Homework assignment #5

1. We have discussed the following technique for smoothing a nondifferentiable convex function $f(x)$: find the conjugate $f^*(y)$, add a small strongly convex term $d(y)$ to it, and take the conjugate $(f^* + d)^*$ of the modified conjugate. The Moreau–Yosida smoothing in lecture 8 is an example with $d(y) = (t/2)\|y\|_2^2$.

   In this problem, we work out two other examples. Find $(f^* + d)^*$ for the following combinations of f and d. In both problems, the variable x is an n-vector and μ is a positive constant.

   (a) $f(x) = \|x\|_1$ and $d(y) = \mu \sum_{i=1}^n (1 - \sqrt{1 - y_i^2})$.

   (b) $f(x) = \max_{i=1,\ldots,n} x_i$ and $d(y) = \mu(\sum_{i=1}^n y_i \log y_i + \log n)$.

2. *Projection on order cone.* Ordering constraints $x_1 \leq x_2 \leq \cdots \leq x_n$ arise in many applications. In this problem we discuss the Euclidean projection on the cone defined by these inequalities, *i.e.*, the problem

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x - a\|_2^2 \\ \text{subject to} & x_1 \leq x_2 \leq \cdots \leq x_n. \end{array} \tag{1}$$

   This is known in statistics as the *isotonic regression problem.* It can be written as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|x - a\|_2^2 \\ \text{subject to} & Ax \preceq 0, \end{array} \tag{2}$$

   where A is the $(n-1) \times n$ matrix

$$A = \left[\begin{array}{ccccccc} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & -1 \end{array}\right].$$

   The following algorithm is called the *Pool Adjacent Violators Algorithm.* We use the following notation. If β is a subset of $\{1, 2, \ldots, n\}$, then a_β is the subvector of a with elements indexed by β, and $\operatorname{avg}(a_\beta)$ denotes the average of the elements of the vector a_β. Thus, if $\beta = \{2, 3, 4\}$, then

$$a_\beta = (a_2, a_3, a_4), \qquad \operatorname{avg}(a_\beta) = \frac{a_2 + a_3 + a_4}{3}.$$

Pool Adjacent Violators Algorithm. Initially, $l = 1$ and $\beta_1 = \{1\}$. For $i = 2, \ldots, n$, execute the following steps.

(a) Set $l := l + 1$ and define $\beta_l = \{i\}$.

(b) While $\mathrm{avg}(a_{\beta_{l-1}}) \geq \mathrm{avg}(a_{\beta_l})$, merge the sets β_{l-1} and β_l:

$$\beta_{l-1} := \beta_{l-1} \cup \beta_l, \qquad l := l - 1.$$

An example is shown in Table 1. When the algorithm terminates, the sets $\beta_1, \ldots, \beta_l$ partition $\{1, 2, \ldots, n\}$. We show that the optimal solution of (1) is given by

$$x_{\beta_i} = \mathrm{avg}(a_{\beta_i})\mathbf{1}, \quad i = 1, \ldots, l. \tag{3}$$

(a) Show that x is optimal for (2) if and only if there exists an $(n-1)$-vector z with

$$Ax \preceq 0, \qquad z \succeq 0, \qquad z^T Ax = 0, \qquad x + A^T z = a.$$

(b) Verify that after cycle $i = 1, \ldots, n$ in the algorithm, the following properties hold.

i. The sets β_i are nonempty sets of consecutive indices in $\{1, 2, \ldots, n\}$ and they follow each other, *i.e.*, $\max \beta_k + 1 = \min \beta_{k+1}$ for $k = 1, \ldots, l-1$. Together, they partition $\{1, 2, \ldots, \max \beta_l\}$.

ii. The averages of the subvectors a_{β_k} are strictly increasing:

$$\mathrm{avg}(a_{\beta_k}) < \mathrm{avg}(a_{\beta_{k+1}}), \quad k = 1, \ldots, l-1.$$

iii. The cumulative sums of the vectors $a_{\beta_k} - \mathrm{avg}(a_{\beta_k})\mathbf{1}$ are nonnegative:

$$\mathrm{cs}(a_{\beta_k}) \succeq 0, \quad k = 1, \ldots, l,$$

where $\mathrm{cs}(u)$ is defined as

$$\mathrm{cs}(u) = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix} (u - \mathrm{avg}(u)\mathbf{1}).$$

(Note that the last element of $\mathrm{cs}(u)$ is necessarily zero.)

(c) Show that the optimality conditions in part (a) are satisfied by the vector x defined in (3) and the $(n-1)$-vector z defined by

$$z_{\beta_i} = \mathrm{cs}(a_{\beta_i}), \quad i = 1, \ldots, l-1, \qquad (z_{\bar{\beta}_l}, 0) = \mathrm{cs}(a_{\beta_l}),$$

where $\bar{\beta}_l = \beta_l \setminus \{n\}$.

(d) Explain why the complexity of the algorithm is linear in n.

i	Subvectors $a_{\beta_1}, \ldots, a_{\beta_l}$	Averages $\mathrm{avg}(a_{\beta_1}), \ldots, \mathrm{avg}(a_{\beta_l})$
1	$\boxed{7}$	7
2	$\boxed{7}\ \boxed{-8}$	7, −8
	$\boxed{7, -8}$	−1/2
3	$\boxed{7, -8}\ \boxed{-6}$	−1/2, −6
	$\boxed{7, -8, -6}$	−7/3
4	$\boxed{7, -8, -6}\ \boxed{18}$	−7/3, 18
5	$\boxed{7, -8, -6}\ \boxed{18}\ \boxed{-9}$	−7/3, 18, −9
	$\boxed{7, -8, -6}\ \boxed{18, -9}$	−7/3, 9/2
6	$\boxed{7, -8, -6}\ \boxed{18, -9}\ \boxed{4}$	−7/3, 9/2, 4
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}$	−7/3, 13/3
7	$\boxed{7, -8, -6)}\ \boxed{18, -9, 4}\ \boxed{16}$	−7/3, 13/3, 16
8	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}\ \boxed{17}$	−7/3, 13/3, 16, 17
9	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}\ \boxed{17}\ \boxed{-10}$	−7/3, 13/3, 16, 17, −10
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16}\ \boxed{17, -10}$	−7/3, 13/3, 16, 7/2
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16, 17, -10}$	−7/3, 13/3, 23/3
10	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16, 17, -10}\ \boxed{-8}$	−7/3, 13/3, 23/3, −8
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4}\ \boxed{16, 17, -10, -8}$	−7/3, 13/3, 15/4
	$\boxed{7, -8, -6}\ \boxed{18, -9, 4, 16, 17, -10, -8}$	−7/3, 4

Table 1: The projection of the vector $a = (7, -8, -6, 18, -9, 4, 16, 17, -10, -8)$ on the order cone is $x = (-7/3, -7/3, -7/3, 4, 4, 4, 4, 4, 4, 4)$.