Po-Cheng Pan (UT EID: pp22828)
Convex Optimization

Problem Set One

Matlab, Python and Computational Assignments

1.

• Algorithm 1 : Least square

. **(a) Did CVX succeed?**

. **(b) If so, how long did it take to solve each instance?**

. **(c) Report the Regression error of the solution computed: $\|X\beta^* - y\|_2$ and also the Testing error: $\|X_{test}\beta - y_{test}\|_2$.**

Ans:

(a) In ps1A, there are three problems. $X_1 \in \mathbb{R}^{50\times 500}$, $X_2 \in \mathbb{R}^{500\times 5000}$ and $X_3 \in \mathbb{R}^{5000\times 50000}$. CVX succeed when performing Least Square Method for first and second problems. However, when it comes to third problem, CVX is stopped because of out of memory. Maybe it succeed in a better processor. So it isn't sure whether CVX succeed for third problem or not.

(b) In first problem, it took less than one second to solve each instance. And in the second problem, it took almost six minutes to solve each instance. However, for third problem, CVX failed. So we didn't how long it took to solve each instance.

(c) For both first and second problems, the least square method find β^* such that the regression errors are sufficient small. However, we can find that the testing errors are large either first and second problem. So, β^* we found is still very different from the true β.

Least Square	Running Time(sec)	$\|X\beta^* - y\|_2$	$\|X_{test}\beta - y_{test}\|_2$
Problem One	0.64	$6.106\text{x}10^{-11}$	$1.98\text{x}10^{1}$
Problem Two	331.52	$3.602\text{x}10^{-9}$	$2.07\text{x}10^{1}$
Problem Three	-	-	-

• Algorithm 2 : Sparse Recovery via an optimization-based algorithm called LASSO.

- **(a) Did CVX succeed?**
- **(b) If so, how long did it take to solve each instance?**
- **(c) Report the Regression error of the solution computed: $\|X\beta^* - y\|_2$ and also the Testing error: $\|X_{test}\beta - y_{test}\|_2$.**
- **(d) What is the support of β? That is, what are the non-zero coefficients of β.**

Ans:

(In this question, I set **λ** to be 0.1.)

(a) In ps1A, there are three problems. $X_1 \in \mathbb{R}^{50\times 500}$, $X_2 \in \mathbb{R}^{500\times 5000}$ and $X_3 \in \mathbb{R}^{5000\times 50000}$. Like least square method, CVX succeed for first and second problems. However, it failed on third problem because of out of memory. Maybe it succeed in a better processor. It isn't sure whether CVX succeed for third problem or not.

(b) In first problem, it took almost one second to solve each instance. And in the second problem, it took several minutes to solve each instance. However, for third problem, CVX failed. So we didn't how long it took to solve each instance.

(c) For both first and second problems, the least square method find β^* such that the regression errors are sufficient small. However, unlike Least Square Method, LASSO finds β^* closer to the true β since the testing errors are very small.

LASSO	Running Time(sec)	$\|X\beta^* - y\|_2$	$\|X_{test}\beta - y_{test}\|_2$
Problem One	1.14	$1.019\text{x}10^{-9}$	$1.27\text{x}10^{-1}$
Problem Two	104.42	$2.468\text{x}10^{-8}$	$7.73\text{x}10^{-2}$
Problem Three	-	-	-

(d) According to the SparseRegressionData.m, we know that **β** is composed of five 1's and other elements are 0's. Hence, the non-zero coefficients of **β** is 1. On the other hand, if we see the composition of **β*** we find by LASSO, we can find that there are 5 elements approach to 1, which have same index as the five 1's of **β** do. And the other elements of **β*** are approach to zero. So, the support of **β*** is exactly the same as the support of the true **β**

2. (OMP – Orthogonal Matching Pursuit)

(a) What is the sparsity pattern found?

(b) How long does the solution take?

(c) What are the regression and testing errors?

Ans:

(a) If we take a look at each element of β^* for each problem, we can find that there are only five elements are not zero and approach to one. Hence, the sparsity pattern found by OMP is exactly the same as the original sparsity pattern for all three problems.

(b) This solution runs much faster than the above two algorithms. Even the third problem took only within a second. The detailed running time are showed on the below table.

(c) The regression errors aren't as small as the above two algorithms. But the testing errors are much smaller than other two algorithms. The detailed errors are showed on the below table.

OMP	Running Time(sec)	$\|X\beta^* - y\|_2$	$\|X_{test}\beta - y_{test}\|_2$
Problem One	0.005	6.270×10^{-2}	3.97×10^{-2}
Problem Two	0.010	2.167×10^{-1}	1.19×10^{-2}
Problem Three	0.820	6.983×10^{-1}	2.60×10^{-3}

4.

We want to run the gradient descent algorithm which iteratively computes

$$\beta^{(n+1)} = \beta^{(n)} - \gamma \nabla f(\beta^{(n)})$$

where γ is a constant step size. The initial $\beta^{(0)}$ is the all-one vector.

For each matrix, find the range of γ that the solution converges to zero and the range of γ that the algorithm diverges, and explain why. Take example values of γ to illustrate the two behaviors, convergence to zero and divergence. Plot $f(\beta^{(n)})$ over n for the two of your values.

Ans:

Because X is symmetric and positive definite matrix, there always exists an eigendecomposition of $X = U\Lambda U^T$ where $U \in R^{m \times m}$ is a unitary matrix and $\Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_n)$ is a diagonal matrix with positive eigenvalues $\lambda_1 \geq \ldots \geq \lambda_m > 0$.

We want to run the gradient descent algorithm which iteratively computes

$$\beta^{(n+1)} = \beta^{(n)} - \gamma \nabla f(\beta^{(n)}) \quad (\nabla f(\beta^{(n)} = X\beta^{(n)})$$

$$=> \quad \beta^{(n+1)} = \beta^{(n)} - \gamma X \beta(n) \qquad (X = U\Lambda U^T)$$

$$=> \quad \beta^{(n+1)} = \beta^{(n)} - \gamma U \Lambda U^T \beta^{(n)}$$

$$=> \quad \beta^{(n+1)} = U(I - \gamma\Lambda)U^T \beta^{(n)}$$

Now we can let $\beta' = U^T\beta$

$$=> \quad U^T\beta^{(n+1)} = U^TU(I - \gamma\Lambda)U^T \beta^{(n)} \text{ (U is unitary => } U^TU = I)$$

$$=> \quad \beta'^{(n+1)} = (I - \gamma\Lambda)\, \beta'^{(n)} = (I - \gamma\Lambda)^2\, \beta'^{(n-1)} = \ldots = (I - \gamma\Lambda)^{(n+1)}\, \beta'^{(0)}$$

Then if we consider for each component of β' as β'_i and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$

Hence, $\beta'^{(n)}_i = (1 - \gamma\lambda_i)^n\, \beta'^{(0)}_i$

If we want to make sure the solution converges to zero, $|1 - \gamma\lambda_i|$ should < 1 for all $i = 1, 2, \ldots, n$

$=> 1 - \gamma\lambda_i < 1$ or $\gamma\lambda_i - 1 < 1 => 0 < \gamma\lambda_i$ or $\gamma\lambda_i < 2$

(Because X is symmetric and positive definite matrix, $0<\lambda_i$)

$=> 0<\gamma<(2/\lambda_i)$

For (a), all eigenvalues are one $\lambda_i=1$, that is . So it converges when $0<\gamma<2$ and diverges when $2<\gamma$.

For (b), a half of the eigenvalues are one and the other half of them are very small. $0<\gamma<(2/\lambda_i)$ must satisfy for all λ_i. Hence, it converges when $0<\gamma<(2/\max(\lambda_i)) = 0<\gamma<2$ and diverges when $2<\gamma$.

For (c), all other than a few very large eigenvalues. We can know that the large eigenvalues are 100 from the data given by the instructor. Like (b), it converges when $0<\gamma<(2/\max(\lambda_i)) = 0<\gamma<2/100 = 0<\gamma<0.02$ and diverges when $0.02<\gamma$.

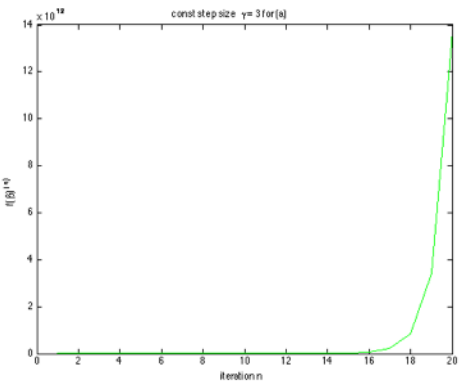
The below table are convergence and divergence conditions for (a), (b) and (c) as stated above.

	convergence	divergence
(a)	$0<\gamma<2$	$2<\gamma$
(b)	$0<\gamma<2$	$2<\gamma$
(c)	$0<\gamma<0.02$	$0.02<\gamma$

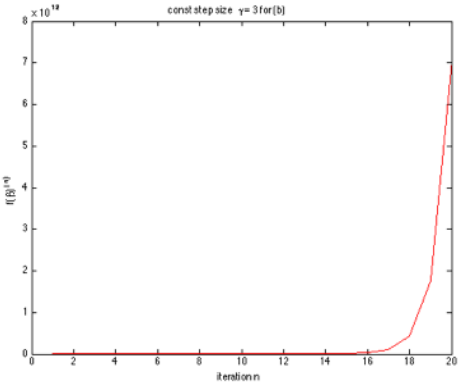
Then using Matlab to plot $f(\beta^{(n)})$ over n for two different γ:

First, $\gamma = 3$, n =20

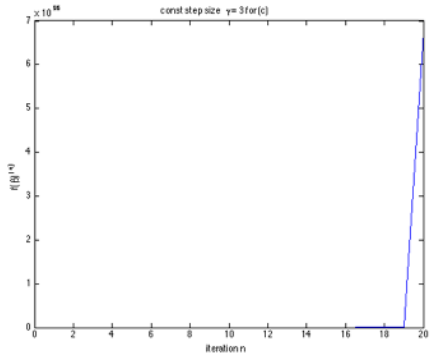
(a)



(b)



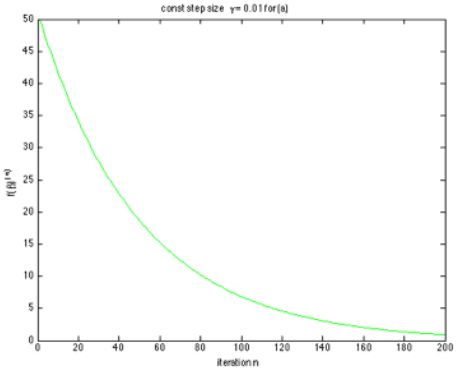
(c)



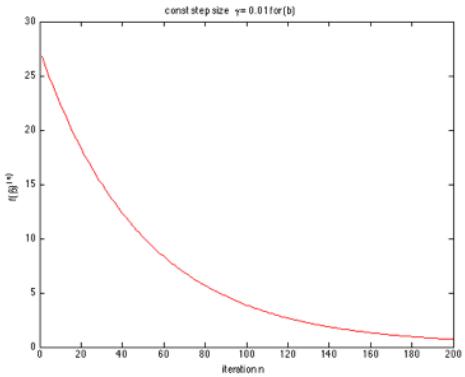
From the above three figures, it's easy to find that all three matrices diverge. And $\gamma = 3$ satisfies the divergence condition of all three matrices.

Then, $\gamma = 0.1$, n =20

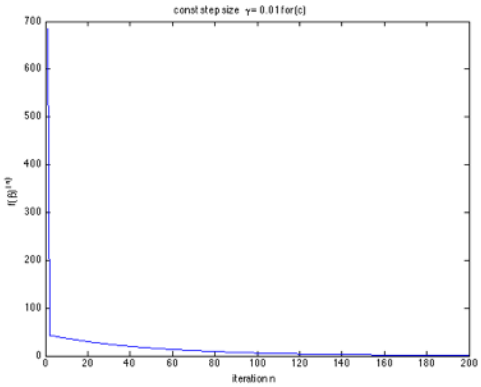
(a)



(b)



(c)

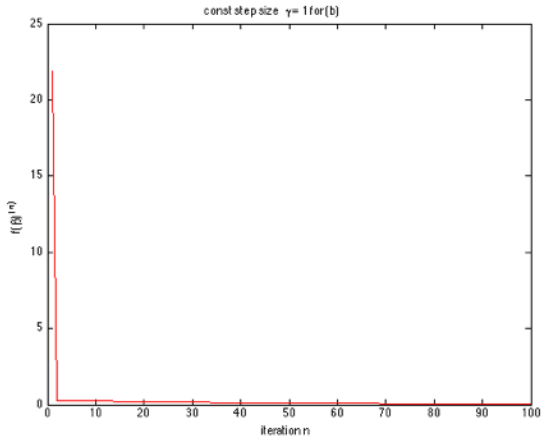


From the above three figures, it's easy to find that all three matrices converge. And $\gamma = 0.01$ satisfies the convergence condition of all three matrices.

5. Take $\gamma = 1$, and plot $f(\beta^{(n)})$ over n for the second matrix (b) of the above three. Explain the convergence behavior of the solution based on the plot.

Ans:

Take $\gamma = 1$, and plot $f(\beta^{(n)})$ over n for the second matrix (b) as below figure:



$f(\beta) = 1/2*\beta^T X\beta = 1/2*\beta^T U\Lambda U^T\beta$ (If let $\beta' = U^T\beta$)

$\Rightarrow = 1/2*\beta'\Lambda\beta' = (1/2)* \sum_{i=1}^{m} \lambda_i \beta_i^2$

As a result, we know that $f(\beta)$ depends on β'.

Besides, if we set $\gamma = 1$ for the matrix (b), because (b) a half of the eigenvalues are one and the other half of them are 0.01 (According to the data provided by the instructor.), we can know

$\beta'^{(n)}_i = (1 - 1*1)^n \beta'^{(0)}_i = 0$ for $i = 1, 2, \ldots, 50$

$\beta'^{(n)}_i = (1 - 1*0.01)^n \beta_i'^{(0)} = 0$ for $i = 51, 52, \ldots, 100$

Because of the first 50 components of β'_i equal to 0, there is a drop at the first iteration. On the other hand, because of the last 50 components of β'_i decreasing geometrically with a factor of 0.99, they contribute to the linear convergence to 0 after the first iteration.

EE381K : Convex Optimization Fall 2016 Po-Cheng Pan UT EID: pp22826

Written Problems:

1. Over-determined

$f(\beta) =$

① $\|X\beta - y\|_2^2 = \beta^T X^T X\beta - \beta^T X^T y - y^T X\beta + y^T y$

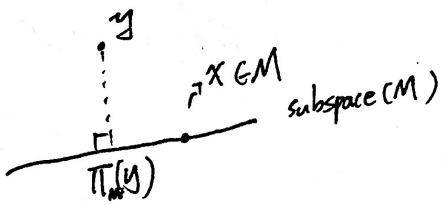
$\nabla^2 f(\beta) = 2X^T X$ $\because z^T \cdot \nabla^2 f(\beta) z = 2 \cdot z^T X^T X z = 2\|Xz\|^2 \geq 0$, $\forall z$

$\Rightarrow f(\beta)$ is a convex function of β #

If $\nabla f(\beta_{LS}) = 0 \Rightarrow 2X^T X\beta_{LS} - X^T y - (y^T X)^T = 0$

$\Rightarrow \beta_{LS} = (X^T X)^{-1} X^T y$ #

② Pic:



As the description of the question says,

$\langle y - \Pi_M(y), x \rangle = 0$ for all $x \in M$

We can say that $M = \text{Range}(X)$

$\Pi_M(y) = X\beta_{LS}$

Hence, $\langle y - \Pi_M(y), X \rangle = \langle y - X\beta_{LS}, X\beta \rangle = 0, \forall \beta$

Because $\forall \beta$, it all satisfies

$\Rightarrow \langle y - X\beta_{LS}, X \rangle = 0$

$(y - X\beta_{LS})^T \cdot X = 0$

$y^T X - \beta_{LS}^T X^T X = 0 \Rightarrow \beta_{LS} = (y^T X (X^T X)^{-1})^T$

$= (X^T X)^{-1} X^T y$ #

Under-determined

③ min : $\|\beta\|_2^2$

subject to $X\beta = y$

Consider any other solution $\beta_1 = \beta_0 + z$

$\|\beta_1\|^2 = \|\beta_0 + z\|^2 = \|\beta_0\|^2 + \|z\|^2$ ($\because \beta_0 \perp z \Rightarrow \beta_0 \cdot z = 0$)

$\because \|z\|^2 \geq 0 \Rightarrow \|\beta_1\|^2 \geq \|\beta_0\|^2$

So, β_0 is the minimum norm solution #

④ $y = X\beta_0$ & $\beta_0 \perp z$ for any $z \in \text{Null}(X)$

$\because$ Null(X) is the set of vectors perpendicular to the rows of X

$\therefore$ The set of vectors perpendicular to Null(X) must be in span of the rows of X

$\Rightarrow \beta_0$ is in the span of the rows of X $\Rightarrow \beta_0 = X^T z = \sum z_i X_i$ (X_i is the ith row of X, z for some vector)

$y = X\beta_0$

$= XX^T z$

$z = (XX^T)^{-1} y$

$\beta_0 = X^T z$

$= X^T (XX^T)^{-1} y$ #

12. $C = \{x \in \mathbb{R}^n : x^TAx + b^Tx + c \le 0\}$ where $A \in S^n$, $b \in \mathbb{R}^n$, $c \in \mathbb{R}^n$

(a) if $A \in S_+^n \Rightarrow x^TAx \ge 0$

Consider $\{x_1 + tv \mid t \in \mathbb{R}\}$ an arbitrary line

C's intersection with $\{x_1 + tv \mid t \in \mathbb{R}\}$

$\Rightarrow (x_1+tv)^TA(x_1+tv) + b^T(x_1+tv) + c \le 0$

$\Rightarrow v^TAv\,t^2 + (2x_1^TAv + b^Tv)t + (b^Tx_1 + c + x_1^TAx_1) \le 0$

It's convex because $A \in S_+^n$ s.t. $v^TAv \ge 0$

So C is convex because its intersection with an arbitrary line is convex.

Q.E.D. #

(b) we consider the intersection of $C \cap H = C_1$ with an arbitrary line $\{x_1 + tv \mid t \in \mathbb{R}\}$

Without loss of generality, we can assume $g^Tx_1 + h = 0$

Hence, the intersection defined by x_1 & v is

$$\{x_1 + tv \mid v^TAvt^2 + (2x_1^TAv + b^Tv)t + (b^Tx_1 + c + x_1^TAx_1) \le 0,\ g^Ttv = 0\}$$

If $g^Tv = 0$, the intersection is the singleton $\{x_1\}$

The set can be reduced to $\{x_1 + tv \mid v^TAvt^2 + (2x_1^TAv + b^Tv)t + (b^Tx_1 + c + x_1^TAx_1) \le 0\}$ which is convex if $v^TAv \ge 0$

Therefore C_1 is convex if $g^Tv = 0 \Rightarrow v^TAv \ge 0$

Consider $\lambda gg^T \in \mathbb{R}^{n \times n}$ $\lambda \in \mathbb{R}$

$$v^TAv = v^T(A + \lambda gg^T)v \ge 0 \quad (\because \lambda gg^Tv = 0)$$

In conclusion, C_1 is convex if there exists $\lambda \in \mathbb{R}$ s.t. $A + \lambda gg^T \in S_+^n$

Q.E.D. #

13. $S = \{(a,b) : a^Tx \le b\ \forall x \in C,\ a^Tx \ge b\ \forall x \in D\}$

It forms a set of homogenous linear inequalities in (a,b)

That means it's the intersection of many half spaces that pass through the origin

$\Rightarrow \because a^Tx \le b \Rightarrow a^Tx - b \le 0 \Rightarrow [x^T\ -1]\begin{bmatrix} a \\ b \end{bmatrix} \le 0\ \forall x \in C$ & $a^Tx - b \ge 0 \Rightarrow [x^T\ -1]\begin{bmatrix} a \\ b \end{bmatrix} \ge 0\ \forall x \in D$

$\Rightarrow$ Hence, it will form a convex cone.

So S is convex.

17. We can consider a hyperplane perpendicular to (V_2-V_1) & lying on $\frac{V_1+V_2}{2}$

Suppose this hyperplane as $(V_2-V_1)^T X = a$

$$\Rightarrow a = (V_2-V_1)^T \cdot \frac{V_1+V_2}{2} = \frac{1}{2}\left(\|V_2\|^2 - V_1^T V_2 + V_2^T V_1 + \|V_2\|^2 \right)$$
$$= \frac{1}{2}\left(\|V_2\|^2 - \|V_1\|^2 \right)$$

$$\{X : \|X-V_1\| \le \|X-V_2\|\} = \{X : \|X-V_1\|^2 \le \|X-V_2\|^2\}$$
$$= \{X : \|X\|^2 - 2V_1^T X + \|V_1\|^2 \le \|X\|^2 - 2V_2^T X + \|V_2\|^2\}$$
$$= \{X : 2V_2^T X - 2V_1^T X \le \|V_2\|^2 - \|V_1\|^2\}$$
$$= \{X : (V_2^T - V_1^T) X \le \frac{\|V_2\|^2 - \|V_1\|^2}{2}\}$$

$c = V_2^T - V_1^T$ $\quad d = \frac{\|V_2\|^2 - \|V_1\|^2}{2}$ #

18. $A \in \mathbb{R}^{k \times n}$ $\quad B \in \mathbb{R}^{k \times m}$

for every $X \in \mathbb{R}^m$, $AX = 0 \Rightarrow BX = 0 \Rightarrow Null(A) \subseteq Null(B)$

From II.(c), we know that $U \subseteq W \Leftrightarrow U^\perp \supseteq W^\perp$

So $Null(A) \subseteq Null(B) \Rightarrow Row\ Space(A) \supseteq Row\ Space(B)$
$\Rightarrow Range(A^T) \supseteq Range(B^T)$

For every $b \in Range(B^T)$, we can find a vector $c \in \mathbb{R}^n$
s.t. $A^T c = b$

Hence, For each column $b_i \in B^T$, we can find a $c_i \in \mathbb{R}^n$ s.t. $A^T c_i = b_i$

$$\Rightarrow A^T \begin{bmatrix} c_1^1 & c_2^1 & c_3^1 & \cdots & c_k^1 \\ | & | & | & & | \end{bmatrix} = B^T = \begin{bmatrix} b_1^1 & b_2^1 & \cdots & b_k^1 \\ | & | & & | \end{bmatrix}$$

$$A^T C_1 = B^T \Rightarrow (A^T C_1)^T = (B^T)^T \Rightarrow C_1^T A = B$$
$$\because C_1 \in \mathbb{R}^{n \times k} \quad \therefore C_1^T \in \mathbb{R}^{k \times n}$$

$\Rightarrow$ There exists a $k \times n$ real matrix C
s.t. $CA = B$

Q.E.D.

#