without bias

$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} w_1 = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \rightarrow \begin{cases} x_1 w_1 = y_1 \\ x_2 w_1 = y_2 \end{cases} \rightarrow f(x) = w_1 x$

with bias

$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \rightarrow \begin{cases} w_0 + x_1 w_1 = y_1 \\ w_0 + x_2 w_1 = y_2 \end{cases} \rightarrow f(x) = w_0 + w_1 x$

EE2211 Tutorial 5

(Linear Regression, bias/offset)

Question 1:

Given the following data pairs for training:

$\{x = -10\} \rightarrow \{y = 5\}$
$\{x = -8\} \rightarrow \{y = 5\}$
$\{x = -3\} \rightarrow \{y = 4\}$
$\{x = -1\} \rightarrow \{y = 3\}$
$\{x = 2\} \rightarrow \{y = 2\}$
$\{x = 8\} \rightarrow \{y = 2\}$

a)
$$\underset{X}{\begin{bmatrix} 1 & -10 \\ 1 & -8 \\ 1 & -3 \\ 1 & -1 \\ 1 & 2 \\ 1 & 8 \end{bmatrix}} \underset{w}{\begin{bmatrix} w_0 \\ w_1 \end{bmatrix}} = \underset{Y}{\begin{bmatrix} 5 \\ 5 \\ 4 \\ 3 \\ 2 \\ 2 \end{bmatrix}}$$

b)
$$\underset{X}{\begin{bmatrix} -10 \\ -8 \\ -3 \\ -1 \\ 2 \\ 8 \end{bmatrix}} \underset{w}{\begin{bmatrix} w_1 \end{bmatrix}} = \underset{Y}{\begin{bmatrix} 5 \\ 5 \\ 4 \\ 3 \\ 2 \\ 2 \end{bmatrix}}$$

$$w = (X^T X)^{-1} X^T y$$
(left inverse)

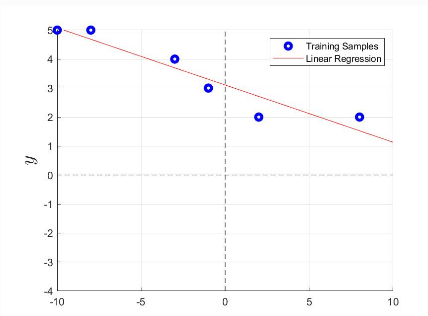
(a) Perform a linear regression with addition of a bias/offset term to the input feature vector and sketch the result of line fitting.
(b) Perform a linear regression without inclusion of any bias/offset term and sketch the result of line fitting.
(c) What is the effect of adding a bias/offset term to the input feature vector?

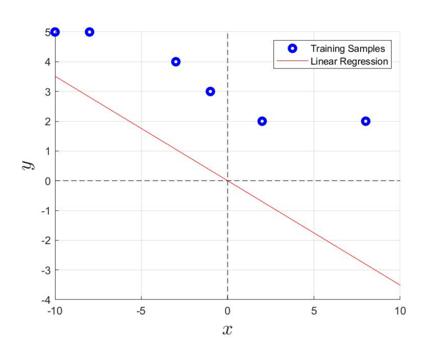
Answer:

(a) This is an over-determined system.

The input feature including bias/offset can be written as $\mathbf{X}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -10 & -8 & -3 & -1 & 2 & 8 \end{bmatrix}$.

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 6 & -12 \\ -12 & 242 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ -10 & -8 & -3 & -1 & 2 & 8 \end{bmatrix} \begin{bmatrix} 5 \\ 5 \\ 4 \\ 3 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 3.1055 \\ -0.1972 \end{bmatrix}.$$





(b) This is an over-determined system.

In this case, the input feature without inclusion of bias/offset is a vector given by $[-10, -8, -3, -1, 2, 8]^T$.

$$\widehat{\mathrm{w}} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y} = [242]^{-1}[-10, -8, -3, -1, 2, 8]\begin{bmatrix}5\\5\\4\\3\\2\\2\end{bmatrix} = -0.3512.$$

(c) The bias/offset term allows the line to move away from the origin (moved vertically in this case).

(Linear Regression, prediction, even/under-determined)

Question 2:

Given the following data pairs for training:

$\{x_1 = 1,\ x_2 = 0,\quad x_3 = 1\} \rightarrow \{y = 1\}$

$\{x_1 = 2,\ x_2 = -1, x_3 = 1\} \rightarrow \{y = 2\}$

$\{x_1 = 1,\ x_2 = 1,\ x_3 = 5\} \rightarrow \{y = 3\}$

(a) Predict the following test data without inclusion of an input bias/offset term.

(b) Predict the following test data with inclusion of an input bias/offset term.

↳ right inverse : $X^T\ (XX^T)\ y$

$\{x_1 = -1,\ x_2 = 2,\ x_3 = 8\} \rightarrow \{y = ?\}$
$\{x_1 = 1,\ x_2 = 5,\ x_3 = -1\} \rightarrow \{y = ?\}$

Answer:

(a) Without bias, this is an even-determined system and **X** is invertible.

$$\widehat{\mathbf{w}} = \mathbf{X}^{-1}\mathbf{y} = \begin{bmatrix}1 & 0 & 1\\2 & -1 & 1\\1 & 1 & 5\end{bmatrix}^{-1}\begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}0.3333\\-0.6667\\0.6667\end{bmatrix}$$

$$\hat{\mathbf{y}}_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}}\widehat{\mathbf{w}} = \begin{bmatrix}-1 & 2 & 8\\1 & 5 & -1\end{bmatrix}\begin{bmatrix}0.3333\\-0.6667\\0.6667\end{bmatrix} = \begin{bmatrix}3.6667\\-3.6667\end{bmatrix}$$

(b) After adding bias, it becomes an under-determined system.

$$\widehat{\mathbf{w}} = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{y} = \begin{bmatrix}1 & 1 & 1\\1 & 2 & 1\\0 & -1 & 1\\1 & 1 & 5\end{bmatrix}\begin{bmatrix}3 & 4 & 7\\4 & 7 & 7\\7 & 7 & 28\end{bmatrix}^{-1}\begin{bmatrix}1\\2\\3\end{bmatrix} = \begin{bmatrix}-0.1429\\0.5238\\-0.4762\\0.6190\end{bmatrix}$$

$$\hat{\mathbf{y}}_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}}\widehat{\mathbf{w}} = \begin{bmatrix}1 & -1 & 2 & 8\\1 & 1 & 5 & -1\end{bmatrix}\begin{bmatrix}-0.1429\\0.5238\\-0.4762\\0.6190\end{bmatrix} = \begin{bmatrix}3.3333\\-2.6190\end{bmatrix}$$

(Linear Regression, prediction, extrapolation)
Question 3:

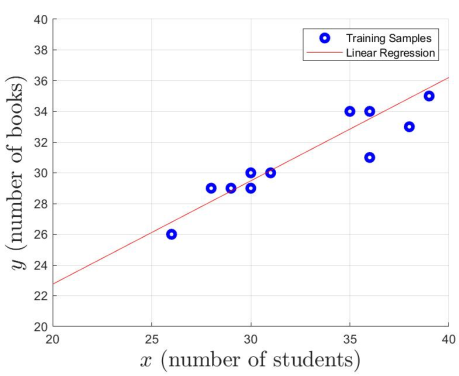
A college bookstore must order books two months before each semester starts. They believe that the number of books that will ultimately be sold for any particular course is related to the number of students registered for the course when the books are ordered. They would like to develop a linear regression equation to help plan how many books to order. From past records, the bookstore obtains the number of students registered, X, and the number of books actually sold for a course, Y, for 12 different semesters. These data are shown below.

Semester	Students	Books
1	36	31
2	28	29
3	35	34
4	39	35
5	30	29
6	30	30
7	31	30
8	38	38
9	36	34
10	38	33
11	29	29
12	26	26

(a) Obtain a scatter plot of the number of books sold versus the number of registered students.
(b) Write down the regression equation and calculate the coefficients for this fitting.
(c) Predict the number of books that would be sold in a semester when 30 students have registered.
(d) Predict the number of books that would be sold in a semester when 5 students have registered.

Answer:

(a)



(b) Regression equation: $\mathbf{y} = \mathbf{X}\mathbf{w}$,

where $\mathbf{w} = [\mathrm{w}_0, \mathrm{w}_1]^T$, $\mathbf{X}^T = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 36 & 28 & 35 & 39 & 30 & 30 & 31 & 38 & 36 & 38 & 29 & 26 \end{bmatrix}$,

$\mathbf{y}^T = [31, 29, 34, 35, 29, 30, 30, 38, 34, 33, 29, 26]$.

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 12 & 396 \\ 396 & 13288 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 36 & 28 & 35 & 39 & 30 & 30 & 31 & 38 & 36 & 38 & 29 & 26 \end{bmatrix} \begin{bmatrix} 31 \\ 29 \\ 34 \\ 35 \\ 29 \\ 30 \\ 30 \\ 38 \\ 34 \\ 33 \\ 29 \\ 26 \end{bmatrix}$$

$$= \begin{bmatrix} 9.30 \\ 0.6727 \end{bmatrix}$$

(c)

$$\hat{y}_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}}\hat{\mathbf{w}} = [1 \ \ 30] \begin{bmatrix} 9.30 \\ 0.6727 \end{bmatrix} = 29.4818$$

(d) ($\hat{y}_{\mathrm{t}} = 12.6636$) This prediction appears to be somewhat over optimistic. Since 5 students is not within the range of the sampled number of students, it might not be appropriate to use the regression equation to make this prediction. We do not know if the straight-line model would fit data at this point, and we might not want to extrapolate far beyond the observed range.

(Linear Regression, prediction, impact of duplicated entries)
Question 4:

Repeat the above problem using the following training data:

Semester	Students	Books
1	36	31
2	26	20
3	35	34
4	39	35
5	26	20
6	30	30
7	31	30
8	38	38
9	36	34
10	38	33
11	26	20
12	26	20

(a) Calculate the regression coefficients for this fitting.
(b) Predict the number of books that would be sold in a semester when 30 students have registered.
(c) Purge those duplicating data and re-fit the line and observe the impact on predicting the number of books that would be sold in a semester when 30 students have registered.
(d) Sketch and compare the two fitting lines.

Answer:

(a),(b),(c),(d)

Using the full data:

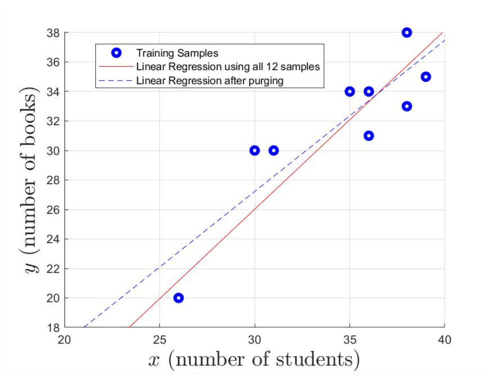
$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 12 & 387 \\ 387 & 12791 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 36 & 26 & 35 & 39 & 26 & 30 & 31 & 38 & 36 & 38 & 26 & 26 \end{bmatrix} \begin{bmatrix} 31 \\ 20 \\ 34 \\ 35 \\ 20 \\ 30 \\ 30 \\ 38 \\ 34 \\ 33 \\ 20 \\ 20 \end{bmatrix} = \begin{bmatrix} -10.4126 \\ 1.2143 \end{bmatrix}$$

$$\hat{y}_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}}\hat{\mathbf{w}} = [1 \;\; 30] \begin{bmatrix} -10.4126 \\ 1.2143 \end{bmatrix} = 26.0177$$

After having the duplicating data purged:

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} = \begin{bmatrix} 9 & 309 \\ 309 & 10763 \end{bmatrix}^{-1} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 36 & 35 & 39 & 30 & 31 & 38 & 36 & 38 & 26 \end{bmatrix} \begin{bmatrix} 31 \\ 34 \\ 35 \\ 30 \\ 30 \\ 38 \\ 34 \\ 33 \\ 20 \end{bmatrix} = \begin{bmatrix} -3.5584 \\ 1.0260 \end{bmatrix}$$

$$\hat{y}_{\mathrm{t}} = \mathbf{X}_{\mathrm{t}}\hat{\mathbf{w}} = [1 \;\; 30] \begin{bmatrix} -3.5584 \\ 1.0260 \end{bmatrix} = 27.2208$$



Note: these results show that duplicating samples can influence the learning and decision too. In this case, purging seems to give a more optimistic prediction for a relatively small number of students (< 37) and more conservative prediction for a relatively large number of students (>37).

(Linear Regression, python)

Question 5:

Download the data file “government-expenditure-on-education.csv” from Canvas Tutorial Folder.
It depicts the government’s educational expenditure over the years (downloaded in July 2021 from https://data.gov.sg/dataset/government-expenditure-on-education)
Predict the educational expenditure of year 2021 based on linear regression. Solve the problem using Python with a plot. Note: please use the file from the canvas link.
Hint: use Python packages like numpy, pandas, matplotlib.pyplot, numpy.linalg.

Answer:

The predicted educational expenditure in year 2021 is 12102904.270643068.

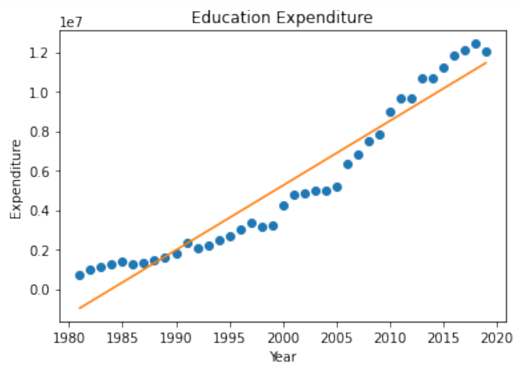
Codes:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from numpy.linalg import inv

df = pd.read_csv("government-expenditure-on-education.csv")

expenditureList = df ['recurrent_expenditure_total'].tolist()

yearList = df ['year'].tolist()

m_list = [[1]*len(yearList), yearList]

X = np.array(m_list).T

y = np.array(expenditureList)

w = inv(X.T @ X) @ X.T @ y

print(w)

y_line = X.dot(w)

plt.plot(yearList, expenditureList, 'o', label = 'Expenditure over the years')

plt.plot(yearList, y_line)

plt.xlabel('Year')

plt.ylabel('Expenditure')
```

```
plt.title('Education Expenditure')

plt.show()

y_predict = np.array([1, 2021]).dot(w)

print(y_predict)
```

Output

```
[-6.4843247e+08  3.2683591e+05]
12102904.270643068
```



(Linear Regression, python)

Question 6:

Download the CSV file for red-wine using " wine = pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv",sep=';') " . Use Python to perform the following tasks. Hint: use Python packages like numpy, pandas, matplotlib.pyplot, numpy.linalg, and sklearn.metrics.

(a) Take `y = wine.quality` as the target output and `x = wine.drop('quality',axis = 1)` as the input features. Assume the given list of data is already randomly indexed (i.e., not in particular order), split the database into two sets: [0:1500] samples for regression training, and [1500:1599] samples for testing.

(b) Perform linear regression on the training set and print out the learned parameters.

(c) Perform prediction using the test set and provide the prediction accuracy in terms of the mean of squared errors (MSE).

Answer:

```
import pandas as pd

#import matplotlib.pyplot as plt

import numpy as np

from numpy.linalg import inv
```

```
from sklearn.metrics import mean_squared_error

## get data from web

wine   =   pd.read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv",sep=';')

wine.info()

y = wine.quality

x = wine.drop('quality',axis = 1)

## Include the offset/bias term

x0 = np.ones((len(y),1))

X = np.hstack((x0,x))

## split data into training and test sets

##  (Note:  this  exercise  introduces  the  basic  protocol  of  using  the  training-test partitioning of samples for evaluation assuming the list of data is already randomly indexed)

##  In  case  you  really  want  a  general  random  split  to  have  a  better  training/test distributions:

## from sklearn.model_selection import train_test_split

## train_X,test_X,train_y,test_y = train_test_split(X,y,test_size=99/1599, random_state = 0)

train_X = X[0:1500]

train_y = y[0:1500]

test_X = X[1500:1599]

test_y = y[1500:1599]

## linear regression

w = inv(train_X.T @ train_X) @ train_X.T @ train_y

print(w)

yt_est = test_X.dot(w);

MSE = np.square(np.subtract(test_y,yt_est)).mean()

print(MSE)

MSE = mean_squared_error(test_y,yt_est)
```

```
print(MSE)
```

[2.22330327e+01   2.68702621e-02 -1.12838019e+00 -2.06141685e-01

    1.22000584e-02 -1.77718503e+00   4.29357454e-03 -3.18953315e-03

  -1.81795124e+01 -3.98142390e-01   8.92474793e-01   2.77147239e-01]

0.34352638122440293

0.343526381224403

Question 7:
This question is related to understanding of modelling assumptions. The function given by $f(\mathbf{x}) = 1 + x_1 + x_2 - x_3 - x_4$ is affine.
a) True
b) False

Answer: a)

Question 8:

MCQ: There could be more than one answer.
Suppose $f(\mathbf{x})$ is a *scalar* function of d variables where $\mathbf{x}$ is a $d \times 1$ vector. Then, without taking data points into consideration, the outcome of differentiation of $f(\mathbf{x})$ w.r.t. $\mathbf{x}$ is
a) a scalar
b) a d×1 vector
c) a d×d matrix
d) a d×d×d tensor
e) None of the above

Answer: b)

(Linear regression with multiple outputs)

Questions 9:

The values of feature vector **x** and their corresponding values of target vector **y** are shown in the table below:

x	[3, -1, 0]	[5, 1, 2]	[9, -1, 3]	[-6, 7, 2]	[3, -2, 0]
y	[1, -1]	[-1, 0]	[1, 2]	[0, 3]	[1, -2]

Find the least square solution of **w** using linear regression of multiple outputs and then estimate the value of **y** when ***x*** = [8, 0, 2].

Answer:

```python
#python
import numpy as np
from numpy.linalg import inv
X = np.array([[1, 3, -1, 0], [1, 5, 1, 2], [1, 9, -1, 3], [1, -6, 7, 2],
[1, 3, -2, 0]])
Y = np.array([[1, -1], [-1, 0], [1, 2], [0, 3], [1, -2]])
W = inv(X.T @ X) @ X.T @ Y
print(W)

newX=np.array([1, 8, 0, 2])
newY=newX@W
print(newY)
```

Outputs

```
W= [[ 1.14668974 -0.95997404]
 [-0.630463   -0.33427088]
 [-1.10601471 -0.24426655]
 [ 1.3595846   1.77953267]]
newY=[-1.17784509 -0.07507572]
```