Lecture 19: CS395T Numerical Optimization for Graphics and AI — Quadratic Programming

Qixing Huang
The University of Texas at Austin
huangqx@cs.utexas.edu

Disclaimer

This note is adapted from

- https://people.eecs.berkeley.edu/~satishr/cs270/sp11/rough-notes/SDP.pdf
- http://www.optimization-online.org/DB_FILE/2001/03/296.pdf
- http://mpc.zib.de/index.php/MPC/article/viewFile/40/20

1 Primal and Dual Program

Primal SDP:

$$\begin{aligned} \min \quad & \langle C, X \rangle & (1)\\ \text{subject to} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, \cdots, m, & (2)\\ & X \succeq 0. & (3) \end{aligned}$$

Dual SDP:

$$\begin{aligned} \max \quad & \langle \boldsymbol{b}, \boldsymbol{y} \rangle & (4)\\ \text{subject to} \quad & C \succeq \sum_{i=1}^{m} y_i A_i. & (5) \end{aligned}$$

Proposition 1. *Weak duality:*

$$\langle C, X \rangle \geq \langle \boldsymbol{b}, \boldsymbol{y} \rangle.$$

Strong duality under the Slater's condition.

2 Max-Cut Problem

The max cut problem is to partition a graph $G = (V, E)$ into two pieces $(S, \overline{S})$ such that the weight of the edges cut is maximized. If OPT is the weight of the maximum cut, there is a simple randomized algorithm that produces cuts with weight at least $OPT/2$. The algorithm constructs set S by independently adding vertices $v \in G$ to S with probability $1/2$. The indicator random variable $I_e = 1$ if the edge e is cut and 0 otherwise, by linearity of expectation we have,

$$\mathbb{E}[w(S, \overline{S})] = \sum_e \mathbb{E}[w(e) I_e] = \sum_e \frac{w_e}{2} \geq \frac{OPT}{2}.$$

The expected weight of the cut produced by the randomized algorithm is half the total weight, if the variance is large cuts with high weight would be produced frequently. The variance can be shown to be small for random graphs, also it is easy to derandomize the algorithm. The simple algorithm was the best known for a long time, in order to achieve an improvement let us write the max cut problem as an integer program,

$$\max \sum_{(i,j)\in E} w_{ij} \frac{1 - x_i x_j}{2} \qquad \forall i, x_i \in \{-1, 1\}.$$

As the integer program is NP hard to solve exactly we look for relaxations of the program that are easier to solve. Instead of one dimensional unit vectors, optimizing over vectors in n dimensional space we have the program,

$$\max \sum_{(i,j)\in E} w_{ij} \frac{1 - \boldsymbol{v}_i^T \boldsymbol{v}_j}{2} \qquad \forall i, \boldsymbol{v}_i \in \mathbb{R}^n,\ \|\boldsymbol{v}_i\| = 1.$$

The program is a semi definite program as the objective function and constraints are linear in the inner products $\boldsymbol{v}_i^T \boldsymbol{v}_j$. The optimal value of the program is denoted by $VP(OPT)$, the solution vectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ can be found by computing the Cholesky decomposition of the matrix A output by the SDP solver. As an example, let us consider the 5 cycle, the size of the maximum cut is 4 while the optimal solution to the vector program is two dimensional and corresponds to the embedding of the five star. The optimum value for the relaxed program is $\frac{5(1-\cos(\frac{4\pi}{5}))}{2} = 4.52$. The one dimensional solution corresponding to the maximum cuts is a solution to the relaxed problem, the value $VP(OPT)$ is therefore greater than OPT. We will show that starting with a solution $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n$ to the vector program a cut with value $0.878VP(OPT)$ can be found,

$$0.878VP(OPT) \leq OPT \leq VP(OPT).$$

Select a random hyper-plane $\boldsymbol{w}^T \boldsymbol{x} = 0$ through the origin and define $S := \{i | \boldsymbol{w}^T \boldsymbol{v}_i \geq 0\}$ to be the set of points that lie on one side of the hyperplane.

Proposition 2. *The expected weight of the cut $(S, \overline{S})$ is at least $0.878VP(OPT)$.*

Sketch Proof.

$$\frac{E[w(S,\overline{S})]}{VP(OPT)} = \sum_e \frac{2 w_e \theta}{w_e(1-\cos(\theta))} \geq \min_{\theta\in[0,\pi]} \frac{2\theta}{1-\cos(\theta)} = 0.878.$$

□

3 Numerical Algorithms Beyond Interior Point Methods

3.1 Solving Semidefinite Programs via Low-Rank Factorizations

Proposition 3. *Consider a semidefinite program over the form:*

$$\begin{aligned} \min \quad & \langle C, X\rangle \\ \text{subject to} \quad & A_i(X) = b_i, i = 1, \cdots, m, \\ & X \succeq 0. \end{aligned} \tag{6}$$

Then there exists an optimal solution $X^\star$ having rank r that satisfies $\frac{r(r+1)}{2} \leq m$.

Lemma 3.1. *Suppose that $X \in F$, where F is a face of the feasible set of (6). Let $d = \dim(F)$, $r = rank(X)$. Then*

$$\frac{r(r+1)}{2} \leq m + d.$$

This motivates us to reformulate (6) as

$$\begin{aligned} \min \quad & \langle C, RR^T \rangle \\ \text{subject to} \quad & A_i(RR^T) = b_i, i = 1, \cdots, m. \end{aligned} \tag{7}$$

We again consider its Lagrangian version:

$$\mathcal{L}(R, \boldsymbol{y}, \sigma) = \langle C, RR^T \rangle - \sum_{i=1}^{m} (A_i(RR^T) - b_i) + \frac{\sigma}{2} \sum_{i=1}^{m} \big(A_i(RR^T) - b_i\big)^2.$$

3.2 ADMM method

ADMM method considers the following Lagrangian (of the dual program):

$$\mathcal{L}_\mu(X, \boldsymbol{y}, S) := -\boldsymbol{b}^T \boldsymbol{y} + \langle X, \mathcal{A}^\star(\boldsymbol{y}) + S - C \rangle + \frac{1}{2\mu} \|\mathcal{A}^\star(\boldsymbol{y}) + S - C\|_{\mathcal{F}}^2.$$

ADMM applies the following recursion:

$$\boldsymbol{y}^{k+1} := \operatorname*{argmin}_{\boldsymbol{y} \in \mathbb{R}^m} \mathcal{L}_\mu(X^k, \boldsymbol{y}, S^k), \tag{8}$$

$$S^{k+1} := \operatorname*{argmin}_{S \in \mathcal{S}^n} \mathcal{L}_\mu(X^k, \boldsymbol{y}^{k+1}, S), \quad S \succeq 0, \tag{9}$$

$$X^{k+1} := X^k + \frac{\mathcal{A}^\star(y^{k+1}) + S^{k+1} - C}{\mu}. \tag{10}$$

After some derivations, we have

$$\boldsymbol{y}(S, X) := -(\mathcal{A}\mathcal{A}^\star)^{-1} \big(\mu(\mathcal{A}(X) - \boldsymbol{b}) + \mathcal{A}(S - C)\big).$$

S^{k+1} is given by

$$S^{k+1} = \operatorname*{argmin}_{S \succeq 0} \|S - \big(C - \mathcal{A}^*(\boldsymbol{y}^{k+1}) - \mu X^k\big)\|_{\mathcal{F}}^2.$$

The next lecture we will cover the proof of convergence.