MA796S/OR791K: Convex Programming and Interior Point Methods
Homework 1
Instructor: *Dr. Kartik Sivaramakrishnan*

INSTRUCTIONS

Due in class on Tuesday, September 11, 2007. You can work in groups of 2-3 students and submit the entire assignment as a group. No late homeworks will be accepted without prior instructor approval. The purpose of this homework is to set up and solve simple conic optimization problems in MATLAB.

1. We will use SDPT3 to solve conic optimization problems in this course. SDPT3 (latest version SDPT3-4.0) is a MATLAB based software. Your first job is to install SDPT3-4.0 on your unity account or personal computer. Please follow these instructions carefully:

   (a) Go to `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`. Download the file `SDPT3-4.0-beta.zip`.

   (b) Unpack the software using `gunzip SDPT3-4.0-beta.zip`.

   (c) Run MATLAB in the directory `SDPT3-4.0-beta`.

   (d) In the MATLAB command window, type: `Installmex`.

   (e) To ensure that you have installed SDPT3 correctly, type: `startup` and then `sqlpdemo`.

   The user's guide is at `http://www.math.nus.edu.sg/~mattohkc/guide4-0-draft.pdf`. You can also download some simple examples illustrating the use of SDPT3 from `http://www.math.nus.edu.sg/~mattohkc/sdpexample.html`.

2. The `Examples` directory in `SDPT3-4.0-beta` contains the file `maxcut.m`, that sets up the SDP relaxation of the maxcut problem that we considered in class, and also solves the relaxation using SDPT3. A nice discussion on the SDP relaxation for the maxcut problem appears in Example 9 on page 14 of Todd [3]. You should look through the file `maxcut.m` carefully to find the calling sequence for the program, and also understand how the program sets up the SDP and solves it using SDPT3. The input to the program is the adjacency matrix of a graph. Test the algorithm on the complete graph with 5 vertices whose adjacency matrix is

$$B = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

The optimal solution X to the SDP relaxation of the maxcut problem is contained in $X\{1\}$.

3. We will now implement the Goemans-Williamson rounding procedure (see Goemans & Williamson [2]) that we discussed in class to round the SDP solution X into an integer solution for the maxcut problem. Perform the following steps:

   (a) Compute a Cholesky factorization $V = \text{chol}(X)$ of the solution X to the SDP relaxation. Let v_i denote the ith column of the matrix V for $i = 1, \ldots, 5$.

   (b) Generate a random vector r on the unit hypersphere in $\mathbb{R}^5$ using the following commands in MATLAB: $r = \text{randn}(5,1)$ and $r = \dfrac{r}{\text{norm}(r,2)}$.

   (c) We will generate a cut as follows: For $i = 1, 2, \ldots, 5$; if $v_i^T r \geq 0$, assign vertex i to the set S; else, assign vertex i to the set V/S. Calculate the objective value for the cut that is generated.

   (d) Repeat steps (b) and (c) 1000 times. What is the average objective value of the cuts generated? What is the best objective value for all the cuts generated?

   (e) Find the optimal maxcut value for the complete graph with 5 vertices by inspection. What is the ratio of the average value of the cuts generated in step (d) and the optimal maxcut value? Is it better than the Goemans-Williamson approximation bound of 0.878? Moreover, how does the best objective value of the cut generated in part (d) compare with the optimal maxcut value?

4. Consider the Petersen graph (see `http://en.wikipedia.org/wiki/Petersen_graph`) with 10 vertices and 15 edges, whose adjacency matrix is given below

$$B = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

   Repeat parts (2) and (3) of the homework on the Petersen graph. It is difficult to estimate the optimal maxcut value for the Petersen graph by insepection. However, we will give bonus points to the group that returns the cut with the best objective value using the Goemans-Williamson algorithm :)

5. Consider the quadratic programming problem (QP)

$$\begin{array}{ll} \min & \frac{1}{2}x^T P x + q^T x + r \\ \text{s.t.} & -1 \leq x_i \leq 1, \quad i = 1, \ldots, 3. \end{array}$$

where

$$P = \begin{bmatrix} 13 & 12 & -2 \\ 12 & 17 & 6 \\ -2 & 6 & 12 \end{bmatrix}, \quad q = \begin{bmatrix} -22.0 \\ -14.5 \\ 13.0 \end{bmatrix}, \quad r = 1.$$

(a) Solve the QP using MATLAB's routine *quadprog*.

(b) Set up the QP as a second order cone program (SOCP) using the discussion in class. You can also look at Section 2.1 on pages 8-9 of Alizadeh & Goldafrb [1]. What is the dual SOCP problem?

(c) Solve the SOCP using SDPT3. Compare your answers with your results in part (a).

References

[1] F. Alizadeh and D. Goldfarb, *Second order cone programming*, Mathematical Programming, 95(2003), pp. 3-51. (see *Second Order Cone Programming* on class reading list).

[2] M. Goemans and D.P. Williamson, *Improved approximation algorithms for max cut and satisfiability problems using semidefinite programming*, Journal of A.C.M 42 (1995), pp. 1115-1145. (Reference 2 in *Applications in Combinatorial Optimization* in class reading list).

[3] M.J. Todd, *Semidefinite programming*, Acta Numerica, 10(2001), pp. 515-560. (Reference 3 in *Semidefinite Programming* in class reading list).