Fast and Efficient Strategies for Model Selection of Gaussian Support Vector Machine

Zongben Xu, Mingwei Dai, and Deyu Meng

***Abstract*—Two strategies for selecting the kernel parameter (σ) and the penalty coefficient (C) of Gaussian support vector machines (SVMs) are suggested in this paper. Based on viewing the model parameter selection problem as a recognition problem in visual systems, a direct parameter setting formula for the kernel parameter is derived through finding a visual scale at which the global and local structures of the given data set can be preserved in the feature space, and the difference between the two structures can be maximized. In addition, we propose a heuristic algorithm for the selection of the penalty coefficient through identifying the classification extent of a training datum in the implementation process of the sequential minimal optimization (SMO) procedure, which is a well-developed and commonly used algorithm in SVM training. We then evaluate the suggested strategies with a series of experiments on 13 benchmark problems and three real-world data sets, as compared with the traditional 5-cross validation (5-CV) method and the recently developed radius-margin bound (RM) method. The evaluation shows that in terms of efficiency and generalization capabilities, the new strategies outperform the current methods, and the performance is uniform and stable.**

***Index Terms*—Data mining, kernel methods, pattern classification, support vector machine (SVM).**

I. Introduction

ROOTED in statistical learning theory (SLT), support vector machine (SVM) realizes the structural risk minimization (SRM) principle [1] by implementing classification to maximize the interclass margin(s) [2]–[5]. Due to its successful applications on character recognition, speaker identification, face recognition, gender classification, stock action prediction, etc. [5]–[12], SVM has significantly been highlighted in areas of data mining and machine learning.

Model selection of SVM is an important issue in SVM research. Specifically, the involved parameters in SVM, such as the kernel parameters (such as σ in Gaussian kernel) and the penalty coefficient C, always have a significant influence on the overall performance of the final obtained classifier. Designing reasonable strategies to tune these parameters has attracted more and more attentions in the latest years [13]–[25].

Manuscript received April 7, 2008; revised October 20, 2008 and January 3, 2009. First published March 31, 2009; current version published September 16, 2009. This work was supported in part by the Natural Science Foundation of China under Contract 60575045 and Contract 70531030 and in part by the National 973 program of China under Grant 2007CB311002. This paper was recommended by Associate Editor S.-F. Su.

The authors are with the Institute for Information and System Sciences, Faculty of Science, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: dymeng@mail.xjtu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCB.2009.2015672

So far, there have mainly been two categories of approaches to the model selection issue of SVM. The first category of approaches primarily takes the cross validation (CV) errors as the criteria to control the selection of the model parameters [13]–[15]. This category of approaches is most widely used nowadays due to its simplicity, reliability, and interpretability. Two typical approaches of this category are 5-cross validation (5-CV) [14] and leave one out (LOO) [15]. The procedures of those types of methods require a complete grid search over the whole parameter space that needs to be located in an interval of the feasible solution, and they also need to take an appropriate sampling step. These bring negativeness to the approaches since an appropriate sampling step varies from kernel to kernel, and the search interval may not be easy to locate without prior knowledge. Moreover, the complete grid search unavoidably brings very high computational burden, frequently excluding the possibility of their application to very-large-scale problems. In the latest years, a series of researches on improving the efficiency of the grid search have been proposed by applying modern optimization techniques, such as particle swarm optimization (PSO) [26]–[28], simulated annealing [29], and genetic algorithms [30]–[32], to guide the searching process of the optimal parameters. However, their effectiveness on large-scale applications still need to be further evaluated.

The second category of approaches takes a certain type of theoretical approximation, such as the influence-function-based estimation [33], [34], or upper bound estimations of CV errors (LOO error commonly), such as the radius-margin bound (RM), as the criteria to guide the model selection of SVM. The main idea is to find the optimal parameters for minimizing the approximations or the approximated error bounds. When the error bound functions are differentiable, the traditional gradient descent techniques are frequently adopted to realize the minimization. In applications (e.g., [16]–[21]), the very often used approximation error bounds are Joachim's bound, span bound, the generalized approximate cross-validation bound, and RM (i.e., Vapnik–Chervonenkis (VC) bound). Compared with the first category of approaches, this category of approaches can select the model parameters in a definite way (that is, through a definite procedure of minimizing an error function), and, therefore, has a much lower computation complexity. However, it is generally uncertain if a satisfied (sufficiently accurate and differentiable) error bound can be obtained, and even if it can, there exist inevitably gaps between the approximation error and the real error, which then might lead to an inappropriate selection of the parameters. Moreover, to iteratively attain the minimal value of the approximation error bounds, generally,

1083-4419/$25.00 © 2009 IEEE

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

multiple times of SVM training have to be implemented, which may then cause heavy burden of computation.

Aiming at alleviating these problems, we propose two novel strategies to specify the model parameters of a Gaussian SVM in this paper. Based on viewing the model selection problem as a recognition problem in visual systems, we derive a direct parameter setting formula for selecting the kernel parameter (σ) through finding a visual scale at which the global and local structures of a data set can be preserved in the feature space, and, meanwhile, the difference between the two structures can be maximized. We also propose a new heuristic for the selection of the penalty coefficient (C) through identifying some quantities to measure the classification extent of each training datum in the implementation process of the sequential minimal optimization (SMO) procedure, which is a well-developed and commonly used algorithm in SVM training. A series of experiments with standard benchmark data sets and real-world ones verify that the suggested strategies are capable of yielding the SVM classifier with higher generalization capability within a significantly less computation time, as compared with the well-known 5-CV method and the recently developed RM method (see [21]).

We organize the rest of this paper as follows. In Section II, we present a brief review of SVM. In Sections III and IV, the strategies to select the model parameters σ and C are respectively introduced. The experimental results are reported in Section V. Finally, we conclude this paper with some useful remarks in Section VI.

II. Brief Review of SVM

SVM is initiated to calculate the underlying classifier $f: R^n \to \{-1, 1\}$ of a given data set $D_l = \{x_i, y_i\}_{i=1}^l$, where $x_i \in R^n$, $y_i \in \{-1, 1\}$, and l is the number of the data set. The attribute x_i with label $y_i = +1$ represents the sample belonging to a positive class; otherwise, it belongs to a negative class. SVM works based on solving the following mathematical model:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{l}\xi_i \\ \text{s.t.} \quad & y_i\left(\langle \omega, x_i\rangle - b\right) \geq 1 - \xi_i, \qquad i = 1, \ldots, l \\ & \xi_i \geq 0, \qquad i = 1, \ldots, l \end{aligned} \tag{1}$$

where ω and ξ_i are unknown variables, $\|\omega\|$ is the reciprocal of the margin between two classes, ξ_i is the permitted classification error for the ith sample, and C is the preset penalty coefficient. Based on the SLT and SRM principles, all pursue of implementing the model is to reach a best compromise between the generalization capability (controlled by $(1/2)\|\omega\|^2$) and the approximation capability (controlled by $\sum_{i=1}^l \xi_i^2$) of the classifier [2], [4].

One of the most prominent developments in SVM is the introduction of kernel functions [35]. Through substituting the inner product $\langle x, y\rangle$ in the low-dimensional original space (R^n) by kernel $K(x, y) = \langle \varphi(x), \varphi(y)\rangle$ in the high-dimensional feature space, a nonlinear classification problem in the original space can equivalently be solved by considering a linear classification problem in the feature space. This greatly expands the scope of applicability of the original SVM. In this paper, we focus on the Gaussian SVM that involves application of the Gaussian kernel function defined by

$$K(x, y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}. \tag{2}$$

This is the type of SVMs most commonly used in applications nowadays.

In SVM, the optimization problem [see (1)] with kernel form is normally solved through transferring it to the following equivalent dual problem:

$$\begin{aligned} \min \quad & W(\alpha_1, \ldots, \alpha_l) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \text{s.t.} \quad & \sum_{i=1}^{l}\alpha_i y_i = 0, \qquad 0 \leq \alpha_i \leq C, \qquad i = 1, \ldots, l \end{aligned} \tag{3}$$

where α_i's $(i = 1, \ldots, l)$ are the Lagrangian variables. The problem (3) is a standard quadratic programming (QP) problem, and, therefore, α_i's $(i = 1, \ldots, l)$ are the solution if and only if the following so-called Karush–Kuhn–Tucker (KKT) conditions are fulfilled: for every $i = 1, \ldots, l$:

$$\begin{aligned} \alpha_i &= 0 \Rightarrow y_i\left(f(x_i) - y_i\right) > 0 \\ 0 < \alpha_i &< C \Rightarrow y_i\left(f(x_i) - y_i\right) = 0 \\ \alpha_i &= C \Rightarrow y_i\left(f(x_i) - y_i\right) < 0. \end{aligned}$$

Due to their simplicity and low complexity of calculation, the KKT conditions play an important role in accelerating the optimization procedure of SVM training. Denoting the optimal solutions of problem (3) by $\alpha_i^*(i = 1, \ldots, l)$, the resultant classifier of SVM is then defined by

$$f(x) = sign\left(\sum_{i=1}^{l}\alpha_i^* y_i K(x_i, x) + b\right).$$

Note that before implementing the Gaussian SVM, we have to preset the model parameters: the Gaussian kernel parameter σ and the penalty coefficient C. All simulations and applications show that these two parameters have a significant influence on the performance of the SVM, even compared with the model itself. Hence, to appropriately select such model parameters is crucial to the success of SVM. In this paper, we aim at developing a fast and efficient method for the selection of the parameters in SVM from a new perspective.

III. Strategy for Selection of the Gaussian Kernel Parameter

The SVM with kernel function K essentially transforms a nonlinear classification problem in the original space (R^n) with Euclidean distance $\|x - y\|$ into a linear classification problem in the feature space $(\mathcal{H})$ with distance $\sqrt{(K(x, x) + K(y, y) - 2K(x, y))}$ induced from the kernel function. The Gaussian kernel is normally preferable by virtue of its many perfect properties [36].

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

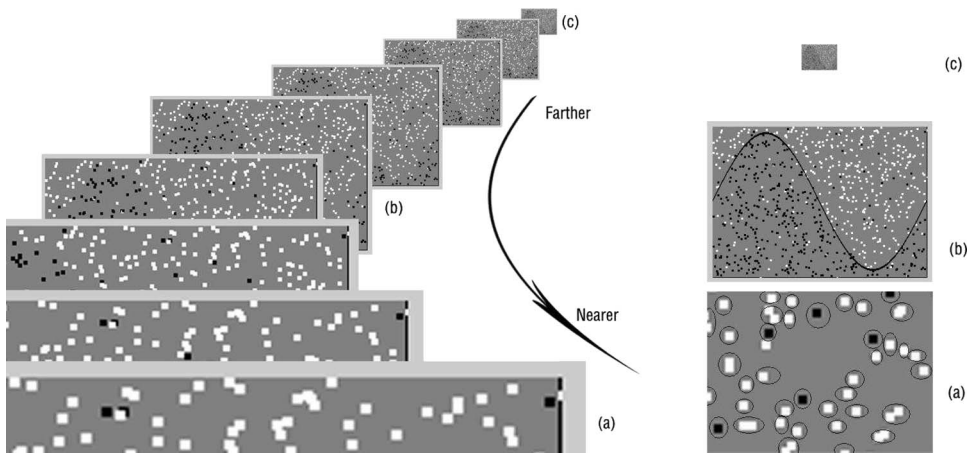


Fig. 1. Observing a data set from different scales.

One of such promising properties is the structure-preserving feature of the Gaussian kernel (that is, it preserves the ranking order of the distances between data pairs in the original and feature spaces). This is because, with the Gaussian kernel, the feature space $\mathcal{H}$ is known to be a reproduced Hilbert space with property $\mathcal{H} = \overline{span\{\varphi_1, \varphi_2, \ldots, \varphi_m, \ldots\}}$, where $\{\varphi_i\}_{i=1}^{\infty}$ is the complete system of orthonormal eigenvectors of a linear positive definite operator $T : L^2(R^n) \to L^2(R^n)$ induced by the Gaussian kernel [4]. In this case, the transformation Φ to realize the mapping from the original space R^n to the feature space $\mathcal{H}$ in SVM is defined by

$$\Phi(x) = (\varphi_1(x), \varphi_2(x), \ldots, \varphi_m(x), \ldots) \in \mathcal{H} \quad \forall x \in R^n.$$

Therefore, it satisfies $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$ for any x, y in R^n, and under the transformation, the original data set $D_l = \{x_i, y_i\}_{i=1}^{l}$ is transformed to the data set $\Phi(D_l) = \{\Phi(x_i), y_i\}_{i=1}^{l}$ in the feature space. With this understanding, we can calculate that the distance of any data pair $(\Phi(x), \Phi(y))$ in $\Phi(D_l) \subseteq \mathcal{H}$ is given by

$$\begin{aligned} \|\Phi(x) - \Phi(y)\| &= \sqrt{(K(x,x) + K(y,y) - 2K(x,y))} \\ &= \sqrt{2\left(1 - e^{-\frac{\|x-y\|^2}{\sigma^2}}\right)} \end{aligned} \tag{4}$$

which exhibits a positively proportional relation between $\|\Phi(x) - \Phi(y)\|$ and $\|x - y\|$. This shows the structure-preserving property of the Gaussian kernel. The structure-preserving property might be the essential reason why the SVM with Gaussian kernel performs more effectively than others in general.

Another promising property of the Gaussian Kernel is its connection with our everyday visual experience [5], [37]. When we watch a data set in plane (cf. Fig. 1), viewed as an image, every individual datum is amplified when the watching distance is very near; then, each datum is observed, and a proper structure (e.g., the classification boundary) gradually appears as the watching distance gets far. If the distance between us and the data image continually becomes far, a more blurred image then appears, and the structure disappears until only a blob is observed when the watching distance becomes sufficiently far away.

This everyday visual experience has been modeled in visual theory, and it is shown (see, e.g., [37]) that the Gaussian filtering of the initial image gives the blurred image of visual observation at scale σ. When we implement the SVM with Gaussian kernel, a completely same phenomenon occurs: when σ is very small, the yielded SVM classifier fully accords with the given label for each training sample, whereas it always appears as a very complicated shape, having a perfect approximation but without any generalization capability. However, when σ is set very large, a very smooth classifier (commonly a linear classifier) is yielded, which has very strong generalization but without approximation capability. Only when the scale is set appropriately, that is, neither too small nor too large, would the obtained classifier possibly find a proper classification surface, reaching an optimal compromise between the two capabilities.

Let us further explain why the SVM performs like this to motivate our idea for the selection of the Gaussian kernel parameter.

When σ is very small, we can deduce from (4) that the distances between all data pairs in the feature space tend to be close to the maximal value $\sqrt{2}$ (as shown in Fig. 2 when $\sigma = 0.1, 1$), so all structures in the data set are nearly dismissed. More specifically, the local structure of the data set (the data pairs with small distances) in the original space is inclined to be destroyed in the feature space, so each data point (including noises) is deviated from its local neighbors to form an isolated plot in the feature space. Hence, each datum can correctly be classified by the yielded classifier, i.e., it has perfect approximation capability. However, since any new input is also inclined to be an isolated plot, no reference data can properly guide its classification. Thus, the classifier has no or very poor generalization capability.

On the other hand, when σ is very large, the distances between all data pairs approach the minimal value 0 (as shown in Fig. 2 when $\sigma = 30, 100$). In this case, the global structure (the data pairs with large distances) in the original space tends to be broken in the feature space, and each original faraway data pair may appear very close in the feature space. In such a

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

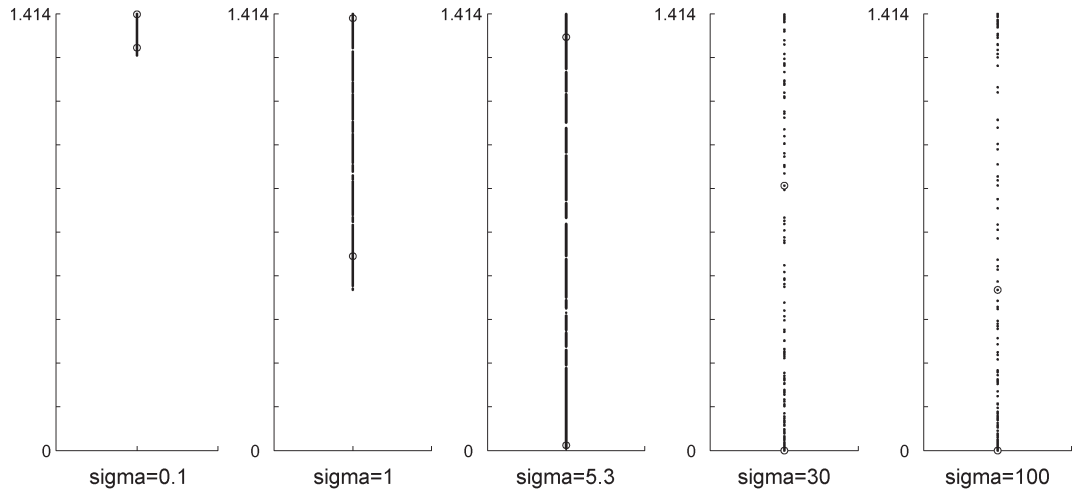


Fig. 2. Demonstration (in y-axis) for the distance distributions of a data set in the Gaussian feature spaces under different kernel parameters. The data set contains 1000 points, which are randomly generated from the 2-D $(0,1) \times (0,1)$ box. The circled points are the $\lfloor 0.9 * N(l-1) \rfloor$th distance and the $\lfloor 0.1 * N(l-1) \rfloor$th distance, respectively, where $N = 1000$.

way, all the data are highly syncretized in the feature space, and the classifier can no longer distinguish labels of the given data, leading to a very poor approximation capability. In this case, the generalization of any rule from such syncretized data does not make sense, so the yielded classifier could be regarded as having a very strong generalization capability (even if it does not necessarily imply a meaningful result).

The above explanation and the visual experience both suggest that there is an appropriate scale, for example, σ^*, at which the classification structure of a data set image can best be observed. From this point of view, the Gaussian kernel parameter selection problem can then be viewed as a recognition problem in a visual system: to find a best visual scale at which the image can most clearly be understood.

The problem is how to find σ^* in a practical, fast, and efficient way. Our idea is to look for such a scale at which the global and local structures of a data set can be preserved in the feature space, and, meanwhile, the difference between the two structures can be maximized.

To realize this, we may first pick a comparatively large distance expressed by a Gaussian kernel as a measure for the global structure and a small distance for the local structure in the feature space (in doing so, some possibly extreme cases caused by noise or outlier of the data should be excluded for reliability). Then, we determine an appropriate value of σ to make the global structure measure as large as possible and the local structure measure as small as possible, so as to maximize the gap (the difference) between the two measures in the feature space. A concrete method may be as follows.

First, we set the global and local structure measures in the feature space according to the following way: Randomly select N $(N \leq l)$ samples from the given data set. Calculate the Euclidean distances between the data in the selected set and all the ones in the given data set, and rank them in increasing order. Pick the $\lfloor (1-\alpha) * (N(l-1)) \rfloor$th distance and the $\lfloor \alpha * (N(l-1)) + 1 \rfloor$th distance $(0 \leq \alpha < 0.5)$ in the ordered sequence as the global and local structure measures d_{far} and d_{near}, respectively.

Note that the effect of the aforementioned α is to set d_{far} and d_{near} as a comparatively large value and a small one from all $N(l-1)$ distances between the selected data with number N and the whole data with number l (N 0-distances between selected samples have been eliminated). Particularly, as α is taken as 0, d_{far} and d_{near} correspond, respectively, to the largest and smallest distances of the whole distance set, and evidently, in this case, the deviation between the two values are maximized. However, when α inclines to 0.5, this deviation tends to vanish. Generally speaking, both of the above extreme cases are not preferred. On one hand, if α is set too small, the possible existence of outliers or noises in the given data set tends to negatively influence the effectiveness of the latter-proposed model selection strategy constructed based on the preset d_{far} and d_{near} (This negative influence can be obviously demonstrated in Fig. 5). On the other hand, if it is set too adjacent to 0.5, then it is evident that d_{far} and d_{near} so calculated have lost the significance of giving global and local structure measures of the data set. Therefore, the value of α should be set neither too adjacent to 0 or 0.5. The strategy to set the value of α will further be discussed toward the end of this paper.

Then, we determine the optimal σ by maximizing the difference between the local and global structure measures in the feature space corresponding to d_{far} and d_{near}. According to (4), we can formulate the difference function of the two squared measures in the feature space mathematically as

$$\begin{aligned} dif(\sigma) &= \left(2 - 2e^{-\frac{d_{\text{far}}^2}{2\sigma^2}}\right) - \left(2 - 2e^{-\frac{d_{\text{near}}^2}{2\sigma^2}}\right) \\ &= 2\left(e^{-\frac{d_{\text{near}}^2}{2\sigma^2}} - e^{-\frac{d_{\text{far}}^2}{2\sigma^2}}\right). \end{aligned}$$

Denote the optimal σ at which $dif(\sigma)$ attains its maximum by σ_{opt}. Then

$$\begin{aligned} \sigma_{\text{opt}} &= \arg\max_{\sigma} dif(\sigma) \\ &= \arg\max_{\sigma} \left(e^{-\frac{d_{\text{near}}^2}{2\sigma^2}} - e^{-\frac{d_{\text{far}}^2}{2\sigma^2}}\right). \end{aligned}$$

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

To simplify the calculation, we define $\gamma = (1/2\sigma^2)$. Then, $dif(\sigma)$ and σ_{opt} can be rewritten as

$$dif(\gamma) = e^{-\gamma d_{\text{near}}^2} - e^{-\gamma d_{\text{far}}^2}$$
$$\gamma_{\text{opt}} = \arg\max_{\gamma} \left(e^{-\gamma d_{\text{near}}^2} - e^{-\gamma d_{\text{far}}^2} \right).$$

Differentiating $dif(\gamma)$ with respect to γ, we get

$$\frac{\partial dif(\gamma)}{\partial \gamma} = e^{-\gamma d_{\text{near}}^2} \left(-d_{\text{near}}^2\right) - e^{-\gamma d_{\text{far}}^2} \left(-d_{\text{near}}^2\right)$$

and γ_{opt} satisfies

$$e^{-\gamma_{\text{opt}} d_{\text{near}}^2} \left(-d_{\text{near}}^2\right) - e^{-\gamma_{\text{opt}} d_{\text{far}}^2} \left(-d_{\text{near}}^2\right) = 0$$

that is

$$-\gamma_{\text{opt}} d_{\text{near}}^2 + \ln d_{\text{near}}^2 = -\gamma_{\text{opt}} d_{\text{far}}^2 + \ln d_{\text{far}}^2.$$

Therefore, we find

$$\gamma_{\text{opt}} = \frac{\ln d_{\text{far}}^2 - \ln d_{\text{near}}^2}{d_{\text{far}}^2 - d_{\text{near}}^2}$$

and, hence, σ_{opt} is given by

$$\sigma_{\text{opt}} = \sqrt{\frac{1}{2\gamma_{\text{opt}}}} = \sqrt{\frac{d_{\text{far}}^2 - d_{\text{near}}^2}{2\left(\ln d_{\text{far}}^2 - \ln d_{\text{near}}^2\right)}}. \tag{5}$$

To summarize, we suggest the following method to specify the Gaussian kernel parameter σ in SVM.

Algorithm 1: Selection for Gaussian Kernel Parameter σ

Step I. Randomly select N samples $\{\overline{x}_i, i = 1, \ldots, N\}$ in D_l. Calculate the Euclidean distances between the selected data and all the ones in the given set, and rank them in increasing order.

Step II. Pick up the $\lfloor (1-\alpha) * (N(l-1)) \rfloor$th and the $\lfloor \alpha * (N(l-1)) \rfloor + 1$th distances from the ranked distances as the global and local structure measures d_{far} and d_{near}, respectively, where α is a small real number (for example, set to be 0.1 in our experiments).

Step III. Calculate the optimal value σ_{opt} according to (5).

Note that when the sample size is not too large, we can set $N = l$; otherwise, we can set N to be an integer less or much less than l to save computation time.

IV. Strategy for Selection of the Penalty Coefficient

In this section, we develop a strategy to select the penalty coefficient C.

As it is known, the role of the penalty coefficient C consists of balancing the generalization capability of an SVM classifier controlled by the term $(1/2)\|\omega\|^2$ and the approximation capability controlled by the term $\sum_{i=1}^{l} \xi_i$ in the SVM model [see (1)]. When C is set too small (e.g., close to 0), the SVM model puts emphasis on the former and comparatively neglects the latter so that it yields a classifier with good generalization capability but poor approximation capability. However, when C is set too large, the corresponding model tends to be degraded as the traditional empirical risk minimization (ERM) model, which highly focuses on the fitness of the obtained classifier on the training examples, but less on generalization. This then leads to the well-known overfitting problem. Hence, the value of C should also be set neither too small nor too large as that for the Gaussian parameter σ.

Once the model parameters are set, an SVM can be trained through many available algorithms. The most well-known and commonly used algorithm, for instance, is the SMO algorithm [38], [39]. The SMO algorithm has been proven to be very fast and effective in applications [40]. Moreover, some useful information, which might had been ignored by previous researches, can be identified and extracted for guiding the selection of model parameters. Our aim in this section is to explore such possibility.

In principle, the SMO algorithm solves model (3) through decomposing the original large QP problem into a series of the smallest possible QP problems, each of which contains only two variables and can analytically be solved. The smallest QP problem with variables α_1 and α_2 is of the form

$$\begin{aligned}
\min_{\alpha_1,\alpha_2} \quad & W(\alpha_1,\alpha_2) = \frac{1}{2}K(x_1,x_1)\alpha_1^2 + \frac{1}{2}K(x_2,x_2)\alpha_2^2 \\
& + y_1 y_2 K(x_1,x_2)\alpha_1\alpha_2 - (\alpha_1+\alpha_2) \\
& + y_1\alpha_1 \sum_{i=3}^{l} y_i\alpha_i K(x_i,x_1) \\
& - y_2\alpha_2 \sum_{i=3}^{l} y_i\alpha_i K(x_i,x_2) + c_1 \\
\text{s.t.} \quad & \alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{l} y_i\alpha_i = c_2, \\
& 0 \le \alpha_i \le C, \quad i = 1,\ldots,l
\end{aligned} \tag{6}$$

where c_1 and c_2 are two constants independent of α_1 and α_2. According to [38] and [39], if we only minimize $W(\alpha_1,\alpha_2)$ without considering the constraints, then the iterative procedures can then be formulated as

$$\begin{aligned}
\alpha_1^{\text{new}} &= \alpha_1^{\text{old}} + \frac{y_1(E_2 - E_1)}{\kappa} \\
\alpha_2^{\text{new}} &= \alpha_2^{\text{old}} + \frac{y_2(E_1 - E_2)}{\kappa}
\end{aligned} \tag{7}$$

where $E_i = f^{\text{old}}(x_i) - y_i$, $i = 1, 2$. If the constraints in (6) are considered, then α_1^{new} and α_2^{new} need to be clipped further to amend the values in the iteration as

$$\alpha_1^{\text{new,clipped}} = \begin{cases} H, & \text{if } \alpha_1^{\text{new}} \ge H \\ \alpha_1^{\text{new}}, & \text{if } L < \alpha_1^{\text{new}} < H \\ L, & \text{if } \alpha_1^{\text{new}} \le L \end{cases}$$

$$\alpha_2^{\text{new,clipped}} = \begin{cases} H, & \text{if } \alpha_2^{\text{new}} \ge H \\ \alpha_2^{\text{new}}, & \text{if } L < \alpha_2^{\text{new}} < H \\ L, & \text{if } \alpha_2^{\text{new}} \le L \end{cases}$$

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

where

$$L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right)$$
$$H = \min\left(C, C + \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right), \quad \text{if } y_1 \neq y_2$$
$$L = \max\left(0, \alpha_2^{\text{old}} - \alpha_1^{\text{old}} - C\right)$$
$$H = \min\left(C, \alpha_2^{\text{old}} - \alpha_1^{\text{old}}\right), \quad \text{if } y_1 = y_2.$$

In application of the SMO algorithm, α_1^{new} and α_2^{new} in (7) are normally used as intermediate values for clipping. There hides, however, very useful information in α_1^{new} and α_2^{new} before clipping. Looking at the computing procedure [see (7)], for example, we can observe the following.

1) When x_i is correctly classified by the current classifier f^{old}, i.e., $y_i(f^{\text{old}}(x_i) - y_i) > 0 (y_i E_i > 0)$, then α_i^{new} tends to be smaller than α_i^{old}. Furthermore, if x_i is far-away from the classification surface, i.e., $y_i(f^{\text{old}}(x_i) - y_i) \gg 0 (y_i E_i \gg 0)$, then α_i^{new} inclines to be much smaller than α_i^{old}.
2) When x_i is incorrectly classified by the current classifier f^{old}, i.e., $y_i(f^{\text{old}}(x_i) - y_i) < 0 (y_i E_i < 0)$, then α_i^{new} tends to be larger than α_i^{old}. In this case, if x_i is further faraway from the classification surface, i.e., $y_i(f^{\text{old}}(x_i) - y_i) \ll 0 (y_i E_i \ll 0)$, then α_i^{new} is inclined to be much larger than α_i^{old}.

From this observation, a very interesting conclusion can be drawn. After all the iterations of the algorithm, the values of all α_i^{new} $(i = 1, \ldots, l)$ in the memory naturally yield measures for classification extent of the training data. Specifically, if α_i^{new} for some sample x_i is small, then it shows that x_i has correctly been classified by the current classifier. If α_i^{new} is quite small, then not only has x_i correctly been classified but it should also be inside the y_i-class area defined by the classifier. However, if α_i^{new} is large, then the corresponding x_i tends to be in the classification surface or be classified incorrectly; if it is quite large, then x_i inclines to be deviated greatly from the y_i-class area. Therefore, α_i^{new} can be viewed as a reasonable indication of classification degree of the corresponding sample x_i.

From this sense, we can see that clipping (i.e., the last step of the SMO algorithm) plays the role of limiting the magnitude of α_i^{new} regulated by the penalty coefficient C. The limitations should be neither too strong nor too weak. With reference to (6), we define the limitation strength of α_i^{new} as

$$\text{limitation}_i = \begin{cases} 0, & \alpha_i^{\text{new}} < C \\ \alpha_i^{\text{new}} - C, & \alpha_i^{\text{new}} \geq C. \end{cases}$$

That is, we hope that clipping does not affect or heavily affect those samples with α_i^{new} smaller than C, and only those samples with α_i^{new} greater than C are limited.

Turning this idea to specify the penalty coefficient C, we then suggest to update the currently set C through successively relaxing its current assignment so as to tightly bound the current α_i's in a certain sense. More specifically, we propose the following strategy for updating the penalty coefficient C. First, we start with a small enough C to guarantee it to be smaller than the real optimal value. Second, we train the SVM model by the SMO algorithm under current C and store all α_i's. Then, we take the mean of limitation strength on all α_i^{new} as the incremental value of C for the next iteration.

In practice, we denote $\Lambda' = \{\alpha_i^{\text{new}} : \alpha_i^{\text{new}} \geq C\}$ and denote by Λ the set formed through deleting the largest 10% ones in Λ' to avoid the infections of the extreme cases (i.e., outliers and noisy samples). Then, we define

$$C^{\text{new}} = C^{\text{current}} + \frac{1}{|\Lambda|} \sum_{i \in \Lambda} \left(\alpha_i^{\text{new}} - C^{\text{current}}\right) \tag{8}$$

where $|\Lambda|$ is the number of elements contained in Λ. Such iteration is continued until a convergence condition is met.

A problem still remains as to when to terminate the above update procedure. To find such a heuristic, we have run a set of simulations of SVM with 13 benchmark data sets provided by [41]. These data sets have originally been used in [42] and are very good for classification tasks [43]. We calculated the correct classification rate of the SVM through training the machine on the given training sets and testing the machine on the validation sets with varying penalty coefficient $C = e^i$, $i = -7, -6, \ldots, 0, 1, \ldots, 8$. The classification rate of SVMs with different C is listed in Table I and Fig. 3. It can clearly be observed that the correct classification rate of SVM first increases as C increases, and after reaching the maximum at a certain C, it no longer stops increasing. This suggest the possibility and rationality of taking the classification/misclassification rate (MR) as the criterion to guide the termination of the update procedure. This criterion has originally been discovered in [47]. In the following, we decide to stop the update procedure of C once the classification rate of the obtained classifier no longer increases.

With such a criterion, we now suggest the following procedure for selection of the penalty coefficient C.

Algorithm 2: Selection of penalty coefficient C

Step I. Calculate the optimal kernel parameter σ_{opt} according to Algorithm 1 and initialize a small value to C^{current}. Set the correct classification rate $R = 0$ and a small threshold $\varepsilon > 0$.

Step II. **Do** the following iteration until convergence:

Step II.1. Apply the SMO algorithm on the training set to get C^{new} by utilizing (8) under σ_{opt} and C^{current}. If $\|C^{\text{new}} - C^{\text{current}}\| < \varepsilon$, stop the iteration.

Step II.2. Evaluate the correct classification rate R' under σ_{opt} and C^{new}. If $R' \leq R$, stop the iteration; otherwise, let $R = R'$, $C^{\text{current}} = C^{\text{new}}$.

End Do

Step III. Output $C_{\text{opt}} = C^{\text{current}}$ as the optimal selection of the penalty coefficient value.

Note that to promise feasibility of the above algorithm, it needs to be designed in advance how to evaluate a reasonable classification rate R' of a classifier (Step II.1). Based on the existing research, CV errors, like LOO error or k-fold CV error, provide the mostly appropriate criterion to measure the classification rate of a classifier. Hence, we suggest the use of such type of error (particularly, 5-fold CV error has been

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

TABLE I
CLASSIFICATION RATES OF SVM WITH DIFFERENT PENALTY COEFFICIENT C

	C							
data	e^{-7}	e^{-6}	e^{-5}	e^{-4}	e^{-3}	e^{-2}	e^{-1}	e^{0}
banana	55.95	55.95	55.95	55.95	55.95	63.66	76.27	82.74
breast-cancer	30.77	69.23	69.23	69.23	69.23	69.23	71.79	71.79
diabetis	32.56	32.56	67.44	67.44	68.44	76.41	75.75	75.75
flare-solar	44.64	55.36	55.36	55.36	55.36	63.84	65.33	65.33
german	27.91	72.09	72.09	72.09	72.09	72.09	77.41	79.40
heart	43.56	56.43	56.43	56.43	66.34	78.22	81.19	82.18
image	42.53	42.53	42.53	64.39	73.78	74.78	84.07	86.05
ringnorm	50.62	49.38	49.38	51.42	97.53	97.68	98.03	98.00
splice	48.02	51.97	51.97	51.97	53.58	85.52	88.28	90.21
thyroid	35.52	35.52	35.52	35.52	64.47	80.26	89.47	90.79
titanic	32.60	32.60	67.40	67.40	67.40	77.05	77.05	77.05
twonorm	50.06	50.06	49.93	82.98	97.61	97.71	97.63	97.06
waveform	32.95	67.05	67.05	67.05	83.33	87.13	88.98	89.33

	C							
data	e	e^{2}	e^{3}	e^{4}	e^{5}	e^{6}	e^{7}	e^{8}
banana	85.27	87.21	87.92	88.35	89.18	89.35	88.57	88.04
breast-cancer	71.79	69.23	66.67	66.67	62.82	66.67	64.10	64.10
diabetis	75.41	75.41	75.75	74.08	73.42	73.08	71.43	72.76
flare-solar	65.58	65.58	65.58	65.58	65.09	64.84	64.84	64.59
german	79.73	78.74	78.07	76.74	74.42	73.75	73.75	73.75
heart	83.17	83.17	85.15	85.15	85.15	85.15	85.15	85.15
image	89.61	90.50	92.08	93.87	95.55	96.24	96.83	97.53
ringnorm	97.93	97.78	97.78	97.78	97.78	97.78	97.78	97.78
splice	90.30	90.39	90.39	90.39	90.39	90.39	90.39	90.39
thyroid	92.10	93.42	94.74	94.74	96.05	94.74	94.74	93.42
titanic	77.05	77.05	77.05	77.05	77.05	77.05	77.05	77.05
twonorm	96.68	95.98	95.93	95.93	95.93	95.93	95.93	95.93
waveform	89.50	89.24	88.50	88.48	88.48	88.48	88.48	88.48

adopted in the following experiments) in the application of the algorithm.

V. EXPERIMENT RESULTS

We provide a series of experiments in this section to demonstrate the rationality, effectiveness, and high efficiency of the suggested new strategies for SVM model selection.

Our experiments were made with two different family of data sets. One was with the standard 13 benchmark problems commonly utilized in previous researches, and the other with three real-world data sets from University of California, Irvine, DELVE, STAT-LOG, and other benchmark repositories. The first family consisted of small- or medium-sized data sets (each less than 1500 samples), designed for facilitating the overall performance evaluation of the new algorithms due to the known nearly optimal solutions checked by the CV technique. The second family was of relatively large-scale data sets (each larger than 30 000 samples), the optimal solutions of which were unknown, and was designed for supporting the real performance assessment of the new strategies. For comparison, we applied our new algorithms together with the most accepted 5-CV method [14] and a newly developed RM method [21]. When applying Algorithm 1 in our method, we set $\alpha = 0.1$, $N = l$ for

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

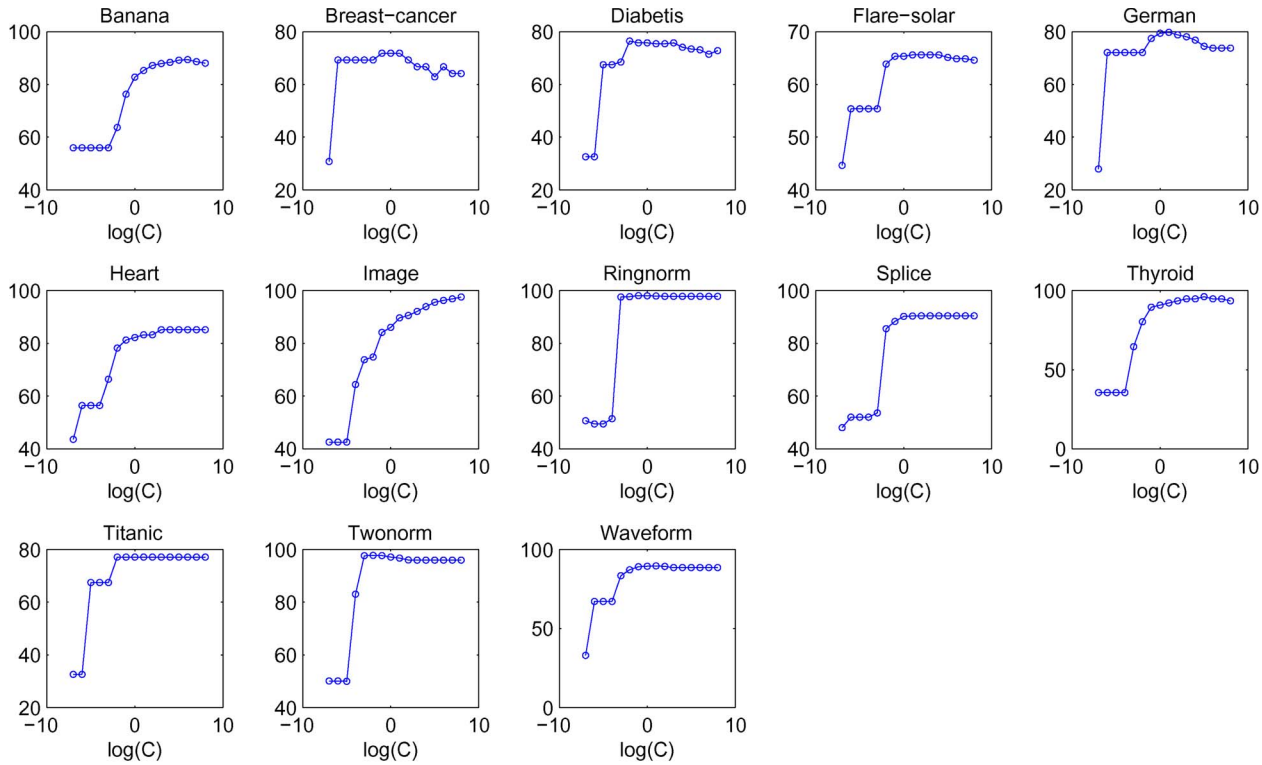


Fig. 3. Evolution of correct classification rates of SVM as C varies when applied to 13 benchmark data sets.

the first family of data sets, and $\alpha = 0.1$, $N = 0.5l$ for the second one. We ourselves have written the program of the 5-CV method, in which a 16 × 16 grid of (σ, C) values varying from e^{-7} to e^{8} were considered as candidates to be selected, whereas that of the RM method was downloaded directly from http://www.csie.ntu.edu.tw/cjlin/libsvmtools/#12. All programs were written in C++ language and realized in the platform of VC++.net 2003. The implementation environment was a personal computer with Genuine Intel T2400 at 1.83 G (CPU), 1024 MB (memory), and Windows XP (operating system).

A. Experiments With Benchmark Problems

In this series of experiments with 13 benchmark problems, our aim is to evaluate the overall performance of the new algorithms as compared with the 5-CV method and the RM method in terms of efficiency and ability of yielding the SVM classifier with higher generalization capability. The related information of the benchmark problems is summarized in Table II, where the number of samples in the training set (Ntrain), the number of samples in the testing set (Ntest), and the dimension of the corresponding data set (Dim) are listed for each data set.

For each problem, the data set contains 100 partitions (except 20 partitions for Image and splice problems) of training sets and testing sets (approximately 60% : 40%). For a fixed SVM implementation scheme (that is, an SVM model with fixed model parameters), through training the SVM on each training set and testing the classifier on the corresponding testing set, we obtained 100 (or 20) SVM classifiers with different MRs for each problem. The average and variance of the MRs over the 100 (or 20) classifiers were then taken as the measures of generalization ability of the SVM scheme when applied to the problem.

TABLE II
INFORMATION OF THE 13 BENCHMARK DATA SETS

name	$Ntrain$	$Ntest$	Dim
banana	400	4900	2
breast-cancer	200	77	9
diabetis	468	300	8
flare-solar	666	400	9
german	700	300	20
heart	170	100	13
image	1300	1010	18
ringnorm	400	7000	20
splice	1000	2175	60
thyroid	140	75	5
titanic	150	2051	3
twonorm	400	7000	20
waveform	400	4600	21

For each involved model selection method, we determined the parameters C and σ in the following way. First, we

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

TABLE III
MODEL PARAMETERS AND NUMBERS OF SVS CONDUCTED BY THE RM METHOD, THE 5-CV METHOD, AND THE NEW METHOD ON THE BENCHMARK DATA SETS

	RM method			5-CV method			New method		
	σ	C	NSV	σ	C	NSV	σ	C	NSV
banana	0.42	0.44	179	0.85	235.9	90	1.18	67.20	113
breast-cancer	5.54	0.0003	115	10.95	828.1	107	2.77	1.75	122
diabetis	0.11	0.001	425	11.88	249.6	239	2.64	0.99	263
flare-solar	0.52	0.07	540	10.43	13.68	505	1.30	1.02	508
german	1.08	4.21	696	12.97	42.82	391	4.27	2.63	428
heart	1.38	4.96	152	23.89	230.3	65	3.45	1.24	92
image	0.54	14.10	753	3.43	768.6	149	3.65	286.10	299
ringnorm	2.27	2.48	144	2.72	91.6	106	4.19	1.39	96
splice	4.70	3.14	811	7.18	702.7	578	8.42	1.11	707
thyroid	1.08	1.03	48	0.93	626.8	41	1.96	13.40	24
titanic	1.11	0.19	83	0.66	220.2	69	0.43	1.23	71
townorm	2.77	3.87	154	5.31	0.74	104	4.33	0.87	104
waveform	1.93	3.27	280	7.44	9.78	115	4.50	2.05	136
mean	1.803	2.905	337	7.587	309.291	197	3.31	29.306	228

estimated the parameters of the Gaussian SVM by applying the method to the first five partitions of each benchmark data set. Then, we computed the median of the obtained five estimations and taken the median values as the final model parameters. Once the parameters C and σ were set in this way, an SVM implementation scheme (an SVM scheme in short) was defined. Then, the experiment results with such an SVM scheme were taken as the results of the corresponding model selection method. In consequence, the capability of a model selection method was measured with the MR of the corresponding SVM scheme, whereas the efficiency of a model selection method was measured directly with the average computation time for the estimation of the parameters. Evidently, the smaller the MR and computation time, the better a model selection method.

The experiment results with the 5-CV method, the RM method, and our new method are summarized in Tables III and IV and Fig. 4. Particularly, the calculated optimal model parameters C and σ, the number of support vectors (NSVs), the computation time (Time), and the corresponding MR are listed in two tables for each method. An the NSVs, computational times, means, and variances of the MRs for each of the 13 benchmark data sets are depicted in Fig. 4.

From Fig. 4 and Tables III and IV, it can be seen that our new method, overall, outperforms the 5-CV method and the RM method. In particular, all cases for 13 benchmark data sets report less computation time for the new method: approximately three times higher than that of the RM method and more than 500 times higher than that of the 5-CV method (in average). As generalization capability is concerned, the new method has minimal average MR among the three methods for both mean and variance. Considering that the 5-CV method has commonly been accepted as the best and reliable method that can very often yield a nearly optimal classifier, we conclude that the new method is not only feasible but reliable and efficient as well.

An interesting observation can be made from Table III. The parameters σ and C found by the new method tend to be smaller than those found by the 5-CV method and larger than those found by the RM method (that is, located between the two values found by the RM and 5-CV methods). As the analysis conducted in Sections III and IV implies, either too small or too large σ and C is inclined to lead to poor performance of SVM in application, and, hence, a moderate value might be more preferable. Our experiments are evidently supported by this assertion. Therefore, to a certain extent, this gives an explanation on why the new method outperforms the 5-CV and RM methods. It can also be observed from Table II and Fig. 4 that the numbers of support vectors (SVs) found by the new algorithm are also in the middle of the two found by the other two methods (in average, the 5-CV method tends to get the least, and the RM method tends to get the largest). This shows that the classifier yielded by the 5-CV method tends to be the simplest one. In this sense, 5-CV might have a better performance than the other two methods. Nevertheless, its huge computation burden has excluded itself from being an even feasible model selection method in SVM when applied to large-scale problems.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

TABLE IV
COMPUTATIONAL COSTS AND MRS CONDUCTED BY THE RM METHOD, THE 5-CV METHOD, AND THE NEW METHOD ON THE BENCHMARK DATA SETS

	RM method		5-CV method		New method	
	Time(s)	MR(%)	Time(s)	MR(%)	Time(s)	MR(%)
banana	0.85	10.48 ± 0.40	211.8	11.53±0.66	0.34	10.68±0.50
breast-cancer	0.33	27.83±4.62	57.9	26.04±4.74	0.12	24.97 ± 4.62
diabetis	2.58	34.56±2.17	332.4	23.53±1.73	0.29	23.16 ± 1.65
flare-solar	1.28	35.51±1.65	435.98	34.43±1.82	1.24	32.37 ± 1.78
german	5.70	28.84±2.03	772.3	23.61±2.07	1.02	23.41 ± 2.10
heart	0.36	21.84±3.70	44.9	15.95 ± 3.26	0.10	16.62±3.20
image	12.10	4.19±0.70	1796.0	2.96 ± 0.60	4.90	5.55±0.60
ringnorm	0.99	1.65 ± 0.11	223.9	1.66±0.12	0.33	2.03±0.25
splice	13.40	10.94±0.70	2523.0	10.88 ± 0.66	3.87	11.11±0.66
thyroid	0.13	4.01 ± 2.18	17.9	4.80±2.19	0.06	4.4±2.40
titanic	0.09	23.04±1.18	18.4	22.42 ± 1.02	0.05	22.87±1.12
townorm	0.84	3.07±0.246	232.1	2.96±0.23	0.29	2.69 ± 0.19
waveform	1.43	11.10±0.50	247.7	9.88 ± 0.43	0.36	10±0.39
mean	3.083	16.697±1.553	531.868	14.665± 1.502	0.998	14.605±1.497

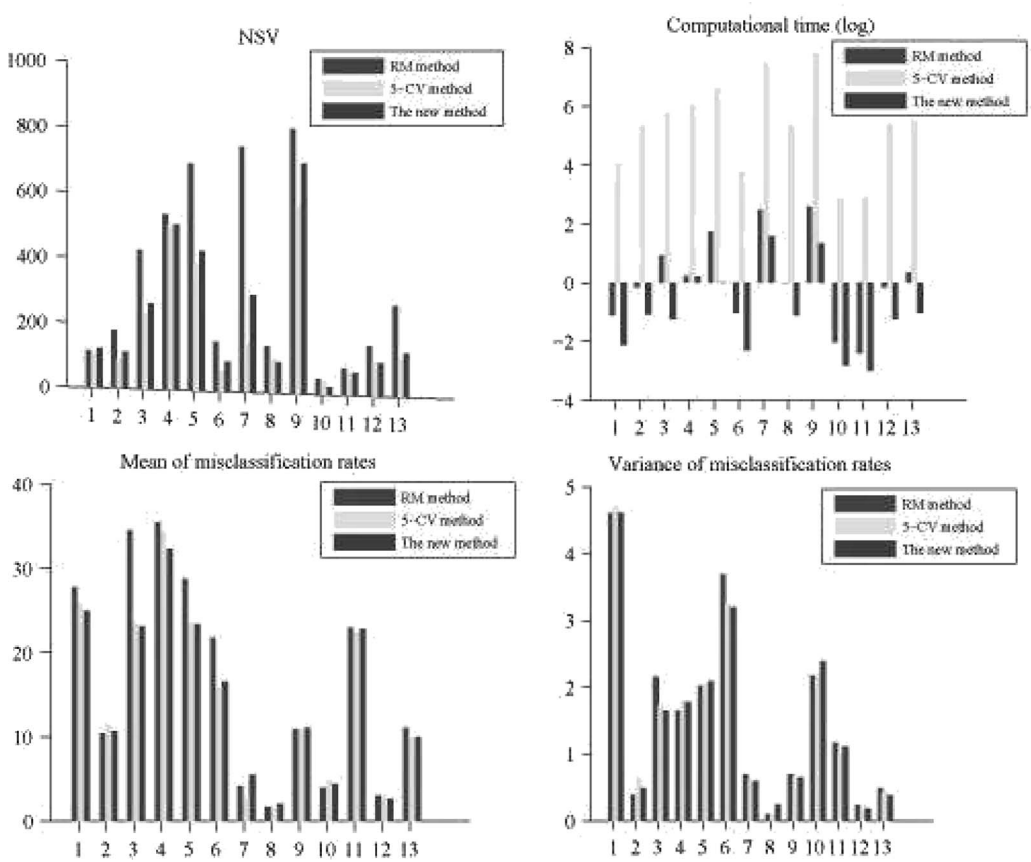


Fig. 4. Output results consulted by the RM method, the 5-CV method, and the new method on 13 benchmark data sets.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

TABLE V
INFORMATION OF THE THREE REAL-WORLD DATA SETS

name	$Ntrain$	$Ntest$	Dim
Adult	32561	16281	123
Web	49749	14951	300
IJCNN	49990	91701	22

B. Experiments With Real-World Problems

In this section, we further testify the effectiveness and efficiency of the new method when applied to larger-scale real-world problems. Three such real-world data sets, i.e., Adult, Web, and International Joint Conference on Neural Networks (IJCNN), were simulated. Herein, Adult data were extracted from the census bureau database at http://www.census.gov/ftp/pub/DES/www/welcome.html, Web data were generated by judging whether a web page data set belongs to a category or not, and each input consists of 300 sparse binary keyword attributes extracted from each web page (http://www.research.microsoft.com/jplatt/web. zip), and IJCNN data were formed from the first problem of IJCNN challenge 2001 [44] by preprocessing the raw data using the winner's transformation, which was downloaded from http://www.csie.ntu.edu.tw/cjlin/libsvmtools/ data sets. Each of these three real-world data sets was composed of one training set and one testing set. Related information on the magnitude and dimensionality of the data sets is shown in Table V. We can see that the size of every data set in this case exceeds 30 000.

In this case, the 5-CV method cannot be utilized due to its extremely huge computation burden. Only the RM method and our method are then simulated and compared in this series of experiments. We still evaluated the performance of the methods in terms of the average computation time (efficiency measure) and the MR (generalization measure) of the corresponding SVM scheme. The generalization measures were collected for each fixed data set in the following way: first, applying the RM method and the new model selection method to the data set, yielding the parameters σ and C, then adopting the obtained parameters (namely, the SVM scheme with the chosen model parameter) to train SVM on the training set, yielding an SVM classifier, and finally calculating the MR of the classifier on the testing set. The efficiency measure was simply taken as the computation time used for the parameter selection run. The experiment results are given in Tables VI and VII.

From Tables VI and VII, it is seen that the new method significantly outperforms the RM method. To be more specific, we can observe the following from the tables: 1) For each data set, the NSVs found by the SVM scheme deduced from the new method is much less than (almost half of) that from the RM method. This shows the notable simplicity of the classifier defined by the new method, and so, it tends to have faster computation speed and better generalization performance. 2) The computational speed of the new method is around 20 times faster than that of the RM method, which shows the high efficiency of the new method. 3) The average MR of the SVM deduced from the new method is also smaller than that from the RM method (about 16% better), which testifies the better prediction capability of the new method.

TABLE VI
MODEL PARAMETERS AND NUMBERS OF SVS CONDUCTED BY THE RM METHOD AND THE NEW METHOD ON THE LARGE-SCALE DATA SETS

	RM method			New method		
	σ	C	NSV	σ	C	NSV
Adult	2.219	0.0673	12545	1.809	0.946	12351
Web	1.940	0.325	5590	3.789	1.520	2951
Icjnn	0.132	7.862	17490	0.868	47.311	2505
mean	1.430	2.751	11875	2.155	16.592	5936

TABLE VII
COMPUTATIONAL COSTS AND MRS CONDUCTED BY THE RM METHOD AND THE NEW METHOD ON THE LARGE-SCALE DATA SETS

	RM method		New method	
	Time(s)	MR(%)	Time(s)	MR(%)
Adult	4317.1	15.85	639.24	14.95
Web	8032.6	1.57	194.65	1.34
Ijcnn	15328.9	2.91	501.52	1.21
mean	9225.7	6.78	445.137	5.83

All these experiments support the feasibility, effectiveness, and high efficiency of the new strategies.

VI. DISCUSSION AND CONCLUDING REMARKS

We conclude by discussing the stability of our proposed strategy for selecting the kernel parameter, comparing the strategy with the two latest model selection methods for SVM, presenting a short summary of the whole paper, and mentioning some open problems for future investigation.

A. Stability and Parameter Tuning of Algorithm 1

Note that the proposed model selection method needs to first calculate the optimal Gaussian scale parameter $\sigma_{\rm opt}$ via Algorithm 1 and then, based on $\sigma_{\rm opt}$, compute the optimal penalty coefficient $C_{\rm opt}$. Therefore, the capability of Algorithm 1 has a significant effect on the whole model selection strategy and, further, on the final SVM implementation. Since there are two parameters, i.e., α and N, involved in the algorithm, it is necessary to discuss how the stability of the algorithm is dependent on these two parameters and how to preset suitable values for them.

In the above issue, by adopting similar 13 benchmark data sets (only the first partition is considered for each data set) utilized in Section V, three series of experiments have been designed. The first series changes α from 0 to 0.2 with fixed $N = 0.5l$, where l is the number of the whole training set. The second changes N from $0.5l$ to l with fixed $a = 0.1$. The aim

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

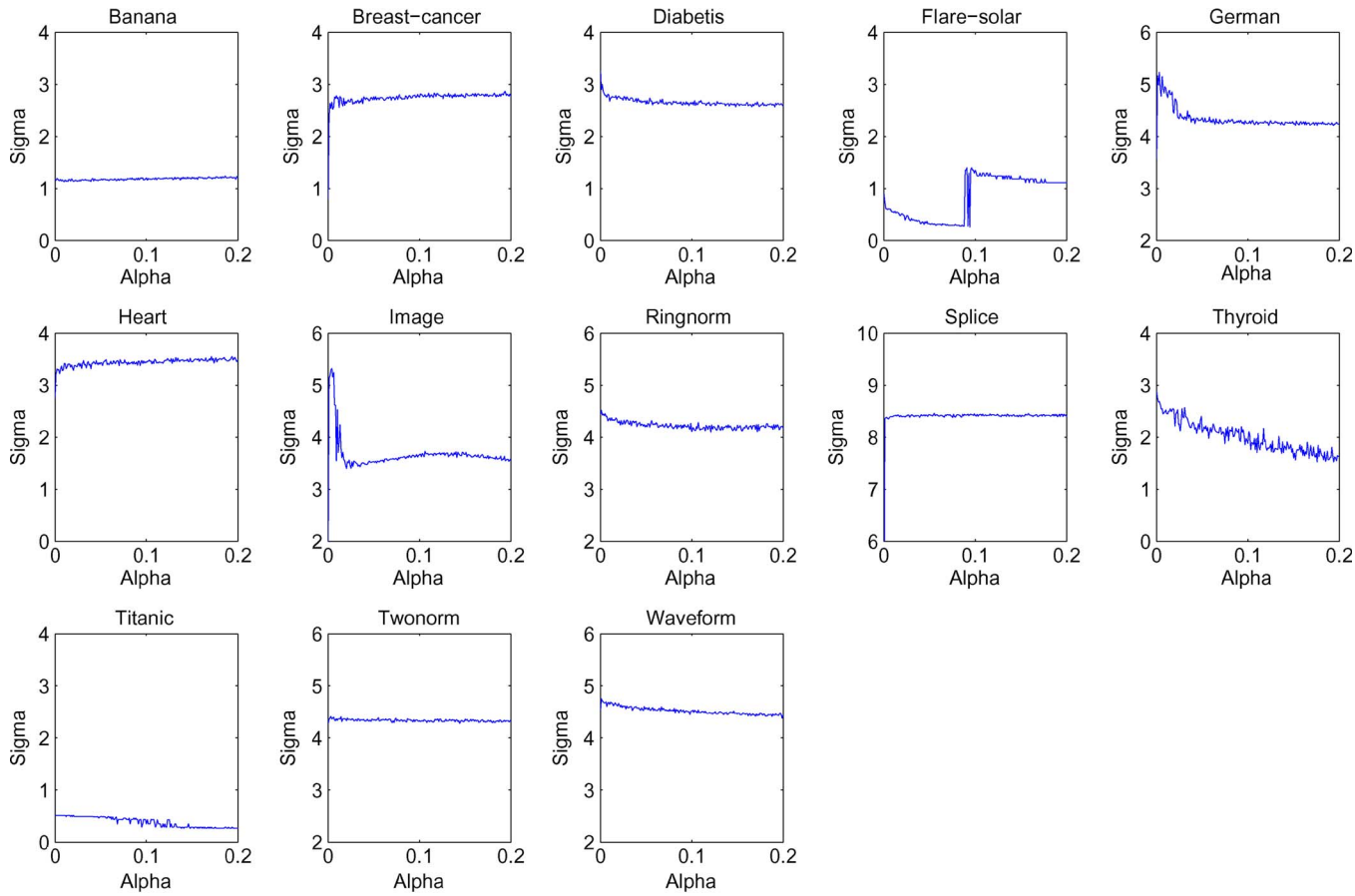


Fig. 5. Evolution of the optimal scale curves obtained from Algorithm 1 as α varies from 0 to 0.2, with fixed $N = 0.5l$ (l is the number of the corresponding training set) when applied to 13 benchmark data sets.

of these two series of experiments is to testify the algorithm's stability related to α and N, respectively. The third lets the algorithm run 100 times with fixed $a = 0.1$ and $N = 0.5l$ to testify the stability of the algorithm related to the randomly selected N samples. The results are demonstrated in Figs. 5–7, respectively. For uniformity, we set all figures in the window boxes with similar height of 4.

Two distinctive features can clearly be observed from Fig. 5. First, when α is set too small, i.e., too adjacent to 0, the algorithm perform unstably on some data sets, such as the Breast-cancer, Flare-solar, German, and Image data sets. This is duo to the fact that in such extreme case the algorithm is more likely to be negatively affected by the possible existence of outliers or noises in the training data, as mentioned in Section III. The second observation is, when α varies from 0.1 to 0.2, the algorithm has a very stable performance. Particularly, the variances of the output scale tendency curves in 12 experiments are less than 0.1 (except that corresponding to Thyroid data, where the variance is less than 0.5). This shows that we only need to preset the parameter α as an arbitrary value in the interval [0.1, 0.2], and then the stability of Algorithm 1 can empirically be verified.

The stability of Algorithm 1 that is related to size N in selecting the optimal scale parameter can easily be observed in Fig. 6. Specifically, the variances of the 13 output scale curves from the algorithm on all of the training data sets are all less than 0.1. This shows that the parameter N can be set as any value between $0.5l$ and l to promise a stable performance of the algorithm.

Fig. 7 further depicts the stable performance of Algorithm 1 when it is running 100 times with fixed parameters α and N. In particular, all of the 13 output scale curves figure as a constant line with little deviations. This shows that the stability of the algorithm is not affected by the randomly selected sample set with number N. This means that if the same experiment (with similar α and N) is run multiple times, then there will not be a vast difference in results, i.e., the stable optimal scale can still be obtained.

The above experimental results verify the stability of Algorithm 1 that is related to α and N and the randomly selected sample set with size N. This also confirms the reasonability of the adopted parameters ($\alpha = 0.1$ and $N = 0.5l$ or l) in all our experiments in Section V.

B. Links With Two Latest Model Selection Methods

In the recent three to four years, several novel parameter selection methods for SVM have arisen. The most typical ones are the regularization path method [45], [46] and the bilevel programming method [47]–[49]. In the following, we will analyze the relations between these two methods and our method.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

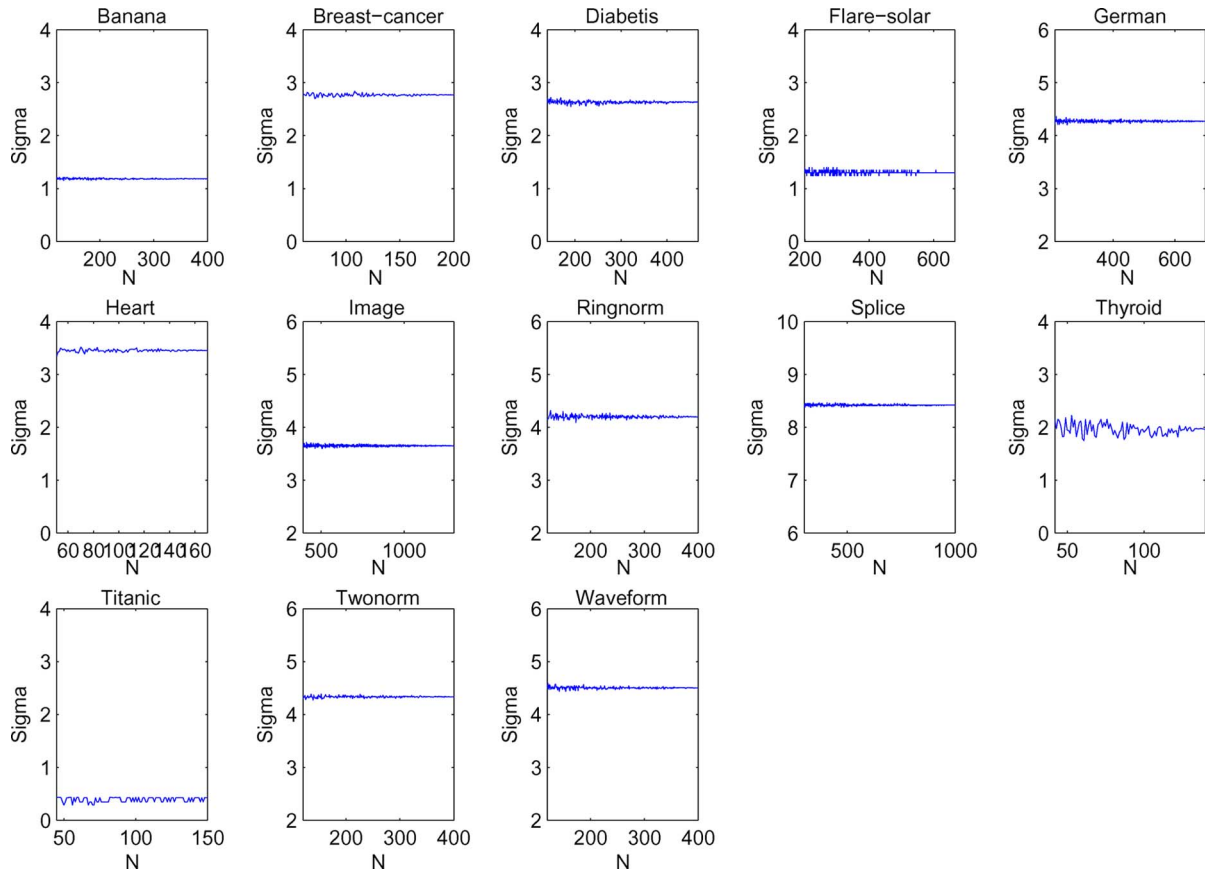


Fig. 6. Evolution of the optimal scale curves obtained from Algorithm 1 as N varies from $0.5l$ to l, with fixed $\alpha = 0.1$ when applied to 13 benchmark data sets.

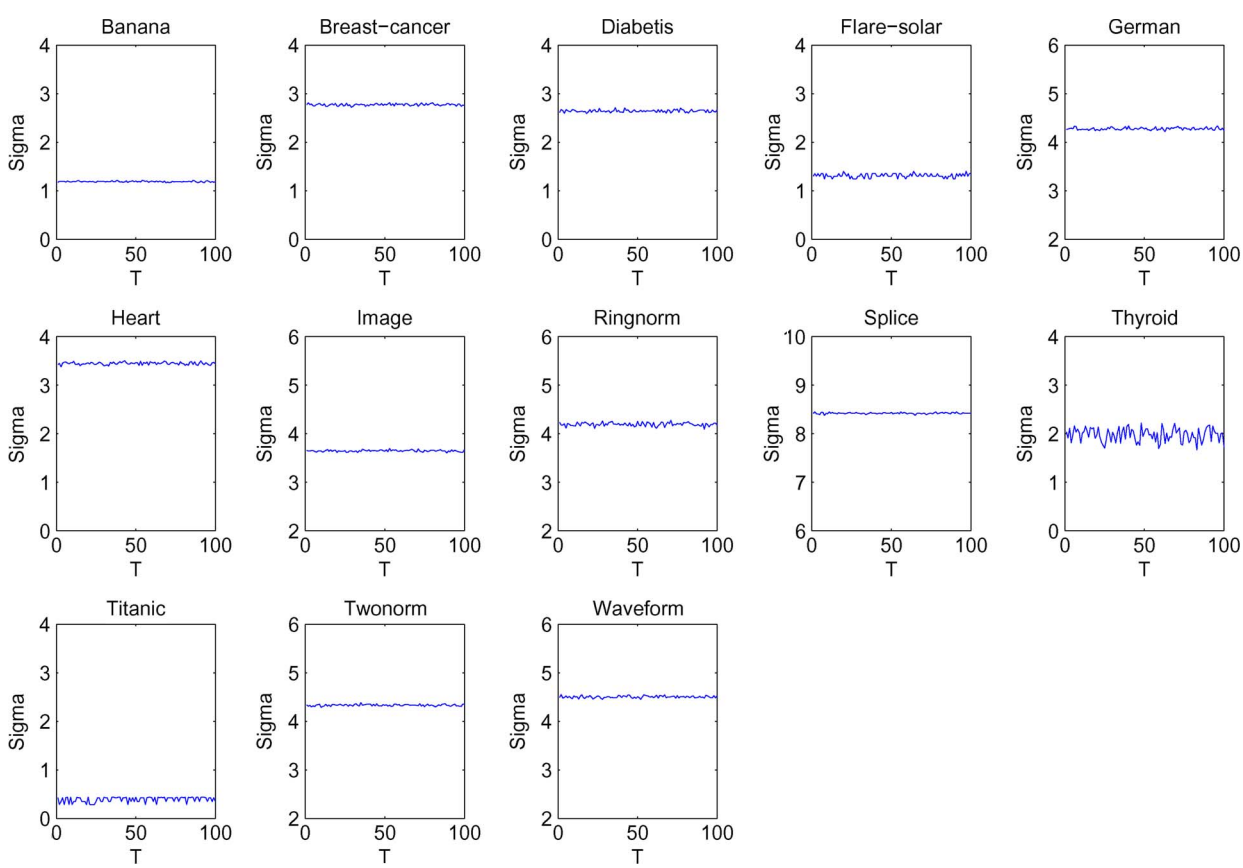


Fig. 7. Optimal scale curves obtained by running Algorithm 1 100 times on 13 benchmark data sets with fixed $\alpha = 0.1$ and $N = 0.5l$.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

The regularization path method is an iterative method specifically proposed to select the optimal penalty coefficient C. The method is constructed based on the principle that the outputs of SVM (such as the Lagrangian variables) are piecewise linear in C, and, hence, by reasonably calculating a breakpoint sequence of C with increasing order from small to large, the entire solution path of SVM with respect to C corresponding to each output variable can be simulated via linear interpolation on the outputs obtained by sequentially running SVM on these breakpoints. Through setting the rational termination condition, the optimal C can be obtained on the iterative process of the method.

Intrinsically, there are many similarities between the proposed method (Algorithm 2) and the regularization path method in selecting C. First, both are iterative methods, and both base on the outputs in the current step to calculate the value of C in the next step in an increasing order. Second, both only take into account the SVs while totally ignoring other samples under the current model parameters to calculate the related results in the iterative process. Third, both embed the model selection procedures into the SVM training process, which means that on the termination of the method, the results of SVM with respect to the corresponding optimal C are simultaneously obtained. Fourth, both set the termination condition as that there are little difference between the next step and the current one.

However, there are also essential differences between the two methods. In each step of the proposed method, the classification degrees of all current SVs are quantitatively evaluated to calculate C in the next step, and by integrating such global information, the parameter is updated in a comparatively fast speed. However, for the regularization path method, as C increases so that a nonbound SV changes to a bound SV, or a non-SV or bound SV to a nonbound SV, it is updated, and, hence, the number of the iterative steps are much larger. Empirically speaking, the latter method generally iterates much more times than the former, particularly for large data sets. Even for the large-scale real-world experiments proposed in Section V-B, no more than ten iterations are implemented by utilizing our method. More experiments and evaluations are needed in our future research to further quantitatively entertain such analysis result. In addition, utilizing the regularization path of the SVM to improve the termination conditions of the algorithm should also be examined in further research.

By integrating the general SVM model and the CV principle, the bilevel programming method initiates a bilevel optimization problem, which means another optimization programming exists in the constraints of the original problem. By virtue of KKT conditions and other mathematical skills, such bilevel programming can further be transformed into a uniform optimization problem with respect to the penalty coefficient and the Lagrangian variables. Then, by solving this optimization, the optimal penalty parameter and the SVM result can both be obtained. However, the presence of complementarity constraints in the model is a major theoretical and computational challenge, as the principles of nonlinear programming theory cannot directly be extended to the current model. That is to say, model selection for a nonlinear SVM, including Gaussian SVM, cannot be implemented by virtue of such method. As mentioned in [49], a major outstanding open question is the development of efficient algorithms for bilevel programs, particularly for a nonlinear SVM.

If we first utilize Algorithm 1 in Section III to select σ_{opt} and substitute it into the nonlinear SVM problem, then the model selection problem of SVM can be transformed to the optimization issue that only involves the penalty coefficient C. Then, through a similar mathematical deduction of linear bilevel programming, the uniform optimization problem with respect to the penalty coefficient and the Lagrangian variables can also be obtained, and, hence, the optimal C_{opt} and the corresponding Lagrangian variables can be calculated by solving this optimization problem. This idea will let the bilevel programming method be feasible for the model selection of nonlinear SVM and will be pursued in further research.

C. Summary and Ongoing Work

Two heuristic strategies to select the parameters, i.e., the kernel parameter σ and the penalty coefficient C, of Gaussian SVM have been suggested in this paper. Based on viewing the model parameter selection problem as a recognition problem in a visual system, we have developed a simple direct parameter setting formula for the kernel parameter σ. The philosophy behind the formula is to find a visual scale with which the global and local structures of a data set can be preserved in the feature space, and the difference between the two structures (the visual effect) will be maximized. We have analyzed the SMO procedure, which is a well-developed and commonly used algorithm nowadays in SVM training. Through constructing classification extents of the training data in the process of training, we have developed a heuristic for updating the penalty coefficient C from a very small value to an appropriate one. The proposed new strategies have been evaluated with a series of experiments on 13 standard benchmark problems and three real-world data sets, as compared with the well-known 5-CV heuristic and the recently developed RM method. The experiments show that in terms of efficiency and generalization capability, the new strategies outperform the current methods, and the performance is very uniform and stable. In particular, we can conclude from the experiments that the new strategies are capable of yielding the SVM classifier with higher generalization capability and fewer NSVs within a significantly less time. Hence, the suggested new strategies can be accepted as an efficient reliable model selection method for Gaussian SVM.

However, certain limitations of the method should be kept in mind. Currently, the proposed model selection strategy is only suitable for SVM with Gaussian kernel, and yet it still cannot directly be utilized on those with other kernels, such as the polynomial kernel

$$k(x, y) = (x \cdot y + 1)^d$$

and the sigmoid kernel

$$k(x, y) = \tanh(\kappa x \cdot y + c).$$

The main reason is that these kernels are not of the structure-preserving property as the Gaussian kernel, i.e., it cannot

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

preserve the ranking order of the distances between data pairs in the original and feature spaces.[1] That is to say, it is not easy to deduct the global and local structure measures in such kernel feature spaces as in a Gaussian one, and, hence, the suggested kernel selection strategy cannot similarly be constructed for these kernels. Therefore, we need to further develop specific model selection strategies for SVMs with non-Gaussian kernels in future research. In addition, although it has been verified that the proposed strategy significantly improves the computational time of the current model selection methods for SVM (this point has been proved by applying paired t-test [50] to the experimental results in this paper), to what extent the new method improves the classification accuracy of other methods still needs further investigation. Furthermore, the interior point method has outperformed SMO on some large-sized data sets in solving SVMs [51], and it is very meaningful to develop the model selection strategies based on interior point methods instead of the SMO algorithm, particularly for large-scale applications. Other problems include developing the theoretical basis of model selection, devising a more robust termination criterion for the proposed algorithms, extending the proposed strategy to the fuzzy hypersphere SVM (FHS-SVM), and comparing more extensively all the known model selection techniques. Future work will be on the research of these cases and applications of the proposed method in more practical areas.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers, whose comments helped to give great improvement to this paper.

REFERENCES

[1] V. N. Vapnik, *Estimation of Dependences Based on Empirical Data*. Moscow, Russia: Nauka, 1979.
[2] B. Schökopf, C. Burges, and A. Smola, Eds., "Introduction to support vector learning," in *Advances in Kernel Methods-Support Vector Learning*. Cambridge, MA: MIT Press, 1999, pp. 1–15.
[3] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[4] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[5] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
[6] C. Cortes and V. N. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[7] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1993.
[8] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.
[9] M. Schmidt, "Identifying speakers with support vector networks," in *Proc. 28th Symp. Interface*, Sydney, Australia, 1996.
[10] E. Osuna, R. Freund, and F. Girosi, "Support vector machines: Training and applications," MIT, Cambridge, MA, Tech. Rep. AIM-1602, 1997.
[11] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. IEEE Comput. Soc. Conf. CVPR*, 1997, pp. 130–136.

[1]For example, set four data in R^2 space as $x_1 = (1, 1)$, $y_1 = (0, 0)$, $x_2 = (1.1, 0)$, $y_2 = (0, 1.1)$, let $d = 2$ in the polynomial kernel and $\kappa = 1$, $c = -1$ in the sigmoid kernel. Then, it is easy to calculate that the distances between x_1 and y_1 as well as x_2 and y_2 in the original R^2 space are $\sqrt{2}$ and $\sqrt{2.2}$, respectively. Correspondingly, in the feature space induced from the polynomial kernel, their distances are 9 and 8.794, and in that from sigmoid kernel, they are 1.523 and 1.176, respectively. That is, in both feature spaces, the ranking order of the distances in the original space cannot be preserved.

[12] H. Drucker, D. Wu, and V. N. Vapnik, "Support vector machines for spam categorization," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1048–1054, Sep. 1999.
[13] K. Duan, S. Sathiya Keerthi, and A. N. Poo, "Evaluation of simple performance measures for tuning SVM hyperparameters," *Neurocomputing*, vol. 51, pp. 41–59, Apr. 2003.
[14] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. R. Stat. Soc., B*, vol. 36, no. 1, pp. 111–147, 1974.
[15] A. Elisseeff and M. Pontil, "Leave-one-out error and stability of learning algorithms with applications," in *Learning Theory and Practice*. Washington, DC: IOS Press, 2002.
[16] T. Joachims, "The maximum-margin approach to learning text classifiers: Method, theory and algorithms," Ph.D. dissertation, Dept. Comput. Sci., Univ. Dortmund, Dortmund, Germany, 2000.
[17] V. N. Vapnik and O. Chapelle, "Bounds on error expectation for support vector machine," in *Advances in Large Margin ClassiBers*. Cambridge, MA: MIT Press, 1999.
[18] G. Wahba, Y. Lin, and H. Zhang, "GACV for support vector machines," in *Advances in Large Margin ClassiBers*. Cambridge, MA: MIT Press, 1999.
[19] O. Chapelle, V. N. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple kernel parameters for support vector machines," *Mach. Learn.*, vol. 46, no. 1–3, pp. 131–159, Jan. 2002.
[20] S. Sathiya Keerthi, "Efficient tuning of SVM hyperparameters using radius/margin bound and iterative algorithms," *IEEE Trans. Neural Netw.*, vol. 13, no. 5, pp. 1225–1229, Sep. 2002.
[21] K. M. Chung, W. C. Kao, C. L. Sun, L. L. Wang, and C. J. Lin, "Radius margin bounds for support vector machines with the RBF kernel," *Neural Comput.*, vol. 15, no. 11, pp. 2643–2681, Nov. 2003.
[22] W. J. Wang, Z. B. Xu, W. Z. Lu, and X. Y. Zhang, "Determination of the spread parameter in the Gaussian kernel for classification and regression," *Neurocomputing*, vol. 55, no. 3, pp. 643–663, Oct. 2003.
[23] N. E. Ayat, M. Cheriet, and C. Y. Suen, "Optimization of the SVM kernels using an empirical error minimization scheme," in *Pattern Recognition with Support Vector Machines*, vol. 2388, S. W. Lee and A. Verri, Eds. Berlin, Germany: Springer-Verlag, 2002, pp. 354–369.
[24] N. Cristianini, C. Campbell, and J. Shawe-Taylor, "Dynamically adapting kernels in support vector machines," in *Advances in Neural Information Processing Systems*, vol. 11, M. S. Kearns, S. A. Solla, and D. A. Cohn, Eds. Cambridge, MA: MIT Press, 1999, pp. 204–210.
[25] O. Chapelle and V.N. Vapnik, "Model selection for support vector machines," in *Proc. 12th Conf. Adv. Neural Inf. Process. Syst.*, S. Solla, T. Leen, and K.-R. Müller, Eds. Cambridge, MA: MIT Press, 2000, vol. 12, pp. 230–236.
[26] C. L. Huang and J. F. Dun, "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1381–1391, Sep. 2008.
[27] C. Y. Sun and D. C. Gong, "Support vector machines with PSO algorithm for short-term load forecasting," in *Proc. IEEE ICNSC*, 2006, pp. 676–680.
[28] X. Li, S. D. Yang, and J. X. Qi, "A new support vector machine optimized by improved particle swarm optimization and its application," *J. Central South Univ. Technol.*, vol. 13, no. 5, pp. 568–572, Oct. 2006.
[29] S. W. Lin, Z. J. Lee, S. C. Chen, and T. Y. Tseng, "Parameter determination of support vector machine and feature selection using simulated annealing approach," *Appl. Soft Comput.*, vol. 8, no. 4, pp. 1505–1512, Sep. 2008.
[30] S. Liu, C. Y. Jia, and H. Ma, "A new weighted support vector machine with GA-based parameter selection," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2005, vol. 7, pp. 4351–4355.
[31] C. L. Huang and C. J. Wang, "A GA-based feature selection and parameters optimization for support vector machines," *Expert Syst. Appl.*, vol. 31, no. 2, pp. 231–240, 2006.
[32] W. C. Hong, "Determining parameters of support vector machines by genetic algorithms—Applications to reliability prediction," *Int. J. Oper. Res.*, vol. 2, no. 1, pp. 1–7, 2005.
[33] M. Debruyne, M. Hubert, and J. Suykens, "Model selection for kernel regression using the influence function," *J. Mach. Learn. Res.*, vol. 9, pp. 2377–2400, Oct. 2008.
[34] M. Debruyne, "Robustness of censored depth quantiles, PCA and kernel based regression, with new tools for model selection," Ph.D. dissertation, Katholieke Universiteit Leuven, Leuven, Belgium, 2007. 91 p.
[35] A. Smola, "Learning with kernels," Ph.D. dissertation, GMD First, Berlin, Germany, 1998.
[36] S. Sathiya Keerthi and C.J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Comput.*, vol. 15, no. 7, pp. 1667–1689, Jul. 2003.

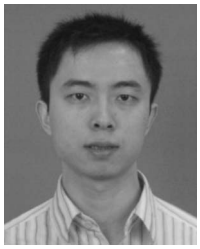
Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.

[37] Y. Leung, J. S. Zhang, and Z. B. Xu, "Clustering by scale-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1396–1410, Dec. 2000.
[38] J. C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," Microsoft Res., Redmond, WA, Tech. Rep. MSR-TR-98-14, 1998.
[39] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
[40] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods-Support Vector Learning*, B. Schökopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 11.
[41] G. Rätsch, *Benchmark Data Sets*, 1999. [Online]. Available: http://ida.first.fhg.de/projects/bench/benchmarks.htm
[42] G. Räatsch, T. Onoda, and K. R. Muller, "Soft margins for AdaBoost," *Mach. Learn.*, vol. 42, no. 3, pp. 287–320, Mar. 2001.
[43] S. Mika, G. Räatsch, J. Weston, B. Scholkopf, and K. R. Muller, "Fisher discriminant analysis with kernels," in *Proc. Neural Netw. Signal Process. IX*, H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds., 1999, pp. 41–48.
[44] C. C. Chang and C. J. Lin, "IJCNN 2001 challenge: Generalization ability and text decoding," in *Proc. IEEE IJCNN*, 2001, pp. 1031–1036.
[45] T. Hastie, S. Rosset, R. Tibishirani, and J. Zhu, "The entire regularization path for the support vector machine," *J. Mach. Learn. Res.*, vol. 5, pp. 1391–1415, Dec. 2004.
[46] M. Y. Park and T. Hastie, "L_1 regularization path algorithm for generalized linear models," *J. R. Stat. Soc. B*, vol. 69, no. 4, pp. 659–677, Sep. 2007.
[47] G. Kunapuli, K. Bennett, J. Hu, and J. S. Pang, "Classification model selection via bilevel programming," *Optim. Methods Softw.*, vol. 23, no. 4, pp. 475–489, Aug. 2008.
[48] K. Bennett, J. G. Hu, X. Y. Ji, G. Kunapuli, and J. S. Pang, "Model selection via bilevel optimization," in *Proc. Int. Joint Conf. Neural Netw.*, 2006, pp. 1922–1929.
[49] G. Kunapuli, K. Bennett, J. Hu, and J. S. Pang, "Bilevel model selection for support vector machines," in *Proc. CRM, Lecture Notes*, 2008, vol. 45, pp. 129–158.
[50] C. H. Goulden, *Methods of Statistical Analysis*, 2nd ed. New York: Wiley, 1956, pp. 50–55.
[51] M. C. Ferris and T. S. Munson, "Interior-point methods for massive support vector machines," *SIAM J. Optim.*, vol. 13, no. 3, pp. 783–804, 2002.

Zongben Xu received the M.S. degree in mathematics and the Ph.D. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 1981 and 1987, respectively.

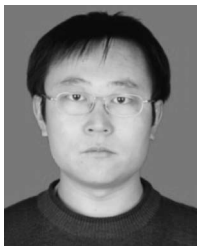
In 1989, he was a Postdoctoral Researcher with the Department of Mathematics, the University of Strathclyde, Glascow, U.K. Since 1982, he has been with the Institute for Information and System Sciences, Faculty of Science, Xi'an Jiaotong University, where he was promoted to Associate Professor in 1987 and a Full Professor in 1991 and is currently a Professor of mathematics and computer science. In 2007, he was appointed as a Chief Scientist of the National Basic Research Program of China (973 Project). He is the author of more than 150 academic papers on nonlinear functional analysis, optimization techniques, neural networks, evolutionary computation, and data mining algorithms, most of which are in international journals. His current research interests include nonlinear analysis, machine learning, and computational intelligence.

Dr. Xu received the title "Owner of Chinese Ph.D. Degree Having Outstanding Achievements" from the Chinese State Education Commission and the Academic Degree Commission of the Chinese Council in 1991. He received the National Natural Science Award of China in 2007 and the CSIAM Su Buchin Applied Mathematics Prize in 2008.



Mingwei Dai received the B.Sc. and M.Sc. degrees from Xi'an Jiaotong University, Xi'an, China, in 2003 and 2006, respectively.

He is currently with the Institute for Information and System Sciences, Faculty of Science, Xi'an Jiaotong University. His current research interests include machine learning, pattern recognition, and artificial intelligence.



Deyu Meng received the B.Sc., M.Sc., and Ph. D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively.

He is currently with the Institute for Information and System Sciences, Faculty of Science, Xi'an Jiaotong University His current research interests include machine learning, pattern recognition, artificial intelligence, manifold learning, and compressed sensing.

Authorized licensed use limited to: Xian Jiaotong University. Downloaded on December 15, 2009 at 05:02 from IEEE Xplore. Restrictions apply.