ISSN: 0067-2904
GIF: 0.851

Using One-Class SVM with Spam Classification

Inas Ali*, Sumaya Saad, Safa Ahmed
Department of Computer, College of Science, University of Baghdad, Baghdad, Iraq

Abstract

Support Vector Machine (SVM) is supervised machine learning technique which has become a popular technique for e-mail classifiers because its performance improves the accuracy of classification. The proposed method combines gain ratio (GR) which is feature selection method with one-class training SVM to increase the efficiency of the detection process and decrease the cost. The results show high accuracy up to 100% and less error rate with less number of feature to 5 features.

Keywords*:* gain ratio, spam, SVM.

استخدام SVM ذات الصنف الواحد لتصنيف البريد المؤذي

ايناس علي*، سمية سعد، صفا احمد
قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق

الخلاصة

SVM تقنية موجهه لتعليم الماكنة والتي اصبحت تقنية شائعة لمصنفات البريد الالكتروني بسبب ادائها الذي يحسن التنصنيف. الطريقة المقترحةتجمع بين نسبة الربح وهي طريقة اختيار الخصائص مع تدريب SVMذات الصنف الواحدلزيادة كفاءة عملية الكشف وتقليل الكلفة. اظهرت النتائج دقة عالية تصل الى 100% ونسبة خطأ اقل مع عدد خصائص يصل الى 5 خصائص.

1. Introduction

E-mail messages can be categorized into normal and spam which is useless electronic form of junk mail that is delivered by the postal services. Spam emails volume is growing widely every year and the traditional methods of spam recognition became slow speed and less accurate [1]. One of the conventional methods is using filters based on header content or sender address. The problem with filtering is that sometimes a legal message may be blocked [2]. The other methods use classification algorithms which provide a high precision in classifying.

SVM is a classification method that use hypotheses constructed in a multidimensional space, driven by an optimization algorithm derived from statistical learning theory [3]. SVM shows many particular advantages in solving nonlinear and high dimensional classification and has obtained a good result in pattern classification, function approaching and probability density [1]. SVM training is a computationally exhaustive process mostly due to its curved quadratic programming challenges related with the dense Hessian Matrix involved during optimization[4]. Many formulas and architectures for improving spam detection problem have been explored and suggested including combining more than one algorithm as in this work where feature selection process is combined with one-class training SVM.

This paper is organized as follows: Section 2 presents several works done previously. Section 3 describes the proposed method including the dataset, feature selection algorithm, and SVM algorithm. Section 4 presents experimental results of the used algorithm. A conclusion of this work is listed in section 5.

*Email: smart_girl8120@yahoo.com

2. Related Work

In Sculley paper [5], the anti-spam controversy is addressed and an expected accuracy is offered. First online SVMs show a state-of-the-art spam detection through experimental tests on several large benchmark data sets of email spam. Then analyse the effect of the trade-off parameter in the SVM objective function, which shows that the expensive SVM methodology may be overkill for spam detection. The computational cost of SVM learning is reduced by relaxing requirement on the maximum margin in online settings, and creates a Relaxed Online SVM suitable for high performance content-based spam filtering in large-scale settings. But not all data allow the relaxation of SVM requirements.

Lee et.al in [6] proposed parameter optimization and feature selection to reduce processing overheads with guaranteeing high detection rates. Parameters optimization is to regulate parameters of spam detection models in order to discover optimal parameters of the detection model. Feature selection is to select only important features. Feature selection enables excluding irrelevant features to avoid processing overheads.

In [7] a spam detection agent based on SVM was presented, several methods were tested to extract numerical features from text documents, and assess the optimal values of SVM parameters needed for this classification problem. While the best results show a good classification accuracy of 94% with large amount of training emails set.

An email classification task proposed in [1] uses the mutual information to extract key features from email while SVM is designed for classification. The simulation experiments of the SVM on emails shows average accuracy up to 98.9.

In [8], the dataset is divided by using J48 tree, then, features selection is applied in each partition. Consistently, selected features are used in SVM training. This method has evaluated some conducted benchmark datasets and the results are compared with other algorithms such as SVM and GA-SVM. The experimental results show that the suggested method is scalable when the number of features is increased with detection rate 91.04%.

3. Proposed Method

This section explains the details of GR and SVM algorithms with the data set used in the experiments (see Figure-1):

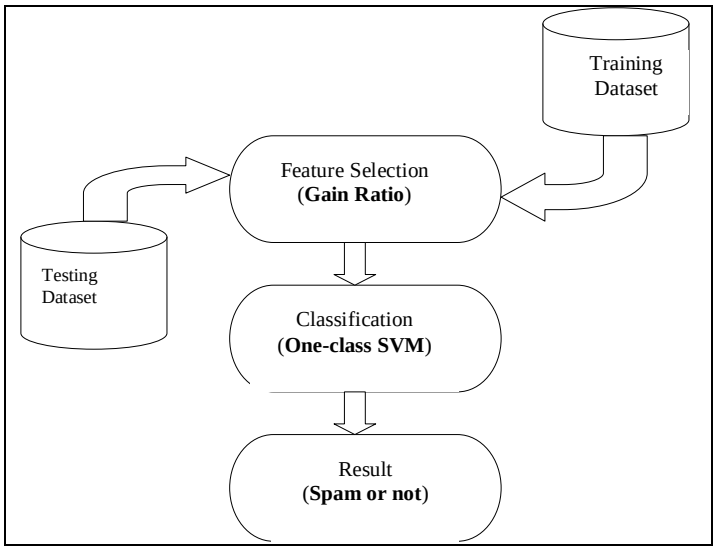

Figure 1-Stages of the proposed method

A. Data Used

The data used in the experiment is available in the spam-base dataset [9]. It contains 4601 records labelled normal and spam; each record contains fifty seven features in numeric form. These features represent relative frequencies of various most important words and characters in emails (for information about features see [9]).

B. Selection of Features

In feature selection process, some of original features is selected to use for training and testing the classifier. The selected features are most relevant to the dataset so many feature selection methods are proposed and GR is one of them. GR selects the features via scores [10]. If classes **c** is symbolized to {c_1, c_2, …, c_k}. **T** samples are partitioned into subsets T_1, T_2, …, T_n where T_i has all samples in **T** that have outcome o_i of the chosen test. Entropy is a criterion of medium uncertainty of collection of data. This represents the average much information which wanted to get from outcome of a data source. If **s** is a collection of samples, then freq (c_i, **s**) points to the set of samples in **s** that is in class c_i and also |**s**| denote the set of samples in the collection **s**. Equation 1 presents the entropy of the set **s**:

$$info(s) = -\sum ((freq(c_i,s) / |s|) * \log_2 (freq(c_i,s) / |s|)) \quad (1)$$

When collection T has been partitioned in accordance with n outcomes of one feature test **X**. Equation 2 presents the entropy of the set **T**:

$$info(T) = \sum ((|T_i| / |T|) * info(T_i)) \quad (2)$$

Gain information creates with the split, is the difference among the amount of information necessary to classify a situation after and before doing the split. Equation 3 presents the new gain rate:

$$Gain(X) = F * (info(T) - info_x(T)) \quad (3)$$

For a given features, total number of samples are divided to control missing values. If only the gain used it is not adequate to make a tree. The gain measure proper splits with many outcomes. Equation 4 defines GR as follow to solve this problem:

$$GainRatio(X) = Gain(X) / split\ info(X) \quad (4)$$

The GR partitions the gain with the evaluated split information. This produces splits with many outcomes as in equation 5.

$$split\ info(X) = -\sum ((S_i / |S|) * \log_2 (S_i / |S|)) \quad (5)$$

Split information is the weighted average of the information using the ratio of states that are sent to each child. The best evaluation function can measure the grace of a subset that produces from generation function and compared with the previous subset.

C. SVM system

The traditional two-class SVM has been exposed to yield state-of-the-art performance on email classification by detecting a hyperplane that separates two classes of data in data space while maximizing the distance among them as shown in Figure-2 [5].

Considers a data set Ω={(x1,y1),(x2,y2),…,(xn,yn)}; points xi∈R in a space where xi is the i-th input data point and yi∈{−1,1} is the i-th output pattern, denoting the class membership [2, 11, 12]. If the two classes are linearly separable, then an optimal weight vector w* can be found such that(Equation 6):

$$y_i(w^* . x_i - b) \geq 1 \quad (6)$$

Then margin among two classes is maximized when the norm of the weight vector ||w*|| is minimized which might be done by maximizing this function with respect to the Lagrange multipliers variables α_j as in Equation 7:

$$w(\alpha) = \sum \alpha_i - 0.5 \sum\sum \alpha_i \alpha_j (x_i . x_j)\, y_i y_j \quad (7)$$

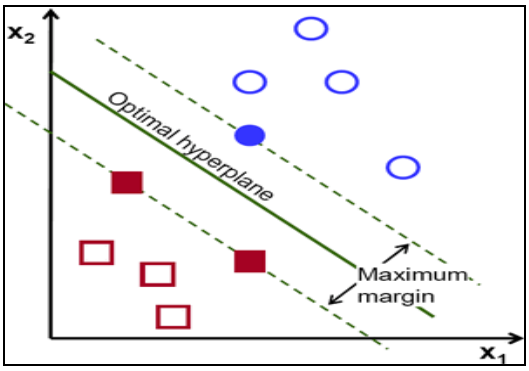

Figure 2- Margins amidst Two-class

Submissive to the constraint: $0 \leq \alpha_j$ where it is supposed there are N training examples, x_i is a of the training vectors, if $\alpha_j > 0$ then x_j is named a support vector. For an uncertain vector x_j classification corresponds to finding function F in Equation 8:

$$F(x_j) = sign\{w^* . x_j - b\} \quad (8)$$

where

$$w^* = \sum \alpha_i x_i y_i \quad (9)$$

And the sum is over the r nonzero support vectors (whose α 's are nonzero).

For the non-linear case a polynomial SVM can create a non-linear decision boundary by projecting the data through a non-linear function ϕ to a space with a higher dimension. This implies that data points which can't be separated by a straight line in their original space *I* are "lifted" to a feature space F where there can be a "straight" hyperplane that separates the data points of one class from an other. When that hyperplane would be projected back to the input space *I*, it would have the form of a non-linear curve. The hyperplane is represented with $w^Tx+b=0$, where $w \in F$ and $b \in R$. To prevent the SVM from over-fitting with noisy data, slack variables ξi are introduced to allow some data points to lie within the margin, If error is between $0 \leq \xi \leq 1$, then data can be properly classified, but if $\xi \geq 1$ then the data is misclassified. The constant C>0 spesifies the trade-off between maximizing the margin and the number of training data points within that margin (and thus training errors). The objective function of the SVM shown in the Equation 10 minimization formulation and its constraint as in Equation 11:

$$minimize \ ½ \ ||w||^2 + C \sum \zeta_i \quad (10)$$

Submissive to:

$$y_i(w^T \Phi(xi) + b) \geq 1 - \zeta_i \quad \text{for } \zeta i > 0 \quad (11)$$

The classification rule for a data point $\boldsymbol{x}$ is changed to Equation 12:

$$f(x) = \text{sgn}(\sum y_i K(x_i, x_j) + b) \quad (12)$$

$$\text{Where Polynomial function } K(x_i, x_j) = (\alpha x_i^T x_j + y)^d \quad y>0 \quad (13)$$

Here d and γ are kernel parameters.

Schölkopf et al. in 1999 proposed a method of acclimating the SVM methodology to the one-class classification problem by separating whole data points from the origin (in feature space *F*) and maximizing the distance from this hyperplane to the origin.

This conducts in a binary function which captures regions in the input space where the probability density of the data lives. Thus the function yields +1 in a "small" region (capturing the training data points) and −1 elsewhere. The quadratic programming minimization function is a bit different from the original stated above, but the sameness is still clear:

$$minimize \ ½ \ ||w||^2 + 1/v \sum \zeta_i - p \quad (14)$$

$$\text{subject to } (w \cdot \Phi(xi)) \geq p - \zeta_i \quad \zeta i > 0 \quad i=1,\ldots,n \quad (15)$$

In the previous formulation the parameter C determined the smoothness. In this formula it is the parameter ν that characterizes the solution; it sets an upper bound on the fraction of outliers and, it is a lower bound on the number of training examples used as Support Vector. The approach is referred to as ν-SVM [13].

Again by using Lagrange techniques and using a kernel function for the dot-product calculations, the decision function becomes:

$$f(x) = sgn(w \Phi(x_i) - p) = sgn(\alpha_i K(x_i, x_j) - p) \quad (16)$$

f(x) will be positive for most examples x_i contained in the training set.

4. Experimental Results

This section shows the results of the proposed GR and SVM combination when training linear and polynomial SVM algorithm on one-class (spam only) and two-class (normal and spam) of E-mails. The evaluation criterion is computed as:

$$\text{Accuracy} = \text{data classified correctly} \ / \ \text{total no. of data} \quad (17)$$

The dataset used in training and testing is 4601 (2788 normal and 1813 spam) distributed as shown in Table-1. Each data-entry contains 58 numerical value, from 1 to 57 are email features, while the 58 column is a label to denote if the e-mail was spam or not. The GR algorithm is applied to reduce the number of features into 5 or 10 or 15 features by selecting only the important features to the dataset Table-2.

Then SVM algorithm with linear and polynomial equations is used to train and test the dataset in two cases: the *first* case train the dataset to **normal** class only and the test is applied ones to **normal**

class, or **normal and spam**. While the *second* case trains the dataset to **normal and spam** classes and the test is applied to **normal and spam**. The outcomes shown in Table-3.

Table 1- Dataset groups used in experiment

	Data Group	Class used	training	testing
One-class	**Group 1**	**normal**	2100	688
		spam	non	non
	Group 2	**normal**	2100	688
		spam	non	1813
Two-class	**Group 3**	**normal**	2100	688
		spam	1502	311

Table 2- Relevant features

No. of features	Name of features
5	char_freq_;, char_freq_(, char_freq_#, word_freq_conference, word_freq_people
10	char_freq_;, char_freq_(, char_freq_#, word_freq_conference, word_freq_people, word_freq_hp, word_freq_1999, word_freq_415, word_freq_3d, word_freq_font
15	char_freq_;, char_freq_(, char_freq_#, word_freq_conference, word_freq_people, word_freq_hp, word_freq_1999, word_freq_415, word_freq_3d, word_freq_font, char_freq_$, word_freq_all, word_freq_your, word_freq_parts, word_freq_make

Table 3- Testing accuracy results (%)

SVM Type	No. of classes in test	No. of features / Equation	5	10	15
One-class	One class	linear	51.59	73.54	64.09
		polynomial	**100**	98.69	98.69
	Two classes	linear	73.05	53.01	86.56
		polynomial	82.88	78.68	79.12
Two-class	Two classes	linear	56.59	57.98	78.86
		polynomial	56.59	56.59	56.59

5. Conclusion

Spam Detection is an automated procedure that tools up security against Trojans, viruses , and resources having potentially unsafe information for particular group of users by detecting emails that contains spam. Many methods used to identify spam such as SVM. To enhance the productivity of SVM with one-class training, the feature was reduced using GR method. The results of the proposed system show a higher performance (up to 100% with minimum no. of features 5 features) than SVM with two-class training.

References

1. Shi, T. **2012**. Research on the Application of Email Classification based on Support Vector Machine. *Advances in intelligent and soft computing, springer-verlag*, 133, pp:987-994.
2. Drucker, H. Wu, D. and Vapnik, V. N. **1999**. Support Vector Machines for Spam Categorization. *IEEE Transactions on Neural Networks*, 10(5), pp: 1048-1054.
3. Vapnik, V. N. and Chervonenkis, A. **1974**. Theory of Pattern Recognition: Statistical Problems of Learning. Moscow, Nauka.
4. Godwin, C., Maozhen, L. and Yang L. **2013**. An Ontology Enhanced Parallel SVM for Scalable filter Training. *Elsevier Science Publishers*. Amsterdam, the Netherlands, 108, pp:45-57.

5. Sculley, D. and Wachman, G. M. **2007**. Relaxed Online SVMs for Spam Filtering, SIGIR'07, Amsterdam, The Netherlands, 23-27 July.

6. Lee, S. M. Kim, D. S. Kim, J. H. and Park, J. S. **2010**. Spam Detection Using Feature Selection and Parameters Optimization. CISIS '10 Proceedings of the 2010 International Conference on Complex, Intelligent and Software Intensive Systems, IEEE Computer Society Washington, DC, USA, pp:883-888.

7. Lazurca, C. and Leon, F. **2010**. An E-Mail Filtering Agent Based on Support Vector Machines. Buletinul Institutului Politehnic din Iaşi, pp:43-56.

8. Shahraki, M. Torabi, Z. S. and Nabiollahi, A. **2015**. Using J48 Tree Partitioning for Scalable SVM in Spam Detection. *Canadian Center of Science and Education*, 8(2), pp: 37-42.

9. UCI Machine Learning Repository, www.ics.uci.edu/~mlearn/MLRepository.html.

10. Zareapoor, M. and Seeja, K. R. **2015**. Feature Extraction or Feature Selection for Text Classification: A Case Study on Phishing Email Detection. *Mathematic Education and Computer Science press, I.J. Information Engineering and Electronic Business*, 2, pp: 60-65.

11. Vapnik, V. **1992.** *Estimation of Dependencies Based on Empirical Data*. Springer-Verlag, New York.

12. Vlasveld, R. **2013**. Introduction to One-class Support Vector Machines. M.Sc. Thesis. Department of Technical Artificial Intelligence, Utrecht University, The Netherlands.

13. Manevitz, L. M. and Yousef, M. **2001**. One-Class SVMs for Document Classification. *Journal of Machine Learning Research*, 2, pp: 139-154.