Available online at www.sciencedirect.com
ScienceDirect
Knowledge-Based Systems xxx (2008) xxx–xxx
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

A comparative study for content-based dynamic spam classification using four machine learning algorithms

Bo Yu [a,*], Zong-ben Xu [b]

[a] School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China
[b] Institute for Information and System Science, School of Science, Xi'an Jiaotong University, Xi'an 710049, China

Received 11 December 2007; accepted 15 January 2008

Abstract

The growth of email users has resulted in the dramatic increasing of the spam emails during the past few years. In this paper, four machine learning algorithms, which are Naïve Bayesian (NB), neural network (NN), support vector machine (SVM) and relevance vector machine (RVM), are proposed for spam classification. An empirical evaluation for them on the benchmark spam filtering corpora is presented. The experiments are performed based on different training set size and extracted feature size. Experimental results show that NN classifier is unsuitable for using alone as a spam rejection tool. Generally, the performances of SVM and RVM classifiers are obviously superior to NB classifier. Compared with SVM, RVM is shown to provide the similar classification result with less relevance vectors and much faster testing time. Despite the slower learning procedure, RVM is more suitable than SVM for spam classification in terms of the applications that require low complexity.

Keywords: Spam classification; Naïve Bayesian; Neural network; Support vector machine; Relevance vector machine

1. Introduction

With the development of the information and network technology, individuals and organizations more and more rely on the emails to communicate and share information and knowledge. However, spam, also known as unsolicited commercial or bulk email, is a bane of email communication [1]. A study estimates that over 70% of the business emails are spam [2]. These spam emails not only consume users' time and energy to identify and remove the undesired messages, but also cause many problems such as taking up limited mailbox space, wasting network bandwidth and engulfing important personal emails.

Many methods have been proposed to deal with these problems [3]. These methods can be grouped into two categories which are static methods and dynamic methods. Static methods base their spam email identification on a predefined address list. For example, Helfman and Isbell strove to build into Ishmail filtering system that users were able to program simple rules for filtering emails into different priority folders [4]. Many email clients have similar, yet less sophisticated, filtering subsystems that allow the users to filter and prioritize the emails. However, constructing and maintaining rules for filtering is a burdensome task [5].

Compared to static methods based their spam email identification on a predefined address list, dynamic methods take the contents of the emails into consideration and adapt their spam filtering decisions with respect to these contents. The technique of filtering is based on a list of words and phrases that characterize spam messages. Most of them use general text categorization and data mining techniques by implementing machine learning methods. Naïve Bayesian algorithms have been generally used by training a classifier on manual spam email filtering. The using of Bayesian formula as a tool to identify spam is initially applied to spam filtering by Sahami et al. For their

* Corresponding author. Tel./fax: +86 10 68949443.
E-mail addresses: xjtuyubo@gmail.com (B. Yu), zbxu@mail.xjtu.edu.cn (Z. Xu).

0950-7051/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.knosys.2008.01.001

model, they used the 500 highest frequency tokens as a binary feature vector, 35 hand-crafted rule phrases, and 20 hand-crafted non-textual features as the sender's domain [6]. Other researchers implemented subsequently Bayesian filters with a high positive precision and a low negative recall [7,8]. However, Carpinter and Hunt pointed out that filters generally used 'Naïve' Bayesian filtering, which assumes that the occurrence of events is independent of each other; i.e. such filters do not consider that the words 'special' and 'offers' are more likely to appear together in spam email than in legitimate email [9].

There have been a few studies in applying neural network (NN). The main disadvantage of NN is that it requires considerable time for parameter selection and network training. On the other hand, previous researches have shown that NN can achieve very accurate results, that are sometimes more accurate than those of the symbolic classifiers [10]. Levent et al. also carried out the experiments that a total of 750 emails (410 spams and 340 normal emails) were used and a success rate of about 90% was achieved [11,12]. These studies showed that neural network can be successfully used for automated email filing into mailboxes and spam email filtering.

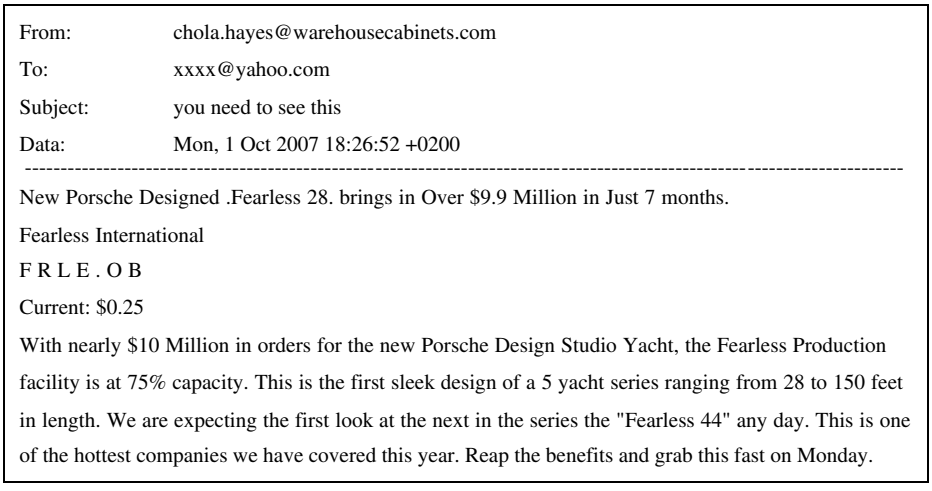
Support vector machine (SVM) is a classification method that directly minimizes the classification error without requiring a statistical data model [13,14]. This method is popular because of its simple implementation and consistently high classification accuracy when applied to many real-world classification situations. Drucker et al. applied the technique to spam filtering, testing it against three other text classification algorithms: Ripper, Rocchio and boosting decision trees. Both boosting trees and SVM provided "acceptable" performances, with SVM given lesser training requirements [15].

Although SVM provides a state-of-the-art technique to tackle this problem, Relevance vector machine (RVM), that relies on Bayesian inference learning, offers advantages such as its capacity to find sparser and probabilistic solutions [16,17]. Silva and Ribeiro found that RVM for text classification could surpass other techniques, both in terms classification performance and response time [18]. Begüm and Sarp also proved that approximately the same classification accuracy was obtained using RVM-based classification, with a significantly smaller relevance vector rate and consequently much faster testing time, compared to SVM-based classification [19].

In this paper, four machine learning algorithms which are NB, NN, SVM and RVM are proposed as dynamic anti-spam filtering methods to compare their performances. For each algorithm, we develop it by changing the topologies of the networks and adjusting some parameters to achieve its best possible predicted result. The paper is organized as follows. Section 2 discusses the morphological analysis of the spam and spam filtering. In Section 3, we explain the algorithms based on the four above mentioned methods, respectively, developed for the classification of emails. The details and the results of the experiments are presented in Section 4. The last section is the conclusions.

2. Content-based spam filtering

The email consists of two parts, one is the body message and another part is called the header, as shown in Fig. 1. The job of the header is to store information about the message and it contains many fields, for example, tracing information about which a message has passed (Received:), authors or persons taking responsibility for the message (From:), intending to show the envelop address of the real sender opposed to the sender used for replying (Return-Path:). Firstly, the email should be pretreated by removing the useless structure information, left with the sender, subject and content. Subsequently, the document containing text is extracted. Each document in the email is expressed as a vector. The dimensions in the vector are corresponding to the word frequency existing in the document. Feature



From: chola.hayes@warehousecabinets.com
To: xxxx@yahoo.com
Subject: you need to see this
Data: Mon, 1 Oct 2007 18:26:52 +0200

New Porsche Designed .Fearless 28. brings in Over $9.9 Million in Just 7 months.

Fearless International

F R L E . O B

Current: $0.25

With nearly $10 Million in orders for the new Porsche Design Studio Yacht, the Fearless Production facility is at 75% capacity. This is the first sleek design of a 5 yacht series ranging from 28 to 150 feet in length. We are expecting the first look at the next in the series the "Fearless 44" any day. This is one of the hottest companies we have covered this year. Reap the benefits and grab this fast on Monday.

Fig. 1. An example of spam email.

Please cite this article in press as: B. Yu, Z. Xu, A comparative study for content-based dynamic spam ..., Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.01.001

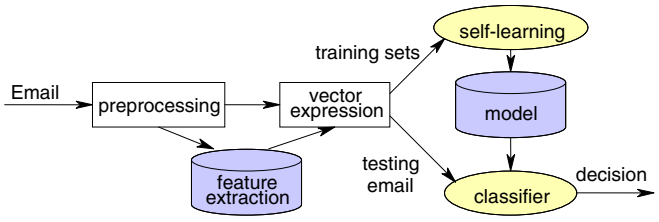


Fig. 2. The process of spam filtering.

extraction procedures will process the words into lowercase, remove empty words, take root of words, etc., to meet the required text feature.

The architecture of spam filtering is shown in Fig. 2. Firstly, the model will collect individual user emails which are considered as both spam and legitimate email. After collecting the emails the initial transformation process will begin. This model includes initial transformation, the user interface, feature extraction and selection, email data classification, and analyzer section. Machine learning algorithms are employed at last to train and test whether the demanded email is spam or legitimate.

3. Classification algorithms

3.1. Naïve Bayesian classifier

Naïve Bayesian algorithm is one of the frequently used machine learning methods for text categorization. The original idea of identifying whether an email is spam or not by looking at which words are found in the message and which words are absent from it. This approach begins by studying the content of a large collection of emails which have already been classified as spam or legitimate email. Then when a new email comes into some user's mailbox, the information gleaned from the "training set" is used to compute the probability that the email is spam or not from the words in the email.

Given a feature vector $\vec{x} = \{x_1, x_2, \ldots, x_n\}$ of an email, where are the values of attributes $X_1, \ldots, X_n$, and n is the number of attributes in the corpus. Let C denote the category to be predicted, i.e., $C \in \{\text{spam, legitimate}\}$. It uses a discriminate function to compute the conditional probabilities of $P(C_i|X)$. Here, given the inputs, $P(C_i|X)$ denotes the probability that, example X belongs to class C_i:

$$P(C_i|X) = \frac{P(C_i) \times P(X|C_i)}{P(X)} \tag{1}$$

$P(C_i)$ is the probability of observing class i. $P(X|C_i)$ denotes the probability of observing the example, given class C_i. $P(X)$ is the probability of the input, which is independent of the classes. A simple example of Naïve Bayesian filtering is shown in Fig. 3.

3.2. Neural network

The neural network adopted in this paper is a standard non-linear feed-forward network with the sigmoid activation function

$$f(x) = \frac{1}{1 + e^{-x}}. \tag{2}$$

Both the hidden and output neurons are shown in Fig. 4. This activation function will produce an output in the range $[0;1]$. A network with one input for each word in the vocabulary, a single hidden layer and one output neuron are constructed. The input and hidden layer contain a bias neuron with a constant activity of 1.

When presenting an email to the network, all inputs corresponding to words found in the message are set to 1, and all other inputs are set to 0. The email is classified as junk if the output value is above 0.5. When training the network, the desired output is set to 0.1 for the good emails and 0.9 for the junk emails. The number of good emails and the number of junk emails presented to the network should not differ too much. If the number of emails from one of the categories is much larger than that of the other, the network might learn to always answer yes or always answer no, simply because that would be true most of the time for a bad training set.

Provided that we extract the following keywords from an e-mail:

Benefits (0.85) million (0.15), current (0.1) reap (0.12) need (0.2)

Avalue of 0.85 for benefits indicates 85% of previously seen emails that included that word were ultimately classified as spam, with the remaining 15% classified as legitimate email.

To calculate the overall probability (P) of an e-mail being spam:

$$P = \frac{x_1 \cdot x_2 \cdots x_n}{x_1 \cdot x_2 \cdots x_n + (1-x_1)\cdot(1-x_2)\cdots(1-x_n)}$$

$$= \frac{0.85 \cdot 0.15 \cdot 0.1 \cdot 0.12 \cdot 0.2}{0.85 \cdot 0.15 \cdot 0.1 \cdot 0.12 \cdot 0.2 + (1-0.85)\cdot(1-0.15)\cdot(1-0.1)\cdot(1-0.12)\cdot(1-0.2)}$$

$$= 0.00377$$

This value indicates that it is unlikely that the email message is spam; however, the ultimate classification decision would depend on the decision boundary set by the filter.

Fig. 3. A simple example of Naïve Bayesian filtering.

Please cite this article in press as: B. Yu, Z. Xu, A comparative study for content-based dynamic spam ..., Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.01.001

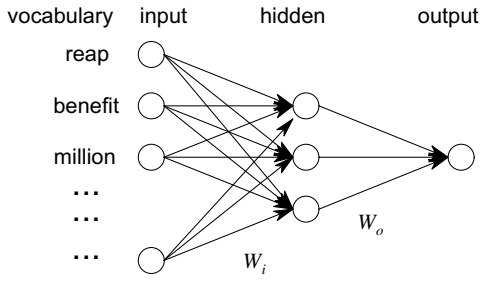


Fig. 4. The structure of neural network.

A gradient descent training algorithm using back propagation of errors is applied for optimizing the weights in the network. Optimization is done by minimizing output mean square error:

$$\min_{W} \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} (O_i - O_{\text{desired},i})^2. \tag{3}$$

3.3. Support vector machine

SVM was originally developed by Vapnik and his coworkers [13,14]. SVM embodies the Structural Risk Minimization (SRM) principle to minimize an upper bound on the expected risk [20]. Because structural risk is a reasonable trade-off between the training error and the modeling complication, SVM has a great generalization capability. An operating model of SVM is shown in Fig. 5. The SVM algorithm seeks to maximize the margin around a hyperplane that separates a positive class (marked by circles) from a negative class (marked by squares). When using SVM for pattern classification, the basic idea is to find the optimal separating hyperplane that gives the maximum margin between the positive and negative samples. According to this idea, spam filtering can be viewed as a simple possible SVM application, classification of linearly separable classes; that is, a new email belongs to the spam category or not.

The key concepts we want to use are the following: there are two classes, $y_i \in \{-1,1\}$, and there are ℓ labeled training examples: $(\mathbf{x}_1,y_1), \ldots (\mathbf{x}_\ell,y_\ell)$, $\mathbf{x} \in \boldsymbol{R}^d$ where d is the dimensionality of the vector. Consider that the primal optimization problem for the maximal margin case is the following:

$$\begin{aligned} &\min_{w,b} \quad \langle \mathbf{w} \ldots \mathbf{w} \rangle, \\ &\text{subject to} \quad y_i(\langle \mathbf{w} \ldots \mathbf{x}_i \rangle + b) \geqslant 1, \quad i = 1, \ldots, \ell. \end{aligned} \tag{4}$$

In order to optimize the margin slack vector we need to introduce slack variables ξ_i to allow the margin constrains to be violated

$$\begin{aligned} &\text{subject to} \quad y_i(\langle \mathbf{w} \ldots \mathbf{x}_i \rangle + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell, \\ &\qquad\qquad \xi_i \geqslant 0, \quad i = 1, \ldots, \ell. \end{aligned} \tag{5}$$

2-norm soft margin contains the ξ_i scaled by the norm of the weight vector $\mathbf{w}$.

Suggesting that an optimal choice for C in the objective function of the resulting optimization problem should be R^{-2}:

$$\begin{aligned} &\min_{\xi,w,b} \quad \langle \mathbf{w} \ldots \mathbf{w} \rangle + C \sum_{i=1}^{N} \xi_i^2 \\ &\text{subject to} \quad y_i(\langle \mathbf{w} \ldots \mathbf{x}_i \rangle + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, \ell, \\ &\qquad\qquad \xi_i \geqslant 0, \quad i = 1, \ldots, \ell. \end{aligned} \tag{6}$$

In practice the parameter C is varied through a wide range of values and the optimal performance assessed using a separate validation set or a technique known as cross-validation for verifying performance using only the training set. The training vector $\boldsymbol{X}$ is mapped into a higher (maybe infinite) dimensional feature space $\boldsymbol{F}$ by the function ϕ. Then SVM finds a linear separating hyperplane with maximal margin γ in a higher dimensional space. $K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \ldots \phi(\mathbf{z}) \rangle$ is called kernel function. It is well known that the choice of the kernel function is crucial to the efficiency of support vector machines. The four types of kernel functions (linear, polynomial, RBF, sigmoid) frequently used with SVM. In this paper, sigmoidal kernel is adopted in the experiments.

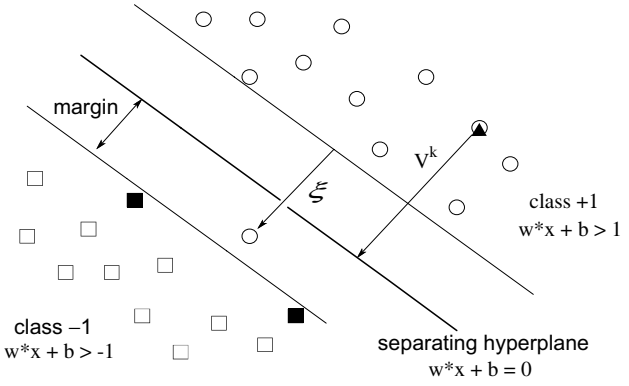


Fig. 5. An operating mode of SVM.

3.4. Relevance vector machine

Despite of the excellent classification performance of SVM, it also has some practical and significant drawbacks. Although relatively sparse, SVM makes unnecessarily liberal using of basis functions since the number of support vectors required typically grows linearly with the size of the training set. Post processing is often required to reduce computational complexity. There is no straightforward method to estimate C and ξ. Sometimes cross validation is used to estimate which is wasteful for both data and computation. The kernel function $K(\mathbf{x},\mathbf{z})$ must satisfy Mercer's condition.

In 2001, Michael E. Tipping proposed relevance vector machine [16]. RVM is a Bayesian treatment of Eq. (6) which does not suffer from any of the limitations stated above. Tipping introduces Lagrange coefficient vector $\mathbf{w}$

as the weight parameters and builds a significant degree framework. The sample corresponding to the obtained nonzero **w** is called relevance vector (RV). RV is equivalent to support vector (SV) in SVM. But for the same training set, the number of RV is less than that of SV of SVM. Thus RVM has a short testing time compared to SVM. However, RVM needs to keep iterative calculations for hyper-parameters, and it has a long training time. Following Tipping [16], the mathematical classification model of RVM is presented here.

Given a dataset of input-target pairs $\{x_n, t_n\}_{n=1}^{\ell}$, the conditional distribution is generalized by applying the logistic sigmoid function $\sigma(y) = 1/(1+e^{-y})$to $y(x)$ and writing the likelihood as

$$P(t|w) = \prod_{n=1}^{N} \sigma\{y(\mathbf{x}_n)\}^{t_n}[1-\sigma\{y(x_n)\}]^{1-t_n} \tag{7}$$

However, we have not the weights to obtain the marginal likelihood analytically, and so utilize an iterative procedure based on that of MacKay [21]:

(1) For the current, fixed, values of α, the most probable weights $\mathbf{w}_{\mathrm{MP}}$ (the location of the posterior mode) is found. This is equivalent to a standard optimization of a regularized logistic model. The efficient iteratively reweighted least squares algorithm to find the maximum.
(2) The Hessian at $\mathbf{w}_{\mathrm{MP}}$ is computed:

$$\nabla\nabla \log p(\mathbf{t}, \mathbf{w}|\alpha)|_{\mathbf{w}_{\mathrm{MP}}} = -(\phi^T \mathbf{B}\phi + \mathbf{A}) \tag{8}$$

where $\mathbf{B}_{nn} = \sigma\{y(\mathbf{x}_n)\}[1-\sigma\{y(\mathbf{x}_n)\}]$, and this is negated and inverted to give the covariance $\sum$ for a Gaussian approximation to the posterior over weights, and from that the hyper-parameters α are updating using the following equation ($\gamma_i = 1 - \alpha_i \sum_{ii}$;$\mu_i$ is ith of posterior weighted average):

$$\alpha_i^{\mathrm{new}} = \frac{\gamma_i}{\mu_i^2} \tag{9}$$

This procedure is repeated until some suitable convergence criteria are satisfied. Note that in the Bayesian treatment of multilayer neural network, the Gaussian approximation is considered a weakness of the method if the posterior mode is unrepresentative of the overall probability mass. However, for the RVM, we note that $p(t,w|\alpha)$ is log-concave which has better Gaussian approximation.

4. Experimental results

4.1. Spam corpora

There are several known and well-defined collections of legitimate and spam messages and many researchers use them as a benchmark to compare the performances [22]. Two standard benchmark corpora, 6000 emails with the spam rate 37.04% gotten from *SpamAssassin*[1] sets and 5000 emails with the spam rate 45.04% gotten from *Babletext*[2] sets, are required to allow meaningful comparison of the reported results of new spam filtering techniques against existing systems.

$$A = \frac{\#\text{ email correctly classified}}{\text{Total \# of emails}}$$

$$SR = \frac{\#\text{ spam correctly classified}}{\text{Total \# of spam messages}}$$

$$SP = \frac{\#\text{ spam correctly classified}}{\text{Total \# of messages classified as spam}}$$

Fig. 6. Common experimental measures for the evaluation of spam filters.

4.2. Preprocessing work

It is necessary to convert the emails into a suitable format for the classification algorithms (namely the extraction of the plain text from the subject and body field of the emails) in the process of constructing an automatic classifier. A generic architecture for text categorization, called LINGER [10], is adopted in this paper. It supports the bag of words representation which is the most commonly used in text categorization. All unique terms (tokens, e.g. numbers, words, special symbols, etc.) in the entire training corpus are identified and each of them is treated as a single feature. A feature selection is applied to choose the most important words and reduce dimensionality. Each document is then represented by a vector that contains a normalized weighting for every word according to its importance.

4.3. Performance measurement

In this section, four classification methods, including NB, NN, SVM and RVM, are evaluated the effects based on different data sets and different feature sizes. The performance of spam filtering is often measured in terms of accuracy (A), spam precision (SP) and spam recall (SR), which are most important performance parameters. Accuracy is the percentage of all emails that are correctly classified by the classifier. SR is the proportion of spam emails in the test set that are classified as spam, i.e. the spam emails that the filter manages to block. It measures the effectiveness of the filter. SP is the proportion of emails in the test data classified as spam that are truly spam, i.e. it measures filter's protection or overprotection ability. Discarding a legitimate email is of the greatest concern to most users than classifying a spam message as legitimate email. This means that high SP is particularly important. Detail definition formulas for the three evaluation measuring parameters are listed in Fig. 6.

[1] Availability: http://www.spamarchive.org and http://spamassassin.apache.org.
[2] Availability: http://www.babeltext.com/spam/.

The first experiments are run with different training and testing sets. The pairs of training and testing sets are created by splitting each corpus at a ratio from 20:80 to 70:30 respectively. The experiment is performed with 100 features extracted from LINGER. The accuracy performances in the case of different training dataset sizes for the two corpora are listed in Table 1.

The other experiments measuring the *SP* and *SR* performance against the size of dataset are conducted using different features from 60 to 140 using LINGER. The most frequent words in spam email are selected as features. The pairs of training and testing set are created by splitting each corpus at a ratio 40:60. Comparison results on measuring *SP* and *SR* are shown in Fig. 7 for *SpamAssassin* corpus and in Fig. 8 for the *Babletext* corpus.

In order to compare the performance of the RVM and SVM classification with different number of features, Fig. 9 shows the number of SVs (Support Vectors) for the SVM classification and RVs (Relevance Vectors) for the RVM classification.

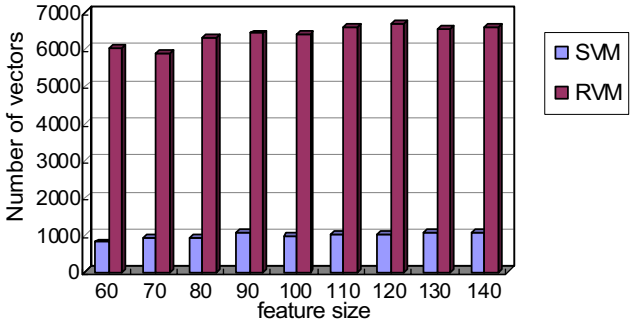


Fig. 9. Number of SVs for SVM and number of RVs for RVM under different feature size.

Table 1
The accuracy for the *SpamAssassin* and *Babletext* corpus using four methods

Size (training: testing)	Method (*SpamAssassin*/*Babletext*)			
	NB	NN	SVM	RVM
20:80	92.7%/93.8%	85.3%/87.7%	95.2%/96.0%	96.1%/93.9%
30:70	91.3%/90.4%	86.6%/89.8%	94.8%/95.4%	95.1%/94.6%
40:60	90.7%/92.0%	86.1%/86.3%	96.3%/96.4%	94.8%/94.6%
50:50	94.0%/94.5%	92.4%/85.9%	95.8%/94.6%	94.2%/95.8%
60:40	92.2%/92.2%	83.5%/90.2%	97.0%/95.9%	95.6%/96.3%
70:30	91.8%/93.1%	84.0%/84.2%	96.0%/96.1%	96.0%/96.5%

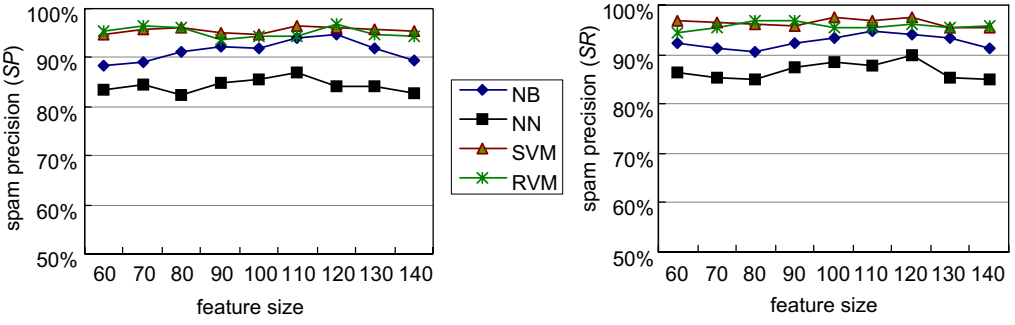


Fig. 7. Spam precision and spam recall under different feature size for *SpamAssassin* corpus.

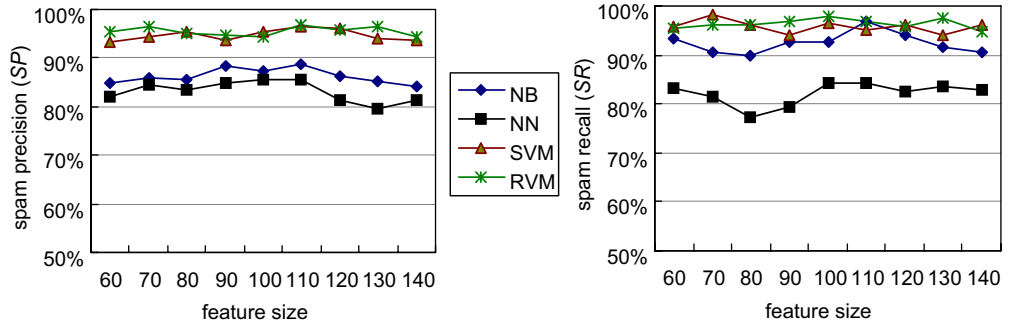


Fig. 8. Spam precision and spam recall under different feature size for *Babletext* corpus.

4.4. Results

A few of observations can be made from these experiments. As shown in Table 1, training dataset sizes have a more obvious impact on the accuracy of NN classifier. NN classifier is more sensitive to the change of training set because the parameters of NN model must be decided upon network size and training algorithm control parameters, and the generalization ability is poor. NN is also likely to be overfitted by the training data. For these reasons, NN is unsuitable to use alone as a spam rejection tool. However, the other three methods seem to be less influenced by the training set size and different data sets. Overall, the accuracy of SVM and RVM classifier is higher than NB classifier. Furthermore, the RVM and SVM classifiers only depends on the relevance or support vectors, and the classifier function is not influenced by the whole data set, as it is the case for many neural network systems.

We also obtain the above similar conclusions observed from Figs. 7 and 8. Besides that, with the increasing of the feature set size, the precision and recall rate start to increase gradually and then decrease. The rate has a maximum value near at the number of 110. It can be explained that there are not enough features to reflect the overall content of the emails and there are too many features may introduce a classification noise by some independent noisy features. NB classifier and NN classifier show an obvious fluctuation with the changes of feature numbers. However, it only has a minor impact on SVM and RVM methods. The characteristics of SVM and RVM make them have the possibility to efficiently deal with a very large number of features due to the exploitation of kernel functions. It is seen that the classification performances of RVM and SVM are more robust with a similar behavior under feature increasing.

As shown in Fig. 9, RVM always results in a significantly smaller number of RVs compared with the number of SVs obtained in SVM. In the Section 3, we know that RVM is superior to SVM in terms of the number of kernel functions that needs to be used in the classification stage. Therefore, RVM is preferable to SVM in terms of the time performance in the test phase. However, it has to be noted that the training time of RVM is longer than SVM because the updating rules for the hyper-parameters depend on computing the posterior weight covariance matrix.

5. Conclusions

Spam is becoming a very serious problem to the Internet community, threatening both the integrity of the networks and the productivity of the users. In this paper, we propose four machine learning methods for anti-spam filtering and present an empirical evaluation for them on the benchmark spam filtering corpora *SpamAssassin* and *Babletext*. These approaches include Naïve Bayesian, neural network, support vector machine and relevance vector machine. Two experiments are carried out to test the performances of these algorithms by changing the training set size and extracted feature size. Experimental results show that NN classifier is more sensitive to the training set size and unsuitable for using alone as a spam rejection tool. Generally, the performances of the SVM and RVM classifiers are less influenced by data sets and feature sizes, and obviously superior to the NB classifier. Compared with SVM, RVM is shown to provide similar classification with a significantly smaller RV rate and much faster testing time. However, the learning procedure of RVM is normally much slower than SVM. Hence, the RVM classification is more suitable to the SVM classification in terms of applications that require low complexity.

Acknowledgement

The research work in this paper is supported by National Natural Science Foundation of China, under the Grant No. 70531030.

References

[1] P.O. Boykin, V.P. Roychowdhury, Leveraging social networks to fight spam, IEEE Computer 38 (4) (2005) 61–68.
[2] Aladdin Knowledge Systems, Anti-spam white paper, <http://www.eAladdin.com>.
[3] L. Özgür, T. Güngör, F. Gürgen, Spam mail detection using artificial neural network and Bayesian filter, in: 5th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL-2004), Exeter, UK, August 2004, Proceedings LNCS, vol. 3177, pp. 505–510.
[4] J. Helfman, C. Isbell, Ishmail: immediate identification of important information, AT& T Labs Technical Report, 1995.
[5] J. Rennie, iFile: an application of machine learning to e-mail filtering, in: Proceedings of KDD-2000 Text Mining Workshop, Boston, 2000.
[6] M. Sahami, S. Dumais, D. Heckerman, E. Horvitz, A Bayesian approach to filtering junk e-mail, in: Learning for Text Categorization: Papers from the 1998 Workshop, Madison, Wisconsin. AAAI Technical Report WS-98-05, 1998.
[7] McCallum, K. Nigam, A comparison of event models for Naive Bayes text classification, in: Sahami, M. (Ed.), Proceedings of AAAI Workshop on Learning for Text Categorization, Madison, WI, 1998, pp. 41–48.
[8] Androutsopoulos, J. Koutsias, K.V. Chandrinos, et al. An evaluation of Naïve Bayesian anti-spam filtering, in: Proceedings of Workshop on Machine Learning in the New Information Age, Barcelona, 2000, pp. 9–17.
[9] J. Carpinter, R. Hunt, Tightening the net: a review of current and next generation spam filtering tools, Computers and Security 25 (2006) 566–578.
[10] Clark I. Koprinska, J. Poon, A neural network based approach to automated e-mail classification, in: Proc. IEEE/WIC International Conference on Web Intelligence (WI), 2003, pp. 702–705.
[11] L. Özgür, T. Güngör, F. Gürgen, Adaptive anti-spam filtering for agglutinative languages: a special case for Turkish, Pattern Recognition Letters 25 (2004) 1819–1831.
[12] L. Özgür, T. Güngör, F. Gürgen, Spam mail detection using artificial neural network and bayesian filter. IDEAL 2004, LNCS 3177 (2004) 505–510.
[13] V. Vapnik, Estimation of Dependencies Based on Empirical Data, Springer-Verlag, New York, 1992.
[14] V. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.

[15] H. Drucker, D. Wu, V. Vapnik, Support vector machines for spam categorization, IEEE Transactions on Neural Networks 10 (5) (1999) 1048–1054.
[16] E. Tipping, The relevance vector machine, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems, vol. 12, MIT Press, Cambridge, MA, 2000.
[17] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, Journal of Machine Learning Research 1 (2001) 211–244.
[18] S. Catarina, R. Bernardete, RVM ensemble for text classification, International Journal of Computational Intelligence Research 3 (1) (2007) 31–35.
[19] B. Demir, S. Ertürk, Hyperspectral image classification using relevance vector machines, IEEE Geoscience and Remote Sensing Letters 4 (4) (2007) 586–590.
[20] C.J. Lin, Foundations of support vector machines: A note from an optimization point view, Neural Computation 13 (2) (2001) 307–317.
[21] D.J.C. Mackay, The evidence framework applied to classification networks, Neural Computation 4 (5) (1992) 720–736.
[22] V. Zorkadisa, D.A. Karrasb, M. Panayotou, Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering, Neural Networks 18 (2005) 799–807.

Please cite this article in press as: B. Yu, Z. Xu, A comparative study for content-based dynamic spam ..., Knowl. Based Syst. (2008), doi:10.1016/j.knosys.2008.01.001