THE UNIVERSITY OF TEXAS
AT AUSTIN

EE381V LARGE SCALE OPTIMIZATION

Problem Set 1

Edited by LaTeX

Department of Computer Science

STUDENT
Jimmy Lin
xl5224

COURSE COORDINATOR
Sujay Sanghavi

UNIQUE NUMBER
17350

RELEASE DATE
September 15, 2014

DUE DATE
September 23, 2014

TIME SPENT
13 hours

September 16, 2014

Table of Contents

List of Figures

Part I
Matlab and Computational Assignment

1 Gradient Descent on three matrices

Command to get executed:

```
>> gd_run_script()
```

1.1 $X1$, $b1$

- Range of γ that leads to convergence: $(0, 2)$
- Range of γ that leads to divergence: $(2, +\infty)$
- Explanation: if $\gamma = 2$, the program indicates that

$$\forall k,\ f(x^{k+1}) = f(x^k)$$

Since the above equation is constantly true (independent of the minima), we can conclude that gradient descent with $\gamma = 2$ goes to the opposite side of that quadratic curve. Intuitively, the program will diverge if we set larger ($\gamma > 2$) and converge if we set smaller ($\gamma < 2$).

- Two illustrative examples: $\gamma = 0.5$ and $\gamma = 3.0$

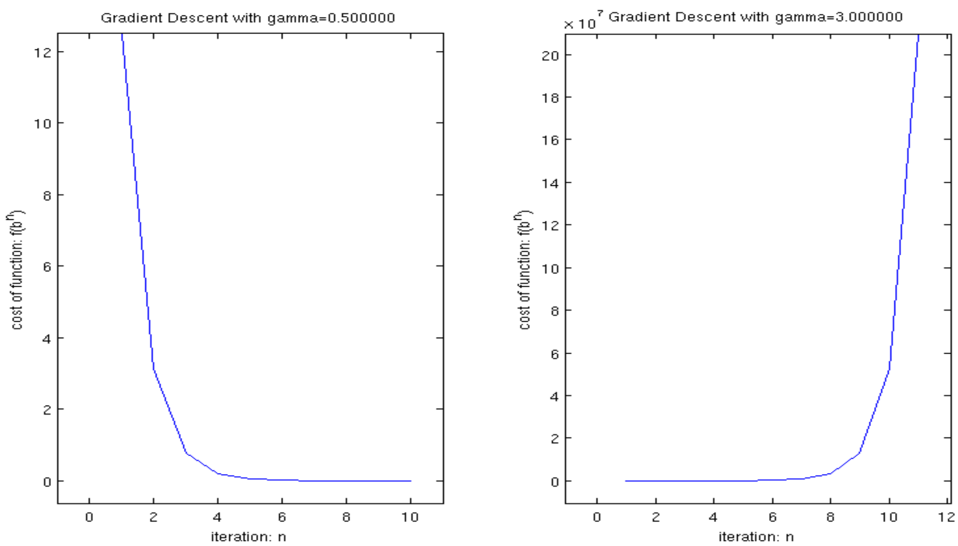

Figure 1: Illustration for gradient descent on $X1$, staring with $b1$ by $\gamma = 0.5$ and 3.0

1.2 $X2$, $b2$

- Range of γ that leads to convergence: $(0, 2)$
- Range of γ that leads to divergence: $(2, +\infty)$
- Explanation: if $\gamma = 2$, the program indicates that

$$\forall k,\ f(x^{k+1}) = f(x^k)$$

  Since the above equation is constantly true (independent of the minima), we can conclude that gradient descent with $\gamma = 2$ goes to the opposite side of that quadratic curve. Intuitively, the program will diverge if we set larger ($\gamma > 2$) and converge if we set smaller ($\gamma < 2$).

- Two illustrative examples: $\gamma = 1.5$ and $\gamma = 3.0$

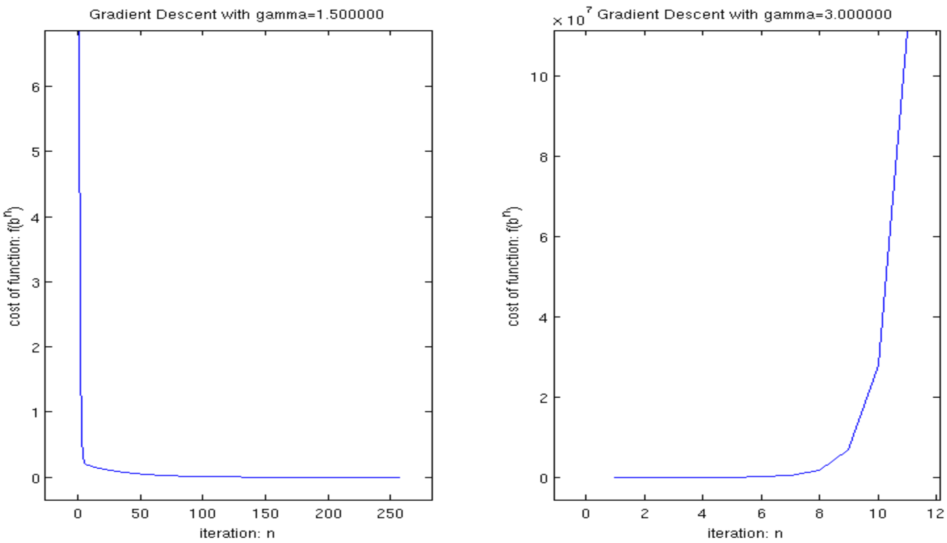

Figure 2: Illustration for gradient descent on $X2$, starting with $b2$ by $\gamma = 1.5$ and 3.0

1.3 $X3$, $b3$

- Range of γ that leads to convergence: $(0, 0.02)$
- Range of γ that leads to divergence: $(0.02, +\infty)$
- Explanation: if $\gamma = 0.02$, the program indicates that

$$\forall k,\ f(x^{k+1}) = f(x^k)$$

  Since the above equation is constantly true (independent of the minima), we can conclude that gradient descent with $\gamma = 0.02$ goes to the opposite side of that quadratic curve. Intuitively, the program will diverge if we set larger ($\gamma > 0.02$) and converge if we set smaller ($\gamma < 0.02$).

- Two illustrative examples: $\gamma = 0.005$ and $\gamma = 0.05$

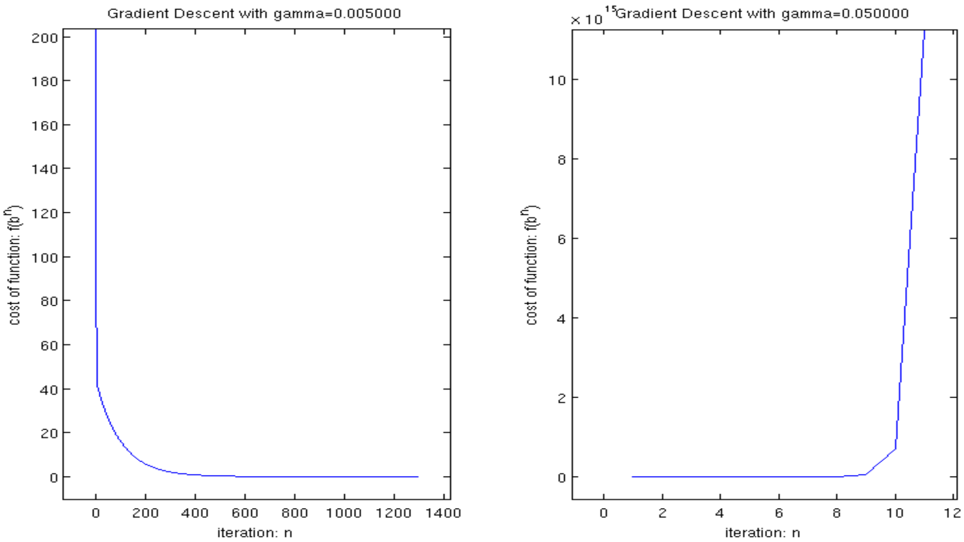

Figure 3: Illustration for gradient descent on $X3$ staring with $b3$ by $\gamma = 0.005$ and 0.05

2 $\gamma = 1$ for the second matrix

Command to get executed:

```
>> gamma = 1;
>> [b2_opt, iters, all_costs] = gd (X2, b2, gamma);
```

Plotting: figure for $\gamma = 1$

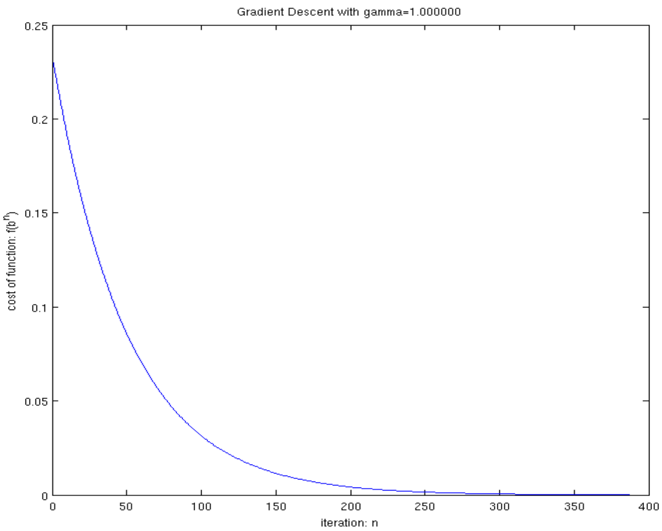

Figure 4: Plotting figure for gradient descent with $\gamma = 1$ on the second matrix

Explanation: Through the smooth plotted curve, we guess that the gradient descent method got linear convergence when $\gamma = 1$ on X_2. Hence, we trace convergence rate conv_rate $= f(x^k)/f(x^{k-1})$ as follows:

```
Iter: 2, Cost: 2.254428e-01, Conv_Rate: 0.980100
Iter: 3, Cost: 2.209565e-01, Conv_Rate: 0.980100
Iter: 4, Cost: 2.165594e-01, Conv_Rate: 0.980100
Iter: 5, Cost: 2.122499e-01, Conv_Rate: 0.980100
Iter: 6, Cost: 2.080261e-01, Conv_Rate: 0.980100
Iter: 7, Cost: 2.038864e-01, Conv_Rate: 0.980100
...
...
Iter: 381, Cost: 1.107934e-04, Conv_Rate: 0.980100
Iter: 382, Cost: 1.085886e-04, Conv_Rate: 0.980100
Iter: 383, Cost: 1.064277e-04, Conv_Rate: 0.980100
Iter: 384, Cost: 1.043098e-04, Conv_Rate: 0.980100
Iter: 385, Cost: 1.022340e-04, Conv_Rate: 0.980100
Iter: 386, Cost: 1.001996e-04, Conv_Rate: 0.980100
Iter: 387, Cost: 9.820558e-05, Conv_Rate: 0.980100
Converged to zeros!
```

In terms of above dumps and the fact that $f(x^*) = 0$, we can conclude that when $\gamma = 1$

$$f(x^{k+1}) - f(x^*) = 0.9801 \cdot \big(f(x^k) - f(x^*)\big)$$

which supports our previous guess that

Gradient Descent with $\gamma = 1$ on second matrix leads to **linear convergence**.

Part II
Written Problems

1 Othorognal Subspace

(a) Show that if U is a subspace, then so is $U^{\perp}$

Proof. Since U is a subspace, then we have U satisfying all three properties shown below:

- $\mathbf{0} \in U$
- $\forall \mathbf{u}_1, \mathbf{u}_2 \in U, \mathbf{u}_1 + \mathbf{u}_2 \in U$
- $\forall \mathbf{u} \in U, \alpha \in \mathbb{R}, \alpha\mathbf{u} \in U$

Now we show that $U^{\perp}$ is also a subspace by indicating $U^{\perp}$ satisfies all three properties as above U does.

- Since $\forall \mathbf{u} \in U, \langle \mathbf{0}, \mathbf{u} \rangle = 0$ and $\mathbf{0} \in V$ ($\mathbf{0} \in U \subseteq V$), then it turned out that $\mathbf{0} \in U^{\perp}$.
- Let $\mathbf{u}$ be arbitrary vector s.t. $\mathbf{u} \in U$, and $\mathbf{x}_1$, $\mathbf{x}_2$ to be distinct vector s.t. $\mathbf{x}_1 \in U^{\perp}$ and $\mathbf{x}_2 \in U^{\perp}$. By definition of $U^{\perp}$, we have $\langle \mathbf{x}_1, \mathbf{u} \rangle = 0$ and $\langle \mathbf{x}_2, \mathbf{u} \rangle = 0$. Then it is obvious that $\langle \mathbf{x}_1 + \mathbf{x}_2, \mathbf{u} \rangle = 0$. That is $\mathbf{x}_1 + \mathbf{x}_2 \in U^{\perp}$. Therefore, $\forall \mathbf{x}_1, \mathbf{x}_2 \in U^{\perp}, \mathbf{x}_1 + \mathbf{x}_2 \in U^{\perp}$ was proved.
- Let $\mathbf{x}$ be arbitrary vector s.t. $\mathbf{x} \in U^{\perp}$, $\mathbf{u}$ be arbitrary vector s.t. $\mathbf{u} \in U$ and arbitrary $\alpha \in \mathbb{R}$. By definition of $U^{\perp}$, we have $\langle \mathbf{x}_1, \mathbf{u} \rangle = 0$. Since inner product is linear operator, it is obvious that $\langle \alpha\mathbf{x}_1, \mathbf{u} \rangle = 0$. That is $\alpha\mathbf{x}_1 \in U^{\perp}$. Therefore, $\forall \mathbf{x} \in U^{\perp}, \alpha \in \mathbb{R}, \alpha\mathbf{x} \in U^{\perp}$ was proved.

Since $U^{\perp}$ contains $\mathbf{0}$, and is closed under addition and scalar multiplication, it turned out that $U^{\perp}$ is a subspace. Therefore, the statement that if U is a subspace, then so is $U^{\perp}$ was proved. □

(b) Show that $(U^{\perp})^{\perp} = U$

Proof. By contradiction. Assume that $(U^{\perp})^{\perp} \neq U$ and then show the contradiction. By definition of $U^{\perp}$, we have $U^{\perp} = \{\forall \mathbf{v} \in V, \langle \mathbf{v}, \mathbf{u} \rangle = 0, \forall \mathbf{u} \in U\}$ and $(U^{\perp})^{\perp} = \{\forall \mathbf{x} \in V, \langle \mathbf{x}, \mathbf{v} \rangle = 0, \forall \mathbf{v} \in U^{\perp}\}$. Since $(U^{\perp})^{\perp} \neq U$, then we can say that $\exists \mathbf{x} \notin U, \langle \mathbf{x}, \mathbf{v} \rangle = 0, \forall \mathbf{v} \in U^{\perp}$. That is to say, $\exists \mathbf{x} \in V$ *but* $\notin U, \langle \mathbf{x}, \mathbf{v} \rangle = 0, \forall \mathbf{v}\ s.t. \langle \mathbf{v}, \mathbf{u} \rangle = 0, \forall \mathbf{u} \in U$. However, such $\mathbf{x}$ does not exist. Hence, we reject initial assumption and conclude that $(U^{\perp})^{\perp} = U$. □

(c) Show that if $U, W \subseteq V$ are subspaces of V, then $U \subseteq W \Leftrightarrow U^{\perp} \supseteq W^{\perp}$

Proof of $U \subseteq W \Rightarrow U^{\perp} \supseteq W^{\perp}$. $U^{\perp} = \{\forall \mathbf{v} \in V, \langle \mathbf{v}, \mathbf{u} \rangle = 0, \forall \mathbf{u} \in U\}$ and $W^{\perp} = \{\forall \mathbf{v} \in V, \langle \mathbf{v}, \mathbf{w} \rangle = 0, \forall \mathbf{w} \in W\} = \{\forall \mathbf{v} \in V, \langle \mathbf{v}, \mathbf{w} \rangle = 0, \forall \mathbf{w} \in W$ and $\langle \mathbf{v}, \mathbf{u} \rangle = 0, \forall \mathbf{u} \in U\}$ (This is valid because $\forall U \subseteq W$ and then $\mathbf{u} \in W$). Now since membership of $W^{\perp}$ requires one more condition, then it is obvious that $\mathbf{v} \in W^{\perp} \Rightarrow \mathbf{v} \in U^{\perp}$, and $\mathbf{v} \in U^{\perp} \not\Rightarrow \mathbf{v} \in W^{\perp}$ hold for arbitrary $\mathbf{v}$. Therefore, we can conclude that $U^{\perp} \supseteq W^{\perp}$. □

Proof of $U^{\perp} \supseteq W^{\perp} \Rightarrow U \subseteq W$. By definition, we have $U^{\perp} = \{\mathbf{v} \in V | \langle \mathbf{v}, \mathbf{u} \rangle = 0, \forall \mathbf{u} \in U\}$ and $W^{\perp} = \{\mathbf{v} \in V | \langle \mathbf{v}, \mathbf{w} \rangle = 0, \forall \mathbf{w} \in W\}$. Since $U^{\perp} \supseteq W^{\perp}$, then we have $U^{\perp} \cap W^{\perp} = W^{\perp}$. Then $U^{\perp} \cap W^{\perp} = \{\mathbf{v} \in V | \langle \mathbf{v}, \mathbf{u} \rangle = 0, \forall \mathbf{u} \in U$ and $\langle \mathbf{v}, \mathbf{w} \rangle = 0, \forall \mathbf{w} \in W\} = \{\mathbf{v} \in V | \langle \mathbf{v}, \mathbf{x} \rangle = 0, \forall \mathbf{x} \in W \cup U\} = W^{\perp}$ Then we have $W \cup U = W$, which naturally derives $U \subseteq W$. □

(d) Show that $X^\perp$ makes sense, $X^\perp$ is a subspace and $(X^\perp)^\perp \supseteq X$

Proof of $X^\perp$ makes sense. $X^\perp$ also makes sense. This is because $\forall \mathbf{v} \in X^\perp, \langle \mathbf{v}, \rangle = 0$ □

Proof of $X^\perp$ is a subspace. We prove $X^\perp$ is a subspace by showing $X^\perp$ satisfies three properties as $U^\perp$ above does.

- Since $\forall \mathbf{x} \in X, \langle \mathbf{0}, \mathbf{x} \rangle = 0$ and $\mathbf{0} \in V$, then it turned out that $\mathbf{0} \in X^\perp$.
- Let $\mathbf{x}$ be arbitrary vector s.t. $\mathbf{x} \in U$, and $\mathbf{x}_1^\perp$, $\mathbf{x}_2^\perp$ to be distinct vector s.t. $\mathbf{x}_1^\perp \in X^\perp$ and $\mathbf{x}_2^\perp \in X^\perp$. By definition of $X^\perp$, we have $\langle \mathbf{x}_1^\perp, \mathbf{x} \rangle = 0$ and $\langle \mathbf{x}_2^\perp, \mathbf{x} \rangle = 0$. Then it is obvious that $\langle \mathbf{x}_1^\perp + \mathbf{x}_2^\perp, \mathbf{x} \rangle = 0$. That is $\mathbf{x}_1^\perp + \mathbf{x}_2^\perp \in X^\perp$. []Therefore, $\forall \mathbf{x}_1^\perp, \mathbf{x}_2^\perp \in X^\perp, \mathbf{x}_1^\perp + \mathbf{x}_2^\perp \in X^\perp$ was proved. (That is, membership of $X^\perp$ is closed under vector addition).
- Let $\mathbf{x}^\perp$ be arbitrary vector s.t. $\mathbf{x} \in X^\perp$, $\mathbf{x}$ be arbitrary vector s.t. $\mathbf{x} \in X$ and arbitrary $\alpha \in \mathbb{R}$. By definition of $X^\perp$, we have $\langle \mathbf{x}^\perp, \mathbf{x} \rangle = 0$. Since inner product is linear operator, it is obvious that $\langle \alpha \mathbf{x}^\perp, \mathbf{x} \rangle = 0$. That is $\alpha \mathbf{x}^\perp \in X^\perp$. Therefore, $\forall \mathbf{x}^\perp \in X^\perp, \alpha \in \mathbb{R}, \alpha \mathbf{x}^\perp \in X^\perp$ was proved. (That is, membership of $X^\perp$ is closed under scalar multiplication.)

Since we have shown that $X^\perp$ satisfies all three above properties, then we conclude that $X^\perp$ is a subspace. □

Proof of $(X^\perp)^\perp \supseteq X$. .. □

(e) Show that any $v \in V$ can be written uniquely as $v = u + u^\perp$

The key to show every vector $v \in V$ can be uniquely represented as $v = u + u^\perp$ lies in the fact that $U \cup U^\perp = V$. The following proof will emphasize to show this.

Proof. .. □

2 Boyd and Vandenberghe, Ex. 2.10

(a) Show that if $A \in \mathbb{S}_+^n$ then the set C is convex

Proof. Assume that $A \in \mathbb{S}_+^n$. Let $\mathbf{x}_1$, $\mathbf{x}_2$ to be arbitrary vector such that $\mathbf{x}_1 \in C$, $\mathbf{x}_2 \in C$ and $\mathbf{x}_1 \neq \mathbf{x}_2$. Then we show arbitrary linear combination $\mathbf{x}_* = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2,\ \lambda \in [0,1]$ also belongs to the set C. According to the definition of set C, we have

$$\mathbf{x}_1^T A\mathbf{x}_1 + b^T\mathbf{x}_1 + c \leq 0 \tag{1}$$
$$\mathbf{x}_2^T A\mathbf{x}_2 + b^T\mathbf{x}_2 + c \leq 0 \tag{2}$$

By positive semidefiniteness of A, we have

$$\mathbf{x}^T A\mathbf{x} \geq 0,\ \forall\mathbf{x} \tag{3}$$

That is

$$\mathbf{x}_1^T A\mathbf{x}_1 \geq 0 \tag{4}$$
$$\mathbf{x}_2^T A\mathbf{x}_2 \geq 0 \tag{5}$$
$$(\mathbf{x}_2 - \mathbf{x}_1)^T A(\mathbf{x}_2 - \mathbf{x}_1) \geq 0 \tag{6}$$

Besides,

$$\lambda \geq 0 \tag{7}$$
$$1 - \lambda \geq 0 \tag{8}$$
$$\lambda - 1 \leq 0 \tag{9}$$

Then we investigate the property of $\mathbf{x}_*^T A\mathbf{x}_* + b^T\mathbf{x}_* + c$.

$$\begin{aligned}
&\mathbf{x}_*^T A\mathbf{x}_* + b^T\mathbf{x}_* + c && (10)\\
&= (\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)^T A(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) + b^T(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) + c && (11)\\
&= \lambda^2\mathbf{x}_1^T A\mathbf{x}_1 + (1-\lambda)^2\mathbf{x}_2^T A\mathbf{x}_2 + \lambda(1-\lambda)(\mathbf{x}_1^T A\mathbf{x}_2 + \mathbf{x}_2^T A\mathbf{x}_1) + \lambda b^T\mathbf{x}_1 + (1-\lambda)b^T\mathbf{x}_2 + c && (12)\\
&= \lambda^2\mathbf{x}_1^T A\mathbf{x}_1 - \lambda\mathbf{x}_1^T A\mathbf{x}_1 + (1-\lambda)^2\mathbf{x}_2^T A\mathbf{x}_2 - (1-\lambda)\mathbf{x}_2^T A\mathbf{x}_2 \\
&+ \lambda(\mathbf{x}_1^T A\mathbf{x}_1 + b^T\mathbf{x}_1 + c) + (1-\lambda)(\mathbf{x}_2^T A\mathbf{x}_2 + b^T\mathbf{x}_2 + c) + \lambda(1-\lambda)(\mathbf{x}_1^T A\mathbf{x}_2 + \mathbf{x}_2^T A\mathbf{x}_1) && (13)\\
&\leq \lambda(\lambda-1)(\mathbf{x}_1^T A\mathbf{x}_1 + \mathbf{x}_2^T A\mathbf{x}_2) + \lambda(1-\lambda)(\mathbf{x}_1^T A\mathbf{x}_2 + \mathbf{x}_2^T A\mathbf{x}_1) && (14)\\
&= \lambda(\lambda-1)((\mathbf{x}_2 - \mathbf{x}_1)^T A(\mathbf{x}_2 - \mathbf{x}_1)) && (15)\\
&\leq 0 && (16)
\end{aligned}$$

Then, it is obvious that

$$\mathbf{x}_* \in C \tag{17}$$

Since $\mathbf{x}_*$ is arbitrary convex combination of $\mathbf{x}_1$ and $\mathbf{x}_2$, then

$$C \text{ is convex set.} \tag{18}$$

Hence, it is proved that

$$A \in \mathbb{S}_+^n \Rightarrow C \text{ is convex.} \tag{19}$$

□

(b) Show that C_1 is convex if there exists $\lambda \in \mathbb{R}$ such that $(A + \lambda gg^T) \in \mathbb{S}_+^n$

Proof. Assume that there exists $\lambda \in \mathbb{R}$ such that $(A + \lambda gg^T) \in \mathbb{S}_+^n$. Let λ^* be the λ that satisfies $(A + \lambda gg^T) \in \mathbb{S}_+^n$. Now we show that

$$\begin{aligned} C_1 &= C \cap \{\mathbf{x} \in \mathbb{R}^n : g^T\mathbf{x} + h = 0\} && (20)\\ &= \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^T A\mathbf{x} + b^T\mathbf{x} + c \le 0 \text{ and } g^T\mathbf{x} + h = 0\} && (21) \end{aligned}$$

is convex.
Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be arbitrary vector that $\mathbf{x}_1 \in C_1$, $\mathbf{x}_2 \in C_1$ and $\mathbf{x}_1 \neq \mathbf{x}_2$. Let $\mathbf{x}_*$ to be the arbitrary convex combination of $\mathbf{x}_1$ and $\mathbf{x}_2$ as $\mathbf{x}_* = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$, and next we show that $\mathbf{x}_* \in C_1$.
By the definition of C_1 (See. (21)), we have

$$\begin{aligned} \mathbf{x}_1^T A\mathbf{x}_1 + b^T\mathbf{x}_1 + c \le 0 \text{ and } g^T\mathbf{x}_1 + h = 0 && (22)\\ \mathbf{x}_2^T A\mathbf{x}_2 + b^T\mathbf{x}_2 + c \le 0 \text{ and } g^T\mathbf{x}_2 + h = 0 && (23) \end{aligned}$$

Then we have

$$\begin{aligned} g^T\mathbf{x}_2 + h - (g^T\mathbf{x}_1 + h) = 0 && (24)\\ g^T(\mathbf{x}_2 - \mathbf{x}_1) = 0 && (25) \end{aligned}$$

Derivation for $g^T\mathbf{x}_* + h = 0$ is as follows:

$$\begin{aligned} & g^T\mathbf{x}_* + h && (26)\\ &= g^T(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) + h && (27)\\ &= \lambda g^T\mathbf{x}_1 + (1-\lambda)g^T\mathbf{x}_2 + h && (28)\\ &= \lambda(g^T\mathbf{x}_1 + h) + (1-\lambda)(g^T\mathbf{x}_2 + h) && (29)\\ &= 0 && (30) \end{aligned}$$

Derivation for $\mathbf{x}_*^T A\mathbf{x}_* + b^T\mathbf{x}_* + c \le 0$ is as follows:

$$\begin{aligned} & \mathbf{x}_*^T A\mathbf{x}_* + b^T\mathbf{x}_* + c && (31)\\ &= (\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2)^T A(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) + b^T(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) + c && (32)\\ &= \lambda^2\mathbf{x}_1^T A\mathbf{x}_1 + (1-\lambda)^2\mathbf{x}_2^T A\mathbf{x}_2 + \lambda(1-\lambda)(\mathbf{x}_1^T A\mathbf{x}_2 + \mathbf{x}_2^T A\mathbf{x}_1) + \lambda b^T\mathbf{x}_1 + (1-\lambda)b^T\mathbf{x}_2 + c && (33)\\ &= \lambda^2\mathbf{x}_1^T A\mathbf{x}_1 - \lambda\mathbf{x}_1^T A\mathbf{x}_1 + (1-\lambda)^2\mathbf{x}_2^T A\mathbf{x}_2 - (1-\lambda)\mathbf{x}_2^T A\mathbf{x}_2 \\ &+ \lambda(\mathbf{x}_1^T A\mathbf{x}_1 + b^T\mathbf{x}_1 + c) + (1-\lambda)(\mathbf{x}_2^T A\mathbf{x}_2 + b^T\mathbf{x}_2 + c) + \lambda(1-\lambda)(\mathbf{x}_1^T A\mathbf{x}_2 + \mathbf{x}_2^T A\mathbf{x}_1) && (34)\\ &\le \lambda(\lambda-1)(\mathbf{x}_1^T A\mathbf{x}_1 + \mathbf{x}_2^T A\mathbf{x}_2) + \lambda(1-\lambda)(\mathbf{x}_1^T A\mathbf{x}_2 + \mathbf{x}_2^T A\mathbf{x}_1) && (35)\\ &= \lambda(\lambda-1)((\mathbf{x}_2 - \mathbf{x}_1)^T A(\mathbf{x}_2 - \mathbf{x}_1)) && (36)\\ &= \lambda(\lambda-1)\Big((\mathbf{x}_2 - \mathbf{x}_1)^T(A + \lambda^* gg^T)(\mathbf{x}_2 - \mathbf{x}_1) - (\mathbf{x}_2 - \mathbf{x}_1)^T(\lambda^* gg^T)(\mathbf{x}_2 - \mathbf{x}_1)\Big) && (37)\\ &\le \lambda(1-\lambda)(\mathbf{x}_2 - \mathbf{x}_1)^T(\lambda^* gg^T)(\mathbf{x}_2 - \mathbf{x}_1) && (38)\\ &= 0 && (39) \end{aligned}$$

Then we have

$$\mathbf{x}_*^T A\mathbf{x}_* + b^T\mathbf{x}_* + c \le 0 \text{ and } g^T\mathbf{x}_* + h = 0 \tag{40}$$

That is to say, for arbitrary convex combination $\mathbf{x}_* \in C_1$. Then

$$C_1 \text{ is convex set.} \tag{41}$$

Hence, it is proved that

$$\exists\lambda \in \mathbb{R},\ (A + \lambda gg^T) \in \mathbb{S}_+^n \Rightarrow C_1 \text{ is convex.} \tag{42}$$

□

3 Boyd and Vandenberghe, Ex. 2.21

Proof. Let $(a_1, b_1) \in S$, $(a_2, b_2) \in S$ and $(a_1, b_1) \neq (a_2, b_2)$, then show that arbitrary convex combination $(a_*, b_*) = (\lambda a_1 + (1-\lambda)a_2, \lambda b_1 + (1-\lambda)b_2) \in S$.
By definition of set S, we have

$$a_1^T x \leq b_1 \; \forall x \in C, \text{ and } a_1^T x \geq b_1 \; \forall x \in D \tag{43}$$
$$a_2^T x \leq b_2 \; \forall x \in C, \text{ and } a_2^T x \geq b_2 \; \forall x \in D \tag{44}$$

Then we turn to discuss $a_*^T x_c$ as follows: let x_c be arbitrary vector in set C,

$$\begin{aligned} a_*^T x_c &= (\lambda a_1 + (1-\lambda)a_2)^T x_c && (45)\\ &= \lambda a_1^T x_c + (1-\lambda) a_2^T x_c && (46)\\ &\leq \lambda b_1 + (1-\lambda) b_2 && (47)\\ &= b_* && (48) \end{aligned}$$

and now let x_d be arbitrary vector in set D,

$$\begin{aligned} a_*^T x_d &= (\lambda a_1 + (1-\lambda)a_2)^T x_d && (49)\\ &= \lambda a_1^T x_d + (1-\lambda) a_2^T x_d && (50)\\ &\geq \lambda b_1 + (1-\lambda) b_2 && (51)\\ &= b_* && (52) \end{aligned}$$

Now since we have $a_*^T x_c \leq b_* \; \forall x_c \in C$ and $a_*^T x_d \geq b_* \; \forall x_d \in D$, then we can conclude that

$$(a_*, b_*) \in S \tag{53}$$

Hence, it is proved that

$$S \text{ is convex.} \tag{54}$$

□

7 $\{x : ||x - v_1|| \leq ||x - v_2||\} = \{x : c^T x \leq d\}$

Proof.

$$\begin{aligned} & ||x - v_1|| \leq ||x - v_2|| && (55)\\ \Leftrightarrow \quad & ||x - v_1||^2 \leq ||x - v_2||^2 && (56)\\ \Leftrightarrow \quad & (x - v_1)^T (x - v_1) \leq (x - v_2)^T (x - v_2) && (57)\\ \Leftrightarrow \quad & x^T x - x^T v_1 - v_1^T x + v_1^T v_1 \leq x^T x - x^T v_2 - v_2^T x + v_2^T v_2 && (58)\\ \Leftrightarrow \quad & -v_1^T x - v_1^T x + v_1^T v_1 \leq -v_2^T x - v_2^T x + v_2^T v_2 && (59)\\ \Leftrightarrow \quad & 2(v_2 - v_1)^T x \leq v_2^T v_2 - v_1^T v_1 && (60) \end{aligned}$$

Let $c = 2(v_2 - v_1)$ and $d = v_2^T v_2 - v_1^T v_1 = ||v_2||^2 - ||v_1||^2$, then

$$c^T x \leq d \tag{61}$$

where $c = 2(v_2 - v_1)$ and $d = ||v_2||^2 - ||v_1||^2$.
Hence, we can conclude that

$$\{x : ||x - v_1|| \leq ||x - v_2||\} = \{x : c^T x \leq d\} \tag{62}$$

□

8 Exists C such that $CA = B$

Proof. By Contradiction. Assume that

$$\nexists C,\ CA = B \tag{63}$$

Then let the A_j denotes the j-th column vector of A and the B_j denotes the j-th column vector of B,

$$\nexists C,\ \forall j,\ CA_j = B_j \tag{64}$$

This naturally derives

$$\nexists C,\ A_j = C^{-1}B_j \tag{65}$$

Let x be the one that $Ax = 0$, that is $\sum_j A_j x_j = 0$.

$$\nexists C,\ \sum_j C^{-1}B_j x_j = 0 \tag{66}$$

$$\Leftrightarrow \nexists C,\ C^{-1}\sum_j B_j x_j = 0 \tag{67}$$

Since we have

$$Ax = 0 \Rightarrow Bx = 0 \tag{68}$$

$$\tag{69}$$

Then we have

$$Bx = 0 \tag{70}$$

$$\Leftrightarrow \sum_j B_j x_j = 0 \tag{71}$$

In terms of (67) and (71), we have

$$\nexists C,\ C^{-1} \cdot 0 = 0 \tag{72}$$

which contradicts the common sense that any arbitrary invertible matrix C would satisfies $C^{-1} \cdot 0 = 0$. Hence, the initial assumption should be rejected and then it is proved that

$$\exists C \text{ such that } CA = B \tag{73}$$

□

A Codes Printout

(a) Gradient Descent Routine

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Problem set 1: Standard Gradient Descent with fixed step size
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Usage:
%     [b, iter, all_costs] = gd (X, b_init, gamma)
% Parameters:
%     X: matrix in quadratic optimization
%     b_init: starting vector of variable b
%     gamma: fixed step size
% Note that stopping criteria is set by absolute eps = 10e-5.

function [b, iter, all_costs] = gd (X, b_init, gamma)
eps = 10e-5;
b = b_init;
last_cost = 0.5 * b' * X * b;
iter = 1;
all_costs = [];
while true,
    %% compute essential numerics and do gradient descent
    gradient = X * b;
    b = b - gamma * gradient;
    cost = 0.5 * b' * X * b;
    rate = (cost / last_cost);
    all_costs = [all_costs cost];
    %% output numeric information of this iteration
    disp(sprintf('Iter: %d, Cost: %e, Conv_Rate: %f',iter,cost,rate));
    %% quadratic optimization converges to zero
    if cost < eps,
        disp('Converged to zeros!')
        break
    end
    %% qudratic optimization diverges
    if cost >= last_cost && iter > 10,
        disp('Problem diverges!')
        break
    end
    %% prepare for next iteration
    last_cost = cost;
    iter = iter + 1;
end

%% uncomment following code for plotting individual gradient descent run
%plot f(b^(n)) with regard to n
%plot (1:iter, all_costs)
%title (sprintf ('Gradient Descent with gamma=%f', gamma))
%xlabel ('iteration: n')
%ylabel ('cost of function: f(b^n)')

end
```

(b) Running Script

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Running scripts for applying gradient descent
%%% on three given dataset
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% for X1, b1
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gamma1_one = 0.5; gamma2_two = 3;
[b1_opt_one, iter1_one, costs1_one] = Gradient_Descent(X1, b1, gamma1_one);
[b1_opt_two, iter1_two, costs1_two] = Gradient_Descent(X1, b1, gamma2_two);
subplot (1, 2, 1)
plot (1:iter1_one, costs1_one)
axis ([-0.1*iter1_one 1.1*iter1_one -0.05*max(costs1_one) max(costs1_one)])
title (sprintf ('Gradient Descent with gamma=%f', gamma1_one))
xlabel ('iteration: n')
ylabel ('cost of function: f(b^n)')
subplot (1, 2, 2)
plot (1:iter1_two, costs1_two)
axis ([-0.1*iter1_two 1.1*iter1_two -0.05*max(costs1_two) max(costs1_two)])
title (sprintf ('Gradient Descent with gamma=%f', gamma2_two))
xlabel ('iteration: n')
ylabel ('cost of function: f(b^n)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% for X2, b2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gamma2_one = 1.5; gamma2_two = 3;
[b2_opt_one, iter2_one, costs2_one] = Gradient_Descent(X2, b2, gamma2_one);
[b2_opt_two, iter2_two, costs2_two] = Gradient_Descent(X2, b2, gamma2_two);
figure()
subplot (1, 2, 1)
plot (1:iter2_one, costs2_one)
axis ([-0.1*iter2_one 1.1*iter2_one -0.05*max(costs2_one) max(costs2_one)])
title (sprintf ('Gradient Descent with gamma=%f', gamma2_one))
xlabel ('iteration: n')
ylabel ('cost of function: f(b^n)')
subplot (1, 2, 2)
plot (1:iter2_two, costs2_two)
axis ([-0.1*iter2_two 1.1*iter2_two -0.05*max(costs2_two) max(costs2_two)])
title (sprintf ('Gradient Descent with gamma=%f', gamma2_two))
xlabel ('iteration: n')
ylabel ('cost of function: f(b^n)')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% for X3, b3
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gamma3_one = 0.005; gamma3_two = 0.05;
[b3_opt_one, iter3_one, costs3_one] = Gradient_Descent(X3, b3, gamma3_one);
[b3_opt_two, iter3_two, costs3_two] = Gradient_Descent(X3, b3, gamma3_two);
figure()
subplot (1, 2, 1)
plot (1:iter3_one, costs3_one)
axis ([-0.1*iter3_one 1.1*iter3_one -0.05*max(costs3_one) max(costs3_one)])
title (sprintf ('Gradient Descent with gamma=%f', gamma3_one))
xlabel ('iteration: n')
ylabel ('cost of function: f(b^n)')
subplot (1, 2, 2)
plot (1:iter3_two, costs3_two)
axis ([-0.1*iter3_two 1.1*iter3_two -0.05*max(costs3_two) max(costs3_two)])
title (sprintf ('Gradient Descent with gamma=%f', gamma3_two))
xlabel ('iteration: n')
ylabel ('cost of function: f(b^n)')
```