Proceedings of the 23rd IASTED International Multi-Conference
ARTIFICIAL INTELLIGENCE AND APPLICATIONS
February 14-16, 2005, Innsbruck, Austria

A COMPARISON OF SUPPORT VECTOR MACHINES, MEMORY-BASED AND NAÏVE BAYES TECHNIQUES ON SPAM RECOGNITION

GULSEN ERYIGIT , A. CUNEYD TANTUG

Department of Computer Engineering
Istanbul Technical University
ITU Ayazaga Kampusu Elektrik-Elektronik Fak. Bilgisayar Muh. Bolumu Maslak
Turkey

gulsen@cs.itu.edu.tr, cuneyd@cs.itu.edu.tr

ABSTRACT

This paper presents a comparison of support vector machines (SVM), memory-based learning (MBL) and Naïve Bayes (NB) techniques for the classification of legitimate and spam mails. Although there are a number of method-comparative studies regarding spam mail filtering, most of the studies are tested on separate data sets. In order to evaluate the effectiveness of SVM, MBL and NB methods, we have used a common publicly available corpus (LINGSPAM). As MBL and NB methods are previously tested with this corpus, the obtained best parameters are used in the experiments with few changes. On the other hand, intense experiments are made to find the best attribute dimensions with SVMs. Results show that SVM has significantly better performance for no-cost and high-cost cases, but NB performs best when the cost is extremely high.

KEY WORDS

Spam Recognition, Support Vector Machines, Memory Based Learning, Naïve Bayes Learning

1. INTRODUCTION

With the rapid growth of the e-mail usage in the recent years, the marketers started to use e-mails as an advertisement opportunity. As e-mails are easy to send and very cheap, the unsolicited commercial messages started to bomb the mailboxes, to waste bandwidth and to fill up file servers. The studies done in the topic of spam mail recognition are not very old: The first technical work on automatic classification of spam and non-spam messages is on 1998 by Sahami et al.[1].

Spam recognition is in essence a two-class classification problem. That is why machine learning techniques are reliable to solve it. The followings are some of the classifiers developed so far: Naïve Bayes [1][2], memory based learning [3], boosting trees [4], support vector machines [5]. In these classifiers, a supervised learning is implemented where the classification is fully automatic except the previously labeled training corpus. The classifiers aim to classify a new incoming e-mail as spam or legitimate according to the knowledge collected from the training stage.

In the classification, each e-mail is considered as a sample and a feature vector is constructed for each of them. Each feature in the feature vector is a word and contains either a binary value showing that the word occurs in the current mail or not, or a number that shows the occurrence frequency of that word in the current mail. These two different representations of features are called multi-variate and multi-nomial [6]. The pre-mentioned classifiers developed for spam filtering use the multi-variate (binary 0/1) model while the work done in [6] emphasizes that some more work should be done on multi-nomial model. Anti-spam filtering differs from other text classification tasks in two ways [3]: first, the spam mails cover a wide spectrum of topics and second it is a cost-sensitive classification area. The misclassification of a legitimate mail as spam is much more crucial than the misclassification of a spam mail as legitimate. There are different cost approaches directly related with what will be done after the classification. The cost should be chosen high for a filter which deletes detected spam messages, similarly the cost can be lower or zero if the filter just marks the e-mails as spam.

The comparison of the results in [1, 2, 3, 4, 5] is impossible as they are not trained on the same corpus and not all of them are formulated within a cost-sensitive framework [3]. In this paper, three spam recognition methods previously examined with different corpora are compared both under a non-cost-sensitive and a cost-sensitive framework. These methods are support vector machines, experimented by Drucker et al. [5] in a non-cost sensitive manner, memory based learning and naïve bayes, compared by Sakkis et al. [3] in a cost-sensitive manner. In the latter one, it is shown that MBL performs

better when the misclassification cost for legitimate messages is high. In the comparison made in this work, it mail is equal to misclassifying a spam mail, SVM performs better on the average than the other two. Also SVM performs very well when the cost is not very high but NB has a better performance when the cost is extremely high.

Subsequent sections describe the details of the used corpus, the feature selection and pattern representation methods to prepare the e-mails for the classification phase, the used classification methods and the comparison of the results obtained from each different method. In the final section, some conclusions and suggestions for future work are given.

2. CORPUS

The experiments are performed on a publicly available e-mail corpus[1] which is a collection of spam and legitimate messages from a mailing list on linguistics "Ling-Spam" [2]. There exist four versions of the corpus which differ from each other by the usage of a lemmatizer which converts each word to its base form and a stoplist that removes from messages the 100 most frequently used words. In this paper, the version with lemmatizer enabled and stoplist enabled is used during the experiments as it is emphasized in [2] that this version performs better for different cost criteria. The corpus consists of 2893 messages: 2412 legitimate and 481 spam. Each version in Ling-Spam is partitioned into 10 parts, with each part maintaining the same ratio of legitimate and spam messages as in the entire corpus. Each experiment was repeated ten times, each time reserving a different part as the testing and using the remaining nine parts as the training corpus.

3. FEATURE SELECTION AND PATTERN REPRESENTATION

The techniques to be compared use multi-variate model as mentioned in the introduction part. That is each mail is considered as a pattern which consists of features with binary values showing the presence or absence of a word in the current mail. Features correspond to words which are selected according to their mutual information calculated with the following formula (1) [1,2,3]. The "n" words with the biggest MI values are selected as the features and each mail is represented by the feature vector $\vec{x} = <x_1, x_2, \ldots . x_n>$.

$$\mathbf{MI}(X,C) = \sum_{x \in \{0,1\}, c \in \{spam, legitimate\}} P(X = x, C = c) . \log_2 \frac{P(X = x, C = c)}{P(X = x).P(C = c)} \quad (1)$$

is seen that when the cost of misclassifying a legitimate

4. CLASSIFICATION METHODS

In this section the classification techniques of NB, MBL and SVM are given. One can look to the following references for a detailed explanation of these methods: Sahami et al. [1], Androutsopoulos et al. [2], Sakkis et al. [3], Drucker et al. [5]. Before going into details with these techniques, the cost-sensitive approach for the classification is given. Mistakenly classifying a legitimate message as spam is generally more severe error than letting a spam message pass the filter that is classifying it as legitimate. Legit→Spam is λ time more costly than Spam→Legit: A mail is classified as spam if the following criterion (2) is met:

$$\frac{P(C = spam \mid \vec{X} = \vec{x})}{P(C = legitimate \mid \vec{X} = \vec{x})} > \lambda \quad (2)$$

In the case of e-mail classification: $P(C = spam \mid \vec{X} = \vec{x}) = 1 - P(C = legitimate \mid \vec{X} = \vec{x})$. As shown with the equations in (3), an instance $\vec{x}$ is classified as spam when the confidence level $W_s(\vec{x})$ is greater than *t* which is a function of λ .

$$\frac{P(C = spam \mid \vec{X} = \vec{x})}{1 - P(C = spam \mid \vec{X} = \vec{x})} > \lambda$$

$$P(C = spam \mid \vec{X} = \vec{x}) > t \qquad t = \frac{\lambda}{\lambda + 1} \quad (3)$$

$$W_s(\vec{x}) > t$$

Additionally, all methods are trained and tested by 10-fold cross-validation technique. The total data set is divided into 10 equal sized parts. In each step, one of these 10 parts is selected and others are used for training. Then the part selected is used for testing. This process repeated ten times for each run.

4.1 Naïve Bayes

From Bayes' theorem and the theorem of total probability, the probability that a document with vector $\vec{x} = <x_1, x_2, \ldots . x_n>$ belongs to category c is:

$$P(C = c \mid \vec{X} = \vec{x}) = \frac{P(C = c).P(\vec{X} = \vec{x} \mid C = c)}{\sum_{k \in \{spam,\, legitimate\}} P(C = k).P(\vec{X} = \vec{x} \mid C = c)} \quad (4)$$

In practice, the probabilities $P(\vec{X}=\vec{x}\,|\,C=c)$ are impossible to estimate without simplifying the assumptions, because the possible values of $\vec{x}$ are too many and there are also data sparseness problems. The Naïve Bayesian classifier assumes that $x_1, x_2, \ldots x_n$ are conditionally independent given the category c, which yields:

$$P\left(C=c\,|\,\vec{X}=\vec{x}\right)=\frac{P(C=c).\prod_{i=1}^{n}P(X_i=x_i\,|\,C=c)}{\sum_{k\in\{spam,legitimate\}}P(C=k).\prod_{i=1}^{n}P(X_i=x_i\,|\,C=c)} \quad (5)$$

4.2 Memory Based Learning

MBL [7] is a differentiated version of a basic k-nearest neighborhood classifier. While K-nn assigns to each new e-mail the majority class among the k closest training e-mails, MBL assigns the majority class among the training e-mails which reside in k closest distances. As a result, if there is more than one neighbor at some of the k closest distances, the neighborhood will contain more than k neighbors. The distance between two instances is computed by the overlap metric which is the same as hamming distance. Given two instances $\vec{x}_i$ and $\vec{x}_j$, their overlap metric $d(\vec{x}_i, \vec{x}_j)$ is calculated as in (6).

$$\vec{x}_i=\langle x_{i1}, x_{i2}.....x_{in}\rangle \quad and \quad \vec{x}_j=\langle x_{j1}, x_{j2}.....x_{jn}\rangle$$
$$\delta(x,y)\equiv\begin{cases}0, & if \quad x=y\\ 1, & otherwise\end{cases} \quad (6)$$
$$d\left(\vec{x}_i,\vec{x}_j\right)\equiv\sum_{r=1}^{n}\delta\left(x_{ir},x_{jr}\right)$$

The confidence level $W_c(\vec{x})$ that an instance $\vec{x}$ belongs to category c is calculated by (7) where $C(\vec{x}_i)$ is the class of the neighbor i. Then, the equation (3) can be used to categorize an e-mail as spam after that the confidence levels are scaled to the interval [0,1].

$$W_c(\vec{x})=\sum_i\left(1-\delta\left(c,C(\vec{x}_i)\right)\right) \quad (7)$$

MBL's performance can be improved by using some weighting schemes. The WMBL's (Weighted Memory Based Learning) weighting schemes are introduced in [3] as attribute weighting and distance weighting.

Distance Weighting

Distance weighting considers neighbors closer to the input instance more important by applying the following formula (8).

$$W_c(\vec{x})=\sum_i f_n(d(\vec{x},\vec{x}_i)).\left(1-\delta\left(c,C(\vec{x}_i)\right)\right) \quad (8)$$

where $f_n(d)=\frac{1}{d^3}$

Attribute Weighting

In the MBL, the attributes are considered as equally as important to each other while this is not the case. That is why attribute weighting in WMBL aims to not treat all attributes as equally important and assigns different importance scores to each attribute according to the following formula (9). The overlap metric $d(\vec{x}_i, \vec{x}_j)$ in (6) becomes as in (9).

$$d\left(\vec{x}_i,\vec{x}_j\right)\equiv\sum_{r=1}^{n}w_r.\delta\left(x_{ir},x_{jr}\right) \quad where$$
$$w_r=H(C)-\sum_{x\in\{0,1\}}P(X=x).H(C\,|\,X=x)$$
$$H\left(C\right)=-\sum_{c\in\{spam,legitimate\}}P\left(C=c\right).\log_2 P(C=c) \quad (9)$$
$$H\left(C\,|\,X=x\right)=-\sum_{c\in\{spam,legitimate\}}P\left(C=c\,|\,X=x\right).\log_2 P(C=c\,|\,X=x)$$

4.3 Support Vector Machines

Vapnik's Support Vector Machines (SVM) [10] is a very useful and effective pattern recognition technique which tries to find a separating hyperplane which maximizes the margin between two classes. SVM is a well-known, 2-class classification method giving good results for high dimensional data sets. A SVM is trained by the following optimization problem:

$$\hat{w}=\arg\min_w\frac{1}{2}\|w\|^2+C\sum_i\xi_i$$
$$y_i(d_i\cdot w+b)\geq 1-\xi_i \qquad \xi_i\geq 0 \quad (10)$$

where each di is a document vector, yi is the label (+1 or -1) for di and w is the vector of weights that defines the optimal separating hyperplane. This form of the optimization is called the "primal." By incorporating the inequality constraints via Lagrange multipliers, we arrive at the "dual" form of the problem,

$$\hat{w}=\arg\max_w\sum_i\alpha_i-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j(d_i\cdot d_j)$$
$$0\leq\alpha_i\leq C \qquad \sum_i\alpha_i y_i=0 \quad \forall i \quad (11)$$

Given optimized values for the αi, the optimal separating hyperplane is

$$\hat{w}=\sum_i\alpha_i y_i d_i \quad (12)$$

The constrained problem above can be solved by quadratic programming. Some fast solving methods like Platt's Sequential Minimal Optimization [11] and Osuna's

method [12]. More information about SVM and solving quadratic problems that is an essence of SVM can be found in [8]. In our work, we have used a library for SVM implementation called LibSVM [9]. The latest version of LibSVM 2.6 has the ability to give us the confidence levels of both classes, which allows us to compare SVM, MBL and Naïve Bayes methods in a cost-sensitive framework. Linear kernel is chosen for solving the quadratic equations.

5. RESULTS

In this section, the results obtained from the implementation of SVM, MBL and NB algorithms are given. Drucker et al. [5] emphasizes that in two-class classification cases, recall (15) and precision rates (16) are useless, the false alarm rate (13) and miss rate (14) should be used instead. However, most of the previous works present their results by means of recall and precision rates. The cost function TCR (Total Cost Ratio) (17) described in [3] is suitable to compare the performances when the classification of a legitimate mail as spam is more costly than the classification of a spam mail as legitimate. The full derivation of TCR function can be found in [3]. "Greater TCR values mean better performance. It can be easily seen from formula (17) that when TCR is less than 1, it is better to not use the filter. So, in our development, all of the three criteria are presented in order to obtain a relation with previous results.

$$FAR\,(False\ Alarm\ Rate) = \frac{N_{S\to L}}{N_S} \tag{13}$$

$$MR\,(Miss\ Rate) = \frac{N_{L\to S}}{N_L} \tag{14}$$

$$RECALL\ \ (R) = \frac{N_{S\to S}}{N_{S\to S} + N_{S\to L}} \tag{15}$$

$$PRESICION\ (P) = \frac{N_{S\to S}}{N_{S\to S} + N_{L\to S}} \tag{16}$$

$$TCR = \frac{N_S}{\lambda N_{L\to S} + N_{S\to L}} \tag{17}$$

Before the comparison of the methods, the parameters ("k" value in MBL and attribute sizes) that give the best results with the used corpus should be determined. Androutsopoulos et al. [2] state that NB gives better results on LINGSPAM with the attribute dimensions dim=100 for λ=1, dim=100 for λ=9, dim=300 for λ=999. In our implementation, we observed that dim=100 for λ=999 gives also better results that dim=300 (Table-1).

Table 1: NB Attr. Size

Method	Dimension	λ=999 TCR
NB	**100**	4.19
NB	**300**	0.15

While using WMBL on LINGSPAM, Sakkis et al. [3] obtain better results with dim=600 and neighborhood size k=8. We obtained better results with k=2 (Table-2) in our experiments.

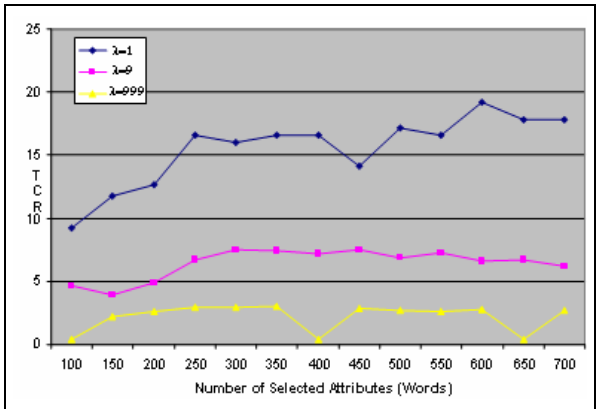
Table 2: WMBL "k" Value

Method	Dimension	λ=1 TCR	λ=9 TCR	λ=999 TCR
WMBL (k=2)	**600**	5.87	3.37	0.15
WMBL (k=8)	**600**	4.86	2.00	0.38

As there is no study in the literature that experiments LINGSPAM with SVM, we experimented the SVM for different attribute sizes varying from 50 to 700 by 50 to find the optimum that gives the better TCR value. Dim=600 is chosen since it gives the better TCR value in the average of the three different cost scenarios (Figure-1).

Comparison of NB, MBL and SVM

The methods are tested with the best observed attributes and the results with FAR/MR are given in Table-3. The results of the simple Memory Based Learning (MBL) is also included in the table so that one can easily see the improvements obtained by attribute and distance weighting.



Dim	λ=1 TCR	λ=9 TCR	λ=999 TCR	Avg TCR
100	9.25	4.63	0.40	4.76
150	11.73	3.94	2.18	5.95
200	12.66	4.90	2.60	6.72
250	16.59	6.68	2.93	8.73
300	16.03	7.51	2.93	8.83
350	16.59	7.40	**3.04**	9.01
400	16.59	7.18	0.42	8.06
450	14.15	**7.51**	2.85	8.17
500	17.18	6.87	2.73	8.93
550	16.59	7.29	2.63	8.83
600	**19.24**	6.59	2.78	**9.54**
650	17.81	6.68	0.41	8.30
700	17.81	6.25	2.70	8.92

Figure 1 – SVM Attribute Size

Table 3: False Alarm/Miss Rate							
Method	**Dim**	**λ=1**		**λ=9**		**λ=999**	
		FAR	**MR**	**FAR**	**MR**	**FAR**	**MR**
MBL (k=2)	**600**	0.3970	0.0000	0.5500	0.0000	0.5500	0.0000
WMBL (k=2)	**600**	0.1470	0.0045	0.2410	0.0012	0.2470	0.0012
NB	**100**	0.1140	0.0029	0.1600	0.0025	0.2390	0.0000
SVM	**600**	0.0350	0.0033	0.1140	0.0008	0.3600	0.0000

As MR (miss rate) increases, the number of misclassifications of legitimate e-mails increases while FAR (false alarm rate) increases, the number of misclassifications of spam e-mails (passing from the filter) increases. So both of FAR and MR should be as small as possible for an acceptable filter (should be 0 for a perfect filter). While considering the cost sensitive cases, the MR is more important and should be penalized more than FAR. Another evaluation criteria set is precision and recall rates. In Table-4, the precision and recall rates are presented per method and different cost values.

Table 4: Precision/Recall							
Meth.	**Dim**	**λ=1**		**λ=9**		**λ=999**	
		R	**P**	**R**	**P**	**R**	**P**
MBL (k=2)	**600**	0.603	1.000	0.451	1.000	0.451	1.000
WMBL (k=2)	**600**	0.852	0.974	0.759	0.992	0.753	0.992
NB	**100**	0.886	0.984	0.840	0.985	0.761	1.000
SVM	**600**	0.965	0.983	0.886	0.995	0.640	1.000

It can be readily seen from Table-4 that SVM performs best for the first two cost values (λ=1, λ=9). For the highest cost value (λ=999) the performance of SVM drastically decreases while WMBL maintains it performance.

Table 5: Total Cost Ratio				
Method	**Dim**	**λ=1 TCR**	**λ=9 TCR**	**λ=999 TCR**
MBL (k=2)	**600**	2.52	1.83	1.83
WMBL (k=2)	**600**	5.87	3.37	0.15
NB	**100**	7.77	3.68	4.19
SVM	**600**	19.26	6.60	2.78

Since TCR represents the performance of the method in the cost-sensitive framework, it can be said as a consequence (Table-5) that SVM has a great performance nearly 3 times greater than the second best method NB when there is no cost (λ=1). SVM has again performs best when the cost value is λ=9**.** But as described in the paragraph above, when the cost is very high (λ=999), NB performs much better than SVM.

6. CONCLUSIONS & FUTURE WORK

This paper aims to compare the performances of Memory Based Learning, Naïve Bayes and Support Vector Machine techniques in the field of spam mail recognition in a cost-sensitive framework. In order to benchmarking the methods and use results of previous related works, a publicly available e-mail list (LINGSPAM) corpus is used. We have implemented NB, MBL and WMBL methods in a cost-sensitive manner and use a library for SVM. The evaluation is performed with three different cost scenarios. Results show that SVM has significantly better performance for no-cost and high-cost cases, but NB performs best when the cost is extremely high. As a consequence, the contribution of our work to spam filtering tasks is the comparison of three methods on the same data set. As an additional contribution, the cost-sensitive version of SVM is applied in the spam filtering subject.

Although spam mail filtering is performed by some pattern recognition techniques, not all of them are tested. Other methods should be implemented for spam filtering and compared with the others. Even some simple methods can achieve surprisingly spam filtering like Naïve Bayes. The linear kernel is used in our tests because a previous work uses this kernel [5] but other kernel types (sigmoid, polynomial, etc.) should be examined and tested in order to find the most effective SVM classification.
The different representations of e-mails are not intensively investigated. Although a multi-variate representation scheme is chosen for keeping the relations with previous works, a multi-nomial representation can have a great impact on the performance of these methods. Investigating effects of other representations is covered by our future objectives. Moreover, some other data which gives clues about the e-mail can be included in representation as domain attributes. In addition to selected words constituting the vector representation of e-mails, the existence of some word patterns like *"FREE MONEY"* can be new attributes. We plan to compare all possible methods and representation schemes in a cost-sensitive framework.

REFERENCES

[1] Sahami, M., S. Dumais, D. Heckerman, and E. Horvitz. 1998. "*A Bayesian Approach to Filtering Junk E-Mail*". Learning for Text Categorization – Papers from the AAAI Workshop, pages 55–62, Madison Wisconsin. AAAI Technical Report WS-98-05.

[2] Androutsopoulos I., Koutsias J., Chandrinos K.V., Paliouras G. and Spyropoulos C.D., 2000. *"An Evaluation of Naive Bayesian Anti-Spam Filtering"*. Proceedings of the workshop on machine Learning in the New Information Age, 11th European Conference on Machine Learning (ECML 2000), Barcelona, Spain, pp. 9-17.

[3] Sakkis G., Androutsopoulos I., Paliouras G., Karkaletsis V.,Spyropoulos C.D.;Stamatopoulos P., 2003, "*A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists*", Information Retrieval 6(1), 49-73, Kluwer Publishing

[4] Boosting Trees for Anti-Spam Email Filtering (2001)Xavier Carreras, Lluís Marquez, Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing

[5] Drucker H., Wu D., Vapnik V.N., 1999. "*Support Vector Machines for Spam Categorization*", IEEE Transactions On Neural Networks, pages 1048-1054.

[6] Karl-Michael Schneider, A Comparison of Event Models for Naive Bayes Anti-Spam E-Mail Filtering, 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 03), pp. 207-314, 2003.

[7] Androutsopoulos I., Paliouras G., Karkaletsis V., Sakkis G., Spyropoulos C.D., Stamatopoulos P., 2000, "*Learning to filter Spam E-Mail: AComparison of a Naïve Bayesian and a Memory-Based Approach* ", Proc. of the workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, France

[8] Duda, R.O. and P.E. Hart. 1973. "*Bayes Decision Theory*". Chapter 2 in Pattern Classification and Scene Analysis, pages 10–43. John Wiley.

[9] Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm

[10] Vapnik, 1995. "The Nature of Statistical Learning Theory". Springer-Verlag, 1995.

[11] J. C. Platt, 1998. "*Sequential minimal optimization: A fast algorithm for training support vector machines*," in Advances in Kernel Method: Support Vector Learning, Scholkopf, Burges, and Smola, Eds. Cambridge, MA: MIT Press, pp. 185–208.

[12] E. Osuna, R. Freund, and F. Girosi, 1997. "*Improved training algorithm for support vector machines*," in Proc. IEEE NNSP'97.