Semidefinite Programming Hierarchies

Daniel Alabi

1 Introduction

Recently, my collaborators and I released a paper [2] (also see concurrent work of [6]) about privately estimating a Gaussian using polynomial-time algorithms. We give the first computationally-efficient algorithms for estimating a Gaussian (in total variation distance) subject to pure or approximate differential privacy (DP) guarantees. In the pure DP setting, via a new lower bound, we show that a dependence on the condition number is necessary. However, in the approximate DP setting, our sample complexity bound does not depend on the condition number and the algorithms rely on a method of stabilizing convex relations. Our work leverages the *powerful* sum-of-squares framework, which I will try to discuss briefly—from my own vantage point—in this blog post. For a more pedagogical and complete introduction, please see textbooks or review articles (e.g., [1, 10, 4, 5]).

2 Semidefinite Programming

Semidefinite Programming (SDP) involves minimizing a linear function of a variable $x \in \mathbb{R}^m$ subject to a matrix inequality:

$$\min c^T x \text{ s.t. } F(x) \succeq 0, \tag{1}$$

where

$$F(x) = F_0 + \sum_{i=1}^{m} x_i F_i.$$

The vector $c \in \mathbb{R}^m$ and $m+1$ symmetric matrices $F_0, \ldots, F_m \in \mathbb{R}^{n \times n}$ are given to the SDP. The inequality $F(x) \succeq 0$ (a linear matrix inequality) enforces that $F(x)$ is positive semidefinite (PSD). i.e., $z^T F(x) z \geq 0$ for all $z \in \mathbb{R}^n$. An equivalent (and probably more standard) form of the SDP is to optimize the following objective over symmetric $n \times n$ matrices

$$\min \mathtt{Tr}(CX) \text{ s.t. } \mathtt{Tr}(A_i X) = b_i \, \forall i \in [m], X \succeq 0, \tag{2}$$

where $C, A_1, \ldots, A_m$ are symmetric $n \times n$ matrices and $\mathtt{Tr}$ is the trace operator. Note that $\mathtt{Tr}(CX) = \sum_{i,j}^{n} C_{ij} X_{ij}$.

An SDP is a convex optimization problem since the objective and constraints are convex. Also SDP includes linear programming (LP) as a special case. Thus, semidefinite programming is a generalization of linear programming where componentwise inequalities between vectors are replaced by matrix inequalities. Many LP solvers can handle semidefinite programs with some important caveats: duality results are weaker for semidefinite programs and some nonlinear, but convex,

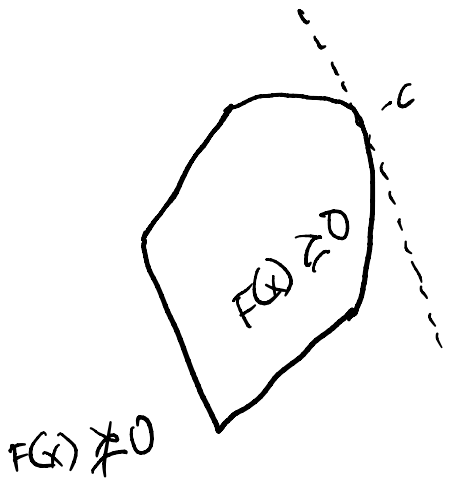

Figure 1: Semidefinite program with $x \in \mathbb{R}^n, F(x) \in \mathbb{R}^{m \times m}$ for some $n, m \in \mathbb{Z}$. e.g., $n = 2, m = 7$.

optimization problems can be cast as SDPs but not an LP. For example, consider the problem

$$\min \frac{(c^T x)^2}{d^T x} \text{ s.t. } Ax + b \geq 0, \tag{3}$$

where

$$d^T x > 0 \text{ whenever } Ax + b \geq 0.$$

Then using *Schur complements* (an exercise for the reader, perhaps) and the introduction of a new auxilliary variable t, we can reformulate the program as the following SDP:

$$\min t, \tag{4}$$

subject to

$$\begin{pmatrix} \texttt{diag}(Ax + b) & 0 & 0 \\ 0 & t & c^T x \\ 0 & c^T x & d^T x \end{pmatrix} \succeq 0.$$

$\texttt{diag}(M)$ denotes the diagonal matrix that represents all entries of M on the diagonal entries of the matrix $\texttt{diag}(M)$. We have thus reformulated the nonlinear but convex problem as a semidefinite program.

The main reason to study and utilize semidefinite programming is because the programs can be solved efficiently both in theory and practice. For example, see this Github page that contains instructions on how to run some SoS solvers: `https://github.com/`

`sums-of-squares/sos` [1].

3 Sum-of-Squares (SoS) Hierarchy

The SoS hierarchy is a family of convex relaxations to polynomial optimization problems (dating back to the early 2000s). The hierarchy was independently formulated by Parrilo, Lasserre, and Shor for the study of polynomial optimization [11, 8, 9, 12].

While its study began as a tool in the optimization and control literature, SoS has found profound use in proof systems. It has also sparked interest in possibly refuting conjectures related to hardness of approximation and average-case complexity [3]. In the area of average-case complexity, the goal is to design algorithms that perform well on typical instances, rather than every instance. For example, Max-Clique is NP-hard to approximate but a greedy algorithm can find a clique of size $\approx \log n$ in polynomial time. [3] shows that the degree-8 SoS hierarchy can efficiently solve integrality gap instances of the UGC (Unique Games Conjecture) problem that other linear and semidefinite programs cannot solve.

The degree-d SoS relaxation for any optimization problem with n variables has size $n^{O(d)}$ and can be shown to run in time $n^{O(d)}$. When $d = 2$, the SoS relaxation is a "simple" semidefinite program.

3.1 Polynomial Optimization

Let $C \subset \mathbb{R}^n$ be a convex region parameterized by polynomials $\{g_i\}_{i\in[m]}$ where for all $x \in C$, $g_i(x) = 0$. The goal is to optimize a polynomial f over C. As we shall see in Section 4, many important problems such as MaxCut can be represented in this form. Although the generality of polynomial optimization is appealing, it is not clear how much time (e.g., exponential or polynomial) is required to optimize the functions. Once the SDP is formulated, we can use the *Ellipsoid* algorithm to solve it (approximately) in polynomial time. [1]

Let $\mathcal{P} = (\min_{x\in C} f(x), C)$ be a polynomial optimization problem. We can relax the problem (by relaxing f, C to $\tilde{f}, \tilde{C}$ respectively) to $\mathcal{Q} = (\min_{y\in\tilde{C}} \tilde{f}(y), \tilde{C})$ so that

$$\texttt{val}(\mathcal{Q}) = \min_{y\in\tilde{C}} \tilde{f}(y) \leq \min_{x\in C} f(x) = \texttt{val}(\mathcal{P}).$$

Note that even though $\texttt{val}(\mathcal{Q}) \leq \texttt{val}(\mathcal{P})$, not every solution in $\mathcal{Q}$ will be satisfiable in $\mathcal{P}$, unless additional/higher-degree constraints are added.

3.2 Sum-of-Squares (SoS) Relaxations and Duality

Essentially, a degree-d SoS relaxation introduces a variable for each monomial of degree at most d. The relaxation also introduces affine constraints in these variables to mimic the polynomial constraints as well as some *eigenvalue constraints*. [2]

[1] Unlike LPs, no poly-time algorithms are known for solving SDPs exactly. The runtime for solving SDPs depends on the separation oracle for the convex body. For SoS, the separation oracle is well-defined in terms of *pseudo-distributions*. Essentially, a degree-d SoS "certificate" for non-negativity of a function f exists iff for all degree-d pseudo-distributions μ, the "expected value" of f over μ is at least 0 [5].

[2] A technical condition that induces the right formulation. e.g., can mimic treating squares of degree-$d/2$ polynomials as non-negative functions.

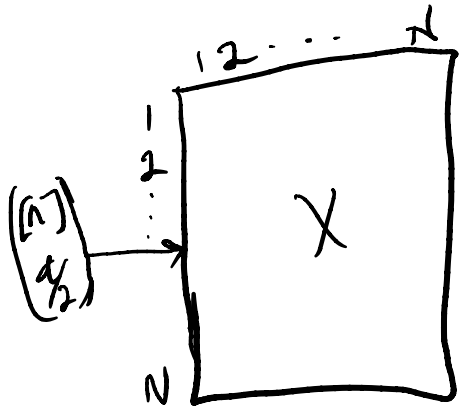

Figure 2: For $N = (n+1)^{d/2}$, we solve the SDP over $X \in \mathbb{R}^{N \times N}$.

4 MaxCut, SoS versus Other SDP Hierarchies

Karp famously showed that MaxCut is NP-hard. Let E be the set of edges in a graph. The problem can be formulated as

$$\max_{x \in \mathbb{R}^n} \sum_{(i,j) \in E} \frac{1}{2}(1 - x_i x_j), \text{ s.t. } x_i^2 = 1 \, \forall i \in [n]. \tag{5}$$

Consider the following degree-2 SoS relaxation of MaxCut

$$\max_{X \in \mathbb{R}^{n \times n}} \sum_{(i,j) \in E} \frac{1}{2}(1 - X_{ij}), \text{ s.t. } X_{ii} = 1 \, \forall i \in [n], X_\emptyset = 1, X \succeq 0. \tag{6}$$

which turns out to be equivalent to the Goemans-Williamson relaxation. Higher degree relaxations give better approximations but with a corresponding increase in runtime.

Thus far, we have only considered the *primal* view of SoS relaxations. The *dual* view is quite powerful: it gives us ways to certify that a SoS relaxation has a value $\geq c$ for some $c \in \mathbb{R}$. The certificate/proof is in terms of polynomials with degree at most d.

There are other semidefinite programming hierarchies, such as Sherali-Adams and the Approximate Lassserre hierarchies [7, 5]. In my opinion, the main difference between SoS and other hierarchies is that SoS treats all polynomials equally and others (e.g., Lasserre) do not and so are not agnostic to the basis choice.

5 Acknowledgements

Thanks to Daniel Hsu for comments on a preliminary draft.

References

[1] G. Valmorbida S. Prajna P. Seiler P. A. Parrilo M. M. Peet A. Papachristodoulou, J. Anderson and D. Jagt. SOSTOOLS: Sum of squares optimization toolbox for MATLAB. 2021. Available from `https://github.com/oxfordcontrol/SOSTOOLS`.

[2] Daniel Alabi, Pravesh K. Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. Privately estimating a gaussian: Efficient, robust and optimal. *CoRR*, abs/2212.08018, 2022.

[3] Boaz Barak, Fernando G.S.L. 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*, STOC '12, page 307–326, 2012.

[4] Grigoriy Blekherman, Pablo A. Parrilo, Rekha R. Thomas, Grigoriy Blekherman, Pablo A. Parrilo, and Rekha R. Thomas. *Semidefinite Optimization and Convex Algebraic Geometry*. Society for Industrial and Applied Mathematics, 2012.

[5] Noah Fleming, Pravesh Kothari, and Toniann Pitassi. Semialgebraic proofs and efficient algorithm design. *Foundations and Trends® in Theoretical Computer Science*, 14(1-2):1–221, 2019.

[6] Samuel B. Hopkins, Gautam Kamath, Mahbod Majid, and Shyam Narayanan. Robustness implies privacy in statistical estimation. *CoRR*, abs/2212.05015, 2022.

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

[8] Jean B. Lasserre. *New Positive Semidefinite Relaxations for Nonconvex Quadratic Programs*, pages 319–331. Springer US, Boston, MA, 2001.

[9] Jean B. Lasserre. Semidefinite programming vs. lp relaxations for polynomial programming. *Mathematics of Operations Research*, 27(2):347–360, 2002.

[10] M. Laurent. *Sums of squares, moment matrices and optimization over polynomials*, pages 155–270. Number 149 in The IMA Volumes in Mathematics and its Applications Series. Springer Verlag, Germany, 2009.

[11] Pablo A. Parrilo. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. *Dissertation (Ph.D.), California Institute of Technology*, 2000.

[12] Naum Z Shor. Quadratic optimization problems. *Soviet Journal of Computer and Systems Sciences*, 25(1):1–11, 1987.