[redacted]

תרגיל בית 1

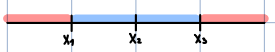
1 (a) נגדיר $f(c)=(c-x_1)^2+(c-x_2)^2+(c-x_3)^2$

נרצה למצוא ערך של c שממזער את $f(c)$, אז נגזור את הביטוי לפי c ונשווה ל-0.

$$f(c)=(c-x_1)^2+(c-x_2)^2+(c-x_3)^2$$
$$=c^2-2cx_1+x_1^2+c^2-2cx_2+x_2^2+c^2-2cx_3+x_3^2$$
$$=3c^2-(2x_1+2x_2+2x_3)c+x_1^2+x_2^2+x_3^2$$
$$f'(c)=6c-(2x_1+2x_2+2x_3)$$
$$f'(c)=0$$
$$6c-(2x_1+2x_2+2x_3)=0$$
$$6c=2x_1+2x_2+2x_3$$
$$3c=x_1+x_2+x_3 \longrightarrow \boxed{c=\frac{x_1+x_2+x_3}{3}}$$

(b) נרצה למצוא ערך של c שממזער את המרחק הגדול ביותר בין c ל-x.

נתון כי $x_1<x_2<x_3$. נבחין כי אם $c>x_3$ אז נקודה x_1 הכי רחוקה מ-c, ובאופן דומה



אם $c<x_1$ אז נקודה x_3 הכי רחוקה מ-c.

אם $x_1<c<x_3$ אז המרחקים קטנים ולכן נרצה ש-c יהיה מתחום זה.

במקרה זה, המרחק המקסימלי יהיה $|c-x_1|$ או $|c-x_3|$. כדי למזער מרחק זה,

נבחר את c להיות נקודת האמצע בין x_1 ל-x_3, כי כל נקודה אחרת שניקח

מהתחום תשנה רק להגדיל את $|c-x_1|$ או $|c-x_3|$.

אז קיבלנו: $\boxed{c=\frac{x_1+x_3}{2}}$.

(c) נרצה למצוא ערך של c שממזער את סכום המרחקים.

באופן דומה לסעיף b, נבחין כי $x_1<c<x_3$. אם כך, הביטוי $|c-x_1|+|c-x_3|$ קבוע לכל c,

אז נרצה שהמרחק $|c-x_2|$ יהיה מינימלי. נפתור את המשוואה $|c-x_2|=0$ ונקבל: $\boxed{c=x_2}$.

(d) בהינתן וקטור עם n משתנים, נוכל לבחור בסעיף a את $A=\begin{bmatrix}1\\1\\\vdots\\1\end{bmatrix}\in\mathbb{R}^{n\times1}$ ואת $b=\begin{bmatrix}x_1\\x_2\\\vdots\\x_n\end{bmatrix}\in\mathbb{R}^n$.

נפתור לפי המשוואה הנורמלית $A^TAc=A^Tb$ ונקבל $c=\frac{x_1+\dots+x_n}{n}$ לכל n.

קיבלנו ש-c הוא הממוצע.

בסעיף c, כפי שהסברנו, c "בחר מהתחום $x_1<c<x_n$ וכדי למזער את סכום המרחקים

נרצה שהמרחק $|c-x_{median}|$ יהיה מינימלי.

נפתור $|c-x_{median}|=0$ ונקבל $c=x_{median}$ לכל n.

2 (a) תהי $A\in\mathbb{R}^{m\times n}$, אז $A^TA\in\mathbb{R}^{n\times n}$. נרצה להראות כי $A^TA=(A^TA)^T$ וגם $x^TA^TAx\geq 0$ לכל $x\in\mathbb{R}^n$.

A^TA סימטרית: $(A^TA)^T = A^T(A^T)^T = A^TA$.

A^TA חיובית חצי-מוגדרת: לפי הגדרה, $x^TA^TAx = (Ax)^TAx = \|Ax\|_2^2 \geq 0$.

(b) תהי $C\in\mathbb{R}^{m\times n}$. נניח כי λ ע"ע של C, אז לפי הגדרה $C\cdot v=\lambda\cdot v$ ($\vec{v}\neq 0$ וקטור עצמי).

כפי שלמדנו, הע"ע של מטריצה מקיימים $\det(C-\lambda I)=0$.

נתבונן ב- $\det((I-C)-(1-\lambda)\cdot I)$. נפתח את הביטוי:

$$\det((I-C)-(1-\lambda)\cdot I)=\det(I-C-I+\lambda I)=\det(\lambda I-C)=\det(C-\lambda I)=0$$

הראנו כי $\det((I-C)-(1-\lambda)\cdot I)=0$ ולכן $1-\lambda$ הוא ע"ע של המטריצה $I-C$.

(c) תהי $A\in\mathbb{R}^{m\times n}$, $m\geq n$. אז $A^TA\in\mathbb{R}^{n\times n}$.

$\Leftarrow$: נניח שהדרגה של A מלאה. אז A מטריצה הפיכה.

יהי $\vec{x}\in\ker A^TA$, אז $A^TAx=0$.

$$A^TAx=0 \xrightarrow{\text{נכפיל את שני האגפים ב-}x^T\text{ משמאל}} x^TA^TAx=0 \xrightarrow{\text{חוקי טרנספוז}} (Ax)^TAx=0 \xrightarrow{\text{הגדרה}} \|Ax\|_2^2=0$$

מכאן ש- $Ax=0$. A הפיכה ולכן הפתרון היחיד למש"ל זו הוא $x=0$ (הפתרון הטריוויאלי).

קיבלנו ש- $\ker(A^TA)=\{0\}$ ולכן לפי הגדרה A^TA הפיכה.

$\Rightarrow$: נניח כי A^TA הפיכה, אז דרגתה מלאה, כלומר $\operatorname{rank}(A^TA)=n$.

בנוסף, לפי הגדרה $\operatorname{rank}(A^TA)\leq\min\{\operatorname{rank}(A),\operatorname{rank}(A^T)\}$.

ידוע כי $\operatorname{rank}(A)=\operatorname{rank}(A^T)$ ונתון $m\geq n$, אז $n=\operatorname{rank}(A^TA)\leq\operatorname{rank}(A)\leq\min\{m,n\}\leq n$.

אז $\operatorname{rank}(A)=n$ ולכן A היא מטריצה שדרגתה מלאה.

(d) תהי $A\in\mathbb{R}^{m\times n}$, $m\geq n$. אז $A^TA\in\mathbb{R}^{n\times n}$.

$\Leftarrow$: נניח שהדרגה של A מלאה. אז לפי סעיף c, A^TA הפיכה ולכן $\ker(A^TA)=\{0\}$.

הוכחנו בסעיף a שלכל מטריצה A, המטריצה A^TA היא סימטרית וחיובית חצי-מוגדרת.

אז מהגדרה, $x^TA^TAx\geq 0$ לכל x.

מהפיכות המטריצה ידוע כי $A^TAx=0$ רק עבור $x=0$.

אז $x^TA^TAx\geq 0$ לכל x וגם לכל $x\neq 0$ מתקיים $A^TAx\neq 0$, מכאן שלכל $x\neq 0$ $x^TA^TAx>0$.

לכן A^TA מטריצה חיובית מוגדרת.

$\Rightarrow$: נניח כי A^TA סימטרית וחיובית מוגדרת.

יהי $v\in\ker(A^TA)$, אז $A^TAv=0$. מהגדרת חיוביות מוגדרת $x^TA^TAx>0$ לכל $x\neq 0$.

מכאן ש- $\vec{v}=0$, אז $\ker(A^TA)=\{0\}$ ולכן A^TA הפיכה.

לפי סעיף c, קיבלנו ש- A מטריצה שדרגתה מלאה.

(e) נניח $\alpha>0$.

המטריצה $\alpha\cdot I$ היא מטריצה סקלרית עם ע"ע יחיד חיובי, α, ולכן היא מטריצה חיובית מוגדרת.

המטריצה A^TA היא מטריצה חיובית חצי-מוגדרת, מסעיף a.

יהי $\vec{x}\neq 0$. אז $x^T\alpha I x>0$ וגם $x^TA^TAx\geq 0$. נרצה להראות: $x^T(A^TA+\alpha I)x>0$.

$x^T(A^TA+\alpha I)x=\overbrace{x^TA^TAx}^{\geq 0}+\overbrace{x^T\alpha I x}^{>0}>0$. אז המטריצה $A^TA+\alpha I$ חיובית מוגדרת.

Question 3:

Code:

```
import numpy as np

A = np.asarray([[2,1,2],[1,-2,1],[1,2,3],[1,1,1]])
b = np.asarray([[6],[1],[5],[2]])
ATA = A.transpose() @ A
ATb = A.transpose() @ b
x = np.linalg.inv(ATA) @ ATb
Ax = A @ x

print("Solution for section a\nVector x:\n", x ,"\n")
print("The best approximation in a least square sense for the system Ax ≈ b
:\n", Ax ,"\n")

p1 = x.transpose() @ A.transpose() @ A @ x
p2 = 2*b.transpose() @ A @ x
p3 = b.transpose() @ b

minimalObjectiveValue = p1 -p2 + p3

print("Solution for section b\nThe minimal objective (loss) value:\n",minim
alObjectiveValue,"\n")

r = A @ x - b

print("The solution x* that we found in the previous section is unique.
Explanation: Matrix A is full rank -> A*A is invertible -
> Normal equations have a single solution.\n")
print("Solution for section c\nThe residual of the least squares system:\n"
,r,"\n")

ATr = A.transpose() @ r

print("Showing that ATr = 0:\n",ATr,"\n")
print("This is not surprising. Explanation: A*r = A*(Ax-b) = A*Ax-
A*b = 0.\n")

W = np.asarray([[1,0,0,0],[0,1000,0,0],[0,0,1,0],[0,0,0,1]])
ATWA = A.transpose() @ W @ A
ATWb = A.transpose() @ W @ b
x = np.linalg.inv(ATWA) @ ATWb
r = A @ x - b

print("Solution for section d\nWeighted least squares solution:\n",r,"\n")
print("As requested, r2 =", r[1][0], "< 10^-3.\n")

x = np.linalg.inv(ATA + 0.5*np.eye(3)) @ ATb

print("Solution for section e\nLeast squares solution with simple Tikhonov
regularization:\n",x,"\n")
```

Output:

Solution for section a

Vector x:

[[1.7]

[0.6]

[0.7]]

The best approximation in a least square sense for the system Ax ≈ b:

[[5.4]

[1.2]

[5.]

[3.]]

Solution for section b

The minimal objective (loss) value:

[[1.4]]

The solution x∗ that we found in the previous section is unique. Explanation: Matrix A is full rank -> A*A is invertible -> Normal equations have a single solution.

Solution for section c

The residual of the least squares system:

[[-0.6]

[0.2]

[0.]

[1.]]

Showing that A⊤r = 0:

[[1.24344979e-14]

[0.00000000e+00]

[1.24344979e-14]]

This is not surprising. Explanation: A*r = A*(Ax-b) = A*Ax-A*b = 0.

Solution for section d

Weighted least squares solution:

[[-6.17628893e-01]

[2.05876298e-04]

[-9.94759830e-14]

[1.02938149e+00]]

As requested, r2 = 0.0002058762978407458 < 10^-3.

Solution for section e

Least squares solution with simple Tikhonov regularization:

[[1.38517367]

[0.53499222]

[0.88958009]]

Question 4:

Solution for section a

$$\underset{X\in\mathbb{R}^{n\times n}}{\arg\min}\|AX-B\|_F^2 = \underset{X\in\mathbb{R}^{n\times n}}{\arg\min}\sum_j\|(AX-B)_j\|_2^2 = \underset{X\in\mathbb{R}^{n\times n}}{\arg\min}\sum_j\|Ax_j-b_j\|_2^2$$

$$= \underset{X\in\mathbb{R}^{n\times n}}{\arg\min}\left(\|Ax_1-b_1\|_2^2+\|Ax_2-b_2\|_2^2+\dots+\|Ax_n-b_n\|_2^2\right)$$

$$[\nabla f(x)]_j = 2A^TAx_j - 2A^Tb_j = 0$$

$$A^TAx_j - A^Tb_j = 0$$

$$A^TAx_j = A^Tb_j$$

(עמודה במטריצה X) $x_j = (A^TA)^{-1}A^Tb_j \implies X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \dots & x_n \\ | & | & & | \end{bmatrix}$

נשים לב שהסכום $\|Ax_1-b_1\|_2^2+\|Ax_2-b_2\|_2^2+\dots+\|Ax_n-b_n\|_2^2$ מינימלי כאשר כל אחד מהמחוברים בו מינימלי. נוכל למצוא כל אחד מהמחוברים $\|Ax_j-b_j\|_2^2$ באמצעות ריבועים פחותים.
נקבל פתרון יחיד רק אם עבור כל x_j יש פתרון יחיד.
לכן, בדומה לריבועים פחותים, זה יקרה כאשר A הפיכה.

Solution for section b

Code:

```
A = np.asarray([[5,6,7,8],[1,3,5,4],[1,0.5,4,2],[3,4,3,1]])
B = np.asarray([[0.57,0.56,0.8,1],[1.5,4,6.7,4.9],[0.2,0.1,1,0.6],[11,30,26
,10]])
D = np.zeros((4,4))

for i in range(4):
  ata = A[i].transpose() @ A[i];
  atb = A[i].transpose() @ B[i];
  D[i][i] = atb/ata

print("Solution D for the given matrices A and B:\n",D,"\n")
```

Output:

Solution D for the given matrices A and B:

```
[[0.11385057 0.         0.         0.        ]
 [0.         1.30588235 0.         0.        ]
 [0.         0.         0.25647059 0.        ]
 [0.         0.         0.         6.88571429]]
```

Question 5:

Code:

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import OneHotEncoder
from sklearn.metrics import mean_squared_error

#section a
df = pd.read_csv('/content/insurData.csv')

#section b
df['variable'] = 1
encoder = OneHotEncoder(handle_unknown = 'ignore')

encoder_df_sex = pd.DataFrame(encoder.fit_transform(df[['sex']]).toarray())
encoder_df_smoker = pd.DataFrame(encoder.fit_transform(df[['smoker']]).toar
ray())
encoder_df_region = pd.DataFrame(encoder.fit_transform(df[['region']]).toar
ray())

df_age = df['age']
df_bmi = df['bmi']
df_children = df['children']
df_charges = df['charges'].multiply(1/1000)
df_variable = df['variable']

df = pd.concat([df_variable,df_age,df_bmi,df_children,encoder_df_sex,encode
r_df_smoker,encoder_df_region,df_charges],axis=1)

df.columns = ['variable','age','bmi','children','female','male','non-
smoker','smoker','northeast','northwest','southeast','southwest','charges']

#section c
for i in range(5):
  final_df = df.sample(frac=1) # shuffle
  train_df = final_df.iloc[0:1070] # 80%
  test_df = final_df.iloc[1070:1339] # 20%

  #train
  X_train = train_df[['variable','age','bmi','children','female','male','no
n-
smoker','smoker','northeast','northwest','southeast','southwest']].to_numpy
()
  y_train = train_df[['charges']].to_numpy()

  XTX = X_train.transpose() @ X_train
  XTy = X_train.transpose() @ y_train
  XTX_inv = np.linalg.inv(XTX)
  a = np.array(np.linalg.lstsq(X_train,y_train,rcond=1))[0]

  Xa = X_train @ a
  MseTrain = mean_squared_error(y_train, Xa)

  #test
  test_X = test_df[['variable','age','bmi','children','female','male','non-
smoker','smoker','northeast','northwest','southeast','southwest']].to_numpy
()
  test_y = test_df[['charges']].to_numpy()
```

```
  test_Xa = test_X @ a
  MseTest = mean_squared_error(test_y, test_Xa)

  #comparing by finding ratio
  ratio = MseTest / MseTrain
  print("Experiment" ,i+1, "results:")
  print("MSE for test:", MseTest)
  print("MSE for train:", MseTrain)
  print("The ratio between train and test is:", ratio)

#section d
plt.hist(np.abs(Xa - y_train), bins=200)
plt.title("Distribution of error values")
plt.ylabel("Frequency")
plt.xlabel("Error values")
plt.legend("Error value")
plt.show()
```

Output:

Solution for section c

Experiment 1 results:
MSE for test: 39.495878621292164
MSE for train: 35.855147832702244
The ratio between train and test is: 1.101539974275865

Experiment 2 results:
MSE for test: 38.30470574312318
MSE for train: 36.09005935972206
The ratio between train and test is: 1.0613644428047893

Experiment 3 results:
MSE for test: 41.55690789877068
MSE for train: 35.351751536003185
The ratio between train and test is: 1.1755261364192378

Experiment 4 results:
MSE for test: 34.48779750652155
MSE for train: 37.03378877200032
The ratio between train and test is: 0.9312522064336101

Experiment 5 results:
MSE for test: 34.753019045753106
MSE for train: 36.991137012365826
The ratio between train and test is: 0.9394958320458078

As shown, the ratio between train and test is ~1 in all the experiments, which means that the model predicts the charges well.

Solution for section d