Project report on CSCI6905 - Machine Learning for Big Data
by **Md Rashadul Hasan Rakib (B00598853)**

# Project Title: Document classification using reduced features based on document frequency

**Introduction:** The goal of this project is to reduce the dimensionality (feature) of the documents so as to obtain better classification result. I have used document frequency (df) for dimensionality reduction and apply different classification algorithms to classify the documents.

**Why Dimensionality reduction (feature selection):** Dimensionality reduction plays an imperative role in classification task because not every feature in a document bears a distinctive characteristic. If all the features are used, then the classification model will be overfitted, at the same time if very few numbers of features are used, then the model will be cursed for underfitting problem. So, the challenge is to find the optimal number of features that will improve the classification result.
There is a significant role of df to build a classification model that will not be affected by overfitting and underfitting problem. If the df is 1, then all the features will be selected to train the classifier and overfitting problem will arise. Similarly, for high df, very few number of features will be selected and will cause underfitting problem. Therefore, the optimal df should be found to select the features in order to avoid these problems. Feature selection using df is really very faster and it will not take more than X iterations where X is the number of documents. **Moreover, I have optimized the number of iterations using sqrt(X).** I have used training data to select features using df; then train the classifier and evaluate it using testing data.

**Why choosing better classification algorithm:** The data sets used in this project are Estrogens, OralHypoglycemics, Triptan, and BB. Among them, Triptan is a balanced one and rests of them are imbalanced. This is because, it is very important to find a classification algorithm that will efficiently handle these imbalanced data sets.
I have implemented Multinomial Naïve Bayes (MNB) to classify documents since it is faster and easy to understand; but it does not perform well for imbalanced data since MNB gets biased to the class (E) that has more instances and misclassifies the documents of minor class (I). Existing implementation of Support vector machine (SVM) has been examined by finding best parameters (slack variable and regularization). SVM requires finding the combination of best karnels and parameters. Moreover it is sensitive to the number of instances, that is why it is quite slower and its running time is a second degree polynomial. Since the documents are high dimensional and have large numbers of irrelevant features; Random Forest (RF) does not give better performance. For the data with different number of levels, random forests are biased in favor of those attributes with more levels. KNN algorithm is a lazy learner, i.e. it does not learn anything from the training data and simply uses the training data itself for classification. KNN must compute the distance from new instance to all instances and find the K closest neighbors at each prediction, which can be slow if there are a large number of training examples.
Considering the problems of above classification algorithms in terms of accuracy and running time, I prefer Compliment Naïve Bayes (CNB) to handle the imbalanced data since CNB predicts the class of an instance more efficiently because it uses more even amount of training data per class, and reduces the bias. Using CNB I get better result than other classifiers. CNB is faster, easy to implement and its performance is not affected due to irrelevant features because

*The whole project has been implemented using R

non-relevant features result in small probability values and are eliminated by the argmax operator at the core of naïve Bayes approach.

**Methodology:** I perform the following steps for document classification. The classification result has been measured using Area under ROC curve (AUC) value. Higher AUC refers to higher performance. AUC is more meaningful than Recall and Precision. High AUC means that the recall values for both major (E) and minor class (I) are good. If any recall value deviates from other at a large scale, high AUC is not possible. It gives the overall performance on the both classes E and I. Therefore, I prefer AUC to measure the performance of the classifiers.

i. Text preprocessing
ii. Feature Selection
iii. Evaluate Document Classification

**i. Text preprocessing:** The data are extracted from the text file and transposed into CSV format. During this step, the stop words, punctuation characters are removed. For all attributes T, A, P and M, space characters are used to split words. Term frequency (tf) weighting is used for the words in a document. Documents that have no class label (E or I) are simply ignored. **A document consists of the values of four attributes T, A, P and M without having prior knowledge about the significance of each attribute. I train and test the classifier in more generic way. Each attribute among four is equally important for any classifier.**

**ii. Feature selection:** The training data is given to the feature selection module. Same document classification algorithm is used for feature selection. For instance, I select features from the training part using SVM as well as evaluate the classifier on testing data using SVM. Following algorithm FSDF shows how document frequency is used to select features. **This algorithm will iterate for SQRT(#docs(data)) times**. I have examined on different datasets for various classification algorithms, and I find the best feature set that have document frequency less than or equal to SQRT(#docs(data)).

```
Algorithm FSDF(data, CName){
    //FSDF=Feature Selection by Document Frequency
    //data=training data
    // CName=Classifier Name
    Df=2  //Df=Document frequency ; I start from Df=2 i.e. the features that appear in at least
        //two documents. For Df=1, all features will be selected which will cause
        //overfitting problem
    MaxResult=0
    BestFtrSet=null //Best feature set, selected by document frequency
    While(Df<=SQRT(#docs(data))){
        FtrDf= Selected features that have document frequency Df
        FilteredData= data[FtrDf] //New data frame with only the selected features
        Tr=randomly selected 70% documents from FilteredData
        Te=Rest of the documents from FilteredData
        Result=Train the CName classifier by Tr; get classification result from
            CName classifier using the testing data Te.
```

*The whole project has been implemented using R

```
                //Area under ROC curve (AUC) has been used as result.
            If(Result> MaxResult){
                MaxResult= Result
                BestFtrSet= FtrDf
            }
        }
        Return (BestFtrSet)
}
```

**iii. Evaluate Document Classification:** I have implemented 10 fold cross validation to get the average AUC for a particular classifier on particular data set. Using 10 fold, each part of the data has been tested. In each iteration, whole data is divided into two parts training and testing set. Using the training set I get the best feature set that is used to classify the testing data and evaluate the performance of the classifier; how accurately it can classify the documents of Major class (E) and Minor class (I).

```
Algorithm CV10F(data, CName){
        //CV10F=10 fold cross validation
        //data=whole data set
        // CName=Classifier Name
        itemsInAFold[10]=getItemsInFolds(#docs(data))
        // itemsInAFold contains the number of items in each fold, determined by the function
        // getItemsInFolds
        sumAUC=0
        For(NumberOfItems in itemsInAFold){
                TestDataSetInAFold=getRandomDocsForAFold(data, NumberOfItems)
                //Random Test Data set obtained in a fold will not appear in any other testing
                //folds
                TrainDataSetInAFold=data[-TestDataSetInAFold]
                //The rest of the docs other than TestDataSetInAFold
                bestFtrSet= FSDF(TrainDataSetInAFold, CName)
                //best feature obtained by the training data set
                FilteredTrData= TrainDataSetInAFold [bestFtrSet]
                //New training data frame with only selected features
                FilteredTeData= TestDataSetInAFold [bestFtrSet]
                //New testing data frame with only selected features
                PredictedClasses= predictClass(FilteredTrData, FilteredTeData, CName)
                //predict the class E or I for the testing data set FilteredTeData
                Result=GetAUC(PredictedClasses, FilteredTeData$class)
                //Calculate AUC from the predicted and true classes
                sumAUC= sumAUC+ Result
        }
        Print(sumAUC/10)
}
```

*The whole project has been implemented using R

**Detail Experimental Result:** Short notations used for different classification algorithms and evaluation measures are given below.
CNB: Compliment Naïve Bayes
MNB: Multinomial Naïve Bayes
RF: Random Forest
SVM: Support vector machine
KNN: K-nearest-neighbor

RecE=Recall for E class
PreE=Precision for E class
RecI=Recall for I class
PreI=Precision for I class
AUC= Area under the ROC curve

| Data Set | CNB: Select feature using Train Data within 10 fold CV | | | | |
|---|---|---|---|---|---|
| | RecE | PreE | RecI | PreI | AUC |
| BB | 0.78843 | 0.91419 | 0.70017 | 0.44066 | 0.74430 |
| Estrogens | 0.80014 | 0.94927 | 0.84071 | 0.55903 | 0.820428 |
| OralHypo | 0.82100 | 0.84420 | 0.68028 | 0.62105 | 0.75064 |
| Triptan | 0.75460 | 0.83619 | 0.89821 | 0.84267 | 0.82640 |

| Data Set | CNB: No Feature selection, apply 10 fold CV | | | | |
|---|---|---|---|---|---|
| | RecE | PreE | RecI | PreI | AUC |
| BB | 0.82166 | 0.88979 | 0.57120 | 0.43233 | 0.69643 |
| Estrogens | 0.80128 | 0.90242 | 0.69961 | 0.51215 | 0.75045 |
| OralHypo | 0.79758 | 0.82978 | 0.630482 | 0.5775 | 0.71403 |
| Triptan | 0.69676 | 0.87857 | 0.93166 | 0.81442 | 0.81421 |

| Data Set | MNB: Select feature using Train Data within 10 fold CV | | | | |
|---|---|---|---|---|---|
| | RecE | PreE | RecI | PreI | AUC |
| BB | 0.69664 | 0.89584 | 0.67152 | 0.35470 | 0.68408 |
| Estrogens | 0.30748 | 0.92426 | 0.91642 | 0.27082 | 0.61195 |
| OralHypo | 0.74351 | 0.83291 | 0.65394 | 0.52476 | 0.69872 |
| Triptan | 0.77956 | 0.77587 | 0.81192 | 0.81035 | 0.79574 |

| Data Set | SVM: Select feature using Train Data within 10 fold CV | | | | |
|---|---|---|---|---|---|
| | RecE | PreE | RecI | PreI | AUC |
| BB | 0.96157 | 0.84383 | 0.24642 | 0.61544 | 0.60399 |
| Estrogens | 0.95714 | 0.82124 | 0.28798 | 0.76 | 0.62256 |
| OralHypo | 0.89200 | 0.79385 | 0.47299 | 0.66289 | 0.68249 |
| Triptan | 0.67965 | 0.80555 | 0.87799 | 0.78817 | 0.77882 |

For SVM, I find the best parameters gamma and cost by tune.svm function to get better classification result

*The whole project has been implemented using R

| Data Set | RF: Select feature using Train Data within 10 fold CV | | | | |
|---|---|---|---|---|---|
| | RecE | PreE | RecI | PreI | AUC |
| BB | 0.99143 | 0.82419 | 0.10914 | 0.75666 | 0.550287 |
| Estrogens | 0.98213 | 0.81331 | 0.20290 | 0.80833 | 0.592518 |
| OralHypo | 0.95348 | 0.78045 | 0.37303 | 0.79888 | 0.663260 |
| Triptan | 0.78239 | 0.80964 | 0.84569 | 0.81936 | 0.814043 |

| Data Set | KNN: Select feature using Train Data within 10 fold CV | | | |
|---|---|---|---|---|
| | RecE | PreE | RecI | PreI |
| BB | 0.82387 | 0.98606 | 0.65957 | 0.11355 |
| Estrogens | 0.83171 | 0.94833 | 0.65853 | 0.34177 |
| OralHypo | 0.73658 | 0.98692 | 0.87096 | 0.2 |
| Triptan | 0.66995 | 0.90666 | 0.90789 | 0.67317 |

For KNN, I use the library (RWeka) by which I find the best K (number of nearest neighbors) to get better classification result

**Discussion about the result:** From the above experimental result, I find that the performance of the classifiers on Triptan data set is better than other data sets because Triptan is more balanced and rest of three are imbalanced. Since Triptan is balanced, so we get better values for RecE, PreE, RecI, PreI and AUC for these data sets.
For imbalanced data set (BB, Estrogens, OralHypo) not every classifier can perform better. I find that among the above classifiers, CNB performs best since it is independent of the number of features. However, applying CNB with feature selection, gives better result for imbalanced data set (BB, Estrogens, OralHypo). So, I conclude that applying classifier with feature selection, really improves the performance of classification result.

*The whole project has been implemented using R