CS 434: Implementation Assignment 2

Due April 24th 11:59PM, 2019

General instructions.

1. The assignment should be implemented in Python 3.
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 all your team members.
4. Your code will be tested on flip, please make sure that your program can run on it without any issue.
5. 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

Dataset. This dataset belongs to the Wisconsin Diagnostic Breast Cancer dataset and the classes are the diagnosis (+1= malignant and -1= benign). It constitutes of 30 features and is a matrix of $N \times 31$ dimension. The first column of the test and train data contains the true class label of the samples and the remaining columns are the features. Note 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 computation, and on the final prediction.

1. (10 pts) Implement the K-nearest neighbor algorithm, where K is a parameter.
2. (15 pts) Test your KNN with 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. (10 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?

Write your code so that you get the results for questions 1_1 and 1_2 using the following command:

python q1.py train.txt test.txt k

The output should include:

- The training error: percentage of training examples incorrectly classified.
- The leave-one-out cross-validation error.
- The testing error.

2 Part II: Decision tree

For this part, please, use the same dataset that is provided for Part I.

1. (20 pts) Implement the algorithm for learning a 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 use the information gain as the selection criterion (i.e., using entropy to measure uncertainty) for building the decision stump and consider only binary splits. In your report, please provide 1) the learned decision stump (be sure to provide the label for each branch); 2) the computed information gain of the selected test, 3) the training and testing error rates (in percentage) of the learned decision stump.

   Write your code so that you get the results for questions 2_1 using the following command:

   python q2_1.py train.txt test.txt

   The output should clearly include:

   - The learned decision stump.
   - The computed information gain values.
   - The training error.
   - The testing error.

2. (25 pts) Implement the top-down greedy induction algorithm using the information gain criterion for learning a decision tree from the training data with a fixed depth limit given by d (note when $d = 1$ this is the decision stump). Consider the following choices for $d \in \{1, 2, 3, 4, 5, 6\}$. Similarly we will only consider binary splits. Please apply the learned

decision trees to the training and testing data and report in a table the training and testing error rates of the learned decision trees for different d values. Plot the error rates as a function of d. What behavior do you observe? Provide an explanation for the observed behavior.

Write your code so that you get the results for questions 2_1 using the following command:

python q2_2.py train.txt test.txt d

The output should clearly include:

- The training error.
- The testing error.