The Chinese University of Hong Kong, Shenzhen

DDA 3020

Machine Learning

Assignment1 Report

Author:
Zhao Rui

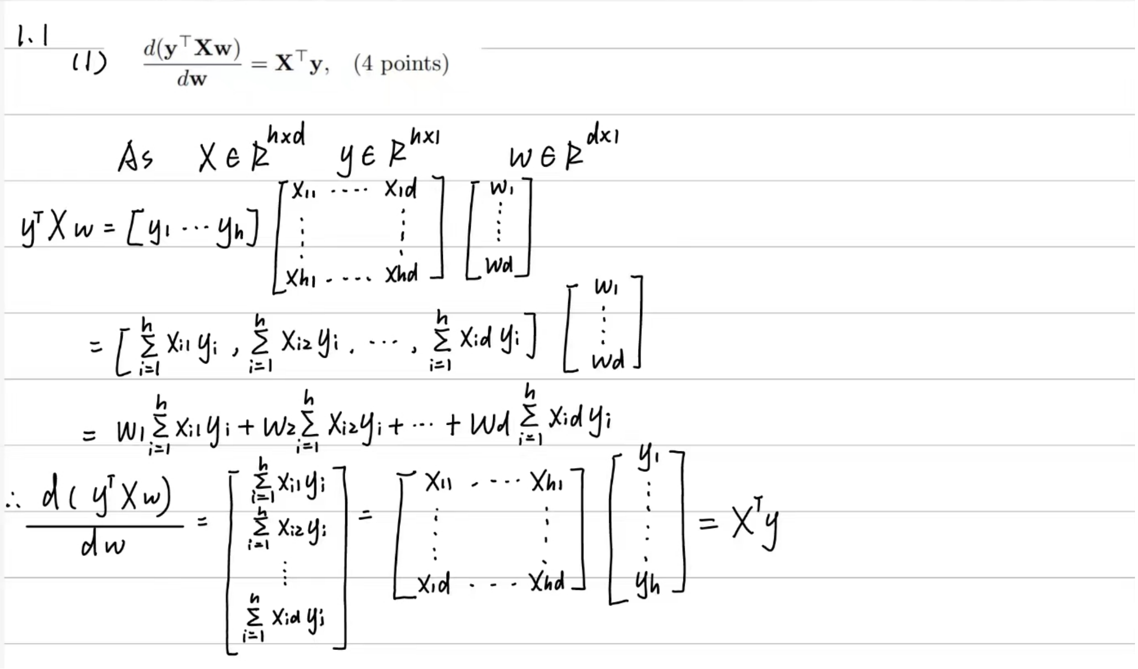
Student Number:
121090820

March 12, 2023

Contents

1 Written Questions

1.1 Question 1.1



1.1

(1) $\frac{d(\mathbf{y}^\top \mathbf{X}\mathbf{w})}{d\mathbf{w}} = \mathbf{X}^\top \mathbf{y}$, (4 points)

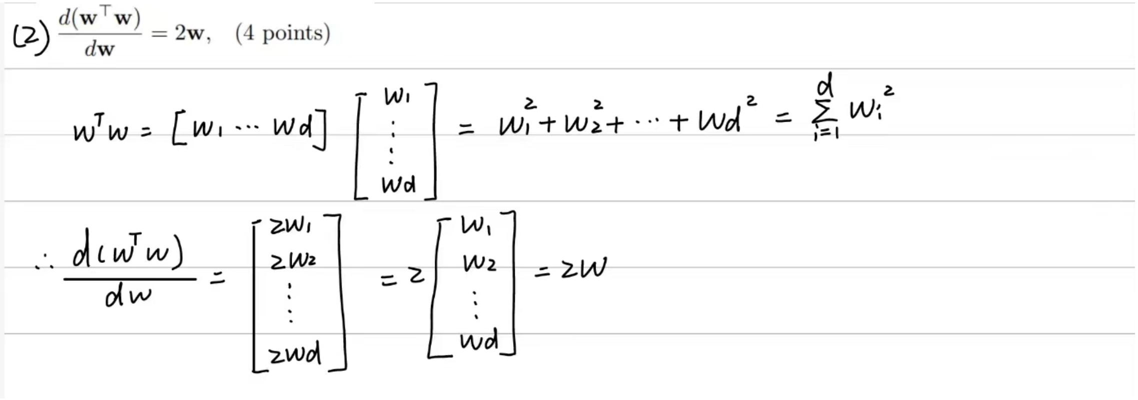
As $X \in R^{h\times d}$ $y \in R^{h\times 1}$ $w \in R^{d\times 1}$

$$y^T X w = [y_1 \cdots y_h] \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & & \vdots \\ x_{h1} & \cdots & x_{hd} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$= \left[\sum_{i=1}^{h} x_{i1} y_i , \sum_{i=1}^{h} x_{i2} y_i , \cdots , \sum_{i=1}^{h} x_{id} y_i \right] \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$= w_1 \sum_{i=1}^{h} x_{i1} y_i + w_2 \sum_{i=1}^{h} x_{i2} y_i + \cdots + w_d \sum_{i=1}^{h} x_{id} y_i$$

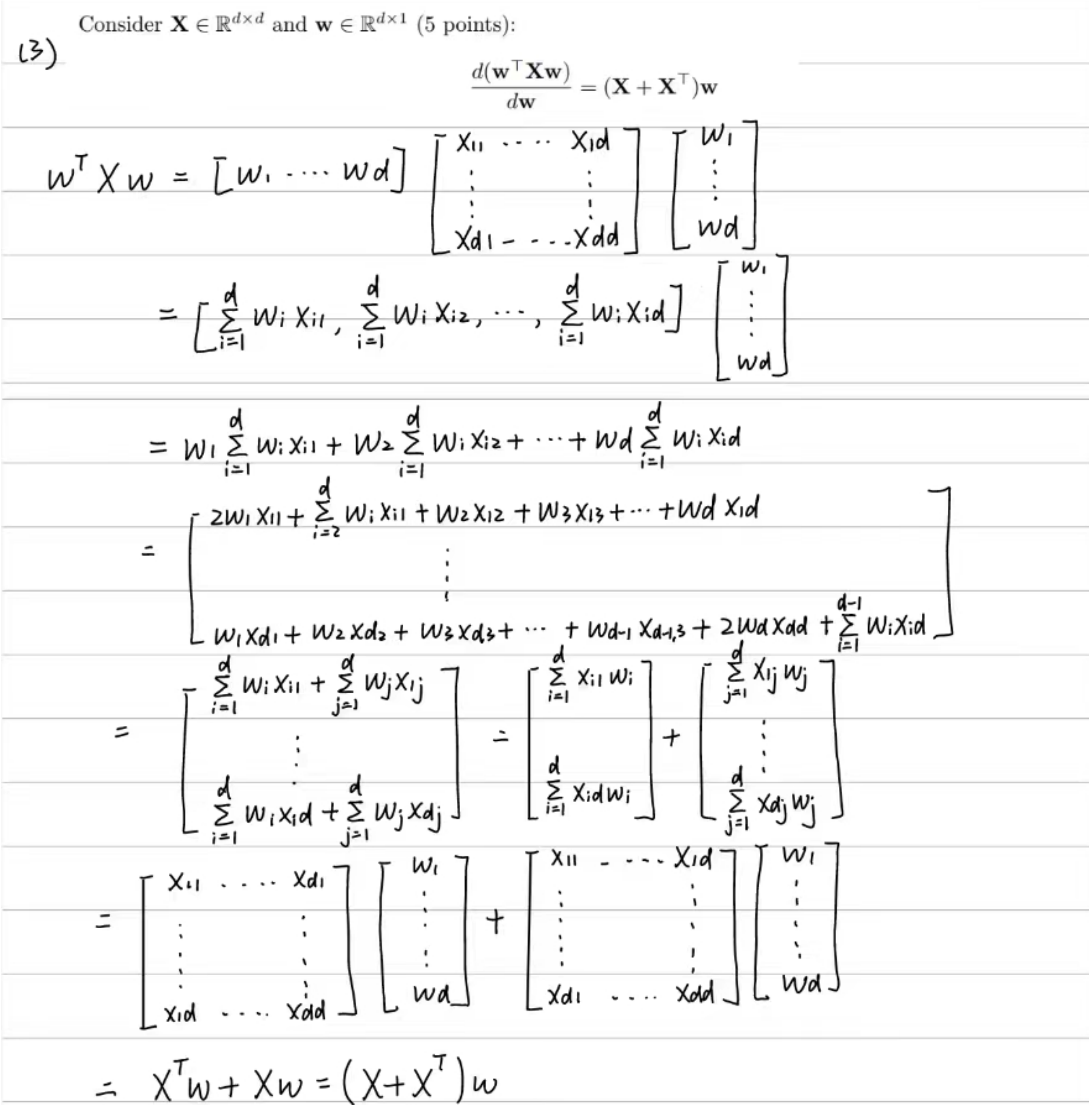
$$\therefore \frac{d(y^T X w)}{dw} = \begin{bmatrix} \sum_{i=1}^{h} x_{i1} y_i \\ \sum_{i=1}^{h} x_{i2} y_i \\ \vdots \\ \sum_{i=1}^{h} x_{id} y_i \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{h1} \\ \vdots & & \vdots \\ x_{1d} & \cdots & x_{hd} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_h \end{bmatrix} = X^T y$$



(2) $\frac{d(\mathbf{w}^\top \mathbf{w})}{d\mathbf{w}} = 2\mathbf{w}$, (4 points)

$$w^T w = [w_1 \cdots w_d] \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} = w_1^2 + w_2^2 + \cdots + w_d^2 = \sum_{i=1}^{d} w_i^2$$

$$\therefore \frac{d(w^T w)}{dw} = \begin{bmatrix} 2w_1 \\ 2w_2 \\ \vdots \\ 2w_d \end{bmatrix} = 2 \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} = 2w$$



(3)

Consider $\mathbf{X} \in \mathbb{R}^{d \times d}$ and $\mathbf{w} \in \mathbb{R}^{d \times 1}$ (5 points):

$$\frac{d(\mathbf{w}^\top \mathbf{X}\mathbf{w})}{d\mathbf{w}} = (\mathbf{X} + \mathbf{X}^\top)\mathbf{w}$$

$$w^T X w = \begin{bmatrix} w_1 & \cdots & w_d \end{bmatrix} \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & & \vdots \\ x_{d1} & \cdots & x_{dd} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{d} w_i x_{i1}, & \sum_{i=1}^{d} w_i x_{i2}, & \cdots, & \sum_{i=1}^{d} w_i x_{id} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$= w_1 \sum_{i=1}^{d} w_i x_{i1} + w_2 \sum_{i=1}^{d} w_i x_{i2} + \cdots + w_d \sum_{i=1}^{d} w_i x_{id}$$

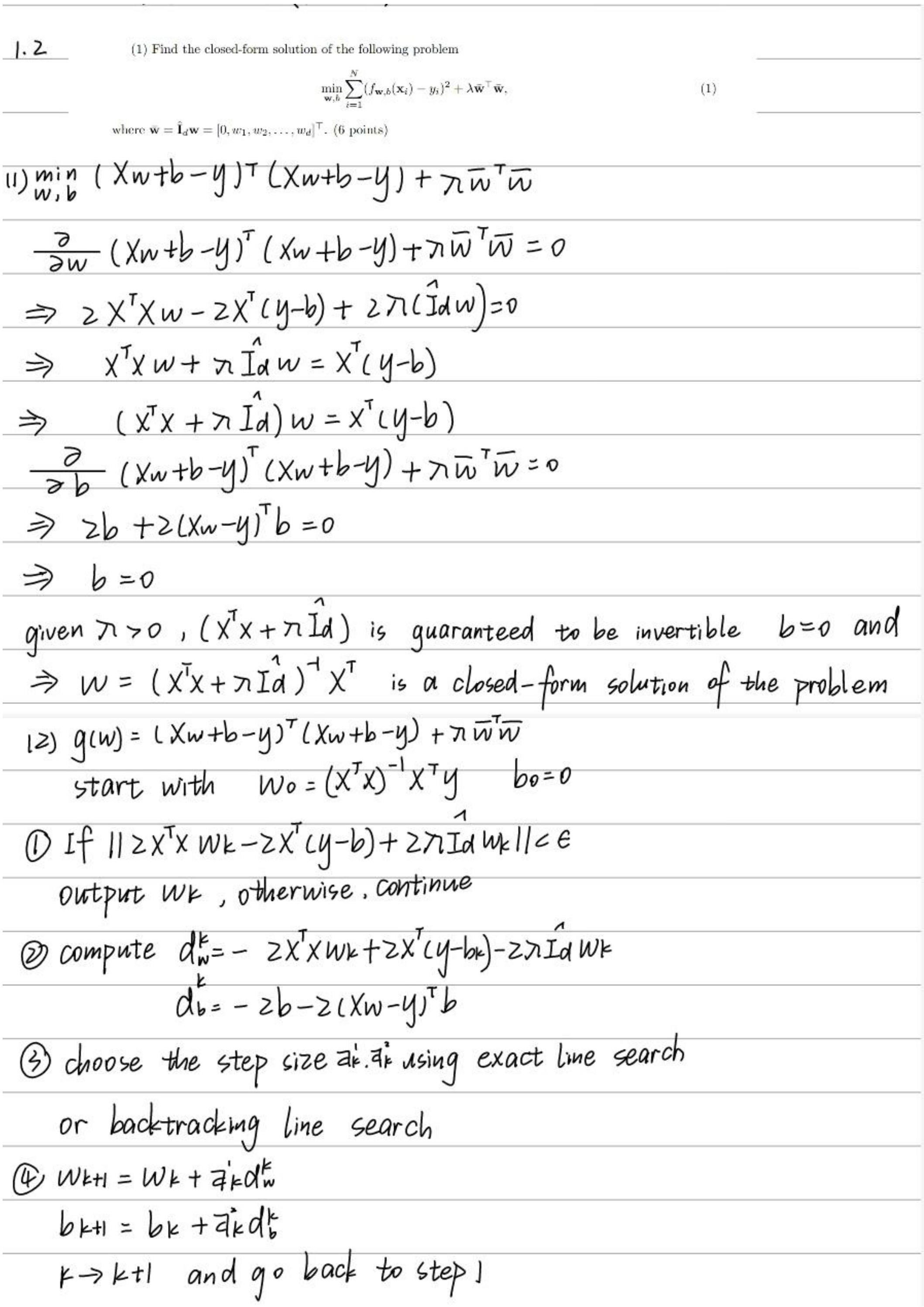
$$= \begin{bmatrix} 2w_1 x_{11} + \sum_{i=2}^{d} w_i x_{i1} + w_2 x_{12} + w_3 x_{13} + \cdots + w_d x_{1d} \\ \vdots \\ w_1 x_{d1} + w_2 x_{d2} + w_3 x_{d3} + \cdots + w_{d-1} x_{d-1,3} + 2 w_d x_{dd} + \sum_{i=1}^{d-1} w_i x_{id} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{d} w_i x_{i1} + \sum_{j=1}^{d} w_j x_{1j} \\ \vdots \\ \sum_{i=1}^{d} w_i x_{id} + \sum_{j=1}^{d} w_j x_{dj} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{d} x_{i1} w_i \\ \\ \sum_{i=1}^{d} x_{id} w_i \end{bmatrix} + \begin{bmatrix} \sum_{j=1}^{d} x_{1j} w_j \\ \vdots \\ \sum_{j=1}^{d} x_{dj} w_j \end{bmatrix}$$

$$= \begin{bmatrix} x_{11} & \cdots & x_{d1} \\ \vdots & & \vdots \\ x_{1d} & \cdots & x_{dd} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix} + \begin{bmatrix} x_{11} & \cdots & x_{1d} \\ \vdots & & \vdots \\ x_{d1} & \cdots & x_{dd} \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$$

$$= X^T w + X w = (X + X^T) w$$

1.2 Question 1.2



1.2 (1) Find the closed-form solution of the following problem

$$\min_{\mathbf{w},b}\sum_{i=1}^{N}(f_{\mathbf{w},b}(\mathbf{x}_i)-y_i)^2+\lambda\bar{\mathbf{w}}^\top\bar{\mathbf{w}}, \qquad (1)$$

where $\bar{\mathbf{w}}=\hat{\mathbf{I}}_d\mathbf{w}=[0,w_1,w_2,\ldots,w_d]^\top$. (6 points)

(1) $\min_{w,b}\ (Xw+b-y)^T(Xw+b-y)+\lambda\bar{w}^T\bar{w}$

$\frac{\partial}{\partial w}(Xw+b-y)^T(Xw+b-y)+\lambda\bar{w}^T\bar{w}=0$

$\Rightarrow\ 2X^TXw-2X^T(y-b)+2\lambda(\hat{I}_dw)=0$

$\Rightarrow\ X^TXw+\lambda\hat{I}_dw=X^T(y-b)$

$\Rightarrow\ (X^TX+\lambda\hat{I}_d)w=X^T(y-b)$

$\frac{\partial}{\partial b}(Xw+b-y)^T(Xw+b-y)+\lambda\bar{w}^T\bar{w}=0$

$\Rightarrow\ 2b+2(Xw-y)^Tb=0$

$\Rightarrow\ b=0$

given $\lambda>0$, $(X^TX+\lambda\hat{I}_d)$ is guaranteed to be invertible $b=0$ and

$\Rightarrow\ w=(X^TX+\lambda\hat{I}_d)^{-1}X^T$ is a closed-form solution of the problem

(2) $g(w)=(Xw+b-y)^T(Xw+b-y)+\lambda\bar{w}^T\bar{w}$

start with $w_0=(X^TX)^{-1}X^Ty$ $\quad b_0=0$

① If $\|2X^TXw_k-2X^T(y-b)+2\lambda\hat{I}_dw_k\|<\epsilon$

output w_k, otherwise, continue

② compute $d_w^k=-2X^TXw_k+2X^T(y-b_k)-2\lambda\hat{I}_dw_k$

$d_b^k=-2b-2(Xw-y)^Tb$

③ choose the step size $\alpha_k^1.\alpha_k^2$ using exact line search

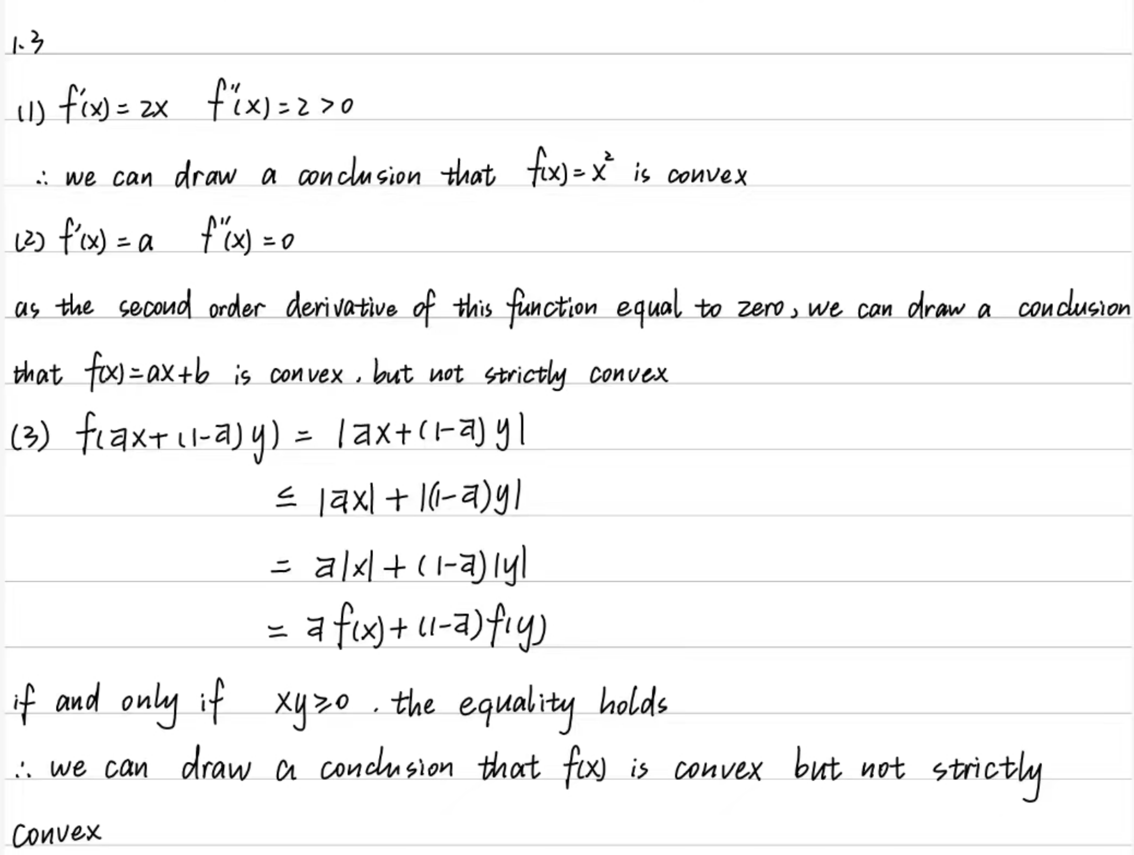
or backtracking line search

④ $w_{k+1}=w_k+\alpha_k^1d_w^k$

$b_{k+1}=b_k+\alpha_k^2d_b^k$

$k\to k+1$ and go back to step 1

1.3 Question 1.3



1.3

(1) $f'(x) = 2x \quad f''(x) = 2 > 0$

$\therefore$ we can draw a conclusion that $f(x) = x^2$ is convex

(2) $f'(x) = a \quad f''(x) = 0$

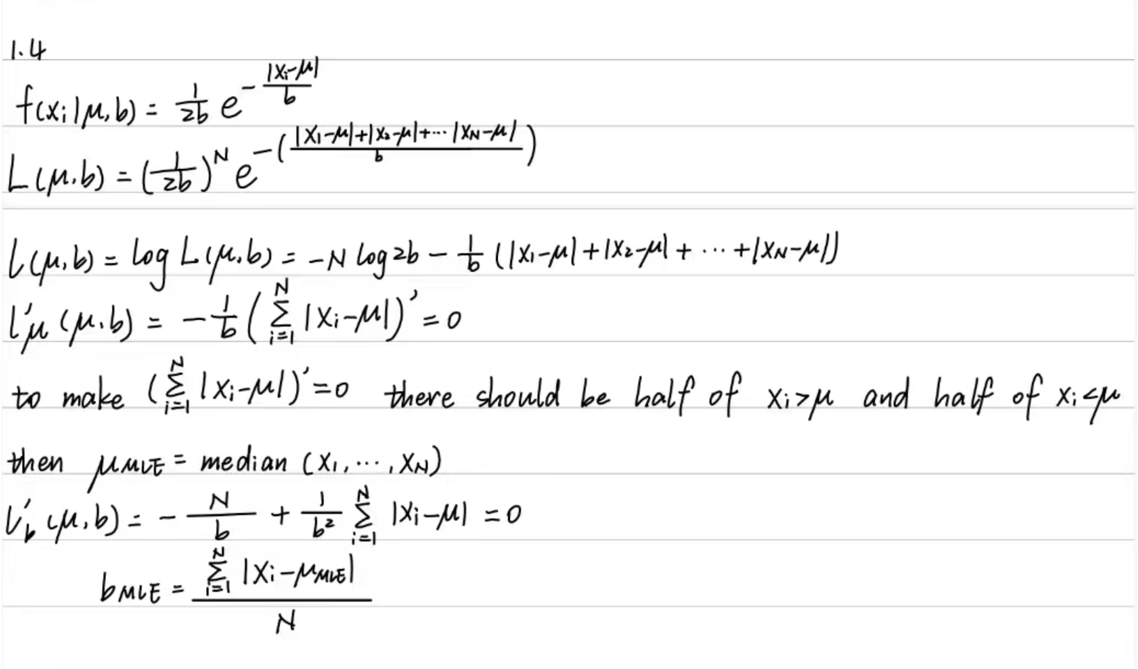
as the second order derivative of this function equal to zero, we can draw a conclusion that $f(x) = ax+b$ is convex. but not strictly convex

(3) $$\begin{aligned} f(\lambda x + (1-\lambda) y) &= |\lambda x + (1-\lambda) y| \\ &\leq |\lambda x| + |(1-\lambda) y| \\ &= \lambda |x| + (1-\lambda)|y| \\ &= \lambda f(x) + (1-\lambda) f(y) \end{aligned}$$

if and only if $xy \geq 0$. the equality holds

$\therefore$ we can draw a conclusion that $f(x)$ is convex but not strictly convex

1.4 Question 1.4



1.4

$$f(x_i|\mu,b) = \frac{1}{2b} e^{-\frac{|x_i-\mu|}{b}}$$

$$L(\mu,b) = \left(\frac{1}{2b}\right)^N e^{-\left(\frac{|x_1-\mu|+|x_2-\mu|+\cdots |x_N-\mu|}{b}\right)}$$

$$l(\mu,b) = \log L(\mu,b) = -N \log 2b - \frac{1}{b}\left(|x_1-\mu|+|x_2-\mu|+\cdots+|x_N-\mu|\right)$$

$$l'_\mu(\mu,b) = -\frac{1}{b}\left(\sum_{i=1}^{N}|x_i-\mu|\right)' = 0$$

to make $(\sum_{i=1}^{N}|x_i-\mu|)'=0$ there should be half of $x_i>\mu$ and half of $x_i<\mu$

then $\mu_{MLE} = \text{median}(x_1,\cdots,x_N)$

$$l'_b(\mu,b) = -\frac{N}{b} + \frac{1}{b^2}\sum_{i=1}^{N}|x_i-\mu| = 0$$

$$b_{MLE} = \frac{\sum_{i=1}^{N}|x_i-\mu_{MLE}|}{N}$$

2 Programming Question

2.1 Step 1

In this assignment, we need to predict the attributes 'MEDV' using other attributes. There are 14 attributes in total. Through simple data analysis, all attributes are float except for chas, rad, and tax, which are integers. In addition, we found no incomplete data points in the original data set such as 'NAN' or 'Null'. The attributes 'CHAS' can only be 0 or 1.

From the definition of different attributes, we can guess the most relevant attribute for MEDV is LSTAT which means percent of lower status of the population.

2.2 Step 2

In the second step, we use the seaborn library **lineplot function** to plot the MEDV distributions over each attribute. The distributions of MEDV is shown as the figure 1.

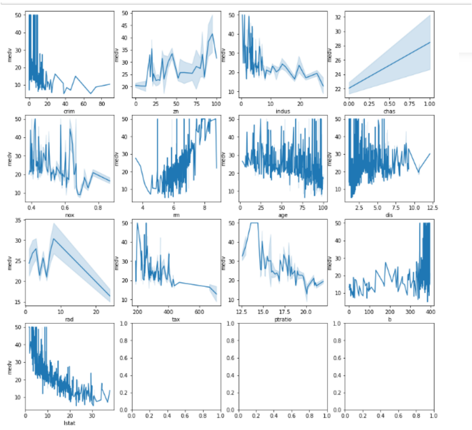


Figure 1: Distributions of MEDV

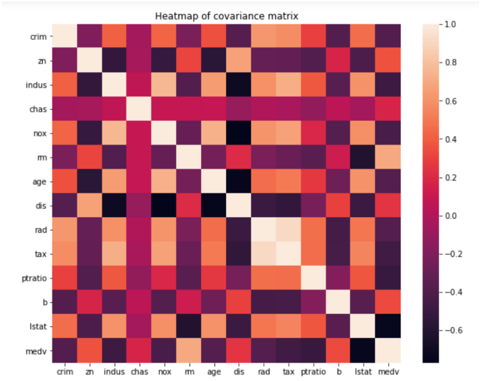


Figure 2: heatmap

From this figure we can draw a conclusion that the attribute INDUS, NOX, RM, RAD, TAX, PTRATIO and LSTAT have high correlation with the MEDV. These attributes have an approximately linear relationship with the attribute MEDV. There is no strong linear correlation between the other properties and MEDV.

2.3 Step 3

In the third step, we use seaborn.heatmap function to plot thr pairwise correlation on data. The heatmap is shown as the figure 2.

From this heatmap we can see that INDUS, RM, TAX, PTRATIO and LSTAT have high correlation with the MEDV. These attributes are good indications of using as predictors. Besides, other attributes have no strong correlation with MEDV.

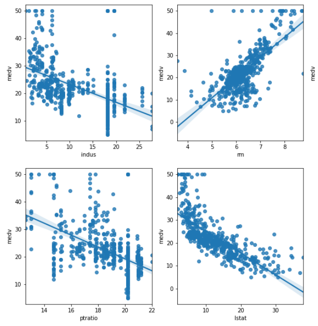
2.4 Step 4

Due to the strong correlation in the third step of selective attributes of the unit is not consistent, so we need to use **sklearn.Preprocessing.MinMaxScaler** function to normalize. The formular to normalize is

$$X_{std} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$X_{scaled} = X_{std} * (X_{max} - X_{min}) + X_{min}$$

Then we use seaborn.regplot to plot the relevance of INDUS, RM, TAX, PTRATIO and LSTAT with MEDV. We set the parameter **confident interval** equal to 95. The relevance is shown as the below figure 3.



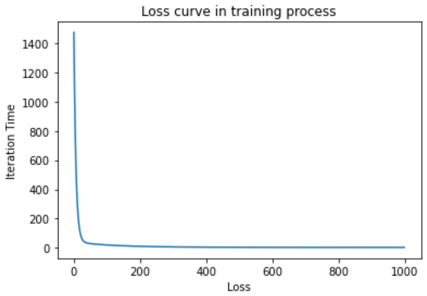


Figure 3: The relevance of choosing column against MEDV

Figure 4: The loss curve in training process

2.5 Step 5

In the step 5, we use the **Grandient Descent Method** to build a linear regression model.

The linear regression is formulated to the following optimization problem

$$\overline{w}^* = \arg\min_{\overline{w}} J(\overline{w}),\ J(\overline{w}) = \tfrac{1}{2}(X\overline{w} - y)^2$$

where $X_{train} \in \mathbb{R}^{404\times(5+1)}$ and $\overline{w} = [b; w] \in \mathbb{R}^{(5+1)\times 1}$

Then $\overline{w}$ can be updated by **Grandient Descent Algorithm**

$$\overline{w} \leftarrow \overline{w} - \alpha \frac{J(\overline{w})}{\overline{w}}$$

where α is called step-size or learning rate, $\frac{J(\overline{w})}{\overline{w}}$ should be the gradient equal to $X^\top(X\overline{w} - y)$, every iteration we update the gradient and find a new $\overline{w}$

Learning rate is set to 0.0001 as default. **Iteration steps** is set to 1000 as default.

LOSS

We calculate $loss = \frac{1}{2}(X\overline{w} - y)^2$ in the training process. Plot the loss curves in the training process as figure 4 above.

And we can use this formular to calculate RMSE:

$$\sqrt{\frac{1}{m}\sum_{i=0}^{m}(y_i - \hat{y}_i)^2}$$

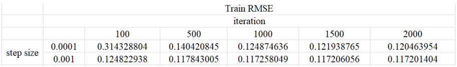
Finally we successfully calculate the training error in terms of RMSE is 0.12487463626425599 and the testing error in terms of RMSE is 0.13751375521100107.

2.6 Step 6

In the step 6, we choose the step size to be 0.0001 or 0.001, and iteration steps to be 100, 500, 1000, 1500, 2000 and calculate the RMSE using formular

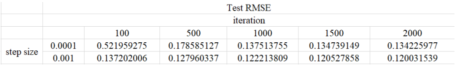
$$\mathrm{RMSE} = \sqrt{\tfrac{1}{m}\textstyle\sum_{i=0}^{m}(y_i - \hat{y}_i)^2}.$$

Then we can see the train RMSE with different parameters in figure 5 below, and the test RMSE with different parameter in figure 6 below.(Keep 15 decimal places)



		Train RMSE				
		iteration				
		100	500	1000	1500	2000
step size	0.0001	0.314328804	0.140420845	0.124874636	0.121938765	0.120463954
	0.001	0.124822938	0.117843005	0.117258049	0.117206056	0.117201404

Figure 5: The Train RMSE with Different Parameters



		Test RMSE				
		iteration				
		100	500	1000	1500	2000
step size	0.0001	0.521959275	0.178585127	0.137513755	0.134739149	0.134225977
	0.001	0.137202006	0.127960337	0.122213809	0.120527858	0.120031539

Figure 6: The Test RMSE with Different Parameters

When iteration steps equal to 100 and step size equal to 0.0001, the training error in terms of RMSE is 0.31432880392173557 and the testing error in terms of RMSE is 0.5219592749349572. When iteration steps equal to 100 and step size equal to 0.001, the training error in terms of RMSE is 0.12482293847357762 and the testing error in terms of RMSE is 0.137202005710657. When iteration steps equal to 500 and step size equal to 0.0001, the training error in terms of RMSE is 0.14042084480439 81 and the testing error in terms of RMSE is 0.178585127233023. When iteration steps equal to 500 and step size equal to 0.001, the training error in terms of RMSE is 0.11784300481753227 and the testing error in terms of RMSE is 0.12796033654211392. When iteration steps equal to 1000 and step size equal to 0.0001, the training error in terms of RMSE is 0.12487463626425599 and the testing error in terms of RMSE is 0.13751375521100107. When iteration steps equal to 1000 and step size equal to 0.001, the training error in terms of RMSE is 0.11725804928031328 and the testing error in terms of RMSE is 0.12221380944381176. When iteration steps equal to 1500 and step size equal to 0.0001, the training error in terms of RMSE is 0.12193876541995663 and the testing error in terms of RMSE is 0.13473914939174692. When iteration steps equal to 1500 and step size equal to 0.001, the training error in terms of RMSE is 0.1172060561214392 and the testing error in terms of RMSE is 0.12052785840730477. When iteration steps equal to 2000 and step size equal to 0.0001, the training error in terms of RMSE is 0.12046395433297512 and the testing error in terms of RMSE is 0.13422597662192484. When iteration steps equal to 2000 and step size equal to 0.001, the training error in terms of RMSE is 0.11720140437734465 and the testing error in terms of RMSE is 0.12003153910056037.

Finally we get the RMSE of different parameters. From the result of operation we can draw a conclusion that the larger iteration step, the smaller the RMSE. Besides, with too large step size, the $\overline{w}$ will not converge and when the step size become larger an larger, the RMSE first become smaller and smaller.