[Lecture 10]

0.1 Semidefinite Programming and Sum-of-Squares

In the previous subsection, we saw how to approximately solve $\max_{\|v\|_\infty \le 1} v^\top \Sigma v$ via the semidefinite program defined by $\max_{M \succeq 0, \mathrm{diag}(M)=1} \langle M, \Sigma \rangle$. In this section we will cover semidefinite programming in more detail, and build up to *sum-of-squares programming*, which will be used to achieve error $\mathcal{O}(\epsilon^{1-1/k})$ when p^* has "certifiably bounded" kth moments (recall that we earlier achieved error $\mathcal{O}(\epsilon^{1-1/k})$ for bounded kth moments but did not have an efficient algorithm).

A **semidefinite program** is an optimization problem of the form

$$\begin{aligned} \text{maximize } & \langle A, X \rangle \\ \text{subject to } & X \succeq 0, \\ & \langle X, B_1 \rangle \le c_1, \\ & \vdots \\ & \langle X, B_m \rangle \le c_m. \end{aligned} \tag{1}$$

Here $\langle X, Y \rangle = \mathrm{tr}(X^T Y) = \sum_{ij} X_{ij} Y_{ij}$ is the inner product between matrices, which is the same as the elementwise dot product when considered as n^2-dimensional vectors.

Here the matrix A specifies the objective of the program, while (B_j, c_j) specify linear inequality constraints. We additionally have the positive semidefinite cone constraint that $X \succeq 0$, meaning that X must be symmetric with only non-negative eigenvalues. Each of A and $B_1, \ldots, B_m$ are $n \times n$ matrices while the c_j are scalars. We can equally well minimize as maximize by replacing A with $-A$.

While (1) is the canonical form for a semidefinite program, problems that are seemingly more complex can be reduced to this form. For one, we can add linear equality constraints as two-sided inequality constraints. In addition, we can replace $X \succeq 0$ with $\mathcal{L}(X) \succeq 0$ for any linear function $\mathcal{L}$, by using linear equality constraints to enforce the linear relations implied by $\mathcal{L}$. Finally, we can actually include any number of constraints $\mathcal{L}_1(X) \succeq 0$, $\mathcal{L}_k(X) \succeq 0$, since this is e.g. equivalent to the single constraint $\begin{bmatrix} \mathcal{L}_1(X) & 0 \\ 0 & \mathcal{L}_2(X) \end{bmatrix}$ when $k = 2$. As an example of these observations, the following (arbitrarily-chosen) optimization problem is also a semidefinite program:

$$\begin{aligned} \underset{x, M, Y}{\text{minimize }} & a^\top x + \langle A_1, M \rangle + \langle A_2, Y \rangle \\ \text{subject to } & M + Y \succeq \Sigma \\ & \mathrm{diag}(M) = 1 \\ & \mathrm{tr}(Y) \le 1 \\ & Y \succeq 0 \\ & \begin{bmatrix} 1 & x^\top \\ x & M \end{bmatrix} \succeq 0 \end{aligned} \tag{2}$$

(As a brief aside, the constraint $[1 \ x^\top; x \ M] \succeq 0$ is equivalent to $xx^\top \preceq M$ which is in turn equivalent to $x^\top M^{-1} x \le 1$ and $M \succeq 0$.)

Semidefinite constraints as quadratic polynomials. An alternative way of viewing the constraint $M \succeq 0$ is that the polynomial $p_M(v) = v^\top M v$ is non-negative for all $v \in \mathbb{R}^d$. More generally, if we have a non-hogoneous polynomial $p_{M,y,c}(v) = v^\top M v + y^\top v + c$, we have $p_{M,y,c}(v) \ge 0$ for all v if and only if $M' \succeq 0$ for $M' = \begin{bmatrix} c & y^\top/2 \\ y/2 & M \end{bmatrix} \succeq 0$.

This polynomial perspective is helpful for solving eigenvalue-type problems. For instance, $\|M\| \leq \lambda$ if and only if $v^\top M v \leq \lambda \|v\|_2^2$ for all v, which is eqvuialent to asking that $v^\top(\lambda I - M)v \geq 0$ for all v. Thus $\|M\|$ can be expressed as the solution to

$$\begin{aligned} &\text{minimize } \lambda \\ &\text{subject to } \lambda I - M \succeq 0 \text{ (equivalently, } v^\top(\lambda I - M)v \geq 0 \text{ for all } v) \end{aligned} \tag{3}$$

We thus begin to see a relationship between moments and *polynomial non-negativity constraints*.

Higher-degree polynomials. It is tempting to generalize the polynomial approach to higher moments. For instance, $M_4(p)$ denote the 4th moment tensor of p, i.e. the unique symmetric tensor such that

$$\langle M_4, v^{\otimes 4} \rangle = \mathbb{E}_{x \sim p}[\langle x - \mu, v \rangle^4]. \tag{4}$$

Note we can equivalently express $\langle M_4, v^{\otimes 4} \rangle = \sum_{ijkl} (M_4)_{ijkl} v_i v_j v_k v_l$, and hence $(M_4)_{ijkl} = \mathbb{E}[(x_i - \mu)(x_j - \mu)(x_k - \mu)(x_l - \mu)]$.

A distribution p has bounded 4th moment if and only if $\langle M, v^{\otimes 4} \rangle \leq \lambda \|v\|_2^4$ for all v. Letting $p_M(v) \stackrel{\text{def}}{=} \langle M, v^{\otimes 4} \rangle$, we thus can express the 4th moment of p as the polynomial program

$$\begin{aligned} &\text{minimize } \lambda \\ &\text{subject to } \lambda(v_1^2 + \cdots + v_d^2)^2 - p_M(v) \geq 0 \text{ for all } v \in \mathbb{R}^d \end{aligned} \tag{5}$$

Unfortunately, in constrast to (2), (5) is NP-hard to solve in general. We will next see a way to approximate (5) via a technique called *sum-of-squares programming*, which is a way of approximately reducing polynomial programs such as (5) to a large but polynomial-size semidefinite program.

Warm-up: certifying non-negativity over $\mathbb{R}$. Consider the one-dimensional polynomial

$$q(x) = 2x^4 + 2x^3 - x^2 + 5 \tag{6}$$

Is it the case that $q(x) \geq 0$ for all x? If so, how would we check this?

What if I told you that we had

$$q(x) = \frac{1}{2}(2x^2 + x - 3)^2 + \frac{1}{2}(3x + 1)^2 \tag{7}$$

Then, it is immediate that $q(x) \geq 0$ for all x, since it is a (weighted) sum of squares.

How can we construct such decompositions of q? First observe that we can re-write q as the matrix function

$$q(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \underbrace{\begin{bmatrix} 5 & 0 & 0 \\ 0 & -1 & 1 \\ 0 & 1 & 2 \end{bmatrix}}_{M} \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}. \tag{8}$$

On the other hand, the sum-of-squares decomposition for q implies that we can also write

$$q(x) = \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}^\top \left(\frac{1}{2} \begin{bmatrix} -3 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} -3 \\ 1 \\ 2 \end{bmatrix}^\top + \frac{1}{2} \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 3 \\ 0 \end{bmatrix}^\top \right) \begin{bmatrix} 1 \\ x \\ x^2 \end{bmatrix}, \tag{9}$$

i.e. we can decompose the matrix M defining $q(x) = [1; x; x^2]^\top M [1; x;^2\,]$ into a non-negative combination of rank-one outer products, which is true if and only if $M \succeq 0$.

There is one problem with this, which is that despite our successful decomposition of q, M is self-evidently not positive semidefinite! (For instance, $M_{22} = -1$.) The issue is that the matrix M defining $q(x)$ is not unique. Indeed, any $M(a) = \begin{bmatrix} 5 & 0 & -a \\ 0 & 2a-1 & 1 \\ -a & 1 & 2 \end{bmatrix}$ would give rise to the same $q(x)$, and a sum-of-squares decomposition merely implies that $M(a) \succeq 0$ for *some* a. Thus, we obtain the following characterization:

$$q(x) \text{ is a sum of squares } \sum_{j=1}^{k} q_j(x)^2 \iff M(a) \succeq 0 \text{ for some } a \in \mathbb{R}. \tag{10}$$

For the particular decomposition above we took $a = 3$.

Sum-of-squares in two dimensions. We can generalize the insights to higher-dimensional problems. Suppose for instance that we wish to check whether $q(x, y) = a_{40}x^4 + a_{31}x^3y + a_{22}x^2y^2 + a_{13}xy^3 + a_{04}y^4 + a_{30}x^3 + a_{21}x^2y + a_{12}xy^2 + a_{03}y^3 + a_{20}x^2 + a_{11}xy + a_{02}y^2 + a_{10}x + a_{01}y + a_{00}$ is non-negative for all x, y. Again, this is hard-to-check, but we can hope to check the sufficient condition that q is a sum-of-squares, which we will express as $q \succeq_{\text{sos}} 0$. As before this is equivalent to checking that a certain matrix is positive semidefinite. Observe that

$$q(x,y) = \begin{bmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{bmatrix}^\top \begin{bmatrix} a_{40} & a_{31}/2 & -b & a_{30}/2 & -c & -b' \\ a_{31}/2 & a_{22}+2b & a_{13}/2 & a_{21}/2+c & -c' & -c'' \\ -b & a_{13}/2 & a_{04} & a_{21}/2+c' & a_{03}/2 & -b'' \\ a_{30}/2 & a_{21}/2+c & a_{21}/2+c' & a_{20}+2b' & a_{11}/2+c'' & a_{10}/2 \\ -c & -c' & a_{03}/2 & a_{11}/2+c'' & a_{02}+2b'' & a_{01}/2 \\ -b' & -c'' & -b'' & a_{10}/2 & a_{01}/2 & a_{00} \end{bmatrix} \begin{bmatrix} x^2 \\ xy \\ y^2 \\ x \\ y \\ 1 \end{bmatrix} \tag{11}$$

for any b, b', b'', c, c', c''. Call the above expression $M(b, b', b'', c, c', c'')$, which is linear in each of its variables. Then we have $q \succeq_{\text{sos}} 0$ if and only if $M(b, b', b'', c, c', c'') \succeq 0$ for some setting of the bs and cs.

Sum-of-squares in arbitrary dimensions. In general, if we have a polynomial $q(x_1, \dots, x_d)$ in d variables, which has degree $2t$, then we can embed it as some matrix $M(b)$ (for decision variables b that capture the symmetries in M as above), and the dimensionality of M will be the number of monomials of degree at most t which turns out to be $\binom{d+t}{t} = \mathcal{O}((d+t)^t)$.

The upshot is that any constraint of the form $q \succeq_{\text{sos}} 0$, where q is linear in the decision variables, is a semidefinite constraint in disguise. Thus, we can solve any program of the form

$$\begin{aligned} &\underset{y}{\text{maximize}} \; c^\top y \\ &\text{subject to } q_1 \succeq_{\text{sos}} 0, \dots, q_m \succeq_{\text{sos}} 0, \end{aligned} \tag{12}$$

where the q_j are linear in the decision variables y. (And we are free to throw in any additional linear inequality or semidefinite constraints as well.) We refer to such optimization problems as *sum-of-squares programs*, in analogy to semidefinite programs.

Sum-of-squares for kth moment. Return again to the kth moment problem. As a polynomial program we sought to minimize λ such that $\lambda(v_1^2 + \cdots + v_d^2)^{k/2} - \langle M_{2k}, v^{\otimes 2k} \rangle$ was a non-negative polynomial. It is then natural to replace the non-negativity constraint with the constraint that $\lambda \|v\|_2^k - \langle M_{2k}, v^{\otimes 2k} \rangle \succeq_{\text{sos}} 0$. However, we actually have a bit more flexibility and it turns out that the best program to use is

$$\begin{aligned} &\text{minimize } \lambda \\ &\text{subject to } \lambda - \langle M_{2k}, v^{\otimes 2k} \rangle + (\|v\|_2^2 - 1)q(v) \succeq_{\text{sos}} 0 \text{ for some } q \text{ of degree at most } 2k-2 \end{aligned} \tag{13}$$

Note that the family of all such q can be linearly parameterized and so the above is indeed a sum-of-squares program. It is always at least as good as the previous program because we can take $q(v) = \lambda(1 + \|v\|_2^2 + \cdots + \|v\|_2^{2k-2})$.

When the solution λ^* to (13) is at most σ^{2k} for $M_{2k}(p)$, we say that p has $2k$th moment *certifiably bounded* by σ^{2k}. In this case a variation on the filtering algorithm achieves error $\mathcal{O}(\sigma\epsilon^{1-1/2k})$. We will not discuss this in detail, but the main issue we need to resolve to obtain a filtering algorithm is to find some appropriate tensor T such that $\langle T, M_{2k}\rangle = \lambda^*$ and T "looks like" the expectation of $v^{\otimes 2k}$ for some probability distribution over the unit sphere. Then we can filter using $\tau_i = \langle T, (x_i - \hat{\mu})^{\otimes 2k}\rangle$.

To obtain T requires computing the dual of (13), which requires more optimization theory than we have assumed from the reader, but it can be done in polynomial time. We refer to the corresponding T as a *pseudomoment* matrix. Speaking very roughly, T has all properties of a moment matrix that can be "proved using only sum-of-squares inequalities", which includes all properties that we needed for the filtering algorithm to work. We will henceforth ignore the issue of T and focus on assumptions on p that ensure that $M_{2k}(p)$ is certifiably bounded. The main such assumption is the *Poincaré inequality*, which we cover in the next section.