Homework Assignment 4: Nearest Neighbors

Due Monday, February 10th, 2020 at 11:59pm

Description

In class we learned KNN and weighted nearest neighbor. In this problem set, you will study the effect of choosing K on the classification accuracy, and implement the weighted voting algorithm.

What to submit

Create a folder ps4_LastName_FirstName. The structure of your folder should be as follows:

ps4_LastName_FirstName /

- input/ - input data, images, videos or other data supplied with the problem set
- output/ - directory containing output images and other generated files
- ps4.m - your Matlab code for this problem set
- ps4_report.pdf - A PDF file that shows all your output for the problem set, including images labeled appropriately (by filename, e.g. ps0-1-a-1.png) so it is clear which section they are for and the small number of written responses necessary to answer some of the questions (as indicated). Also, for each main section, if it is not obvious how to run your code please provide brief but clear instructions (no need to include your entire code in the report).
- *.m - Any other supporting files, including Matlab function files, etc. It's a good practice to add the keyword 'end' at the end of each function file
- ps4_LastName_FirstName_debugging.m – one m-file that has all of your codes from all the files you wrote for this assignment. It should be a concatenation of your main script and all of your functions in one file (simply copy all the codes and pate them in this file). In fact, this file in itself can be executed and you can regenerate all of your outputs using it.

Zip it as `ps4_LastName_FirstName.zip`, and submit on canvas.

Guidelines

1. Include all the required images in the report to avoid penalty.
2. Include all the textual responses, outputs and data structure values (if asked) in the report.
3. Make sure you submit the correct (and working) version of the code.

4. Include your name and ID on the report.
5. Comment your code appropriately.
6. Please avoid late submission. Late submission is not acceptable.
7. Plagiarism is prohibited as outlined in the Pitt Guidelines on Academic Integrity.

Questions

1- **Effect of K**: In this part, you will use MATLAB functions `fitknn` and `predict` to study the effect of K on the prediction accuracy and determine the optimal value of K for the given multiclass dataset. You will also learn about cross-validation. Load the data file 'hw4_data1.mat' into MATLAB. The data contains 5 equally sized folds obtained from a large training set. The folding was done, to enable us to test the algorithm using a proportion from the original training set. For each fold, you will find a training matrix (e.g., X1) and the corresponding labels vector (e.g., y1). Your results reported below should be an average of the results when you **train** on four folds and **test** on the remaining one. Thus, you will need to train **5 different KNN classifiers** and test them. For example, the first classifier is trained using the first 4 folds and tested using the fifth, where the second classifier is trained using the first three folds in addition to the fifth and then tested using the fourth fold; and so on for the remaining classifiers.

a. Compute the average accuracy (over the five folds) for the following values of K = 1:2:15.
Hint: To compute accuracy for one fold, check the ratio of test samples whose predicted labels are the same as the ground-truth labels, out of all test samples.
Output: A figure showing average accuracy vs K as ps4-1-a.png
Text output: what value of K do you suggest for this particular problem? Is this value robust to any other problem? Why?

2- **Weighted NN**: In this part you will implement and apply the Gaussian weighted neighbors classifier, using the equation in lecture slides, and apply it to some testing examples. For this example, you will need to use the training and testing samples in 'hw4_data2.mat': X_train and y_train are the training example and the corresponding class labels. X_test are the testing features that you are required to predict a class for, and y_test are the ground truth labels that you can use to compute the accuracy.

a. Write a function, `y_predict = weightedKNN(X_train,y_train,X_test,sigma)` that uses all neighbors to make a prediction on the test set, but weighs them according to their distance to the test sample. Use the Euclidian distance as your distance metric. Hint: you may find the MATLAB function `pdist2` useful.
   - `X_train` is an $m \times n$ features matix, where m is the number of training instances and n is the feature dimension,
   - `y_train` is an $m \times 1$ labels vector for the training instances,
   - `X_test` is an $d \times n$ feature matrix, where d is the number of test instances,
   - `sigma` is a scaler denoting the bandwidth of the Gaussian weighing function,
   - `y_predict` should be a $d \times 1$ vector that contains the predicted labels for the test instances.

   Function file: `weightedKNN.m` containing function `weightedKNN`

b. Test your function using the provided training matrix (X_train), training labels (y_train), and testing features matrix (X_test). Use the following values for sigma: 0.1, 0.5, 1, 3, and 5. Compute the classification accuracy for each value of sigma.
Output: a table that list the accuracy vs sigma
Text output: comment on your results and the effect of sigma.