International Journal of Operations Research

International Journal of Operations Research Vol. 6, No. 1, 24–32 (2009)

On Multiclass Support Vector Machines: One-Against-Half Approach

JYING-NAN WANG[1,*], SY-MING GUU[2], and SHENG-TE CHOU[3]

[1] Department of Finance Minghsin University of Science and Technology No.1, Xinxing Rd., Xinfeng 30401, Hsinchu, Taiwan, R.O.C.

[2] Department of Business Administration Yuan Ze University 135, Far East Road, Taoyuan, Taiwan, R.O.C

[3] Department of Accounting Chinese Culture University 55, Hwa-Kang Road, Yang-Ming-Shan, Taipei, Taiwan, R. O. C.

Received Auguest 2009; Accepted September 2009

Abstract—Support vector machines (SVM) was originally designed for binary classification. SVM has been recently applied to solve multi-class problems. And there lies the unsolving research issues on developing 2-class SVM into multi-class SVM. In this paper, five common multi-class SVMs have been reviewed and a new multi-class SVM "one-against-half method" has been proposed along with the comparison between the performance of one-against-half method and the other five multi-class SVMs. The experiments proved one- against-half method to be a qualified multi-class SVM.

Keyword—One-against-half method, Support vector machines, Multi-classification.

1. INTRODUCTION

The Support Vector Machines (SVM), based on Statistical Learning Theory, is a new technique for solving a variety of learning and function estimation problems. SVM has fairly extensive applications, such as image recognition, text categorization, hand-written digit recognition, data mining, and bioinformatics. SVM was originally designed for binary classification. It has been recently applied to solve multi-class problems. And there lies the unsolving research issues on developing 2-class SVM into multi-class SVM.

Currently there are two types of approaches for multi-class SVM. One constructs a multi-class SVM by combining several 2-class SVMs, including "one-against-all" (Bottou, et al., 1994), "one-against-one" (Friedman [1996] and KreBel [1999]), and "DAGSVM" (Platt, et al., 2000). The other one considers all classes at once, including "considering all data at once" (Vapnik, 1998) and "C&S method" (Crammer and Singer, 2000). The experiments of Chang (2000) indicate that none of the above-mentioned methods is entirely better than the others. So how to effectively extend 2-class SVM for multi-class SVM is still an on-going research issue. For solving *k*-class problems, "one-against-one" and "one-against-all" need to deal $k\times(k-1)/2$ and k 2-class SVMs respectively. Under different multi-class problems, the amount of 2-class SVMs may produce different performance and computational time. Thus, we argue that a more elastic method which constructs $n\times k$ 2-class SVMs may yield a better model under different multi-class problems. In this paper, we propose a new idea of multi-class SVM: one-against-half method, which is a new method of solving multi-class problem. Although this method constructs exactly $2k$ 2-class SVMs, people can extend it to $n\times k$ case easily.

In Section2, we first review 2-class SVM and multi-class SVMs. In the next Section, we give an introduction of one-against-half method. Numerical experiments are in Section 4 where we compare one-against-half with the other multi-class SVMs, and we also propose the improvement approach for one-against-half method when the accuracy rate is bad. Finally we give the conclusion and future works in Section 5.

2. SUPPORT VECTOR MACHINE

2.1 2-Class SVM

We will start with the separable case (Burges [1998], Cristianini and Shawf-Taylor [2000], Schölkopf, et al. [1999], and Vapnik [1998]). Label the training data $\{x_i, y_i\}$, $i=1,...,l$, $y_i\in\{-1,1\}$, $x_i\in R^d$. Suppose there are some hyperplane that separates the positive from the negative examples. The points x which lie on the separating hyperplane satisfy $w\cdot x+b=0$, where w is normal to the hyperplane. Define the "margin" of a separating hyperplane to be $d_+ + d_-$, where $d_+\ (d_-)$ is the

* Corresponding author's e-mail: s929603@mail.yzu.edu.tw

shortest distance from the separating hyperplane to the closest positive (negative) training data. In the separable case, all the training data satisfy the following constraints:

$$x_i \cdot w + b \geq +1 \quad \text{for } y_i = +1 \tag{1}$$

$$x_i \cdot w + b \leq -1 \quad \text{for } y_i = -1 \tag{2}$$

These can be combined into one set of inequalities:

$$y_i \left(x_i \cdot w + b \right) - 1 \geq 0 \quad \forall i \tag{3}$$

By constraints Eq.(1) and Eq.(2), $d_+ = d_- = 1/\|w\|$ and the margin is simply $2/\|w\|$. Thus we can find the separating hyperplane which gives the maximum margin by minimizing $\|w\|^2$, subject to constraints Eq.(3). Using the Lagrange multiplier technique, a *positive* Lagrange multipliers α_i, $i = 1,...,l$, one for each of the inequality constraints Eq.(3) is introduced. This gives Lagrangian:

$$\min \; L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i y_i \left(x_i \cdot w + b \right) + \sum_{i=1}^{l} \alpha_i \tag{4}$$

$$\alpha_i \geq 0$$

In order to deal properly with nonlinear SVM, we transform L_P into its dual problem:

$$\max \; L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

$$\alpha_i \geq 0 \tag{5}$$

$$\sum_i \alpha_i y_i = 0$$

In the case where the training data cannot be separated by a hyperplane without errors, Cortes and Vapnik (1995) propose that introducing positive slack variables $\xi_i, i = 1,...,l$, the constraints become:

$$x_i \cdot w + b \geq +1 - \xi_i \quad \text{for } y_i = +1 \tag{6}$$

$$x_i \cdot w + b \leq -1 + \xi_i \quad \text{for } y_i = -1 \tag{7}$$

$$\xi_i \geq 0 \tag{8}$$

The goal is to construct hyperplane that makes the smallest number of errors. Hence the objection function becomes minimize $\|w\|^2 / 2 + C\left(\sum_i \xi_i\right)$, where C is a parameter to be chosen by the user, a larger C corresponding to assigning a higher penalty to errors. The optimization problem becomes:

$$\max \; L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

$$0 \leq \alpha_i \leq C \tag{9}$$

$$\sum_i \alpha_i y_i = 0$$

Now suppose that the data is mapped to some higher dimension space (feature space), using a mapping which is called Φ (Boser, 1992):

$$\Phi : R^d \rightarrow F \tag{10}$$

Then of course the training algorithm would only depend on the data through dot products in F, i.e. on functions of the form $\Phi\left(x_i\right) \cdot \Phi\left(x_j\right)$. Kernel function is the important concept of SVM, The definition of kernel is:

$$k\left(x_i, x_j\right) = \left(\Phi\left(x_i\right) \cdot \Phi\left(x_j\right)\right) \tag{11}$$

So the optimization problem of nonlinear SVM is:

$$\max \; L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k\left(x_i, x_j\right)$$

$$0 \leq \alpha_i \leq C \tag{12}$$

$$\sum_i \alpha_i y_i = 0$$

After solving this optimization problem, those points for which $\alpha_i > 0$ are called "support vectors". Then they determine w by Eq.(13). And b can be found by KKT (Fletcher, 1987) "complementarily" condition Eq.(14), where s_j are support vectors and Ns is the number of support vectors.

$$w = \sum_j^{Ns} \alpha_j y_j \Phi\left(s_j\right) \tag{13}$$

$$\alpha_i \left(y_j \left(w \cdot \Phi\left(s_j\right) + b \right) - 1 \right) = 0 \tag{14}$$

Finally, the class of x is

$$\operatorname{sgn}\left(w \cdot x + b\right) = \operatorname{sgn}\left(\sum_{j}^{Ns} \alpha_j y_j k\left(s_j, x\right) + b \right) \tag{15}$$

2.2 Multi-class SVM

One-against-all method (Bottou, et al., 1994) is probably the earliest multi-class SVM. It constructs k SVM models where k is the number of classes. The i th SVM is trained with all of training data in the i th class with positive labels, and all other data with negative labels. Given l training data $\left(x_1, y_1\right),...,\left(x_l, y_l\right)$, where $x_j \in R^n$, $j = 1,...,l$ and $y_j \in \left\{1,...,k\right\}$ is the class of x_i. Solving the following problem get the i th SVM:

$$\begin{aligned}
&\min \ \frac{1}{2}\left(w^i\right)^T w^i + C\sum_{j=1}^{l} \xi_j^i \\
&\left(w^i\right)^T \Phi\left(x_j\right) + b^j \geq 1 - \xi_j^i, \ \text{if } y_j = i \\
&\left(w^i\right)^T \Phi\left(x_j\right) + b^j \leq -1 + \xi_j^i, \ \text{if } y_j \neq i \\
&\xi_j^i \geq 0, j = 1,...,l.
\end{aligned} \tag{16}$$

The unknown-class data x is in the class which has the largest value of the decision function:

$$\textit{class of } x \equiv \arg\max_{i=1,...,k} \left(\left(w^i\right)^T \Phi\left(x\right) + b^i \right) \tag{17}$$

Another method is called one-against-one method (Friedman [1996] and KreBel [1999]). It constructs $k\left(k-1\right)/2$ SVM models where each one is trained on data from two classes. One-against-one method uses voting strategy (Friedman, 1996) to decide which class of x: if decision function says x is in the i th class, then the vote for the i th class is added by one. Then we predict x is in the class with the largest vote.

The third method is the Directed Acyclic Graph Support Vector Machines (DAGSVM) (Platt, et al., 2000) which extends from one-against-one method. Its training phase also constructs $k\left(k-1\right)/2$ SVM models, but DAGSVM use a directed acyclic graph to predict the class of x in the testing phase. Its testing time is less than the one-against-one method.

The other two multi-class SVMs , "considering all data at once method" (Vapnik, 1998) and "C&S method" (Crammer and Singer, 2000), directly consider all data in one optimization formulation. The difference between these methods is that C&S method deceases the number of variables. These two multi-class SVMs are dissimilar to our new multi-class SVM; so we omit the introduction of them.

3. ONE-AGAINST-HALF METHOD

"One-against-half method" constructs $2k$ SVM models where k is the number of classes. Each class constructs two SVM models (ex. $SVM^{i,1}$ and $SVM^{i,2}$). $SVM^{i,1}$ is trained on data from class i and the front half of all classes except class i; $SVM^{i,2}$ is trained on data from class i and the later half of all classes except class i. Both they are in the ith class with positive labels, and the others with negative labels. Thus given l training date $\left(x_1, y_1\right)$, $\left(x_2, y_2\right),...,\left(x_l, y_l\right)$, where $x_j \in R^n$, $j = 1,...,l$ and $y_j \in \left\{1,...,k\right\}$ is the class of x_i, the $SVM^{i,1}$ solves the following problem:

$$\begin{aligned}
&\min \ \frac{1}{2}\left(w^{i,1}\right)^T w^{i,1} + C\sum_{t} \xi_t^{i,1} \\
&y_t \left(\left(w^{i,1}\right)^T \Phi\left(x_t\right) + b^{i,1} \right) \geq 1 - \xi_t^{i,1} \\
&\xi_t^{i,1} \geq 0, i = 1,2,...k
\end{aligned} \tag{18}$$

The approach of constructing other SVM models is similar to $SVM^{i,1}$. After solving $2k$ optimization problems Eq.(18), we will have $2k$ decision functions. Adding every two "decision values", then we can get k "sum of decision values", $D^1,...,D^k$:

$$\begin{aligned}
&\left(w^{1,1}\right)^T \phi\left(x\right) + b^{1,1} \ + \ \left(w^{1,2}\right)^T \phi\left(x\right) + b^{1,2} \ = D^1 \\
&\qquad \vdots \qquad\qquad\qquad \vdots \qquad\qquad \vdots \\
&\left(w^{k,1}\right)^T \phi\left(x\right) + b^{k,1} \ + \ \left(w^{k,2}\right)^T \phi\left(x\right) + b^{k,2} = D^k
\end{aligned}$$

We say x is in the class which has the largest value of the "sum of decision values":

$$\textit{class of } x \equiv \arg\max_{i=1,...,k} \left(D^i \right) \tag{19}$$

3.1 Example

We use "one-against-half method" to deal with the multi-class dataset: *satimage* (Blake and Merz, 1998), which contains 36 attributes. Note that for this problem, there is one missing class. That is, in the original application there is one more class but in the data set no examples are with this class. Thus the goal is to classify the class of *satimage* based on these 36 attributes. The statistic of this problem is displayed in Table 1.

Table 1. Multi-class problem: *satimage*

Description	Training Data	Test Data
1 red soil	1072(24.17%)	461 (23.05%)
2 cotton crop	479 (10.80%)	224 (11.20%)
3 grey soil	961 (21.67%)	397 (19.85%)
4 damp grey soil	415 (09.36%)	211 (10.55%)
5 soil with vegetation stubble	470 (10.60%)	237 (11.85%)
6 mixture class	0	0
7 very damp grey soil	1038(23.40%)	470 (23.50%)

First, we must group the training data in order to construct 12 (2×6) SVM models. Table 2 presents the result after dividing into 12 groups. Next we decide kernel function: *rbf* kernel and parameters: $\gamma = 0.001$, $C = 100$, where rbf kernel is:

$$k\left(x_i, x_j\right) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \tag{20}$$

Then we solve 12 SVM models by Eq.(18). For example, $SVM^{3,1}$ is trained on data from three classes in the 3rd class with positive labels, and the others with negative labels. After constructing 12 SVM models, we start testing phase. If randomly select one example x and test it by these models, we will get 12 decision values. Adding every two "decision values", then we can get 6 "sum of decision values". The result is presented in Table 3. We easily find that the maximum sum of decision values is 2.9097. So we say x is 1st class through Eq.(19). After testing 2000 testing data in the same models, 1844 data of all will be predicted correct. So the accuracy rate is 92.2%. But it is unfair to use only one parameter set. Practically for any method people find the best parameters by performing the model selection. Then the best parameter set is used for constructing the model for future testing. Note that details of how we conduct the model selection will be discussed in section 4.

Table 2. The result of grouping

Class	Label (+)	Label (-)	Models
1	1	2,3	$SVM^{1,1}$
	1	4,5,7	$SVM^{1,2}$
2	2	1,3	$SVM^{2,1}$
	2	4,5,7	$SVM^{2,2}$
3	3	1,2	$SVM^{3,1}$
	3	4,5,7	$SVM^{3,2}$
4	4	1,2	$SVM^{4,1}$
	4	3,5,7	$SVM^{4,2}$
5	5	1,2	$SVM^{5,1}$
	5	3,4,7	$SVM^{5,2}$
7	7	1,2	$SVM^{6,1}$
	7	3,4,5	$SVM^{6,2}$

Table 3. Sum of decision value

Class	Models	Decision Value	Sum
1	$SVM^{1,1}$	1.3327	**2.9097**
	$SVM^{1,2}$	1.5770	
2	$SVM^{2,1}$	-1.1969	-1.7065
	$SVM^{2,2}$	-0.5096	

3	$SVM^{3,1}$	-1.2329	-1.8165
	$SVM^{3,2}$	-0.5836	
4	$SVM^{4,1}$	-1.0806	-2.3986
	$SVM^{4,2}$	-1.3180	
5	$SVM^{5,1}$	-1.5501	-1.0287
	$SVM^{5,2}$	0.5214	
7	$SVM^{6,1}$	-1.0730	-1.8449
	$SVM^{6,2}$	-0.7719	

3.2 Discussions on "Sum of Decision Values"

We will discuss the characters of one-against-half method by the sum of decision value. Suppose 12 SVM models were constructed by example 3.1. Now we use these models to predict one unknown-class data x, which is **1st class in reality**.

- **Discussion on D^1** D^1 is the sum of decision value of $SVM^{1,1}$ and $SVM^{1,2}$. Table 4 presents four kinds of the results of D^1. If both $SVM^{1,1}$ and $SVM^{1,2}$ predict correct, the two decision values are positive numbers. Of course, D^1 is positive. If only $SVM^{1,2}$ occurs fault, the decision value of $SVM^{1,2}$ is negative. But x is 1st class in reality, the decision value of $SVM^{1,2}$ will be very close to zero. Besides, the decision value of $SVM^{1,1}$ is still positive. So D^1 is positive in all probability. The 3rd kind of the results is similar to the above-mentioned. In the 4th circumstance, both $SVM^{1,1}$ and $SVM^{1,2}$ predict faults. Even though D^1 is negative, it will be very close to zero, too. Because of the real class of x is 1st class.

Table 4. Discussion on D^1

	Prediction		Decision Value		D^1
	$SVM^{1,1}$	$SVM^{1,2}$	$SVM^{1,1}$	$SVM^{1,2}$	
1	Correct	Correct	>0	>0	>0
2	Correct	Incorrect	>0	<0	Uncertain
3	Incorrect	Correct	<0	>0	Uncertain
4	Incorrect	Incorrect	<0	<0	<0

- **Discussion on D^2** D^2 is the sum of decision value of $SVM^{2,1}$ and $SVM^{2,2}$. Table 5 presents two kinds of the results of D^2. If $SVM^{2,1}$ predicts correct, the decision value of $SVM^{2,1}$ will be negative. However, $SVM^{2,2}$ is trained on data from class 2 and class 4, 5, 7. It causes using $SVM^{2,2}$ to predict x is unstable; the decision value may be positive or negative. Suppose the data from every class distribute similarly wide. In general, the probability that $SVM^{2,2}$ predicts x is class 2 is less than the other classes. So the expected value of the decision value of $SVM^{2,2}$ is negative, and getting negative D^2 is a strong probability. In the worst case, if $SVM^{2,1}$ predicts incorrect, the decision value will be positive. Because x is 1st class in reality, the decision value will be very closed to zero and D^2 has more probability than first case to be positive. When this situation occurs, one- against-half has more probability to predict faults. The discussions on $D^3, ..., D^6$ are similar to D^2.

Table 5. Discussion on D^2

	Prediction		Decision Value		D^2
	$SVM^{2,1}$	$SVM^{2,2}$	$SVM^{2,1}$	$SVM^{2,2}$	
1	Correct	--	<0	--	Uncertain
2	Incorrect	--	>0	--	Uncertain

Combining the two points at previous issue, we will get one conclusion: in the generally cases, D^1 is positive and $D^2, ..., D^6$ are negative. By Eq.(19), we easily say x is in the 1st class. In the last of this section, we randomly choose 100 testing data, which are **1st class in reality**, to test them and observe the sum of decision values. Figure 1 illustrate that in general D^1 are positive and D^2 are negative. Table 6 shows the expected value of "sum of decision values" and we find our inference is sensible.

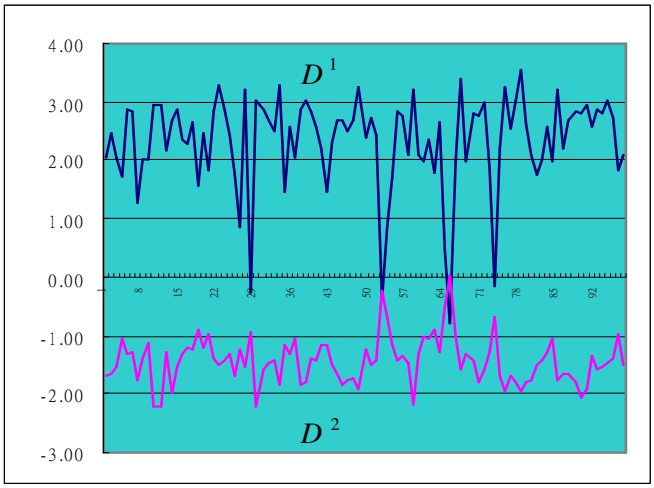

Figure 1. The distritution of D^1 and D^2

Table 6. Expected value of D^i

Class	Models	Exp (decision value)	Exp (D^i)
1	$SVM^{1,1}$	1.1909	2.3438
	$SVM^{1,2}$	1.1529	
2	$SVM^{2,1}$	-1.1251	-1.4400
	$SVM^{2,2}$	-0.3149	
3	$SVM^{3,1}$	-1.1128	-1.4819
	$SVM^{3,2}$	-0.3691	
4	$SVM^{4,1}$	-1.0818	-2.1247
	$SVM^{4,2}$	-1.0429	
5	$SVM^{5,1}$	-1.1299	-1.0762
	$SVM^{5,2}$	0.0537	
7	$SVM^{6,1}$	-1.0694	-1.7781
	$SVM^{6,2}$	-0.7086	

4. NUMERICAL EXPERIMENTS

4.1 Implement and result

In this section we present experimental results on 10 multi-class problems from UCI Repository (Blake and Merz, 1998) and LIACC [4]. From UCI Repository we choose the multi-class datasets: *iris*, *wine*, *glass*, and *vowel*. From LIACC we choose the datasets: *vehicle*, *segment*, *dna*, *satimage*, *letter*, and *shuttle*. We give the statistics of the problems in Table 7. In the last column we give the best accuracy rate listed in LIACC. Note that originally the problems *glass* and *satimage* both are 7-class problems, but they are regarded as 6-class problems because in the dataset no examples are with one class. All experiments in this section were done on a PentiumIV1500 with 384MB RAM. We implement one-against-half using MATLAB and every binary class SVM is constructed by LIBSVM (Chang and Lin, 2001). For each problem we stop the optimization algorithm if the constraints violation is less than 10^{-3}.

We use the same kernel function: **rbf** kernel and different kernel parameters γ and cost parameters C ($\gamma = [2^4, 2^3, \ldots, 2^{-10}]$ and $C = [2^{12}, 2^{11}, \ldots, 2^{-2}]$). So we try 225 combinations for every problem. We use holdout method to estimate the accuracy rates of problems *dna*, *satimage*, *letter*, and *shuttle*, because they are divided into training data (70%) and testing data (30%). For the other six smaller problems, we use 10-fold cross-validation (Kohavi, 1995) to estimate their accuracy rates.

Table 7. Problem Statistics

Problem	#class	#attributes	#training data	#testing data	statlog rate
iris	3	4	150	0	
wine	3	13	178	0	
glass	6	9	214	0	
vowel	11	10	528	0	
vehicle	4	18	846	0	78.3
segment	7	19	2310	0	96.9
dna	3	180	2000	1186	95.9
satimage	6	36	4435	2000	90.6
letter	26	16	15000	5000	93.6
shuttle	7	9	43500	14500	99.9

Table 8 is the result of comparing six multi-class SVMs. We present the optimal parameters (C,γ) and the corresponding accuracy rates. The best rates of six methods are represented by bold-faced, and "*" indicates using **improvement approach** (we will discuss it in 4.2). Note that we only estimate the accuracy rates by one-against-half method, and the other methods are experimented by Chang, et al. (2000). However the design of our experiments is the same as Chang, et al. (2000). Therefore we can compare one-against-half method with the other methods. Among the ten problems, one-against-half method obtains the best accuracy rates on *iris*, *wine*, *vowel*, and *dna*. For the other problems except *vehicle*, the accuracy rates are competitive with other methods. Although the accuracy of *vehicle* is the worst among these multi-SVMs, it (82.98%) still better than earlier results (78.3%) listed in statlog (see Table 6). We also illustrate the result of comparing these methods with Figure 2.

4.2 Improvement approach

We discuss the relation between "the standard deviation of every attribute(σ_A)"and "the correlation coefficient of every attribute and class ($\rho_{A,C}$)". Then we will find it affects the value of decision values. First, we show the decision value of x:

$$D=\sum_{i=1}^{Ns}\alpha_i y_i \Phi(s_i)\Phi(x)+b=\sum_{i=1}^{Ns}\alpha_i y_i k(s_i,x)+b \tag{21}$$

Suppose x has *n* attributes, $x=(x^1,x^2,...,x^n)$, and kernel function is rbf kernel. So the value of $k(s_i,x)$ is:

$$\begin{aligned} k(s_i,x)&=\exp\left(-\gamma\|s_i-x\|^2\right)\\ &=\exp\left(-\gamma\left(\sqrt{\left(s_i^1-x^1\right)^2+...+\left(s_i^n-x^n\right)^2}\right)^2\right)\end{aligned} \tag{22}$$

According to Eq.(22), when "the standard deviation of the attribute **a**"(σ_a) is much greater than the other attributes, it causes the variation of $\left(s_i^a-x^a\right)^2$ is also much greater than the others. It means that "attribute **a**" has much more influence on the decision values. If σ_a is much greater than others, the accuracy of one-against-half has two kinds of possibilities:

1. If σ_a is much greater and $\rho_{a,C}<0$, the accuracy of one-against-half method is usually worse.
2. If σ_a is much greater and $\rho_{a,C}>0$, the accuracy of one-against-half method is usually better.

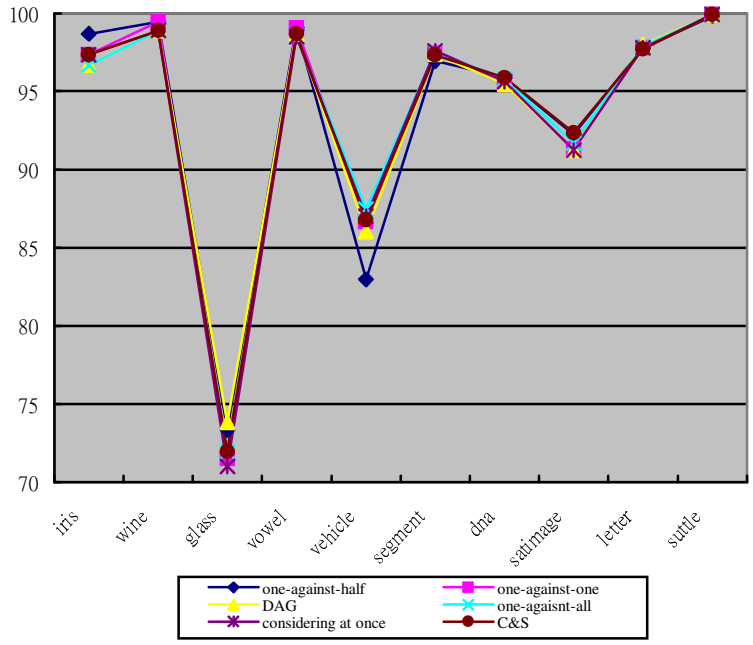

Figure 2. Comparing six multi-class SVMs

Table 8. Comparing six multi-class SVMs
(* indicates using improvement approach, before using *wine*:80.34%, *vehicle*:71.87%, *segment*: 96.49%)

Problem	One-against-half (C,γ) rate	One-against-one (C,γ) rate	DAG (C,γ) rate	One-against-all (C,γ) rate	Considering at once (C,γ) rate	C&S (C,γ) rate
iris	$(2^{-1},2^{-1})$ ***98.67***	$(2^{12},2^{-9})$ 97.33	$(2^{12},2^{-8})$ 96.67	$(2^{9},2^{-3})$ 96.67	$(2^{12},2^{-8})$ 97.33	$(2^{10},2^{-7})$ 97.33
wine	$(2^{5},2^{-10})$ ***99.44****	$(2^{7},2^{-10})$ ***99.44***	$(2^{6},2^{-9})$ 98.88	$(2^{7},2^{-6})$ 98.88	$(2^{0},2^{-2})$ 98.88	$(2^{1},2^{-3})$ 98.88
glass	$(2^{3},2^{-1})$ 73.36	$(2^{11},2^{-2})$ 71.50	$(2^{12},2^{-3})$ ***73.83***	$(2^{11},2^{-2})$ 71.96	$(2^{9},2^{-4})$ 71.03	$(2^{4},2^{-1})$ 71.96
vowel	$(2^{1},2^{0})$ ***99.05***	$(2^{4},2^{0})$ ***99.05***	$(2^{2},2^{2})$ 98.67	$(2^{4},2^{1})$ 98.49	$(2^{3},2^{0})$ 98.49	$(2^{1},2^{3})$ 98.67
vehicle	$(2^{6},2^{-10})$ 82.98*	$(2^{9},2^{-3})$ 86.64	$(2^{11},2^{-5})$ 86.05	$(2^{11},2^{-4})$ ***87.47***	$(2^{10},2^{-4})$ 87.00	$(2^{9},2^{-4})$ 86.76
segment	$(2^{3},2^{-10})$ 96.93*	$(2^{6},2^{0})$ 97.40	$(2^{11},2^{-3})$ 97.36	$(2^{7},2^{0})$ 97.53	$(2^{5},2^{0})$ ***97.58***	$(2^{0},2^{3})$ 97.32
dna	$(2^{4},2^{-6})$ ***95.87***	$(2^{3},2^{-6})$ 95.45	$(2^{3},2^{-6})$ 95.45	$(2^{2},2^{-6})$ 95.78	$(2^{4},2^{-6})$ 95.62	$(2^{1},2^{-6})$ ***95.87***
satimage	$(2^{4},2^{-10})$ 92.25	$(2^{4},2^{0})$ 91.30	$(2^{4},2^{0})$ 91.25	$(2^{2},2^{1})$ 91.70	$(2^{3},2^{0})$ 91.25	$(2^{2},2^{2})$ ***92.35***
letter	$(2^{6},2^{-4})$ 97.76	$(2^{4},2^{2})$ ***97.98***	$(2^{4},2^{2})$ ***97.98***	$(2^{2},2^{2})$ 97.88	$(2^{1},2^{2})$ 97.76	$(2^{3},2^{2})$ 97.68
shuttle	$(2^{4},2^{-10})$ 99.83	$(2^{11},2^{3})$ 99.92	$(2^{11},2^{3})$ 99.92	$(2^{9},2^{4})$ 99.91	$(2^{9},2^{4})$ 99.91	$(2^{12},2^{4})$ ***99.94***

When the first possibility occurs, our improvement approach is to reduce the influence of "attribute **a**". In other words, it lets every attribute be divided by its standard deviation. Then the standard deviation of every attribute is equal to one, and the influence of "attribute a" is equal to others. For example, Table 9 presents "the standard deviation of every attribute" and "the

correlation coefficient of every attribute and class " of problem *wine*. The attributes 5 and 13 occur first possibility, so the accuracy rate of one-against-half is 80.34%. But after using the improvement approach, its accuracy rate improves to 99.44%. Note that not every situation is suitable for using this approach. When second possibility occurs, this approach may reduce the accuracy rates of one-against-half method.

Table 9. σ_A and $\rho_{A,C}$ of wine

	1	2	3	4	**5**	6	7	8	9	10	11	12	**13**
σ_A	0.8	1.1	0.2	3.3	**14**	0.6	1	0.1	0.6	2.3	0.2	0.7	**315**
ρ_{AC}	-0.3	0.4	-0.1	0.5	**-0.2**	-0.7	-0.9	0.5	-0.5	0.3	-0.6	-0.8	**-0.6**

5. CONCLUSIONS AND FUTURE WORK

From the results of our experiments (see Table 8), we find that none of these six methods is absolutely better than others and one-against-half is also a good multi-class SVM. We also propose the improvement approach to modify one-against-half method when the first possibility occurs. Thus, one-against-half method is the other selection when we deal with a multi-class problem.

Finally, we simply discuss the extension of one-against-half: each class originally constructs two SVM models (total *2k* models), and we change it to construct three SVM models (total *3k* models). This extension uses the similar idea of "the sum of decision values"; every binary SVM is trained on data from 1/3 classes. In the future multi-class SVM research, we could also consider the idea of this extension.

REFERENCES

1. Blake C. L. and Merz. C. J. (1998). UCI repository of machine learning databases. Technical report, University of California, Department of Information and Computer Science, Irvine, CA. Available at http://www.ics.uci.edu/~mlearn/MLRepository.html.
2. Boser B. E., Guyon I. M., and Vapnik V. (1992). A training algorithm for optimal margin classifiers. In Fifth Annual Workshop on Computational Learning Theory, Pittsburgh, ACM.
3. Bottou L., Cortes C., Denker J., Drucker H., Guyon I., Jackel L., LeCun Y., Muller U., Sackinger E., Simard P., and Vapnik V. (1994). Comparision of classifier methods: a case study in handwriting digit recognition. In International Conference on Pattern Recognition, IEEE Computer Society, pp. 77-87.
4. Brazdil P., Gama J., LIACC, University of Porto Rua Campo Alegre 823 4150 Porto, Portugal. Available at http://www.liacc.up.pt/ML/statlog/
5. Burges C.J.C. (1998). A tutorial on support vector machines for pattern recogition. *Data Mining and Knowledge Discovery*, 2 (2): 955-974.
6. Chang C.-C. and Lin C.-J., LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm
7. Chang C.-C., Hsu C.-W., and Lin C.-J. (2000). The analysis of decomposition methods for support vector machines. *IEEE Transactions on Neural Networks* 11 (4): 1003-1008.
8. Cortes C. and Vapnik V. (1995). Support vector networks. *Machine Learning*, 20:273- 297.
9. Crammer K. and Singer Y. (2000). One the learnability and design of out put codes for multiclass problems. In *Computational Learning Theory*, pp. 35-46.
10. Cristianini N., Shawf-Taylor J. (2000). *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press.
11. Fletcher R. (1987). *Practical Methods of Optimization*. John Wiley and Sons, Inc., 2nd edition.
12. Friedman J. (1996). Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University.
13. Kohavi R. (1995). *A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection*. IJCAI 1995: 1137-1145.
14. KreBel U. (1999). Pairwise Classification and Support Vector Machines. *Advances in Kernel Methods—Support Vector Learning*, Cambridge, MA , MIT, pp. 254-268.
15. Platt J. C., Cristianini N., and Shawe-Taylor J. (2000). Large margin DAGs for multiclass classification. *Advances in Neural Information Processing Systems* 12: 547-553.
16. Schölkopf B., Burges C.J.C., Smola A.J. (1999). Introduction to Support Vector Learning, *Advances in Kernel Methods—Support Vector Learning*, Cambridge, MA, MIT, pp. 1-15.
17. Vapnik V. (1998). *Statistical Learning Theory*. Wiley.