Assignment 1

CS6510: Applied Machine Learning
IIT-Hyderabad
Aug-Nov 2018

Max Points: 60
Due: 16 Sep 2018 11:59 pm

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

- k-NN, Decision Trees, Model Selection

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 10 grace days for late submission of assignments. Late submissions will automatically use your grace days balance, if you have any left.
- We recommend using PYTHON for the programming assignments, although you can use Java/C/C++/R if you like too.
- 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. **Decision Trees:** You have been provided with 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 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.

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

(a) ***(10 points)*** **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) ***(10 points)*** **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) ***(15 points)*** **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.

2. ***(25 points)*** **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 or the k-NN classifier (to align with what we have covered in class so far). Use `train.json` to train your classifier (decision tree or k-NN, no random forests too). 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 if required.)

   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.

2 Theory and Practice Questions (No submission required)

The questions below are only for your practice - no submission required.

1. **Numpy:** The following questions should be done using `numpy` operations. Each exercise can be solved with a maximum of 4-5 lines of Python code. Please avoid the use of iterative constructs (such as for loops) to the extent possible, and use matrix/vector operations to achieve the objectives.

   (a) Import the numpy package under the name np. Print the numpy version and the configuration.

   (b) Create a null vector of size 10, and output the vector to the terminal.

   (c) Create a null vector of size 10 but the fifth value which is 1. Output the vector to the terminal.

   (d) Reverse a vector (first element becomes last).

   (e) Create an $n \times n$ array with checkerboard pattern of zeros and ones.

   (f) Given an $n \times n$ array, sort the rows of array according to m^{th} column of array.

   (g) Create an $n \times n$ array with $(i+j)^{th}$-entry equal to $i+j$.

   (h) Consider a $(6, 7, 8)$ shape array, what is the index (x, y, z) of the 100th element (of the entire structure)?

   (i) Multiply a 5×3 matrix by a 3×2 matrix (real matrix product).

   (j) Create random vector of size 10 and replace the maximum value by 0.

   (k) How to find the closest value (to a given scalar) in an array?

   (l) Subtract the mean of each row from each corresponding row of a matrix.

   (m) Consider a given vector, how to add 1 to each element indexed by a second vector (be careful with repeated indices - you should consider it only once)?

   (n) How to find the most frequent value in an array?

   (o) Extract all the contiguous 3×3 blocks from a random 10×10 matrix.

   (p) Compute the rank, trace and determinant of a matrix.

2. **k-NN:** 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) Describe what happens to the training error (using all available data) when the neighbor size k varies from n to 1.

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

Price	Maintenance	Capacity	Airbag	Profitable
Low	Low	2	No	Yes
Low	Med	4	Yes	No
Low	Low	4	No	Yes
Low	High	4	No	No
Med	Med	4	No	No
Med	Med	4	Yes	Yes
Med	High	2	Yes	No
Med	High	5	No	Yes
High	Med	4	Yes	Yes
High	High	2	Yes	No
High	High	5	Yes	Yes

(d) 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.

3. **Decision Trees:** Consider the dataset in the table above:

(a) Considering 'profitable' as the binary-valued attribute we are trying to predict, which of the attributes would you select as the root in a decision tree with multi-way splits using the entropy-based impurity measure?

(b) For the same data set, suppose we decide to construct a decision tree using only binary splits and the Gini index impurity measure. Which feature and split point combination would be the best to use as the root node assuming that we consider each of the input features to be unordered?