The Chinese University of Hong Kong, Shenzhen
School of Data Science

DDA 3020 · Homework 1
Due: 23:59, March 9th, 2024

Instructions:

- This assignment accounts for 14/100 of the final score.
- You must independently complete each assignment.
- Late submission will get discounted score: 20 percent discount on $(0, 24]$ hours late; 50 percent discount on $(24, 120]$ hours late; no score on late submission of more than 120 hours.

---

1 Written Problems (50 pts.)

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.

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.

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 $P = Q$.

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.

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.

2 Coding Problems (50 pts.)

Problem 1 (25pts)
The *house_prices.csv* file contains the house price dataset for regression. There are 128 samples with 5 features. The following describes the dataset features:

- SqFt - The area of house (square feet)
- Bedrooms - The number of bedrooms
- Bathrooms - The number of bathrooms
- Neighborhood - The neighborhood where the house is located

- Price - The price of the house

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

- **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.

- **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. Briefly analyze the potential patterns between "Price" and other attributes.

- **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.

- **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 predicting ($model.predict$) on train and test dataset, we could use *mean_squared_error* function in *sklearn.metrics* to evaluate the performance of fitted model. Report the training error and testing error in terms of RMSE.

Problem 2 (25pts).
The diebetes dataset is available in sklearn package for linear regression. We use *load_diabetes* function in *sklearn.datasets* to load the dataset. There are 442 samples obtained from diabetes patients with 11 attributes. The first 10 attributes are baseline variables, age, sex, body mass index, average blood pressure, and six blood serum measurements. These 10 attributes have been mean centered and scaled by the standard deviation times the square root of the number of samples. The last attribute 11 is a quantitative measure of disease progression one year after baseline. We want to use attributes 1-10 to predict attributes 11.

- **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 parameters with gradient descent methods. (Notice: Do not use the linear regression packages of Sklearn).

- **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.

- **Step 3:** Repeat the splitting, training, 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.

Submission Format

- code.ipynb (without Input Data included) - your jupyter notebook files should **contain the running output** of each step (numbers, plots, etc.). If your notebook has only code but no output results, you will get a discounted score.
- Submit report containing task description, dataset, models, implememt steps, settings, important outputs (errors, plots, figures), and relevant analysis required in above steps. This should be included as part of written assignment in pdf file.
- Note: Please include all the results from your model in the report. You will receive no credits if we have to run the code to get outputs. The recommended length of the report is about 3-5 pages. If the report is too short, the score will be deducted for lacking sufficient contents. There is also no credit bonus for too long reports. The overall submission format is **one ipynb file** and **one pdf** file containing both answers for both written and report.

File list

- Datasets: *house_prices.csv* for coding problem 1. *diabetes.csv* for coding problem 2.
- *LR_example.ipynb*: example for linear regression for beginners, including basic operations such as load datasets, sklearn usage, etc.
- $DDA3020\_2024\_spring\_homework1.pdf$: PDF file of homework.
- Report template: $report\_example1.pdf$, $report\_example2.pdf$.