CS 434: Assignment 3

Due April 28th 11:59PM, 2017

General instructions.

1. The following languages are acceptable: Java, C/C++, Matlab, Python and R.
2. You can work in team of up to 3 people. Each team will only need to submit one copy of the source code and report.
3. You need to submit your source code (self contained, well documented and with clear instruction for how to run) and a report via TEACH. In your submission, please clearly indicate your team members' information.
4. Be sure to answer all the questions in your report. Your report should be typed, submitted in the pdf format. You will be graded based on both your code as well as the report. In particular, the clarity and quality of the report will be worth 10 % of the pts. So please write your report in clear and concise manner. Clearly label your figures, legends, and tables.

1 Part I: Model selection for KNN

The dataset constitutes of 30 features and is a matrix of $N \times 31$ dimension. The first column of the test and train data is the true class label of the samples and the remaining columns are the features. This dataset belongs to the Wisconsin Diagnostic Breast Cancer dataset and the classes are the diagnosis (+1= malignant and -1= benign). Do not forget that features should be normalized to have the same range of values (e.g., [0,1]), otherwise features with larger ranges will have higher impact on the distance.

1. (15 pts) Implement the K-nearest neighbor algorithm. Ideally, you want to implement this algorithm as a subroutine with K as a parameter.
2. (20 pts) Consider the following range of K values: 1, 3, 5,..., 51. (This is a suggested range, feel free to explore more possible K values). For each possible value of K, please compute the following: 1) the training error (measured as the number of mistakes) 2) the leave-one-out cross-validation error on the training set; and 3) the number of errors on the provided test data. Plot these three errors as a function of K.

3. (15 pts) Discuss what you observe in terms of the relationship between these three different measure of errors. Perform model selection. What is your choice of K?

2 Part II: Decision tree

For this part, please, use dataset provided for Part I.

1. (15 pts) Implement the algorithm for learning decision stump, i.e. a decision tree with only a single test. To build a decision stump, simply apply the top down decision tree induction algorithm to select the root test and then stop and label each of the branches with its majority class label. For this assignment, please only use binary split. Please use the information gain as the selection criterion for building the decision stump. Provide the learned stump, and information gain. Provide the training and testing error rate of the learned stump.

2. (35 pts) Implement the top-down greedy induction algorithm for learning decision tree with depth $d = 6$ (where level 1 is a root of the tree and level 6 contains the leaves). Please use the information gain as the selection criterion for building the decision tree. For this assignment, please only use binary split. Provide your learned decision tree and for each test its information gain. Provide the training and testing error rate of the learned tree. Compare decision stump rates with decision tree rates. What behavior do you observe? Explain.

Remark 1. *We strongly suggest you to create a separate function to implement both the decision stump and the decision tree.*

3 Part III: extra credit

- (15 pts) Explore the ways you can use the results from Part II to improve KNN.