Machine Learning

Homework 1

Faculty of New Sciences & Technologies
University Of Tehran
Fall 2023

Part 1

General Homework Policies

1. Due date of this homework is on *Sunday 5 Aban 402*, so you need to submit it before the due date[midnight 5 Aban], otherwise you won't get the total score! We consider the following policy for the late homework,
   - Homework is worth full credit at the beginning of class on the due date.
   - It is worth half credit for the next 48 hours.
   - It is worth zero credit after that.
2. You are welcome to collaborate, cooperate, and consult with your classmates provided that you write-up the solutions independently.
3. Don't plagiarize! Write everything in your own words, and properly cite every outside source you use. Taking credit for work as well as ideas that are not your own is plagiarism. Students who plagiarize will not get any score and they will be introduced in the class.
4. Please create reference for all sources(books, papers, websites) which you use.
5. Please create a cover letter for your report which simply is the Homework#, title of the course, your name, surname, and student number.
6. You may post questions asking for clarifications and alternate perspectives on concepts on piazza or in the class.
7. Upload your final file of assignment on the course website at UT elearn by naming style as [PRML_2023_hw# Surname] which # indicates number of the homework.

Part 2

Questions

Comments on the dataset

Based on students request about the size of CDC Diabetes Health Indicators dataset, you can use *Employee_Data_Classification* which is uploaded in UT Elearn. Please check the dataset description at Employee_Data_Classification. The dataset comprises 7 features and a label feature named as "LeaveOrNot". Also, you can skip the imbalanced items in following questions.
The added points are considered for students who use CDC Diabetes Health Indicators dataset.

1 k-NN implementation

(80 points, 50 points for parts i to vi and 30 points for part vii) A nearest neighbor classifier requires a parameter (the number k of neighbors used to classify). We will use cross validation to select the value of k for a specific type of data, CDC Diabetes Health Indicators. You can Download the data set from the course website at elearn and the details of the dataset is given at diabetes dataset. *diabetes_ 012_health_indicators_BRFSS2015.csv* a clean dataset of 253,680 survey responses to the CDC's BRFSS2015. The target variable *Diabetes_012* has 3 classes. 0 is for no diabetes or only during pregnancy, 1 is for prediabetes, and 2 is for diabetes. There is class imbalance in this dataset. This dataset has 21 feature variables.

(i) Read in the data into Python from the dataset.

(ii) Plot the histogram or barchart to represent the dataset containing label and features.

(iii) Randomly split the data into,

- Training data (80% of the entire data set).
- Test data (20%).

(iv) Train a 1-nearest neighbor classifier using the training data and predict the labels in the testing data. What is the test error (the empirical error rate on the test set)?

(v) Select k as follows:

- For $k \in \{1; 3; 5; 7; 9; 11; 13\}$, train the k-nn classifier(that you've written) on the training data and classify the samples in the test set. Compute the test error for each k.
- Which value of k did you choose? Why?
- Compute the error rate of the classifier for the optimal value of k on the validation set.

(vi) There is a clean balanced dataset available on diabetes dataset. Compare the results of imbalanced data to the balanced one based on the chosen k in part v. (optional points: create a balanced data by yourself based on some well-known techniques such as dealing with imbalanced data). Is there a significant difference between the obtained results for imbalanced data and balanced one? Try to figure out the optimal k in part v using balanced data.

(vii) (30 points)The aim of this part is to use some strategies to improve the computational cost of nearest neighbor classifier. (Use the balanced data for this part.)

a. Using a Priority Queue Heap Data structure: First initialize the heap with the k arbitrary points from the training dataset based on their distances to the query point. Then, as we iterate through the dataset to find the first nearest neighbor of the query point, at each step, we make a comparison with the points and distances in the heap. If the point with the largest stored distance in the heap is farther away from the query point that the current point under consideration, we remove the farthest point from the heap and insert the current point. Once we finished one iteration over the training dataset, we now have a set of the k nearest neighbors. Rewrite your knn using heap data structure and compare your running time performance on this data set with the ordinary knn with $k = 5$.

b. Use k-d tree data structure K-d Tree to improve the efficiency of your implementation and compare the obtained running time to the previous parts by considering $k = 5$.

viii (optional 10 points) Because of the large number of samples for this dataset, the computational complexity is a bit large. Try to use some dimension reduction techniques and then apply k-nn method. What is the difference between attained running time to the original dataset versus the reduced dataset? (with $k = 5$)

2 Probability theory

(20 points)

(i) Solve exercise 2.7 of [1].

(ii) (optional) Solve exercise 2.14 of [1].

Bibliography

[1] Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2007, chapter 2.

Good Luck!