Assignment 1

CS6510: Applied Machine Learning
IIT-Hyderabad
Jan-Apr 2019

Max Marks: 40
Due: 4 Feb 2019 11:59 pm

This homework is intended to cover programming exercises in the following topics:

- k-NN, Decision Trees, Model Selection, Naive Bayes classifier

Instructions

- Please use Google Classroom to upload your submission by the deadline mentioned above. Your submission should comprise of a single file (PDF/ZIP), named `<Your_Roll_No>_Assign1`, with all your solutions.
- For late submissions, 10% is deducted for each day (including weekend) late after an assignment is due. Note that each student begins the course with 7 grace days for late submission of assignments. Late submissions will automatically use your grace days balance, if you have any left. You can see your balance on the `CS6510 Marks and Grace Days` document under the course Google drive (soon to be shared).
- You have to use PYTHON for the programming questions.
- Please read the department plagiarism policy. Do not engage in any form of cheating - strict penalties will be imposed for both givers and takers. Please talk to instructor or TA if you have concerns.

1 Questions

1. **k-NN:** ***(7 marks)*** In k-nearest neighbors (k-NN), the classification is achieved by majority vote in the vicinity of data. Given n points, imagine two classes of data each of $n/2$ points, which are overlapped to some extent in a 2-dimensional space.

   (a) ***(1 mark)*** Describe what happens to the training error (using all available data) when the neighbor size k varies from n to 1.

   (b) ***(2 marks)*** Predict and explain with a sketch how the generalization error (e.g. holding out some data for testing) would change when k varies? Explain your reasoning.

(c) ***(2 marks)*** Give two reasons (with sound justification) why k-NN may be undesirable when the input dimension is high.

(d) ***(2 marks)*** Is it possible to build a univariate decision tree (with decisions at each node of the form is $x > a$, is $x < b$, is $y > c$, or is $y < d$ for any real constants a, b, c, d) which classifies exactly similar to a 1-NN using the Euclidean distance measure ? If so, explain how. If not, explain why not.

2. **Bayes Classifier:** ***(6 marks)***

(a) ***(3 marks)*** A training set consists of one dimensional examples from two classes. The training examples from class 1 are $\{0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.35, 0.25\}$ and from class 2 are $\{0.9, 0.8, 0.75, 1.0\}$. Fit a (one dimensional) Gaussian using Maximum Likelihood to each of these two classes. You can assume that the variance for class 1 is 0.0149, and the variance for class 2 is 0.0092. Also estimate the class probabilities p1 and p2 using Maximum Likelihood. What is the probability that the test point $x = 0.6$ belongs to class 1?

(b) ***(3 marks)*** You are now going to make a text classifier. To begin with, you attempt to classify documents as either sport or politics. You decide to represent each document as a (row) vector of attributes describing the presence or absence of the following words.

x = (goal,football,golf,defence,offence,wicket,office,strategy)

Training data from sport documents and from politics documents is represented below in a matrix in which each row represents the 8 attributes.

$$x_{\text{politics}} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$x_{\text{sport}} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Using a maximum likelihood naive Bayes classifier, what is the probability that the document $x = (1, 0, 0, 1, 1, 1, 1, 0)$ is about politics?

3. **Decision Trees:** ***(15 marks)*** In this question, you will use the Wine dataset[1], a popular dataset to evaluate classification algorithms. The classification task is to determine, based on various parameters, whether a wine quality is over 7. The dataset has already been preprocessed to convert this into a binary classification problem (scores less than 7 belong to the "*zero*" class, and scores greater than or equal to 7 belong to the "*one*" class). Each line

[1] https://archive.ics.uci.edu/ml/datasets/Wine+Quality

describes a wine, using 12 columns: the first 11 describe the wines characteristics (details), and the last column is a ground truth label for the quality of the wine (0/1). You must not use the last column as an input feature when you classify the data.

(a) ***(5 marks)*** **Decision Tree Implementation:** Implement your own version of the decision tree using binary univariate split, entropy and information gain. (If you are using Python, you can use the skeleton code provided.)

(b) ***(5 marks)*** **Cross-Validation:** Evaluate your decision tree using 10-fold cross validation. Please see the lecture slides for details. In a nutshell, you will first make a split of the provided data into 10 parts. Then hold out 1 part as the test set and use the remaining 9 parts for training. Train your decision tree using the training set and use the trained decision tree to classify entries in the test set. Repeat this process for all 10 parts, so that each entry will be used as the test set exactly once. To get the final accuracy value, take the average of the 10 folds accuracies. With correct implementation of both parts (decision tree and cross validation), your classification accuracy should be around 0.78 or higher.

(c) ***(5 marks)*** **Improvement Strategies:** Now, try and improve your decision tree algorithm. Some things you could do include (not exhaustive):

- Use Gini index instead of entropy
- Use multi-way split (instead of binary split)
- Use multivariate split (instead of univariate)
- Prune the tree after splitting for better generalization

Report your performance as an outcome of the improved strategies.

Deliverables:

- Code
- Brief report (PDF) with: (i) Accuracy of your initial implementation; (ii) Accuracy of your improved implementation, along with your explanation of why the accuracy improved with this change.

4. ***(12 marks)*** **Kaggle - What's cooking:** The next task of this assignment is to work on a (completed) Kaggle challenge: "what's cooking?" As part of this task, please visit https://www.kaggle.com/c/whats-cooking to know more about this problem, and download the data. (You may have to create a Kaggle account to download the data, if you don't have one already.)

   In this assignment, you are allowed to use any existing machine learning library of your choice: scikitlearn, pandas, Weka (we recommend `scikitlearn`) - but you should use only the decision tree, k-NN or the naive Bayes classifier (to align with what we have covered in class so far, random forests not allowed too). Use `train.json` to train your classifier. Predict the cuisine on the data in `test.json`, and report your best 2 scores in your report. (Note that Kaggle will not publish the scores of a completed contest on its leaderboard, but will reveal the scores to you - please report them. We will also upload your codes randomly to confirm the scores.)

   Deliverables:

   - Code

- Brief report (PDF) with top-2 scores of your methods, and a brief description of the methods that resulted in the top 2 scores.
- Your report should also include your analysis of why your best 2 methods performed better than others you tried.