Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Large margin classification for combating disguise attacks on spam filters*

Xi-chuan ZHOU†1, Hai-bin SHEN2, Zhi-yong HUANG1, Guo-jun LI3

([1]College of Communications Engineering, Chongqing University, Chongqing 400044, China)
([2]Institute of VLSI Design, Zhejiang University, Hangzhou 310027, China)
([3]Chongqing Communication Institute, Chongqing 400032, China)
†E-mail: zxc@cqu.edu.cn

Received Sept. 2, 2011; Revision accepted Oct. 25, 2011; Crosschecked Feb. 8, 2012

Abstract: This paper addresses the challenge of large margin classification for spam filtering in the presence of an adversary who disguises the spam mails to avoid being detected. In practice, the adversary may strategically add good words indicative of a legitimate message or remove bad words indicative of spam. We assume that the adversary could afford to modify a spam message only to a certain extent, without damaging its utility for the spammer. Under this assumption, we present a large margin approach for classification of spam messages that may be disguised. The proposed classifier is formulated as a second-order cone programming optimization. We performed a group of experiments using the TREC 2006 Spam Corpus. Results showed that the performance of the standard support vector machine (SVM) degrades rapidly when more words are injected or removed by the adversary, while the proposed approach is more stable under the disguise attack.

Key words: Large margin, Spam filtering, Second-order cone programming (SOCP), Adversarial classification
doi:10.1631/jzus.C1100259 **Document code:** A **CLC number:** TP393.098

1 Introduction

Statistical classifiers such as support vector machines (SVM) have been widely used for spam filtering applications (Drucker *et al.*, 1999). They offer high accuracy and the ability to detect novel spam messages. However, as statistical learning systems become more and more popular, the motivation to defeat them increases. It was reported that a spammer could potentially profit \$25 000 even with the response rate being as low as 0.001% (Carpinter and Hunt, 2006), resulting in over \$10 billion spam-related cost worldwide (Jennings, 2005). In this paper, we address the challenge of spam filtering based on large margin classification in the presence of an adversary who strategically disguises the spam massages to avoid being detected.

Recently, spammers have developed more sophisticated methods to circumvent spam filters. Lowd and Meek (2005b) demonstrated a good word attack (GWA) that adds words indicative of a legitimate message to spam messages, making spam messages appear closer to legitimate messages. The authors showed that, by adding as few as 150 good words, the average spam messages could pass off as non-spam. Webb *et al.* (2005) also examined the effectiveness of the GWA on statistical spam filters. Their experimental results showed that, on clean emails where no good words are injected, filters based on the Naive Bayes or standard SVM classifiers could achieve an accuracy of as high as 98%. However, when testing on the messages that have been injected with good words, the classification accuracy of the

* Project supported by the National Natural Science Foundation of China (No. 61103212) and the Natural Science Foundation of CQ CSTC, China (No. cstcjjA40005)

filters could drop by more than 40%.

To resist the GWA, Jorgensen *et al.* (2008) proposed a multiple instance learning strategy, which splits a message into a bag of multiple segments. Each segment was treated as an instance. An email was classified as spam if at least one instance in the corresponding bag was spam, and as legitimate if all the instances in it were legitimate. They showed that the multiple instance strategy was very effective against the GWA. Besides adding good words, however, the adversary could also delete the words indicative of spam from the spam messages. The multiple instance approach was not designed for this attack strategy. Our experiments showed that the performance of the multiple instance classifier is sensitive to the missing bad words.

In a more theoretical perspective of the adversarial learning research, Dalvi *et al.* (2004) explored the possibility of anticipating attacks by computing the adversary's optimal strategy. They modeled the computation of the adversary's strategy as a constrained optimization problem and approximated its solution based on dynamic programming. Their experiments showed that the game-theoretic approach outperforms traditional classification algorithms in the presence of an adversary. However, Dalvi *et al.* (2004) assumed that both the classifier and the adversary have perfect knowledge of each other, which is not common in practice.

Lowd and Meek (2005a) extended Dalvi *et al.*'s work by removing the assumption of prior knowledge. They proposed an adversarial reverse engineering learning approach based on the concept 'adversarial cost'. They assumed that each feature of the spam data has a different utility to the adversary, and the adversary could modify the spam data only to some extent to keep the utility of the data. For example, some words in the spam messages, such as 'deals', 'drug', and 'price', have more commercial value than others. Removing these bad words will damage the utility of the involved spam messages. To describe the adversarial cost, Lowd and Meek (2005a) used the L_1-norm of the modification vector. In this paper, we use the L_2-norm of the modification vector to represent the adversarial cost. This is mainly because the L_1-norm loss function is not continuous, which makes the problem difficult for convex relaxation.

In this paper we focus on the large margin classification based spam filtering with an adversary strategically adding good words to or deleting bad words from the spam mails. Both strategies can be modeled as changing the features of the spam data, which we generally call 'disguise attack'. We follow the assumption that the adversary affords to lose only limited utility by modifying the spam messages (Lowd and Meek, 2005a). Furthermore, in the L_2-norm based adversary cost function an 'untrusted region' is introduced for each spam data. To improve the robustness of large margin classification under the disguise attack, we use the second-order cone programming (SOCP) formulation, which minimizes the errors over both the training points and the untrusted regions.

One important issue in computing the adversarial cost is how to decide the utility of different features. In this paper, we estimate the feature-wise utility using the Bayes method, which leads to larger utility weights for the bad words. This result matches the intuition that the bad words are usually more valuable for the spammer.

Attempts have been proposed to improve the robustness of SVM. Shivaswamy *et al.* (2006) proposed a Robust SVM formulation to handle the Gaussian noise in the training data. Their research focused mainly on handling environmental noise instead of intentional attacks. A group of researchers also proposed several SVM reformulations with modified hinge loss to improve robustness against outliers (Song *et al.*, 2002; Krause and Singer, 2004; Xu *et al.*, 2006; Wu and Liu, 2007). These attempts usually employ nonlinear hinge loss that ceases to increase after a certain point, resulting in non-convex formulation or non-scalable algorithms. Recently, Chechik *et al.* (2008) proposed a max-margin classifier which is robust when some features are missing from the data. This reformulation could potentially resist the bad-word-removing attack, but it is not robust under the good-word-injection attack.

2 Adversarial cost assumption

Inspired by Lowd and Meek (2005a), we model the disguise attack of adding good words or removing bad words as: the adversary tries to disguise the spam data $\boldsymbol{x}$ with $\boldsymbol{x}^+$ by strategically changing features. The adversarial cost function represents the increased cost (or decreased utility) of using some instances as compared to others. Different from Dalvi *et al.* (2004), the cost function used in this paper is defined as the weighted squared difference between feature values in the disguise instance, $\boldsymbol{x}^+$, and those in the original instance $\boldsymbol{x}$:

$$L(\boldsymbol{x}, \boldsymbol{x}^+) = \sum_{i=1}^{D} q_i(x_i^+ - x_i)^2 = (\boldsymbol{x}^+ - \boldsymbol{x})^{\mathrm{T}} \boldsymbol{Q}(\boldsymbol{x}^+ - \boldsymbol{x}), \tag{1}$$

where D is the number of words that occur in the corpus. The weight $q_i \geq 0$ represents the cost of modifying the feature corresponding to the ith word, admitting that some features have more value than others. $\boldsymbol{Q}$ is a diagonal matrix with $Q_{ii} = q_i$. We assume the adversary can afford to modify the spam message only to some extent, specifically, described by threshold parameter γ:

$$L(\boldsymbol{x}, \boldsymbol{x}^+) = (\boldsymbol{x}^+ - \boldsymbol{x})^{\mathrm{T}} \boldsymbol{Q}(\boldsymbol{x}^+ - \boldsymbol{x}) \leq \gamma^2. \tag{2}$$

One important issue in defining the adversarial cost function is the way to calculate q_i. In practice, the weight q_i should be decided according to the adversary. The features corresponding to the bad words should have larger weights, because they are more valuable for the spammer. In this work, we estimate the utility value to be the posterior probability conditional on the event of spam, which could be calculated using the following Bayes model:

$$q_i = \frac{p(f_i|D_{\mathrm{s}})}{p(f_i|D_{\mathrm{s}}) + p(f_i|D_{\mathrm{l}})}, \tag{3}$$

where $p(f_i|D_{\mathrm{s}})$ represents the probability of the ith word occurring in the spam corpus, and $p(f_i|D_{\mathrm{l}})$ represents the probability of the ith word occurring in the legitimate corpus. When the ith word is a good word, Eq. (3) results in a small utility value. Words indicative of spam have larger weights. Note that, as feature selection is common practice in spam filtering, not all words in the corpus affect the classification result.

3 Secure support vector machine

Spam filters based on the standard SVM are known to be vulnerable to adversarial attacks (Lowd and Meek, 2005b). Specifically, the adversary could disguise the spam data $\boldsymbol{x}$ with $\boldsymbol{x}^+$ by strategically changing features. In this work, we focus on this type of attack and present a large margin training formulation which is more robust under this attack. We begin with introducing the SOCP based large margin training formulation proposed by Debnath *et al.* (2004).

Given a set of n training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is drawn from a domain $\mathcal{X}$ and each of the label y_i is an integer from $\mathcal{Y} = \{+1, -1\}$, the goal of binary-class classification in SVM is to train a model in which the correct label is assigned to unseen test samples. The SOCP based learning process of SVM can be written as

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{b}} \sum_{i=1}^{n} \xi_i \\ &\text{s.t. } y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots, n, \\ &\qquad \xi_i \geq 0, \quad i = 1, 2, \ldots, n, \\ &\qquad \|\boldsymbol{w}\| \leq C. \end{aligned} \tag{4}$$

The remainder of this section focuses on reformulating Eq. (4) to improve classification robustness under the disguise attack. Given the adversarial cost threshold γ as shown in Eq. (2), the untrusted region of the spam data $\boldsymbol{x}$ is defined as

$$\mathcal{D}_{\boldsymbol{x}} = \{\boldsymbol{x}^+ | L(\boldsymbol{x}, \boldsymbol{x}^+) \leq \gamma^2\}.$$

By strategically deleting or adding words in the spam messages, the adversary could change the spam data $\boldsymbol{x}$ to any point $\boldsymbol{x}^+$ in $\mathcal{D}_{\boldsymbol{x}}$. To resist this attack, we consider the conservative strategy of labeling all the points in the untrusted region $\mathcal{D}_{\boldsymbol{x}}$ as spam. Formally, this strategy can be written as

$$y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^+ + b) \geq 1, \ \ \forall \boldsymbol{x}^+ \in \mathcal{D}_{\boldsymbol{x}}. \tag{5}$$

Note that each point in the untrusted region can be represented by a constraint function. Thus, the above inference leads to an infinite number of constraints. Therefore, before incorporating them in the SOCP formulation (4), we need to reduce the number of constraints. The main results are presented in the following theorem:

Theorem 1 Suppose $\boldsymbol{x}$ represents a spam message and an adversary may select any point $\boldsymbol{x}^+ \in \mathcal{D}_{\boldsymbol{x}}$ to disguise $\boldsymbol{x}$. Then Eq. (5) is equivalent to

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \geq 1 + \gamma \| \boldsymbol{Q}^{-\frac{1}{2}} \boldsymbol{w} \|. \tag{6}$$

The proof of Theorem 1 is straightforward, given the following lemma:

Lemma 1 Given the spam data $\boldsymbol{x}$ and the weights $\boldsymbol{w}$, the following optimization

$$\mathrm{argmin}_{\boldsymbol{x}^+\in\mathcal{D}_{\boldsymbol{x}}}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}^+ \tag{7}$$

has the minimum value of $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}-\gamma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{w})^{\frac{1}{2}}$, which can be achieved at

$$\boldsymbol{x}^+=\boldsymbol{x}-\gamma(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-1}\boldsymbol{w})^{-\frac{1}{2}}\boldsymbol{Q}^{-1}\boldsymbol{w}.$$

Proof First we substitute $\boldsymbol{v}=\boldsymbol{Q}^{\frac{1}{2}}(\boldsymbol{x}^+-\boldsymbol{x})$, and the minimization problem (7) can be equivalently reformulated as

$$\min_{\boldsymbol{v}}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\boldsymbol{v}\quad\text{s.t.}\quad\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}\le\gamma^2.$$

Since $\boldsymbol{w}$ is known, due to the Cauchy-Schwartz inequality, we have

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\boldsymbol{v}\ge-\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\|\;\|\boldsymbol{v}\|\ge-\gamma\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\|. \tag{8}$$

And the minimal value is achieved at

$$\boldsymbol{v}^*=\frac{-\gamma\boldsymbol{Q}^{-\frac{1}{2}}\boldsymbol{w}}{\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\|}. \tag{9}$$

Given the minimal point $\boldsymbol{v}^*$, inequality (8) leads to

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\boldsymbol{v}^*=-\gamma\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\|. \tag{10}$$

Substituting the definition of $\boldsymbol{v}$ in Eq. (10) yields $\boldsymbol{w}^{\mathrm{T}}(\boldsymbol{x}^+-\boldsymbol{x})=-\gamma\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\|$, which proves the first claim in Lemma 1. And substituting the definition of $\boldsymbol{v}$ in Eq. (9) yields $\boldsymbol{x}^+=\boldsymbol{x}-\gamma\|\boldsymbol{w}^{\mathrm{T}}\boldsymbol{Q}^{-\frac{1}{2}}\|^{-1}\boldsymbol{Q}^{-1}\boldsymbol{w}$, which proves the second claim in Lemma 1.

The virtue of Lemma 1 is that, given the spam point $\boldsymbol{x}$, it analytically describes the disguise point $\boldsymbol{x}^+$ that is most likely to pass the filter as a legitimate message. In other words, the large margin classifier can correctly classify the whole untrusted region if it correctly classifies the points described by Lemma 1. Furthermore, the infinite number of constraint functions can be reduced to the one described in Theorem 1.

Before incorporating the constraint (6) into Eq. (4), one needs to consider the inseparable situation. We introduce an extra variable ϵ_i to describe the tolerable error over the untrusted region of the spam data $\boldsymbol{x}_i$. Suppose there are s spam messages and l legitimate messages out of an n-message training set. The messages in the training corpus are sorted so that the first s messages are all spam and the rest are legitimate. Thus, the training process of the Secure SVM can be written as

$$\begin{gathered}\min_{\boldsymbol{w},b,\boldsymbol{\xi},\boldsymbol{\epsilon}}\sum_{i=1}^{s}\epsilon_i+\sum_{j=1}^{l}\xi_j\\ \text{s.t.}\quad y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i+b)\ge1-\epsilon_i+\gamma\|\boldsymbol{Q}^{-\frac{1}{2}}\boldsymbol{w}\|,\\ i=1,2,\ldots,s,\\ y_{s+j}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_{s+j}+b)\ge1-\xi_j,\ j=1,2,\ldots l,\\ \epsilon_i\ge0,\ \xi_j\ge0,\ i=1,2,\ldots,s,\ j=1,2,\ldots,l,\\ \|\boldsymbol{w}\|\le C,\end{gathered} \tag{11}$$

where $y_i=1$ if $\boldsymbol{x}_i$ represents a spam message and $y_i=-1$ otherwise. Note that the adversary is interested only in altering the spam data to avoid being detected.

The standard SVM training is formulated as a quadratic programming optimization with $2n$ constraint functions, while the proposed Secure SVM takes the form of SOCP with $2n+1$ constraint functions. The revised formulation becomes more robust against the disguise attack at the expense of more computational cost. In practice, Eq. (11) can be solved using interior point optimization methods with publicly available solvers.

4 Experimental setting

We designed a series of experiments to evaluate the effectiveness of the proposed large margin classifier for spam filtering in the presence of an adversary. Our experimental data consists of 37 822 spam and legitimate messages (12 910 legitimate and 24 912 spam) from the 2006 TREC Spam Corpus. We preprocessed the entire corpus by stripping HyperText Markup Language (HTML) and non-textual parts and applying stemming and a stop list to all terms. Messages that had an empty body after preprocessing were discarded.

We sorted the emails in the corpus chronologically by the received date and divided them almost evenly into 18 subsets $\{S_1,S_2,\ldots,S_{18}\}$. In other words, the messages in subset n come chronologically before the messages in subset $n+1$. Experiments were run in an online fashion, that is, training on subset n $(n=1,2,\ldots,17)$ and testing on subset $n+1$. Each subset contained approximately 2100 messages. The percentage of the spam messages in each subset varied (Fig. 1).

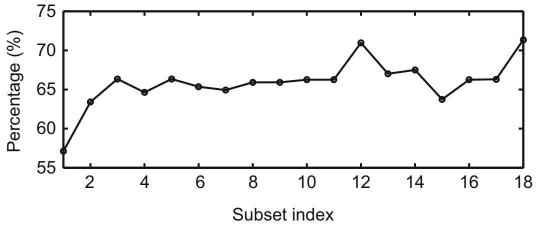

Fig. 1 The percentage of spam messages in the 18 subsets

4.1 Feature calculation and selection

The value for each feature was calculated using the common term frequency–inverse document frequency (TF–IDF) scheme. Suppose f is the number of occurrences of the given term in the given message. We first normalized f by dividing it by the maximum value of f for the given term over all messages in the corpus. The inverse document frequency was then calculated by $\log_2(n_1/n_2)$, where n_1 is the total number of messages in the corpus and n_2 is the number of messages in the corpus that contains the given term. Then the feature corresponding to the given term is calculated as $\mathrm{norm}(f) \times \log_2(n_1/n_2)$.

Since it is common practice to apply feature selection before classification in spam filtering, we reduced the features used for describing the messages to the top features ranked using a greedy-forward-selection (GFS) method. The GFS method starts with an empty feature set, and adds the feature whose addition provides the Naive Bayes classifier with the best five-cross-validation performance. Fig. 2 shows the dependance of classification accuracy on the number of features selected. We used the top 400 features in our experiments, which provided a good compromise among the classifiers in terms of efficiency and performance.

4.2 Attacking words selection

We evaluated the proposed algorithm under two types of attack. In the first attack strategy, GWA, the adversary selects a list of good words indicative of a legitimate message and adds these words to the spam messages. In the second attack strategy, addressed as the bad word attack (BWA) in the following, the adversary selects a list of bad words indicative of a spam message and strategically removes these words from the spam messages.

Lowd and Meek (2005b) described several strategies of GWA on spam filters. The most effective are the frequent word attack (FWA) and frequency ratio attack (FRA). FWA disguises the spam mails by adding words that occur most often in legitimate messages. FRA adds words that occur very often in legitimate messages but not in spam messages. We evaluated the revised large margin classifier under both types of attack. For FWA, we ranked every unique word in the whole corpus according to its frequency in the legitimate messages. For FRA, we ranked every unique word in the whole corpus according to the ratio of its frequency in the legitimate messages to its frequency in the spam messages. For both types of attack, we selected the top M words from the ranking as our good word list. We considered the scenario in which the classifier does not have knowledge about the value of M, which is decided according to the adversary with respect to an affordable cost.

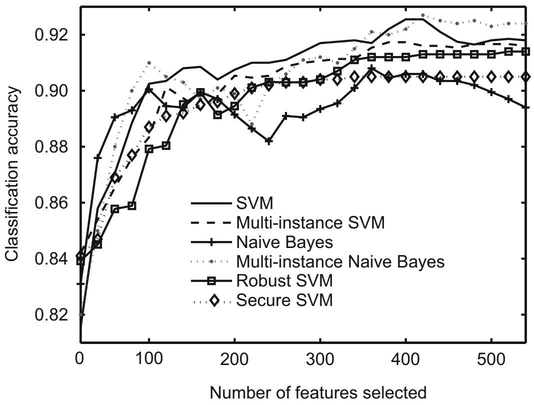

Fig. 2 Effect of the number of features selected on classification accuracy

For BWA, we ranked every unique word in the whole corpus according to the ratio of its frequency in the spam messages to its frequency in the legitimate messages. Then we selected the top M words from the ranking as our bad word list. In practice, if the given spam message does not contain any word in the bad word list, the classification result of the message is not affected.

Note that, since the lists were generated from the entire corpus rather than from the training subset, and since we represented emails using a vector of 400 features, some of the words in the lists would not affect the classification of the message, no matter whether they were injected into or removed from the message. Such lists were more representative of what an adversary would be able to produce in practice, since the adversary would have no way of knowing the exact features used by the target filter.

5 Experiment results

We now present the results of the experiments in which we evaluated the effectiveness of the proposed Secure SVM classifier. In the first experiment, we tested the algorithms on the emails that had been injected with good words. In the second experiment, we tested the algorithms on emails with some bad words being deleted. We compared the proposed classifier with the standard SVM, the Naive Bayes classifier, and the Robust SVM proposed in Shivaswamy *et al.* (2006). We used the LIBSVM package (Chang and Lin, 2011) in our experiments for the standard SVM. The SOCP optimization of the proposed algorithm was realized using the MOSEK tool (MOSEK, 2011).

Since the multi-instance learning strategy has recently been reported to be robust against GWA (Lowd and Meek, 2005b), we extended SVM and Naive Bayes to their multi-instance variants. Specifically, each message was evenly split into two segments. Each segment was treated as one instance, and the multi-instance variants predicted the test message to be a spam message if one of the segments was classified as spam. We split an email down the middle into two approximately equal halves. This splitting method was called split-half (split-H) in Jorgensen *et al.* (2008). The messages in the TREC 2006 Spam Corpus were sorted chronologically and split into 18 subsets. All the experiments were performed in an online fashion. Specifically, the classifiers were trained on subset S_n $(n = 1, 2, \ldots, 17)$ and tested on subset S_{n+1}. The averaged classification accuracy over the test subsets was calculated and reported in the remainder of this section.

5.1 Classification under good word attack

For the first experiment, we chose the words from the top of the good word list and injected one instance of each selected word into the spam messages. We tested the algorithms with a range (M=0–200) of words injected and fixed γ to 5.0. Note that, we injected good words randomly into only 50% of the test spam messages, because in practice, the disguised spam emails account for only a subset of the spam emails processed by a given filter.

Fig. 3 illustrates the change in classification accuracy while the number of good words injected increased from 0 to 200. The attacking method in Fig. 3a was named FWA because the good words were selected according to their frequency of occurring in the legitimate messages. As one can see, SVM showed the highest classification accuracy (over 0.923) when no words were injected. But the SVM performance dropped dramatically while more good words were injected. Similar results were obtained for the Naive Bayes classifier. Specifically, by adding 200 words from the good word list, the classification accuracy of SVM and the Naive Bayes classifier dropped to 0.562 and 0.516, respectively. In contrast, the multi-instance strategy substantially improved the robustness of SVM under GWA. The accuracy of the multi-instance SVM was higher than that of the standard SVM by 0.067 with 200 words injected.

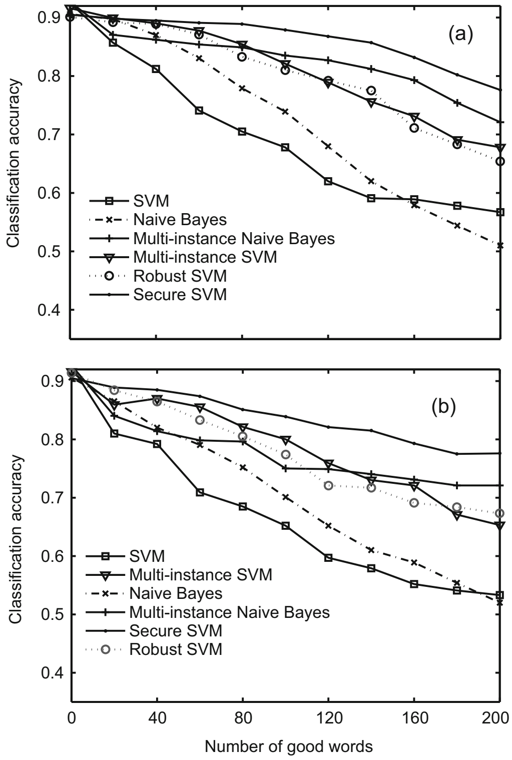

Fig. 3 Comparison of the classifiers under the frequent word attack (a) and frequent ratio attack (b)

Fig. 3b shows the results for FRA, which was more effective than FWA in terms of degradation in the performance of the SVM classifier. For both types of attack, the proposed large margin classifier had the best performance, and its accuracy was higher than that of the standard SVM by over 0.08, when more than 40 words were injected in the spam mails.

5.2 Classification under bad word attack

For the second experiment, we chose the words from the top of the bad word list and deleted one instance of the selected word if it occurred in the given message. Similarly, we tested the algorithms by deleting different numbers of bad words. In this experiment, we removed the bad words randomly from only 50% of the test spam messages due to the same reason as in the first experiment.

Fig. 4 illustrates the change in classification accuracy while the number of bad words to be deleted increased from 0 to 100. As one can see, the SVM performance dropped dramatically while more bad words were added to the deleting list, with a classification accuracy of 0.535 when the number of bad words in the deleting list reached 100. Similar results could be obtained for the Naive Bayes classifier. Different from the first experiment, the multi-instance SVM was not robust for the bad word deleting attack. Specifically, the accuracy of the multi-instance SVM was higher than that of the standard version by only about 0.021 when 20 bad words were checked to be deleted. Compared to the multi-instance classifiers and Robust SVM, the proposed large margin classifier was less sensitive when words indicative of spam were deleted, and its accuracy was higher than that of the standard SVM by over 0.12 when more than 20 bad words were deleted from the spam mails.

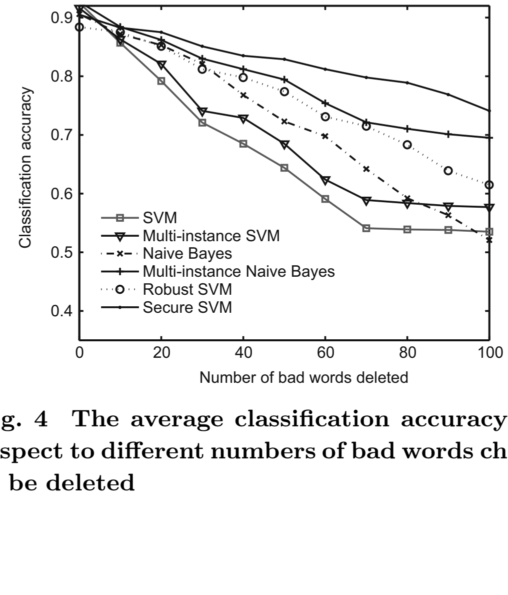

Fig. 4 The average classification accuracy with respect to different numbers of bad words checked to be deleted

5.3 Parameter selection

One important issue in Secure SVM is the selection of the parameter γ, which represents the threshold of the adversarial cost. Intuitively, γ should be large enough so that the classifier could allow the spam messages to be disguised without misclassification. To select an appropriate γ, we implemented the proposed large margin classifier with a range of γ. Fig. 5 shows the change in classification accuracy under FWA when γ was varied. In practice, the selecting of γ depends on the method of estimating the weight q_i in the adversarial cost. With the Bayes method proposed to estimate q_i, the performance of the Secure SVM classifier was not very sensitive to γ as long as it exceeded a certain value. On the other hand, large values of γ were not advisable either, because the classification error over the untrusted region increased significantly in our experiment when γ exceeded 20, resulting in the decline of classification accuracy. Note that the adversary who could afford more cost to disguise the spam messages will reduce the performance of the proposed classifiers to a lower extent.

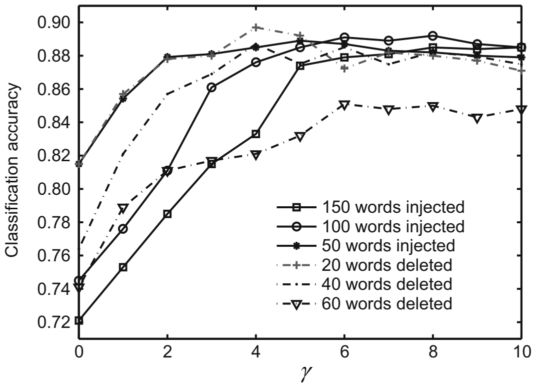

Fig. 5 The average classification accuracy with respect to different values of adversarial cost γ

Another metric to evaluate the classifiers is the area under the receiver operating characteristic (ROC) curve, AUC. In terms of AUC, the presented algorithm was better than SVM when more than 40 words were injected or 20 words were deleted. The AUC of the secure variant of SVM with 200 words injected was 0.917, higher than that of the standard SVM by 0.152. As the value of the threshold increased from 1 to 6, the AUC stayed in the region between 0.932 and 0.953 with 40 words injected. When a larger value of the threshold was used, the AUC began to drop. One reason for the decrease is that, as the threshold value increases, more legitimate messages could be found in the untrusted regions, resulting in a higher false positive rate.

Fig. 6 shows the average classification accuracy with 40 good words injected in the testing data. Similar to the standard SVM, the accuracy of the proposed algorithm was not very sensitive to the value of C as long as C was larger than a certain threshold, which was determined by the training data. In our experiments, C was in the range of 0.01 to 100.

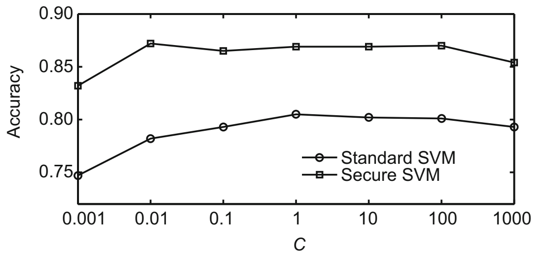

Fig. 6 Classification accuracy with respect to different values of C as given in Eq. (11)

5.4 Training on attacked messages

In the previous experiments, we trained all the classifiers with clean messages. However, the training corpus may be polluted by the disguised spam messages. In this experiment, we still tested the classifiers on the 18 chronologically sorted data sets in an online fashion. In addition to randomly modifying 50% of the testing spam messages, we modified 10% of the training spam messages. We considered the attacks of both adding good words and removing bad words.

For GWA, we selected different numbers (M=0–200) of good words in the legitimate messages. Both the training and testing spam messages were injected with the selected good words. All the classifiers were trained using the polluted corpus. Different from Jorgensen *et al.* (2008), we did not assume that the words injected in the training set and the testing set are the same. Instead, we randomly selected two groups of words from the good word list and injected them in the training and testing sets, respectively. The classification accuracy results are shown in Fig. 7a. Similarly, the SVM performance dropped dramatically while more good words were injected. Similar results were obtained for the Naive Bayes classifier. By adding 120 words from the good word list, the classification accuracy of SVM and the Naive Bayes classifier dropped to 0.533 and 0.520, respectively, lower than that of training with a clean corpus by about 0.1. The accuracy of the multi-instance SVM was higher than that of the standard version by 0.101 with 160 words injected. The Secure SVM outperformed other classifiers when more than 40 words were injected, and its accuracy dropped by 0.092 with 200 words injected in the training spam message.

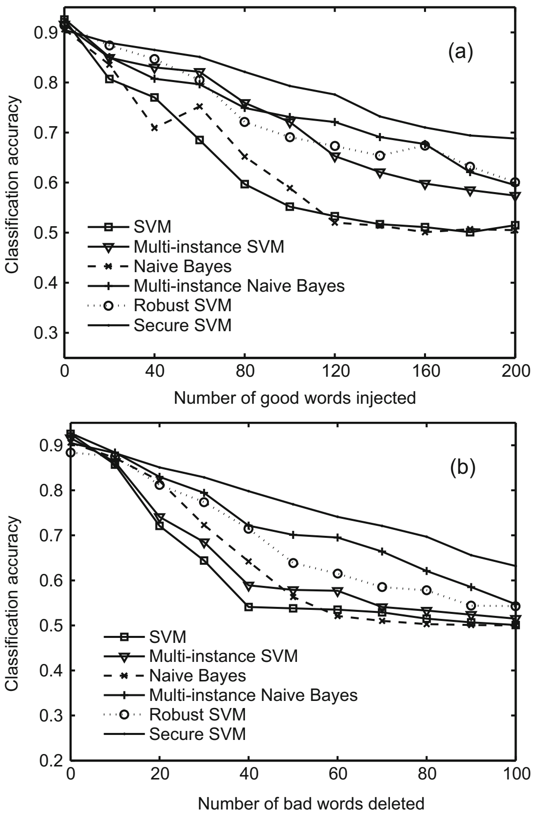

Fig. 7 Comparison of classifiers trained with good word attack (a) and bad word attack (b) messages

For BWA, we selected 10% of the training spam messages and 50% of the testing spam messages. The top words in the bad word list were deleted if they occurred in the selected spam messages. All the classifiers were trained using the polluted corpus. Fig. 7b shows the classification accuracy results under this attack for each of the classifiers. Compared with the SVM trained with a clean corpus, the polluted training corpus caused a drop of 0.18 in terms of classification accuracy for the standard SVM when

40 bad words were deleted. The classification accuracy of the Secure SVM also dropped by 0.149, but still higher than those of other classifiers.

6 Conclusions and future work

In this paper we propose a revised SVM training algorithm which is robust under the disguise attack in a spam filtering application. We assume that the adversary may strategically add good words or delete bad words, but the spammer could afford to modify a spam message only to some extent, without damaging its utility. To improve the robustness of SVM, we define an adversarial cost function, which leads to an untrusted region for each spam data. The weights for the feature-wise disguise cost are estimated using the Bayes method. We consider a conservative strategy of labeling all the points in the untrusted region as spam, resulting in an infinite number of constraint functions. Then, we reformulate the constraint functions and further incorporate them into an SOCP based Secure SVM formulation. Results of experiments using the 2006 TREC Spam Corpus show the effectiveness of the proposed algorithm.

One important advantage of SVM is that it can be reformulated into its dual form and further generalized for nonlinear classification using the kernel trick. After reformulating Eq. (11) into its dual form, we find that it cannot be generalized to its kernel form due to such terms as $y_i\boldsymbol{x}_i$ in the constraint functions. Instead of applying the kernel trick in the dual form, Chapelle (2007) generalized the standard SVM for nonlinear classification in the primal form. In the future, we plan to follow her inspiration for nonlinear generalization of Secure SVM formulation.

References

Carpinter, J., Hunt, R., 2006. Tightening the net: a review of current and next generation spam filtering tools. *Comput. Secur.*, **25**(8):566-578. [doi:10.1016/j.cose.2006.06.001]

Chang, C., Lin, C., 2011. LIBSVM: a library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, **2**:27:1-27:27.

Chapelle, O., 2007. Training a support vector machine in the primal. *Neur. Comput.*, **19**(5):1155-1178. [doi:10.1162/neco.2007.19.5.1155]

Chechik, G., Heitz, G., Elidan, G., Abbeel, P., Koller, D., 2008. Max-margin classification of data with absent features. *J. Mach. Learn. Res.*, **9**:1-21.

Dalvi, N., Domingos, P., Mausam, Sanghai, S., Verma, D., 2004. Adversarial Classification. Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, p.99-108. [doi:10.1145/1014052.1014066]

Debnath, R., Muramatsu, M., Takahashi, H., 2004. The Support Vector Machine Learning Using the Second Order Cone Programming. Proc. IEEE Int. Joint Conf. on Neural Networks, **4**:2991-2996.

Drucker, H., Wu, D., Vapnik, V.N., 1999. Support vector machines for spam categorization. *IEEE Trans. Neur. Networks*, **10**(5):1048-1054. [doi:10.1109/72.788645]

Jennings, R., 2005. The Global Economic Impact of Spam. Technical Report, Ferris Research, San Diego, CA, USA.

Jorgensen, Z., Zhou, Y., Inge, M., 2008. A multiple instance learning strategy for combating good word attacks on spam filters. *J. Mach. Learn. Res.*, **9**:1115-1146.

Krause, N., Singer, Y., 2004. Leveraging the Margin More Carefully. Int. Conf. on Machine Learning.

Lowd, D., Meek, C., 2005a. Adversarial Learning. Proc. 11th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining.

Lowd, D., Meek, C., 2005b. Good Word Attacks on Statistical Spam Filters. Proc. 2nd Conf. on Email and Anti-Spam.

MOSEK, 2011. The MOSEK Optimization Tools Version 6.0. User's Manual and Reference 2011. Available from www.mosek.com

Shivaswamy, P.K., Bhattacharyya, C., Smola, A.J., 2006. Second order cone programming approaches for handling missing and uncertain data. *J. Mach. Learn. Res.*, **7**:1283-1314.

Song, Q., Hu, W., Xie, W., 2002. Robust support vector machine with bullet hole image classification. *IEEE Trans. Syst. Man Cybern. C*, **32**(4):440-448. [doi:10.1109/TSMCC.2002.807277]

Webb, S., Chitti, S., Pu, C., 2005. An Experimental Evaluation of Spam Filter Performance and Robustness Against Attack. 1st Int. Conf. on Collaborative Computing: Networking, Applications and Worksharing, p.19-21.

Wu, Y., Liu, Y., 2007. Robust truncated hinge loss support vector machines. *J. Am. Stat. Assoc.*, **102**(479):974-983. [doi:10.1198/016214507000000617]

Xu, L., Crammer, K., Schuurmans, D., 2006. Robust Support Vector Machine Training via Convex Outlier Ablation. Proc. National Conf. of Artificial Intelligence, p.1-7.