

Lecture 13: Semidefinite Programming

Scribe: Erasmo Tani

From Linear to Semidefinite Programming (SDP)

We can think of SDPs as a generalization of Linear Programming. We have seen two different relaxation: the metric relaxation that was solved by a linear program, and the spectral relaxation that was solved by an eigenvector problem. We will see today that SDP is the correct way to generalize the type of problem we solve in the spectral relaxation case.

Linear Programming Recap

Let's start with a generic LP:

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

The input data is the matrix $A \in \mathbb{R}^{m \times n}$, and the vectors $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. The constraint $x \geq 0$ is known as the conic constraint. The nice thing about linear programs is that one can define a natural notion of dual program, for which strong duality always hold. Consider the following program derived from the above one:

$$\begin{aligned} \max \quad & y^T b \\ \text{s.t.} \quad & A^T y \leq c \\ & y \in \mathbb{R}^m \end{aligned}$$

the optimal function value for the above program will always match the optimal function value for the previous one.

$$\max_{\substack{y \in \mathbb{R}^m \\ s \geq 0}} \min_{\substack{x \in \mathbb{R}^n \\ s \geq 0}} c^T x - y^T (Ax - b) - s^T (x - 0) = \min_{\substack{x \in \mathbb{R}^n \\ s \geq 0}} \max_{\substack{y \in \mathbb{R}^m \\ s \geq 0}} x^T (c - A^T y - s) + y^T b$$

SDPs

SDPs are a generalization of the eigenvector problems.:

$$\begin{aligned} \lambda_{\min}(A) = \max_{\substack{x \in \mathbb{R}^n \\ x^T x = 1}} x^T A x \end{aligned}$$

The above program is not convex, however we can change the parametrization of the solution space, so as to make it easier to solve it:

$$X = xx^T \quad X_{i,j} = x_i x_j$$

we then have:

$$\begin{aligned} A &= \max_{x \in \mathbb{R}^n} A \cdot X \\ I \cdot X &= 1 \\ X &\in \mathbb{R}^{n \times n} \end{aligned}$$

recall that we defined the dot (/ inner) product between two matrices to be:

$$A \cdot B = \text{Tr}(A^T B)$$

Note that we can use the fact that the trace is invariant under cyclic permutation of its arguments to see that:

$$\text{Tr}(x^T A x) = \text{Tr}(A x x^T) = \text{Tr}(A X) = A \cdot x x^T$$

Note that the program of the form:

$$\begin{aligned} A &= \max_{x \in \mathbb{R}^n} A \cdot X \\ I \cdot X &= 1 \\ X &\in S^{n \times n} \end{aligned}$$

is still not convex. Ideally we would like to enforce a constraint that makes X rank-1, however that also makes the program non-convex. We may instead optimize over convex combinations of rank-1 vectors:

$$X = \sum p_i x_i x_i^T$$

with:

$$\|x_i\|^2 = 1.$$

We then must have:

$$I \cdot X = \sum_i p_i = 1$$

and furthermore:

$$X \succeq 0.$$

Moreover, any X obeying:

$$\begin{aligned} I \cdot X &= 1 \\ X &\succeq 0 \end{aligned}$$

can be written as the convex combination of rank-1 matrices, with the coefficients living in the simplex. We have a linear objective, with a linear and a conic constraint: that is because:

$$PSD_n = \{X \in S^{n \times n} \mid X \succeq 0\}$$

is a cone. This can easily be shown: given $X, Y \in PSD_n$, we have:

$$z^T(aX + bY)z = a(z^T X z) + b(z^T Y z) \geq 0$$

for any $a, b \geq 0$.

An important thing about convex programs is that you can solve them efficiently if you have a polynomial time procedure to check whether a certain constraint is satisfied and produce a certificate otherwise. In this case we have algorithms that check with high precision whether a matrix is PSD or not.

Eigenvector Problem

Let's now see how SDPs can help us solve the eigenvector problem:

$$\begin{aligned} \max \quad & A \cdot X \\ & I \cdot X = 1 \\ & X \succeq 0 \end{aligned}$$

The above will solve the eigenvector problem.

The equivalent version for linear programming is:

$$\begin{aligned} \max \quad & a^T x \\ & q^T x = 1 \\ & x \geq 0 \end{aligned}$$

which solves for the maximum entry of the vector a . Fun fact: optimization problems for quantum computers are SDPs. Consider the following program:

$$\begin{aligned} \min \quad & \gamma \\ & A \preceq \gamma Y \end{aligned}$$

This is the dual of the above program.

In general, for a program:

$$\begin{aligned} \min \quad & C \cdot X \\ & \forall i \in [m], A_i \cdot X = b_i \\ & X \succeq 0 \end{aligned}$$

we can apply the lagrangian trick:

$$\begin{aligned} \max_{\substack{y \in \mathbb{R}^n \\ s \in \mathbb{R}^{n \times n} \\ S \succeq 0}} \min_X \quad & C \cdot X - \sum_i y_i (A_i \cdot X - b_i) - s(X - 0) \\ = \min_X \quad & \max_{\substack{y \in \mathbb{R}^n \\ s \in \mathbb{R}^{n \times n} \\ S \succeq 0}} X \cdot (C - \sum_i y_i A_i - s) + y^T b \end{aligned}$$

we then get:

$$\begin{aligned}
& \max \quad y^T b \\
& c = \sum y_i A_i + s \\
& \sum y_i A_i \preceq c \\
& y \in \mathbb{R}^m \\
& s \geq 0
\end{aligned}$$

SDP vector embeddings

if $X \succeq 0$ then it can be written as: $X = V^T V$ where V is the squared root of X . We then have:

$$X = \langle v_i, v_j \rangle$$

this is useful for instance in the following example: given a graph with Laplacian L , I want to find an embedding solving:

$$\begin{aligned}
& \min \quad \sum_{\{i,j\} \in E} ||v_i - v_j||^2 \iff L \cdot X \\
& \text{s.t.} \quad \sum_{i < j} ||v_i - v_j||^2 = 1 \iff L(K_G) \cdot X = 1 \\
& ||v_i - v_j||^2 = \langle ||v_i||^2 + ||v_j||^2 - 2\langle v_i, v_j \rangle = L_{ij} c(X).
\end{aligned}$$