Lecture 0: The Sum-of-Squares "proofs to algorithms" paradigm

In this introductory lecture, we will introduce the sum-of-squares (SoS) algorithm and the "proofs to algorithms" paradigm, using the problem of robust mean estimation as an illustrative example. Some bibliographic remarks will be deferred to the end.

These notes have not been reviewed with the same scrutiny applied to formal publications. There may be errors.

1 Context: statistical inference with constrained resources

Statistics is all about drawing conclusions from data, using algorithms. This course will be concerned with **statistical inference**, which basically means that we want to draw *rigorous, provable* conclusions about the data and its underlying distribution. As mortal, non-omniscient practitioners of statistical inference, we are *resource-constrained*. We often have limited access to:

Data/Information: The more data we have, the more information we have, and the better we can understand the underlying distribution. However, collecting data requires time, money, and care; if, say, a team of scientists performs an experiment and collects some fixed number of observations, it is typically not easy for them to later collect more.

Computation: With our data in hand, we will run an algorithm on it to perform inference. We are limited in the amount of time (and memory) we can use to process the data: computing resources cost money, and we want to make our inference in a timely manner.

Our goal is to design algorithms that will use our resources, information and computation, as efficiently as possible; or alternatively, to understand when information- and computation-efficient algorithms are impossible. Sample-efficient inference is an age-old focus in statistics, and computation-efficient algorithms are the core topic of study in computer science. The focus of this course will be at the confluence of these concerns. As we will see, the interaction of limited information and computational resources produces interesting effects that we would remain blind to if we were studying them in isolation. Hence this more modern perspective is essential to understanding what is possible in statistical inference.

How do we measure information? The best way to model information content depends on the setting. In some situations, it makes sense to parametrize the quantity information in terms of *sample complexity*, that is, the number of data points that we see. In other situations, when we only expect to see a single sample, it makes more sense to parametrize the quantity of information in terms of a *signal-to-noise* ratio; we'll see some examples of this throughout the class.

Throughout much of the course, we'll be working in the *high-dimensional* setting, in which we think of the algorithm's input, our data, as lying in $\mathbb{R}^d$ where $d \to \infty$; our goal will be to design algorithms which perform inference for any $d \in \mathbb{N}$. At a first pass it makes sense to say that an algorithm is *sample-efficient* if the number of samples it requires, n, is at most polynomial in d. That is, there exists some fixed universal constant C, such that $\lim_{d\to\infty} \frac{n}{d^C} < \infty$. We'll write $n = O(d^C)$ for this specific C, and $n = \mathrm{poly}(d)$ if there exists such a C. Of course, the smaller this C, the more sample-efficient the algorithm is.

Even when we think of the dimension d as fixed, we may wish to know how many samples we need to guarantee error rate at most $\varepsilon \in [0,1]$; leaving this vague for now, we'll just say that an algorithm is considered sample-efficient if it uses $\mathrm{poly}(\frac{1}{\varepsilon})$ samples (though we may hope to do better, and at times it makes sense to tolerate worse).

How do we measure computation? Our input to our algorithm is our data; say, n samples $v_1, \dots, v_n \in \mathbb{R}^d$. All told, the input has size $m = n \cdot d \cdot B$, where B is the number of bits we used to represent each entry of the $\{v_i\}_{i\in[n]}$. It's reasonable that an algorithm should run in time proportional to the amount of input data, and we measure algorithmic efficiency asymptotically according to the size of the input. We'll say that an algorithm is *time-efficient* if it terminates in time $\mathrm{poly}(m)$. As above, we may sometimes wish to do even better than polynomial time; it is much better to have linear time ($O(m)$ time) algorithms.

Sometimes, we also think of *space* or memory as a computational resource. But mostly we'll be concerned with time-efficiency, so when we say "computational efficiency" we mean time-efficiency by default.

2 Proofs-to-Algorithms

The main topic of this course is an algorithmic methodology that we call *sum-of-squares* algorithms. The incredible thing about sum-of-squares algorithms is that they allow us to take a restricted class of mathematical proofs (called sum-of-squares proofs) about properties of our input, and automatically transform these proofs into algorithms. Rather than trying to define this rigorously from the start, let's illustrate it with an example. We consider the well-studied problem of robust mean estimation. Since our focus will be on the algorithmic methodology, I'll save comments on the history of this problem for later.

Problem 2.1 (Robust Mean Estimation)**.** Let D be a distribution over $\mathbb{R}^d$ with mean u, covariance $\Sigma \preceq \mathbb{1}$, and bounded 4th moments. Let $\varepsilon > 0$ be a real number. Our goal is to estimate the mean u from ε-corrupted samples: we observe $v_1, \dots, v_m \in \mathbb{R}^d$, a $(1-\varepsilon)$-fraction sampled iid from D and the remaining ε-fraction are adversarially-chosen vectors in $\mathbb{R}^d$.

There are certainly situations where this problem is impossible, for example, if $\varepsilon = 1$. So, when is estimating the mean u possible? We will give a *proof of identifiability*, showing that if $n = \mathrm{poly}(d)$ sufficiently large, then with high probability given samples $v_1, \dots, v_n$ it is possible to estimate u in $\|\cdot\|_2$ with error at most $O(\sqrt{\varepsilon})$. More specifically, we'll show:

Lemma 2.2. *If $n = \mathrm{poly}(d)$ sufficiently large, and if $S \subset [n]$ of size $|S| = (1-\varepsilon)n$ satisfies, for $u_S := \frac{1}{|S|}\sum_{i\in S} v_i$,*

$$\frac{1}{|S|}\sum_{i\in S}(v_i - u_S)(v_i - u_S)^\top \preceq 2 \cdot \mathbb{1},$$

then with high probability over the samples, $\|u_S - u\| \leqslant O(\sqrt{\varepsilon})$.

Further, with high probability the uncorrupted samples form such a set S. So this lemma automatically implies an estimation algorithm: perform a brute force search over subsets of $[n]$ until you find a set satisfying this condition, then use its empirical mean as an estimate of u. But this algorithm runs in time $\geqslant \binom{n}{(1-\varepsilon)n}$, which is exponential in n and thus **inefficient**.

However, because the proof of Lemma 2.2 itself will be a **low-degree sum-of-squares proof**, it will automatically give us an efficient algorithm, based on solving a semidefinite program. This is amazing!

2.1 Sum-of-squares proofs

We'll now define sum-of-squares proofs.

Definition 2.3. For multivariate polynomials $p, q \in \mathbb{R}[x]$ in variables $x \in \mathbb{R}^N$, we say that $p \geqslant q$ is a *degree-k sum-of-squares inequality* if there exist polynomials $s_1, s_2, \ldots \in \mathbb{R}[x]$ of degree at most $k/2$ such that

$$p - q = \sum_t s_t^2. \tag{1}$$

In such a case, we say that $p \geqslant q$ has a degree-k sum-of-squares proof.

We say that there is a *degree-k sum of squares proof of $p \geqslant q$ modulo the axioms $A = \{f_i = 0\}_{i\in[M]} \cup \{g_j \geqslant 0\}_{j\in[K]}$* if there exist polynomials $a_1, \ldots, a_M, b_1, \ldots, b_K, s_1, \ldots \in \mathbb{R}[x]$ such that $\deg(a_i f_i) \leqslant k$ for all $i \in [M]$ and $\deg(b_j^2 g_j) \leqslant k$, for all $j \in [K]$, $\deg(s_t^2) \leqslant k$ for all $t \geqslant 1$, and

$$p - q = \sum_t s_t^2 + \sum_{i=1}^{M} a_i f_i + \sum_{j=1}^{K} b_j^2 g_j. \tag{2}$$

If there is a degree-k sum-of-squares proof that $p \geqslant q$ modulo the axioms A, we write $A \vdash_k p \geqslant q$.

Notice that if we can write $p - q$ as in (1), then this indeed constitutes a proof that $p \geqslant q$ on any real input. The same is true for (2), assuming that all of the axioms in A hold.

You may ask, what kind of statements have sum-of-squares proofs? First, we'll give some meta-statements, and then some examples.

Theorem 2.4. *If $p, q \in \mathbb{R}[x]$ for $x \in \mathbb{R}$ (that is, $N = 1$) and $p \geqslant q$ then there is always a sum-of-squares proof of this fact of degree at most* $\max(\deg(p), \deg(q))$.

What about $N > 1$? For $N > 1$, there are p, q such that $p \geqslant q$ but there is no sum-of-squares proof of this fact. However, the following theorem gives an alternative result; the question goes back to David Hilbert's 17th problem, and its resolution is due to Artin, Krivine, and Stengle.

Theorem 2.5 (Positivestellensatz)**.** *Any non-negative (modulo the axioms A) polynomial can be written as a sum of squares of rational functions.*

In most of the settings that we consider, where we have additional constraints on x such as $\|x\|^2 \leqslant 1$ or $x_i^2 = x_i$, every true polynomial inequality has a sum-of-squares proof (of degree as large as $\mathrm{poly}(N)$).

Examples of sum-of-squares proofs

Now we will give some examples of well-known facts that have low-degree sum-of-squares proofs.

Claim 2.6 (SoS Cauchy-Schwarz)**.** Let a, b be vector-valued polynomials of degree at most k. Then for any $\varepsilon > 0$,

$$\vdash_{2k} \langle a, b\rangle \leqslant \frac{\varepsilon}{2}\|a\|^2 + \frac{1}{2\varepsilon}\|b\|^2$$

and

$$\vdash_{4k} \langle a, b\rangle^2 \leqslant \|a\|^2\|b\|^2.$$

Proof. We can write $\langle a, b\rangle + \frac{1}{2}\|\sqrt{\varepsilon}a - \frac{1}{\sqrt{\varepsilon}}b\|^2 = \frac{\varepsilon}{2}\|a\|^2 + \frac{1}{2\varepsilon}\|b\|^2$, and $\langle a, b\rangle^2 = \|a\|^2\|b\|^2 - \frac{1}{2}\left(\sum_{i,j}(a_i b_j - a_j b_i)^2\right)$. □

Claim 2.7 (SoS operator norm)**.** Let $y \in \mathbb{R}^n$, $M \in \mathbb{R}^{n \times n}$, and $B \in \mathbb{R}^{n \times k}$. Then

$$\{M = \lambda \mathbb{1} - BB^\top\} \vdash_k y^\top M y \leqslant \lambda \|y\|^2,$$

for $k \geqslant \deg(y^\top M y + y^\top B B^\top y)$.

Proof. Our axioms imply that $y^\top M y = \lambda y^\top \mathbb{1} y - y^\top B B^\top y = \lambda \|y\|^2 - \|B^\top y\|^2$, which is a sum-of-squares proof that $y^\top M y \leqslant \lambda \|y\|^2$. □

2.2 Pseudoexpectations

The connection between sum-of-squares proofs and algorithms derives from the fact that if $A \vdash_k p \geqslant q$,[1] and A, p, q have at most N variables and $M + K$ axioms, then there is an algorithm running in time $(MNK)^{O(k)}$, which solves a *semidefinite program* to find such a proof. We'll prove this in the next class.

Here, we'll use a consequence of this statement (derived via convex duality). We'll need the following definition:

Definition 2.8. For a set of polynomial axioms $A = \{f_i = 0\}_{i \in [M]} \cup \{g_j \geqslant 0\}_{j \in [K]}$, we say that $\widetilde{\mathbf{E}} : \mathbb{R}[x] \to \mathbb{R}$ is a *degree-k pseudoexpectation satisfying A* if it is a linear operator with the following properties:

1. Scaling: $\widetilde{\mathbf{E}}[1] = 1$
2. Non-negativity of squares: $\widetilde{\mathbf{E}}[h^2] \geqslant 0$ for any polynomial $h \in \mathbb{R}[x]$ with $\deg(h) \leqslant k/2$
3. Respecting axioms: $\widetilde{\mathbf{E}}[a f_i] = 0$ for all $i \in [M]$ and $a \in \mathbb{R}[x]$ satisfying $\deg(a f_i) \leqslant k$, and $\widetilde{\mathbf{E}}[b^2 g_j] \geqslant 0$ for all $j \in [K]$ and $b \in \mathbb{R}[x]$ satisfying $\deg(b^2 g_j) \leqslant k$.

The pseudoexpectation operator is a *relaxation* of an expectation operator for some distributions over solutions x to the polynomial system of equations defined by A. We have no guarantee that $\widetilde{\mathbf{E}}$ corresponds to an actual distribution over solutions; on the other hand, it behaves a little bit like the expectation of a distribution. In particular, if $A \vdash_k p \geqslant q$, then we must also have $\widetilde{\mathbf{E}} p \geqslant \widetilde{\mathbf{E}} q$.

The following theorem we will prove in the next class:

Theorem 2.9. *Let $A = \{f_i = 0\}_{i \in [M]} \cup \{g_j \geqslant 0\}_{j \in [K]}$ be a set of polynomial axioms with $f_i, g_j \in \mathbb{R}[x]$ and $x \in \mathbb{R}^N$. If there exists a degree-k pseudoexpectation satisfying A, then there is an $(MKN)^{O(k)}$ time algorithm that finds it.*

Notice it's enough for the algorithm to print out the value of $\widetilde{\mathbf{E}}[x^\alpha]$ for all N^k degree-k monomials x^α; by linearity we can know its value on the rest.

2.3 A sum-of-squares proof of identifiability for robust mean estimation

Now, we set up a system of polynomial equations to solve robust mean estimation. Recall that we observe $v_1, \dots, v_n \in \mathbb{R}^d$, with the guarantee that $(1 - \varepsilon)n$ are drawn from D and εn may be arbitrary. We can think of this as a two-step process: first, n "pure" samples $z_1, \dots, z_n$ are sampled independently from D, and we view the "corrupted" copies $v_1, \dots, v_n$, with the guarantee that for $(1 - \varepsilon)n$ coordinates $i \in [n]$, $v_i = z_i$.

We'll take the following axioms, where the goal is that the solution to the system of polynomial equations identifies the uncorrupted samples.

[1] Technically we also require that the sum-of-squares proof that $p \geqslant q$ mod A has polynomial bit complexity; we'll brush this under the rug here

Problem 2.10 (Polynomial system for robust mean estimation)**.** We define a polynomial system $\mathcal{A}$ in the following variables: $Z_1, \dots, Z_n \in \mathbb{R}^d$ represent the vectors $z_1, \dots, z_n$; $W_1, \dots, W_n \in \mathbb{R}$ with W_i representing the indicator that $z_i = v_i$ (or that $i \in S$ for the S of Lemma 2.2); $B \in \mathbb{R}^{d \times d}$ is a matrix of "slack" variables. We include the following polynomial constraints:

$$W_i^2 = W_i \quad \forall\, i \in [n] \tag{3}$$

$$\sum_{i=1}^{m} W_i = (1-\varepsilon)n \tag{4}$$

$$W_i(Z_i - v_i) = 0 \quad \forall\, i \in [n] \tag{5}$$

$$\overline{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i, \qquad \frac{1}{n}\sum_{i=1}^{n}(Z_i - \overline{Z})(Z_i - \overline{Z})^\top = 2\mathbb{1} - BB^\top. \tag{6}$$

The constraints from (3) enforce that the W_i are $0/1$ valued. The constraints from (4) and (5) together ensure that $(1-\varepsilon)n$ of the Z_i are equal to the corresponding v_i. Finally, the constraint (6) ensures that the covariance matrix of the Z_i is bounded by $2\mathbb{1}$. We have normalized the averages by n rather than by $|S| = (1-\varepsilon)n$; this will be more convenient to work with. You can check that this is not a big deal for the final result, since this is a $(1 \pm O(\varepsilon))$ difference.

We'll show the following:

Lemma 2.11. *Let $\overline{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$ be the empirical mean of the uncorrupted samples. So long as $n = \mathrm{poly}(d)$, with high probability over $v_1, \dots, v_n$,*

$$\mathcal{A} \vdash_6 \|\overline{Z} - \overline{z}\|^4 \leqslant O(\varepsilon) \cdot \|\overline{Z} - \overline{z}\|^2.$$

We'll prove the lemma shortly; first, we make a couple of observations.

Identifiability. From this lemma, we get Lemma 2.2 immediately: in any solution to the polynomial system, the vectors v_i for which $W_i = 1$ form the set S have the property that $\|\overline{z} - \overline{Z}\| \leqslant O(\sqrt{\varepsilon})$. So long as $n = \mathrm{poly}(d)$ is large enough, with high probability $\|\overline{z} - u\| \leqslant \sqrt{\varepsilon}$, and the conclusion follows from the triangle inequality.

Proofs-to-algorithms. Because this is a low-degree sum-of-squares proof, we also automatically get a sum-of-squares algorithm! In polynomial time, we solve for a degree-6 pseudoexpectation operator which satisfies $\mathcal{A}$, as guaranteed by Theorem 2.9. From Lemma 2.11, we are guaranteed that $\widetilde{\mathbf{E}}[\|\overline{Z} - \overline{z}\|^4] \leqslant O(\varepsilon) \cdot \widetilde{\mathbf{E}}[\|\overline{Z} - \overline{z}\|^2]$. Then, by the non-negativity of $\widetilde{\mathbf{E}}$ applied to squares,

$$0 \leqslant \widetilde{\mathbf{E}}\left[\left(\|\overline{z} - \overline{Z}\|^2 - \widetilde{\mathbf{E}}[\|\overline{z} - \overline{Z}\|^2]\right)^2\right] = \widetilde{\mathbf{E}}[\|\overline{z} - \overline{Z}\|^4] - \widetilde{\mathbf{E}}[\|\overline{z} - \overline{Z}\|^2]^2 \leqslant \widetilde{\mathbf{E}}[\|\overline{z} - \overline{Z}\|^2](O(\varepsilon) - \widetilde{\mathbf{E}}[\|\overline{z} - \overline{Z}\|^2]),$$

which implies that $\widetilde{\mathbf{E}}[\|\overline{z} - \overline{Z}\|^2] \leqslant O(\varepsilon)$. By a similar logic, $\|\overline{z} - \widetilde{\mathbf{E}}[\overline{Z}]\|^2 \leqslant O(\varepsilon)$. So the quantity $\widetilde{\mathbf{E}}[\overline{Z}]$ computed by our algorithm is a good estimate for $\overline{z}$.

We'll now finally prove Lemma 2.11. The proof will be dull—how could it not be, when the point is that every inequality is certified as a sum-of-squares? The beauty is that, once we know that this simple, dull proof exists, we also know that a computer can find it and find a corresponding $\widetilde{\mathbf{E}}$.

Proof. Let $z_1, \dots, z_n$ be the uncorrupted samples from D, such that $v_i = z_i$ for a $(1-\varepsilon)$ fraction of $i \in [n]$. Recall $\overline{Z} = \frac{1}{n}\sum_{i=1}^n Z_i$, and define $\Sigma_Z = \frac{1}{n}\sum_{i=1}^n (Z_i - \overline{Z})(Z_i - \overline{Z})^\top$. Recall also that W_i is our variable which represents $\mathbf{1}_{Z_i = v_i}$. Let $\overline{z} = \frac{1}{n}\sum_{i=1}^n z_i$, and since we have chosen n large enough, with high probability the empirical covariance of the uncorrupted samples concentrates, so that $\Sigma_z = \mathbf{Cov}(z_1, \dots, z_n) \preceq 2\mathbb{1}$. We have that

$$\|\overline{z} - \overline{Z}\|^4 = \langle \overline{z} - \overline{Z}, \overline{z} - \overline{Z}\rangle^2 = \left(\frac{1}{n}\sum_{i=1}^n (1 - W_i \mathbf{1}_{z_i = v_i}) \langle z_i - Z_i, \overline{z} - \overline{Z}\rangle + \frac{1}{n}\sum_{i=1}^n W_i \mathbf{1}_{z_i = v_i} \langle z_i - Z_i, \overline{z} - \overline{Z}\rangle \right)^2.$$

Since we have enforced the constraint $W_i(v_i - Z_i) = 0$ in (5), the second term is 0. So we have

$$= \left(\frac{1}{n}\sum_{i=1}^n (1 - W_i \mathbf{1}_{z_i = v_i}) \langle z_i - Z_i, \overline{z} - \overline{Z}\rangle \right)^2$$

Now, we apply the $\vdash \langle p, q\rangle^2 \leqslant \|p\|^2\|q\|^2$ version of degree-6 SoS Cauchy-Schwarz (Claim 2.6),

$$\leqslant \left(\frac{1}{n}\sum_{i=1}^n (1 - W_i \mathbf{1}_{z_i = v_i})^2 \right) \left(\frac{1}{n}\sum_{i=1}^n \langle z_i - Z_i, \overline{z} - \overline{Z}\rangle^2 \right)$$

If $A \leqslant B$ is an SoS inequality, then so is $As \leqslant Bs$ for any sum-of-squares s, since $Bs - As = (B - A)s$. So, we can bound the parenthesized terms one at a time. For the first term, notice that (3) $\vdash_2 (1 - W_i \mathbf{1}_{z_i = v_i})^2 = 1 - W_i \mathbf{1}_{z_i = v_i}$.[2] Also, (4), (3) $\vdash_1 \frac{1}{n}\sum_{i=1}^n (1 - W_i \mathbf{1}_{z_i = v_i}) \leqslant 2\varepsilon$.[3] So (3), (4) $\vdash \frac{1}{m}\sum_{i=1}^n (1 - W_i \mathbf{1}_{z_i = v_i})^2 \leqslant 2\varepsilon$.

We now bound the second term. Using the shorthand $b = \overline{z} - \overline{Z}$, we expand

$$\langle z_i - Z_i, b\rangle = \langle z_i - Z_i + b - b, b\rangle = \langle z_i - \overline{z}, b\rangle - \langle Z_i - \overline{Z}, b\rangle + \|b\|^2,$$

and now applying these manipulations,

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^n \langle z_i - Z_i, \overline{z} - \overline{Z}\rangle^2 &= \frac{1}{n}\sum_{i=1}^n \left(\langle z_i - \overline{z}, b\rangle - \langle Z_i - \overline{Z}, b\rangle + \|b\|^2 \right)^2 \\ &\leqslant \frac{1}{n}\frac{10}{3}\sum_{i=1}^n \langle z_i - \overline{z}, b\rangle^2 + \langle Z_i - \overline{Z}, b\rangle^2 + \|b\|^4, \end{aligned}$$

Where we have used that for real A, B, C, $(A+B)^2 \leqslant 2A^2 + 2B^2$ (since $2A^2 + 2B^2 = (A+B)^2 + (A-B)^2$, and by similar reasoning $(A+B+C)^2 \leqslant 2A^2 + 4B^2 + 4C^2$, which can be improved to a uniform $\frac{10}{3}$ by averaging over permutations of A, B, C). Proceeding, we can re-write the above in terms of quadratic forms with the empirical covariance matrices of the Z_i and z_i,

$$= \frac{10}{3}\left(b^\top \Sigma_z b + b^\top \Sigma_Z b + \|b\|^4 \right),$$

And applying Claim 2.7 we have that (6) $\vdash_4 b^\top \Sigma_Z b \leqslant 2\|b\|^2, b^\top \Sigma_z b \leqslant 2\|b\|^2$ by the concentration of the spectrum of Σ_z, so we conclude that

$$\leqslant \frac{10}{3}(4\|b\|^2 + \|b\|^4).$$

So, putting everything together, we conclude that $\|\overline{z} - \overline{Z}\|^4 \leqslant O(\varepsilon) \cdot (4\|\overline{z} - \overline{Z}\|^2 + \|\overline{z} - \overline{Z}\|^4)$ has a degree-6 sum-of-squares proof, as desired. □

[2] Since $(1 - W_i \mathbf{1}_{z_i = v_i})^2 = 1 - 2W_i \mathbf{1}_{z_i = v_i} + W_i^2 \mathbf{1}^2_{z_i = v_i} = 1 - W_i \mathbf{1}_{z_i = v_i}$.

[3] Since $1 - W_i \mathbf{1}_{z_i = v_i} = 1 - W_i + W_i \mathbf{1}_{z_i \neq v_i}$, (4) $\vdash_1 \sum_i W_i = n(1-\varepsilon)$ and (3) $\vdash_2 W_i \mathbf{1}_{z_i \neq v_i} \leqslant \mathbf{1}_{z_i \neq v_i}$.

3 Conclusion

We have seen how a proof of identifiability which is captured by low-degree sum-of-squares proofs can automatically yield a polynomial time algorithm via sum-of-squares relaxations. This is the "sum-of-squares algorithmic paradigm" after which the course is named. The theme of proofs-to-algorithms will show up again and again throughout the course.

Bibliographic remarks. Sum-of-squares algorithms originated in several independent works by Lasserre [Las01], Nesterov [Nes00], Parrilo [Par00], and Shor [Sho87] near the end of the 20th century. The proofs-to-algorithms paradigm was popularized in the algorithms community starting with the work of Barak, Brandao, Harrow, Kelner, Steurer and Zhou [BBH+12] (see also [OZ13, BKS14, BKS15]). The proof of the SoS Cauchy-Schwarz inequality $\langle a, b\rangle^2 \leqslant \|a\|^2\|b\|^2$ is taken from Ma-Shi-Steurer [MSS16], Lemma A.1.

The problem of estimating the mean under adversarial corruptions goes back as far as the 1960's (e.g. [Ans60, Tuk60]). The first polynomial-time algorithm with dimension-independent error was given by Diakonikolas, Kamath, Kane, Li, Moitra, and Stewart [DKK+19] (see also [LRV16]); their convex programming approach bears some similarity to the SoS program that we use here, but the analysis is more complicated. Since then there have been numerous works on this topic, including the time- and sample-efficient algorithms [DL19, DHL19, CDGS20]. See e.g. [Li18] for a more complete survey. The presentation in this lecture was based on the works of Hopkins-Li [HL18] and Kothari-Steinhardt-Steurer [KSS18], with invaluable advice from Sam B. Hopkins. Thanks also to Sam for suggesting robust mean estimation as a topic for the introductory lecture.

Thanks to Jay Mardia and Louigi Addario-Berry for helpful suggestions in improving the presentation of these notes.

Contact. Comments are welcome at tselil@stanford.edu.

References

[Ans60] Frank J Anscombe. Rejection of outliers. *Technometrics*, 2(2):123–146, 1960. 7

[BBH+12] Boaz Barak, Fernando GSL Brandao, Aram W Harrow, Jonathan Kelner, David Steurer, and Yuan 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, 2012. 7

[BKS14] Boaz Barak, Jonathan A Kelner, and David Steurer. Rounding sum-of-squares relaxations. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 31–40, 2014. 7

[BKS15] Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 143–151, 2015. 7

[CDGS20] Yu Cheng, Ilias Diakonikolas, Rong Ge, and Mahdi Soltanolkotabi. High-dimensional robust mean estimation via gradient descent. In *International Conference on Machine Learning*, 2020. 7

[DHL19] Yihe Dong, Samuel Hopkins, and Jerry Li. Quantum entropy scoring for fast robust mean estimation and improved outlier detection. In *Advances in Neural Information Processing Systems*, pages 6067–6077, 2019. 7

[DKK+19] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019. 7

[DL19] Jules Depersin and Guillaume Lecué. Robust subgaussian estimation of a mean vector in nearly linear time. *arXiv preprint arXiv:1906.03058*, 2019. 7

[HL18] Samuel B Hopkins and Jerry Li. Mixture models, robustness, and sum of squares proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1034, 2018. 7

[KSS18] Pravesh K Kothari, Jacob Steinhardt, and David Steurer. Robust moment estimation and improved clustering via sum of squares. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1035–1046, 2018. 7

[Las01] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001. 7

[Li18] Jerry Zheng Li. *Principled approaches to robust machine learning and beyond.* PhD thesis, Massachusetts Institute of Technology, 2018. 7

[LRV16] Kevin A Lai, Anup B Rao, and Santosh Vempala. Agnostic estimation of mean and covariance. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674. IEEE, 2016. 7

[MSS16] Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 438–446. IEEE, 2016. 7

[Nes00] Yurii Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000. 7

[OZ13] Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*, pages 1537–1556. SIAM, 2013. 7

[Par00] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization.* PhD thesis, California Institute of Technology, 2000. 7

[Sho87] Naum Zuselevich Shor. An approach to obtaining global extremums in polynomial mathematical programming problems. *Cybernetics*, 23(5):695–700, 1987. 7

[Tuk60] John W Tukey. A survey of sampling from contaminated distributions. *Contributions to Probab. and Statist*, pages 448–485, 1960. 7