Convex and Nonsmooth Optimization
HW4: Mostly about Semidefinite Programming

Michael Overton

Spring 2020

1. (20 pts) BV Ex 5.13

2. (40 pts) Consider the primal SDP

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

with

$$C = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, A_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, A_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, b_1 = 1, b_2 = 0$$

(a) Does the Slater condition hold for this primal SDP, i.e., does there exist a strictly feasible $\tilde{X}$?

(b) What is the optimal value of the primal SDP? Is it attained, and if so, by what X?

(c) Write down the dual SDP.

(d) Does the Slater condition hold for the dual SDP, i.e., does there exist a strictly feasible dual variable $\tilde{y}$?

(e) What is the optimal value of the dual SDP? Is it attained, and if so, by what dual variable y?

(f) Does strong duality hold?

(g) What can you say in general about strong duality if the Slater condition holds for at least one of a primal-dual pair of SDPs?

3. (40 pts) Exercise 5.39 on p. 285–286 in BV (see also p. 219–220). Assume that the matrix W is componentwise nonnegative, so that $W_{ij} = W_{ji}$ can be interpreted as a nonnegative weight on the edge joining vertex i to vertex j. As well as answering the questions in the exercise, also do the following. First, here is some useful information.

   - The easiest way to set up an SDP in CVX is to use "cvx_begin sdp" instead of "cvx_begin". Then, all matrix inequalities before the next "cvx_end" will be interpreted as semidefinite inequalities. Be sure to declare any symmetric matrix variables as symmetric, like this: "variable X(n,n) symmetric". See here for more details.
   - If you declare a variable as "dual variable Z" and then put ": Z" after an equality or inequality constraint, you will have access to the computed optimal dual variable for that constraint. If you want more than one dual variable, use "dual variables Z y" (no comma).
   - In MATLAB, and hence also CVX, if X is a matrix, diag(X) is a vector, and if x is a vector, diag(x) is a matrix. Type "help diag" for more.
   - Because the rank of a matrix is a discontinuous function, "rank(X)" is not a reliable way to find the approximate rank of a matrix, especially one that has been computed with CVX. Instead, compute the eigenvalues with "eig" (assuming X is symmetric) and estimate the rank from the eigenvalues.

   (a) Using W from data set 1 and data set 2, solve the SDP given in BV (5.114) with CVX.

   (b) Also, solve the SDP given in BV (5.115) by CVX and compare its optimal value with the one for (5.114). For the smaller data set 1, compare the computed optimal dual variables from (5.115) with the computed optimal primal variables from (5.114) and vice versa. What are the approximate ranks of the computed optimal primal and dual matrices? Do the matrices satisfy approximate complementarity, e.g. is the matrix product $X * Z$ approximately zero?

   (c) Here is another way to motivate the SDP relaxation (5.115). Instead of insisting that the variables x_i in (5.113) have the values ± 1, replace each scalar x_i by a vector $v_i \in \mathbb{R}^n$ with $\|v_i\|_2 = 1$, and then write $V = [v_1, \ldots, v_n]$ and $X = V^T V$. Does such an X

satisfy the constraints in (5.115)? This is the motivation used in Goemans and Williamson's celebrated 1994 paper and this leads to a simple randomized procedure for assigning the vertices to the two sets: see equations (1)–(3) on p. 1120 of their paper. Solving the SDP gives you $X \succeq 0$ and then you need V such that $X = V^T V$. Is such a V unique? If not, what is a convenient choice? Show that it gives you V whose columns have norm one as required. Would this work if X is exactly low rank, or low rank to machine precision, instead of only approximately low rank? Why or why not?

Using this V, carry out the assignment algorithm on p. 1120 for the data sets 1 and 2 using r with $r_i = 1/\sqrt{n}$ (instead of a random vector) and print the resulting partitioning of the vertices and the corresponding cut value. How does it compare to the optimal value of the SDP?

(d) Explicitly solve (5.113) for data set 1 only. This problem is NP-hard so you will have to write a brute force method to solve it: there is no way to do it efficiently, but it should run fast enough on the smaller data set 1. According to Goemans and Williamson, the optimal value in their SDP relaxation (which CVX computes in polynomial time up to a given accuracy) should be within a factor of ≈ 0.878 of the optimal value of the max cut problem. Is it? If not, perhaps there are some issues of scaling or constants that need working through.

An additional note of interest, not part of the homework: Håstad showed that there is no polynomial time algorithm to improve this max-cut guaranteed approximation factor from 0.878 to $16/17 \approx 0.941$ (assuming P $\neq$ NP), and Courant's Subhash Khot and his collaborators showed in this 2005 paper that if Subhash's "unique games conjecture" is true, then SDP is optimal for max-cut: one cannot get a better approximation guarantee than ≈ 0.878 in polynomial time unless P=NP. This would mean that SDP is somehow a very fundamental notion. Amazing!