Ex. 1

See code and output at the end of this document.

Ex, 2

$$g_\theta = \theta^T X \qquad [\theta_0, \theta_1, \theta_2] \begin{bmatrix} x_{bmi} \\ x_{ht} \\ 1 \end{bmatrix}$$

$$\hat{\theta} = \arg\min \sum_{n=1}^{N} (y_n - g_\theta(x_n))^2$$

$$\Rightarrow \hat{\theta} = \arg\min \| y - X\theta \|^2$$

a) $J(\theta) = \| y - X\theta \|^2 = \sqrt{(y - X\theta)^2}^{\,2} = (y - X\theta)^2$

$\nabla_\theta J(\theta) = 0$

$$2X^T(y - X\hat{\theta}) = 0.$$

$$X^T y - X^T X \hat{\theta} = 0$$

$$\boxed{\hat{\theta} = (X^T X)^{-1} X^T y}$$

To have a unique minimizer, $\nabla^2_\theta J(\theta)$ needs to be positive definite.

If $X^T X$ is singular, we can use regularization & LASSO regression.

c) See code

d). Find $\nabla \mathcal{E}_{train}(\theta^k)$.

$$\theta^{k+1} = \theta^k - \alpha^k \nabla \mathcal{E}_{train}(\theta^k)$$

$$\nabla \mathcal{E}_{train}(\theta^k) = \nabla_\theta \| y_n - X\theta^{(k)} \|^2$$

$$= 2X^T(y_n - X\theta^{(k)})$$

$$= 2X^T y - 2X^T X \theta^k$$

$$= 2(X^T y - X^T X \theta^k) \quad = \underline{d}$$

$$\theta^{k+1} = \theta^k - \alpha^k (b - A\theta^k)$$

$$\boxed{\theta^{k+1} = \theta^k - \alpha^k b + \alpha^k A \theta^k.}$$

$$\min_\alpha J(\underline{\theta} + \alpha \underline{d}) = \| \underline{y} - X(\underline{\theta} + \alpha d) \|^2.$$

$$\nabla_\alpha J(\underline{\theta} + \alpha d) = \frac{d}{d\alpha}\left[\|y\|^2 + \|X(\underline{\theta} + \alpha \underline{d})\|^2 - 2\underline{y}^T X(\theta + \alpha \underline{d}) \right]$$

$$= \frac{d}{d\alpha}\left[\|y\|^2 + (\underline{\theta} + \alpha \underline{d})^T X^T X (\underline{\theta} + \alpha \underline{d}) - 2y^T X(\theta + \alpha d) \right]$$

$$= \frac{d}{d\alpha}\left[(\underline{\theta} + \alpha \underline{d})^T X^T X(\underline{\theta} + \alpha \underline{d}) - 2y^T X \alpha \underline{d} \right].$$

$$= \frac{d}{d\alpha}\left[\underline{\theta}^T X^T X(\underline{\theta} + \alpha \underline{d}) + \alpha \underline{d}^T X^T X(\underline{\theta} + \alpha \underline{d}) \right] - 2y^T X \underline{d}$$

$$= \theta^T X^T X \underline{d} + \underline{d}^T X^T X \underline{\theta} + 2\alpha d^T X^T X \underline{d} - 2y^T X \underline{d}$$

$$= 2\theta^T X^T X \underline{d} + 2\alpha d^T X^T X \underline{d} - 2y^T X \underline{d}$$

$$= 2(\underline{\theta}^T X^T X \underline{d} + \alpha d^T X^T X \underline{d} - y^T X \underline{d})$$

$$\nabla_\alpha J(\underline{\theta} + \alpha \underline{d}) = 0.$$

$$\underline{\theta}^T X^T X \underline{d} + \alpha \| X \underline{d} \|^2 - y^T X \underline{d} = 0,$$

$$\alpha^k = - \frac{\underline{\theta}^T X^T X \underline{d}^k - \underline{y}^T X \underline{d}^k}{\| X \underline{d}^k \|^2}$$

$$= - \frac{(\theta^T X^T X - y^T X) \underline{d}^k}{\| X d^k \|^2}$$

$$= - \frac{\frac{1}{2} d^{kT} \cdot d^k}{d^{kT} X^T \cancel{X} d^k}$$

$$= -\frac{1}{2} \frac{d^{kT} \cdot d^k}{d^{kT} \cdot A \cdot d^k}$$

4) $$\hat{\theta}_\lambda = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \|X\theta - y\|_2^2 + \alpha\|\theta\|_2^2 \quad (3)$$

$$\hat{\theta}_\alpha = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \|X\theta - y\|_2^2 \quad \text{s.t.} \quad \|\theta\|_2^2 \leq \alpha \quad (4)$$

$$\hat{\theta}_\epsilon = \underset{\theta \in \mathbb{R}^d}{\operatorname{argmin}} \|\theta\|_2^2 \quad \text{s.t.} \quad \|X\theta - y\|_2^2 \leq \epsilon \quad (5)$$

a) Looking at (3).

$$J(\hat{\theta}) = \|X\hat{\theta} - y\|^2 + \alpha\|\hat{\theta}\|^2 = (X\hat{\theta} - y)^2 + \alpha\hat{\theta}^2$$

$$\nabla_\theta J(\hat{\theta}) = 0$$

$$2X^T(X\hat{\theta} - y) + 2\alpha\hat{\theta} = 0$$

$$2X^TX\hat{\theta} - 2X^Ty + 2\alpha\hat{\theta} = 0$$

$$(X^TX + \lambda I)\hat{\theta} = X^Ty$$

$$\hat{\theta}_\lambda = (X^TX + \alpha I)^{-1}X^Ty$$

4 b). $\hat{\theta}_\alpha = \underset{\theta \in \mathbb{R}^d}{\arg\min} \| X\theta - y\|_2^2 \quad s.t. \quad \|\theta\|_2^2 \leq \alpha \quad (4)$

$\hat{\theta}_\epsilon = \underset{\theta \in \mathbb{R}^d}{\arg\min} \| \theta\|_2^2 \quad s.t. \quad \|X\theta - y\|_2^2 \leq \epsilon \quad (5)$

(i) For (3)

$$\mathcal{L}(\theta) = \|X\theta - y\|_2^2 + \lambda \|\theta\|_2^2$$

For (4) :

$$\mathcal{L}(\theta, \gamma_\alpha) = \|X\theta - y\|_2^2 - \gamma_\alpha (\alpha - \|\theta\|_2^2)$$

For (5)

$$\mathcal{L}(\theta, \gamma_\varepsilon) = \|\theta\|^2 - \gamma_\varepsilon (\varepsilon - \|X\theta - y\|^2)$$

4 b ii)

(3) : Stationarity : $\nabla_\theta \mathcal{L} = (X^T X + \lambda I)\,\hat{\theta} - X^T y = 0$

Primal feasibility : None because it has no constraint.

Dual feasibility : None . . . .

Complementary slackness : None . . . .

---

(4). Stationarity : $\nabla_\theta \mathcal{L} = 0$.

$$\therefore\ X^T(X\theta - y) - \gamma_\alpha\,\theta = 0$$

Primal feasibility : $(\alpha - \|\theta\|_2^2) \geq 0$

Dual feasibility : $\gamma_\alpha \geq 0$.

complementary slackness : $\gamma_\alpha \cdot (\alpha - \|\theta\|_2^2) = 0$.

---

(5) Stationarity : $\nabla_\theta \mathcal{L} = 0$

$$\theta - \gamma_\varepsilon X^T(X\theta - y) = 0.$$

Primal feasibility : $(\varepsilon - \|X\theta - y\|_2^2) \geq 0$

Dual feasibility : $\gamma_\varepsilon \geq 0$

Complementary slackness : $\gamma_\varepsilon\,(\varepsilon - \|X\theta - y\|_2^2) = 0$

4b) iii) $\hat{\theta}_\alpha = (X^T X + \alpha I)^{-1} X^T y$.

$X^T (X \hat{\theta}_\alpha - y) - \gamma_\alpha \hat{\theta}_\alpha = 0$ (Stationarity)

$X^T X \hat{\theta}_\lambda - X^T y - \gamma_\alpha \hat{\theta}_\alpha = 0$

$X^T X \hat{\theta}_\alpha - (X^T X + \lambda I) \hat{\theta}_\alpha - \gamma_\alpha \hat{\theta}_\alpha = 0$.

$(\cancel{X^T X} - \cancel{X^T X} - \lambda I - \gamma_\alpha I) \hat{\theta} = 0$.

$\boxed{\gamma_\alpha = \lambda} > 0$ (Primal feasibility).

$\gamma_\alpha (\alpha - \|\hat{\theta}_\alpha\|^2) = 0$ (complementary slackness)

$\boxed{\alpha = \|(X^T X + \lambda I)^{-1} X^T y\|_2^2}$

4 b) iv).

$$\hat{\theta}_\lambda - \gamma_\varepsilon X^T(X\hat{\theta}_\lambda - y) = 0.$$

$$\hat{\theta}_\lambda - \gamma_\varepsilon X^T X \hat{\theta}_\lambda - \gamma_\varepsilon X^T y = 0,$$

$$\hat{\theta}_\lambda - \gamma_\varepsilon X^T X \hat{\theta}_\lambda - \gamma_\varepsilon (X^T X + \lambda I)\hat{\theta}_\lambda = 0$$

$$\left[I - \gamma_\varepsilon(\cancel{X^T X} - \cancel{X^T X} + \lambda I)\right]\hat{\theta}_\lambda = 0$$

$$\left[(1 - \gamma_\varepsilon \lambda) I\right]\hat{\theta}_\lambda = 0$$

$$\boxed{\gamma_\varepsilon = \frac{1}{\lambda}} > 0 \quad \text{(stationarity)}$$

complementary slackness: $\gamma_\varepsilon(\varepsilon - \|X\theta - y\|_2^2) = 0$

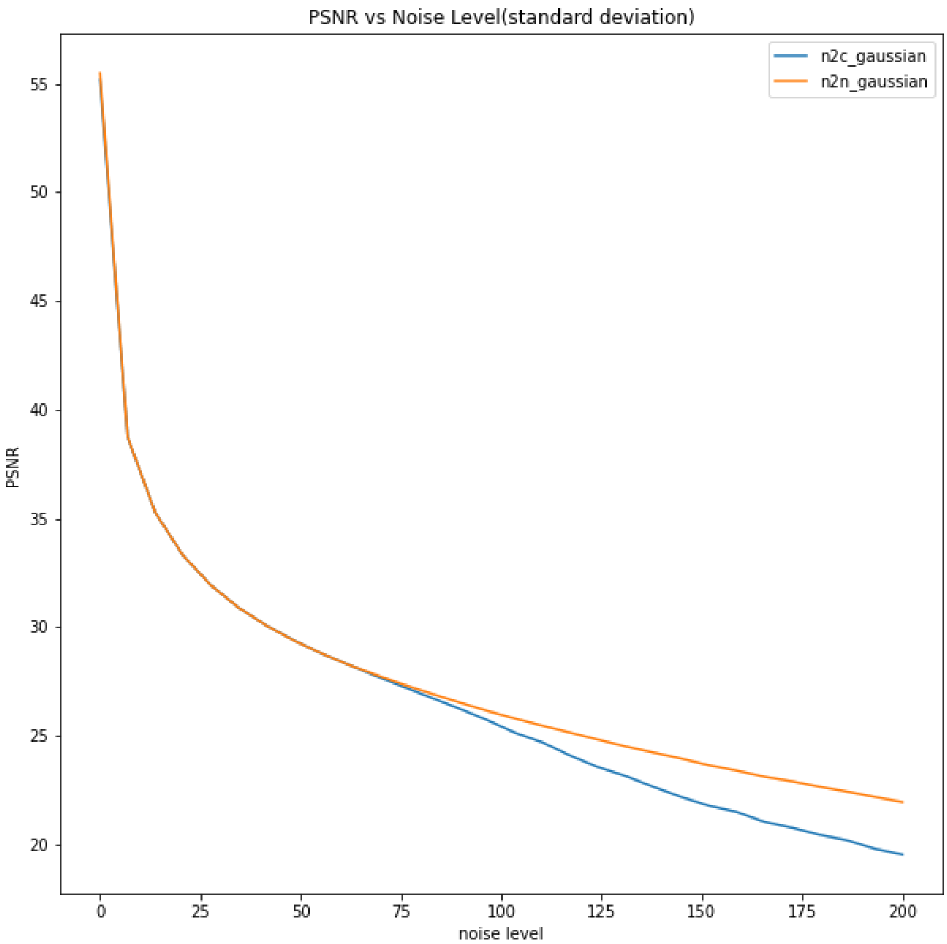
$$\varepsilon - \|X\hat{\theta}_\lambda - y\|_2^2 = 0$$

$$\varepsilon = \|X\hat{\theta}_\lambda - y\|_2^2$$

$$\boxed{\varepsilon = \| X(X^T X + \lambda I)^{-1} X^T y - y \|_2^2}$$

4 b) v). KKT condition are necessary but not conclusive to guarantee a solution. However, since the problem we are optimizing is smooth, these conditions are sufficient for optimality, and we can claim that $\hat{\theta}_\lambda$ is the solution to (4).

5.



Noise levels are generate from 0 to 200 at normal distribution, then normalized by dividing by 255.

When the noise level is low, it's clear that both models perform similarly. However, as noise level get higher, the model trained with noise data (n2n) has a higher average PSNR score than the one trained with clean data.

hw2

February 19, 2021

1 Homework 2

[1]:
```
%config IPCompleter.use_jedi = False
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
import cvxpy as cp
import pandas as pd
np.set_printoptions(precision=4)

from pathlib import Path
fig_path = str(Path().absolute())+'/figures/hw2/'
print(fig_path)
```

```
/home/zpyang/grad_courses/2021_spring/ece595_ml/figures/hw2/
```

1.1 Exercise 1

[2]:
```
data_path = str(Path().absolute())+'/hw2/data/'

male_train_data = pd.read_csv(data_path+'male_train_data.csv')
female_train_data = pd.read_csv(data_path+'female_train_data.csv')
m_bmi = male_train_data['male_bmi']*0.1
m_stature = male_train_data['male_stature_mm']*0.001
f_bmi = female_train_data['female_bmi']*0.1
f_stature = female_train_data['female_stature_mm']*0.001

print('Female BMI: '+str(f_bmi.head(10).values))
print('Female Stature(m): '+str(f_stature.head(10).values))
print('Male BMI: '+str(m_bmi.head(10).values))
print('Male Stature(m): '+str(m_stature.head(10).values))
```

```
Female BMI: [2.82 2.22 2.71 2.81 2.55 2.3  3.56 3.11 2.46 4.3 ]
Female Stature(m): [1.563 1.716 1.484 1.651 1.548 1.665 1.564 1.676 1.69  1.704]
Male BMI: [3.   2.56 2.42 2.74 2.59 2.53 2.27 2.54 3.41 3.34]
Male Stature(m): [1.679 1.586 1.773 1.816 1.809 1.662 1.829 1.686 1.761 1.797]
```

2 Exercise 2

2.1 2a) see hand written part

2.2 2b)

[3]:
```
N_male = m_bmi.values.shape[0]
N_female = f_bmi.values.shape[0]
N = N_male+N_female

X = np.block([
    [np.ones(N)],
    [m_bmi.values, f_bmi.values],
    [m_stature.values, f_stature.values],
]).T
y = np.block([np.ones(N_male), -np.ones(N_female)])
yT = np.array([y]).T

theta = np.linalg.lstsq(X, y, rcond=None)[0]
theta
```

[3]:
```
array([-10.7018,  -0.1234,   6.6749])
```

2.3 2c)

[4]:
```
P = 3
var = cp.Variable(P)
objective = cp.Minimize(cp.sum_squares(y-X @ var))
constraints = []
prob = cp.Problem(objective=objective, constraints=constraints)
prob.solve()
theta = var.value
theta
```

[4]:
```
array([-10.7018,  -0.1234,   6.6749])
```

2.4 2 d) e)

[5]:
```
n = 50000
theta0 = np.zeros(3);
theta_k = theta0

A = X.T @ X
b = X.T @ y
cost_normal = []
for k in range(n):
    d_k = -2*A @ theta_k + 2*b
```

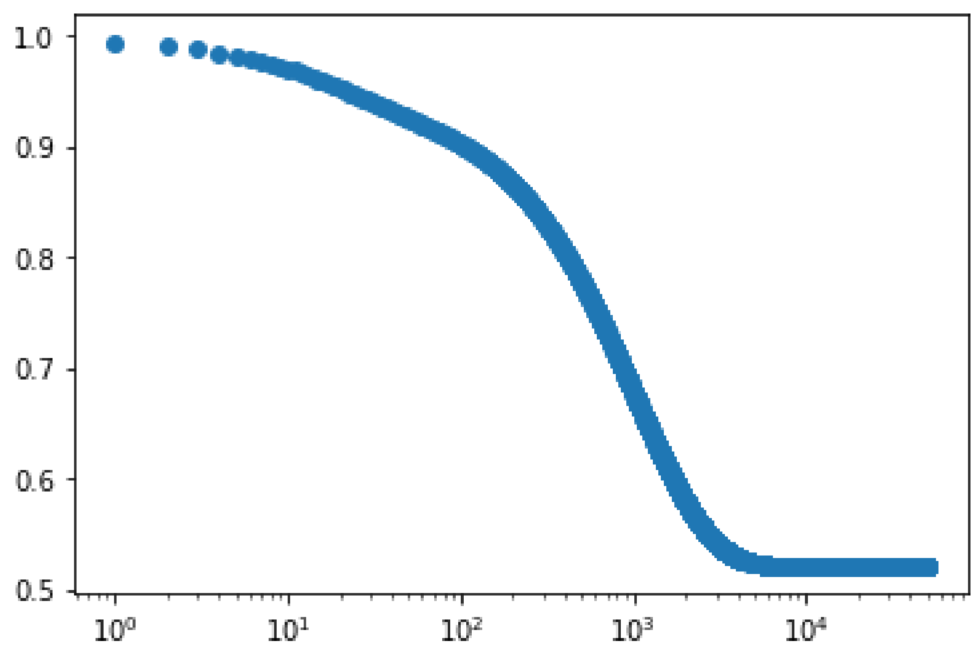
```
    alpha_k = -0.5 * (d_k.T @ d_k) /(d_k.T @ A @ d_k)
    theta_k = theta_k - alpha_k * d_k
    cost_normal.append(np.linalg.norm(y - X @ theta_k)**2/N)
cost_normal = np.array(cost_normal)
theta = theta_k
theta
```

[5]: array([-10.7018, -0.1234,   6.6749])

2.5 2 f)

[17]:
```
plt.figure()
plt.semilogx(cost_normal, 'o')
```

[17]: [<matplotlib.lines.Line2D at 0x7f88e6964700>]



2.6 2 g)

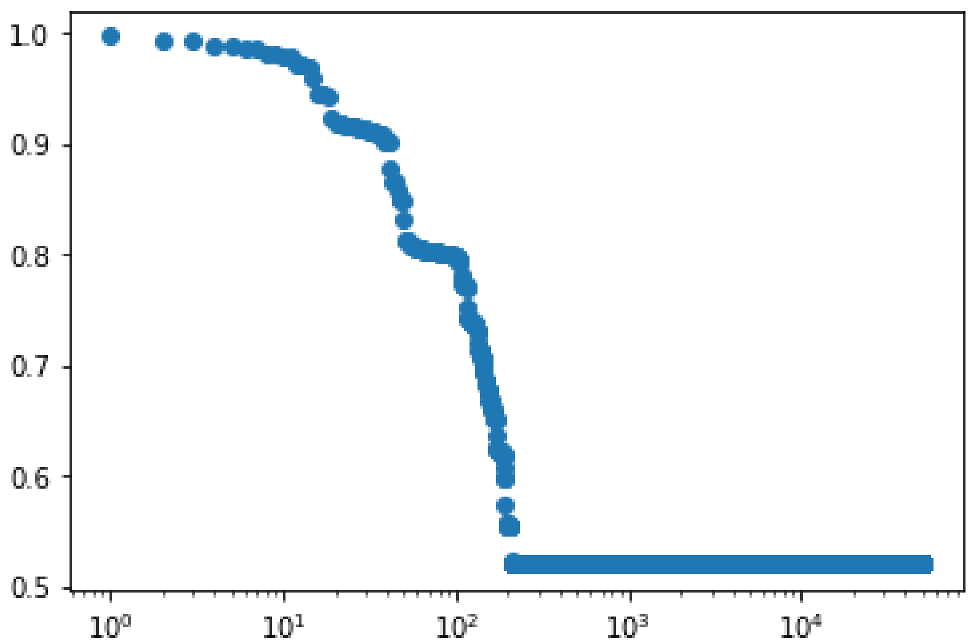
[14]:
```
iterations = 50000
theta0 = np.zeros(3)
A = X.T @ X
b = X.T @ y
beta = 0.9

theta_k = theta0
```

```python
d_k0 = np.zeros(3)
cost_momentum = np.zeros(iterations)
for k in range(iterations):
        d_k = -2 * A @ theta_k + 2 * b
        fuck = beta * d_k0 + (1-beta) * d_k

        alpha_k = -0.5 * (d_k.T @ fuck)/(fuck.T @ A @ fuck)
        theta_k = theta_k - alpha_k * fuck
        d_k0 = d_k
        cost_momentum[k] = np.linalg.norm(y - X @ theta_k)**2/N
print(theta_k)
```

```
[-10.7018  -0.1234   6.6749]
```

[16]:
```python
plt.figure()
plt.semilogx(cost_momentum, 'o')
```

[16]: `[<matplotlib.lines.Line2D at 0x7f88e6a8d490>]`

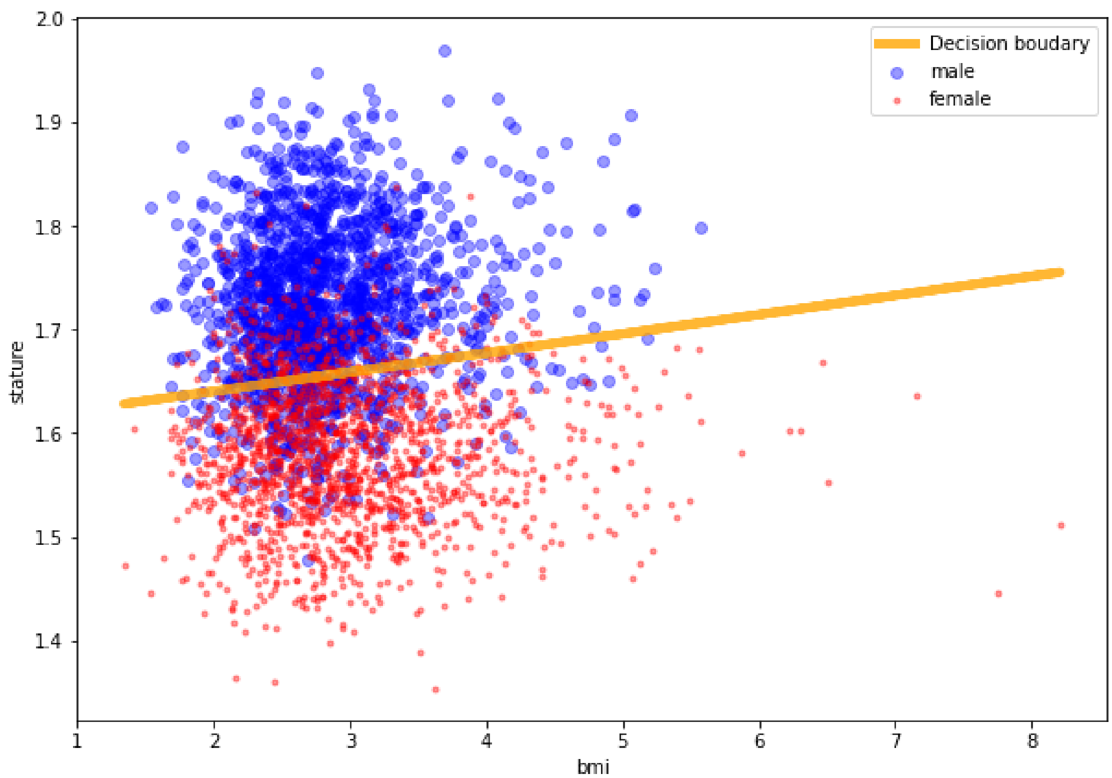


3 Exercise 3

3.1 3 a)

[9]:
```
plt.figure(figsize=(10,7))
# plt.axis('equal')
plt.scatter(m_bmi.values, m_stature.values, color='blue',␣
 ↪label='male',marker='o', alpha=0.4)
plt.scatter(f_bmi.values, f_stature.values, color='red', label='female',␣
 ↪marker='.',alpha=0.4)
plt.xlabel('bmi')
plt.ylabel('stature')
x1 = np.hstack([m_bmi.values, f_bmi.values])
x2 = (-theta[0]-theta[1]*x1)/theta[2]
plt.plot(x1,x2, linewidth='5', color='orange', label='Decision boudary',␣
 ↪alpha=0.8)
plt.legend()
```

[9]: <matplotlib.legend.Legend at 0x7f88e852e9d0>



3.2 3 b)

[10]:
```
data_path = str(Path().absolute())+'/hw2/data/'

male_test_data = pd.read_csv(data_path+'male_test_data.csv')
female_test_data = pd.read_csv(data_path+'female_test_data.csv')
m_bmi_t = male_test_data['male_bmi']*0.1
m_stature_t = male_test_data['male_stature_mm']*0.001
f_bmi_t = female_test_data['female_bmi']*0.1
f_stature_t = female_test_data['female_stature_mm']*0.001

N_male_t = m_bmi_t.values.shape[0]
N_female_t = f_bmi_t.values.shape[0]
N_t = N_male_t + N_female_t

X_test_male = np.block([
    [np.ones(N_male_t)],
    [m_bmi_t.values],
    [m_stature_t.values],
]).T

X_test_female = np.block([
    [np.ones(N_female_t)],
    [f_bmi_t.values],
    [f_stature_t.values],
]).T

label_true_m = np.ones(N_male_t)
label_true_f = -np.ones(N_female_t)

label_f = np.sign(X_test_female @ theta)
label_m = np.sign(X_test_male @ theta)


# 3 b) i)
N_false_male = sum(label_f-label_true_f)/2
print(N_false_male)
type1_err_male = N_false_male/N_female_t
print('Type I error:',type1_err_male)

# 3 b) ii)
N_false_female = -sum(label_m - label_true_m)/2
print(N_false_female)
type2_err_male = N_false_female/N_male_t
print('Type II error:',type2_err_male)

# 3 b) iii)
```

```python
N_model_male = N_false_male + N_male_t - N_false_female
precision = (N_male_t - N_false_female)/N_model_male
print('Precision:',precision)

recall = (N_male_t-N_false_female)/N_male_t
print('Recall:', recall)
```

```
71.0
Type I error: 0.14171656686626746
90.0
Type II error: 0.17964071856287425
Precision: 0.8526970954356846
Recall: 0.8203592814371258
```

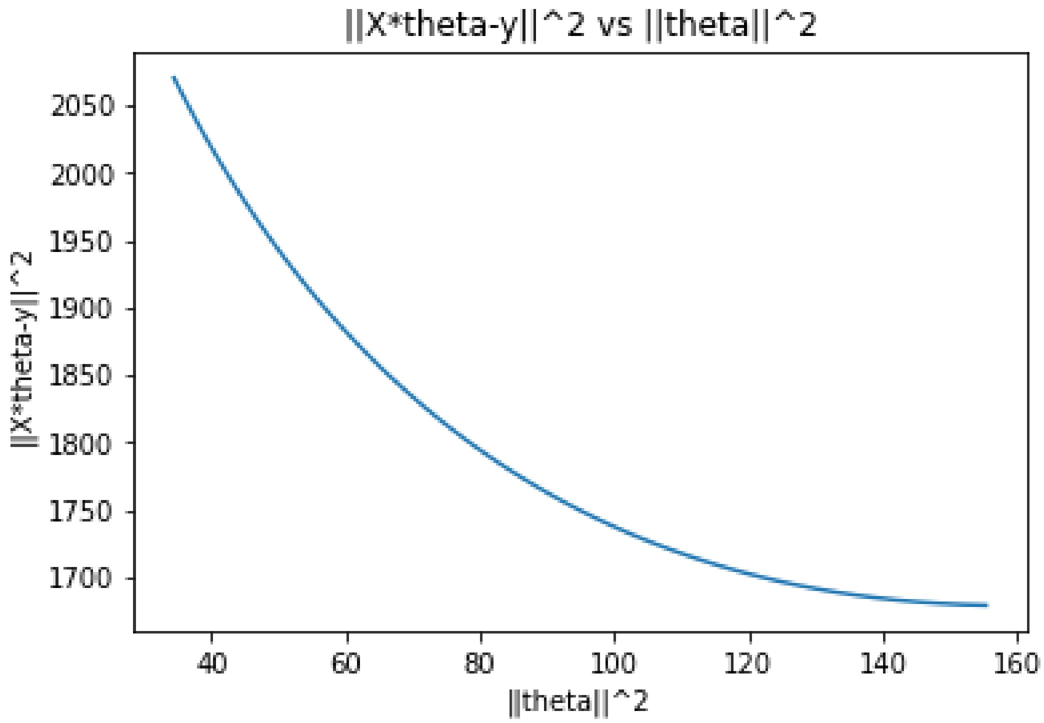
3.3 Exercise 4

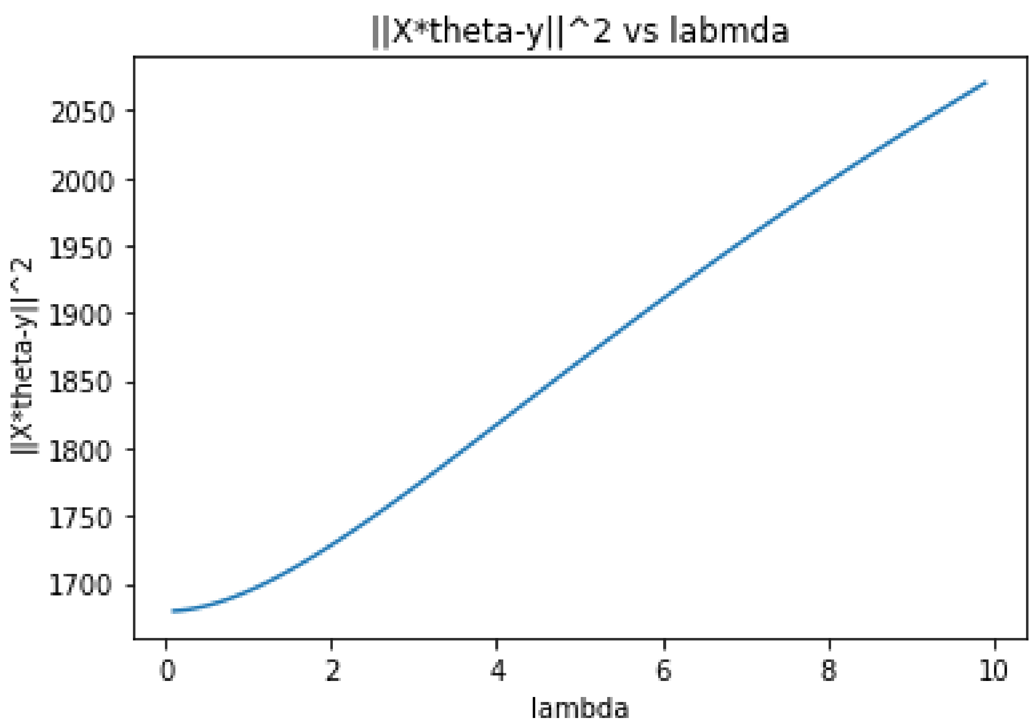
3.4 4 a)

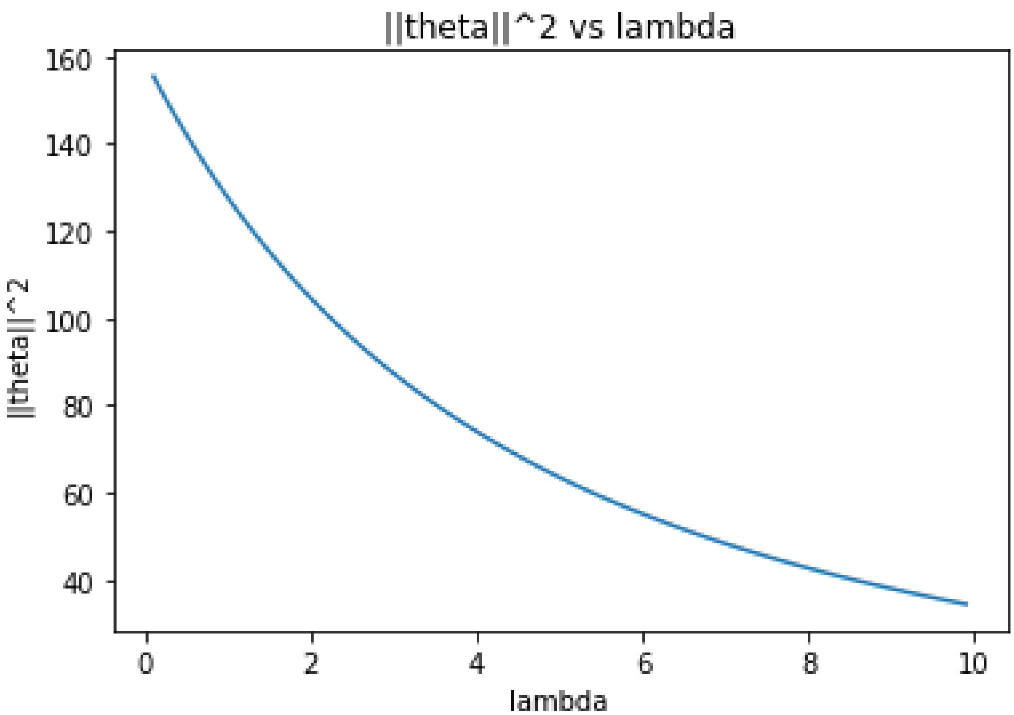
[11]:
```python
lambd = np.arange(0.1, 10, 0.1)

f_theta_l = lambda lamb: np.linalg.inv(X.T @ X + np.eye(3) * lamb) @ X.T @ y
theta_l_vec = [f_theta_l(lamb) for lamb in lambd]
first_term = [np.linalg.norm(X@theta-y)**2 for theta in theta_l_vec]
theta_sqr = [np.linalg.norm(theta)**2 for theta in theta_l_vec]
```

[12]:
```python
plt.figure()
plt.plot(theta_sqr, first_term)
plt.title('||X*theta-y||^2 vs ||theta||^2')
plt.xlabel('||theta||^2')
plt.ylabel('||X*theta-y||^2')

plt.figure()
plt.plot(lambd, first_term)
plt.title('||X*theta-y||^2 vs labmda')
plt.xlabel('lambda')
plt.ylabel('||X*theta-y||^2')

plt.figure()
plt.plot(lambd,theta_sqr)
plt.title('||theta||^2 vs lambda')
plt.xlabel('lambda')
plt.ylabel('||theta||^2')
```

[12]: `Text(0, 0.5, '||theta||^2')`


||X*theta-y||^2 vs ||theta||^2
||X*theta-y||^2
2050
2000
1950
1900
1850
1800
1750
1700
40
60
80
100
120
140
160
||theta||^2


||X*theta-y||^2 vs labmda
||X*theta-y||^2
2050
2000
1950
1900
1850
1800
1750
1700
0
2
4
6
8
10
lambda



[13]:
```
P = 3
var = cp.Variable(P)
objective = cp.Minimize(cp.sum_squares(y-X @ var)+ lambd[0]*cp.sum_squares(var))
constraints = []
prob = cp.Problem(objective=objective, constraints=constraints)
prob.solve()
theta = var.value
np.linalg.norm(theta)**2
```

[13]: 155.44352320635605

[ ]:

[ ]: