Support Vector Machines Parameter Selection Based on Combined Taguchi Method and Staelin Method for E-mail Spam Filtering

Wei-Chih Hsu [1], Tsan-Ying Yu [2,*]

[1] Department of Computer and Communication, National Kaohsiung First University of Science and Technology, Kaohsiung City, Taiwan, ROC

[2] Department of Electrical Engineering, Kao Yuan University, Kaohsiung, Taiwan, ROC.

Received 13 December 2011; received in revised form 20 January 2012; accepted 09 February 2012

Abstract

Support vector machines (SVM) are a powerful tool for building good spam filtering models. However, the performance of the model depends on parameter selection. Parameter selection of SVM will affect classification performance seriously during training process. In this study, we use combined Taguchi method and Staelin method to optimize the SVM-based E-mail Spam Filtering model and promote spam filtering accuracy. We compare it with other parameters optimization methods, such as grid search. Six real-world mail data sets are selected to demonstrate the effectiveness and feasibility of the method. The results show that our proposed methods can find the effective model with high classification accuracy

Keywords: Support Vector Machines, Taguchi Method, Grid Search

1. Introduction

Spamming is the abuse of electronic messaging systems to send unsolicited bulk e-mails or to promote services or products, which are usually undesired. Spamming is economically viable because advertisers have no operating costs beyond the management of their mailing lists. The sender cannot be specified, because the sender of spamming has only temporary e-mail address and the reply of them is not reached to the original sender. Therefore, undesired emails have been increased everyday so that it is hard to read an important e-mail.

Previously, the spam filtering black and white list was applied usually. Although its characteristics are fast and simple, but the drawback is that users have to update the spam mail filtering rules and maintain a black list. Spam filtering based on the textual content of e-mail messages can be regarded as a special case of text categorization, with the categories being spam and normal (non-spam). Content-based filters can be divided into rule-based methods and probabilistic methods. Rule-based methods such as Ripper [1, 2] , Boosting [3] , Decision Tree [4] , Rough Sets [5] and so on strongly dependent on the existence of key terms, therefore, specific terms can cause the failure of filtering. Methods based on probability and statistics such as K-Nearest Neighbor [5] and Support Vector Machine (SVM) [6] and so on. Besides, the prevailing machine learning method for spam message filtering is the Bayesian approach [7] used with good results.

* Corresponding author. E-mail address: allen@nfu.edu.tw
Tel.: +886-5-6315368; Fax: +886-5-6314486

SVM proposed by Vapnik [6] in 1995, has been widely applied in many applications such as function approximation, modeling, forecasting, optimization control, etc and has yielded excellent performance. It is a statistical theory to deal with the dual categories of classification and can find the best hyperplane to partition a sample space. Huang [8] demonstrated that the SVM-based model is very competitive to back-propagation neural network (BPN), genetic programming (GP) and decision tree in terms of classification accuracy. Selection of kernel function is a pivotal factor which determines performance of SVM. RBF kernel function requires only two parameters: the penalty factor C and the Gaussian kernel width γ. However, for the SVM-based model, its classification performance is sensitive to the parameters of the model, thus, parameters selection is very important. The function (C, γ) of the optimization parameters will make the SVM have the best performance. In spam filtering, the Bayesian algorithm in the mail system is very extensive. Compared with Bayesian algorithm, if SVM is used with linear kernel function or default parameters, the Bayesian algorithm will be better than the accuracy of SVM. In order to enhance the accuracy of SVM, it is necessary to develop a search mechanism to tune the hyperparameters. Most of the previous researches focus on the grid search (GS), pattern search based on principles from design of experiments (DOE) such as Staelin[9] and genetic algorithm (GA) [8, 9] to choose the parameters. GS is simple and easily implemented, but it is very time-consuming. DOE is like GS but it reduces the searching grid density and can reduce the computational time greatly. Although GA does not require setting an initial search range, it introduces some new parameters to control the GA searching process, such as the population size, generations, and mutation rate.

The Taguchi method [10], a robust design approach, uses many ideas from statistical experimental design for evaluating and implementing improvements in products, processes, and equipment. The fundamental principle is to improve the quality of a product by minimizing the effect of the causes of variation without eliminating the causes. One of the major tools used in the Taguchi method is orthogonal array (OA) to reduce the number of experiments and obtain good experimental results. The parameters (C, γ) of SVM are regarded as control factors in OA. Experiment is conducted through Multilevel-column OA after selecting the parameters of SVM. We verify the classification results and compared with GS. As far as we know, this is the first attempt to introduce Taguchi method to optimize the SVM for spam filtering models.

The remainder of this paper is organized as follows. In Section 2, the SVM and Taguchi method are described briefly. Section 3 presents implementation for our approach to classify the spam e-mails. Section 4 gives experimental results and discussion. Finally, the research results are summarized and also present future work.

2. Related Work

The proposed approach is based on SVM, Staelin Method and Taguchi method. In this section, SVM, Staelin Method and Taguchi method are introduced briefly.

2.1. The brief description of SVM

To search existing design models or study an available new design model with required specifications and to establish the topological structure of these models are the first step of the methodology. The goal of this step is to select some of these models for researching their equivalent mechanism skeleton and kinematic chain for developing the new designs.

The textual and non-textual features representing an email, obtained through the method mentioned previously, are as the input to the spam email filtering algorithm. In the approach, the filtering algorithm is represented by SVM.

SVM is a powerful supervised learning paradigm based on the structured risk minimization principle from statistical learning theory, which is currently placed among of the best-performing classifiers and have a unique ability to handle extremely large feature spaces (such as text), precisely the area where most of the traditional techniques fail due to the “curse of the dimensionality”. SVM has been reported remarkable performance on text categorization task. In our evaluation, we used the Library for SVM [11] to build SVM models. In the following, we give a brief introduction to the theory and implementation of SVM classification algorithm.

Consider the problem of separating the set of training set vectors belonging to two separate classes in some feature space. Given one set of training example vectors:

$$(x_1, y_1),...(x_l, y_l), x_i \in R_n, y_i \in \{-1,+1\} \tag{1}$$

we try to separate the vectors with a hyperplane

$$(w \cdot x) + b = 1 \tag{2}$$

so that

$$y_i[(w \cdot x) + b] \geq 1, (i = 1,2,...,l) \tag{3}$$

The hyperplane with the largest margin is known as the optimal separating hyperplane. It separates all vectors without error and the distance between the closest vectors to the hyperplane is maximal. The distance is given by

$$d(w,b) = \frac{2}{\|w\|} \tag{4}$$

Hence, the hyperplane that separates the data optimally is the one that minimizes the following equation:

$$Minimize \frac{1}{2}\|w\|^2 \tag{5}$$

subject to the constraints of (4).

To solve above problems, Lagrange multipliers α are introduced. Let *i* = 1,2,…,*l* and define

$$w(\alpha) = \sum_{i=1}^{l} \alpha_i y_i x_i \tag{6}$$

With Wolfe theory the problem can be transformed to its dual problem:

$$\max W(\alpha) = \sum_i \alpha_i - \frac{1}{2} w(\alpha) \cdot w(\alpha), s.t. \ \ \alpha_i \geq 0 \tag{7}$$

$$\sum_i \alpha_i y_i = 0 \tag{8}$$

With the optimal separating hyperplane found, the decision function can be written as:

$$f(x) = (w_0 \cdot x) + b_0 \tag{9}$$

Then the test data can be labeled with

$$label(x) = \operatorname{sgn}(f(x)) = \operatorname{sgn}((w_0 \cdot x) + b_0) \tag{10}$$

Training vectors that satisfy $y_i[(w_0 \cdot x) + b_0] = 1$ are termed support vectors, which are always corresponding to nonzero α_i. The region between the hyperplane through the support vectors on each side is called the margin band.

In the case of linearly non-separable training data, by introducing slack variables the primal problem can be rewritten as:

$$Min\left(\frac{1}{2}\|w\|^2 + C\sum_i \xi_i\right) \tag{11}$$

subject to . $y_i[(w \cdot x) + b] \geq 1 - \zeta_i, \zeta_i \geq 0$

Similarly, we can get the corresponding dual problem

$$\begin{aligned} &\max W(\alpha) = \sum_i \alpha_i - \frac{1}{2} w(\alpha) \cdot w(\alpha), \\ &s.t. \quad C \geq \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0 \end{aligned} \tag{12}$$

Problems described as in Eq. (11) and Eq. (12) are typical quadratic optimization questions, and have been approached using a variety of computational techniques. Recent advances in optimization methods have made support vector learning in large-scale training data possible.

All the training vectors corresponding to nonzero αi are called support vectors, which form the boundaries of the classes. The maximal margin classifier can be generalized to nonlinearly separable data via transforming input vectors into a higher dimensional feature space by a map function $K(x_i, x_j) = (\varphi(x_i), \varphi(x_j))$, followed by a linear separation there. The expensive computation of inner products can be reduced significantly by using a suitable kernel function . We implemented the SVM classifier using the LIBSVM library[12] and adopted radial basis function (RBF) defined as the kernel $K(x_i, x_j) = \exp\left(-\gamma\|x_i - x_j\|^2\right)$. In this study, the RBF is used as the basic kernel function of SVM. There are two parameters associated with the RBF kernels: C and γ. Vapnik found that a different kernel function of SVM has little effect on the performance but parameters of kernel function are key factor.

2.2. The brief description of Taguchi Method

The above original designs are transformed individually into their corresponding generalized chains (kinematic chains). The generalized chain will be involved in various types of members (edges) and joints (vertices, or said kinematic pairs) for all possible assembly in the following steps.

In this section, we briefly introduce the basic concept of the structure and Taguchi method. Taguchi method is quite common in the design of industrial experiments [12, 13]. Taguchi method requires a significantly small number of experiments compared with other statistical techniques [14]. Although some information is lost due to these two approximations, it is still worth choosing this approach according to the time consuming nature. OA is a very important tool for Taguchi method. Many designed experiments use matrices called OA for determining which combinations of factor levels to use for each experimental run and for analyzing the data. An OA is a fractional factorial matrix, which assures a balanced comparison of levels of any factor or interaction of factors. It is a matrix of numbers arranged in rows and columns where each row represents the level of the factors in each run, and each column represents a specific factor that can be changed in each run. The array is called orthogonal because all columns can be evaluated independently of one another.

The general symbol for m-level standard OA is

$$L_n(m^{n-1}) \tag{13}$$

where

$n=m^k$ number of experimental runs;

k a positive integer which is greater than 1;

m number of levels for each factor;

n-1 number of columns in the OA.

The letter “L” comes from “Latin,” the idea of using OA for experimental design having been associated with Latin square designs from the outset. The two-level standard OA which are most often used in practice are $L_4(2^3)$, $L_8(2^7)$, $L_{16}(2^{15})$, and $L_{32}(2^{31})$. Table 1 shows an OA $L_8(2^7)$. The number to the left of each row is called the run number or experiment number and runs from 1 to 8.

Table 1 $L_8(2^7)$ Orthogonal array

$L_8(2^7)$

Experiment No	cloumn						
	1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

2.3. Parameters selection using Staelin method

This parameters selection method for SVM-based model inspired by design of experiment (DOE) was proposed by Staelin [15]. This method called Staelin method which can reduce the complexity sharply relative to grid search method. Set initial range for X and Y as $[X_{\min}, X_{\max}]$, $[Y_{\min}, Y_{\max}]$, which is a combination of a standard N-parameter, three-level, or experiment design with another standard N-parameter, two-level, or experiment design, resulting in thirteen points per iteration in the two parameter case are as shown in Fig. 1. After each iteration, the system evaluates all sampled points and chooses the point with the best performance. The search space is centered at the best point, unless this would cause the search to go outside the user-given bounds, in which case the center is adjusted so the whole space is contained within the user-given bounds. At this point, the range is halved, and the best point is used as the center point of the new range, unless the new range would extend outside the user-specified range. The system repeats this process as many times as possible, and at the end, the best point is chosen.

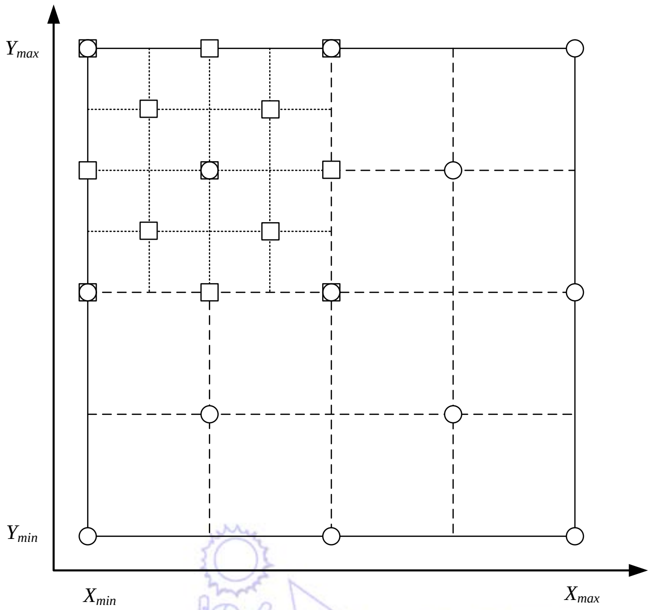


Fig. 1 Points of evaluation in two iteration search with Staelin method

3. Implementation

In this paper, the flow chart of e-mail spam filtering based on SVM with Taguchi method for parameter selection is shown in Fig. 2. First stage is data pre-processing as depicted in Fig. 3. Vector space model is a text representing approach, which is widely used and has good performance in text categorization. In its simple form, spam filtering can be recast as text categorization task where the classes to be predicted are spam and normal. Therefore, email can be regarded as a vector space, which is composed of a group of orthogonal key words.

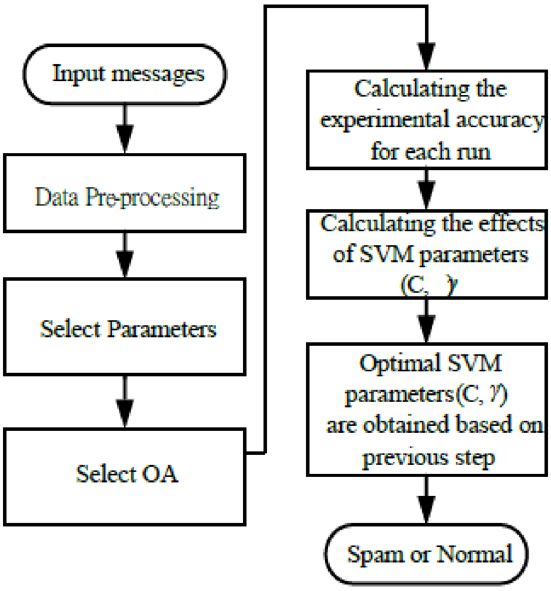

Input messages
Data Pre-processing
Select Parameters
Select OA
Calculating the experimental accuracy for each run
Calculating the effects of SVM parameters (C, γ)
Optimal SVM parameters(C, γ) are obtained based on previous step
Spam or Normal

Fig. 2 The flow chart of e-mail spam filtering based on SVM with Taguchi method for parameter selection

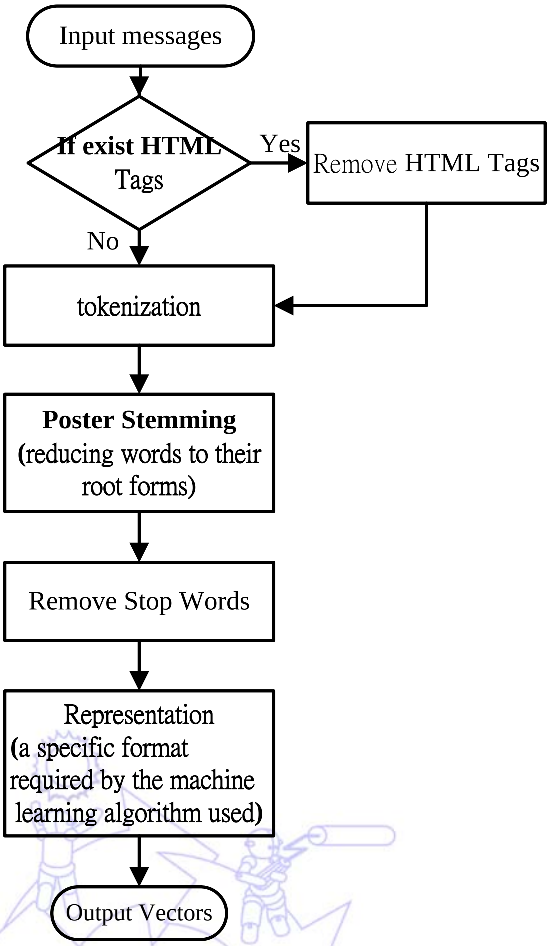


Fig. 3 Data Pre-processing

For each email, its textual portion was represented by a concatenation of the subject line and the body of the message. Due to the prevalence of html and binary attachments in modern email, a degree of pre-processing is required on messages to allow effective feature extraction. Therefore, we adopt the following data pre-processing steps:

(1) If there are HTML tags, then remove HTML tags. Then, tokenization is the process of reducing a message to its colloquial components.

(2) To avoid treating forms of the same word as different attributes, a lemmatizer was applied to the corpora to convert each word to its base form (e.g., "got" becomes "get").

(3) The stopping process is adopted to remove the high frequent words with low content discriminating power in an email document such as "to", "a","and","it", etc. Removing these words will save spaces for storing document contents and reducing time taken during the subsequent processes.

We obtain word frequencies and convert into vectors. We introduce Taguchi method to our approach. In content-based spam filtering performance analysis, a commonly used evaluation criteria measuring the efficiency of the classification is accuracy (Acc). It is regarded as response variable, defined as:

$$Acc = \frac{A + D}{N} \tag{14}$$

where N is the number of all messages; A is as spam and the actual system to determine the number of spam; and D is the actual system for normal mail and e-mail to determine the number of normal emails.

In order to reduce the number of times of experiments and the cost of design, we have to choose appropriate OA by numbers of control factors and levels. To explain how to employ OA to obtain the solution, on the other hand as the search scope is suggested by Lin [11] and we expand to different combinations of parameters C and γ with 8 levels: $log_2(C)$ = (-15, -11, -6, -2, 3, 7, 12, 16) and $log_2(\gamma)$ = (-15, -11, -6, -2, 3, 7, 12, 16) to find the best combination. In this work, both of the factor $log_2(C)$, $log_2(\gamma)$ are set at eight levels. Seven degrees of freedom (d.f.) are required for each factor. Consider that 14 d.f. are required in total, an OA type $L_{16}(2^{15})$ with 16 trials and 15 d.f. , as indicated in left side of Table 2, is adopted. A conversion of the L_{16} array of two levels to one multilevel with 8 levels had to be performed to accommodate two factors $log_2(C)$, $log_2(\gamma)$ with 8 levels. This modification of the OA should be planned in such a way that respects the d.f. of the L_{16}. In general, three main concepts were used in the orthogonal arrays theory [16].

Table 2 Experiment set-up and data for $L_{16}(8\times8\times2)$

Exp. No.	*Orginal Columns* *1, 2, 4*	 *13, 6, 1*	*Acc*					
	Modified columns *1*	 *2*						
	Factor $Log_2(C)$	 $Log_2(\gamma)$	Enronspam1	Enronspam2	Enronspam3	Enronspam4	Enronspam5	Enronspam6
1	-20.00	-20.00	98.10	80.00	80.00	80.00	97.00	80.00
2	-20.00	2.86	98.10	80.00	80.00	80.00	97.00	80.00
3	-14.29	-8.57	98.10	80.00	80.00	80.00	97.00	80.00
4	-14.29	14.29	98.10	80.00	80.00	80.00	97.00	80.00
5	-8.57	-8.57	98.10	80.00	80.00	80.00	97.00	80.00
6	-8.57	14.29	98.10	80.00	80.00	80.00	97.00	80.00
7	-2.86	-20.00	98.10	80.00	80.00	80.00	97.00	80.00
8	-2.86	2.86	98.10	80.00	80.00	80.00	97.00	80.00
9	2.86	-14.29	98.80	85.90	87.80	84.00	97.30	86.40
10	2.86	8.57	98.10	80.20	80.00	80.00	97.00	80.40
11	8.57	-2.86	98.80	83.60	80.50	79.90	97.00	81.20
12	8.57	20.00	98.10	80.40	80.00	80.00	97.00	80.40
13	14.29	-2.86	98.80	83.40	80.30	80.10	97.00	81.10
14	14.29	20.00	98.10	80.40	80.00	80.00	97.00	80.40
15	20.00	-14.29	99.40	96.00	95.60	94.10	97.50	94.90
16	20.00	8.57	98.10	80.20	80.00	80.00	97.00	80.20

1. Balance, for each factor the levels occur equally often.

2. Estimability, every parameter could be capable of being estimated.

3. Orthogonality, a term which implies that it is easy to extract and separate out the effect of different factors equally.

Multilevel factors could be created by the appropriate multilevel columns in two-level arrays. This is generally achieved at expense of 3 columns which are replaced by a new column whose levels directly correspond to every level-combination of the original 2 columns. The only requirement for the creation of multilevel columns in this way is that four interaction columns must exist for the 3 sacrificed columns which are deleted. Consequently, only one two-level column is left to remain after conversion and $L_{16}(8\times8\times2)$ are achieved.

In order to verify whether the arithmetic is valid or not, we employ 5-fold cross validation for our experiment. 5-fold cross validation is to separate e-mails into 5 parts. We make use of the 4 parts for training, and the remaining for testing. The procedure loops 5 times, so every part has been tested. Finally, the average of tests values is used as the result of test for evaluation. Each run of L_{16} $(8\times8\times2)$ will proceed 5-fold cross validation. The accuracy for each run and the average accuracy for each level and each factor need to be evaluated. We pick the level which is with maximum accuracy for each factor. Therefore, we can obtain approximation results.

4. Experiment results and discussion

In our test, the program runs with LIBSVM toolbox provide by Lin [11] on an IBM compatible PC. Six public data sets have been used in this study. The experiments were conducted on enronspam corpora [17]. The enronspam corpus, which contains six non-encoded data sets : enronspam1, enronspam2, enronspam3, enronspam4, enronspam5 and enronspam6, respectively. Each contains ham messages of particular users and fresh spam messages and includes spam messages from various sources. Attachments, HTML tags, and duplicate spam messages received on the same day are not included. We mix this enronspam and take 500 normal messages and 500 spam messages randomly. Table 3 shows the summary of the data sets.

Table 3 Description of data sets

	Orginal		Our method	
Data set	Non-Spam	Spam	Non-Spam	Spam
enronspam1	3672	1500	500	500
enronspam2	4316	1496	500	500
enronspam3	4012	1500	500	500
enronspam4	1500	4500	500	500
enronspam5	1500	3675	500	500
enronspam6	1500	4500	500	500

Experiment set-up and data for $L_{16}(8\times8\times2)$ is shown in Table 2. In this table, the conversion of $L_{16}(8\times8\times2)$ from $L_{16}(2^{15})$ still keep orthogonal. It indicates that the accuracy of SVM will become worse without careful selection for parameters C and γ. We list accuracy averages of both parameters $log_2(C)$ and $log_2(\gamma)$ for every level in different data sets and evaluate effective of control factors for all levels as illustrated in Table 4. Here accuracy is desirable as larger as in possible.

The maximum of both parameters $log_2(C)$ and $log_2(\gamma)$ accuracy average for each level each data set are marked. The difference between maximum accuracy and minimum accuracy of main effect for parameters $log_2(C)$ and $log_2(\gamma)$ implies the impact for accuracy. By observing the effective and variance of control factor $log_2(\gamma)$ and $log_2(C)$ for all level, The difference of parameters $log_2(\gamma)$ is larger than the one of parameters $log_2(\gamma)$. It means that parameter γ is more significant than parameter C for all data sets. The experiment of both methods used identical training and testing sets with 5-fold cross validation.

The average classification accuracy of 5-fold cross validation of both methods for in other data sets, the average accuracy for Taguchi method is close to the results for GS but not good enough. Furthermore, we apply Taguchi method with more levels OAs as depicted in Table 5(b)(c). This improvement is significant between Table 5(a) and (b). However, the improvement is little between Table 5(b) and (c). Taguchi approach with more the number of levels has more effective detective points, so the accuracy will get higher. Meanwhile, the difference in accuracy between GS and our proposed method will decrease. The comparison of both methods is based on the same levels in this experiment. Fig. 4 is available by GS on C = $(2^{-15}, 2^{-14}, 2^{-13}, 2^{-12}, ..., 2^{15}, 2^{-16})$ and $\gamma = (2^{-15}, 2^{-14}, 2^{-13}, 2^{-12}, ..., 2^{15}, 2^{-16})$ for each data set. Higher accuracies concentrated in the lower right corner of the contour graph. These contributions are similar to all data sets.

Compared with GS, Naïve Bayes, SVM(Linear), SVM(Taguchi Method L_{32}) and our algorithm SVM (L_{64}(32X32X2)) for enronspam corpa, the results of our confirm test are shown in Table 6. The SVM(Linear) shows that the accuracy of SVM will become worse without careful selection for parameters C and γ.

No parameter needs to be set up for SVM with linear kernel; the accuracy will lower than that of Naïve Bayes algorithms and our proposed method. Although our method is not the best accuracy of the proposed method, it is little lower than that of GS(32×32). GS required searching and computing 32×32 = 1024 times but our proposed method need only 64 times. Our approach is 15 times faster and the accuracy of our method is very close to that of GS. The experimental results show that our proposed method can select good parameters for SVM with kernel RBF and the accuracy is very close to that of GS.

Table 4 $L_{16}(8\times8\times2)$ OA and experiment data

Factor	enronspam1		enronspam2		enronspam3		enronspam4		enronspam5		enronspam6	
	$\text{Log}_2(C)$	$\text{Log}_2(\gamma)$	$\text{Log}_2(C)$	$\text{Log}_2(\gamma)$	$\text{Log}_2(C)$	$\text{Log}_2(\gamma)$	$\text{Log}_2(C)$	$\text{Log}_2(\gamma)$	$\text{Log}_2(C)$	$\text{Log}_2(\gamma)$	$\text{Log}_2(C)$	$\text{Log}_2(\gamma)$
-20.00	98.10	98.10	80.00	80.00	80.00	80.00	80.00	80.00	97.00	97.00	80.00	80.00
-14.29	98.10	$\boxed{99.00}$	80.00	$\boxed{90.75}$	80.00	$\boxed{92.35}$	80.00	$\boxed{88.50}$	97.00	$\boxed{97.45}$	80.00	$\boxed{90.75}$
-8.57	98.10	98.10	80.00	80.00	80.00	80.00	80.00	80.00	97.00	97.00	80.00	80.00
-2.86	98.10	98.80	80.00	83.40	80.00	80.30	80.00	80.05	97.00	97.00	80.00	81.20
2.86	98.45	98.10	82.95	80.00	83.85	80.00	81.95	80.00	97.15	97.00	83.35	80.00
8.57	98.45	98.10	81.90	80.40	80.25	80.00	80.00	80.00	97.00	97.00	80.80	80.30
14.29	98.45	98.10	81.80	80.00	80.05	80.00	80.05	80.00	97.00	97.00	80.80	80.00
20.00	$\boxed{98.65}$	98.10	$\boxed{88.20}$	80.30	$\boxed{88.50}$	80.00	$\boxed{86.55}$	80.00	$\boxed{97.30}$	97.00	$\boxed{87.70}$	80.40
Max	98.65	99.00	88.20	90.75	88.50	92.35	86.55	88.50	97.30	97.45	87.70	90.75
Min	98.10	98.10	80.00	80.00	80.00	80.00	80.00	80.00	97.00	97.00	80.00	80.00
Effect	0.55	0.90	8.20	10.75	8.50	12.35	6.55	8.50	0.30	0.45	7.70	10.75

Table 5 Results for different OAs

Data set	GS(8×8)			$L_{16}(8\times8\times2)$			Acc. Diff.(%)
	Log(C)	Log(γ)	Acc(%)	Log(C)	Log(γ)	Acc(%)	
enronspam1	20.00	-8.57	99.60	20.00	-14.29	99.40	0.20
enronspam2	14.29	-20.00	97.40	20.00	-14.29	96.00	1.40
enronspam3	8.57	-14.29	97.30	20.00	-14.29	95.60	1.70
enronspam4	8.57	-14.29	95.80	20.00	-14.29	94.10	1.70
enronspam5	8.57	-8.57	97.90	20.00	-14.29	97.50	0.40
enronspam6	8.57	-8.57	95.50	20.00	-14.29	94.90	0.60
Avg.							**1.00**

(a)

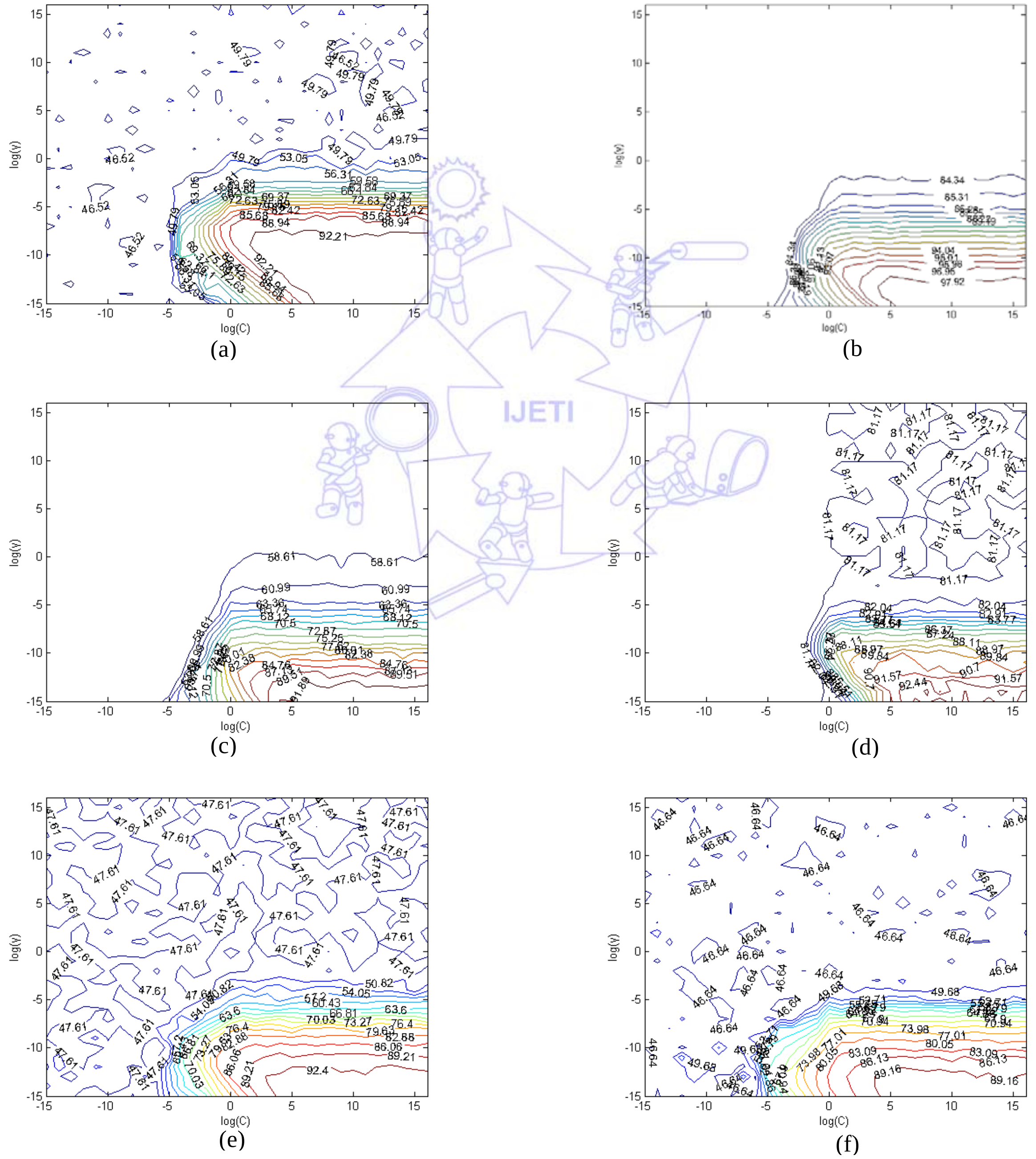
Data set	GS(16×16)			$L_{32}(16\times16\times2)$			Acc. Diff.(%)
	Log(C)	Log(γ)	Acc(%)	Log(C)	Log(γ)	Acc(%)	
enronspam1	4.00	-6.67	99.6	4.00	-6.67	99.60	0.00
enronspam2	12.00	-17.33	97.30	12.00	-12.00	96.50	0.80
enronspam3	9.33	-12.00	97.10	17.33	-12.00	96.30	0.80
enronspam4	14.67	-17.33	96.40	12.00	-12.00	95.50	0.90
enronspam5	20.00	-12.00	98.10	17.33	-12.00	98.00	0.10
enronspam6	20.00	-17.33	96.30	17.33	-12.00	94.70	1.60
Avg.							**0.70**

(b)

Table 5 Results for different OAs (Continued)

Data set	GS(32×32)			L_{64}(32×32×2)			Acc. Diff.(%)
	Log(C)	Log(γ)	Acc(%)	Log(C)	Log(γ)	Acc(%)	
enronspam1	18.71	-5.81	99.6	1.94	-8.39	99.60	0.00
enronspam2	12.26	-14.84	97.7	9.68	-16.13	96.40	1.30
enronspam3	16.13	-16.13	97.30	10.97	-16.13	96.90	0.40
enronspam4	17.42	-18.71	96.5	10.97	-16.13	95.80	0.70
enronspam5	7.10	-9.68	98.20	16.13	-10.97	98.20	0.00
enronspam6	17.42	-18.71	96.20	9.68	-16.13	93.80	2.40
Avg.							**0.80**

(c)



(a)

(b

(c)

(d)

(e)

(f)

Fig. 4 The contour plots of GS on C = $(2^{-15}, 2^{-14}, 2^{-13}, 2^{-12}, ..., 2^{15}, 2^{-16})$ and γ = $(2^{-1}5, 2^{-14}, 2^{-13}, 2^{-12}, ..., 2^{15}, 2^{-16})$ for (a) Enronspam1 (b) Enronspam2 (c) Enronspam3 (d) Enronspam4 (e) Enronspam5 (f) Enronspam6

Table 6 A comparison of the accuracy(%) of different models for enronspam corpora

	eronspam1	eronspam2	eronspam3	eronspam4	eronspam5	eronspam6
Naïve Bayesian	86.99	92.93	91.53	92.19	86.57	80.71
SVM(Linear)	80.23	83.41	85.23	85.69	87.77	88.98
SVM(Taguchi Method L_{32})	94.56	94.39	95.5	93.30	97.10	95.78
SVM (GS 32X32)	99.60	97.70	97.3	96.50	98.20	96.20
SVM (L_{64}(32X32X2))	99.60	96.40	96.90	95.80	98.20	93.80

5. Conclusions and future work

Our proposed approach based on Taguchi method does not like other approximation methods or heuristics may cause exhaustive parameter searches. On the other hand, our proposed approach sometimes may obtain approximation results but not optimal. However, compared with much computational time to find the optimal parameter values by the grid-search, it is worth for our methods to obtain approximation results at expense of little accuracy.

From above experiments, appropriate OA could achieve high accuracy but high multilevel OA make little improvement. In order to achieve appropriate multilevel-column OA, we convert from 2-level OA and still keep multilevel-column OA orthogonal. In our method, the parameter selection by orthogonal table will obtain high accuracy. If we would like to obtain higher accuracy, we could extend OA L_{64} to an OA such as L_{128} to promote the accuracy.

References

[1] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 115-123.

[2] W. W. Cohen, "Learning rules that classify e-mail," in *Proceedings of the 1996 AAAI Spring Symposium in Information Access*, 1996, pp. 18-25.

[3] I. Androutsopoulos, G. Paliouras, and E. Michelakis, "Learning to filter unsolicited commercial e-mail," " DEMOKRITOS", National Center for Scientific Research Technical report 2004/2, 2004.

[4] M. Collins, R. E. Schapire, Y. Singer, P. Domingos, W. Fan, S. J. Stolfo, J. Zhang, P. K. Chan, Y. Freund, and R. Schapire, "Boosting Trees for Anti-Spam Email Filtering," *4th International Conference on Recent Advances in Natural Language Processing*, 2001, pp. 1189-1232.

[5] I. Androutsopoulos, G. Paliouras, V. Karkaletsis, G. Sakkis, C. D. Spyropoulos, and P. Stamatopoulos, "Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach," presented at the Proceedings of the workshop "Machine Learning and Textual Information Access", 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, 2000.

[6] V. N. Vapnik, *The nature of statistical learning theory*. New York: Springer Verlag, 2000.

[7] J. Provost, "Naive-bayes vs. rule-learning in classification of email. The University of Texas at Austin," Artificial Intelligence Lab. Technical Report AI-TR-99-284, 1999.

[8] C. L. Huang and C. J. Wang, "A GA-based feature selection and parameters optimizationfor support vector machines," *Expert Systems With Applications*, vol. 31, pp. 231-240, 2006.

[9] T. Howley and M. G. Madden, "The genetic kernel support vector machine: Description and evaluation," *Artificial Intelligence Review*, vol. 24, pp. 379-395, 2005.

[10] G. Taguchi and S. Chowdhury, *Robust engineering*, New Work: McGraw-Hill, 2000.

[11] C. C. Chang and C. J. Lin. (2008). *LIBSVM -- A Library for Support Vector Machines*. http://www.csie.ntu.edu.tw/~cjlin/libsvm/

[12] G. Taguchi, *Introduction to quality engineering*, Tokyo: Asian Productivity Organization, 1990.

[13] M. Phadke, *Quality engineering using robust design*, U.S.A: Prentice Hall PTR Upper Saddle River, 1995.

[14] D. C. Montgomery, *Design and analysis of experiments*, New York: Wiley, 2006.

[15] C. Staelin, "Parameter selection for support vector machines," *Hewlett-Packard Company, Tech. Rep. HPL-2002-354R1*, 2003.

[16] N. Logothetis and H. P. Wynn, *Quality through design: experimental design, off-line quality control, and Taguchi's contributions*, Oxford: Clarendon Press, 1989.

[17] *Enronspam*. http://www.aueb.gr/Users/ion/data/enron-spam/