DDA3020 Assignment1

Your name

Student ID

Date: March 7, 2023

1 Written Homework

1.1 Please prove the following derivatives: Consider $\mathbf{X} \in \mathbb{R}^{h \times d}$ and $\mathbf{y} \in \mathbb{R}^{h \times 1}$, which are not functions of $\mathbf{w}$:

$$\frac{d\left(\mathbf{y}^{\top} \mathbf{X} \mathbf{w}\right)}{d \mathbf{w}}=\mathbf{X}^{\top} \mathbf{y}, \quad \text{(4 points)}$$
$$\frac{d\left(\mathbf{w}^{\top} \mathbf{w}\right)}{d \mathbf{w}}=2 \mathbf{w}, \quad \text{(4 points)}$$

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

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

Proof. For convenience, the notation of vectors are not under `mathbf` environment.

1. To prove the first equation, it's sufficient to prove $\dfrac{d(Aw)}{dw} = A^{\top}$ for $\forall A^{1\times d}$. For original equation, we only need to take $A = \mathbf{y}^{\top}\mathbf{X}$.

$$Aw = a_1w_1 + a_2w_2 + \ldots + a_dw_d.$$
$$\frac{d(\mathbf{Aw})}{d\mathbf{w}} = \begin{bmatrix} \frac{\partial(Aw)}{\partial w_1} \\ \vdots \\ \frac{\partial(Aw)}{\partial w_d} \end{bmatrix} = \begin{bmatrix} a_1 \\ \vdots \\ a_d \end{bmatrix} = A^{\top}.$$

2. For $w^{d\times 1} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$, $w^{\top}w = \sum_{i=1}^{d} w_i^2$.

$$\frac{d\left(\mathbf{w}^{\top} \mathbf{w}\right)}{d \mathbf{w}} = \begin{bmatrix} 2w_1 \\ \vdots \\ 2w_d \end{bmatrix} = 2w.$$

3. Denote $X^{d\times d} = \begin{bmatrix} X_{11} & \dots & X_{1d} \\ \vdots & & \vdots \\ X_{d1} & \dots & x_{dd} \end{bmatrix}$, $w^{d\times 1} = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}$.

$$w^\top X w = \sum_{i=1}^{d}\sum_{j=1}^{d} w_i w_j X_{ij}.$$

$$\frac{d\left(\mathbf{w}^\top \mathbf{x}\mathbf{w}\right)}{d\mathbf{w}} = \begin{bmatrix} (x_{11} + x_{12} + \ldots + x_{1d}) + (x_{11} + x_{21} + \ldots + x_{d1}) \\ \vdots \\ (x_{d1} + x_{d2} + \ldots + x_{dd}) + (x_{1d} + x_{2d} + \ldots + x_{dd}) \end{bmatrix} = \left(x + x^\top\right) w.$$

□

1.2

1.3 Prove that:

(1) $f(x) = x^2$ is convex. (4 points)

Proof.

□

1.4

2 Programming Report

Step 1: use pandas library to check the data in the dataset. Process incomplete data point such as 'NaN' or 'Null'. Briefly summarize the characteristics of this dataset and guess which is the most relevant attribute for MEDV.

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	b	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

There is no incomplete data (NaN or Null) in the dataset. Some attributes like ZN, INDUS, CHAS, NOX, RAD, TAX, PTRATIO, B are relatively concentrated distributed compared with the others.

Guess: The attributes like CRIM, ZN, INDUS, CHAS, NOX, AGE, DIS, RAD, TAX, PTRATIO, B, LSTAT may not have an serious impact on MEDV. While some of the attributes like RM may be positvely correlated to MEDV.

Step 2: use seaborn library to visualize the dataset. Plot the MEDV distributions over each attribute. Briefly analyze the characteristics of the attributes and revise the assumption in Step 1 if necessary.

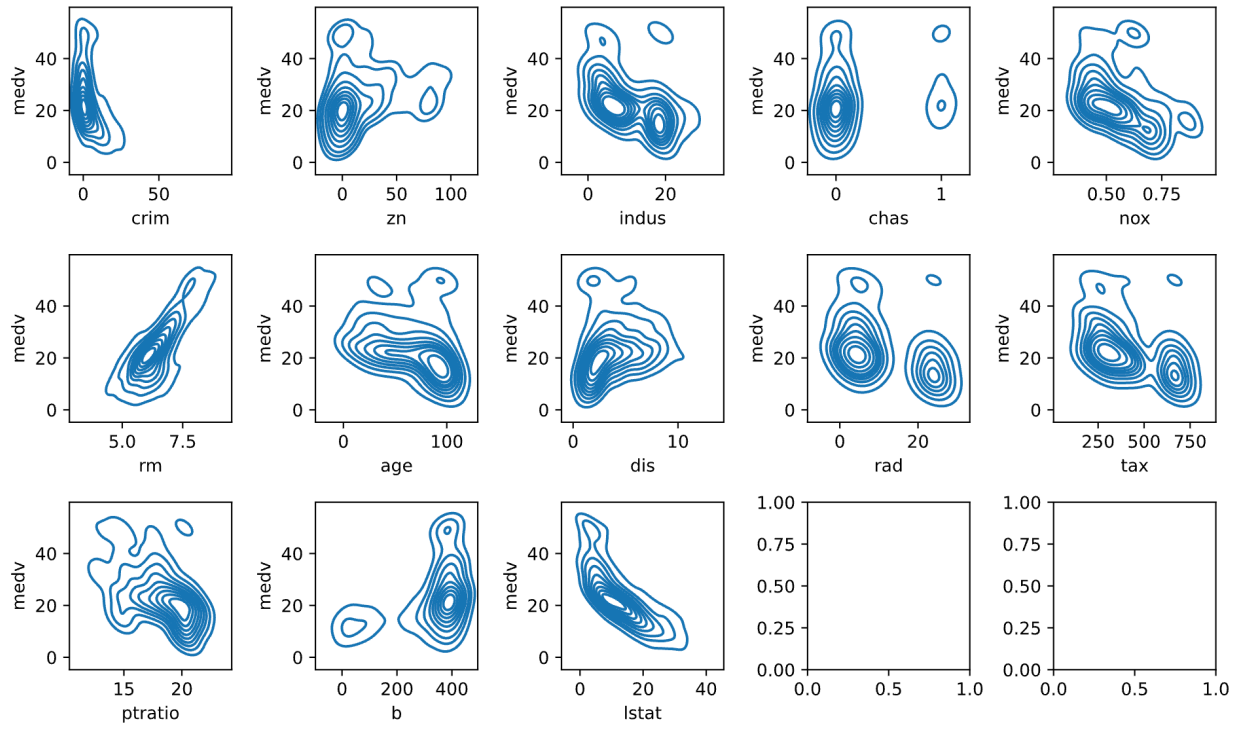

Figure 1: MEDV distributions (Step 2)

From the above figures, I guess rm, dis, ptratio and lstat may correlated to the medv in a relatevely higher degree compare with other attributes.

Step 3: use seaborn.heatmap function to plot the pairwise correlation on data. Select the good attributes which are good indications of using as predictors. Report your findings.

Figure 2: Step 3. covariance matrix

The attributes `indus`, `nox`, `rm`, `tax`, `ptratio` and `lstat` may be good predictors.

The criteria: $|Cov(\text{attributes}, \text{medv})| > 0.4$.

Step 4: use `sklearn.preprocessing.MinMaxScaler` function to scale the columns you select in Step 3. Then use `seaborn.regplot` to plot the relevance of these columns against MEDV with 95% confidence interval.

Step 5: Randomly split the data into two parts, one contains 80% of the samples and the other contains 20% of the samples. Use the first part as training data and train a linear regression model and make prediction on the second part with gradient descent methods. X should be the attributes you select in previous steps. Report the training error and testing error in terms of RMSE. Plot the loss curves in the training process.

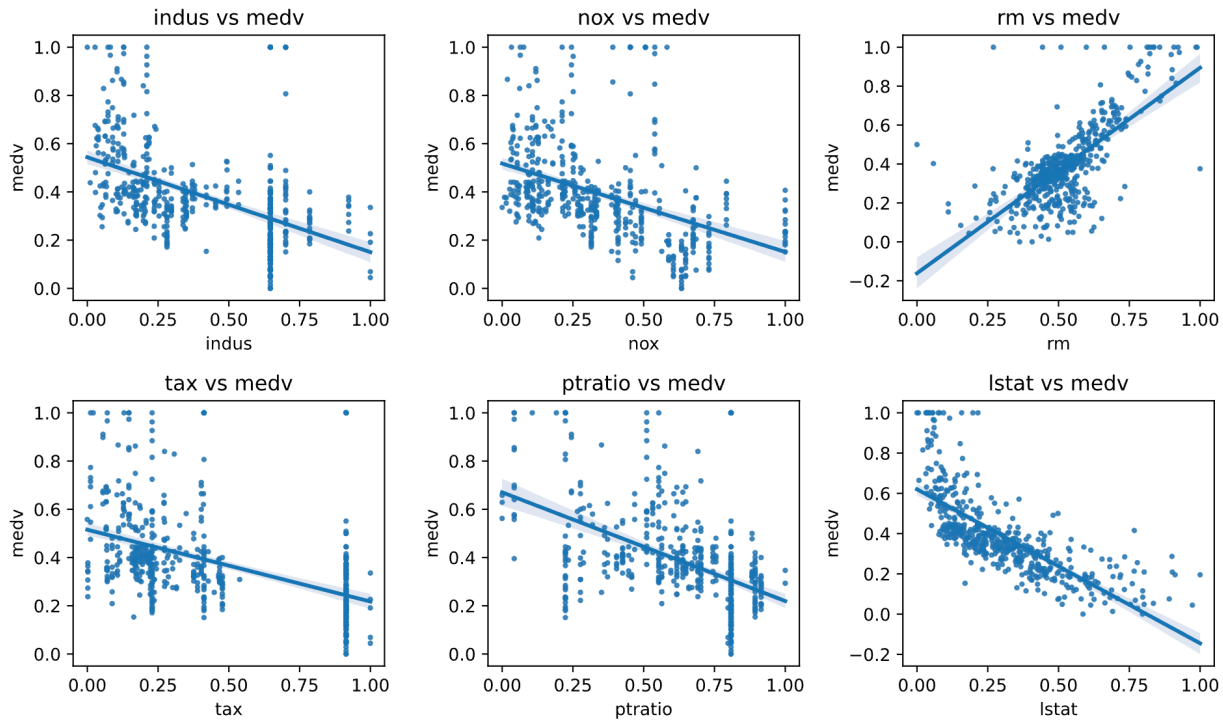

Figure 3: Relevance of MEDV (Step 4)

Linear Model: The attributes selected in step 3 are used for constructing linear model. The goal of the model is to explore the linear relationship between the these attributes and MEDV. To train the linear model, the Boston housing Dataset is separated into training and testing parts. After testing this model by different combinations of hyperparameters, the model with best testing performance is selected as final model.

Loss function:

$$J(w,b) = \frac{1}{2m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

$$\frac{\partial J(w,b)}{\partial w} = \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})x^{(i)}$$
$$\frac{\partial J(w,b)}{\partial b} = \frac{1}{m}\sum_{i=1}^{m}(f_{w,b}(x^{(i)}) - y^{(i)})$$

Hyperparameter settings:

Learning rate: $\alpha = 10^{-5}$.

Iteration times: `iter_nums` = 1000 (by default).

Result: $w : [1.145 \quad -0.481 \quad 0.178 \quad 0.002 \quad 0.606 \quad -0.542], b : 0.0374.$

Training error: 8.979474 (in RMSE) Testing error: 9.921338 (in RMSE)

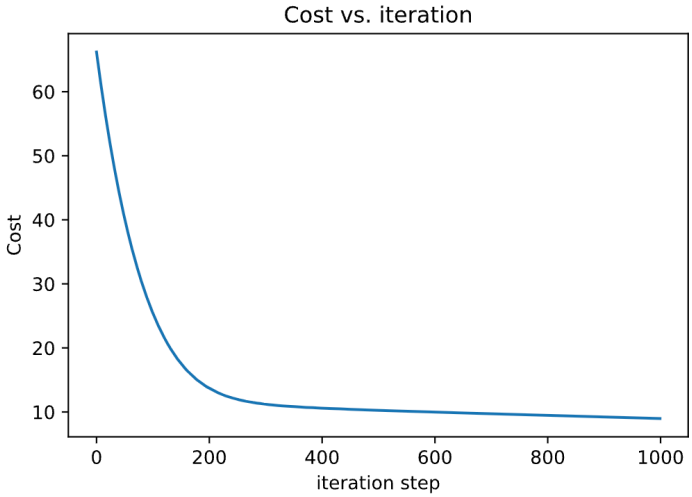

Figure 4: Loss vs. iteration steps (step 5)

Step 6: Repeat the splitting, training, and testing for 10 times with different parameters such as step size, iteration steps, etc. Use a loop and print the RMSEs in each trial. Analyze the influence of different parameters on RMSE.

Learning rate	Iteration number	Training error	Testing error
10^{-7}	100	33.24884	35.58079
10^{-7}	1000	15.98171	16.45367
10^{-7}	10000	8.88256	7.27234
5×10^{-7}	100	9.92086	10.53758
5×10^{-7}	1000	14.96718	14.80721
5×10^{-7}	10000	10.79177	10.60971
10^{-6}	100	20.46792	19.47656
10^{-6}	1000	8.58127	8.43788
10^{-6}	10000	8.06653	6.40161
5×10^{-6}	100	12.96556	13.31157
5×10^{-6}	1000	12.65944	13.53177
5×10^{-6}	10000	6.22781	6.77105
10^{-5}	100	13.77507	11.49493
10^{-5}	1000	10.42047	9.25771
10^{-5}	10000	∞	∞

The best test performance is attained with parameters:

w : $[1.071 \quad 0.217 \quad 1.982 - 0.052 \quad 0.961 \quad 0.083]$, b : 0.00605

learning_rate: 1e-06 num_iters: 10000 test_error: 6.40161 (in RMSE)

Findings: Different parameters influence the performance of linear. With the increasing of iteration numbers, the test accuracy always improved. However, the learning rate is not obviously correlated with the test performance. When the learning rate $< 10^{-6}$, the model converges slowly. When the learning rate $> 10^{-5}$, the model may also diverge.