CS4013/5013 Assignment 4

Fall 2024

Due Date: Oct 20, 2024.

In this assignment, we will implement some simple machine learning processes in Python and report observations. Below is a concise description of each programming task. We will provide one template for each task. See detailed instructions in the templates.

Part I. Regression

Task 1. Implement the learning process of a regression model and report the impact of training data size on the model's prediction performance in Figure 1. Specifically, the x-axis of this figure is percent of data used for training and the y-axis is prediction error (mean-squared-error). Your figure should contain two curves: one is for training error and the other is for testing error.

Goal: you should aim to observe overfitting in Figure 1.

Task 2. Implement the learning process of a regression model and report the impact of hyper-parameter on the model's prediction performance in Figure 2. Specifically, the x-axis of this figure is hyper-parameter value and the y-axis is prediction error. Your figure should contain two curves: one is for training error and the other is for testing error.

Goal: you should aim to observe both overfitting and underfitting in Figure 2.

Task 3. Implement the k-fold cross-validation technique and apply it to select an optimal hyper-parameter for a regression model. (You need to the data splitting, training and evaluation process by yourself, not to call a cross-validation function.) Pick 5 candidate values for the hyper-parameter and report the k-fold cross validation error of each value in Table 1.

Hyper-Parameter					
Validation Error					

Table 1:

Goal: your optimal error should occur when the hyper-parameter is neither too small or large.

Part II. Classification

Task 4. Implement the learning process of a classification model and report the impact of training data size on the model's prediction performance in Figure 3. Specifically, the x-axis of this figure is percentage of data used for training and the y-axis is prediction error (not mean-square-error but classification error). Your figure should contain two curves: one is for training error and the other is for testing error.

Goal: you should aim to observe overfitting in Figure 3.

Task 5. Implement a learning process for classification model on an unbalanced data set and evaluate model performance (both classification error and AUC score).

In addition, you need to develop your own method to improve the AUC score while maintaining classification error as much as possible. A useful reference is "Haibo He and Edwardo A. Garcia, Learning from Imbalanced Data, IEEE Transactions on Knowledge and Data Engineering, 2009". (pdf is in the assignment folder)

You should report testing performance of both the baseline method and your method in Figure 4 and Figure 5. Figure 4 shows model accuracy versus training data size, while Figure 5 shows model AUC score versus training size.

Goal: your method's AUC curve should be higher than the baseline's curve, and your method's accuracy curve should be as close to the baseline's curve as possible.

Bonus Point: if your method has some hyper-parameter and you can show a figure demonstrating the (reasonable) impact of that hyper-parameter on your model AUC score in Figure 6, you get a bonus point.

Submissions Instructions

You should generate 5 (or 6) figures and 1 table for the programming tasks.

You should place all figures and the table in a pdf file named 'hw5.pdf' and upload it to Canvas through the submission page for hw5.

You also need to upload the code that generate each figure, including

– hw4_task1.py for Figure 1

– hw4_task2.py for Figure 2

– hw4_task3.py for Table 1

– hw4_task4.py for Figure 3

– hw4_task5.py for Figure 4 and Figure 5

– hw4_task5b.py for Figure 6 (if you have)

You can generate the pdf file using any tool, although you are encouraged to generate it using Overleaf. A latex template 'hw4_Latex.txt' will be provided to you.