Homework 5 for "Convex Optimization" Part. 1

1500010611 汪祎非

1 Problem 1

At first, we directly use CVX by calling solver *mosek*. It takes about $8.07s$, and the optimal value is 0.0851. Then, we use CVX by calling solver *gurobi*. It takes about $7.51s$, and the optimal value is 0.0851. The error of *cvx gurobi* to *cvx mosek* is 1.91×10^{-6}. We plot the exact solution and solutions from *cvx mosek* and *cvx gurobi*:

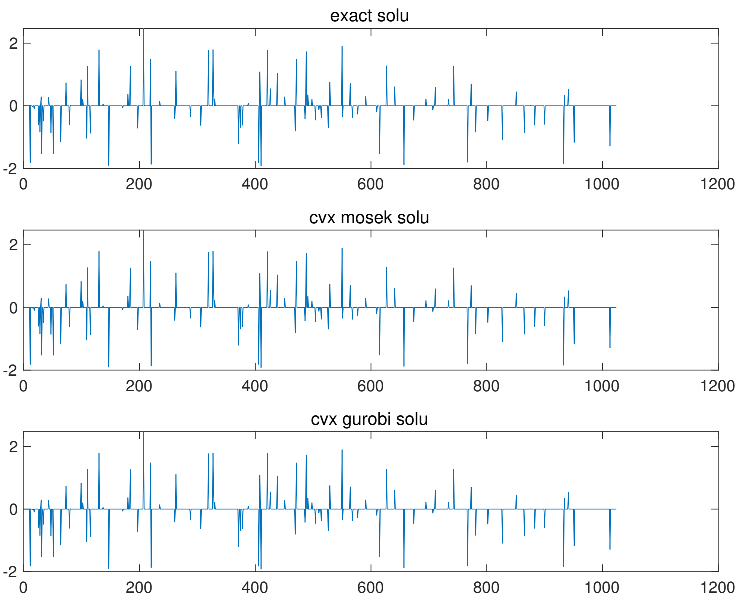

Figure 1 *Solutions*

From figure1, both *cvx mosek* and *cvx gurobi* give exactly the exact solution.

2 Problem 2

The l_1-regularized problem

$$\min_x \frac{1}{2}\|Ax-b\|_2^2 + \mu\|x\|_1 \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $\mu > 0$ are given.

It is equivalent to the following optimization problem

$$\begin{cases} \min \frac{1}{2}\|A(x^+ - x^-) - b\|_2^2 + \mu\mathbf{1}^T(x^+ + x^-) \\ \text{s.t.} x^+ \geq 0,\ x^- \geq 0, \end{cases} \tag{2}$$

We can rewrite it into a quadratic optimization problem:

$$\begin{cases} \min & \frac{1}{2}\begin{bmatrix} x^+ \\ x^- \end{bmatrix}^T \begin{bmatrix} A^TA & -A^TA \\ -A^TA & A^TA \end{bmatrix} \begin{bmatrix} x^+ \\ x^- \end{bmatrix} + \begin{bmatrix} \mu\mathbf{1} - A^Tb \\ A^Tb + \mu\mathbf{1} \end{bmatrix}^T \begin{bmatrix} x^+ \\ x^- \end{bmatrix} + \frac{1}{2}b^Tb \\ \text{s.t.} & x^+ \geq 0,\ x^- \geq 0, \end{cases} \tag{3}$$

The problem 3 can be solved by *mosek* and *gurobi*. We randomly generate A and b again, so A and b in this section are not the same in Section 1. *mosek* takes $6.12s$ and the optimal value is 0.0759. The error of *mosek* to *cvx mosek* is 3.33×10^{-4}. *gurobi* takes $29.85s$ and the optimal value is 0.0735. The error of *gurobi* to *cvx mosek* is 1.98×10^{-6}. We plot the exact solution and solutions from *mosek* and *gurobi*:

From figure 2, both *mosek* and *gurobi* give exactly the exact solution.

3 Problem 3(a)

The problem 3 is a quadratic program with box constraints. We consider to use continuation method. Let $\mu_i = \alpha^{N-i}\mu$, $i = 1, 2 \ldots N$, where $\alpha > 1$ and N are parameters for continuation method. Then, $\mu_N = \mu$. Then the problem 3 with μ_i is equivalent to:

$$\begin{cases} \min & \frac{1}{2}(x^+ - x^-)^TA^TA(x^+ - x^-) + (\mu_i\mathbf{1} - A^Tb)^Tx^+ + (\mu_i\mathbf{1} + A^Tb)^Tx^- + \frac{1}{2}b^Tb \\ \text{s.t.} & x^+ \geq 0, \quad x^- \geq 0 \end{cases} \tag{4}$$

We denote $f_i(x^+, x^-) = \frac{1}{2}(x^+ - x^-)^TA^TA(x^+ - x^-) + (\mu\mathbf{1} - A^Tb)^Tx^+ + (\mu_i\mathbf{1} + A^Tb)^Tx^- + \frac{1}{2}b^Tb$. Then, $\nabla_{x^+}f_i = A^TA(x^+ - x^-) + \mu_i\mathbf{1} - A^Tb$, $\nabla_{x^-}f_i = A^TA(x^- - x^+) + \mu_i\mathbf{1} + A^Tb$.

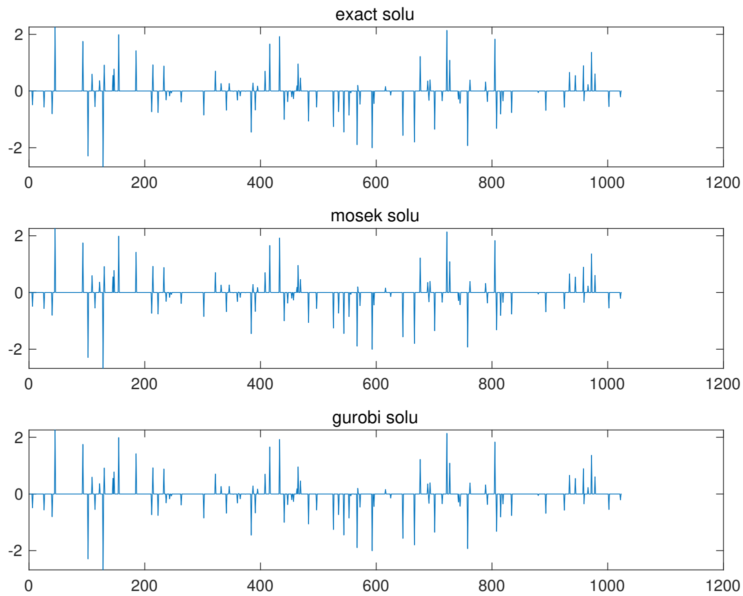

Figure 2 *Solutions*

The projection on C is given by:

$$P_C(x^+)_j = \max\{x_j^+, 0\}, \quad P_C(x^-)_j = \max\{x_j^-, 0\}, \quad j = 1, 2 \ldots n$$

The initial guess of x^+, x^- is given by

$$(x^+)_j = \max\{(x_0)_j, 0\}, \quad (x^-)_j = \max\{-(x_0)_j, 0\}, \quad j = 1, 2 \ldots n$$

The solution x is given by $x = x^+ - x^-$. Then we get the following projection gradient method with continuation method:

Algorithm 1 Projection gradient method with continuation method

Input: initial guess x_0, step size s, continuation parameter α, N, K.

1: Let $(x^+)_j = \max\{(x_0)_j, 0\}, \quad (x^-)_j = \max\{-(x_0)_j, 0\}, \quad j = 1, 2 \ldots n$.
2: **for** $i = 1 : N$ **do**
3:     **for** $k = 1 : K$ **do**
4:         $d^+ = A^T A(x^+ - x^-) + \mu_i \mathbf{1} - A^T b$, $d^- = A^T A(x^- - x^+) + \mu_i \mathbf{1} + A^T b$
5:         $x^+ = P_C(x^+ - sd^+)$, $x^- = P_C(x^- - sd^-)$
6:     **end for**
7: **end for**
8: $x = x^+ - x^-$

We take $s = 4 \times 10^{-4}$, $\alpha = 10$, $N = 6$, $K = 180$. The program is named *l1_projgrad.m*. Then, we compare our solution with the solution from *cvx mosek*. Our algorithm 1 takes $7.77s$, and the optimal value is 0.0758. *cvx mosek* takes $8.07s$, and the optimal value is 0.0758. The error of x between algorithm 1 to *cvx mosek* is 2.8×10^{-6} and the error of the optimal value is -1.8992×10^{-7}. We plot the exact solution and solutions from algorithm 1 and *cvx mosek*:

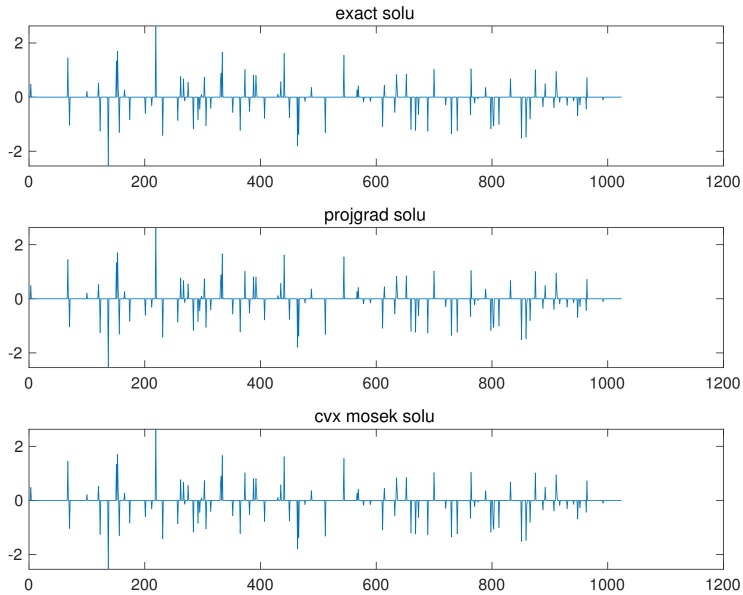

Figure 3 *Solutions*

4 Problem 3(b)

Let us denote $F_\mu(x) = \frac{1}{2}\|Ax - b\|_2^2 + \mu\|x\|_1$. Then, the primal problem can be written as:

$$\min_{x \in \mathbb{R}^n} F_\mu(x) \tag{5}$$

We know that $g_\mu(x) = A^T(Ax - b) + \mu \text{sign}(x)$ is a subgradient of $F_\mu(x)$, where

$$\text{sign}(x)_j = \begin{cases} 1, & x_j > 0 \\ 0, & x_j = 0 \quad j = 1, 2 \ldots n \\ -1, & x_j < 0 \end{cases}$$

We apply continuation method in our implementation as well. Let $\mu_i = \alpha^{N-i}\mu$, $i = 1, 2 \ldots N$, where $\alpha > 1$ and N are parameters for continuation method. Then, $\mu_N = \mu$. Then, we get the following subgradient method with continuation method:

Algorithm 2 Subgradient method with continuation method

Input: initial guess x_0, step size s, continuation parameter α, N, max iteration number for each stage K.
1: Let $x = x_0$.
2: **for** $i = 1 : N$ **do**
3:     **for** $k = 1 : K$ **do**
4:         $x = x - s_i \left(A^T Ax - A^T b + \mu_i \text{sign}(x)\right)$
5:     **end for**
6: **end for**

We take $s = 2.8 \times 10^{-4}, \alpha = 10, N = 6, K = 300$. The program is named *l1_subgrad.m*. Then, we compare our solution with the solution from *cvx mosek*. Our algorithm 1 takes $6.47s$, and the optimal value is 0.0728. *cvx mosek* takes $10.56s$, and the optimal value is 0.0728. The error of x between algorithm 2 to *cvx mosek* is 2.34×10^{-6} and the error of the optimal value is 1.7117×10^{-7}. We plot the exact solution and solutions from algorithm 1 and *cvx mosek*:

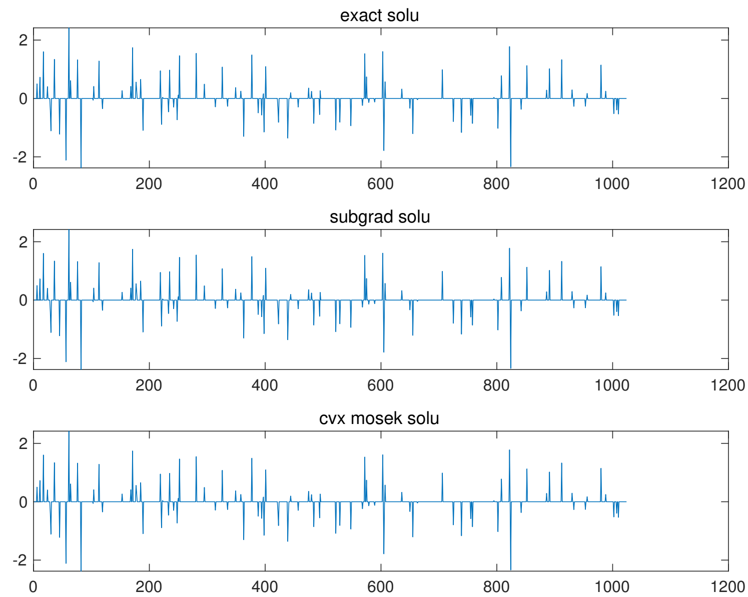

Figure 4 *Solutions*

The whole test program is named *Test_hw05_01.m*.

5 Numerical result

This time we run all algorithms mentioned before with same A and b. The numerical result is given in the following table:

Table 1 Random seed is 4. The cpu time of cvx mosek is 1.18

Method	cpu time	objval to cvx mosek	error to cvx mosek
cvx gurobi	1.66	-1.19×10^{-7}	2.09×10^{-6}
mosek	3.41	2.14×10^{-5}	2.42×10^{-4}
gurobi	8.81	-1.20×10^{-7}	2.11×10^{-6}
projection gradient	1.13	-1.19×10^{-7}	1.99×10^{-6}
subgradient	0.94	-2.73×10^{-9}	1.49×10^{-6}