

Machine Learning: Homework #1

Due on February 21, 2021

Professor Xiao Li

Peng Deng

Problem 1

[20 points] Suppose that $Y = \{0, 1\}$, $P(Y = 0) = P(Y = 1) = \frac{1}{2}$. X is a continuous random variable with probability density function given by

$$p_{X|Y}(x | y = 0) = \begin{cases} e^{-x}, & x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

$$p_{X|Y}(x | y = 1) = \begin{cases} \frac{1}{2}, & x \in [a, a+2] \\ 0, & \text{otherwise} \end{cases}$$

where $a \geq 0$

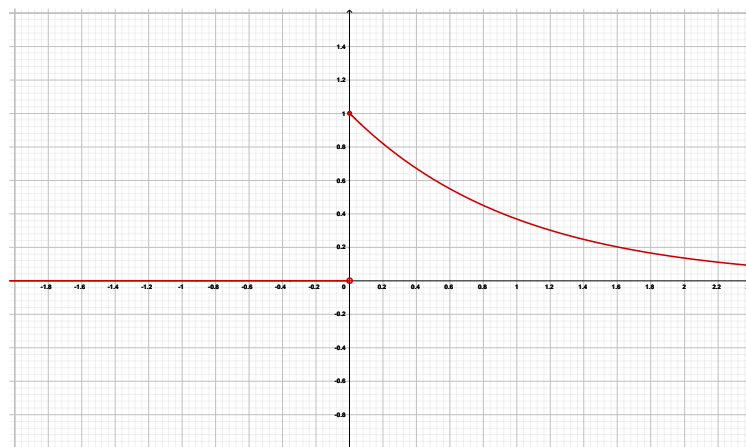
(a) Sketch $p_{X|Y}(x | y = 0)$ and $p_{X|Y}(x | y = 1)$

(b) If $X = \frac{1}{2}$, find the most likely value of Y by using Bayes' Theorem. What about $X = 1$?

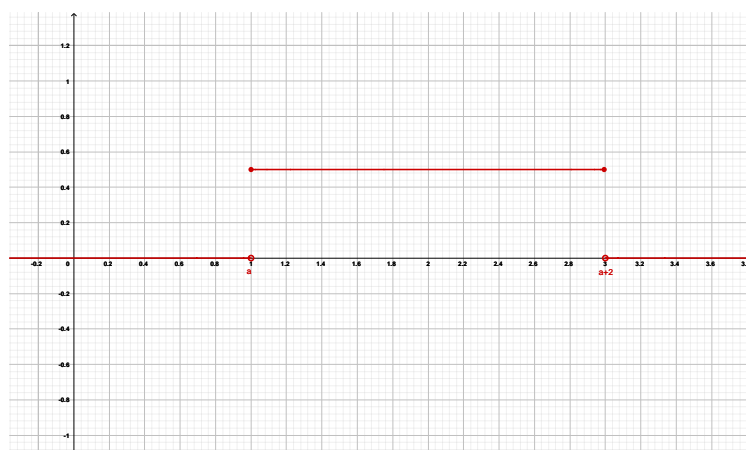
Solution

Subproblem (a)

The plots of $p_{X|Y}(x | y = 0)$ and $p_{X|Y}(x | y = 1)$ are showed in Figure 1.



(a) Probability density function 1



(b) Probability density function 2

Figure 1 The plot of probability density functions

Subproblem (b)

◦ When $X = \frac{1}{2}$, we can have

$$\begin{aligned} p\left(Y = 0 \mid X = \frac{1}{2}\right) &= \frac{p(Y = 0) \cdot p\left(X = \frac{1}{2} \mid Y = 0\right)}{p\left(X = \frac{1}{2}\right)} = \frac{\frac{1}{2} \cdot e^{-\frac{1}{2}}}{p\left(X = \frac{1}{2}\right)} \\ p\left(Y = 1 \mid X = \frac{1}{2}\right) &= \frac{p(Y = 1) \cdot p\left(X = \frac{1}{2} \mid Y = 1\right)}{p\left(X = \frac{1}{2}\right)} = \frac{\frac{1}{2} \cdot p\left(X = \frac{1}{2} \mid Y = 1\right)}{p\left(X = \frac{1}{2}\right)} \\ &\leq \frac{\frac{1}{2} \cdot \frac{1}{2}}{p\left(X = \frac{1}{2}\right)} < \frac{\frac{1}{2} \cdot e^{-\frac{1}{2}}}{p\left(X = \frac{1}{2}\right)} = p\left(Y = 0 \mid X = \frac{1}{2}\right) \end{aligned} \quad (1)$$

As we can see from equation 1, the conditional probability $p(Y = 0 \mid X = \frac{1}{2})$ is larger than $p(Y = 1 \mid X = \frac{1}{2})$. Thus, the most likely value of Y is 0.

◦ When $X = 1$, we can have

$$\begin{aligned} p(Y = 0 \mid X = 1) &= \frac{p(Y = 0) \cdot p(X = 1 \mid Y = 0)}{p(X = 1)} = \frac{\frac{1}{2} \cdot e^{-1}}{p(X = 1)} \\ p(Y = 1 \mid X = 1) &= \frac{p(Y = 1) \cdot p(X = 1 \mid Y = 1)}{p(X = 1)} = \frac{\frac{1}{2} \cdot p(X = 1 \mid Y = 1)}{p(X = 1)} \end{aligned} \quad (2)$$

According to equation 2, we can see that different value of a will result in different conclusion.

- $1 \in [a, a + 2]$

In this situation, we have

$$\begin{cases} a \geq 0 \\ a \leq 1 \\ a + 2 \geq 1 \end{cases} \implies a \in [0, 1] \quad (3)$$

Then, we can have

$$\begin{aligned} p(Y = 1 \mid X = 1) &= \frac{\frac{1}{2} \cdot p(X = 1 \mid Y = 1)}{p(X = 1)} = \frac{\frac{1}{2} \cdot \frac{1}{2}}{p(X = 1)} \\ &> \frac{\frac{1}{2} \cdot e^{-1}}{p(X = 1)} = p(Y = 0 \mid X = 1) \end{aligned} \quad (4)$$

As we can see from equation 3 and 4, when $a \in [0, 1]$, the conditional probability $p(Y = 0 \mid X = 1)$ is less than $p(Y = 1 \mid X = 1)$. Thus, the most likely value of Y is 1.

- $1 \notin [a, a + 2]$

In this situation, we have

$$\begin{cases} a \geq 0 \\ a > 1 \text{ or } a + 2 < 1 \end{cases} \implies a \in (1, +\infty) \quad (5)$$

Then, we can have

$$\begin{aligned} p(Y = 1 \mid X = 1) &= \frac{\frac{1}{2} \cdot p(X = 1 \mid Y = 1)}{p(X = 1)} = \frac{\frac{1}{2} \cdot 0}{p(X = 1)} = 0 \\ &< \frac{\frac{1}{2} \cdot e^{-1}}{p(X = 1)} = p(Y = 0 \mid X = 1) \end{aligned} \quad (6)$$

As we can see from equation 5 and 6, when $a \in (1, +\infty)$, the conditional probability $p(Y = 0 \mid X = 1)$ is larger than $p(Y = 1 \mid X = 1)$. Thus, the most likely value of Y is 0.

Problem 2

[10 points] Markov's inequality is the most elementary tail bound which means that if a non-negative random variable X has finite mean, then we have

$$\Pr[X \geq t] \leq \frac{E[X]}{t} \quad \forall t > 0$$

For a random variable X with finite variance, then show that it satisfies the Chebyshev's inequality

$$\Pr[|X - \mu| \geq t] \leq \frac{\text{Var}(X)}{t^2} \quad \forall t > 0$$

Solution

We set $D = \{x : |x - \mu| \geq t\}$, thus we have

$$\frac{|x - \mu|}{t} \geq 1, \quad x \in D \quad (7)$$

Then, we set $f(x)$ as the probability density function of X , so we have

$$\begin{aligned} \Pr[|X - \mu| \geq t] &= \int_D f(x) dx = \int_D 1 \cdot f(x) dx \\ &\leq \int_D \frac{|x - \mu|}{t} \cdot f(x) dx \quad (\text{equation 7}) \\ &\leq \int_D \left(\frac{|x - \mu|}{t} \right)^2 \cdot f(x) dx \quad (\text{equation 7}) \\ &= \frac{1}{t^2} \int_D (x - \mu)^2 \cdot f(x) dx \\ &\leq \frac{1}{t^2} \int_{-\infty}^{+\infty} (x - \mu)^2 \cdot f(x) dx \\ &= \frac{E[(X - \mu)^2]}{t^2} = \frac{\text{Var}(X)}{t^2} \end{aligned} \quad (8)$$

Problem 3

[20 points] Let $X \in R^{m \times n}$ be a matrix of full column rank. Show that

$$\min_{\theta \in R^n} \|y - X\theta\|_2^2 = \|P_{b_n}^\perp \cdots P_{b_2}^\perp P_{b_1}^\perp y\|_2^2,$$

where $b_1 = x_1, b_2 = P_{b_1}^\perp x_2, b_3 = P_{b_2}^\perp P_{b_1}^\perp x_3, \dots, b_n = P_{b_{n-1}}^\perp \cdots P_{b_2}^\perp P_{b_1}^\perp x_n$. (Hint: $P_{b_i}^\perp$ is the projection of orthogonal complementary space of b_i .)

Solution

Because X is a matrix of full column rank, we have that $\{x_i\}, i = 1, 2, \dots, n$ are linearly independent and the solution of $\arg\min_{\theta \in R^n} \|y - X\theta\|_2^2$ is $\hat{\theta} = (X^T X)^{-1} X^T y$. Then we can have:

$$\min_{\theta \in R^n} \|y - X\theta\|_2^2 = \|y - X (X^T X)^{-1} X^T y\|_2^2 = \|(I - P)y\|_2^2 \quad (9)$$

where $P = X (X^T X)^{-1} X^T$ is the projection matrix onto the range space $\mathcal{R}(X)$, thus, $I - P$ is the projection matrix onto the orthogonal complement space of $\mathcal{R}(X)$.

• Then, we would like to prove that $P_{b_k}^\perp b_k = 0$. We know that the vector b_k can be decomposed uniquely onto the space b_k^\perp and space $R^m - b_k^\perp + \{0\}$. So we have

$$\begin{aligned} b_k &= P_{b_k}^\perp b_k + P_{R^m - b_k^\perp + \{0\}} b_k \implies P_{b_k}^\perp b_k = 0 \\ &= P_{b_k}^\perp b_k + b_k \end{aligned} \quad (10)$$

Thus, we can have (suppose $i > j$)

$$\begin{aligned} b_i \cdot b_j &= \langle P_{b_{i-1}}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp x_i, b_j \rangle \\ &= \langle P_{b_{i-1}}^\perp \dots P_{b_{j-1}}^\perp P_{b_{j+1}}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp x_i, P_{b_j}^\perp b_j \rangle \\ &= \langle P_{b_{i-1}}^\perp \dots P_{b_{j-1}}^\perp P_{b_{j+1}}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp x_i, 0 \rangle \\ &= 0 \quad (\forall i \neq j) \end{aligned} \quad (11)$$

since $P_{b_k}^\perp$ is self-adjoint and $P_{b_k}^\perp x_k = 0$. Thus, $\{b_i\}, i = 1, 2, \dots, n$ are orthogonal to each other, so they are independent to each other. Then, we can have $W = \text{span}\{b_1, b_2, \dots, b_n\} = \text{span}\{x_1, x_2, \dots, x_n\} = \mathcal{R}(X)$.

• Then, we would like to prove that $P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp$ is the projection matrix onto the orthogonal complement space of W . Choose a vector $\mathbf{u} \in \mathbb{R}^m$, it is equal to prove that $\mathbf{v} = P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp \mathbf{u} \in W^\perp$. Then it is equal to prove that \mathbf{v} is orthogonal to any vector in W . We denote the vector $\mathbf{w}_i \in W$ as follow

$$\mathbf{w}_i = a_1 b_1 + a_2 b_2 + \dots + a_n b_n \quad (12)$$

Thus, we have

$$\begin{aligned} \mathbf{v} \cdot \mathbf{w}_i &= \langle P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp \mathbf{u}, a_1 b_1 + a_2 b_2 + \dots + a_n b_n \rangle \\ &= \langle P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp \mathbf{u}, a_1 b_1 \rangle + \langle P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp \mathbf{u}, a_2 b_2 \rangle + \dots + \langle P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp \mathbf{u}, a_n b_n \rangle \\ &= \langle P_{b_n}^\perp \dots P_{b_3}^\perp P_{b_2}^\perp \mathbf{u}, P_{b_1}^\perp a_1 b_1 \rangle + \langle P_{b_n}^\perp \dots P_{b_3}^\perp P_{b_1}^\perp \mathbf{u}, P_{b_2}^\perp a_2 b_2 \rangle + \dots + \langle P_{b_{n-1}}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp \mathbf{u}, P_{b_n}^\perp a_n b_n \rangle \\ &= 0 \end{aligned} \quad (13)$$

Thus, $P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp$ is the projection matrix onto the orthogonal complement space of W , as well as the orthogonal complement space of $\mathcal{R}(X)$. Due to the unique of projection matrix, we have $P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp = I - P$, which shows that

$$\min_{\theta \in \mathbb{R}^n} \|y - X\theta\|_2^2 = \|(I - P)y\|_2^2 = \|P_{b_n}^\perp \dots P_{b_2}^\perp P_{b_1}^\perp y\|_2^2 \quad (14)$$

Problem 4

[50points] MLE for robust regression. Suppose we have the generative linear regression model

$$Y = X\theta^* + \varepsilon$$

where ε is the error term and $\varepsilon \sim N(0, \Sigma)$. The maximum likelihood estimator for θ is:

$$\begin{aligned} \hat{\theta}_{LS} &= \operatorname{argmin}_{\theta \in \mathbb{R}^d} \|X\theta - y\|_2^2 \\ &= (X^T X)^{-1} X^T y \end{aligned}$$

- (a) Suppose the error term, $\varepsilon = [\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n]$ follows the Laplace distribution, i.e. $\varepsilon_i \stackrel{\text{i.i.d.}}{\sim} L(0, b), i = 1, 2, \dots, n$ and the probability density function is $P(\varepsilon_i) = \frac{1}{2b} e^{-\frac{|\varepsilon_i - 0|}{b}}$ for some $b > 0$. Under the MLE principle, what is the learning problem? Please write out the derivation process. (15 points)

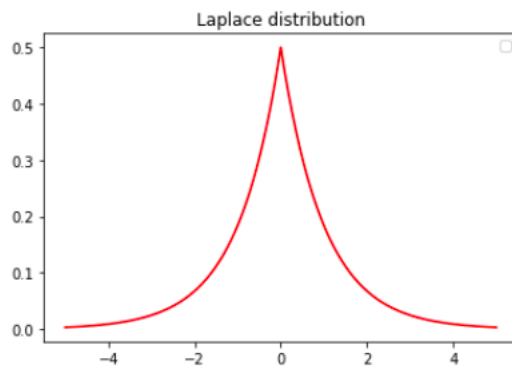


Figure 2 PDF of Laplace distribution

(b) **Huber-smoothing.** $L1$ - norm minimization

$$\hat{\theta}_{L1} = \underset{\theta}{\operatorname{argmin}} \|X\theta - y\|_1$$

is one possible solution for robust regression. However, it is nondifferentiable. We utilize smoothing technique for approximately solving the $L1$ - norm minimization. Huber function is one possibility. The definition and sketch map are shown as below.

$$h_{\mu}(z) \begin{cases} |z|, & |z| \geq \mu \\ \frac{z^2}{2\mu} + \frac{\mu}{2}, & |z| \leq \mu \end{cases}$$

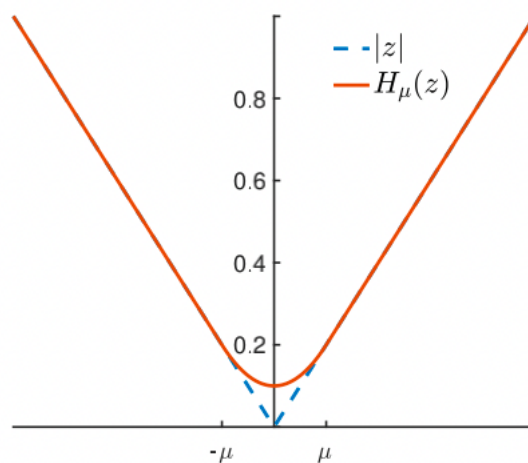


Figure 3 Huber smoothing

Then,

$$H_{\mu}(Z) = \sum_{j=1}^n h_{\mu}(z_j)$$

By using Huber smoothing, the approximation of the optimization of $L1$ - norm can be changed to

$$\min_{\theta} H_{\mu}(X\theta - y)$$

Let

$$f(\theta) = H_\mu(X\theta - y)$$

find the gradient $\nabla f(\theta)$. (10 points)

- (c) Gradient descent for minimizing $f(\theta)$. The process of gradient descent algorithm is shown in the following table.

Algorithm 1: The Process of Gradient Descent Algorithm

```

1 Input: observed data  $X, y$  and initialization parameter  $\theta_0$ ,
  Huber smoothing parameter  $\mu$ ,
  total iteration number  $T$ ,
  learning rate  $\alpha$ .
2 for  $k = 0, 1, 2, \dots, T$  do
3    $\theta_{k+1} = \theta_k - \alpha \nabla f(\theta_k)$ 
4 end for
5 return  $\theta_T$ 

```

The data set is generated by the linear model

$$Y = X\theta^* + \varepsilon_1 + \varepsilon_2$$

where $\varepsilon_1 \in R^n$ follows Gaussian distribution, ε_2 are outliers. Given the observed data $(x, y) = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ and true value θ^* ,

- (1) calculate the estimation $\hat{\theta}_{LS}$ by using linear least squares and compute $\|\hat{\theta}_{LS} - \theta^*\|_2$. (5 points)
- (2) suppose $n = 1000, d = 50$, use python to implement the gradient descent algorithm to minimize $f(\theta)$, the parameters are set as $\mu = 10^{-5}, \alpha = 0.001, T = 1000$, plot the error $\|\theta_k - \theta^*\|_2$ as a function of iteration number. You can download the data $\{Y, X, \theta^*\}$ from Blackboard. (20 points)

Solution

Subproblem (a)

Our model is

$$y_i = \mathbf{x}_i^\top \boldsymbol{\theta} \quad (15)$$

To be more explicit, consider

$$y_i = \mathbf{x}_i^\top \boldsymbol{\theta} + \epsilon_i \quad \text{with} \quad \epsilon_i \sim L(0, b) \quad (16)$$

where ϵ_i are i.i.d. for $i = 1, \dots, n$.

Equivalently,

$$\epsilon_i = y_i - \mathbf{x}_i^\top \boldsymbol{\theta} \sim L(0, b) \quad (17)$$

Thus, we can get the likelihood function as follow

$$\begin{aligned}
 L(\boldsymbol{\theta}) &= \prod_{i=1}^n P(\epsilon_i) = \prod_{i=1}^n \frac{1}{2b} e^{-\frac{|\epsilon_i|}{b}} \\
 &= (2b)^{-n} e^{-\frac{\sum_{i=1}^n |\epsilon_i|}{b}}
 \end{aligned} \quad (18)$$

Then, we can get the log-likelihood function as follow

$$\begin{aligned}\log L(\boldsymbol{\theta}) &= -n \log(2b) - \frac{1}{b} \sum_{i=1}^n |\epsilon_i| \\ &= \text{Constant} - \frac{1}{b} \sum_{i=1}^n |\epsilon_i|\end{aligned}\tag{19}$$

In order to make log-likelihood function to reach maximum value, we can derive the learning problem as follow

$$\begin{aligned}\hat{\boldsymbol{\theta}}_{MLE} &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^n |\epsilon_i| \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{i=1}^n |y_i - \mathbf{x}_i^\top \boldsymbol{\theta}| \\ &= \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\theta}\|_1\end{aligned}\tag{20}$$

Subproblem (b)

We can derive the gradient of $f(\boldsymbol{\theta})$ as follow

$$\begin{aligned}\nabla f(\boldsymbol{\theta}) &= \begin{pmatrix} \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_1} \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_2} \\ \vdots \\ \frac{\partial f(\boldsymbol{\theta})}{\partial \theta_d} \end{pmatrix} = \begin{pmatrix} \frac{\partial H_\mu(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}{\partial \theta_1} \\ \frac{\partial H_\mu(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}{\partial \theta_2} \\ \vdots \\ \frac{\partial H_\mu(\mathbf{X}\boldsymbol{\theta} - \mathbf{y})}{\partial \theta_d} \end{pmatrix} = \begin{pmatrix} \frac{\partial \sum_{i=1}^n h_u(\mathbf{x}_i^\top \boldsymbol{\theta} - y_i)}{\partial \theta_1} \\ \frac{\partial \sum_{i=1}^n h_u(\mathbf{x}_i^\top \boldsymbol{\theta} - y_i)}{\partial \theta_2} \\ \vdots \\ \frac{\partial \sum_{i=1}^n h_u(\mathbf{x}_i^\top \boldsymbol{\theta} - y_i)}{\partial \theta_d} \end{pmatrix} \\ &= \begin{pmatrix} \sum_{i=1}^n \frac{\mathbf{x}_i^\top \boldsymbol{\theta} - y_i}{\max\{|\mathbf{x}_i^\top \boldsymbol{\theta} - y_i|, \mu\}} x_{i1} \\ \sum_{i=1}^n \frac{\mathbf{x}_i^\top \boldsymbol{\theta} - y_i}{\max\{|\mathbf{x}_i^\top \boldsymbol{\theta} - y_i|, \mu\}} x_{i2} \\ \vdots \\ \sum_{i=1}^n \frac{\mathbf{x}_i^\top \boldsymbol{\theta} - y_i}{\max\{|\mathbf{x}_i^\top \boldsymbol{\theta} - y_i|, \mu\}} x_{id} \end{pmatrix} \\ &= \mathbf{X}^\top \cdot \begin{pmatrix} \frac{\mathbf{x}_1^\top \boldsymbol{\theta} - y_1}{\max\{|\mathbf{x}_1^\top \boldsymbol{\theta} - y_1|, \mu\}} \\ \frac{\mathbf{x}_2^\top \boldsymbol{\theta} - y_2}{\max\{|\mathbf{x}_2^\top \boldsymbol{\theta} - y_2|, \mu\}} \\ \vdots \\ \frac{\mathbf{x}_n^\top \boldsymbol{\theta} - y_n}{\max\{|\mathbf{x}_n^\top \boldsymbol{\theta} - y_n|, \mu\}} \end{pmatrix}\end{aligned}\tag{21}$$

Subproblem (c)

- (1) Since $n > d$, we suppose $\mathbf{X} \in \mathbb{R}^{n \times d}$ has full column rank. Thus, we can derive that $\mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{d \times d}$ is invertible. Then, we can calculate $\hat{\boldsymbol{\theta}}_{LS}$ as follow

$$\hat{\boldsymbol{\theta}}_{LS} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}\tag{22}$$

By using Python, we can compute $\|\hat{\boldsymbol{\theta}}_{LS} - \boldsymbol{\theta}^*\|_2$, the result is as follow

$$\|\hat{\boldsymbol{\theta}}_{LS} - \boldsymbol{\theta}^*\|_2 = 144.695\tag{23}$$

The Python code to solve this problem is showed as follow


```

import pandas as pd
import numpy as np
X = pd.read_csv("Sample data of X.csv", header=0, index_col=0)
y = pd.read_csv("Sample data of y.csv", header=None)
theta_star = pd.read_csv("data of theta_star.csv", header=None)

X = np.array(X)
y = np.array(y)
theta_star = np.array(theta_star)

theta_LS = np.dot(np.dot(np.linalg.inv(np.dot(X.T, X)), X.T), y)
error = np.linalg.norm(theta_star - theta_LS, ord = 2)
print(error)

```

- (2) By using Python, we can plot the error $\|\theta_k - \theta^*\|_2$ as a function of iteration number as Figure 4

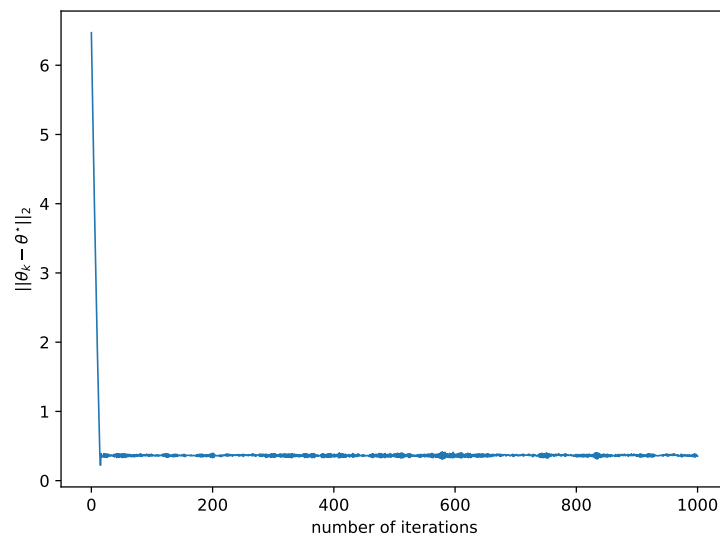


Figure 4 Error $\|\theta_k - \theta^*\|_2$ vs. iteration number

The Python code to solve this problem is showed as follow

```

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

def hu(z):
    if abs(z) >= mu:
        return abs(z)
    else:
        return z*z/(2*mu) + mu/2
def f(theta):
    return sum(map(hu, np.dot(X, theta) - y))[0]
def df(theta):
    t = np.dot(X, theta) - y
    t1 = np.dot(X, theta) - y
    t[abs(t)>=mu] = abs(t)[abs(t)>=mu]
    t[abs(t)<=mu] = mu
    return np.dot(X.T, t1/t)

```

```

def GM(theta):
    thetak = theta
    norm_list = []
    norm_list.append(np.linalg.norm(thetak-theta_star))
    for k in np.arange(T):
        thetak = thetak - alpha * df(thetak)
        norm_list.append(np.linalg.norm(thetak-theta_star))
    norm_list = np.array(norm_list)
    plot_convergence(norm_list)
    return thetak

def plot_convergence(norm_list):
    number = norm_list.size
    x = np.arange(number)
    #y = np.log(norm_list)
    y = norm_list
    plt.plot(x, y, linewidth=1)
    plt.xlabel('number of iterations')
    plt.ylabel(r'$||\theta_k - \theta^{\star}||_2$')
    plt.tight_layout()
    plt.savefig('./plt.pdf', dpi=1000)

X = pd.read_csv("Sample data of X.csv", header=0, index_col=0)
y = pd.read_csv("Sample data of y.csv", header=None)
theta_star = pd.read_csv("data of theta_star.csv", header=None)

X = np.array(X)
y = np.array(y)
theta_star = np.array(theta_star)

n = X.shape[0]
d = X.shape[1]
mu = 1e-5
alpha = 1e-3
T = 1000
theta0 = np.zeros(d).reshape(d,1)
thetaT = GM(theta0)

```