward but important observations.

First, notice by repeated usage of the AMGM inequality, if $\{u\}$ satisfies $\tilde{\mathbb{E}}_u\left[\|u\|_2^2\right] = 1$ then it also satisfies $\tilde{\mathbb{E}}_u\left[\|u\|_d^d\right] \leq 1$, thus Lemma 4.17 implies that

$$\left|\tilde{\mathbb{E}}_u\left[P(u)\right] - \tilde{\mathbb{E}}_u\left[\tilde{P}(u)\right]\right| \leq \tau, \tag{4.17}$$

so the inequality holds in pseudo-expectation as well.

Second, we see that if u is unit with $P(u) \geq 1$ then it must hold that $\|v\|_d^d \geq 1 - \tau$, but for fixed ϵ, and for τ sufficiently small and d sufficiently large, this implies that there is some i such that $v_k^2 = \langle a^k, u\rangle^2 \geq 1 - \epsilon$. Indeed, otherwise

$$1 - \tau \leq \|v\|_d^d = \sum v_k^d \leq \max_k v_k^{d-2} \sum v_k^2 \leq (1-\epsilon)^d \cdot O(1)$$

and the RHS would be smaller than 1/2 if d is a large enough constant. Importantly, this implies that we have the following:

Corollary 4.18. *There is an oracle so that given a vector u, returns* Accept *if there exists a_k so that $\langle a_k, u\rangle \geq 1 - O(\tau)$, and* Reject *otherwise.*

Proof. Plug the goddamn thing into $\tilde{P}(u)$. □

Finally, if $\{u\}$ is an actual distribution over unit u's with $P(u) \geq 1$ then every vector in the support would have $\langle a^k, u\rangle^2 \geq 1 - \tau$ for some k. We wish to show that this holds even if it is only a pseudo-distribution. This is captured in the following lemma:

Lemma 4.19. *Let $\{u\}$ be the pseudo-distribution returned by step 2 of our algorithm. If $t > c\log m$ is an even integer and c sufficiently large then there exists some k_0 such that*

$$\tilde{\mathbb{E}}_u\left[\langle u, a_{k_0}\rangle^t\right] \geq (1-\tau)^{O(t)}\ . \tag{4.18}$$

At this point, if $\{u\}$ were a real distribution, then we could just sample from it and we'd be happy, since this lemma would also imply that with some nontrivial probability, $\langle u, a_{k_0}\rangle^2 \geq 1 - O(\epsilon)$. However, it's a pseudo-distribution (boo). The normal thing to do here is just to match the first two moments with a Gaussian, then sample form that. However, if we dropped Step 3-4 and simply tried to define $M_{i,j} = \tilde{\mathbb{E}}\left[u_i u_j\right]$ then sample form a distribution matching these two moments, this will not work:

Example 4.20. Let us assume that the psuedo-distribution $\{u\}$ was simply the uniform distribution over $\{\pm a_1, \ldots, \pm a_m\}$. This does satisfy all of our conditions. The first moments of this distribution are all zero, and the second moments are $\mathbb{E}\left[u_i u_j\right] = 0$ if $i \neq j$ and $\mathbb{E}\left[u_i^2\right] = 2\sum_k a_{ki}^2$, so what we get is a random linear combination of the a_k. This will not give us any information about the a^k's (in fact can be shown that without loss of generality this would be simply a random vector in $\mathbb{R}^n$, if for instance $a_i = e_i$ where e_i is the ith standard basis vector).

However, the reweighing we do in step 3-4 has the effect that if we are lucky, it will isolate one of the a_k's. To see why in the case of Example 4.20 note that the matrix M we compute in the particular case above is simply:

$$M = 2\sum f_W(a_k) \cdot (a_k)^{\otimes 2}$$

where for $W = (w_1, \ldots, w_t)$, $f_W(a_k) = \prod_{\ell=1}^t \langle w_\ell, a_k\rangle^2$ and for every vector z, $z^{\otimes 2}$ is the matrix Z such that $Z_{i,j} = z_i z_j$.

Intuitively, the idea is that with some inverse polynomial probability, the correlation of each random Gaussian we pick with a_1 will be twice as much as the correlation it has with every other a_k, and hence since we take $O(\log n)$ of these, the weighting will be heavily skewed to $(a_1)^{\otimes 2}$.

In this lucky case we will have that for every i, j $M_{i,j} = f_W(a_1)a_1(i)a_1(j) \pm o(f_W(a_1)/n)$. Therefore, if we sample a random v such that $\mathbb{E}\left[v_i v_j\right] = M_{i,j}$ then (using $\|a_1\| = 1$), we have

$$\mathbb{E}\left[\|v\|^2\right] = \sum_i M_{i,i} = f_W(a_1) \pm o(n f_W(a_1)/n)$$

and

$$\begin{aligned}
\mathbb{E}\left[\langle v, a_1\rangle^2\right] &= \sum a_1(i)a_1(j)M_{i,j} \\
&= f_W(a_1)\left(\sum_{i,j}(a_1(i)a_1(j))^2 \pm o(1/n)\sum_{i,j} a_1(i)a_1(j)\right) \\
&= f_W(a_1)(1 + o(1/n)\left(\sum a_1(i))^2\right) = f_W(a_1)(1 \pm o(1))
\end{aligned}$$

Thus, if we scale v to a unit vector $\tilde{v}$, we will get that $\langle \tilde{v}, a_1\rangle^2 \geq 1 - o(1)$.

In general, this behavior is captured in the following lemma:

Lemma 4.21. *For any degree k pseudo-distribution $\{u\}$ satisfying $\{\|u\|_2^2 = 1\}$ so that there exists some c so that $\tilde{\mathbb{E}}_u\left[\langle u, c\rangle^t\right] \geq e^{-\epsilon k}$, the sampling procedure described in steps 3-5 outputs a c' with $\langle c, c'\rangle \geq 1 - O(\epsilon)$ with probability $2^{-k/\mathrm{poly}(\epsilon)}$.*

By what we've done above, these three lemmas together prove Lemma 4.16. Since by Corollary 4.18 we have an oracle to check the correctness of a candidate solution, by repeatedly doing this a polynomial number of times, we conclude that with high probability, we will succeed in finding the desired solution. Now all we have to do is prove a bunch of lemmas, which we do in the remaining sections.

4.9 Proof of Lemma 4.17

We restate Lemma 4.17 here for convenience.

Lemma 4.22. *Let $P(u) = \|A^T u\|_d^d$. For $S = \mathrm{poly}(\tau, d)$ then with arbitrarily high probability*

$$P(u) - \tau\|u\|_d^d \preceq \tilde{P}(u) \preceq P(u) + 2\tau\|u\|_d^d.$$

Proof. We first show that we can replace $\tilde{P}$ with its expectation. Recall that $\tilde{P}(u) = \frac{1}{S}\sum_{i=1}^{S}\langle y, u\rangle^d$. Let $Q(u) = \mathbb{E}_y\big[\langle y, u\rangle\big]^d$. Associate to any degree d polynomial $f(x) = \sum_{|\alpha|\le d} c_\alpha x^\alpha$ a matrix $M(f)$ whose rows and columns are indexed by monomials of degree at most $d/2$ (recall d is even), so that for every monomial β_1, β_2 with $|\beta_1|, |\beta_2| \le d/2$,

$$M_{\beta_1,\beta_2} = \frac{1}{T_{\beta_1+\beta_2}} c_{\beta_1+\beta_2}$$

where $T_\alpha = \#\{\alpha_1, \alpha_2 : \alpha_1 + \alpha_2 = \alpha\}$. Then it is straightforward to prove that f is a sum of squares if and only if this matrix M is PSD.

We know that $M(\tilde{P}) \to M(Q)$ in the Frobenius norm as we take more and more samples, since all the monomials will converge, and moreover, we know that if we take $\mathrm{poly}(d, \tau)$ samples, we will have that with high probability, $\|M(\tilde{P}) - M(Q)\|_F \le \tau/2$ and hence in the spectral norm as well, which implies that the matrices $\tau I \pm (M(\tilde{P}) - M(Q))$ are PSD, which by the above is equivalent to the statement that $\pm(\tilde{P} - Q) \preceq \tau\|u\|_d^d$.

We now show that $P(u) \preceq Q(u) \preceq P(u) + \tau\|u\|_d^2$. This combined with what we just proved suffices to complete the proof of Lemma 4.17. Let us open up this expression for Q. Letting $v = A^\top u$ and recalling that $y = Ax$, we have

$$Q(u) = \mathbb{E}_y\big[\langle y, u\rangle^d\big] = \mathbb{E}_y\big[\langle x, A^T, u\rangle^d\big] = \sum_{|\alpha|\le d} \mathbb{E}_x\big[x^\alpha v^\alpha\big]$$

noting that the non-square moments here vanish, and that the moments that have more than one variables are at most τ (both by the niceness assumption we place on $\{x\}$), we can see that

$$\|v\|_d^d \preceq Q(u) \preceq \|v\|_d^d + \tau \sum_{|\beta|\le d/2} v^{2\beta} \preceq \|v\|_d^d + \tau d!(\sum_k v_k^2)^{d/2} \tag{4.19}$$

where the last inequality follows by repeated application of the AMGM inequality. Note that $\sum_k v_k^2 = \|A^\top u\|_2^2 \preceq O(\|u\|_2^2)$ under our assumption that $\sigma = O(1)$, where σ is the largest singular value of A, so if we choose $\tau \le O(\frac{1}{d!}) \cdot O(1)^d$ we obtained the desired result.

□

4.10 Proof of Lemma 4.19

We restate Lemma 4.19 here for convenience.

Lemma 4.23. *Let $\{u\}$ be the pseudo-distribution returned by step 2 of our algorithm. If $t > c\log m$ is divisible by $d-2$ and c sufficiently large then there exists some k_0 such that*

$$\tilde{\mathbb{E}}_u\big[\langle u, a_{k_0}\rangle^t\big] \ge e^{-\epsilon t/d}\,. \tag{4.20}$$

where $\epsilon = O(\tau + \log\sigma + d\log\frac{m}{k})$.

We note that if we assume Marley's conjecture, we can prove something much stronger, which is intuitively what we are trying to replicate with this lemma:

Proposition 4.24. *If the $\{u\}$ returned by our algorithm is a real distribution, then there exists a k_0 so that*

$$\tilde{\mathbb{E}}\left[\langle u, a\rangle^t\right] \geq (1-\tau)^{\Omega(t)} \geq e^{-\tau \cdot \Omega(t)}.$$

Proof. Since every vector in the support of $\{u\}$ is close to *some* k, there exist k_0 such that with probability at least $1/m$, $\langle u, a\rangle^2 \geq 1 - o(1)$. That means that

$$\tilde{\mathbb{E}}\left[\langle u, a\rangle^t\right] \geq \tfrac{1}{m}(1-\tau)^t \geq (1-\tau)^{t-\log m} \geq 1-\tau)^{\Omega(t)}.$$

□

Proof of Lemma 4.19. First notice by Lemma 4.17 we know that there is some $\{u\}$ satisfying $\|u\|_2^2$ with $\tilde{\mathbb{E}}_u\left[\tilde{P}(u)\right] \geq 1-\tau$; take the real distribution which is identically a_1, for instance. Thus, for the pseudo-expectation that we get, this is satisfied as well, and we get that $\tilde{\mathbb{E}}_u\left[\|A^T u\|_d^d\right] \geq 1 - 2\tau$.

This implies by a straightforward averaging argument as above that there exists some k_0 so that

$$\tilde{\mathbb{E}}_u\left[\langle a_{k_0}, u\rangle^d\right] \geq \frac{1}{m}(1-2\tau).$$

To demonstrate this for larger t, we appeal to the following form of Hölder's inequality:

$$(\|v\|_d^d)^{t/d-2} \preceq (\|v\|_2^2)^{t/(d-2)} \cdot \|v\|_t^t$$

which holds whenever t is an integer multiple of $d-2$. If we substitute in $v = A^T u$, and since moreover $\|A^T u\|_2^2 \preceq \sigma \|u\|_2^2$ where $\sigma = O(1)$ and we assume that our pseudo-expectation satisfies $\{\|u\|_2^2 = 1\}$ we obtain by an additional application of Hölder's inequality

$$\sigma^{t/(d-2)}\, \tilde{\mathbb{E}}_u\left[\|A^T u\|_t^t\right] \geq \tilde{\mathbb{E}}\left[\|v\|_d^d\right]^{t/(d-2)} \geq (1-2\tau)^{t/(d-2)} \geq e^{-2\tau t/(d-2)},$$

so $\tilde{\mathbb{E}}_u\left[\|A^T u\|_t^t\right] = e^{-\Omega(t)}$. We are playing fast and loose with constants a little bit, and the big-Oh here hides some dependencies, but it is all morally correct. Then by the same averaging trick as before, we get the desired result. □

4.11 Proof of Lemma 4.21

We again restate the lemma we want to prove here for convenience.

Lemma 4.25. *For any degree k pseudo-distribution $\{u\}$ satisfying $\{\|u\|_2^2 = 1\}$ so that there exists some c so that $\tilde{\mathbb{E}}_u\left[\langle u, c\rangle^t\right] \geq e^{-\epsilon k}$, the sampling procedure described in steps 3-5 outputs a c' with $\langle c, c'\rangle \geq 1 - O(\epsilon)$ with probability $2^{-k/\mathrm{poly}(\epsilon)}$.*

4.11.1 Motivation

Here is a crude argument as to why this should happen with good probability in the case described in Example 4.20, which is roughly what we should expect is the hard case. In this case, for every particular random random vector w, with probability 0.99 that $\max_{i\geq 1}\langle w, a_k\rangle^2 \leq \log m$ but with probability $\exp(-c\log n) = n^{-O(1)}$ we would have that $\langle w, a^k\rangle^2 \geq 2\log m$, and these events are essentially independent if the a^k's are sufficiently close to orthogonal. (In general we can't assume that, but it turns out that this doesn't matter for our final argument.) Hence with $n^{-O(t)} = n^{-O(\log n)}$ probability we would have that for every ℓ and $k > 1$, $\langle w^\ell, a^1\rangle^2 \geq 2\langle w^\ell, a^k\rangle^2$ meaning that for every $k > 1$, $f_W(a^1) \geq 2^t f_W(a^k) = n^2 f_W(a^k)$ if we set $t = 2\log n$.

4.11.2 Less Heuristic Analysis

First, we need to prove the following technical lemma which says that the sampling procedure is well-defined, since the covariance matrices for Gaussians only make sense if they're PSD. Recall that given $\{u\}$, the matrix that we wish to use to produce our Gaussian is defined as $M_{ij} = \tilde{\mathbb{E}}\big[\prod_{\ell=1}^{t}\langle w_\ell, u\rangle^2 u_i u_j\big]$. For any choice of $W = (w_1, \dots, w_t)$ let us define $f_W(u) = \prod_{\ell=1}^{t}\langle w_\ell, u\rangle^2$.

Lemma 4.26. *The matrix M is positive semi-definite.*

Proof. For any $x \in \mathbb{R}^n$, we have

$$\begin{aligned} x^T M x &= \sum_{i,j} M_{ij} x_i x_j \\ &= \sum_{ij} \tilde{\mathbb{E}}\big[f_W(u) x_i u_i u_j x_j\big] = \tilde{\mathbb{E}}\big[f_W(u) \sum_{ij} x_i u_i u_j x_j\big] = \tilde{\mathbb{E}}\big[f_W(u)\langle x, u\rangle^2\big] \geq 0 \end{aligned}$$

as f_W is a square polynomial and so is $\langle x, u\rangle^2$. □

We want to prove that with decent (i.e., $n^{-O(1)}$) probability over the choice of the vectors $W = (w_1, \dots, w_t)$, if we select v that matching the first two moments of

$$\tilde{\mathbb{E}}\big[f_W(u) u^{\otimes 2}\big] \tag{4.21}$$

then it will satisfy

$$\langle v, a\rangle^2 \geq (1 - O(\epsilon))\|v\|^2 \,. \tag{4.22}$$

We will prove that (with some decent probability over the choice of W) condition (4.22) holds in expectation. Since

$$\mathbb{E}\big[\langle v, a\rangle^2\big] = \sum_{ij} \mathbb{E}\big[a_i a_j v_i v_j\big] = \sum_{ij} a_i a_j \,\tilde{\mathbb{E}}\big[f_W(u) u_i u_j\big] = \tilde{\mathbb{E}}\big[f_W(u)\langle u, a\rangle^2\big]$$

and

$$\mathbb{E}\big[\|v\|^2\big] = \sum_{ij} \mathbb{E}_{v_i v_j}\big[=\big] \,\tilde{\mathbb{E}}\big[f_W(u) \sum_{i,j} u_i u_j\big] = \tilde{\mathbb{E}}\big[f_W(u)\|u\|^2\big]$$

as we choose $\{v\}$ to match the second moments of $\{u\}$, this is equivalent to showing that

$$\tilde{\mathbb{E}}\big[f_W(u)\langle u, a\rangle^2\big] \geq 0.99\,\tilde{\mathbb{E}}\big[f_W(u)\|u\|^2\big] = 0.99\,\tilde{\mathbb{E}}\big[f_W(u)\big], \tag{4.23}$$

where the final equality holds because $\{u\}$ satisfies $\{\|u\|^2 = 1\}$. One needs to add an additional argument to show that this actually happens with decent probability, but it is not a very deep one, and so we skip it here— as always, see the paper for details.

If we select a random standard Gaussian vector w then by the rotation invariance of the Gaussian distribution, $\langle w, c\rangle$ is a standard Gaussian (i.e., distributed per $N(0,1)$), and so $\mathbb{E}\big[\langle w, c\rangle^2\big] = 1$ and the probability that $\langle w, c\rangle^2 \geq 11$ equals some Wikipedia-computable constant $p > 0$.

Let A be this event and let $C \geq 10$ be the the expectation of $\langle w, c\rangle^2 - 1$ conditioned on A.

Note that by the rotation invariance of the Gaussian distribution, $\langle w, b\rangle$ is distributed like $N(0, \|b\|)$ for every $b \perp a$ even after conditioning on A.

For every vector unit u, we can write $u = \langle u, c\rangle c + b$ where $b \perp a$ has norm $\sqrt{1 - \langle u, c\rangle^2}$, and so conditioning on A

$$\mathbb{E}_{w|A}\big[\langle u, w\rangle^2\big] = \langle u, c\rangle^2 \,\mathbb{E}_{w|A}\big[\langle c, w\rangle^2 + 1 - \langle u, c\rangle^2\big] = C\langle c, u\rangle^2 + 1$$

Since $w^1, \dots, w^t$ are chosen independently, if we condition on A happening for every ℓ (which would occur with probability $p^t = \exp(-O(\log n))$) then, letting $Q(u) = (C\langle u, c\rangle^2 + 1)^t$,

$$\mathbb{E}_{W|A}\big[\tilde{\mathbb{E}}_u\big[f_W(u)\big]\big] = \tilde{\mathbb{E}}_u\big[Q(u)\big]$$

by linearity, and

$$\mathbb{E}_{W|A}\big[\tilde{\mathbb{E}}_u\big[f_W(u)\langle u, c\rangle^2\big]\big] = \tilde{\mathbb{E}}_u\big[Q(u)\langle u, c\rangle^2\big]$$

So, we just need to prove that if $\{u\}$ satisfies our conditions, then

$$\mathbb{E}_u\big[Q(u)\langle u, c\rangle^2\big] \geq 0.99\,\mathbb{E}_u\big[Q(u)\big] \tag{4.24}$$

We will show that (4.24) follows from the assumption that $\tilde{\mathbb{E}}_u\big[\langle u, c\rangle^t\big] \geq e^{-\epsilon k}$. Indeed, write $Q(u) = Q'(u) + Q''(u)$ by expanding the expression $Q(u) = (C\langle u, c\rangle^2 + 1)^t$, and letting $Q'(u)$ contains all the terms where we take $C\langle u, c\rangle^2$ to a power larger than $t/2$ and letting $Q''(u)$ contain the rest of the terms.

First, $\mathbb{E}\big[Q''(u)\big]$ is negligible compared to $\mathbb{E}\big[Q(u)\big]$, since the terms in $Q''(u)$ are each of the form $C^s\langle u, c\rangle^{2s}$ for some $s \leq d/2$ and thus since $C^s\langle u, c\rangle^{2s} \preceq C^s\|u\|^{2s}\|c\|^{2s} \preceq C^s\|u\|^{2s}$ and $\{u\}$ satisfies $\{\|u\|^2 = 1\}$, we have that in pseudo-expectation, each of the at most $\binom{t}{t/2}$ terms in $Q''(u)$ is bounded by $(C+1)^{t/2}$, while $Q(u)$ contains the the much larger term $C^t\,\mathbb{E}\big[\langle u, c\rangle^{2t}\big] \geq 0.999^{2t}C^t$.

Thus we can assume $Q(u) = Q'(u)$, but then

$$\tilde{\mathbb{E}}\big[Q'(u)\langle u, a\rangle^2\big] \geq (1 - O(\epsilon))\,\tilde{\mathbb{E}}\big[Q'(u)\big]$$

since we can show this ratio holds for every term of $Q'(u)$ since for every $k \geq t$

$$\tilde{\mathbb{E}}\big[\langle u, a\rangle^{k+2}\big] \geq \tilde{\mathbb{E}}\big[\langle u, a\rangle^k\big]^{(k+2)/k} = \tilde{\mathbb{E}}\big[\langle u, a\rangle^k\big]\,\tilde{\mathbb{E}}\big[\langle u, a\rangle^k\big]^{2/k} \geq (1 - O(\epsilon))\,\tilde{\mathbb{E}}\big[\langle u, a\rangle^k\big]$$

where the first inequality uses Hölder's inequality for psuedo-expectations and the last uses our assumption (4.20).

This concludes the proof of the Main Lemma for actual distributions.

Chapter 5

Sparsest Cut and the ARV algorithm

At MIT this was a guest lecture given by Jonathan Kelner for which lecture notes are not available. The following are notes from the SOS summer course scribed by Tal Wagner.

In this lecture we revisit the (Uniform) Sparsest Cut problem, a thoroughly studied optimization problem introduced in Lecture 2. We present a breakthrough result of Arora, Rao and Vazirani [**?**], that achieves an efficient $O\left(\sqrt{\log n}\right)$-approximation algorithm. It is of interest to us since it can be interpreted as an instantiation of the degree-4 SoS algorithm.

We will not give the full proof of [**?**], since it is very long and involved, but rather focus on presenting the main ideas.

5.1 Problem Definition

For simplicity we define the problem over regular graphs, and remark that the results presented here extend to general graphs as well.

Definition 5.1 (sparsity)**.** Let $G(V, E)$ be an r-regular graph on n vertices. For a subset of vertices $S \subset V$ such that $S \neq \emptyset, V$, denote $\bar{S} = V \setminus S$, and denote by $\mathcal{E}(S, \bar{S})$ the number of edges crossing the cut $(S, \bar{S})$.

The sparsity of the cut $(S, \bar{S})$, denoted $\phi(S, \bar{S})$, is defined as

$$\phi(S, \bar{S}) = \frac{n\mathcal{E}(S, \bar{S})}{r|S||\bar{S}|}.$$

The sparsity of G, denoted $\phi(G)$, is defined as

$$\phi(G) = \min_{S \subset V : S \neq \emptyset, V} \phi(S, \bar{S}).$$

To get some intuition for this definition, observe that up to the multiplicative factor n/r (which can be thought of as normalization), $\phi(S, \bar{S})$ is the ratio of the number of edges that actually cross the cut to the number of edges that *could have* crossed it (had all the possible edges been present).

Definition 5.2. The Uniform Sparsest Cut problem is, given a d-regular graph $G(V, E)$ over n vertices, to find $S \subset V$ such that $\phi(S, \bar{S}) = \phi(G)$.

5.2 Main Theorem

The main result of this lecture is an $O\left(\sqrt{\log n}\right)$-approximation for Uniform Sparsest Cut.

Theorem 5.3 (Arora-Rao-Vazirani [**?**]). *There is a randomized polynomial time algorithm, that given an r-regular graph $G(V,E)$ on n vertices, finds with high probability $S \subset V$ such that $\phi(S,\bar{S}) = O(\sqrt{\log n}) \cdot \phi(G)$.*

The best multiplicative approximation factor prior to [**?**] was $O(\log n)$ due to Leighton and Rao [**?**]. The Cheeger-Alon-Milman inequality, discussed in Lecture 2, achieves a square-root approximation, i.e. finds $S \subset V$ such that $\phi(S,\bar{S}) = O(\sqrt{\phi(G)})$.[1]

On the other hand, while Uniform Sparsest Cut is known to be NP-hard [**?**], all currently known inapproximability results rely on stronger hardness assumptions.

5.3 The ARV Algorithm

We will work under the simplifying assumption that $\phi(G)$ is attained by a bisection - that is, there is $S^* \subset V$ with $|S^*| = n/2$ such that $\phi(G) = \phi(S^*, \bar{S^*})$. Of course, this assumption is not required for the analysis of [**?**].

We identify our vertex set V with $[n] = \{1,\ldots,n\}$. Let $x^* \in \{\pm 1\}^n$ be the indicator vector of S^*, i.e. for all $i \in V$, $x^*_i = 1$ if $i \in S^*$ and $x^*_i = -1$ otherwise. Observe that we have

$$\mathcal{E}(S^*, \bar{S^*}) = \frac{1}{4} \sum_{\{i,j\} \in E} (x^*_i - x^*_j)^2.$$

Moreover since $(S^*, \bar{S^*})$ is a bisection we have $|S^*| = |\bar{S^*}| = \frac{n}{2}$, and since G is r-regular we have $|E| = \frac{nr}{2}$. Plugging these into Definition 5.1, we may write

$$\phi(G) = \frac{1}{2|E|} \sum_{\{i,j\} \in E} (x^*_i - x^*_j)^2,$$

and now we can formulate our problem in the SoS framework.

5.3.1 The SoS Program

Our algorithm (that will be fully described in the next section) runs the SoS algorithm to get a degree-4 pseudo-distribution $\{x\}$ satisfying the constraints:

- $x_i^2 = 1 \quad \forall\, i \in [n];$ (5.1)
- $\sum_{i=1}^{n} x_i = 0;$ (5.2)
- $\frac{1}{2|E|} \sum_{\{i,j\} \in E} (x_i - x_j)^2 \le \phi.$ (5.3)

The constraints in eq. (5.1) ensure that the pseudo-distribution is over vectors in $\{\pm 1\}^n$. The constraint eq. (5.2) ensures that the pseudo-distribution is over bisections (same number of +1's

[1] As explained in Lecture 2, the Cheeger-Alon-Milman inequality applies to a slight variation of Uniform Sparsest Cut of which optimum cannot exceed 1, and hence a square-root approximation makes sense, i.e. $\phi(G) \le \sqrt{\phi(G)}$.

and -1's). Eq. (5.3) is the minimization of our objective function: we can perform a binary search in order to find the minimal ϕ for which these constraints are satisfiable (this is a standard reduction of optimization to feasibility). Note that $\phi \leq \phi(G)$, since we have x^* that attains $\phi(G)$.

Our goal is to show that we can obtain from $\{x\}$ a subset $S \subset V$ that meets our approximation requirement. We make one more assumption, that our pseudo-distribution is "well spread", in the sense that

$$\mathbb{E}_{i,j\in[n]}\left[\tilde{\mathbb{E}}\left[(x_i - x_j)^2\right]\right] \geq \frac{1}{10}. \tag{5.4}$$

It turns out that in the complementary case, it is not difficult to obtain a constant-factor approximation (and this was used prior to ARV). Hence this assumption focuses us on the challenging case.

5.3.2 The Algorithm

To obtain an approximate solution from $\{x\}$, our approach is by reduction to a vertex separation problem on graphs.

Definition 5.4. Let H be a graph over n vertices. We say H is *separable* if there are disjoint subsets of vertices L, R, such that $|L|, |R| \geq \Omega(n)$ and there are no edges crossing between them.

The key structural result of ARV is the following.

Lemma 5.5 (ARV main lemma)**.** *Let $\{x\}$ be a pseudo-distribution obtained from the SoS program in Section 5.3.1 on the input graph $G(V, E)$.*

Put $\Delta = c/\sqrt{\log n}$ for a sufficiently small constant c. Define the graph $H(V, E_H)$ on the same vertex set as G, with $\{ij\} \in E_H$ iff $\tilde{\mathbb{E}}\left[(x_i - x_j)^2\right] < \Delta$.

Then H is separable, and the subsets L, R can be found with high probability in randomized polynomial time.

Using this lemma we can obtain a sparse cut using standard techniques.

Lemma 5.6. *Given subsets L, R as guaranteed by Lemma 5.5, we can efficiently find $S \subset V$ such that $\phi(S, \bar{S}) \leq O(\phi/\Delta) = O(\sqrt{\log n})\phi$.*

Now we can fully state the algorithm.

ARV approximation algorithm for Uniform Sparsest Cut:

1. Solve the SoS program stated in Section 5.3.1.
2. Construct the graph H as described above.
3. Apply Lemma 5.5 to find disjoint subsets $L, R \subset V$ as in Definition 5.4.
4. Use Lemma 5.6 to obtain from L, R a subset $S \subset V$ such that $\phi(S, \bar{S}) \leq O(\sqrt{\log n})\phi(G)$.

5.4 Analysis

We will perform the analysis under the assumption that $\{x\}$ is an actual distribution, and later verify that all arguments used hold remain intact when $\{x\}$ is a pseudo-distribution (of degree 4, in this case). This is fairly common when working with the SoS algorithm.

5.4.1 Preliminaries

Suppose that $\{x\}$ is an actual distribution over vectors in $\{\pm 1\}^n$. The entries of a vector drawn from $\{x\}$ are correlated random variables $x_1, \ldots, x_n$ taking values in $\{\pm 1\}$. We visualize $\{x\}$ as a $\{\pm 1\}$-matrix $A_{\{x\}}$ of size $\ell \times n$, with columns corresponding to $x_1, \ldots, x_n$ and rows corresponding to the points in the sample space of $\{x\}$ (so ℓ is the size of the sample space). The x_i's can now be thought of as (column) vectors in $\{\pm 1\}^\ell$.[2] We stress that $\{x\}$ is a distribution over vectors in $\{pm1\}^n$ describing bisections in G (and corresponding to rows of $A_{\{x\}}$), while its random coordinates x_i's are interpreted as vectors in $\{\pm 1\}^\ell$ (corresponding to columns of $A_{\{x\}}$). The reader is alerted to avoid confusion.

We go on to define a notion of distance between the x_i's. For $t = 1, \ldots, \ell$, let p_t denote the probability to sample from $\{x\}$ the value of the t^{th} row in $A_{\{x\}}$. For simplicity one may think of $p_1, \ldots, p_\ell$ as the uniform distribution; it does not change the analysis.

Definition 5.7. Define $d : [n] \times [n] \to \mathbb{R}_{\geq 0}$ by

$$d(i,j) = \sum_{t=1}^{\ell} p_t (x_i(t) - x_j(t))^2 = \mathbb{E}\left[(x_i - x_j)^2\right],$$

It straightforward to verify that d is a *distance function*, in the sense that it satisfies the following properties:

1. $d(i,i) = 0$ for all i;

2. (Symmetry) $d(i,j) = d(j,i)$ for all i, j;

3. (Triangle inequality) $d(i,k) \leq d(i,j) + d(j,k)$ for all i, j, k.

In fact, d is equivalent to both the Hamming distance and the $\|\cdot\|_1$-distance.[3]

5.4.2 Why is $\Delta \ll 1/\sqrt{\log n}$ Necessary?

Before turning to the main part of the analysis, let us show why our approach can only work up to $\Delta \ll 1/\sqrt{\log n}$ and not greater values, which would have given a better approximation factor. Note that we are showing this limitation even for actual distributions (rather than just pseudo-distributions).

We rely on the following theorem, stated here without proof.

Theorem 5.8 (expansion of the boolean hypercube)**.** *Let $c > 0$ be a sufficiently large constant. For $L \subset \{\pm 1\}^\ell$, denote*

$$\tilde{L} = \left\{ u \in \{\pm 1\}^\ell : \exists u \in L \;\; s.t. \;\; \|v - u\|_1 \leq c\sqrt{\ell} \right\}.$$

If $|L| \geq \Omega(2^\ell)$, then $|\tilde{L}| \geq (1 - o(1))2^\ell$.

[2] We remark in the usual presentation of the ARV analysis, which is outside of the SoS framework, the vectors are not assumed to have entries in $\{\pm 1\}$.

[3] The equivalence is up to normalization and weighting by $p_1, \ldots, p_\ell$, and up to a factor of 4 (for Hamming distance) or 2 (for $\|\cdot\|_1$-distance..

Put $\ell = \log n$, and let A be a $\{\pm 1\}$-matrix of dimensions $\ell \times n$, such that its n columns are all the $2^\ell = n$ possible $\{\pm 1\}$-vectors of length ℓ. (The order of the columns does not matter.) Consider the distribution $\{x\}$ defined by uniformly sampling a row from A. Note that for this distribution, A is exactly the matrix $A_{\{x\}}$ defined in the previous section.

Since $\{x\}$ is uniform over its support, it can be easily seen that

$$d(i,j) = \mathbb{E}\big[(x_i - x_j)^2\big] = \frac{\|x_i - x_j\|_1}{2\ell}.$$

Recall that we have $\Delta = c/\sqrt{\log n} = c/\sqrt{\ell}$. In the graph H defined in Lemma 5.5, a pair of vertices i, j is adjacent if $d(i,j) < \Delta$, or equivalently (by the above), if $\|x_i - x_j\|_1 < 2\Delta\ell = 2c\sqrt{\ell}$. If we take c to be a large constant (rather than a small constant as in Lemma 5.5), then we can apply Theorem 5.8. It tells us that for any choice of L such that $|L| \geq \Omega(n)$, the subset R of vertices that have no neighbors in L would have size no larger than $o(n)$. This means that H is not separable, and the conclusion of Lemma 5.5 does not hold.

5.4.3 Why is $\Delta \ll 1/\sqrt{\log n}$ Sufficient? Proof of Lemma 5.5

We now turn to the core of the analysis, of showing that if $\Delta = c/\sqrt{\log n}$ then H is separable (with constant probability, that can then be boosted).

Apply the Quadratic Sampling Lemma (see previous lecture for the formal statement and proof) to obtain a Gaussian distribution $\{y\}$ that matches the first two moments of $\{x\}$. This is a distribution over vectors in $\mathbb{R}^n$ (with correlated entries), such that y_i is associated with the vertex i in H. Sample $y \sim \{y\}$ and define

$$L = \{i : y_i < -10\} \ \ , \ \ R = \{i : y_i > 10\}.$$

By the SoS program constraints, for each i we have $\mathbb{E}\big[y_i\big] = \mathbb{E}\big[x_i\big] \in [-1,1]$ and $\mathbb{E}\big[y_i^2\big] = \mathbb{E}\big[x_i^2\big] = 1$. Hence by standard properties of Gaussian random variables, we see that with high probability, $|L|, |R| \geq \Omega(n)$. If there are no edges crossing between L and R, then we are done. Otherwise, we wish to remove a small subset of vertices from L and R in a way that eliminates all the edges crossing between them, but retains their linear sizes. For convenience, we treat all edges crossing between L and R as oriented edges from L to R.

To analyze this approach, let us derive a simple bound on the probability of an edge to cross from L to R. Let $\{ij\}$ be some edge in H. By definition this means that $d(i,j) = \mathbb{E}\big[(x_i - x_j)^2\big]\Delta$. Moreover we know that $\mathbb{E}\big[y_i - y_j\big] = \mathbb{E}\big[x_i - x_j\big] = \in [-2,2]$. Hence $y_i - y_j$ is a Gaussian random variable with mean in $[-2,2]$ and variance bounded by Δ, and therefore,

$$\Pr\left[|y_i - y_j| > 20\right] = \exp(-1/\Delta). \tag{5.5}$$

This means that the edge $\{ij\}$ crosses from L to R with probability at most $\exp(-1/\Delta)$.

Now let us consider some examples of how we can avoid edges crossing from L to R.

- Suppose Δ is as small as $\Delta \leq 1/3 \log n$. Then the probability in eq. (5.5) is roughly $1/n^3$, which means, by a union bound over all edges in H (of which there are at most n^2), that with constant probability there are no edges crossing from L to R. In this case the proof is finished.

- Suppose H has at most $2^{O(\sqrt{\log n})}$ edges. Recalling our choice of Δ as $c/\sqrt{\log n}$ for a sufficiently small constant c, we see that we can apply the same union bound argument as above.

-

5.4.4 From Vertex Separation to Sparse Cut: Proof of Lemma 5.6

We now prove Lemma 5.6. The proof is standard and is outside the core analysis of ARV (which is Lemma 5.5); it is presented here for the sake of completeness.

Relying on the definition of d as a distance function between points in $[n]$, we can define a notion of distance between a point and a subset of points: For $B \subset [n]$ and $i \in [n]$, we let

$$d(B,i) = \min_{b \in B} d(b,i).$$

It is easy to verify that from the standard triangle inequality of d, we can obtain the following triangle inequality for subsets: For $B \subset [n]$ and $i,j \in [n]$,

$$d(B,i) \leq d(B,j) + d(j,i). \tag{5.6}$$

To obtain our sparse cut, we sample $\tau \in (0,\Delta)$ uniformly at random (recall that Δ was defined in Lemma 5.5) and let

$$S = \{i \in [n] : d(L,i) \leq \tau\}.$$

Clearly we have $L \subset S$. Observe that in Lemma 5.5, the graph H was defined such that i,j are neighbours in H iff $d(i,j) < \Delta$. By hypothesis of Lemma 5.6 there are no edges crossing between L and R, and therefore $R \subset \bar{S}$. Since $|L|, |R| \geq \Omega(n)$, we conclude that

$$|S| \cdot |\bar{S}| \geq \Omega(n^2). \tag{5.7}$$

We turn to counting the edges crossing the cut $(S,\bar{S})$. Let $\{ij\}$ be an edge in G and let χ_{ij} be the 0-1 random variable indicating whether the edge crosses $(S,\bar{S})$. Suppose w.l.o.g. that $d(L,i) \leq d(L,j)$. The edge crosses $(S,\bar{S})$ iff both $\tau \geq d(L,i)$ and $\tau < d(L,j)$ occur, and hence

$$\mathbb{E}_\tau\big[\chi_{ij}\big] = \Pr[\tau \in [d(L,i), d(L,j))] = \frac{d(L,j) - d(L,i)}{\Delta} \leq \frac{d(i,j)}{\Delta},$$

where the final inequality is by eq. (5.6). Summing over all edges,

$$\mathbb{E}_\tau\big[\mathcal{E}(S,\bar{S})\big] = \mathbb{E}_\tau\Big[\sum_{\{i,j\} \in E} \chi_{ij}\Big] = \sum_{\{i,j\} \in E} \mathbb{E}_\tau\big[\chi_{ij}\big] \leq \frac{\sum_{\{i,j\} \in E} d(i,j)}{\Delta},$$

and now using Markov's inequality we can find with high probability a threshold τ such that

$$\mathcal{E}(S,\bar{S}) \leq O(1) \cdot \frac{\sum_{\{i,j\} \in E} d(i,j)}{\Delta}. \tag{5.8}$$

Finally, note that by eq. (5.3) we have

$$\sum_{\{i,j\} \in E} d(i,j) = \sum_{\{i,j\} \in E} \mathbb{E}\big[(x_i - x_j)^2\big] \leq 2|E|\phi = 2nr\phi.$$

Combining this with eq. (5.7) and eq. (5.8) we find that

$$\phi(S,\bar{S}) = \frac{n\mathcal{E}(S,\bar{S})}{r|S||\bar{S}|} \leq O(1) \cdot \frac{\phi}{\Delta},$$

as required.

5.4.5 From Actual Distribution to Pseudo-Distribution

5.5 Temp Graveyard

5.5.1 Squared Triangle Inequality for $\{\pm 1\}$

The reason we need a degree-4 SoS program, is for $\{x\}$ to have the following property.

Lemma 5.9. *Let $\{x\}$ be a degree-4 pseudo-distribution satisfying the constraints $\{\forall i, \quad x_i^2 = 1\}$. Then for all i, j, k,*

$$\tilde{\mathbb{E}}\big[(x_i - x_k)^2\big] \le \tilde{\mathbb{E}}\big[(x_i - x_j)^2\big] + \tilde{\mathbb{E}}\big[(x_j - x_k)^2\big]. \tag{5.9}$$

Proof. We first note that if $\{x\}$ was an actual distribution over $\{\pm 1\}^n$ then the lemma would be easy, since the inequality $(x_i - x_k)^2 \le (x_i - x_j)^2 + (x_j - x_k)^2$ would hold for every i, j, k (this is trivial to verify by case analysis) and hence would clearly hold in expectation. However, in order to prove the lemma for pseudo-expectation, we need to give an SOS proof.

By linearity of pseudo-expectation, eq. (5.9) is equivalent to

$$\tilde{\mathbb{E}}\big[(x_i - x_j)^2 + (x_j - x_k)^2 - (x_i - x_k)^2\big] \ge 0.$$

By rearranging, this becomes

$$\tilde{\mathbb{E}}\big[(x_j - x_i)(x_j - x_k)\big] \ge 0.$$

Denote $P(x) = (x_j - x_i)(x_j - x_k)$. We need to show that $\tilde{\mathbb{E}}\big[P(x)\big] \ge 0$, so by definition, we need to find a polynomial $Q(x)$ such that $P = Q^2$ (over $\{\pm 1\}$).

We put $Q = \frac{1}{2}P$. The fact that $P = (\frac{1}{2}P)^2$ can be verified either by explicitly expanding Q^2 and plugging $x_i^2 = x_j^2 = x_k^2 = 1$, or by just observing that $P(x) \in \{0, 4\}$ over $\{\pm 1\}$, which renders $P = (\frac{1}{2}P)^2$ immediate to see.

Note that our SoS proof Q^2 of eq. (5.9) is a polynomial of degree 4, and this is why we need $\{x\}$ to be a pseudo-distribution of atleast this degree. □

Chapter 6

The SOS approach to refuting the Unique Games Conjecture

Morally speaking, the Unique Games Conjecture (UGC) asserts that a simple algorithm— namely the degree 2 SOS program— is the *optimal efficient algorithm* for a wide range of optimization problems. Thus if the UGC is true one might expect that the SOS hierarchy is very often *useless*— there is no point in going beyond the degree two case unless you go for a large degree that would amount to the exponential-time brute force algorithm. We do not know whether the UGC as stated is true, but we will see that this strong version of it is false— we have already seen examples where we can get non-trivial improvements by the SOS algorithm with moderate degree, and as we will see today, we can get such guarantees even for the Unique Games problem itself. Namely, the SOS algorithm yields a sub-exponential ($2^{n^{\epsilon}}$) time algorithm for the Unique Games problem. While falling short of disproving the UGC, this algorithm does significantly outperform the trivial brute-force algorithm.

The SOS hierarchy represents the most promising approach I know of towards refuting the UGC. Progress towards this goal has been somewhat slow, with papers that show the algorithm works for particular instances, or work for general instances but with parameters that are far from what's needed to refute the UGC. But it has also been steady, and an algorithm-optimistic (or complexity-pessimistic) view towards it is that the problem might eventually succumb by the Grothendieck "Rising Sea" method:

> I can illustrate the second approach [to solving a problem] with the same image of a nut to be opened. The first analogy that came to my mind is of immersing the nut in some softening liquid, and why not simply water? From time to time you rub so the liquid penetrates better, and otherwise you let time pass. The shell becomes more flexible through weeks and months— when the time is ripe, hand pressure is enough, the shell opens like a perfectly ripened avocado!
>
> A different image came to me a few weeks ago. The unknown thing to be known appeared to me as some stretch of earth or hard marl, resisting penetration. . . the sea advances insensibly in silence, nothing seems to happen, nothing moves, the water is so far off you hardly hear it. . . yet it finally surrounds the resistant substance.

Another, perhaps slightly less classy metaphor that comes to my mind is the Austin Powers steamroller, though at this point it's still unclear if the SOS steamroller will stop short of running over the UGC...

I should note that the SOS hierarchy also plays an important role in the most promising approach I know of to *prove* the conjecture. This is the work of Khot and Moshkovitz which Dana discussed in our reading group and I will refer to briefly today. I do believe in the "rising sea" approach in the sense that, true or false, settling the UGC will eventually be only a minor part of a much larger theory that will give us a broad understanding of the power of the SOS algorithm, and of efficient algorithms in general, for many classes of optimization problems.

6.1 The Small Set Expansion Hypothesis

There is actually a family of problems related to the Unique Games Conjecture. All these problems share the feature that the best known approximation algorithm for them is the degree 2 SOS and that we have no proof one can't do better. The three most prominent problems in this class are:

- $SSE(\epsilon)$: Distinguishing, given a d-regular graph $G = (V, E)$ of n vertices, between the YES case where there exists a set S of $n/\log n$ vertices such that $|E(S, \overline{S})| \leq \epsilon d|S|$ and the NO case where every set S of at most $n/\log n$ vertices satisfies $|E(S, \overline{S})| \geq (1-\epsilon)d|S|$.

- $UG(\epsilon)$: Distinguishing, given a set of linear equations over n variables taking values in $\mathbb{Z}_{\log n}$ such that each equation only involves two variables, between the YES case where there exists an assignment to the variables that satisfies a $1-\epsilon$ fraction of the equations, and the NO case where every assignment satisfies at most ϵ fraction of them.[1]

- $2LIN(\epsilon)$: Distinguishing, given a set of linear equations over n variables taking values in $\mathbb{Z}_2$ such that each equation only involves two variables, between the YES case where there exists an assignment to the variables that satisfies a $1-\epsilon$ fraction of the equations, and the NO case where every assignment satisfies a $1-\sqrt{\epsilon}/10$ fraction of them. (There are also two extremely related problems to $2LIN(\epsilon)$— $MAXCUT(\epsilon)$ and $SPARSESTCUT(\epsilon)$ which should arguably be added to this list as well.)

In all cases we think of ϵ as a small constant tending to zero (e.g., think $\epsilon = 0.01$ or $\epsilon = 0.001$). We have the following relation between these problems

$$SSE(\epsilon) \preceq UG(\epsilon) \preceq 2LIN(\epsilon)$$

by which we mean that there is a polynomial-time reduction from $SSE(\epsilon)$ to $UG(\epsilon')$ (for some ϵ' depending on ϵ and tending to 0 with ϵ) and from $UG(\epsilon)$ to $2LIN(\epsilon')$. Reductions in the other directions are not known, but current knowledge suggests that all three problems are likely to be computationally equivalent. In particular all known algorithmic and hardness results apply equally well to the $SSE(\epsilon)$ and $UG(\epsilon)$. The best polynomial-time algorithm known for all three problems is the degree 2 SOS algorithm. In particular for SSE this algorithm corresponds to a generalization of Cheeger's Inequality, while for $2LIN$ it corresponds to a (small) generalization of the Goemans-Williamson Max-Cut algorithm. The version of $2LIN(\epsilon)$ of distinguishing between value $1-\epsilon$ vs. $1-1.01\epsilon$ is known to be NP-hard (for some value of 1.01) and in fact this reduction has quasilinear blowup and so under standard assumptions this problem cannot be solved in $2^{n^{0.999}}$ time.

The *Unique Games Conjecture* (UGC) asserts that for every $\epsilon > 0$, $UG(\epsilon)$ is NP hard. The *Small Set Expansion Hypothesis* (SSEH) asserts that for every $\epsilon > 0$, $SSE(\epsilon)$ is NP hard. One can also phrase a *2LIN hypothesis* (2LINH) that for every $\epsilon > 0$, $2LIN(\epsilon)$ is NP-hard. Given

[1] For every $c < 1/2$, the version of the $UG(\epsilon)$ where we compare $1-\epsilon$ vs $1-\epsilon^c$ is known to be equivalent to $UG(\epsilon')$ (for ϵ' related to ϵ) using Rao's parallel repetition theorem.

the discussion above, the SSEH implies the UGC and is very likely to be equivalent to it. Since it also implies all consequences of the UGC (including the 2LINH), the SSEH is a natural anchor for the problems in the "Unique Games Sphere" and so a natural object of study towards refuting the UGC. Conversely, the 2LINH is a natural object of study towards proving the UGC (which is indeed the Khot-Moshkovitz approach).

Note: All these problems are typically stated with more parameters than ϵ, and we fixed the other parameter to be $\log n$ in SSE and UG for simplicity. This version can be shown to be equivalent to the more standard version by known reductions.

6.2 2 to q norm and small set expansion

The main current approach to the attacking the small-set expansion problem via SOS goes through *hyper-contractive norms*. Specifically we will use the following result. Informally, we call a d-regular graph $G = (V, E)$ a *small set expander* if subsets S of size $o(|V|)$ satisfy $|E(S, \overline{S})| \geq (1 - o(1))d|S|$.

Theorem 6.1 (Informal). *For every even $q > 2$, a graph G is a* small set expander *if and only if every vector w in G's top eigenspace satisfies*

$$\mathbb{E}_i\, w_i^q \leq O(\mathbb{E}\, w_i^2)^{q/2} \tag{6.1}$$

Thus a sufficiently good approximation algorithm for (6.1) would yield an algorithm for the small set expansion problem. This approach seems to be extremely ambitious in the sense that we try to approximate the ratio of the q and 2 norms over an *arbitrary* subspace W, forgetting any additional structure W may have had since it is the top eigenspace of some graph. However, this makes the problem also cleaner and presumably, if there are hard instances for it, then it would be easier to find them.

Keeping to a rough, informal level, the proof that $O(1)$ levels of the SOS algorithm solve all previously known hard instances rely on the fact that it can certify (6.1) for the subspace of low degree polynomials over $\{\pm 1\}^{\log n}$. With Kelner and Steurer, we showed that $O(1)$ rounds of SOS yield an $\dim(W)^{1/3}$ approximation (in some precise sense) of the 2 to 4 problem. If this result could be improved to a constant or even *polylog*(n) (perhaps even $n^{o(1)}$), even at the expense of using *polylog*(n) (or perhaps even $n^{o(1)}$) rounds this would refute the SSEH and likely can be extended to refute the UGC as well.

We will now give an informal intuition behind the theorem, and sketch how it implies a sub-exponential algorithm for small-set expansion (that can be extended to the Unique Games problem as well), and then describe in full the $\dim(W)^{1/3}$ algorithm. (Depending on time, we may or may not cover the full proof of Theorem 6.1.)

6.2.1 The relation between the 2 to 4 norm and small set expansion

We now give some intuition on the relation between the 2 to q norm and small set expansion. For simplicity, we focus on the case $q = 4$ although it is not hard to see that the same intuition holds for every even $q > 2$. In the second lecture we saw the following results that says that if $\mathbb{E}\, w_i^4 \leq O(\mathbb{E}\, w_i^2)^2$ for every w in the top eigenspace of G then G is a small set expander:

Lemma 6.2. *Let $G = (V, E)$ be regular graph, $\lambda \in (0, 1)$ and W be the span of eigenvectors of G's normalized adjacency matrix corresponding to eigenvalue at least $1 - \lambda$. If every $w \in W$ satisfies:*

$$\mathbb{E}_i\, w_i^4 \leq C \left(\mathbb{E}_i\, w_i^2\right)^2 \tag{6.2}$$

then for every set S of measure δ set,

$$\phi(S) \geq \lambda(1 - \sqrt{C\delta})$$

The proof of the lemma (which we saw) is a fairly straightforward contrapositive argument— if S is a set of $o(1)$ measure that doesn't expand, then the projection of 1_S to the top eigenspace will still have large 4 norm compared to its 2 norm.

The other direction— transforming a vector w in the top eigenspace with large 4 to 2 norm ratio into a small set that doesn't expand— is trickier. It is instructive to compare this with Cheeger's Inequality. The difficult direction of Cheeger's Inequality transforms a vector w in the top eigenspace (but orthogonal to the all 1's vector) into a set of measure at most 1/2 that does not expand. In fact, Cheeger doesn't need the vector w to be an eigenvector at all. As long as

$$w^\top G w \geq (1 - \epsilon)\|w\|^2 \tag{6.3}$$

the transformation of Cheeger (which involves choosing a random threshold τ with probability proportional to τ^2 and taking the set of all coordinates of w that are larger than τ) will yield such a set.

One could hope that if w satisfies $\mathbb{E}\, w_i^4 \gg (\mathbb{E}\, w_i^2)^2$ then by using the same or a similar transformation we can get a *small* set that doesn't expand. However, this is a bit tricky— in particular we cannot do so by only assuming (6.3), since it is trivial to modify every vector w to have high 4 norm without hurting (6.3) too much. For example if w satisfies $\mathbb{E}\, w_i^2 = 1$, we can add $n^{0.3}e_1$ to w. This change will make the 4-norm of w huge, but will be negligible in the 2 norm and hence will not hurt (6.3). Therefore, to make the proof go through we must use the fact that w is completely contained in the top eigenspace, as opposed to merely satisfying (6.3).

Intuition for the actual proof. To get some intuition for the proof, lets assume that the graph G is "nice" in the following sense: for every vector w, if w is in the eigenspace of G corresponding to eigenvalues larger than $1 - \epsilon$ then the vector $w^{\odot q}$, defined as $w_i^{\odot q} = w_i^q$, is in the eigenspace corresponding to eigenvalues larger than $1 - q\epsilon$. For example, Cayley graphs over the Boolean cube are "nice":

Exercise 6.1: Prove that if G is a Cayley graph over $GF(2)^\ell$ (i.e., G's vertices are elements of $GF(2)^\ell$ and x is connected to y if $x \oplus y \in S$ for some subset $S \subseteq GF(2)^\ell$) then it is nice. See footnote for hint[2] Can you generalize this to other Cayley graphs?

Now this means that if there is a vector w in G's top eigenspace satisfying $\mathbb{E}\, w_i^4 \gg (\mathbb{E}\, w_i^2)^2$ then the vector $v = w^{\odot 2}$ is also in G's top eigenspace (for a slightly looser definition of "top") and satisfies $\mathbb{E}\, v_i^2 \gg (\mathbb{E}\, |v_i|)^2$. However it turns out that the Cheeger transformation does actually produce a set of measure at most $O(\delta)$ if you apply it to a vector v satisfying $\delta\, \mathbb{E}\, v_i^2 > (\mathbb{E}\, |v_i|)^2$. Indeed, without loss of generality we can normalize so that $\sum |v_i| = 1$, and so we can think of $|v_i|$ as a probability distirbution and this condition means that it has collision probability at least $1/(\delta n)$. Let S be the set of i's such that $|v_i| > 1/(100\delta n)$. Note that $|S| \leq 100\delta n$. Now one can show that if we do the Cheeger transformation then with high probability we will output a subset of S. (For a formal argument see the paper of Dimitriou and Impagliazzo (1998) or the appendix of Arora,Barak Steurer (2010).)

The argument for "non nice" graphs is substantially more complicated. Specifically, rather than giving a simple transformation that takes any w satisfying the conditions into a set S that does not

[2] **Hint:**The eigenvectors for such a graph are always the functions $\{\chi_\alpha\}_{\alpha \in GF(2)^\ell}$ where $\chi_\alpha(x) = -1^{\langle x, \alpha \rangle}$. Note that $\chi_\alpha \odot \chi_\beta = \chi_{\alpha \oplus \beta}$.

expand, the argument needs to assume that w is (close to) the *optimal* vector in the subspace in terms of the relation between its q and 2 norm. The full proof is enclosed below and we will cover it in class based on time constraints.

Exercise 6.2: (Open) Find a simpler proof for Theorem 6.1

Theorem 6.1 immediately implies a *sub-exponential time* algorithm for small set expansion using the following exercise (a version of which for $q = 4$ we've already seen in a previous lecture):

Exercise 6.3: Let q be any even constant and W be a subspace of R^n with dimension $\gg n^{2/q}$. Then there exists $w \in W$ such that $(\mathbb{E}\, w_i^q) \gg (\mathbb{E}\, w_i^2)^2$. See footnote for hint[3]

This yields a sub-exponential algorithm since for $SSE(\epsilon)$ we can take consider the subspace corresponding to eigenvalues larger than $1 - \epsilon$ and so q to be roughly $1/\epsilon$. This means that either G's top eigenspace has dimension at most $O(n^{2/q})$, in which case we can enumerate over (a sufficiently fine net of) it in $\exp(O(n^{2\epsilon}))$ time, or if the dimension of the subspace is higher and then we know that it can't be a small set expander by the combination of this exercise and Theorem 6.1. This algorithm can be extended for the Unique Games problem as well (see Arora, Barak and Steurer, 2010). Note that this means that if we take $\epsilon = 0.01$, we would need to look at graphs of size roughly 2^{50} before this algorithm is slower than a quadratic time one (for $\epsilon = 0.001$ this would be 2^{500}). So, even if the SSEH/UGC are true, they do not seem to tell us very much on inputs that actually fit in the world's storage capacity. This is in contrast to problems such as SAT where, despite progress in SAT solvers, its exponential behavior is in fact quite observable even on relatively modest sized inputs of a few thousand variables or so (not to mention variants of SAT arising from private-key cryptography, where we can see the exponential behaviour even on inputs as small as a few dozen variables— e.g. we still don't know of a much better than brute force algorithm to break the 56-bit cipher DES.) That said, even if the SSEH hypothesis is false, it would be still very interesting to know if a *linear time* algorithm exists for the $SSE(\epsilon)$ problem.

6.3 Using SOS for the 2 to 4 problem

In a previous lecture we saw that SOS can certify that the span of low degree polynomials over the Bolean cube has bounded 4 to 2 norm ratio. We did not show how this implies that the SOS algorithm solves the UG/SSE/Max-Cut problems on previously suggested candidate hard instances (see our STOC 2012 paper for that), but in any case this work was only for *specific* instances. We now describe an approach for *general* instances of the problem.

A variant of Theorem 6.1 shows that for any constants $\beta > \alpha$ and $\delta > 0$, an algorithm for the following problem is sufficient to solve $SSE(\epsilon)$ (with ϵ related to the parameters below— in the application we will let W be the subspace corresponding to eigenvalues of G larger than $1 - O(\epsilon)$): given a subspace $W \subseteq \mathbb{R}^n$, distinguish between the YES case when there is a set S with $|S| \leq n/\beta$ such that the projection of 1_S to W has norm $(1-\delta)\|1_S\|$ and the NO case where for every $w \in W$ such that $\mathbb{E}\, w_i^4 \leq \alpha(\mathbb{E}\, w_i^2)^2$. For simplicity lets consider the version with $\delta = 0$. (This seems potentially easier, but we don't know of any better algorithm than the one for $\delta > 0$.)

Therefore we consider the following problem: given a d-dimensional subspace $W \subseteq \mathbb{R}^n$, distinguish between the YES case: there is a set S with $|S| \leq n/\beta$ such that $1_S \in W$ and the NO case: $\mathbb{E}\, w_i^4 \leq \alpha(\mathbb{E}\, w_i^2)^2$ for every $w \in W$. We identify the approximation factor of this problem with β/α. Ideally we would like an algorithm solving the problem for $O(1)$ approximation factor, but what we

[3] **Hint:**Given an orthonormal basis $w_1, \ldots, w_d$ for W, we want to find some signs $\sigma_1, \ldots, \sigma_d \in \{\pm 1\}$ so that some coordinate i of the vector $w = \sum \sigma_i w_i$ will satisfy $|w_i| \geq \Omega(d/\sqrt{n})$ which would imply $\mathbb{E}\, w_i^q \geq d^q/n^{q/2+1}$, while of course $\|w\|^2 = d$ and so $\mathbb{E}\, w_i^2 = d/n$.

would show is an $O(d^{1/3})$ approximation algorithm (taken from the work with Kelner and Steurer which also handled the case of $\delta > 0$).

Theorem 6.3. *There is some constant c such that if $\beta \geq cd^{1/3}\alpha$, then if there exists a degree 20 psuedo-distribution $\{w\}$ over $\mathbb{R}^n$ satisfying the constraints $w \in W$, $\|w\|^2 = 1$, $w_i^2 = w_i$ for all i, $\mathbb{E}_i w_i^4 \geq \beta(\mathbb{E}\, w_i^2)^2$, then there exists some $v \in W$ satisfying $\mathbb{E}\, v_i^4 \geq \alpha(\mathbb{E}\, v_i^2)^2$. Moreover, we can efficiently find such a v from the moments of $\{w\}$.*

Proof. Let Π be the projector to W and let $\delta^i = \Pi e^i$ where e_i is the i^{th} standard basis vector.

The algorithm will use the combination of the following steps:

Random vector rounding: pick a random $w \in W$.

Projection rounding: try all vectors of the form δ^i.

Conditioning: find $i_1, \ldots, i_4$ and change $\{w\}$ to the distribution where we re-weigh the probability of every vector by a factor of $w_{i_1}^2 \cdots w_{i_4}^4 = w_{i_1} \cdots w_{i_4}$.

Quadratic sampling: sample a gaussian distribution $\{v\}$ that matches the first two moments of $\{w\}$.

We will show that if the first two methods fail, then after conditioning, quadratic sampling will yield a good solution.

If there exists an i such that $\mathbb{E}_j(\delta_j^i)^4 \geq \alpha(\mathbb{E}_j(\delta_j^i)^2)^2$ then we're done, so we can assume that $\mathbb{E}_j(\delta_j^i)^4 \leq \alpha(\mathbb{E}_j(\delta_j^i)^2)^2$ for every $i \in [n]$.

Note that $w_i = \langle w, e^i \rangle = \langle w, \delta^i \rangle$ for every $w \in W$.

Note that since the w's are characteristic vectors of sets of size n/β, $\mathbb{E}_i w_i^4 = 1/\beta$. Thus, by Cauchy-Schwarz

$$\tfrac{1}{\beta} = \tilde{\mathbb{E}}_w \mathbb{E}_i \langle w, \delta^i \rangle^4 \leq \sqrt{\tilde{\mathbb{E}} \langle w, w' \rangle^4 \, \mathbb{E}_{i,j} \langle \delta^i, \delta^j \rangle^4} \tag{6.4}$$

(**Exercise 6.4:** verify this.)

Under our assumption for every i,

$$\mathbb{E}_j \langle \delta^i, \delta^j \rangle^4 = \mathbb{E}_j (\delta_j^i)^4 \leq \alpha \left(\mathbb{E}(\delta_j^i)^2 \right)^2$$

Lets assume the RHS is the same up to a factor of α for every j (**Exercise 6.5:** show that otherwise random vector rounding succeeds). Then we get that

$$\mathbb{E}_i \, \mathbb{E}_j \langle \delta^i, \delta^j \rangle^4 \leq \alpha^2 \left(\mathbb{E}_{i,j} (\delta_j^i)^2 \right)^2$$

but $\delta_j^i = \langle e^j, \Pi e^i \rangle$ and so $\mathbb{E}_{i,j}(\delta_j^i)^2$ is simply $1/n^2$ times the Frobenius norm squared of Π which is d. Hence we get that

$$\mathbb{E}_i \, \mathbb{E}_j \langle \delta^i, \delta^j \rangle^4 \leq \alpha^2 d^2 / n^4$$

Plugging this into (6.4), squaring and dividing both sides by $\alpha^2 d^2 / n^4$ we get that

$$\tilde{\mathbb{E}} \langle w, w' \rangle^4 \geq \tfrac{n^4}{\alpha^2 \beta^2 d^2}$$

since $d = \beta^3 / (c^3 \alpha^3)$ we get

$$\tilde{\mathbb{E}} \langle w, w' \rangle^4 \geq \tfrac{n^4 c^6 \alpha^4}{\beta^8}$$

We now make the following claim, which crucially depends on the fact that $w_i^2 = w_i$ for all i:

CLAIM: There exists $i_1, \ldots, i_4$ such that if we modify the distribution $\{w\}$ by multiplying the probability of every w with $w_{i_1}^2 \cdots w_{i_4}^2$ then

$$\tilde{\mathbb{E}}_{new}\langle w, w'\rangle \geq \left(\tilde{\mathbb{E}}_{old}\langle w, w'\rangle^4\right)^{1/4}$$

Exercise 6.6: prove this claim

The claim implies that

$$\tilde{\mathbb{E}}\langle w, w'\rangle \geq \tfrac{nc\alpha}{\beta^2}$$

Now if we pick v to be a random vector matching the first two moments of $\{w\}$, then we claim that $\mathbb{E}_i v_i^4 \geq \frac{c\alpha}{\beta^2}$ (**Exercise 6.7:** verify this.) but on the other hand $\mathbb{E}\, v_i^2 = \mathbb{E}\, w_i^2 = 1/\beta$, hence concluding the proof. □

Theorem 6.3 and the Khot-Moshkovitz construction. The factor $d^{1/3}$ seems rather arbitrary and it is natural to ask whether by using more rounds we can improve it further, perhaps getting a factor of $d^{\Omega(1/r)}$ for degree-r SOS. This should be sufficient to refute the SSEH and quite possible extended to refute the UGC and 2LINH as well.

In contrast, Khot and Moshkovitz gave a candidate integrality gap for the $2LIN$ problem. Specifically for arbitrarily large constants c, r they construct an instance I of $2LIN$ (mod 2) for which there is a degree-r pseudo-distribution consistent with satisfying $1-\epsilon$ fraction of I's constraint, but they conjecture that in fact one cannot satisfy more than $1-c\epsilon$ fraction of I's constraints. Lets call this conjecture the Khot-Moshkovitz conjecture. Even if true, the Khot-Moshkovitz conjecture does not contradict the conjecture that r-rounds of SOS yield an $n^{O(1/r)}$ approximation algorithm for the SSE,UG and 2LIN problems, since one would need to take $r = \Omega(\log n)$ for the latter algorithm to reach the range of parameters of the UGC and its ilk.

However, the Khot-Moshkovitz construction is only "step zero" in their plan to eventually prove the UGC/SSEH and hence contradict the conjecture that r-SOS rounds yield an $n^{O(1/r)}$ approximation for this problem. Specifically, to obtain a proof of the UGC, one would need to improve the KM paper in the following aspects: Step 1 would be to prove the Khot-Moshkovitz conjecture that no assignment can satisfy more than a $1-c\epsilon$ fraction of the equations in their instance. Step 2 would be to improve the gap from $1-\epsilon$ vs $1-c\epsilon$ to $1-\epsilon$ vs $1-\Omega(\sqrt{\epsilon})$ and improve the number of rounds from a constant to $n^{\Omega(1)}$. Step 3 is to extend the result from $2LIN(\epsilon)$ to $UG(\epsilon)$ (and maybe also to $SSE(\epsilon)$). Step 4 would be to to extend the result from a lower bound on SOS to an NP-hardness proof. In my (personal and quite possibly wrong) opinion, the significance of these hurdles is in the order listed. In particular, if the first and second step are completed then this would be sufficient to rule out what seems to be the most natural scenario under the assumption that the UGC is false— that SOS solves $SSE(\epsilon), UG(\epsilon)$ and $2LIN(\epsilon)$ in polynomial or quasipolynomial time, and I believe that achieving Steps 3 and 4 in this case should not be that much harder. Thus, despite the fact that Steps 3 and 4 seem more *qualitative* in nature than Step 2, I actually view Steps 1 and 2 as the most significant hurdles to overcome (or the more likely to be false if the UGC is false).

6.4 Formal statement and proof of Theorem 6.1

In this section (which is more or less copied from (Barak, Brandao, Harrow, Kelner, Steurer, and Zhou 2012) we show that a graph is a *small-set expander* if and only if the projector to the subspace of its adjacency matrix's top eigenvalues has a bounded $2 \to q$ norm for even $q \geq 4$. While the "if" part was known before, the "only if" part is novel. This characterization of small-set expanders is of general interest, and also leads to a reduction from the SSE problem to the problem of obtaining a good approximation for the $2 \to q$ norms. For simplicity of notation, throughout this section we use *expectation norms* — i.e. $\|w\|_p = (\mathbb{E}_i |w_i|^p)^{1/p}$.

Notation For a regular graph $G = (V, E)$ and a subset $S \subseteq V$, we define the *measure* of S to be $\mu(S) = |S|/|V|$ and we define $G(S)$ to be the distribution obtained by picking a random $x \in S$ and then outputting a random neighbor y of x. We define the *expansion* of S, to be $\Phi_G(S) = \Pr_{y \in G(S)}[y \notin S]$, where y is a random neighbor of x. For $\delta \in (0, 1)$, we define $\Phi_G(\delta) = \min_{S \subseteq V : \mu(S) \leq \delta} \Phi_G(S)$. We often drop the subscript G from Φ_G when it is clear from context. We identify G with its normalized adjacency (i.e., random walk) matrix. For every $\lambda \in [-1, 1]$, we denote by $V_{\geq \lambda}(G)$ the subspace spanned by the eigenvectors of G with eigenvalue at least λ. The projector into this subspace is denoted $P_{\geq \lambda}(G)$. For a distribution D, we let $\mathrm{cp}(D)$ denote the collision probability of D (the probability that two independent samples from D are identical).

Our main theorem of this section is the following:

Theorem 6.4. *For every regular graph G, $\lambda > 0$ and even q,*

1. (Norm bound implies expansion) *For all $\delta > 0, \epsilon > 0$, $\|P_{\geq \lambda}(G)\|_{2 \to q} \leq \epsilon/\delta^{(q-2)/2q}$ implies that $\Phi_G(\delta) \geq 1 - \lambda - \epsilon^2$.*

2. (Expansion implies norm bound) *There is a constant c such that for all $\delta > 0$, $\Phi_G(\delta) > 1 - \lambda 2^{-cq}$ implies $\|P_{\geq \lambda}(G)\|_{2 \to q} \leq 2/\sqrt{\delta}$.*

One corollary of Theorem 6.4 is that a good approximation to the $2 \to q$ norm implies an approximation of $\Phi_\delta(G)$.

Corollary 6.5. *If there is a polynomial-time computable relaxation $\mathcal{R}$ yielding good approximation for the $2 \to q$, then the* Small-Set Expansion Hypothesis *is false.*

(Note: here I use the standard notion of the SSEH with δ being an arbitrary small constant as opposed to equaling $1/\log n$; I didn't verify that the Raghvandra-Steurer-Tulsiani reduction can be used for the setting of $\delta = 1/\log n$ as well.)

Proof. Using (Raghavendra, Steurer, Tulsiani), to refute the small-set expansion hypothesis it is enough to come up with an efficient algorithm that given an input graph G and sufficiently small $\delta > 0$, can distinguish between the *Yes* case: $\Phi_G(\delta) < 0.1$ and the *No* case $\Phi_G(\delta') > 1 - 2^{-c \log(1/\delta')}$ for any $\delta' \geq \delta$ and some constant c. In particular for all $\eta > 0$ and constant d, if δ is small enough then in the *No* case $\Phi_G(\delta^{0.4}) > 1 - \eta$. Using the first part of Theorem 6.4, in the *Yes* case we know $\|V_{1/2}(G)\|_{2 \to q} \geq 1/(10\delta^{1/4})$, while in the *No* case, we can choose η to be sufficiently small so that the condition $\Phi_G(\delta^{0.2}) \geq 1 - \eta$ implies (via the second part of Theorem 6.5) that $\|V_{1/2}(G)\|_{2 \to q} \leq 2/\delta^{0.1}$. Thus an $O(\delta^{-0.15})$ approximation for the $2 \to q$ norm will refute the SSEH. □

The first part of Theorem 6.4 follows from previous work (e.g., see [?]). For completeness, we include a proof in Appendix ??. The second part will follow from the following lemma:

Lemma 6.6. *Set $e = e(\lambda, q) := 2^{cq}/\lambda$, with a constant $c \leq 100$. Then for every $\lambda > 0$ and $1 \geq \delta \geq 0$, if G is a graph that satisfies $\mathrm{cp}(G(S)) \leq 1/(e|S|)$ for all S with $\mu(S) \leq \delta$, then $\|f\|_q \leq 2\|f\|_2/\sqrt{\delta}$ for all $f \in V_{\geq \lambda}(G)$.*

Proving the second part of Theorem 6.4 from Lemma 6.6 We use the variant of the local Cheeger bound obtained in [**?**, Theorem 2.1], stating that if $\Phi_G(\delta) \geq 1-\eta$ then for every $f \in Ł2(V)$ satisfying $\|f\|_1^2 \leq \delta\|f\|_2^2$, $\|Gf\|_2^2 \leq c\sqrt{\eta}\|f\|_2^2$. The proof follows by noting that for every set S, if f is the characteristic function of S then $\|f\|_1 = \|f\|_2^2 = \mu(S)$, and $\mathrm{cp}(G(S)) = \|Gf\|_2^2/(\mu(S)|S|)$. □

Proof of Lemma 6.6. Fix $\lambda > 0$. We assume that the graph satisfies the condition of the Lemma with $e = 2^{cq}/\lambda$, for a constant c that we'll set later. Let $G = (V, E)$ be such a graph, and f be function in $V_{\geq\lambda}(G)$ with $\|f\|_2 = 1$ that maximizes $\|f\|_q$. We write $f = \sum_{i=1}^m \alpha_i\chi_i$ where $\chi_1, \dots, \chi_m$ denote the eigenfunctions of G with values $\lambda_1, \dots, \lambda_m$ that are at least λ. Assume towards a contradiction that $\|f\|_q > 2/\sqrt{\delta}$. We'll prove that $g = \sum_{i=1}^m (\alpha_i/\lambda_i)\chi_i$ satisfies $\|g\|_q \geq 5\|f\|_q/\lambda$. This is a contradiction since (using $\lambda_i \in [\lambda, 1]$) $\|g\|_2 \leq \|f\|_2/\lambda$, and we assumed f is a function in $V_{\geq\lambda}(G)$ with a maximal ratio of $\|f\|_q/\|f\|_2$.

Let $U \subseteq V$ be the set of vertices such that $|f(x)| \geq 1/\sqrt{\delta}$ for all $x \in U$. Using Markov and the fact that $\mathbb{E}_{x\in V}[f(x)^2] = 1$, we know that $\mu(U) = |U|/|V| \leq \delta$, meaning that under our assumptions any subset $S \subseteq U$ satisfies $\mathrm{cp}(G(S)) \leq 1/(e|S|)$. On the other hand, because $\|f\|_q^q \geq 2^q/\delta^{q/2}$, we know that U contributes at least half (in fact $1 - 2^{-q}$) of the term $\|f\|_q^q = \mathbb{E}_{x\in V} f(x)^q$. That is, if we define α to be $\mu(U)\,\mathbb{E}_{x\in U} f(x)^q$ then $\alpha \geq \|f\|_q^q/2$. We'll prove the lemma by showing that $\|g\|_q^q \geq 10\alpha/\lambda$.

Let c be a sufficiently large constant ($c = 100$ will do). We define U_i to be the set $\{x \in U : f(x) \in [c^i/\sqrt{\delta}, c^{i+1}/\sqrt{\delta})\}$, and let I be the maximal i such that U_i is non-empty. Thus, the sets $U_0, \dots, U_I$ form a partition of U (where some of these sets may be empty). We let α_i be the contribution of U_i to α. That is, $\alpha_i = \mu_i\,\mathbb{E}_{x\in U_i} f(x)^q$, where $\mu_i = \mu(U_i)$. Note that $\alpha = \alpha_0 + \cdots + \alpha_I$. We'll show that there are some indices $i_1, \dots, i_J$ such that:

(i) $\alpha_{i_1} + \cdots + \alpha_{i_J} \geq \alpha/(2c^{10})$.

(ii) For all $j \in [J]$, there is a non-negative function $g_j : V \to \mathbb{R}$ such that $\mathbb{E}_{x\in V} g_j(x)^q \geq e\alpha_{i_j}/(10c^2)^{q/2}$.

(iii) For every $x \in V$, $g_1(x) + \cdots + g_J(x) \leq |g(x)|$.

Showing these will complete the proof, since it is easy to see that for two non-negative functions and even q, g', g'', $\mathbb{E}(g'(x) + g''(x))^q \geq \mathbb{E}\, g'(x)^q + \mathbb{E}\, g''(x)^q$, and hence **(ii)** and **(iii)** imply that

$$\|g\|_4^4 = \mathbb{E}\, g(x)^4 \geq (e/(10c^2)^{q/2}) \sum_j \alpha_{i_j} \ . \tag{6.5}$$

Using **(i)** we conclude that for $e \geq (10c)^q/\lambda$, the right-hand side of (6.5) will be larger than $10\alpha/\lambda$.

We find the indices $i_1, \dots, i_J$ iteratively. We let $\mathcal{I}$ be initially the set $\{0..I\}$ of all indices. For $j = 1, 2, ...$ we do the following as long as $\mathcal{I}$ is not empty:

1. Let i_j be the largest index in $\mathcal{I}$.

2. Remove from $\mathcal{I}$ every index i such that $\alpha_i \leq c^{10}\alpha_{i_j}/2^{i-i_j}$.

We let J denote the step when we stop. Note that our indices $i_1, \dots, i_J$ are sorted in descending order. For every step j, the total of the α_i's for all indices we removed is less than $c^{10}\alpha_{i_j}$ and hence we satisfy **(i)**. The crux of our argument will be to show **(ii)** and **(iii)**. They will follow from the following claim:

Claim 6.7. *Let $S \subseteq V$ and $\beta > 0$ be such that $|S| \leq \delta$ and $|f(x)| \geq \beta$ for all $x \in S$. Then there is a set T of size at least $e|S|$ such that $\mathbb{E}_{x\in T}\, g(x)^2 \geq \beta^2/4$.*

The claim will follow from the following lemma:

Lemma 6.8. *Let D be a distribution with $\mathrm{cp}(D) \leq 1/N$ and g be some function. Then there is a set T of size N such that $\mathbb{E}_{x\in T}\, g(x)^2 \geq (\mathbb{E}\, g(D))^2/4$.*

Proof. Identify the support of D with the set $[M]$ for some M, we let p_i denote the probability that D outputs i, and sort the p_i's such that $p_1 \geq p_2 \cdots p_M$. We let β' denote $\mathbb{E}\, g(D)$; that is, $\beta' = \sum_{i=1}^{M} p_i g(i)$. We separate to two cases. If $\sum_{i>N} p_i g(i) \geq \beta'/2$, we define the distribution D' as follows: we set $\Pr[D' = i]$ to be p_i for $i > N$, and we let all $i \leq N$ be equiprobable (that is be output with probability $(\sum_{i=1}^{N} p_i)/N$). Clearly, $\mathbb{E}\,|g(D')| \geq \sum_{i>N} p_i g(i) \geq \beta'/2$, but on the other hand, since the maximum probability of any element in D' is at most $1/N$, it can be expressed as a convex combination of flat distributions over sets of size N, implying that one of these sets T satisfies $\mathbb{E}_{x\in T}\,|g(x)| \geq \beta'/2$, and hence $\mathbb{E}_{x\in T}\, g(x)^2 \geq \beta'^2/4$.

The other case is that $\sum_{i=1}^{N} p_i g(i) \geq \beta'/2$. In this case we use Cauchy-Schwarz and argue that

$$\beta'^2/4 \leq \left(\sum_{i=1}^{N} p_i^2\right)\left(\sum_{i=1}^{N} g(i)^2\right) . \tag{6.6}$$

But using our bound on the collision probability, the right-hand side of (6.6) is upper bounded by $\frac{1}{N}\sum_{i=1}^{N} g(i)^2 = \mathbb{E}_{x\in[N]}\, g(x)^2$. □

Proof of Claim 6.7 from Lemma 6.8. By construction $f = Gg$, and hence we know that for every x, $f(x) = \mathbb{E}_{y\sim x}\, g(y)$. This means that if we let D be the distribution $G(S)$ then

$$\mathbb{E}\,|g(D)| = \mathbb{E}_{x\in S}\,\mathbb{E}_{y\sim x}\,|g(y)| \geq \mathbb{E}_{x\in S}\,|\mathbb{E}_{y\sim x}\, g(y)| = \mathbb{E}_{x\in S}\,|f(x)| \geq \beta .$$

By the expansion property of G, $\mathrm{cp}(D) \leq 1/(e|S|)$ and thus by Lemma 6.8 there is a set T of size $e|S|$ satisfying $\mathbb{E}_{x\in T}\, g(x)^2 \geq \beta^2/4$. □

We will construct the functions $g_1, \ldots, g_J$ by applying iteratively Claim 6.7. We do the following for $j = 1, \ldots, J$:

1. Let T_j be the set of size $e|U_{i_j}|$ that is obtained by applying Claim 6.7 to the function f and the set U_{i_j}. Note that $\mathbb{E}_{x\in T_j}\, g(x)^2 \geq \beta_{i_j}^2/4$, where we let $\beta_i = c^i/\sqrt{\delta}$ (and hence for every $x \in U_i$, $\beta_i \leq |f(x)| \leq c\beta_i$).
2. Let g'_j be the function on input x that outputs $\gamma \cdot |g(x)|$ if $x \in T_j$ and 0 otherwise, where $\gamma \leq 1$ is a scaling factor that ensures that $\mathbb{E}_{x\in T_j}\, g'(x)^2$ equals exactly $\beta_{i_j}^2/4$.
3. We define $g_j(x) = \max\{0, g'_j(x) - \sum_{k<j} g_k(x)\}$.

Note that the second step ensures that $g'_j(x) \leq |g(x)|$, while the third step ensures that $g_1(x) + \cdots + g_j(x) \leq g'_j(x)$ for all j, and in particular $g_1(x) + \cdots + g_J(x) \leq |g(x)|$. Hence the only thing left to prove is the following:

Claim 6.9. $\mathbb{E}_{x\in V}\, g_j(x)^q \geq e\alpha_{i_j}/(10c)^{q/2}$

Proof. Recall that for every i, $\alpha_i = \mu_i \mathbb{E}_{x\in U_i} f(x)^q$, and hence (using $f(x) \in [\beta_i, c\beta_i)$ for $x \in U_i$):

$$\mu_i \beta_i^q \leq \alpha_i \leq \mu_i c^q \beta_i^q \ . \tag{6.7}$$

Now fix $T = T_j$. Since $\mathbb{E}_{x\in V} g_j(x)^q$ is at least (in fact equal) $\mu(T)\,\mathbb{E}_{x\in T} g_j(x)^q$ and $\mu(T) = e\mu(U_{i_j})$, we can use (6.7) and $\mathbb{E}_{x\in T} g_j(x)^q \geq (E_{x\in T} g_j(x)^2)^{q/2}$, to reduce proving the claim to showing the following:

$$\mathbb{E}_{x\in T} g_j(x)^2 \geq (c\beta_{i_j})^2/(10c^2) = \beta_{i_j}^2/10 \ . \tag{6.8}$$

We know that $\mathbb{E}_{x\in T} g_j'(x)^2 = \beta_{i_j}^2/4$. We claim that (6.8) will follow by showing that for every $k < j$,

$$\mathbb{E}_{x\in T} g_k'(x)^2 \leq 100^{-i'} \cdot \beta_{i_j}^2/4 \ , \tag{6.9}$$

where $i' = i_k - i_j$. (Note that $i' > 0$ since in our construction the indices $i_1, \ldots, i_J$ are sorted in descending order.)

Indeed, (6.9) means that if we let momentarily $\|g_j\|$ denote $\sqrt{\mathbb{E}_{x\in T} g_j(x)^2}$ then

$$\|g_j\| \geq \|g_j'\| - \|\textstyle\sum_{k<j} g_k\| \geq \|g_j'\| - \displaystyle\sum_{k<j} \|g_k\| \geq \|g_j'\|(1 - \sum_{i'=1}^{\infty} 10^{-i'}) \geq 0.8\|g_j'\| \ . \tag{6.10}$$

The first inequality holds because we can write g_j as $g_j' - h_j$, where $h_j = \min\{g_j', \sum_{k<j} g_k\}$. Then, on the one hand, $\|g_j\| \geq \|g_j'\| - \|h_j\|$, and on the other hand, $\|h_j\| \leq \|\sum_{k<j} g_k\|$ since $g_j' \geq 0$. The second inequality holds because $\|g_k\| \leq \|g_k'\|$. By squaring (6.10) and plugging in the value of $\|g_j'\|^2$ we get (6.8).

Proof of (6.9) By our construction, it must hold that

$$c^{10}\alpha_{i_k}/2^{i'} \leq \alpha_{i_j} \ , \tag{6.11}$$

since otherwise the index i_j would have been removed from the $\mathcal{I}$ at the k^{th} step. Since $\beta_{i_k} = \beta_{i_j} c^{i'}$, we can plug (6.7) in (6.11) to get

$$\mu_{i_k} c^{10+4i'}/2^{i'} \leq c^4 \mu_{i_j}$$

or

$$\mu_{i_k} \leq \mu_{i_j} (2/c)^{4i'} c^{-6} \ .$$

Since $|T_i| = e|U_i|$ for all i, it follows that $|T_k|/|T| \leq (2/c)^{4i'} c^{-6}$. On the other hand, we know that $\mathbb{E}_{x\in T_k} g_k'(x)^2 = \beta_{i_k}^2/4 = c^{2i'}\beta_{i_j}^2/4$. Thus,

$$\mathbb{E}_{x\in T} g_k'(x)^2 \leq 2^{4i'} c^{2i'-4i'-6} \beta_{i_j}^2/4 \leq (2^4/c^2)^{i'} \beta_{i_j}^2/4 \ ,$$

and now we just choose c sufficiently large so that $c^2/2^4 > 100$. □

□

Chapter 7

Linear programming extension complexity

This was a guest lecture by Ankur Moitra for which notes are not available at this time.

Chapter 8

Semidefinite programming extension complexity

In this course we have alluded to an intuition that, at least in some domains, the SOS algorithm is *optimal*, in the sense that no other efficient algorithm could beat it. There are several ways to try to justify this intuition:

Ideally, we would want to simply *prove* this, under assumptions such as $\mathbf{P} \neq \mathbf{NP}$. There are two main results along those lines:

- Siu-On Chan showed that for every $\epsilon > 0$ and predicate $P : \mathbb{F}_q^k \to \{0, 1\}$ of the form 1_V where where V is an affine subspace of $\mathbb{F}_q^k$ such that the uniform distribution on V is pairwise independent, it is NP-hard to distinguish between the case that an CSP-P instance is $1 - \epsilon$ satisfiable and the case that it is $|P^{-1}(1)|/q^k + \epsilon$ satisfiable. Up to the ϵ, this exactly matches the SOS lower bound of Tusliani (which itself is a natural extension of Grigoriev's 3XOR lower bound that we saw in class).

- Prasad Raghavendra showed that if the Unique Games Conjecture is true, then for every $\epsilon > 0$ and predicate P, beating the performance of the degree 2 SOS algorithm on Max-P by $\epsilon > 0$ is NP-hard. Thus, if the UGC is true, that would be very strong evidence for optimality of SOS. Even if the UGC is false, but it is refuted by the SOS algorithm, one could hope that there would be a "modified Raghavendra theorem" showing that the SOS algorithm is optimal. The ideal version would translate a degree d SOS lower bound into a reduction that maps a (suitable variant of) label cover instance of n variables into an instance of the target problem of size $N = poly(n)2^{O(n/d)}$, hence ruling out an $N^{o(d)}$-time algorithm under the exponential time hypothesis.

However, given current knowledge in complexity, such proofs are always conditional on some assumption. Even if we are not too concerned with taking $\mathbf{P} \neq \mathbf{NP}$, or even the ETH, as an axiom (we cannot take such a blasé attitude towards the UGC), these assumptions are inherently limited in the sense that they don't apply (again, based on current knowledge) to *average-case* complexity. Therefore, for several reasons it is also interesting to try to prove that some natural algorithms do not beat the SOS algorithm in some interesting domains.

Perhaps the first question to ask is whether one can beat the SOS algorithm by simply using stronger semidefinite programs. The degree d SOS algorithm can be thought of as obtaining a tighter SDP by adding a set of very specific n^d constraints to the original basic SDP, but perhaps it is possible to add a different set of n^d constraints that would give better performance. The formal way to phrase this question is *extension complexity.* In this lecture we will discuss a very recent result of Lee, Raghavendra and Steurer giving a "Raghavendra Theorem for Semidefinite extension complexity", by translating SOS lower bounds into semidefinite programming extension complexity lower bounds. This can be thought of as the analog of the prior work of Chan,Lee,Raghavendra and Steurer that showed a similar connection between Sherali-Adams lower bounds and linear programming extension complexity lower bounds.

For example using this translation they use Grigoriev's 3XOR lower bound to show the following theorem:

Theorem 8.1. *For every $\epsilon > 0$ and subspace U of the functions from $\{\pm1\}^n$ to $\mathbb{R}$ of dimension less than $n^{o(\log n/\log\log n)}$, there is an instance I of $3XOR$ of value (i.e., maximum fraction of satisfied constraints) at most $1/2+\epsilon$, such that there is no U-proof that the value of I is less than $1 = \epsilon$.*

The definition of a U *prood* that some function $f : \{\pm1\}^n \to \mathbb{R}$ satisfies $f \leq \alpha$ is that there are functions $g_1, \ldots, g_t \in U$ such that

$$f(x) = \alpha - \sum_{i=1}^{t} g_i(x)^2$$

for every $x \in \{\pm1\}^n$. Note that a degree-d SOS proof corresponds to a U-proof where U is the span of all monomials of degree at most d, and hence U proofs are generalization of degree $\log_n \dim(U)$-SOS proofs.

It turns out that the notion of *SDP rank* plays here the same role that the notion of *non-negative rank* plays for linear programming extension complexity. We say that a non-negative $p \times q$ matrix M has *psd-rank* at most r if there exist $r \times r$ psd matrices $\{A_i\}_{i=1}^{p}$ and $\{B_j\}_{j=1}^{q}$ such that $M_{i,j} = \mathrm{Tr}(A_i B_j)$. **Exercise 8.1:** Prove that M has *non-negative rank* at most r if and only if it has a decomposition such as the one above where the A_i's and B_j's are diagonal .

At the heart of their work is the following theorem:

Theorem 8.2. *For $d < m < n/2$, $f : \{\pm1\}^m \to [0,\infty)$, let $M = M_n^f$ be the $\binom{n}{m} \times 2^n$ matrix such that $M(S,x) = f(x_S)$ for every $S \in \binom{[n]}{m}$ and $x \in \{\pm1\}^n$. If there is no degree d SOS proof that $f \geq 0$ then*

$$\mathrm{rank}_{\mathsf{psd}}(M) \geq n^{\Omega(d)}$$

The rest of this lecture will be devoted to outlining the proof of Theorem 8.2. To make things concrete, we will consider the particular $f : \{\pm1\}^m \to \mathbb{R}$ that one obtains by taking the instance of 3XOR I arising from Grigoriev's 3XOR lower bond. That is, we let $f(x)$ equal 0.7 minus the fraction of I's constraints that are satisfied by the assignment x. As we saw in class, for every x, $f(x) \geq 0.1$ but there is a degree (say) $d = m/1000$ pseudo-distribution D over $\{\pm1\}^m$ that satisfies the constraint $\{f(x) = -0.3\}$.

There are several ways to represent such a pseudo-distribution. One representation (which is the one we used in class) is to simply use the pseudo-expectation operator mapping a polynomial P to $\tilde{\mathbb{E}}_{x\sim D} P$, but another one is to define for every $x \in \{\pm1\}^x$, $D(x) \in \mathbb{R}$ to be a number such that

$$\tilde{\mathbb{E}}_{x\sim D} P = \mathbb{E}_{x\in\{\pm1\}^n} D(x)P(x) \tag{8.1}$$

we shall follow the language of LRS and call this a "pseudo-density" operator. To move from the previous representation to (8.1), we can simply define

$$D(x) = \sum_{|\alpha|\leq d} \left(\tilde{\mathbb{E}}_{x'\sim D}\ \chi_\alpha(x')\right) \chi_\alpha(x)$$

where for $\alpha \subseteq [m]$, we define $\chi_\alpha(x) = \prod_{i\in\alpha} x_i$.

We suppose, toward a contradiction that $\mathrm{rank}_{\mathsf{psd}}(M_n^f) = r$ for some $r = n^{o(1)}$ demonstrated by some decomposition $\{P(S)\}_{S\in\binom{[n]}{m}}, \{Q(x)\}_{x\in\{\pm1\}^n}$. We will consider the following quantity:

$$\mathbb{E}_{S\in\binom{[n]}{m}} \mathbb{E}_{x\in\{\pm1\}^n} D(x_S)f(x_S) \quad (*)$$

On one hand for every fixed S and fixing of $x_{\overline{S}}$, (*) equals to $\mathbb{E}_{x'\in\{\pm1\}^m} D(x')f(x') = -0.3$.

On the other hand, by the psd decomposition this equals

$$\mathbb{E}_S \mathbb{E}_{x\in\{\pm1\}^n} D(x_S)\mathrm{Tr}(P(S)Q(x)) = \mathbb{E}_S \mathbb{E}_{x''\in\{\pm1\}^{\overline{S}}} \mathbb{E}_{x'\in\{\pm1\}^S} D(x')\|\sqrt{P(S)}\sqrt{Q(x',x'')}\|_F^2 \qquad (8.2)$$

(where the square roots of P and Q are defined since these are p.s.d matrices.) **Exercise 8.2:** Prove that for every psd matrices PQ, $\mathrm{Tr}(PQ) = \|\sqrt{PQ}\|_F^2 = \sum_{i,j}\sqrt{PQ}_{i,j}^2$, where for a psd matrix $A = \sum \lambda_i v_i v_i^\top$, $\sqrt{A} = \sum \sqrt{\lambda_i} v_i v_i^\top$.

Note that for every fixed S, x'', the function $x' \mapsto \|\sqrt{P(S)}\sqrt{Q(x,x'')}\|_F^2$ is the sum of the squares of the entries of these matrices, and so this function is some sum of squares $g(x')$. If by some luck the degree of g was smaller than $d < 2$ we would get our contradiction and be done since we know that

$$\mathbb{E}_{x'} D(x')g(x') = \tilde{\mathbb{E}}\, g(x') \geq 0$$

for every SOS g of degree at most $d/2$.

The heart of the LRS proof is therefore in showing that this function g can in fact be sufficiently well approximated by a low degree polynomial. They do so using two tools:

- *Quantum learning* — they use an instance of the following general principle that has turned out to be useful again and again in computer science:

  Simple tests can be fooled by (fairly) simple objects.

  At a high level this phenomena underlies the whole theory of pseudorandomness, but we will be focused on a particular set of manifestations of it, known as "Boosting", "Impagliazzo's Hardcore Lemma", "Dense Model Theorem" and more. LRS use a quantum version of this principle.

  Their idea is that because in our case we test the function $Q(x)$ against the degree m function $\mathbb{E}_S P(S)D(x_S)$, we can approximate it with some function of the form $h(x) = R^2(x)$ where R has degree at most $\tilde{O}(m)$.

- *Random restrictions* — the above is still not sufficient since we need to reduce the degree below $m/1000$. For this we use the other general tool

  Simple functions become even simpler if we fix most of their inputs at random.

  Specifically, using tools from Fourier analysis of Boolean functions, one can show that, once we established the degree of h is not too large, if we choose a random S and fix the values of x in $\overline{S}$ then we significantly reduce the degree of h further, and can assume that it is $o(m)$ (up to some an error of $r^{\omega(1)}$ in the norm which would be negligible in our setting), and hence we can apply the result above.

8.1 Boosting, dense model, hardcore lemma, multiplicative weights, computational entropy, and their quantum/matrix/semidefinite cousins

There is a set of results that has been used across many areas in mathematics and computer science. Here are some examples of these results. (This is not meant to be comprehensive review of the related literature— these are the kind of ideas that seem to have been rediscovered again and again by people in different communities and for different purposes.)

- Suppose that you are trying to *learn* some unknown function $F : \Omega \to \{0, 1\}$ and that for every distribution D over Ω, you can find a function $f_D : \Omega \to \{0, 1\}$ that has $1/2 + \epsilon$ agreement with F, then you can boost this to $1 - \delta$ agreement by combining $\mathrm{poly}(1/\epsilon, \log(1/\epsilon))$ of these functions.

  An algorithm achieving such boosting was first put forward by Schapire in 1989, with some later improvements by Freund culminating in their 1995 *AdaBoost* algorithm.

- Suppose that you know that some function $F : \Omega \to \{0, 1\}$ is *mildly hard* in the sense that every efficient algorithm A has at most $1 - \delta$ agreement with it. It turns out that there is a not-too-small subset $H \subseteq \Omega$ (in fact, of measure 2δ) on which F is *extremely hard* in the sense that every efficient algorithm has at most $1/2+\epsilon$ (where the ϵ plays a part in the quantitative losses between our two instantiations of the word "efficient").

  This theorem, which turns out to be extremely useful in the theory of pseudo-randomness, is known as "Impagliazzo's Hardcore Lemma", proven by Russell Impagliazzo in 1995 (with an important quantitative improvement by Holenstein in 2005. (The connection for Boosting was noted by Klivans and Servedio in 2003; see also my paper with Hardt and Kale.)

- Suppose that $S \subseteq \Omega$ is pseudorandom in the sense that one cannot distinguish a uniform element of S from a uniform element of Ω via efficient algorithm, and suppose that $P \subseteq S$ satisfies $|P| \geq \delta|S|$. Then you can find some $M \subseteq \Omega$ with $|M| \geq \delta|\Omega|$ such that one cannot distinguish a uniform element of P from a uniform element of M.

  This theorem is known as the *dense model theorem*, and was first shown by Green and Tao in the context of their 2004 work establishing that the set of primes contains arbitrarily long arithmetic progressions. A more general explicit version was later given by Tao and Zeigler. Some simplified proofs were later found by Gowers and (independently) Reingold, Trevisan and Vadhan. Roughly speaking, Green and Tao used in their theorem number theoretic results showing that the set S of *pseudo primes* (integers having only few large divisors) are pseudorandom. Since the set P of primes has constant density in S, the dense model theorem shows that it is indistinguishable from a set M dense in the set of all integers, but such sets contain large arithmetic progressions by Szemeredi's Theorem.

All these results share some the following properties:

- They are counterintuitive when you (or at least I) first hear about them.
- They are incredible useful.
- They are proven via the multiplicative weights algorithm or von Neumann's min-max theorem (aka linear programming duality or Hahn-Banach theorem).

- They are actually not that hard to prove once you have the nerve to guess that the result might be true.

It turns out that they are all essentially equivalent. Let me sketch how you might prove the Boosting result. You can think of this as a game between two players— Player I comes up with a distribution D, and Player II responds with an algorithm f_D that has $1/2+\epsilon$ agreement with f. We know that by the min-max theorem that Player II could come up with a single distribution over algorithms (or equivalently a probabilistic algorithm) A that would have such agreement with *every* distribution, which means that it succeeds in solving F on any input with probability $1/2+\epsilon$— probability that can be boosted to $1-\delta$ via $O(\log 1/\delta)$ repetitions. Now to converge to this algorithm we can use a fairly simple process of back and forth between the distribution player and the algorithm player. The distributions would be updated according to a multiplicate update rule, and the final algorithm would be some weighted average of the algorithms obtained in each round. So, if these intermediate algorithms are simple, then so will be the final one.

At a very high level, the approach they use is the following one— suppose that you have a convex optimization problem $\max f(x)$ for some convex f and you know that there is a solution Q that has a lot of entropy (in some well prescribed sense). It turns out that this means that there should be a "simple" solution. The idea is that if you run the multiplicative update algorithm to solve the convex program then it starts with the uniform solution that has maximal entropy and at each step it roughly maintains the invariant of having a solution with some entropy H that is (approximately) the best solution possible with at least this much entropy. Now, if there is a solution Q with very high entropy it means that we would achieve a solution with value at least as good as Q with a small number of steps of this algorithm, and that solution will be "simple". If this discussion doesn't make any sense, hopefully the more technical derivation below will be more informative.

Further reading on the "classical" versions. Luca Trevisan had several blog posts related to this, see `https://lucatrevisan.wordpress.com/2008/12/07/applications-of-low-complexity-approximat` and also a survey in the 2011 theory of cryptography conference. Sitanshu Gakkhar's Master's thesis `http://summit.sfu.ca/item/12349`, Russell Impagliazzo's talk `https://video.ias.edu/csdm/densemodelthm` and scribe notes `https://www.math.ias.edu/files/russell_scribe.pdf`, paper of Trevisan, Tulsiani and Vadhan `http://ttic.uchicago.edu/~madhurt/Papers/regularity-full.pdf`. These results can also be phrased in the language of computational entropy— intuitively a set P as in the dense model theorem, has "pseudo-entropy" at least $\log|\Omega| - \log(1/\delta)$. This was first explored in my 2003 paper with Shaltiel and Wigderson (see also Dziembowski and Pietrzak 2008). (Note that our paper had a bug and proved a weaker result than originally claimed; see the 2011 paper of Benjamin Fuller and Leonid Reyzin for discussion.)

Semidefinite/quantum extensions People have looked at extension of these results to the *quantum* setting. One can think of this as extending results from classical to quantum, from numbers to matrices, or from linear programming to semidefinite programming. In any case one obtains similar results, see the LRS paper. Note that (as we will see) LRS uses a very restricted special case of this general principle focusing on a single test.

8.2 Random restrictions

The idea of random restriction is to take a function $f:\{\pm 1\}^n \to \mathbb{R}$ and change it into the function $g:\{\pm 1\}^m \to \mathbb{R}$ obtain by picking an m-sized set $S \subseteq [n]$ at random, and $x'' \in \{\pm 1\}^{\overline{S}}$, and then define $g(x') = f(x', x'')$. The hope is that g is significantly simpler than f. This is not necessarily always the case. For example, if f is simply the parity function $x_1 \cdots x_n$, then g is a parity as well, but if f is "simpler" than the parity in some sense, then g could be significantly even simpler. This idea has been used in Hastad's switching lemma, where one can use random restriction to show that if f has a small constant depth circuit, then g has a circuit of even smaller depth. In the current context we need an even simpler statement about the Fourier degree. (See Ryan O'Donnel's book for a thorough treatment of random restrictions.) If you have a monomial $\prod_{i \in T} x_i$, then after restricting to a random set S you are left with the monomial, $\prod_{i \in S \cap T} x_i$ which is expected to have size about $|S||T|/n$ which would be much smaller than both $|S|$ and $|T|$ if they are both much smaller than n. Specifically, if we take a unit norm function $f : \{\pm 1\}^n \to \mathbb{R}$ of degree at most $\ell = \tilde{O}(m)$, and restrict it to a random m sized set S, then the norm squared of the part of f that has degree at least $m/10^4$ would be in expectation less than $\binom{\ell}{m/10^4}(m/n)^{m/10^4}$ which for $n \gg m$ would be $n^{-\Omega(m)}$. If the presumed PSD rank r satisfies $r = n^{o(m)}$ then it turns out that this is small enough to be treated as negligible.

8.3 Proof sketch

We want to obtain a contradiction to the statement

$$\mathbb{E}_{S \in \binom{[n]}{m}} \mathbb{E}_{x \in \{\pm 1\}^n} D(x_S) f(x_S) = -0.3 \tag{8.3}$$

under the assumption that $f(x_s) = \mathrm{Tr}(P(S)Q(x))$ where $P(S)$ and $Q(x)$ are $r \times r$ psd matrices for $r = m^{o(1)}$.

We first phrase the LHS of (8.3) as

$$\mathrm{Tr}(MQ)$$

where M is the block matrix with 2^n $r \times r$ blocks with the x^{th} block corresponding to $\mathbb{E}_S D(x_S)P(S)$, and the Q is the block matrix 2^n $r \times r$ blocks with the x^{th} block corresponding to $2^{-n}Q(x)$.

We will later show using a quantum learning type argument that there exists a matrix $R = p(M)$, for some polynomial p with degree $\tilde{O}(1)$, such that $\mathrm{Tr}(R^2) = \mathrm{Tr}(Q)$ and

$$\mathrm{Tr}(MR^2) \le \mathrm{Tr}(MQ) + 0.1 = -0.2 \tag{8.4}$$

Let's defer the proof of this for a moment and see what it yields. Since every element of M is a polynomial in x of degree at most m, we can think of R as a matrix-valued $\tilde{O}(m)$ degree polynomial in x and rewrite (8.4) as

$$\mathbb{E}_{S \in \binom{[n]}{m}} \mathbb{E}_{x \in \{\pm 1\}^n} D(x_S) \mathrm{Tr}(P(S)R^2(x)) = \mathbb{E}_{S \in \binom{[n]}{m}} P(S)\, \mathbb{E}_{x' \in \{\pm 1\}^S} D(x')\, \mathbb{E}_{x'' \in \{\pm 1\}^{\overline{S}}} R^2(x', x'') \tag{8.5}$$

If we fix a "typical" value for S and x'', then as mentioned above, we can argue using random restrictions that the function $R'(x') = R(x', x'')$ equals $L(x') + H(x')$ where $L(x')$ is a degree $o(m)$ polynomial and $\|H(x')\| \le r^{-\omega(1)}$. Since without loss of generality, using the fact that $P(S)$ is an $r \times r$ matrix, $\|P(S)\| \le r^2$, the effect of $H(x)$ will be negligible and we get that

$$\mathbb{E}_{x' \in \{\pm 1\}^S} D(x') \mathrm{Tr}(P(S)L^2(x')) \le -0.2 + o(1) \tag{8.6}$$

Now define $g(x')$ to be $\mathrm{Tr}(P(S)L^2(x')) = \|\sqrt{P(S)}L(x')\|_F^2$. Since for a fixed S, every entry of the matrix $\sqrt{P(S)}L(x')$ is a degree $o(m)$ polynomial in x', this quantity is a polynomial of degree $o(m)$ which is a sum of squares, and so we get

$$\tilde{\mathbb{E}}_D\, g(x') = \mathbb{E}_{x'}\, D(x')g(x') \le -0.2 + o(1) < 0$$

contradicting the fact that D is a degree $m/1000$ pseudo distribution.

8.3.1 Quantum learning argument

We only need a rather restricted version of quantum learning (see the LRS paper for a much more general statement). We want to prove the following:

Lemma 8.3. *Let M, Q be $s \times s$ matrices such that Q is psd and $\mathrm{Tr}(Q) = 1$, then there exists a degree $\mathrm{poly}(\log \|Q\|s, 1/\epsilon, \|M\|)$ polynomial p such that*

$$\mathrm{Tr}(Mp(M)^2)/\mathrm{Tr}(p(M)^2) \leq \mathrm{Tr}(MQ) + \epsilon \tag{8.7}$$

Proof. We will show that this holds where instead of using a polynomial we write a matrix exponential $p(M) = \alpha e^{(\theta/2)M}$ for $\theta = \mathit{poly}(\|M\|, \log \|Q\|s, 1/\epsilon)$, and the result would then follow from a Taylor approximation. (Note that in our case $\|Q\| = \mathit{poly}(r)/s$; more generally LRS work with a lower bound on the von Neumann entropy of Q which is simply the Shannon entropy of the eigenvalues. If Q has trace 1 and $\|Q\| \leq \alpha/s$ then the von-Neumann entropy is at least $\log s - \log(1/\alpha)$ using known relations between min-entropy and Shannon entropy.)

By making the transformation $M \mapsto I - M/\|M\|$ we can change to the case that M is psd of norm at most 1 and that our goal is to show that for $\theta = O(\log \|Q\|s, \mathit{poly}(1/\epsilon))$

$$\mathrm{Tr}(Me^{\theta M}) \geq \mathrm{Tr}(MQ) - \epsilon \tag{8.8}$$

What is the maximum value a matrix Q can achieve for $\mathrm{Tr}(MQ)$ subject to being psd, trace 1, and having norm at most some value α/s? It's not hard to see that this would be obtained by having $Q = (\alpha/s)\sum_{i=1}^{s/\alpha} v_i v_i^\top$ where $v_1, v_2, v_3, \ldots$ are the eigenvectors of M sorted in descending order of the eigenvalues $\lambda_1, \lambda_2, \ldots$. In this case $\mathrm{Tr}(MQ)$ will simply be the average of the s/α top eigenvalues of M. Once again, one can see that the most extreme situation would be if all the top s/α eigenvalues would be equal to 1 while the rest are zero, since that would maximize $\mathrm{Tr}(MQ)$ under our conditions. Now, if we consider the matrix

$$e^{\theta M} = \sum e^{\theta \lambda_i} v_i v_i^\top$$

we can see that it gives weight 1 to the eigenvectors corresponding to zero, and weight e^θ to the eigenvectors corresponding to 1, since there are at most α times more of the former than the latter, if $\theta \gg \log \alpha$ then almost of all of the weight will be on the top eigenvectors, and we get that the matrix (after normalizing to trace 1) will give a value very close to $\mathrm{Tr}(MQ)$. □

Chapter 9

Open problems

Here are some open problems regarding the Sum-of-Squares algorithm. In most cases I phrased the problem as asking to show a particular statement, though of course showing the opposite statement would be very interesting as well. These are not meant to be a complete or definitive list, but could perhaps spark your imagination to think of those or other research problems of your own. The broader themes these questions are meant to explore are:

- Can we understand in what cases do SOS programs of intermediate degree (larger than 2 but much smaller than n) yield non-trivial guarantees?
- Can we give more evidence to, or perhaps refute, the intuition that the SOS algorithm is *optimal* in some broad domains?
- Can we understand the performance of SOS in *average-case* setting, and whether there are justifications to consider it optimal in this setting as well? This is of course interesting for both machine learning and cryptography.
- Can we understand the role of *noise* in the performance of the SOS algorithm? Is noise a way to distinguish between "combinatorial" and "algebraic" problems in the sense of `http://windowsontheory.org/2013/10/07/structure-vs-combinatorics-in-computational-complexity/`?

Well posed problems

Problem 1: Show that for every constant C there is some $\delta > 0$ and a quasipolynomial ($n^{polylog(n)}$) time algorithm that on input a subspace $V \subseteq \mathbb{R}^n$, can distinguish between the case that V contains the characteristic vector of a set of measure at most δ, and the case that $\mathbb{E}_i v_i^4 \leq C(\mathbb{E}_i v_i^2)^2$ for every $v \in V$. Extend this to a quasipolynomial time algorithm to solve the small-set expansion problem (and hence refute the small set expansion hypothesis). Extend this to a quasipolynomial time algorithm to solve the unique-games problem (and hence refute the unique games conjecture). If you think this cannot be done then even showing that the $d = \log^2 n$ (in fact, even $d = 10$) SOS program does not solve the unique-games problem (or the 4/2 norms ratio problem as defined above) would be very interesting.

Problem 2: Show that there is some constant d such that the degree-d SOS problem can distinguish between a random graph and a graph in which a clique of size $f(n)$ was planted for some $f(n) =$

$o(\sqrt{n})$, or prove that this cannot be done. Even settling this question for $d = 4$ would be very interesting.

Problem 3: Show that the SOS algorithm is optimal in some sense for "pseudo-random" constraint satisfaction problems, by showing that for every predicate $P : \{0,1\}^k \to \{0,1\}$, $\epsilon > 0$ and pairwise independent distribution μ over $\{0,1\}^k$, it is NP hard to distinguish, given an instance of MAX-P (i.e., a set of constraints each of which corresponds to applying P to k literals of some Boolean variables $x_1, \ldots, x_n$), between the case that one can satisfy $1-\epsilon$ fraction of the constraints, and the case that one can satisfy at most $\mathbb{E}_{x\sim\mu} P(x) + \epsilon$ fraction of them. (In a recent work with Chan and Kothari, we show that small degree SOS programs cannot distinguish between these two cases.)

Problem 4: More generally, can we obtain a "UGC free Raghavendra Theorem"? For example, can we show (without relying on the UGC) that for every predicate $P : \{0,1\}^k \to \{0,1\}$, $c > s$ and $\epsilon > 0$, if there is an n-variable instance of MAX-P whose value is at most s but on which the $\Omega(n)$ degree SOS program outputs at least c, then distinguishing between the case that a CSP-P instance as value at least $c - \epsilon$ and the case that it has value at most $s + \epsilon$ is NP-hard?

Problem 5: Show that there is some $\eta > 1/2$ and $\delta < 1$ such that for sufficiently small $\epsilon > 0$, the degree n^δ SOS program for Max-Cut can distinguish, given a graph G, between the case that G has a cut of value $1-\epsilon$ and the case that G has a cut of value $1-\epsilon^\eta$. (Note that Kelner and Parrilo have a conjectured approach to achieve this.) Can you do this with arbitrarily small $\delta > 0$?

Problem 6: If you think the above cannot be done, even showing that the degree $d = 10$ (or even better, $d = \log^2 n$) SOS program cannot achieve this, even for the more general Max-2-LIN problem, would be quite interesting. As an intermediate step, prove or disprove Khot-Moshkovitz's question whether for an arbitrarily large constant c the Max-2-LIN instance they construct (where the degree d SOS value is $1-\epsilon$) has actual value at most $1 - c\epsilon$. Some intermediate steps that could be significantly easier are: the Khot-Moshkovitz construction is a reduction from a k-CSP on N variables that first considers all n-sized subsets of the N original variables and then applies a certain encoding to each one of those $\binom{N}{n}$ "cloud". Prove that if they used a single cloud then the reduction would be "sound" in the sense that there would be no integral solution of value larger than $1 - c\epsilon$.[1] Another statement that can show the challenge in proving the soundness of the KM construction: Recall that the KM boundary test takes a function $f : \mathbb{R}^n \to \{\pm 1\}$ and checks if $f(x) = f(y)$ where x and y have standard Gaussian coordinates that are each $1-\alpha$ correlated for some $\alpha \ll 1/n$. Their intended solution $f(x) = (-1)^{\lfloor \langle a,x\rangle \rfloor}$ for $a \in \{\pm 1\}^n$ will fail the test with probability $O(\sqrt{\alpha n})$. Prove that there is a function f that passes the test with $c\sqrt{\alpha n}$ for some c but such that for every constant d and function g of the form $g(x) = (-1)^{\lfloor p(x) \rfloor}$ where p a polynomial of degree at most d, $|\mathbb{E}\, p(x) f(x)| = o(1/n)$.

Problem 7: Show that there are some constant $\eta < 1/2$ and d, such that the degree d-SOS program yields an $O(\log^\eta n)$ approximation to the *Sparsest Cut* problem. If you think this can't be done, even showing that the $d = 8$ algorithm doesn't beat $O(\sqrt{\log n})$ would be very interesting.

Problem 8: Give a polynomial-time algorithm that for some sufficiently small $\epsilon > 0$, can (approximately) recover a planted ϵn-sparse vector v_0 inside a random subspace $V \subseteq \mathbb{R}^n$ of dimension

[1] This should be significantly easier to prove than the soundness of the Khot-Moshkovitz construction since it completely does away with their consistency test; still to my knowledge it is not proven in their paper. The reduction will not be "complete" in this case, since it will have more than exponential blowup and will not preserve SOS solutions; but I still view this as an interesting step. Also if this step is completed, perhaps one can think of other ways than the "cloud" approach of KM to reduce the blowup of this reduction to $2^{\delta N}$ for some small $\delta > 0$; perhaps a "biased" version of their code could work as well.

$\ell = n^{0.6}$. That is, we choose $v_1, \ldots, v_\ell$ as random Gaussian vectors, and the algorithm gets an arbitrary basis for the span of $\{v_0, v_1, \ldots, v_\ell\}$. Can you extend this to larger dimensions? Can you give a quasipolynomial time algorithm that works when V has dimension $\Omega(n)$? Can you give a quasipolynomial time algorithm for certifying the *Restricted Isometry Property* (RIP) of a random matrix?

Problem 9: Improve the dictionary learning algorithm of [Barak-Kelner-Steurer] (in the setting of constant sparsity) from *quasipolynomial* to *polynomial* time.

Problem 10: (Suggested by Prasad Raghavendra.) Can SDP relaxations simulate local search? While sum of squares SDP relaxations yield the best known approximations for CSPs, the same is not known for bounded degree CSPs. For instance, MAXCUT on bounded degree graphs can be approximated better than the Goemans-Willamson constant 0.878.. via a combination of SDP rounding and local search. Here local search refers to improving the value of the solution by locally modifying the values. Show that for every constant Δ, there is some $\epsilon > 0, d \in \mathbb{N}$ such that d rounds of SOS yield an $0.878.. + \epsilon$ approximation for MAXCUT on graphs of maximum degree Δ. Another problem to consider is maximum matching in 3-uniform hypergraphs. This can be approximated to a 3/4 factor using only local search (no LP/SDP relaxations), and some natural relaxations have a 1/2 integrality gap for it. Show that for every $\epsilon > 0$, $O(1)$ rounds of SOS give a $3/4 - \epsilon$ approximation for this problem, or rule this out via an integrality gap.

Problem 11: (Suggested by Ryan O'Donnell) Let G be the n vertex graph on $\{0, 1 \ldots, n-1\}$ where we connect every two vertices i, j such that their distance (mod n) is at most Δ for some constant Δ. The set S of $n/2$ vertices with east expansion is an arc. Can we prove this with an SOS proof of constant (independent of Δ) degree? For every $\delta > 0$ there is a c such that if we let G be the graph with $n = 2^\ell$ vertices corresponding to $\{0,1\}^\ell$ where we connect vertices x, y if their Hamming distance is at most $c\sqrt{n}$, then for every subsets A, B of $\{0,1\}^\ell$ satisfying $|A|, |B| \geq \delta n$, there is an edge between A and B. Can we prove this with an SOS proof of constant degree?

Fuzzier problems

The following problems are not as well-defined, but this does not mean they are less important.

Problem 12: Find more problems in the area of unsupervised learning where one can obtain an efficient algorithm by giving a proof of identifiability using low degree SOS.

Problem 13: The notion of pseudo-distributions gives rise to a computational analog of Bayesian reasoning about the knowledge of a computationally-bounded observer. Can we give any interesting applications of this? Perhaps in economics? Or cryptography?

SOS, Cryptography, and NP $\cap$ coNP. It sometimes seems as if in the context of combinatorial optimization it holds that "**NP** $\cap$ **coNP** = **P**", or in other words that all proof systems are automatizable. Can the SOS algorithm give any justification to this intuition? In contrast note that we do not believe that this assertion is actually true in general. Indeed, many of our candidates for public key encryption (though not all— see discussion in [Applebaum,Barak, Wigderson]) fall inside **NP** $\cap$ **coNP** (or **AM** $\cap$ **coAM**). Can SOS shed any light on this phenonmenon? A major issue in cryptography is (to quote Adi Shamir) the lack of diversity in the "gene pool" of problems that can be used as a basis for public key encryption. If quantum computers are built, then essentially the only well-tested candidates are based on a single problem— Regev's "Learning With Errors" (LWE) assumption (closely related to various problems on integer lattices). Some concrete questions along these lines are:

Problem 14: Find some evidence to the conjecture of Barak-Kindler-Steurer (or other similar conjectures) that the SOS algorithm might be optimal even in an *average case* setting. Can you find applications for this conjecture in cryptography?

Problem 15: Can we use a conjectured optimality of SOS to give *public key encryption schemes*? Perhaps to justify the security of LWE? One barrier for the latter could be that breaking LWE and related lattice problems is in fact in $\mathbf{NP} \cap \mathbf{coNP}$ or $\mathbf{AM} \cap \mathbf{coAM}$.

Problem 16: Understand the role of *noise* in the performance of the SOS algorithm. The algorithm seems to be inherently noise robust, and it also seems that this is related to both its power and its weakness– as is demonstrated by cases such as solving linear equations where it cannot get close to the performance of the Gaussian elimination algorithm, but the latter is also extremely sensitive to noise. Can we get any formal justifications to this intuition? What is the right way to define noise robustness in general? If we believe that the SOS algorithm is optimal (even in some average case setting) for noisy problems, can we get any quantitative predictions to the amount of noise needed for this to hold? This may be related to the question above of getting *public key cryptography* from assuming the optimality of SOS in the average case (see Barak-Kindler-Steurer and Applebaum-Barak-Wigderson).

Problem 17: Related to this: is there a sense in which SOS is an optimal noise-robust algorithm or proof system? Are there natural stronger proof systems that are still automatizable (maybe corresponding to other convex programs such as hyperbolic programming, or maybe using a completely different paradigm)? Are there natural noise-robust algorithms for combinatorial optimizations that are *not* captured by the SOS framework? Are there natural stronger proof systems than SOS (even non automatizable ones) that are noise-robust and are stronger than SOS for natural combinatorial optimization problems? Can we understand better the role of the *feasible interpolation property* in this context?

Problem 18: I have suggested that the main reason that a ”robust” proof does not translate into an SOS proof is by use of the probabilistic method, but this is by no means a universal law and getting better intuition as to what types of arguments do and don’t translate into low degree SOS proofs is an important research direction. Ryan O’Donnell’s problems above present one challenge to this viewpoint. Another approach is to try to use techniques from derandomization such as use of additive combinatorics or the Zig-Zag product to obtain ”hard to SOS” proofs. In particular, is there an SOS proof that the graph constructed by Capalbo, Reingold, Vadhan and Wigderson (STOC 2002) is a ”lossless expander” (expansion larger than *degree*/2)? Are there SOS proofs for the pseudorandom properties of the condensers we construct in the work with Impagliazzo and Wigderson (FOCS 2004, SICOMP 2006) or other constructions using additive combinatorics? I would suspect the answer might be ”no”. (Indeed, this may be related to the planted clique question, as these tools were used to construct the best known Ramsey graphs.)

Bibliography

[ABS10] S. Arora, B. Barak, and D. Steurer. Subexponential algorithms for unique games and related problems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 563–572. IEEE, 2010.

[ACMM05] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev. $o(\sqrt{\log n})$ approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 573–581. ACM, 2005.

[ALP$^+$10] R. Adamczak, A. E. Litvak, A. Pajor, N. Tomczak-Jaegermann, et al. Quantitative estimates of the convergence of the empirical covariance matrix in log-concave ensembles. *J. Amer. Math. Soc*, 23(2):535–561, 2010.

[ALPTJ11] R. Adamczak, A. E. Litvak, A. Pajor, and N. Tomczak-Jaegermann. Sharp bounds on the rate of convergence of the empirical covariance matrix. *Comptes Rendus Mathematique*, 349(3):195–200, 2011.

[ARV09] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *Journal of the ACM (JACM)*, 56(2):5, 2009.

[BBH$^+$12a] B. Barak, F. G. Brandao, A. W. Harrow, J. Kelner, D. Steurer, and Y. Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 307–326. ACM, 2012.

[BBH$^+$12b] B. Barak, F. G. Brandao, A. W. Harrow, J. Kelner, D. Steurer, and Y. Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 307–326. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1245-5. doi: 10.1145/2213977.2214006.

[BHHS11] B. Barak, M. Hardt, T. Holenstein, and D. Steurer. Subsampling mathematical relaxations and average-case complexity. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 512–531. SIAM, 2011.

[BKS14] B. Barak, J. A. Kelner, and D. Steurer. Rounding sum-of-squares relaxations. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 31–40. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591886.

[BRS11] B. Barak, P. Raghavendra, and D. Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 472–481. IEEE, 2011.

[CLRS13] S. O. Chan, J. R. Lee, P. Raghavendra, and D. Steurer. Approximate constraint satisfaction requires large lp relaxations. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 350–359. IEEE, 2013.

[DS] D. Dadush and D. Steurer. personal communication.

[FS02] U. Feige and G. Schechtman. On the optimality of the random hyperplane rounding technique for max cut. *Random Structures & Algorithms*, 20(3):403–440, 2002.

[GS11] V. Guruswami and A. K. Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 482–491. IEEE, 2011.

[GW95] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[HD13] P. Hand and L. Demanet. Recovering the Sparsest Element in a Subspace. *ArXiv e-prints*, October 2013.

[Jan97] S. Janson. *Gaussian hilbert spaces*, volume 129. Cambridge university press, 1997.

[LR99] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM (JACM)*, 46(6):787–832, 1999.

[Mar77] B. Marley. Three little birds. In *Exodus*. 1977.

[O'D14] R. O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[RS10] P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 755–764. ACM, 2010.

[RST10] P. Raghavendra, D. Steurer, and M. Tulsiani. Reductions between expansion problems. *arXiv preprint arXiv:1011.2586*, 2010.

[SS12] W. Schudy and M. Sviridenko. Concentration and moment inequalities for polynomials of independent random variables. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 437–446. SIAM, 2012.

[SWW13] D. A. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 3087–3090. AAAI Press, 2013.

[Tao12] T. Tao. Topics in random matrix theory, volume 132 of graduate studies in mathematics. *American Mathematical Society*, 25:76, 2012.

[Tro12] J. A. Tropp. User-friendly tools for random matrices: An introduction. Technical report, DTIC Document, 2012.

[Ver10] R. Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[WS11] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms.* Cambridge University Press, 2011.