CS1675: Homework 4

Due: 10/4/2018, 11:59pm

This assignment is worth 40 points.

In this assignment, you will implement and experiment with KNN classification.

Part I: K-nearest-neighbors (10 pts)

Implement KNN. For each test instance, compute its distance to all training instances, pick the closest K training instances, pick the most common among their labels, and return it as the label for that test instance. Use the Matlab function `pdist2`.

Inputs:

- an NxD feature matrix `X_train` where N is the number of training instances and D is the feature dimension,
- an Nx1 label vector `y_train` for the training instances,
- an MxD feature matrix `X_test` where M is the number of test instances, and
- a scalar `K`.

Outputs:

- an Mx1 predicted label vector `y_test` for the test instances.

Part II: Weighted KNN (10 pts)

Implement a Gaussian-weighed KNN classifier using the equation given in class, in a function `weighted_knn.m`. This version uses all neighbors to make a prediction on the test set, but weighs them according to their distance to the test sample.

Inputs: The first three are as for `my_knn.m`. The fourth one is:

- a scalar `sigma` denoting the bandwidth of the Gaussian.

Outputs: same as above

Part II: Testing KNN on the Pima Indians dataset (20 pts)

You will use the Pima Indians Diabetes dataset (originally from the UCI repository, but link is now disabled) to test your KNN implementation and experiment with different values of K. Write your code in a script `knn_classification.m`.

1. [5 pts] Download the data file. The last value in each row contains the target label for that row, and the remaining values are the features. Split the data into 10 approximately equally-sized "folds". Your results reported below should be an average of the results when you train on the first 9 folds and test on

the remaining 1, then if you train on the folds numbered 1 through 8 and the 10th fold and testing on the 9th fold, etc. For simplicity, you can use folds of size 76 and drop the remaining 8 instances.

2. [2 pts] Make sure to normalize the data X by subtracting the mean and dividing by the standard deviation over each dimension, computed using the training data only, as in HW3.
3. [5 pts] Apply your KNN function and compute the accuracy over all folds, then average the results. To compute accuracy, check the ratio of test samples whose predicted labels are the same as the ground-truth labels, out of all test samples.
4. [5 pts] Experiment with the following values: `K = 1:2:15`. Plot the results (with values of K on the x-axis and accuracy on the y-axis) and include the plot in a file `report.pdf/docx`.
5. [3 pts] So far, we have been weighing neighbors equally. Now we want to experiment with weighing them according to their distance to the test sample of interest. Experiment with 3 different values of the bandwidth parameter σ, create a plot with σ on the x-axis and accuracy on the y-axis, and include the plot in the report file.

Submission: Please include the following files:

- `my_knn.m`
- `weighted_knn.m`
- `knn_classification.m`
- `report.pdf/docx`