CS5785 Homework 2

Due date: October 29

The homework is split into programming exercises and written exercises.You should turn in an electronic copy of your solutions to the homework. Please submit your homework to CMS. You are responsible for submitting clear, organized answers to the questions. Please include all relevant information for a question, including text response, equations, figures, graphs, etc. Please pay attention to the discussion board for relevant information regarding updates, tips, and policy changes. You are required to work in groups of 2 (unless given an exemption from the course staff). Version 1.

1 Programming Exercises

1. In this problem we are going to use k-nearest neighbors for prediction of taxi trip times. The datasets used in this assignment will be same as those used in Homework 1. Download the additional code provided on CMS.

   (a) Using train_data.csv perform density estimation for

   $P(\textit{passenger\_count} = 1 | \textit{dropoff\_longitude,dropoff\_lattitude})$

   $P(\textit{passenger\_count} = 3 | \textit{dropoff\_longitude,dropoff\_lattitude})$

   Plot and compare both densities. Describe the method you used for performing density estimation (it could be nearest neighbors, or a parametric method).

   (b) Implement a 1-nearest neighbor prediction system for predicting taxi trip time, using pickup and drop off lattitudes and longitudes. Using train_data.csv as the training set, and first hundred thousand trips from trip_data_1.csv as the test set, calculate Root Mean Squared Error, Correlation Coefficient, and the Mean Absolute Error between expected and predicted trip times.

   (c) Extend the model above to incorporate pickup_time (with month and year stripped) and trip_distance. Using train_data.csv as the training set, and the first hundred thousand trips from trip_data_1.csv as the test set, calculate Root Mean Squared Error, Correlation Coefficient, Mean Absolute Error between expected and predicted trip times.

   (d) Unlike linear regression, nearest neighbors is scale dependent. Find a scaling method for pickup and drop off lattitudes and longitudes, trip_distance and pickup_time, which will improve performance. Use train_data.csv as training set, and first ten thousand trips from trip_data_1.csv as test set. Use Mean Absolute Error between expected and predicted trip times as objective. Use 1-nearest neighbors.

   (e) Use the scaling parameters determined from the model above, and find the optimal value of k betwen [5 , 20]. Use median voting. Use train_data.csv as training set, and first ten thousand trips from trip_data_1.csv as test set. Use Mean Absolute Error between expected and predicted trip times as objective.

(f) Computation of distance between test and all example in training set is computationaly expensive. Describe one method for implementing approximate nearest neighbors on model defined in [A]. [Karma problem: Implement the method.]

Karma Using the Map Reduce framework *YELP MRJob* provided, write code for Mapper, Combiner and Reducer for 10nearest neighbors. Use only pickup and drop off lattitudes and longitudes as features. Use train_data.csv as the training set, and the first ten thousand trips from trip_data_1.csv as the test set. Run this code locally. Compare the results with your code from part 1. [Hint: Assume that training data is stored on HDFS and test data is sent to all mapper processes]

Karma Run the above code on an AWS EMR cluster.

2. In this problem, we are going to explore decision trees to predict ratings from Amazon food reviews. Go to `http://snap.stanford.edu/data/web-FineFoods.html` and download the dataset. Download the additional code provided on CMS.

   (a) Print out one example of a review and label pair. Look at several more and write a sentence or two about what is inherent to a positive review and what is inherent to a negative review.

   (b) One of the simplest techniques in natural language processing (the analysis of plaintext) is look at text as a "bag of words". In this view, we ignore grammar and analyze text by the frequency of the words which appear in it. This is the technique (simplifying assumption) we will use for this problem. An n-gram is a continugous set of n itmes. 1-grams have a special name: unigram. As a first attempt to analyze a plain-text review, we are going to consider its unigrams. Look into `utils.py` and use the methods there to extract unigrams from reviews. Let's first try to analyze the unigrams in the dataset. Find and report the 30 most popular unigrams among all reviews, 30 most popular unigrams among the positive reviews, and the 30 most popular unigrams among the negative reviews. Write one or two sentences as to whether or not we can deduce anything from this and why.

   (c) One of the problems you should have found in the last subpart is the presence of common but nonuseful words (for example: the, and, but, a, etc.). Use the flag in the `scan.scan` and after removing stopwords find the 30 most popular unigrams among all reviews, 30 most popular unigrams among positive reviews, and the 30 most popular unigrams among negative reviews.

   (d) The metric we are going to optimize at each step is information gain. For this part, go into `util.py` and write the `entropy` and `information_gain` methods. For more information, see the lecture notes or `http://en.wikipedia.org/wiki/ID3_algorithm`.

   (e) Look at the `decision_tree.py` file. There should be a `DecisionTree` object which contains the basic structure of a tree. In this subpart we will fill in the method `train`. We will write these methods. To train the decision tree, create a list of the top 500 non-stopwords for positive and top 500 non-stopwords for negative. At each step, we will find the word which has the highest information gain after using its presence or absence to split. At each step define the `decision` method so that it works on a generic text review.

   (f) In this subpart, fill in the method `test`. Record the accuracy of the decision tree. Write two or three sentences to describe how well the classifier does and two or three sentences on possible extensions to our feature selection.

Karma Try this for the non-binary case. See the `binary_label` flag in `main` and `scan.scan` methods.

2 WRITTEN EXERCISES

1. In this problem, we are going to explore ROC curves.

   (a) Consider a classifier which takes in any point and flips a fair coin. If the coin ends on H, then the classifier will predict 1. If the coin ends on T, then the classifier will predict 0. Say we test this classifier on a balanced test set (equal number of positive and negatives). Draw the expected ROC point and label it A.

   (b) Additionally, say that we have a series of lucky rolls on one of our test sets and get 30 true positives, 10 false positives, 30 true negatives, and 10 false negatives. Draw this point on our graph and label it B.

   (c) Consider a classifier which has a parameter p. This classifier takes in a point and flips a biased coin which will end on heads with probability p. Again, if the coin ends on H, then the classifier will predict 1. If the coin ends on T, then the classifier will predict 0. Draw the expected ROC curve and label it C.

   (d) Consider a classifier which has a parameter p. This classifier takes in a point and flips a biased coin which will end on heads with probability p. If the coin ends on H, then the classifier will return the correct class. If the coin ends on T, then the classifier will return the wrong class. Draw the ROC curve and label it D.

   (e) If we are able to influence any parameters, which classifier A, C, or D is the best? Briefly explain why.

2. This problem concerns some of the main ideas in statistics.

   (a) A random variable is a variable whose value is subject to variations due to chance (randomness). In other words, a random variable can take one value out of a set and does so with certain probability. One common example of a random variable is the result of a coin flip. Say X is a random variable which represents a fair coin flip. What are the possible values it can take and what is the probability it takes those values?

   (b) The expected value of a random value is intuitively the value it will take in the long run after a lot of repetitions. More precisely we define it in the discrete case as $E[X] = x_1 p_1 + \ldots + x_k p_k$ where X is a random variable, $x_1, \ldots, x_k$ are the values X can take, and p_i is the probability X will take value x_i. If we consider a coin flip and say heads equals 1 and tails equals 0, what is the expected value of a coin flip. Similarly say Y is a random variable which represents a fair 6-sided dice roll. What is $E[Y]$ (the expected value of Y)?

   (c) We can perform most mathematical operations on random variables. Let both X_1 and X_2 be rolls of a 6-sided dice. Let $Y = X_1 + X_2$. First, give an interpretation/explanation for what Y represents. Second, write down $E[Y]$.

   (d) One of the most important/useful facts about expectation is that it is linear, in that $E[X_1 + X_2] = E[X_1] + E[X_2]$. Let's try to prove this. Using the definition of random variable and expectations, write out the full equation for $E[X_1 + X_2]$ and the full equation for $E[X_1]$ and $E[X_2]$. With these values, show that linearity is true as claimed.

   (e) Variance is defined as $Var[X] = E[(X - E[X])^2]$. This can also be re-written as $Var[X] = E[X^2] - (E[X])^2$. What is the variance of a fair 6-sided die?

(f) Using techniques you learned above, prove that $Var[a+X] = Var[X]$. where a is a scalar and X is a random variable.

Karma Problems:
Assignments may include optional problems called karma problems. These problems exist to provide an extra challenge for those who are up to it. Karma problems will be graded, but will not affect your score — they mostly just give good karma. Students who do karma problems will be noticed, and doing them may be taken into account when assigning final grades. For example, karma might make a difference if you are right on the line between a B+ and an AâĂŞ. However, spending your time on regular problems is almost always a more effective way to improve your overall score. Tackle karma problems only after you are sure you have the rest of the assignment well in hand.