Homework III

Due date: Feb. 28, 2019 (Before the Class)

Problem 1: kNN Classifier (20 points)

You are asked to build a k-Nearest Neighbor (kNN) classifier. The data set for evaluation is the heart data set. More information about the data can be found here `https://archive.ics.uci.edu/ml/datasets/Statlog+(Heart)`. The data set is included in homework-3-data-set.zip on ICON. In the heart data folder, there are three files: "trainSet.txt", "trainLabels.txt", and "test.txt". Each row of "trainSet.txt" corresponds to a data point whose class label is provided in the same row of "trainLabels.txt". Each row of "testSet.txt" corresponds to a data point whose class label needs to be predicted. You will train a classification model using "trainSet.txt" and "trainLabels.txt", and use it to predict the class labels for the data points in "testSet.txt".

Use the leave one out cross validation on the training data to select the best k among $\{1, 2, \ldots, 10\}$. Report the averaged leave-one-out error (averaged over all training data points) for each $k \in \{1, 2, \ldots, 10\}$ and the best k used for predicting the class labels for test instances. You should also report the predicted labels for the testSet.

Problem 2: Naïve Bayes for Text Classification (50 points)

In this problem, you are asked to implement a Naïve Bayes Classifier for text categorization. In particular, given a document $\mathbf{x} = (x_1, \ldots, x_m)$, where x_j is the occurrence of word w_j in the document, we model $\Pr(\mathbf{x}|C_k)$ as

$$\Pr(\mathbf{x}|C_k) \propto \prod_{j=1}^{m} [p(w_j|C_k)]^{x_j}$$

where $p(w_j|C_k)$ stands for the probability of observing the word w_j in any document from category C_k. Given a collection of training documents $\mathbf{x}^1, \ldots, \mathbf{x}^{n_k}$ in category C_k, where $\mathbf{x}^i = (x_1^i, \ldots, x_m^i)$, the probabilities of $p(w_j|C_k)$ can be estimated by MLE, i.e.,

$$p(w_j|C_k) = \frac{\sum_{i=1}^{n_k} x_j^i}{\sum_{j'=1}^{m} \sum_{i=1}^{n_k} x_{j'}^i}, \forall j, k$$

To avoid the issue that some words may not appear in training documents of a certain class, the estimated probabilities are usually smoothed. One smoothing method is Laplace smoothing which computes $p(w_j|C_k)$ by

$$p(w_j|C_k) = \frac{\sum_{i=1}^{n_k} x_j^i + 1}{\sum_{j'=1}^{m} \sum_{i=1}^{n_k} x_{j'}^i + m}, \forall j, k$$

The log-likelihood for $(\mathbf{x}, y = k)$ is given by

$$\log \Pr(\mathbf{x}|C_k) \Pr(C_k) = \underbrace{\sum_{j=1}^{m} x_j \log p(w_j|C_k) + \log \Pr(C_k)}_{f_k(\mathbf{x})} + const$$

where $\Pr(C_k)$ can be estimated by $\Pr(C_k) = \frac{n_k}{\sum_{k=1}^{K} n_k}$ and $f_k(\mathbf{x})$ can be considered as a prediction score of $\mathbf{x}$ for the k-th class. The class label of a test document $\mathbf{x}$ can be predicted by $k^* = \arg\max_{1 \le k \le K} f_k(\mathbf{x})$.

The data set for training and evaluation is the 20NewsGroup data, which is included in the provided zip file. You will find six **text** files in this data set: train.data, train.label, train.map, test.data, test.label, and test.map, where the first three files are used for training and the last three files are for testing. In the train.data file, you will find the word histograms of all documents; each row is a tuple of format (document-id, word-id, word-occurrence). The class labels of training documents can be found in train.label with the order corresponding to training documents' id, and the topic of each class can be found in train.map. Similarly, the word histograms and the class assignments of test documents can be found in test.data and test.label, respectively. In this problem, you need to train a Naïve Bayes classifier with the Laplace smoothing using the training data and apply the learned classifier to predict the class labels for the test documents. You need to (a) submit your code for implementing the Naive Bayes classifier and (b) report the classification accuracy (the proportion of test documents that are classified correctly) over the test documents. **Note: You code should be able to generate a file that contains the predicted labels of test documents in the same order. Include instructions on how to run your code so that TA can run your code.**

Problem 3: PCA (30 points)

You are asked to build a k-Nearest Neighbor (kNN) classifier based on dimenationality reduced data by PCA. The data set for evaluation is the gisette data set. More information about the data can be found here `https://archive.ics.uci.edu/ml/datasets/Gisette`. The data set is included in homework-3-data-set.zip on ICON. The data is in the same format as that in Problem 1. You will train a classification model using "trainSet.txt" and "trainLabels.txt", and use it to predict the class labels for the data points in "testSet.txt". First, train a kNN based on the original features and then conduct PCA on the data and learn a kNN model using the reduced data. Report the performance of both models on the testing data. For learning both models, you should also conduct cross-validation (of your choice) to select the best k. Describe the cross-validation approach and report the best value of k for both models. You need to submit your code as well.

Problem 4: Median Distance of Nearest Neighbor (Optional: 20 points)

Consider N data points uniformly distributed in a p-dimensional unit ball centered at origin. Consider the nearest neighbor of the origin. Prove that the median distance from the origin to the

closest data point is:

$$d(p, N) = \left(1 - 2^{-1/N}\right)^{1/p}$$