Convex Optimization

Homework # 2

Instructions

Please, submit your homeworks to alexey.barsukov@matfyz.cuni.cz not later than November, 20, 23:59 local time. The subject of your email should be of the following form: `Convex Optimization HW *#ofHW* *yourSurname* *yourName*`, for example: `Convex Optimization HW 2 Havlik Tomas`

Exercise 1 (10 points)

Find two 2×2 symmetric matrices A, B that are incomparable with respect to the cone $\mathbb{S}^2_+$ of positive semidefinite matrices, i.e., neither $A \preceq B$, nor $B \preceq A$.

Exercise 2 (10 points)

Let A and B be two square matrices (of shapes $n \times n$ and $m \times m$). Show that the block matrix $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$ lies in $\mathbb{S}^{n+m}_+$ if and only if $A \in \mathbb{S}^n_+$ and $B \in \mathbb{S}^m_+$.

Exercise 3 (20 points)

For a $n \times n$ matrix A, define its spectral norm $\|A\|_2$ as

$$\|A\|_2 = \sup\{\|Av\|_2 \mid v \in \mathbb{R}^n,\ \|v\|_2 = 1\}$$

Show that $\|A\|_2 = \inf\{s \in \mathbb{R}_+ \mid A^T A \preceq s^2 I_n\}$, where I_n is the $n \times n$ identity matrix.

Exercise 4 (25 points)

Our portfolio might contain four types of stocks, call them $1, 2, 3$, and 4. The profit from investing into these stocks is a random variable. If x_i is the amount invested into the ith stock, then the expected value of our profit is

$$r = 0.07x_1 + 0.08x_2 + 0.09x_3 + 0.1x_4$$

Assume that $x_1 + x_2 + x_3 + x_4 = 1$ and that $x_i \geq 0$. We know the covariance matrix

$$\Sigma = \begin{pmatrix} 1.1 & -1.3 & 2.5 & -0.9 \\ -1.3 & 6.5 & 0.7 & -1.5 \\ 2.5 & 0.7 & 11.1 & 0.7 \\ -0.9 & -1.5 & 0.7 & 16.1 \end{pmatrix}$$

which allows us to calculate the variance (or risk) of our profit as $x^T \Sigma x$. Our goal is to maximize the expected value and to minimize the risk. We do this by choosing our risk-aversion parameter γ and maximizing

$$r - \gamma x^T \Sigma x$$

Formulate this problem as a convex optimization problem and solve it for $\gamma \in \{0, 0.5, 1, 1.5, 2, 2.5, 3\}$.

Exercise 5 (35 points)

Let $G = (V, E)$ be an undirected graph on $|V| = n$ vertices. The problem $\mathrm{MaxCut}(G)$ searches for a cut of G that has the maximal size and can be represented as an optimization problem:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2} \sum_{(v_i, v_j) \in E} (1 - v_i v_j) \qquad (1) \\ \text{subject to} \quad & v_i \in \{-1, 1\} \text{ for } i = 1, \dots, n \end{aligned}$$

From lectures, you may know that this optimization problem has an SDP-relaxation $\mathrm{MaxCut}_{\mathrm{SD}}(G)$, where variables are allowed to take values on the sphere $S^{n-1} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$:

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2} \sum_{(v_i, v_j) \in E} (1 - v_i^T v_j) \\ \text{subject to} \quad & v_i \in S^{n-1} \text{ for } i = 1, \ldots, n \end{aligned}$$

1. Show that the following problem is equivalent to $\mathrm{MaxCut}_{\mathrm{SD}}(G)$

$$\begin{aligned} \text{maximize} \quad & \frac{1}{2}\mathrm{tr}(AA) - \frac{1}{4}\mathrm{tr}(AX) \\ \text{subject to} \quad & X_{ii} = 1 \text{ for all } i \in [n]; \\ & X \succeq 0, \end{aligned}$$

   where A is the adjacency matrix of G.

2. Let G be a graph with the following adjacency matrix: `https://tinyurl.com/ypy4jrw8`

   Using `cvxpy`, compute the optimal value for the problem $\mathrm{MaxCut}_{\mathrm{SD}}(G)$.

Exercise 6 (10 bonus points)

After obtaining the optimal solution for $\mathrm{MaxCut}_{\mathrm{SD}}(G)$ we would like to have a solution which approximates the original problem $\mathrm{MaxCut}(G)$. For this, we want to use the *rounding procedure*. Firstly, we find the vectors $v_1, \ldots, v_n \in S^{n-1}$ corresponding to the solution. For example, they can be found from the eigendecomposition $X = V^T L V$, where L is the diagonal matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and the columns of V are the corresponding eigenvectors that are exactly $v_1, \ldots, v_n$ modulo multiplying by scalars. Then, we *randomly* choose a hyperplane $c^T x = 0$ that goes through 0 and assign "1" to all v_i's on one side of the hyperplane and "-1" to all on the other side. That is, we implement the function

$$r_c(v_i) = \begin{cases} 1, & c^T v_i \geq 0, \\ -1, & \text{otherwise} \end{cases}$$

Using `cvxpy` (and `numpy`), apply the rounding procedure and obtain an approximated solution $s \in \{-1, 1\}^n$ for $\mathrm{MaxCut}(G)$. Plug the vector s in the objective function (1) of $\mathrm{MaxCut}(G)$ and compute the cut value for it. Repeat the rounding procedure several times, and write the best value obtained for (1). You will get the points if the obtained value is in the interval $[\alpha v^\star, v^\star]$, where $v^\star$ is the maximum cut value for G and $\alpha = 0.87856$. You do not need to send a source code for the program solving the problem.