ENGS/QBS 108 Fall 2017 Assignment 2

Due October 3, 2017
Instructors: George Cybenko and Saeed Hassanpour

1 K means and K nearest neighbors clustering

Problem: Clustering [13 points]. In this problem, you will solve a clustering task using the k-Means and k-NN algorithms you learned in class. The dataset for this problem has been provided in the *clustering* folder as csv files. There are 3,100 examples in the dataset. Each entry has two features (x_1, x_2) and a target cluster y. The dataset has been split into two sets: *train_data.csv* and *test_data.csv*. Use the training set solely for all analysis, but report performance results on the test set.

1. 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, are in the dataset?

   (c) What is the dataset distribution across the k clusters? Use a histogram to illustrate this.

   (d) Is the training data balanced?

2. Using the k-Means algorithm, implement a clustering model for the training data. You can use Matlab, Python (Scikit-learn) or any other tools at your disposal. Be sure to include code or screenshots of the implementation. Report the clustering accuracy of your model on the test set.

3. Repeat the above but using the k-Nearest Neighbors algorithm. Note that k here refers to the number of neighbors, not clusters. By repeated training, find the optimal k that produces a similar number of clusters as you first estimated. Report the clustering accuracy of this model on the test set.

2 Decision Tree Classification

3 Logistic Regression

[**25 points + 10 additional implementation points**] This assignment offers you two choices, one is an applied implementation that follows the class material, and the other one is an complete implementation for those interested in deeper understanding of logistic regression. The complete implementation will be awarded with extra-credit points which will be counted towards a grade-boost over the course of the term.

For this problem, you will need to use the logistic regression. You will evaluate your algorithm on the so-called *parkinsons* dataset contained in *parkinsons.mat*, which we obtained from the UCI Machine Learning Repository. 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.

This assignment offers two options, one is an applied implementation that follows the class material, and the other one is an complete implementation for those interested in deeper understanding of logistic regression.

3.1 Applied implementation

Problem: Programming: Naive Logistic Regression [13 points]. Train the logistic regression via gradient ascent using a small fixed learning rate $\alpha = 10^{-6}$. Your solution should report both the training and the text error, the number of iterations needed to reach convergence, and you should plot a curve of log likelihood as a function of the number of iterations (*hint:* if you have implemented the functions correctly, the log likelihood curve should be monotonically increasing).

Problem: Programming: Line search optimization [8 points]. Train the logistic regression using 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., $\alpha_0 = 10^{-4}$. Your solution should report, for both the version using the fixed α (as coded for the previous question) and the one using line search, the following information: the training and the test errors, the number of iterations needed to reach convergence, and the log likelihood as a function of the number of iterations.

Based on the results, answer the questions below:

Problem: Writing [2 points]. Compare the training and test errors of the two variants of gradient ascent. Are the errors different in the two cases? Explain why or why not.

Problem: Writing [2 points]. Now compare the two log likelihood curves. Does the method using line search converge faster or slower than the version using a fixed step size? Explain the result.