UCLA CS97 Homework Assignment 4

2024 Summer

July 3, 2024

0 Instruction and Preparation

Due date: Friday, 7/5 at 10:00pm PT

Instructions: Be sure to clearly label where each problem and sub-problem begins. All problems must be submitted in order.

Goal: This homework aims to help you go through main concepts after HW3 with toy examples.

1 (20pt) Classification Evaluation

For binary classification, a confusion matrix (table) records the number of True Positives (TP), False Negatives (FN), False Positives (TP), and True Negatives (TN). Based on a confusion matrix, different evaluation metrics can be computed. Assuming a logistic regression model gives us the following predicted probability for heart disease, work out the following exercises.

Observed class label (Y)	Predicted probability for heart disease ($P(Y = 1)$)
1	0.9
1	0.8
0	0.7
1	0.65
1	0.6
0	0.55
0	0.45
0	0.40
1	0.35
0	0.30

Table 1.1: A table on observed class label and predicted probability for heart disease. The positive class (with disease) is denoted as 1, and the negative class (with no disease) is denoted as 0.

(5pt) Exercise: Given a cut-off probability 0.5, i.e., classify a data point into positive class if the predicted probability is bigger than or equal to 0.5, and classify a data point into negative class if the predicted probability is less than 0.5, fill in the confusion table below.

	Predicted class = 1	Predicted class = 0
Actual class = 1		
Actual class = 0		

Table 1.2: Confusion Table

(5pt) Exercise: Calculate Accuracy, Error, Precision, Recall, and F_1 score.

(5pt) Exercise: Suppose a different logistic regression model gives us a different prediction result, and

the class labels for the sorted training point (descending with predicted probability) are shown below (Table 1.3). What will the ROC curve look like? What is the area under ROC curve in this case?

Observed class label ($Y = 1$)	Predicted probability for heart disease ($P(Y = 1)$)
1	0.95
1	0.85
1	0.75
1	0.65
1	0.6
0	0.55
0	0.45
0	0.40
0	0.35
0	0.30

Table 1.3: A table on observed class label and predicted probability for heart disease. The positive class (with disease) is denoted as 1, and the negative class (with no disease) is denoted as 0.

(5pt) Exercise: If a binary classification task has an imbalanced label, which evaluation metrics is better, accuracy or area under ROC curve? Why? (Hint: consider a case where only 1% of the data points are positive and a random classifier.)

2 (20pt) K Nearest Neighbor (KNN)

KNN is an algorithm that makes prediction based on a data point's K nearest neighbors.

2.1 KNN for Regression

(10pt) Exercise: Consider the house price dataset in Table 2.1, and answer the following questions. What are the 2 nearest neighbors for a new house with living area 1700 sq.ft. and 3 bedrooms, in terms of Euclidean distance? Based on these two neighbors, what is your prediction for its price?

area (100 sq.ft.)	bedroom	price ($100K)
15	2	5
20	3	7
18	3	6
16	3	5.5

Table 2.1: House Price Training Dataset

2.2 KNN for Classification

(5pt) Exercise: Consider the credit card transaction dataset in Table 2.2, for a new transaction with $250 amount, what are the 3 nearest neighbors to it? Based on these 3 nearest neighbors, what is its probability to be a fraud? Based on this probability, do you classify it into fraud or not?

2.3 Selecting K

(5pt) Exercise: When K becomes bigger, does the model become more complex or simpler? For the classification case, when K becomes bigger, does the decision boundary become smoother or more bumpy? How do you decide K in practice?

amount ($100)	fraud
0.1	0 (No)
1	0 (No)
2	0 (No)
3	1 (Yes)
5	1 (Yes)
10	1 (Yes)

Table 2.2: Credit Card Transaction Dataset

3 (20pt + 5 bonus pt) Support Vector Machine (SVM)

Linear SVM aims at finding the best linear form decision boundary to separate two classes. In the case where we have two predictors, X_1 and X_2, the decision boundary can be written as a line:

$$w_1 X_1 + w_2 X_2 + b = 0 \tag{1}$$

The closest data points to the decision boundary are called *support vectors*. The goal is to maximize the distance between support vectors to the decision boundary, or equivalently the distance between two lines spanned by support vectors in each of the two classes, which is called *margin*.

3.1 Linear SVM

(5pt) Exercise: In the linearly separable case, where positive class and negative class points can be completely separated by a line, how many lines are there with a training error 0?

(5pt) Exercise: Suppose there are two lines, one is with margin 5 and the other is with margin 4, for the same training dataset, which one do you want to choose? Why?

(5pt) Exercise: Suppose we have a decision boundary $X_1 + X_2 - 1 = 0$ for a binary classification task, and a data point with $x_1 = 1$ and $x_2 = 1$, which class should we classify it into, positive or negative? Why?

(5pt) Bonus Exercise: Suppose we have identified three support vectors through some optimizer, which are (1,0), (0,1) with positive label, and (0,0) with negative label. Can you write down the decision boundary based on these three support vectors? (Hint: (1) What is the line spanned by (1,0) and (0,1)? (dotted lines in our lecture slides) (2) what is the line for the negative support vector? Remember the two lines have to be parallel to each other. (3) What is the decision boundary, which has to be in the middle of those two lines?)

3.2 Kernel SVM

When the dataset is not linearly separable, meaning there is no way to use a line to partition data points from two classes into two sides, we can use kernel SVM. The idea behind Kernel SVM is to transform a data point to an even higher dimensional space. For example, from (x_1, x_2) to $(x_1, x_2, \sqrt{x_1^2 + x_2^2})$. This transformation might be very complicated and hard to design. In practice, what we really need to know is a kernel function that tells us the similarity between any two data points. For example, we can define the similarity between two data points $\boldsymbol{x}_i = (x_{i1}, x_{i2})$ and $\boldsymbol{x}_j = (x_{j1}, x_{j2})$ using polynomial kernel with degree 2:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (x_{i1}x_{j1} + x_{i2}x_{j2} + 1)^2 \tag{2}$$

(5pt) Exercise: Given three data points, (1,1), (-1,0), and (-1,1), which point is more similar to (1,1) according to the polynomial kernel with degree 2 defined above?

4 (20pt) Decision Tree

Decision Tree is an algorithm that divides the data into different regions, which correspond to leaf nodes in a tree structure, and the decisions will be made based on the prediction on the specific region a data belongs to.

To construct a tree, we need to choose the "best" predictor to split the data. If the predictor is categorical, the splitting is based on the categorical values; if the predictor is numerical, we also need to decide the cut-off value.

(10pt) Exercise: Consider a dataset with two predictors Ages and Credit_Rating, with the goal to classify whether a person is going to purchase xbox. The class labels are shown for data points in each partition after splitting for both Ages and Credit_Rating in Fig. 4.1. (1) Calculate weighted average classification error for both Ages and Credit_Rating, with the help of Table 4.1 and Table 4.2. (Some of the numbers are provided.) (2) Which predictor do you want to choose for the splitting?

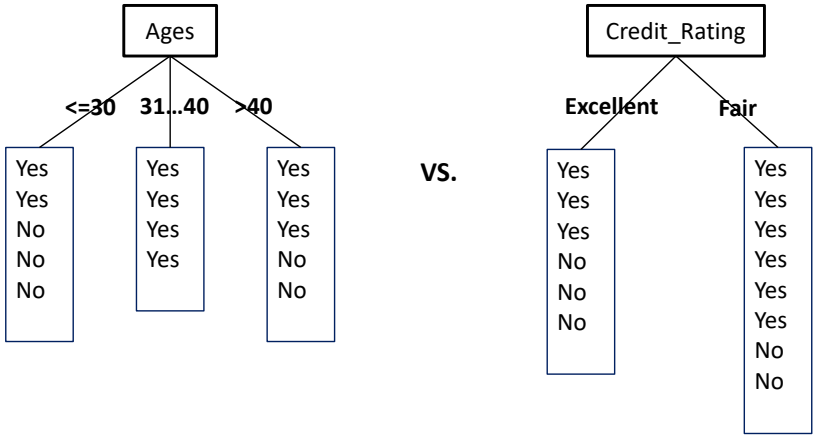

Figure 4.1: Predictor Selection for Splitting

	Class = Yes	Class = No	Classification Error for each Region	weight of each region
R_1	2	3	2/5	5/14
R_2				
R_3				

Table 4.1: Help table for weighted average classification error calculation for Ages

	Class = Yes	Class = No	Classification Error for each Region	weight of each region
R_1				
R_2				

Table 4.2: Help table for weighted average classification error calculation for Credit_Rating

(10pt) Exercise: Given a regression tree as shown in Fig. 4.2, predict the salary for a baseball player with 5 years service and 150 hits. (left branches mean yes and right branches mean no.) Explain your reasons.

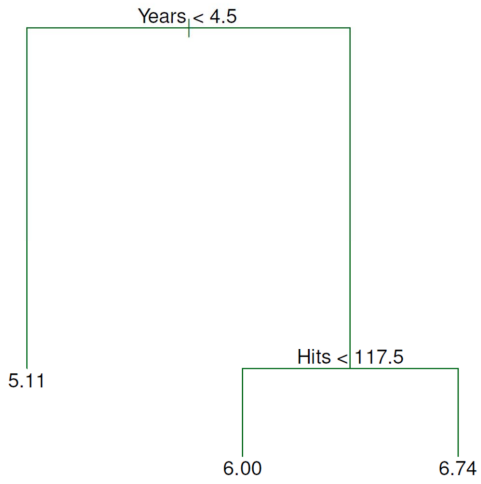

Figure 4.2: A Regression Tree for *Salary* Prediction Based on *Hits* and *Years*.

5 (20pt) Neural Networks

A neural network defines a function that maps an input vector to an output vector. The function is determined by the architecture of the neural network, and the activation functions of each neuron. The parameters of the network are weights on the connections between neurons as well as the bias terms associated with each neuron. Training a neural network involves (1) a forward step, where predictions are calculated in a forward fashion for a mini-batch of inputs; and (2) a backpropagation step, where predictions are compared with the observed labels and the loss is propagated backward to update parameters involved in each layer. The two steps are repeated iteratively until convergence or early stopping (based on validation dataset).

(10pt) Exercise: For a fully connected multi-layer feed-forward neural network with 1 input layer (3 units), 2 hidden layers (4 units and 4 unites), and one output layer (3 units), how many parameters are there in total?

(10pt) Exercise: Given a neural network architecture shown in Fig. 5.1, weights and bias shown in Table 5.1, and activation function for all neurons as ReLU function, compute the output for the input data point (0,1).

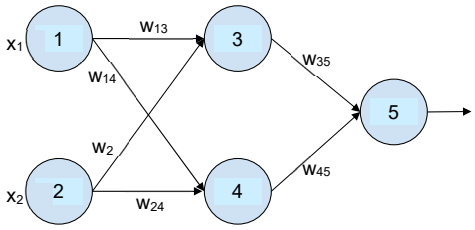

Figure 5.1: Fully Connected Multi-layer Neural Network. Number in each neuron denotes its index.

w_{13}	w_{14}	w_{23}	w_{24}	w_{35}	w_{45}	b_3	b_4	b_5
-0.3	0.2	0.4	-0.1	-0.2	-0.3	0.2	-0.4	0.1

Table 5.1: Weights and biases for each connection and neuron.