ENGS/QBS 108 Fall 2017 Assignment 2

Due October 3, 2017

Instructors: George Cybenko and Saeed Hassanpour

Problem: K-Means Clustering [15 points]. In this problem, you will solve a clustering task using the k-means algorithm and an associated classification task using k nearest neighbors algorithm, both of which you learned in class. The dataset for this problem is a synthetic two-dimensional dataset synth_all.csv. Each entry has two features (x_1, x_2).

1. [5 points] A reasonable first step in every machine learning task is to understand the dataset at hand. Proceed to explore this problem's dataset by addressing the following:
   (a) Choose a suitable type of plot and visualize the training data.
   (b) From your plot, how many clusters, k, would you estimate are represented in the dataset?
   (c) What is the dataset distribution across the k clusters? Use a histogram to illustrate this.
2. [10 points]
   (a) Using the k-Means algorithm, implement a clustering model. You can use Matlab, Python (Scikit-learn) or any other tools at your disposal. Be sure to include code.
   (b) Train the clustering model on several reasonable values of k, taking into account your visual inspection from 1b. Plot the Bayesian information criterion (BIC), Akaike information criterion (AIC), and gap statistic scores for each value of k.
   (c) Which value is optimal? How does it compare to your visual inspection?

Problem: K-NN Classification [10 points]. In this problem, you will utilize data deriving from the same synthetic dataset as above. This time, the data has been separated into synth_train.csv, synth_valid.csv and synth_test.csv files. Furthermore, each sample now includes a class label found in the y column. These class labels come from the set $\{1, 2, \dots, 31\}$.

1. [10 points]
   (a) Train an implementation of the k-Nearest Neighbors algorithm on the training dataset. Note that k here refers to the number of neighbors, not clusters.
   (b) Report the classification accuracy of this model on the validation set for different values for k. Plot these accuracies against k and report the optimal value for k.
   (c) Report the classification accuracy of this model on the data in *synth_test.csv* using the optimal value of k that you found in 1b.

Problem: Decision Tree Classification [20 points]. In this problem you will use decision trees to classify the quality of red vinho verde wine samples based on their physicochemical properties. The dataset has been separated into red_train.csv, red_valid.csv and red_test.csv files. For all of these files, the rightmost column ("quality") is the target label for each datapoint. All other columns are features.

1. [5 points] First let's explore the datasets through the following exercises. Note that we cannot plot the data in a meaningful way given that number of features exceed the physical dimensions:
   (a) How many datapoints are in the training, validation, and testing sets?
   (b) How many features are available for each datapoint?
   (c) What are the average *alcohol* and *pH* values for *training* samples?

2. [15 points] Decision Trees:
   (a) Implement a binary decision tree model for the training data. You may use whatever libraries you prefer.
   (b) There are a number of hyperparameters that can be tuned to improve your model, one of which is the criteria for ending the splitting process. Two common ways of terminating the splitting process are *maximum depth* of the tree or *minimum number of samples* left. Tune the *maximum depth* of the tree by reporting the accuracy of the classifier in 2a on the validation set for different settings of *maximum depth*. Plot your findings.
   (c) Use the optimum setting of *maximum depth* found in 2b to report the accuracy of the classifier on the test dataset.

Problem: Logistic Regression [25 points]. For this problem, you will need to use the logistic regression. You will evaluate your algorithm on a dataset listing biometric measurements on patients suffering from Parkinson's Disease The description of the dataset can be found here (*hint*: in MATLAB, you can check the description of the features using *parkinsons.names* command). The objective is to recognize healthy people from those with Parkinson's disease using a series of biomedical voice measurements. The dataset has been separated into parkinsons_train.csv and parkinsons_test.csv files. For each sample, the rightmost ("total_UPDRS") column is the target value. All other columns are features.

1. [10 points] Train a logistic regression model using whatever toolkit you prefer on the training data. Use a fixed learning rate $\alpha = 10^{-6}$. Report the training error and the number of iterations needed before convergence.

2. [5 points] Plot the curve of the loglikelihood of your model as a function of the number of iterations (*hint:* the log likelihood curve should be monotonically increasing).

3. [10 points] Repeat 1 and 2, but this time train the logistic regression using the line search algorithm (aka newton line-search) to refine the step size α adaptively at each iteration. The line search method requires an initial value α_0. This value should be chosen fairly large, e.g., $\alpha_0 = 10^{-4}$.

4. [5 points] How do the training and testing errors of the two approaches compare? How do the loglikelihood curves compare? Does one method converge faster than the other? If so, why?