SI151A
Convex Optimization and its Applications in Information Science, Fall 2024
Homework 3

Due on Dec. 16, 2024, 11:59 AM

Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- You are required to write down all the major steps towards making your conclusions; otherwise you may obtain limited points ($\leq 20\%$) of the problem.
- Write your homework in English; otherwise you will get no points of this homework.
- Do your homework by yourself. Any form of plagiarism will lead to 0 point of this homework. If more than one plagiarisms during the semester are identified, we will prosecute all violations to the fullest extent of the university regulations, including but not limited to failing this course, academic probation, or expulsion from the university.
- If you have any doubts regarding the grading, you need to contact the instructor or the TAs within two days since the grade is announced.

I. CVX Programming

1. Consider the following Tschebyshev approximation problem:

$$\min_{\mathbf{x}\in\mathbb{R}^p} \|\mathbf{A}\mathbf{x} - b\|_\infty,$$

where $\mathbf{A} \in \mathbb{R}^{n\times p}$ and $\mathbf{b} \in \mathbb{R}^n$, and $\|\mathbf{u}\|_\infty := max\{|\mathbf{u}_i| \mid 1 \leq i \leq p\}$.

   (a) Reformulate it in LP.. (10 points)

   (b) Use CVX to solve the original problem and the LP form and report your results respectively. The initialisation part is given below.(10 points)

```
n = 10; p = 20;
A = randn(n, p);
x_org = randn(p, 1);
b = A*x_org+1e-2*randn(n, 1);
```

Solution: LP:

$$\begin{aligned} \min_{t,\mathbf{x}} \quad & t, \\ s.t. \quad & -(\mathbf{A}\mathbf{x} - \mathbf{b})_i \leq t, \forall i = 1, \ldots, n, \\ & (\mathbf{A}\mathbf{x} - \mathbf{b})_i \leq t, \forall i = 1, \ldots, n. \end{aligned}$$

Code for the original problem:

```
cvx_begin
    variable x(p)
    minimize(norm(A*x-b, Inf))
cvx_end
```

Code for LP:

```
cvx_begin
    variable x(p) t
    minimize(t)
    subject to
        abs(A*x-b)<=t
cvx_end
```

2. Consider the SOCP problem in HW2:

$$\min_{x\in\mathbb{R}^p} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 + \lambda\|\mathbf{D}\mathbf{x}\|_1,$$

where $\mathbf{A} \in \mathbb{R}^{n\times p}$, $\mathbf{D} \in \mathbb{R}^{n\times p}$, and $\mathbf{b} \in \mathbb{R}^n$

   (a) Use CVX to solve the original problem and SOCP form and report your results respectively. The initialisation part is given below.(10 points)

   (b) Compare the results, and answer 1), how many iterations do both problems cost respectively? 2), What if we change the solver?(10 points)

```
n = 10; p = 20;
A = randn(n, p);
x_org = randn(p, 1);
b = A*x_org+1e-2*randn(n, 1);
D = randn(n, p);
```

Solution: Code for the original problem:

```
%you need to tune the solver in order to compare  the results
cvx_solver sdpt3

cvx_begin
    variables x(p)
    mnimize (norm(A*x-b, 2)+lambda * norm(D*x, 1))
cvx_end
```

Code for the SOCP:

```
%you need to tune the solver in order to compare  the results
cvx_solver sdpt3

cvx_begin
    variables x(p) s(n) t
    minimize (t+lambda*sum(s))
    subject to
        norm(A*x-b, 2)<=t
        -s<=D*x<=s
cvx_end
```

Any reasonable results based on your code will get full points.

II. Sparse Optimization

Consider a *linear system of equations* $\mathbf{x} = D\boldsymbol{\alpha}$, where D is an *underdetermined* $m \times p$ matrix ($m < p$) and $\mathbf{x} \in \mathbb{R}^m$, $\boldsymbol{\alpha} \in \mathbb{R}^p$. The matrix D (typically assumed to be full-rank) is referred to as the *dictionary*, and $\mathbf{x}$ is a signal of interest. The core sparse representation problem is defined as the quest for the sparsest possible representation $\boldsymbol{\alpha}$ satisfying $\mathbf{x} = D\boldsymbol{\alpha}$. Due to the underdetermined nature of D, this linear system admits in general infinitely many possible solutions, and among these, we seek the one with the fewest non-zeros. This is the most popular problem in *compress sensing*.

1. Given $\mathbf{x}$ and $\mathbf{D}$, we want to find the α with the fewest non-zeros. Derive the problem. (5 points) Relax the problem so that the **objective function** is convex. (5 points) (*hint*: You can use the L_0 norm to form the problem and use some other norm to approximate the L_0 norm.)
   Solution:
   Original problem:
$$\min_{\boldsymbol{\alpha}} \quad \|\boldsymbol{\alpha}\|_0, \qquad s.t. \quad \mathbf{x} = D\boldsymbol{\alpha},$$
   where $\|\cdot\|_0$ is the l_0 norm, equal to the numbers of the non-zero entries of the vector.
   Relaxation:
$$\min_{\boldsymbol{\alpha}} \quad \|\boldsymbol{\alpha}\|_1, \qquad s.t. \quad \mathbf{x} = D\boldsymbol{\alpha}.$$

2. Derive the **Lagrangian** $L(\mathbf{x}, \lambda, \mathbf{v})$ of the relaxed problem. (5 points) Consider the problem $\min_{\mathbf{x}} L(\mathbf{x}, \lambda, \mathbf{v})$, reformulate it in the ADMM form. (5 points)

Solution:
Lagrangian:
$$L(\boldsymbol{\alpha}, \boldsymbol{\lambda}) = \|\boldsymbol{\alpha}\|_1 + \boldsymbol{\lambda}^T(\mathbf{x} - D\boldsymbol{\alpha}),$$
where $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the Lagrange multiplier.
ADMM form:
$$\min_{\boldsymbol{\alpha},\mathbf{z}} \quad \|\boldsymbol{\alpha}\|_1 + \boldsymbol{\lambda}^T(\mathbf{x} - D\mathbf{z}), \qquad s.t. \quad \mathbf{z} = \boldsymbol{\alpha},$$

III. Low-Rank Optimization

1. Consider a rating matrix $\mathbf{R} \in \mathbb{R}^{m\times n}$ with R_{ij} representing the rating user i gives to movie j. But some R_{ij} are unknown since no one watches all the movies. Thus, the $\mathbf{R}$ may look like blow
$$\mathbf{R} = \begin{bmatrix} 2 & 3 & ? & ? & 5 & ? \\ 1 & ? & 4 & ? & 3 & ? \\ ? & ? & 3 & 2 & ? & 5 \\ 4 & ? & 3 & ? & 2 & 4 \end{bmatrix}$$
   According to the above background, we would like to predict how users will like unwatched movies. Unfortunately, the rating matrix is very big, $480,189$ (number of users) times $17,770$ (number of movies) in the Netflix case. But there are much fewer types of people and movies than there are people and movies. So it is reasonable to assume that for each user i, there is a k-dimensional vector $\mathbf{p}_i$ explaining the user's movie taste, and for each movie j, there is also a k-dimensional vector $\mathbf{q}_j$

explaining the movie's appeal. The inner product between these two vectors, $\mathbf{p}_i^\top \mathbf{q}_j$, is the rating user i gives to movie j, i.e.,

$$R_{ij} = \mathbf{p}_i^\top \mathbf{q}_j.$$

Or equivalently in matrix form, $\mathbf{R}$ is factorized as

$$\mathbf{R} = \mathbf{P}^\top \mathbf{Q},$$

where $\mathbf{P} \in \mathbb{R}^{k \times m}$, $\mathbf{Q} \in \mathbb{R}^{k \times n}$, $k \ll \min(m, n)$. It is the same as assuming the matrix $\mathbf{R}$ is of low rank. However, the true rank k is unknown. A natural approach is to find the minimum rank solution $\mathbf{X}$.

(a) Given the rating matrix $\mathbf{R}$ with known entries R_{ij}, where $(i, j) \in \Omega$ and Ω is the set of observed entries, derive the optimization problem. (Not that $\mathbf{P}$ and $\mathbf{Q}$ are also unknown, so they should not be used in the formulation.)(10 points)

(b) In practice, instead of requiring strict equality for the observed entries, we may allow some error ϵ between the entry of solution X_{ij} and the corresponding entry of the observation R_{ij}. Modify the problem in 1 to satisfy the above requirements.(10 points)

Solution:
Original problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{rank}(\mathbf{X}), \\ s.t. \quad & X_{ij} = R_{ij}, \forall (i, j) \in \Omega. \end{aligned}$$

Modified problem (any reasonable answers will obtain full points):

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{rank}(\mathbf{X}), \\ s.t. \quad & (X_{ij} - R_{ij})^2 \le \epsilon, \forall (i, j) \in \Omega. \end{aligned}$$

2. Consider the low-rank matrix recovery problem

$$\begin{aligned} \min_{\mathbf{L}} \quad & \sum_{i=1}^{m} (y_i - \mathbf{x}_i^T \mathbf{L} \mathbf{x}_i)^2, \\ s.t. \quad & \text{rank}(\mathbf{L}) \le r, \end{aligned}$$

where $\mathbf{L} \in \mathbb{R}^{p \times p}$ is a low-rank matrix with $\text{rank}(\mathbf{L}) \le r \ll p$, $y_i \in \mathbb{R}$, and $\mathbf{x} \in \mathbb{R}^p$. Since the low-rank property is hard to address in practice, we would like to introduce some assumptions on $\mathbf{L}$. Assume that $\mathbf{L} \in \mathbb{S}_+^p$, the factorization method can be used. (Fracterization Method: in the previous problem, the low–rank matrix $\mathbf{R}$ can be factorized as $\mathbf{R} = \mathbf{P}^T \mathbf{Q}$, where $\mathbf{P} \in \mathbb{R}^{k \times m}$, $\mathbf{Q} \in \mathbb{R}^{k \times n}$, $k \ll \min(m, n)$.)

(a) Derive the optimization problem with factorization. (10 points)

(b) Derive the gradient of the objective function in (a). (5 points) Is the objective function convex, concave, or neither? (5 points)

Solution:
Factorization:

$$\min_{\mathbf{W}} \quad \sum_{i=1}^{m} (y_i - \mathbf{x}_i^T \mathbf{W} \mathbf{W}^T \mathbf{x}_i),$$

where $\mathbf{W} \in \mathbb{R}^{p \times r}$ is the low-rank factor.

Gradient:

$$g = 4 \sum_{i=1}^{m} \left(\|\mathbf{x}_i^T \mathbf{W}\|_2^2 - y_i \right) \mathbf{x}_i \mathbf{x}_i^T \mathbf{W}.$$

This is a non-convex problem.

References

[1] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp, "Living on the edge: Phase transitions in convex programs with random data," *Inf. Inference*, vol. 3, pp. 224-294, Jun. 2014.