Part 2

Related work:

Problem representation: The second part of the project consisting in implement text classification using non-linear method. According to the literature, K-Nearest Neighbor is considered pretty suitable for dealing with text classification problems [http://user.ceng.metu.edu.tr/~e120321/paper.pdf]. Thus, the method that we choose to implement a non-linear classifierfor the second part of the project is KNN. In addition, a large number of initial files which are called “training documents” required by KNN text classification are available to us, which make our work perfectly feasible.

Problem representation:

For the KNN text classification, the data pre-processing that is done in order to construct the feature vectors for all our training samples is the following:

First, all the stop-words are removed from the documents of our training samples. Furthermore, all the words are reduced to their root expressions using the Porter Stemmers from NLTK package. As for the feature design, according to the literature, the feature representation of most of text classification methods are done via a bag of words, In this manner, each document can be represented by the presence/absence of a word in the ‘bag of word’ and by the frequency of its occurrences in the document. Therefore, for our KNN classifier, we first obtain our bag of words by manipulating and analyzing our training documents. Afterwards, for each document, we construct a feature array where each element is an integer variable which indicates whether or not a particular word in the bag of words is present in the document or not. If it is present, the value is the frequency of that word in the document. In case of absence, the value is simply 0. The important choices that we need to make for KNN classifier are the way we construct the feature matrix and the number of k neighbors that we take into consideration every time we make a prediction. The distance metrics, we simply choose the Euclidian distance, which is a common approach for most of text classification problems.

We use k-fold cross validation to split the data in train\_in.csv into our training and testing data set. The cross validation is done in order to make intelligent decision about the feature choices and the suitable number of k neighbors for our KNN classifier. The cross validation also permits a better estimation of prediction error. The number of k for k-fold cross validation is 137. It is chosen in attempt to make data manipulation easier: we have 88639 examples and this number can only be divided by 137 in order to include all the samples in the training and testing sets for each round of cross validation.

Testing and validation:

Using a k value of 10, we first perform cross validation to find out our optimal feature encoding method. We consider two options. The first one is that we represent the feature data as binary variables that can only indicate a specific word is present in the document or not. The second is that the features become all integer variables that encode not only the presence, but also the number of occurrences of the word in the occurrence. The table XXX shows the results of the cross-validation. As a result, we decide to implement the feature vector as indications of word occurrence frequency for our KNN classifier. The recall and precision values are average values obtained from 4 classes.

According to previous work of XXX, [https://www.researchgate.net/profile/Ahmad\_Hassanat/publication/265168466\_Solving\_the\_Problem\_of\_the\_K\_Parameter\_in\_the\_KNN\_Classifier\_Using\_an\_Ensemble\_Learning\_Approach/links/5403b2cf0cf23d9765a5d02f.pdf], it is stated that the rule of thumb which dictates that the square root of the number of samples is often the optimal choose does not hold, and large values of k does not improve the accuracy of classification. Starting from XXX to XXX, we use cross validation to producing the Table XXX and choose the optimal K, which is XXX. The recall and precision values are average values obtained from 4 classes.

Discussion:

The KNN method that we use to implement a non-linear classifier for the problem present some advantages. First, after careful pre-processing of the data using online packages, many noisy data are efficiently removed, which allows us to avoid a drawback of KNN, which is that it is vulnerable to noise. Second, KNN benefits from having a large amount of training data and allow to make non-linear predictions. Third, by determining the most suitable number of k for KNN and the feature encoding method, its prediction error can be improved. However, KNN requires a lot of query time and memory. Also, the choice of features, of distance metric and of the k hyper-parameter is subtle, so the results of prediction can be negatively affected if some variation or noisy data are introduced into the training set.