Problem 1 (10pts) Linear Algebra.

1. A rotation in 3D by angle α about the z axis is given by the following matrix:

$$\mathbf{R}(\alpha) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Prove that $\mathbf{R}$ is an orthogonal matrix, i.e., $\mathbf{R}^T\mathbf{R} = \mathbf{I}$, for any α.

2. Prove that the eigenvalue of an orthogonal matrix must be 1 or -1.

1. $$R^T = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R \cdot R^T = \begin{pmatrix} \cos^2(\alpha)+\sin^2(\alpha) & -\sin(\alpha)\cos(\alpha)+\cos(\alpha)\sin(\alpha) & 0 \\ -\sin(\alpha)\cos(\alpha) & \sin^2(\alpha)+\cos^2(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I$$

so R is orthogonal matrix

2. Let $(R - I\lambda)X = 0$

we have $(R^TR - R^T\lambda)X = 0$

$\Rightarrow X = \lambda R^T X$

$\Rightarrow X^TX = \lambda(X^TR^T)X$

$= \lambda(\lambda X)^T X$

$= \lambda^2 X^TX$

$\Rightarrow \lambda = 1$ or -1

Problem 2 (10pts) Optimization.
Prove that:

(1) $f(x) = |x|$ is convex;

(2) $f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ is convex, where $\mathbf{A}$ is a matrix.

(1). Let $t \in [0, 1]$

$t|x_1| + (1-t)|x_2|$

$= |tx_1| + |(1-t)x_2|$

$\geq |tx_1 + (1-t)x_2|$

i.e. $tf(x_1) + (1-t)f(x_2) \geq f(tx_1 + (1-t)x_2)$

so $f(x)$ is convex

(2). $f(x) = (Ax)^T Ax - 2Ax \cdot b + b^2$

$\nabla f(x) = 2A^TAx - 2Ab + b^2$

$\nabla^2 f(x) = 2A^TA$

$x^T \nabla^2 f(x) x = 2(Ax)^T Ax \geq 0$

so $\nabla^2 f(x)$ is semi-definite

so $f(x)$ is convex

Problem 3 (10pts) Information Theory.
Proof that cross-entropy is not smaller than entropy, i.e., $H_{P,Q}(\mathcal{X}) \geq H_P(\mathcal{X})$, and the equality holds only when $P = Q$.

$$H_P(X) = -\sum p(x) \log p(x)$$

$$H_{P,Q}(X) = -\sum p(x) \log q(x)$$

$$= -\sum p(x) \log p(x) + \sum p(x) \log \frac{p(x)}{q(x)}$$

$$= H_P(X) + \sum p(x) \log \frac{p(x)}{q(x)}$$

$$= H_P(x) - \sum p(x) \log \frac{q(x)}{p(x)}$$

since $f(x) = \log x$ is concave, according to Jensen's inequality

$$\sum p(x) \log \frac{q(x)}{p(x)} \leq \log \left(\sum p(x) \cdot \frac{q(x)}{p(x)} \right) = 0$$

the equality holds only when $P = Q$

so $H_{P,Q}(X) \geq H_P(X)$, and the equality holds only when $P = Q$

Problem 4 (10pts) Linear Regression.
Suppose we have training data $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}, i = 1, 2, \ldots, N$. Consider $f_{\mathbf{w},b}(\mathbf{x}_i) = \mathbf{x}_i^T\mathbf{w} + b$, where $\mathbf{w} = [w_1, w_2, \ldots, w_d]^T$.

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

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

where $\bar{\mathbf{w}} = \hat{\mathbf{I}}_d \mathbf{w} = [0, w_1, w_2, \ldots, w_d]^T$. Note that $\hat{\mathbf{I}}_d = \begin{bmatrix} 0 & 0 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ & & & \vdots & \\ 0 & \ldots & 0 & 0 & 1 \end{bmatrix} \in \mathbf{R}^{(d+1)\times d}$

(2) Show how to use gradient descent to solve the problem.

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

$\Rightarrow 2X^TXw - 2X^Ty + 2\lambda \hat{I}_d w = 0$

$\Rightarrow X^TXw + \lambda \hat{I}_d w = X^Ty$

$\Rightarrow (X^TX + \lambda \hat{I}_d) w = X^Ty$

when $\lambda > 0$, $X^TX + \lambda \hat{I}_d$ is invertible

$w = (XX^T + \lambda \hat{I}_d)^{-1} X^Ty$

Actually, w here involves [b, w], which means the the real w is the solution without the first column.

(2). $w \leftarrow w - \alpha \frac{\partial J(w)}{\partial w}$

where $J(w) = \frac{1}{2} (X \cdot \bar{w} - y)^2 + \lambda \bar{w}^T w$,

$\frac{\partial J(w)}{\partial w} = X^T(X^Tw - y) + 2\lambda w$,

α is the learning rate.

Update w until stopping criteria is satisfied

Problem 5 (10pts) MLE.

Consider a linear regression model with a 2 dimensional response vector $\boldsymbol{y}_i \in \mathbb{R}^2$. Suppose we have some binary input data, $x_i \in \{0, 1\}$. The training data is as follows:

x	y
0	$(-1,-1)^T$
0	$(-1,-2)^T$
0	$(-2,-1)^T$
1	$(1,1)^T$
1	$(1,2)^T$
1	$(2,1)^T$

Let us embed each x_i into 2 d using the following basis function:

$$\phi(0) = (1,0)^T, \quad \phi(1) = (0,1)^T$$

The model becomes

$$\hat{\boldsymbol{y}} = \mathbf{W}^T \boldsymbol{\phi}(x)$$

where $\mathbf{W}$ is a 2×2 matrix. Compute the MLE for $\mathbf{W}$ from the above data.

$$W_{MLE} = \underset{W}{\text{argmax}} \log(W; D)$$

$$= \underset{W}{\text{argmax}} \left(m \log\left(6 \cdot (2\pi)^{\frac{1}{2}}\right) - \frac{1}{2 6^2} \sum_{i=1}^{m} (y_i - W^T x_i)^2 \right)$$

$$= \underset{w}{\arg\min} \frac{1}{2} \sum_{i=1}^{m} (y_i - W^T x_i)^2$$

so $\hat{W} = (X^T X)^{-1} X^T y$

wher $X = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}^T$

$y = \begin{bmatrix} -1 & -1 & -2 & 1 & 1 & 2 \\ -1 & -2 & -1 & 1 & 2 & 1 \end{bmatrix}^T$

$$\hat{W} = \begin{bmatrix} \frac{1}{3} & 0 \\ 0 & \frac{1}{3} \end{bmatrix} \cdot \begin{bmatrix} -4 & -4 \\ 4 & 4 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{4}{3} & -\frac{4}{3} \\ \frac{4}{3} & \frac{4}{3} \end{bmatrix}$$

DDA3020 homework1

Mengkang Li
122090264

February 2024

1 Written Homework

Written homework are attached to the head of the file.

2 Programming Report

2.1 Problem 1

We want to use apppropriate attributes in "SqFt, Bedrooms, Bathrooms, Neighborhood" to predict the attributes "Price" in dataset "house_prices".

2.1.1 Step 1

Use *pandas* library to load the csv file into *pandas.Dataframe*. Use *Dataframe.astype* function to convert the data type of "Neighborhood" attribute to "category" type. Use *Dataframe.info* and *Dataframe/describe* functions to check the dataset. Briefly summarize the information of the dataset.

Table 1: Data Information

Column	**Non-Null Count**	**Dtype**
SqFt	128 non-null	int64
Bedrooms	128 non-null	int64
Bathrooms	128 non-null	int64
Neighborhood	128 non-null	category
Price	128 non-null	int64

The dataset has 128 rows in total, and there's no null data in it. Most of the houses have 3 bedrooms. More than 50% of the houses have 2 bathrooms. SqFt are between 1450 and 2590. Prices are between 130427 and 211200.

	SqFt	Bedrooms	Bathrooms	Price
count	128	128	128	128
mean	2000.937500	3.023438	2.445312	130427.343750
std	211.572431	0.725951	0.514492	26868.770371
min	1450.000000	2.000000	2.000000	69100.000000
25%	1880.000000	3.0000000	2.000000	111325.000000
50%	2000.000000	3.000000	2.000000	125950.000000
75%	2140.000000	3.000000	3.000000	148250.000000
max	2590.000000	5.000000	4.0000000	211200.000000

Table 2: Description of the Data

2.1.2 Step 2

Use *seaborn* library to visualize dataset. Use *seaborn.pairplot* function to plot the "Price" against each numeric attributes "SqFt", "Bedrooms" and "Bathrooms" with data points colored differently based on the values of the "Neighborhood" category attributes. Use *seaborn.heatmap* function to plot the pairwise correlation on data. Breifly analyze the potential patterns between "Price" and other attributes.

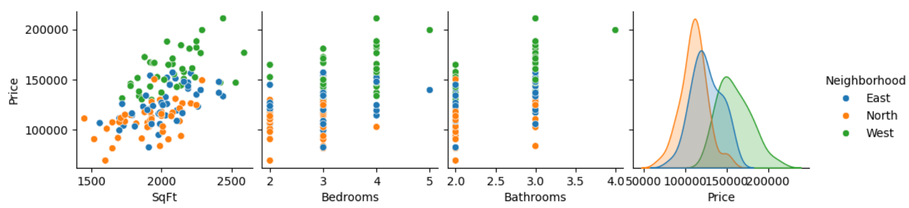

Figure 1: Pair Plot

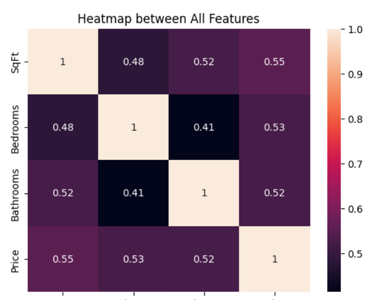

Figure 2: Heatmap of correlation on price and other features

According to the heatmap and the pair plot, as the increment of SqFt and the number of bedrooms and bathrooms, the price will increase. Besides, the

SqFt has the strongest correlation with the price, although only a few percentage points higher than other two features.

2.1.3 Step 3

Use *sklearn* library to process the category variable. For category variable "Neighborhood", we could use *ColumnTransformer* function in *sklearn.compose* and *OneHotEncoder* function in *sklearn.preprocessing* to convert the category column into a one-hot numeric matrix in the dataset. Use sklearn library to split data into train and test subset. We use *train_test_split* in *sklearn.model_selection* to randomly split the data into two parts, one contains 80% of the samples as train data and the other contains 20% of the samples as test data.

X_{train}	X_{test}	y_{train}	y_{test}
(102, 6)	(26, 6)	(102,)	(26,)

Table 3: Dataset Sizes

2.1.4 Step 4

Use sklearn library to train and evaluate a linear regression model. We use *LinearRegression* function in *sklearn.linear_model* to train a linear regression model with "Price" as target and "SqFt", "Bedrooms", "Bathrooms", "Neighborhood" as predictors. After training(*model.fit*) and predictng(*model.predict*) on train and test dataset, we could use *mean_squared_error* function in *sklearn.metrics* to evaluate the performance of the fitted model. Report the training error and testing error in terms of RMSE.

The random seed is 157.

Train RMSE	Test RMSE
14349.994710239715	15291.173754894138

Table 4: RMSE Result

2.2 Problem 2

We want to use use attributes 1-10 to predict attributes 11 in diabetes dataset.

2.2.1 Step 1

Use numpy library to conduct the training of linear regression model. We use matrix operations in numpy to write the codes of learning the parameter with gradient descent method.

Implementation detail: I have three function in the linear_regression class. I use __init__ to give four attributes to the class. In train function, I implement gradient descent method using a fixed learning rate and stopping criteria, and use a list to collect the training loss in the process. In the predict function, I use the atrribute "W" to compute the final prediction and return it.

2.2.2 Step 2

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. Report the training error and testing error in terms of RMSE. Plot the loss curves in the training process.

Train RMSE	Test RMSE
53.79446095679401	53.51328957925159

Table 5: RMSE of the Linear Model

The mean squared error loss curve are as followed.

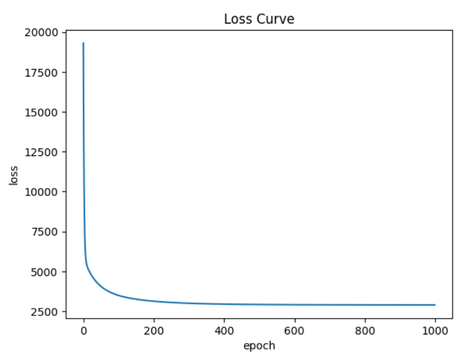

Figure 3: Loss Curve of the Linear Model

2.2.3 Step 3

Repeat the splitting, trainning, and testing for 10 times with different parameters such as step size, iterations, etc. Use a loop and print the RMSE in each trial. Analyze the influence of different parameters on RMSE.

The train and loss RMSE are as followed.

The best performance is attained with parameters:

$w : [-14.22754289\ \ -215.11133207\ 521.36643532\ 337.01578351\ -256.83988241\ 40.50294799\ -167.06394612\ \ 119.86015846\ \ 527.1779997\ \ 63.03044506]$

$b\ =\ 151.97752793$

Findings:

- Different parameters influence the linear model differently.

Learning Rate	Iteration Number	Train RMSE	Test RMSE
0.0005	100	73.25617	71.80073
0.0005	1000	59.13736	58.16296
0.0005	10000	53.79476	53.51389
0.001	100	70.09623	68.56696
0.001	1000	55.90847	55.56655
0.001	10000	53.77071	53.45386
0.005	100	59.11107	58.13960
0.005	1000	53.79459	53.51341
0.005	10000	53.69891	53.36556
0.01	100	$3.115\,97 \times 10^{42}$	$3.115\,72 \times 10^{42}$
0.01	1000	∞	∞
0.01	10000	∞	∞

Table 6: Training Results

- As the iteration number increasing in the accurate range, the performance improves to some extent.
- When the learning rate is too large (larger than or equal to 0.01 literally), the model will diverge when there's more than 100 epochs.
- In the accurate range as the learning rate increasing, it accelerate the speed of the convergence of the model, and the model performs better.