Comparative Study on Feature Space Reduction for Spam Detection

Azam. N
Department of Computer
Engineering.
National University of Sciences
and Technology, EME College
Rawalpindi Pakistan.

Dar. H. A
Department of Computer
Engineering.
National University of Sciences
and Technology, EME College
Rawalpindi Pakistan.

Marwat. S
Department of Computer
Science, Agriculture University
Peshawar Pakistan.

Abstract

The growing volume of spam email has resulted in the need for accurate spam solutions. The accuracy of any solution will depend on classification algorithm coupled with the feature selection method. Selection of most discriminating features is a critical decision in any text categorization task. Efficient features selection method will help us improve our classification accuracy with reduced time and memory. In this paper we present a comparative study between three feature selection methods, namely: Mutual Information, latent semantic indexing (PCA) and thresh holding word frequency. The effects on classification accuracy for varying features set sizes obtained from different methods were compared and analyzed. The results in some cases were quite promising even for as smaller as 20 features. The classification algorithm used was k-Nearest Neighbor

1 Introduction

Electronic mail is one of the most reliable and inexpensive source of communication world wide. The wide use and easy access of the medium makes it prone to spam emails. Such mails not only waste a lot of bandwidth but also can cause serious damages to personal computers in the form of computer viruses. Statistics shows that spam increases day by day to huge volumes and contributes to large percentages of the total mails[1]. Spam contributes to about 40% of all incoming emails[2](6 spam emails per user everyday). A partial and largely ineffective solution is the change of email addresses by the users. Therefore, there is an exigent need for effective solution that deals with the problem of junk mails.

A spam solution consists of a classification algorithm coupled with feature reduction technique. A lot of research had been conducted for finding an algorithm that will do a good job of classification. The Naïve Bayesian approach had been discussed in [1]. Memory based approach and its comparison with the naïve Bayesian has been discussed in [2]. Classification based on support vector machine has been discussed in [3], AdaBoost Boosting algorithm in [4], common vector approach in [5] [6]. All of the methods achieve high accuracy rates with the classification algorithms specified but that's not enough. The thing which is still missing is good algorithm for feature set reduction which will do the same job in lesser time and with lesser memory.

Different feature reduction techniques have been investigated so far for text categorization tasks. Mutual information has been discussed in [1] [2] and Latent semantic indexing or PCA has been discussed in [5] which achieve good accuracy rates when used by the classifiers. Other reduction techniques like CHI-test, information gain, document frequency thresh holding are described in [7].

We investigated the performance of three dimensionality reduction techniques using cost sensitive measures when used for classification. Two out of three i.e. latent semantic indexing and thresh holding word frequency, do not considers class while mutual information does. The performances of the techniques were then judged with suitable metrics for accuracy. The classification algorithm used for measuring the performances was k-Nearest Neighbor.

[1] Consult http://www.junk-o-meter.com/stats/index.php and http://spamlinks.net/stats.htm

[2] http://spam-filter-review.toptenreviews.com/spam-statistics.html

2 Preprocessing of data

The corpus for the experimentation was of large size. So we perform some of the preprocessing which is normally done in text classification tasks on the textual data with the aim of removing less informative, redundant data and to make data reduced in size before being used for the experimentations.

First we remove all the words that have length lesser than or equal to two (this helped a lot in removing the commas, full stops and lots of other useless terms). Than we remove all of the alpha numeric words. Some people will object to it with a claim that alpha numeric words help a lot for the class identification but our results without them are quite acceptable. Then a defined set of stop terms were removed from the data. .

Figure 1: list of stop words used

```
then, there, that, which, the,
those, now, when, which, was, were,
been, had, have, has, will, subject,
here, they, them, may, can, for,
such, and, are, but, not, with,
your.
```

The data obtained after removing the stop terms were then stemmed according to porter's Stemming algorithm. The preprocessing reduced the corpus to about half in size. Next we need proper representation of our data.

We took the bag of words technique (without normalizing the data) with two dimensional representation of the data where the columns corresponds to the examples and the rows corresponds to the features or attributes. Each message in the dataset were represented as a columns vector where rows corresponds to features. Features will correspond to words in the emails. In this way each message is represented as a vector of frequencies of words i.e. the entry in the ith column and jth row will denote that how many time in example i the feature j is repeated.

3 Reduction Techniques

Feature set reduction techniques falls mainly in two broad classes [9]. In the first class we reduce the number of features by selecting a subset of the original features based on certain criteria. While in the second class we transform our features to get new transformed features.

We took two techniques which comes under the first class i.e. mutual information and thresh holding word frequencies and one technique from the second class that is latent semantic indexing. Next we are going to describe these techniques in detail.

3.1 Mutual Information

Before applying the mutual information feature reduction technique we converted our original real data to Boolean data such that if the entry in the ith column and jth row is 1 than it means that feature j is present in the example i and if 0 then otherwise. The MI scores are then calculated for every feature according to the following formula as described in [1] [2].

$$MI(X,C) = \sum_{X\text{€}\{1\ ,\ 0),\ C\ \text{€}\ \{\text{legitimate, Spam}\}} P(X\ ,\ C).\ \text{Log}\ \{P\ (X,\ C)/P(X\ ).P(C)\}.$$

The features were then arranged based on the highest MI scores and highest MI features were selected for the classification purposes**.**

3.2 Latent Semantic Indexing

Also widely known as Principal Component Analysis and Karhunen-Loève transform. It is one of the most widely used techniques for reducing multidimensional datasets to lower dimensions for effiecient computation and analysis.

We used the covaraince method of the PCA algorithm. The main steps of the algorithms were as follows. (Remember our data after preprocessing were arranged in X = [M*N] matrix, where M equals the number of features and N equal the number of examples and X is our data set with $X_1, X_2 \ldots . X_n$ column vectors of examples)

1. Find the mean along each feature dimension for m = 1,2,….M.. Place the mean in the mean column vector $\boldsymbol{\mu} = \mathbf{1/N} \sum \mathbf{X[M*N]}$

   μ is M*1 vector now.

2. Next we find the mean adjusted data by subtracting the mean vector $\boldsymbol{\mu}$ from each of the column of data matrix X so that $\boldsymbol{\mu}_{adjusted\_data} = \mathbf{X - \boldsymbol{\mu} * h}$ (where h is 1*N row vector of all ones).

3. Find the covariance matrix **C** $\mathbf{C} = (\ \boldsymbol{\mu}_{adjusted\_data} * \boldsymbol{\mu}_{adjusted\_data}^{T}\ )\ \mathbf{/N.}$

4. Compute the Eigen values matrix $\boldsymbol{\lambda}$ and Eigen vectors matrix **V** of the covariance matrix **C** So that $\mathbf{C * V = V * \boldsymbol{\lambda}.}$

5. Obtain the Eigen values from the matrix λ. Arrange them in increasing order and select the top few. Also select the Eigen vectors corresponding to the top most Eigen values selected. Make sure to get the correct pairings of the Eigen values and Eigen vectors.

6. Obtain the transformed data by the following operation

$$\mathbf{Eigen\_vectors}_{\text{selected}}{}^{T} * \boldsymbol{\mu}_{\text{adjusted\_data.}}$$

3.3 Thresh Holding Word Frequencies

This technique is fairly simple. Those features whose frequencies in the entire data set is lesser than some predefined value will be discarded with the argument that these features will not be as good for the classification as those having greater frequencies. Though greater frequencies does not necessarily mean that these features would be present in large number of documents but still our results shows that they are good enough to be considered.

4 Evaluation Measures

We used the evaluation measures that were established in [1] [2]. Let $\mathbf{N}_{\text{Spam}}$ and $\mathbf{N}_{\text{Leg}}$ be the total number of spam and legitimate emails in our data set. Furthermore let $\mathbf{N}_{\text{Y-Z}}$ be the number of emails that are classified as Z but belong to class Y {Y, Z € (spam, legitimate)}. We can define the accuracy and error as

$$\text{Accuracy} = \mathbf{N}_{\text{Spam–Spam}} + \mathbf{N}_{\text{Leg-Leg}} / \mathbf{N}_{\text{Spam}} + \mathbf{N}_{\text{Leg}}$$

$$\text{Error} = \mathbf{N}_{\text{Leg-Spam}} + \mathbf{N}_{\text{Spam-Leg}} / \mathbf{N}_{\text{Spam}} + \mathbf{N}_{\text{Leg}}$$

In the above formulas both types of errors have been assigned equal weights(i.e. weight of $\mathbf{N}_{\text{Leg–Spam}}$ is equal to weight of $\mathbf{N}_{\text{Spam-Leg}).}$ How ever identifying legitimate email as spam is more costly and most often unacceptable then identifying spam as legitimate. So we will introduce a constant and say that $\mathbf{N}_{\text{Leg–Spam}}$ is $\boldsymbol{\lambda}$ times more costly than $\mathbf{N}_{\text{Leg–Spam.}}$ To reflect this cost difference we treat every legitimate message as if it were $\boldsymbol{\lambda}$ messages. Furthermore when ever we identify a legitimate message correctly it will mean $\boldsymbol{\lambda}$ success and if identified as spam it will be count as $\boldsymbol{\lambda}$ errors. So after this cost sensitive measures the accuracy and error takes the form

$$\mathbf{WAC} = \mathbf{N}_{\text{Spam–Spam}} + \boldsymbol{\lambda}{*}\mathbf{N}_{\text{Leg-Leg}} / \mathbf{N}_{\text{Spam}} + \boldsymbol{\lambda}{*}\mathbf{N}_{\text{Leg}}$$

$$\mathbf{WE} = \boldsymbol{\lambda}{*}\mathbf{N}_{\text{Leg–Spam}} + \mathrm{N}_{\text{Spam-Leg}} / \mathbf{N}_{\text{Spam}} + \boldsymbol{\lambda}{*}\mathbf{N}_{\text{Leg}}$$

in addition to weighted accuracy we also measure our results in terms of spam recall and spam precision to have better understanding of the results. They are given as

$$\mathbf{SP} = \mathbf{N}_{\text{Spam–Spam}} / \mathbf{N}_{\text{Spam–Spam}} + \mathbf{N}_{\text{Legitimate–Spam}}$$

$$\mathbf{SR} = \mathbf{N}_{\text{Spam–Spam}} / \mathbf{N}_{\text{Spam}}$$

5 Experimental Settings

All experiments were conducted using the ling Spam corpus[3]. The corpus is used in [2] [8] and contains 2412 legitimate and 481 spam emails. All the feature reduction techniques were used to select best features of 20, 50, 100, and 250 and in some cases 500 for the comparison purposes. Before selecting top features the data was thresh holded with word frequencies greater than 29 (after preprocessing the feature set which was over 40,000). The reasons for thresh holding was fairly simple as computation would be extremely hard if we represent every example with over 40,000 features. Thresh holding reduces the features to about 3000 features which saved a lot of memory and computation time. Thresh hold value of 29 seems debatable but look a good choice to us. If we assume that words are distributed across the emails evenly then word frequency of 29 will roughly mean that about 1% of the emails had this word so its better not to have it. Off course some of them might be very useful but most of them will be not.

The first sets of features were selected with thresh holding of word frequencies. We took the thresh holds of 332, 583, 1079, 1730 and 2315 (as these corresponds to 500, 250, 100, 50 and 20 features respectively). The second features sets were obtained from the MI scores that were computed on the entire data set. The third features sets were obtained after dividing the whole data set into ten files and then finding the MI scores of the features of individual files and select the top most (we used 10 files because experiments were conducted with 10 folds cross validation). Finally the last sets of features were also computed using the LSI(PCA) algorithm applied to every file out of 10 and the top most eigen vectors corresponding to top most eigen values of 20,50,100,250 were selected.

6 Experimental Results

We used 10 folds cross validation in our experiments. The ling spam corpus was divided into ten parts and then the experiments were repeated ten times, each time reserving different part for the testing and the remaining nine for the training purposes. All the results were then averaged over the entire set of experiments.

All the feature sets gathered were then tested with K-Nearest Neighbor using k = {1,3,5}.The following tables summarizes the results.

[3] Corpus is available at http://www.iit.demokritos.gr/skel/i-config/downloads/lingspam_public.tar.gz

Table 1: Experimental results of the feature reduction techniques when applied with the nearest neighbor k = 1

No Feature	LSI (PCA)					Thresh holding					MI(Entire data set)					MI(individual files)				
	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP
20	94.0	94.1	94.1	90.9	80.5	94.5	94.8	94.9	76.4	77.4	97.5	98.0	98.0	73.9	89.1	96.4	96.9	96.9	74.4	85.6
50	92.7	92.8	92.8	90.9	78.4	94.7	94.9	94.9	82.6	78.4	97.5	98.0	98.0	74.7	89.5	95.4	95.9	95.9	73.9	83.9
100	90.7	90.7	90.7	91.0	75.0	93.2	93.3	93.4	84.3	76.9	96.4	96.8	96.8	75.3	87.2	93.9	94.3	94.4	73.8	82.0
250	88.5	88.5	88.5	85.9	70.9	90.4	90.5	90.6	85.2	74.0	93.7	94.1	94.1	74.8	83.5	92.8	93.1	93.1	76.9	82.4
500	-	-	-	-	-	91.0	91.1	91.1	83.4	73.9	-	-	-	-	-	-	-	-	-	-

Table 2: Experimental results of the feature reduction techniques when applied with the nearest neighbor k = 3

No Feature	LSI (PCA)					Thresh holding					MI(Entire data)					MI(individual File)				
	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP	WAC λ=9 (%)	WAC λ=99 (%)	WAC λ=999 (%)	% SR	% SP
20	92.9	92.9	92.9	91.1	79.6	94.3	94.6	94.6	79.6	78.9	97.5	98.0	98.0	73.9	89.1	96.0	96.5	96.5	74.6	84.8
50	91.3	91.3	91.3	92.4	77.0	93.2	93.4	93.4	83.1	76.7	97.8	98.3	98.3	74.0	90.7	95.5	95.9	96.0	73.7	84.1
100	89.0	89.0	89.0	91.9	74.1	91.9	92.1	92.1	84.1	77.5	96.3	96.7	96.8	74.2	87.4	93.8	94.2	94.2	72.8	82.9
250	88.5	88.6	88.6	83.8	74.3	90.8	91.0	91.0	83.9	75.8	95.1	95.6	95.6	73.1	85.8	92.7	93.1	93.2	72.5	83.3
500	-	-	-	-	-	90.0	90.1	90.1	83	74	-	-	-	-	-	-	-	-	-	-

Experimental results with k = 5 also reveal similar sort of results with little difference which were omitted for the sake of briefness.

6.1 Results of Weighted Accuracy

MI Scores based methods performs better here than the counter parts. The best accuracy that the LSI and thresh holding achieves is almost about 94% while MI feature sets achieves as high as 98.3%. Careful investigation reveals that classification based on MI Scores of entire data set have slightly better results then those computed on individual files.

6.2 Results of Spam Recall

The following diagrams summaries the results of the Spam Recall.

Figure 2: Spam Recall values for K = 1

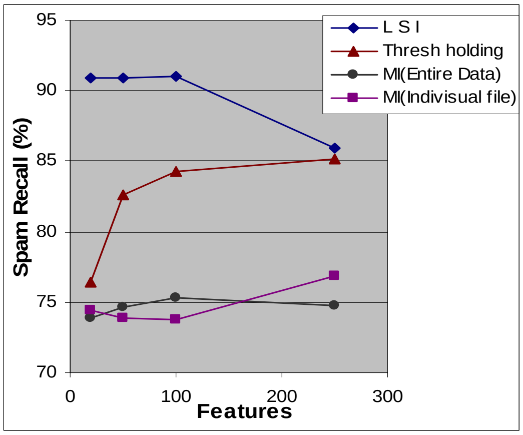
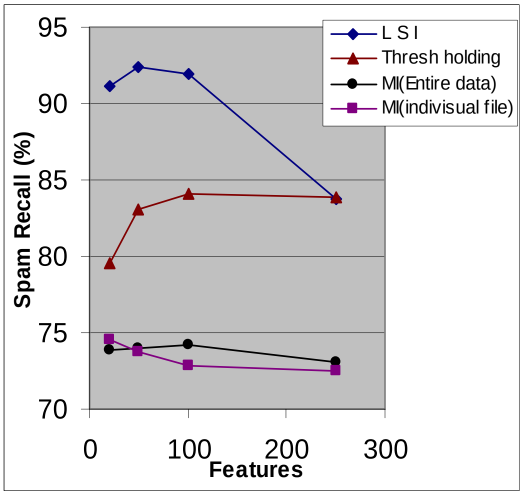


Figure 3: Spam Recall values for K = 3



The Graphs for the Spam Recalls of the four methods (as shown in figure 2, figure 3) reveals LSI to be the obvious winner. Furthermore Thresh holding proves it self to be a strong competitor for LSI at higher feature set. The remaining two MI Scores based methods fails to impress. It can be seen that the MI Scores based techniques remains stable with their results while the other two methods does not.

6.3 Results of Spam Precision

Results with spam precision are quite different from that of spam recall. Here both MI Scores based methods over run the LSI and thresh holding methods. Furthermore, the MI Scores calculated over the entire data set performs better than the ones calculated on the individual files. LSI and thresh holding goes neck to neck but the winner is LSI in terms of numbers. Apart from few exceptions, there is decrease in the spam precision as the feature set increases for all of the four methods.

Figure 3: Spam Precision values for K = 1

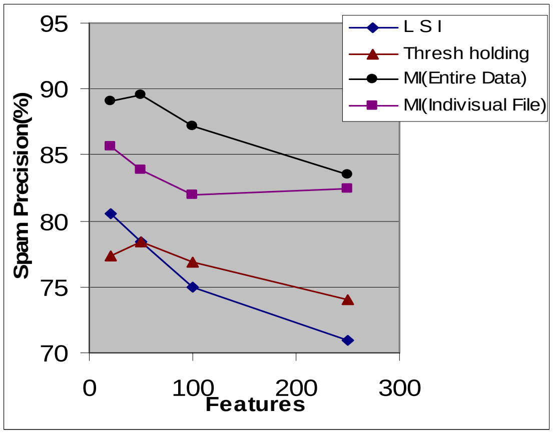
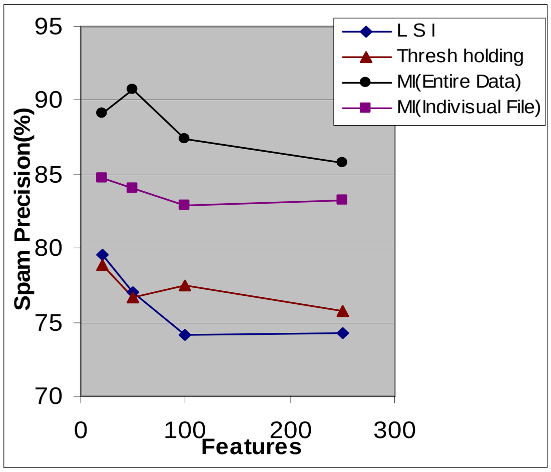


Figure 4: Spam Precision values for K = 3



6.4 Best Feature Set Configuration

Almost all of the feature reduction techniques achieve their best accuracy when the feature set has only 20 attributes. For LSI best precision values corresponds to a transformed features set of 20 while in thresh holding the best values corresponds to feature set of 20 (when k = 3) and 50 (when k= 1). For MI Scores of entire data set the best precision is at 50 while for individual files at 20. Lastly, analysis of spam recall shows that LSI achieves its highest values at 50 and 100 transformed features, thresh holding achieves it at 100 and 250 while in MI scores of entire data set it is achieved at 100 and MI Score of individual file achieves it at 20 and 250.

Following the above discussion it turns out that over all, the best feature sets sizes are 20 and 50. This is really great improvement over the original feature space with thousands of features without sacrificing greatly for accuracy.

7 Conclusions

We performed a thorough analysis of the three feature space reduction techniques on the domain of spam detection with the corpus of ling spam that is publicly available for research purposes. All the three feature reduction techniques were analyzed using cost sensitive measures. These techniques achieve quite high accuracy rate keeping in mind that no phrasal or domain specific features were used. The best of the three seems to be the MI scores based on the entire data set achieving as high as 98.3% of accuracy (with as little as 20 features) and over running all others in spam precision. Further improvement in accuracy can be achieved by adding domain specific, phrasal features and non textual features like attachments and pictures etc.

We are currently investigating other feature reduction techniques which can further help improve the accuracy. Though nearest neighbor seems to have quite better results but one should also look into more algorithms for the classification in order to find the best couple of feature reduction technique and classification algorithm.

8 References

[1]. M.Sahami, S.Dumais, D.Heckerman, and E.Horvitz, A bayesian approach to filtering junk e-mail, *In Learning for Text Categorization Papers from the AAAI Workshop*, pages 55–62, 1998.

[2]. I.An-droutsopoulos, J. Koutsias, K.V. Chandrinos, and D. Spyropoulos, Learning to filter spam email: A comparison of a Naive Bayesian and a memory-based approach. *In Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*. Lyon, France, 1–13, 2000.

[3]. Islam, M. R., Chowdhury, M. U., and Zhou, W, An Innovative Spam Filtering Model Based on Support Vector Machine, *In Proceedings of the international Conference on Computational intelligence For Modeling, Control and Automation and international Conference on intelligent Agents, Web Technologies and internet Commerce* Vol-2 (Cimca-Iawtic'06) CIMCA. IEEE Computer Society, Washington, DC, 348-353, 2005.

[4]. X. Carreras and L. Mrquez, Boosting trees for anti-spam email filtering, *In Proceedings of RANLP-01, Jth International Conference on Recent Advances in Natural Language Processing*, Tzigov Chark, BG, 2001.

[5]. S Günal, S Ergin, ÖN Gerek, Spam E-mail Recognition by Subspace Analysis, *International Symposium on Innovations in Intelligent Systems and Applications*, Yýldýz Technical University 2005.

[6]. S Günal, S Ergin, MB Gülmezolu, ÖN Gerek, On Feature Extraction for Spam E-Mail Detection, Page 1. B. Gunsel et al. (Eds.): MRCS 2006, LNCS 4105, pp. 635–642, 2006.

[7]. Yang, Y. Pedersen, J. O, A comparative study on feature selection in text categorization, *In Proceedings of the 14th International Conference on Machine Learning*, ICML-97. pp. 412-420. 1997.

[8]. Ion Androutsopoulos, John Koutsias, Konstantinos V. Chandrinos, George Paliouras and Constantine D. Spyropoulos, An Evaluation of Naive Bayesian Anti-Spam Filtering, *Proceedings of the workshop on Machine Learning in the New Information Age, G. Potamias, V. Moustakis and M. van Someren (eds.), 11th European Conference on Machine Learning,* Barcelona, Spain, pp. 9-17, 2000.

[9].Fuka, K. and Hanka, R, Feature Set Reduction for Document Classification Problems, *IJCAI-01 Workshop: Text Learning: Beyond Supervision*, Seattle 2001.