Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Neurocomputing 51 (2003) 41–59

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Evaluation of simple performance measures for tuning SVM hyperparameters

Kaibo Duan*, S. Sathiya Keerthi, Aun Neow Poo

Department of Mechanical Engineering, National University of Singapore, 10 Kent Ridge Crescent, 119260 Singapore, Singapore

Received 16 June 2001; accepted 9 March 2002

Abstract

Choosing optimal hyperparameter values for support vector machines is an important step in SVM design. This is usually done by minimizing either an estimate of generalization error or some other related performance measure. In this paper, we empirically study the usefulness of several simple performance measures that are inexpensive to compute (in the sense that they do not require expensive matrix operations involving the kernel matrix). The results point out which of these measures are adequate functionals for tuning SVM hyperparameters. For SVMs with L1 soft-margin formulation, none of the simple measures yields a performance uniformly as good as k-fold cross validation; Joachims' Xi-Alpha bound and the GACV of Wahba et al. come next and perform reasonably well. For SVMs with L2 soft-margin formulation, the radius margin bound gives a very good prediction of optimal hyperparameter values.

Keywords: SVM; Model selection; Generalization error bound

1. Introduction

Support vector machines (SVMs) [17] are extensively used as a classification tool in a variety of areas. They map the input (x) into a high-dimensional feature space ($z = \phi(x)$) and construct an optimal hyperplane defined by $w \cdot z - b = 0$ to separate examples from the two classes. For SVMs with L1 soft-margin formulation, this is

* Corresponding author.
E-mail addresses: engp9286@nus.edu.sg (K. Duan), mpessk@nus.edu.sg (S.S. Keerthi), mpepooan@nus.edu.sg (A.N. Poo).

0925-2312/03/$ - see front matter
PII: S0925-2312(02)00601-X

done by solving the primal problem

$$(\mathrm{P}) \quad \begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 + C\sum_i \xi_i \\ \text{s.t.} \quad & y_i(w \cdot z_i - b) \geqslant 1 - \xi_i, \quad \xi_i \geqslant 0 \ \forall i, \end{aligned}$$

where x_i is the ith example and y_i is the class label value which is either $+1$ or -1. (Throughout the paper, l will denote the number of examples.) This problem is computationally solved using the solution of its dual form

$$(\mathrm{D}) \quad \begin{aligned} \max \quad & \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j k(x_i, x_j) \\ \text{s.t.} \quad & 0 \leqslant \alpha_i \leqslant C \ \forall i, \quad \sum_i y_i \alpha_i = 0, \end{aligned}$$

where $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is the kernel function that performs the non-linear mapping. Popular kernel functions are

$$\text{Gaussian kernel} : k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right),$$

$$\text{Polynomial kernel} : k(x_i, x_j) = (1 + x_i \cdot x_j)^d.$$

To obtain a good performance, some parameters in SVMs have to be chosen carefully. These parameters include:

- the regularization parameter C, which determines the tradeoff between minimizing the training error and minimizing model complexity; and
- parameter (σ or d) of the kernel function that implicitly defines the nonlinear mapping from input space to some high-dimensional feature space. (*In this paper we entirely focus on the Gaussian kernel.*)

These "higher level" parameters are usually referred as hyperparameters. Tuning these hyperparameters is usually done by minimizing the estimated generalization error such as the k-fold cross-validation error or the leave-one-out (LOO) error. While k-fold cross-validation error requires the solution of several SVMs, LOO error requires the solution of many (in the order of the number of examples) SVMs. For efficiency, it is useful to have simpler estimates that, though crude, are very inexpensive to compute. After the SVM is obtained for a given set of hyperparameters, these estimates can be obtained with very little additional work. In particular, they do not require any matrix operations involving the kernel matrix. During the past few years, several such simple estimates have been proposed. The main aim of this paper is to empirically study the usefulness of these simple estimates as measures for tuning the SVM hyperparameters.

The rest of the paper is organized as follows. A brief review of the performance measures is given in Section 2. The settings of the computational experiments are described in Section 3. The experimental results are analyzed and discussed in Section 4. Finally, some concluding remarks are made in Section 5.

2. Performance measures

In this section, we briefly review the estimates (performance measures) mentioned above.

2.1. k-Fold cross-validation and LOO

Cross-validation is a popular technique for estimating generalization error and there are several versions. In *k-fold cross-validation*, the training data is randomly split into k mutually exclusive subsets (the folds) of approximately equal size. The SVM decision rule is obtained using $k-1$ of the subsets and then tested on the subset left out. This procedure is repeated k times and in this fashion each subset is used for testing once. Averaging the test error over the k trials gives an estimate of the expected generalization error.

LOO can be viewed as an extreme form of k-fold cross-validation in which k is equal to the number of examples. In LOO, one example is left out for testing each time, and so the training and testing are repeated l times. It is known [12] that the LOO procedure gives an almost unbiased estimate of the expected generalization error.

k-Fold cross-validation and LOO are applicable to arbitrary learning algorithms. In the case of SVM, it is not necessary to run the LOO procedure on all l examples and strategies are available in the literature to speed up the procedure [3,11,16]. In spite of that, for tuning SVM hyperparameters, LOO is still very expensive.

2.2. Xi-Alpha bound

In [9], Joachims developed an estimate which is an upper bound on the LOO error. This estimate can be computed using α from the solution of SVM dual problem (D) and ξ from the solution of SVM primal problem (P):

$$\mathrm{Err}_{\xi\alpha} = \frac{1}{l}\,\mathrm{card}\{i : (2\alpha_i R_\Delta^2 + \xi_i) \geqslant 1\}. \tag{1}$$

Here card denotes cardinality and R_Δ^2 is a number that satisfies $c \leqslant k(x_i, x_j) \leqslant c + R_\Delta^2$ for all x_i, x_j and some constant c. We will refer to the estimate in (1) as the *Xi-Alpha bound*.

2.3. Generalized approximate cross-validation

The generalized comparative Kullback–Liebler distance (GCKL) [19] for SVM is defined as

$$\begin{aligned}\mathrm{GCKL}(\lambda) &= E_{\mathrm{true}}\,\frac{1}{l}\sum_{i=1}^{l}(1 - y_i f_{\lambda i})_+ \\ &\equiv \frac{1}{l}\sum_{i=1}^{l}\{p_i(1 - f_{\lambda i})_+ + (1 - p_i)(1 + f_{\lambda i})_+\},\end{aligned} \tag{2}$$

where $f_\lambda(x) = w \cdot \phi(x) - b$ is the decision function, $f_{\lambda i} = f_\lambda(x_i)$, $p_i = p(x_i)$ is the conditional probability that $y_i = 1$ given x_i, and the expectation is taken over new y_i's at the observed x_i's. Here, $(\tau)_+ = \tau$ if $\tau > 0$ and 0 otherwise. λ represents all the tunable parameters (C and other parameters inside kernel function) of SVM. GCKL is seen as an upper bound on misclassification rate and it depends on the underlying distribution of the examples. However, since we do not know p_i, we cannot calculate GCKL directly.

Wahba et al. [21] developed generalized approximate cross-validation (GACV) as a computable proxy for GCKL based on training data. Choosing λ to minimize the GACV is expected to come close to minimizing the GCKL. GACV is defined as

$$\mathrm{GACV}(\lambda) = \frac{1}{l}\left[\sum_{i=1}^{l}\xi_i + 2\sum_{y_i f_{\lambda i} < -1}\alpha_i K_{ii} + \sum_{y_i f_{\lambda i} \in [-1,1]}\alpha_i K_{ii}\right], \tag{3}$$

where $K_{ij}=k(x_i,x_j)$. Note that the definition of GACV given above is in a form different from that in [21], but they are equivalent (we use a different version of SVM primary problem description). GACV can be computed directly once the SVM is trained on the whole training data. Preliminary simulations in [21] suggested that minimizer of GACV is a reasonable estimate of the minimizer of GCKL.

2.4. Approximate span bound

Vapnik et al. [18] introduced a new concept called *span* of support vectors. Based on this new concept, they developed a new technique called *span-rule* (specially for SVMs) to approximate the LOO estimate. The span-rule not only provides a good functional for SVM hyperparameter selection, but also reflects better the actual error rate. However, it is expensive to compute. We do not consider the span-rule for evaluation in this paper.

The following upper bound on LOO error was also proposed in [18]:

$$\frac{N_{\mathrm{LOO}}}{l} \leqslant \frac{S\max(D, 1/\sqrt{C})\sum_{i=1}^{n^*}\alpha_i + m}{l}, \tag{4}$$

where N_{LOO} is the number of errors in LOO procedure; $\sum_{i=1}^{n^*}\alpha_i$ is the summation of Lagrange multipliers α_i taken over support vectors of the first category (those for which $0 < \alpha_i < C$); m is the number of support vectors of the second category (those for which $\alpha_i = C$); S is the span of support vectors (see [18] for the definition of S); D is the diameter of the smallest sphere containing the training points in the feature space; and the Lagrange multipliers α_i are obtained from the training of SVM on the whole training data of size l.

Although the right-hand side bound in (4) has a simple form, it is expensive to compute the span S. The bound can be further simplified by replacing S with D_{SV}, the diameter of the smallest sphere in the feature space containing the support vectors of the first category. It was proved in [18] that $S \leqslant D_{\mathrm{SV}}$. Thus, we get

$$\frac{N_{\mathrm{LOO}}}{l} \leqslant \frac{D_{\mathrm{SV}}\max(D, 1/\sqrt{C})\sum_{i=1}^{n^*}\alpha_i + m}{l}. \tag{5}$$

The right-hand side of (4) is referred as the *span bound*. Since the bound in (5) is a looser bound than the span bound, we refer to it as the *approximate span bound*. This bound is the one that is empirically evaluated in this paper.

Remark. The span-rule based estimate in [18] is an *excellent* bound on generalization error, but expensive to compute. On the other hand, the approximate span bound in (5) is a very crude bound, but very cheap to compute.

2.5. VC bound

SVMs are based on the idea of *structural risk minimization* introduced by *statistical learning theory* [17]. For the two-class classification problem, the learning machine is actually defined by a set of functions $f(x,\alpha)$, which perform a mapping from input pattern x_i to class label $y_i \in \{-1,+1\}$. A particular choice of the adjustable parameter α gives a "trained machine". Suppose a set of training examples $(x_1, y_1),\ldots,(x_l, y_l)$ are drawn from some unknown probability distribution $P(x, y)$. Then, the expected test error for a trained machine is

$$R(\alpha) = \int \frac{1}{2}\,|y - f(x,\alpha)|\,\mathrm{d}P(x,y).$$

The quantity $R(\alpha)$ is called *expected risk*. "*Empirical risk*" is defined as the measured mean error rate on the training set:

$$R_{\mathrm{emp}} = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(x_i,\alpha)|.$$

For a particular choice of α, with probability $1-\eta$ $(0 \leqslant \eta \leqslant 1)$, the following bound holds [17]:

$$R(\alpha) \leqslant R_{\mathrm{emp}}(\alpha) + \sqrt{\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}}, \tag{6}$$

where h is the VC-dimension of a set of functions $f(x,\alpha)$ and it describes the capacity of the set of functions. The right-hand side of (6) is referred as *risk bound*. The second term of the risk bound is usually referred as the *VC confidence*.

For a given learning task, the *structural risk minimization principle* [17] chooses the parameter α so that the risk bound is minimal. The main difficulty in applying the risk bound is that it is difficult to determine the VC-dimension of the set of functions. For SVMs, a *VC bound* was proposed in [2] by approximating the VC-dimension in (6) by a loose bound on it:

$$h \leqslant D^2\|w\|^2 + 1. \tag{7}$$

The right-hand side of (7) is a loose bound on VC-dimension and, if we use this bound to approximate h, sometimes we may get into a situation where $1/h$ is so small that the term inside the square root in (6) may become negative. To avoid this problem, we do the following. Since h is also bounded by $l+1$, we simply set h to $l+1$ whenever $D^2\|w|^2 + 1$ exceeds $l+1$.

2.6. Radius-margin bound

For SVMs with hard-margin formulation, it was shown by Vapnik et al. [18] that the following bound holds:

$$\text{LOO Err} \leqslant \frac{1}{4l} D^2 \|w\|^2, \tag{8}$$

where w is the weight vector computed by SVM training and D is the diameter of the smallest sphere that contains all the training examples in the feature space. The right-hand side of (8) is usually referred as the *radius-margin bound.*

Remark. Chapelle [4] rightly pointed out to us that, since (8) is based on hard-margin analysis, it is inappropriate for use in tuning hyperparameters associated with the L1 soft-margin formulation. (In Section 4, we give a detailed analysis to show this.) He also suggested the following modified bound (it is based on the equation appearing before Eq. (6) of [5]):

$$\text{LOO Err} \leqslant \frac{1}{l}\left[D^2 \sum_{i=1}^{l} \alpha_i + \sum_{i=1}^{l} \xi_i\right].$$

It can be shown by equating the primal and dual objective function values that the above can be equivalently written as

$$\text{LOO Err} \leqslant \frac{1}{l}\left[D^2\|w\|^2 + (D^2C + 1)\sum_{i=1}^{l} \xi_i\right].$$

We will refer to the above bound as the *modified radius-margin bound.*

The SVM problem with L2 soft-margin formulation corresponds to replacing the term $\sum \xi_i$ in (P) with $1/2 \sum \xi_i^2$. In other words, the primal problem is

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 + \frac{C}{2}\sum_i \xi_i^2 \\ &\text{s.t.} \quad y_i(w \cdot z_i - b) \geqslant 1 - \xi_i \quad \forall i. \end{aligned}$$

The kernel function is as usual, $k(x_i, x_j) = z_i \cdot z_j$. It is easy to show [6] that this problem is equivalent to the hard-margin problem:

$$\begin{aligned} &\min \ \frac{1}{2}\|w\|^2 \\ &\text{s.t.} \quad y_i(w \cdot z_i - b) \geqslant 1 \quad \forall i, \end{aligned}$$

where $z_i \cdot z_j = k(x_i, x_j) + \delta_{ij}/C$; $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$.

Chapelle et al. [5] explored the computation of gradient of D^2 and $\|w\|^2$, and their results make these gradient computations very easy. In their experiments, they minimized radius-margin bound using gradient descent technique and the results showed that radius-margin bound could act as a good functional to tune the degree of polynomial kernel.

In this paper, we will study the usefulness of $D^2\|w\|^2$ as a functional to tune the hyperparameters of a SVM with Gaussian kernel (both L1 soft-margin formulation and L2 soft-margin formulation).

3. Computational experiments

The purpose of our experiments is to see how good the various estimates (bounds) are for tuning the hyperparameters of SVMs. In this paper, we only focus on SVMs with Gaussian kernel. For one given estimator, goodness is evaluated by comparing the true minimum of the test error with the test error at the optimal hyperparameter set found by minimizing the estimate. We did the simulations on five benchmark data sets: Banana, Image, Splice, Waveform and Tree. General information about the data sets is given in Table 1. Detailed information concerning the first four data sets can be found in [13]. The Tree data set was originally used by Bailey et al. [1] and was formed from a geological remote sensing data; it has two classes: one consists of patterns of trees, and the other consists of non-tree patterns. *Note that each of the data sets has a large number of test examples so that performance on the test set, the test error, can be taken as an accurate reflection of generalization performance.*

One experiment was set up for SVM with L1 soft-margin formulation. The simple performance measures we tested in this experiment are: 5-fold cross-validation error, Xi-Alpha bound, GACV, VC bound, approximate span bound, $D^2\|w\|^2$ (radius-margin bound) and the modified radius-margin bound.

As we mentioned in Section 2, the SVM problem with L2 soft-margin formulation can be converted to the hard-margin SVM problem with a slightly modified kernel function. For SVM hard-margin formulation, the radius-margin bound can be applied. So, we set up an experiment to see how good the radius-margin bound ($D^2\|w\|^2$) is for the L2 soft-margin formulation.

For the above two experiments, first we fix the regularization parameter C at some value and vary the width of Gaussian kernel σ^2 in a large range, and then we fix the value of σ^2 and vary the value of C. The fixed values of C and σ^2 are chosen so that the combination achieves a test error close to the smallest test error rate.

Tables 2–5 describe the performance of the various estimates. Both test error rates and the hyperparameter values at the minima of different estimates are shown there. However, we must point out that we only searched in a finite range of the hyperparameter space and hence the minima are confined to this finite range. Due to lack of space, we give detailed plots of the estimates as functions of C and σ^2, only for the Image data set (Figs. 1–4). The plots for the other data sets show similar variations with respect to the two hyperparameters. We make the plots of other data sets available at http://guppy.mpe.nus.edu.sg/~mpessk/ncfigures.pdf. In order to show the variations

Table 1
General information about the data sets

Data sets	Number of input variables	Number of training examples	Number of test examples
Banana	2	400	4900
Image	18	1300	1010
Splice	60	1000	2175
Waveform	21	400	4600
Tree	18	700	11,692

Table 2
The value of test err at the σ^2-minima of different criteria for fixed C values, for SVM L1 soft-margin formulation. The values in parentheses are the corresponding logarithms of σ^2 at the minima

Criterion	Banana $\log C = 5.20$	Image $\log C = 4.0$	Splice $\log C = 0.40$	Waveform $\log C = 1.40$	Tree $\log C = 8.60$
Test err	0.1043	0.0188	0.0947	0.1022	0.1089
	(0.60)	(1.00)	(3.40)	(3.20)	(3.80)
5-fold CV err	**0.1276**	**0.0198**	**0.0975**	0.1159	**0.1144**
	(1.30)	(1.20)	(3.20)	(4.40)	(5.0)
Xi-Alpha bound	0.1453	0.0257	0.0979	**0.1035**	0.1551
	(−2.10)	(2.00)	(3.80)	(3.0)	(1.0)
GACV	0.3520	0.0376	**0.0975**	0.1143	0.1627
	(−6.60)	(−1.20)	(3.20)	(1.40)	(−2.40)
VC bound	0.4094	0.2564	0.1766	0.3293	0.2609
	(8.90)	(10.0)	(8.40)	(10.0)	(−10.0)
Approxi span bound	0.3943	0.1436	0.1407	0.1243	0.1356
	(6.60)	(6.50)	(5.60)	(5.20)	(9.80)
$D^2\|\|w\|\|^2$	0.5594	0.2564	0.4800	0.3293	0.1627
	(10.0)	(10.0)	(10.0)	(10.0)	(−2.40)
Modified radius-margin bound	0.3520 (−6.60)	0.0376 (−1.20)	0.4800 (10.0)	0.3293 (10.0)	0.1627 (−2.40)

Table 3
The value of test err at the C-minima of different criteria for fixed σ^2 values, for SVM L1 soft-margin formulation. The values in parentheses are the corresponding logarithms of C at the minima

Criterion	Banana $\log \sigma^2 = 0.60$	Image $\log \sigma^2 = 1.0$	Splice $\log \sigma^2 = 3.40$	Waveform $\log \sigma^2 = 3.20$	Tree $\log \sigma^2 = 3.80$
Test err	0.1045	0.0178	0.0947	0.1022	0.1089
	(5.20)	(4.30)	(0.40)	(1.40)	(8.60)
5-fold CV err	**0.1278**	**0.0198**	**0.0947**	**0.1102**	0.1218
	(9.00)	(6.10)	(0.50)	(0.0)	(4.80)
Xi-Alpha bound	0.1286	**0.0198**	0.3398	0.1487	**0.1160**
	(9.30)	(6.70)	(−2.70)	(−2.80)	(9.60)
GACV	0.2449	0.0782	0.1122	0.1157	0.1415
	(−1.40)	(−1.20)	(−0.80)	(−1.20)	(−1.60)
VC bound	0.3987	0.1584	0.4800	0.3293	0.2609
	(−3.0)	(−3.6)	(−10.0)	(−10.0)	(−10.0)
Approxi span bound	0.1251	0.0535	0.1136	**0.1102**	0.1363
	(1.80)	(−0.60)	(−0.90)	(0.0)	(1.20)
$D^2\|\|w\|\|^2$	0.5594	0.2564	0.4800	0.3293	0. 2609
	(−10.0)	(−10.0)	(−10.0)	(−10.0)	(−10.0)
Modified radius-margin bound	0.3727 (−2.60)	0.1158 (−2.20)	0.1375 (−1.80)	0.1324 (−2.0)	0.2609 (−9.40)

of different estimates in one figure, normalization was done on the estimates when necessary. Since what we really concern is how the variation of the estimate relates to the variation of the test error rather than how their values are related, this normalization does no harm.

Table 4
The value of test err at the σ^2-minima of different criteria for fixed C values, for SVM L2 soft-margin formulation. The values in parentheses are the corresponding logarithms of σ^2 at the minima

Criterion	Banana $\log C = -0.90$	Image $\log c = 0.44$	Splice $\log C = 6.91$	Waveform $\log C = 0$	Tree $\log C = 9.80$
Test err	0.1118	0.0238	0.0947	0.0991	0.1049
	(−1.40)	(0.50)	(3.30)	(2.80)	(4.60)
$D^2\|w\|^2$	0.1141	0.0297	0.1002	0.1011	0.1627
	(−1.60)	(−0.30)	(3.10)	(2.20)	(−2.40)

Table 5
The value of test err at the C-minima of different criteria for fixed σ^2 values, for SVM L2 soft-margin formulation. The values in parentheses are the corresponding logarithms of C at the minima

Criterion	Banana $\log \sigma^2 = -1.39$	Image $\log \sigma^2 = -0.29$	Splice $\log \sigma^2 = 3.07$	Waveform $\log \sigma^2 = 2.80$	Tree $\log \sigma^2 = 4.60$
Test err	0.1118	0.0218	0.1007	0.0991	0.1049
	(0.0)	(2.40)	(2.20)	(0.0)	(9.80)
$D^2\|w\|^2$	0.1127	0.0297	0.1016	0.1007	0.1413
	(−0.90)	(0.40)	(9.20)	(−0.60)	(−1.40)

Another experiment was set up to see how the size of the training set affects the performance of different estimates. The Waveform data set was used in this experiment. We vary the number of training examples from 200 to 1000. For comparison purpose, for each training set of different size, we use the same test set that has 4000 examples. As in the other experiments, the performance of each estimate is evaluated by comparing the test error rates at the optimal hyperparameter set found by minimizing the estimate. Fig. 5 shows the performance of the various measures as a function of training size.

4. Analysis and discussion

Let us analyze the performance of the various estimates, one by one.

4.1. k-Fold cross-validation

In all experiments, we used $k = 5$ for the number of folds. On each data set, k-fold cross-validation produced curves (by curves we mean the variation with respect to σ^2 and C) that not only have minima very close to those of the test error curves, but also have shapes very similar to the curves of the test error. *Of all the estimates, k-fold cross-validation yielded the best performance overall.* Even for a small training set with 200 examples, k-fold cross-validation gave a quite good estimate of generalization error (see Fig. 5).

Recently, a lot of research work has been devoted to speed up the LOO procedure so that it can be used to tune the hyperparameters of SVMs. Some of those speed-up

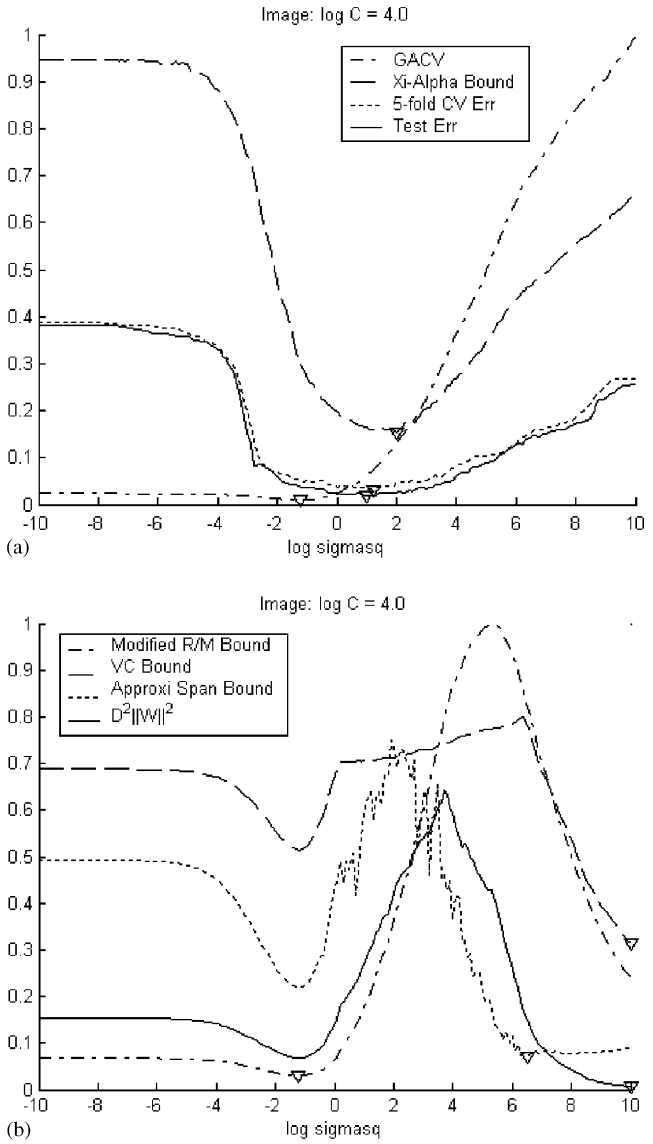

Fig. 1. Variation of GACV, Xi-Alpha bound, 5-fold CV err, test err, modified radius-margin bound, VC bound, approximate span bound, and $D^2\|w\|^2$ with respect to σ^2 for fixed C value, for SVM L1 soft-margin formulation. The vertical axis is normalized differently for GACV, VC bound, approximate span bound and $D^2\|w\|^2$. For each curve, ∇ denotes the minimum point.

strategies, such as alpha seeding [8] and loose tolerance [10,11], can be easily carried from LOO to k-fold cross-validation. Thus, k-fold cross-validation is also an efficient technique for tuning SVM hyperparameters.

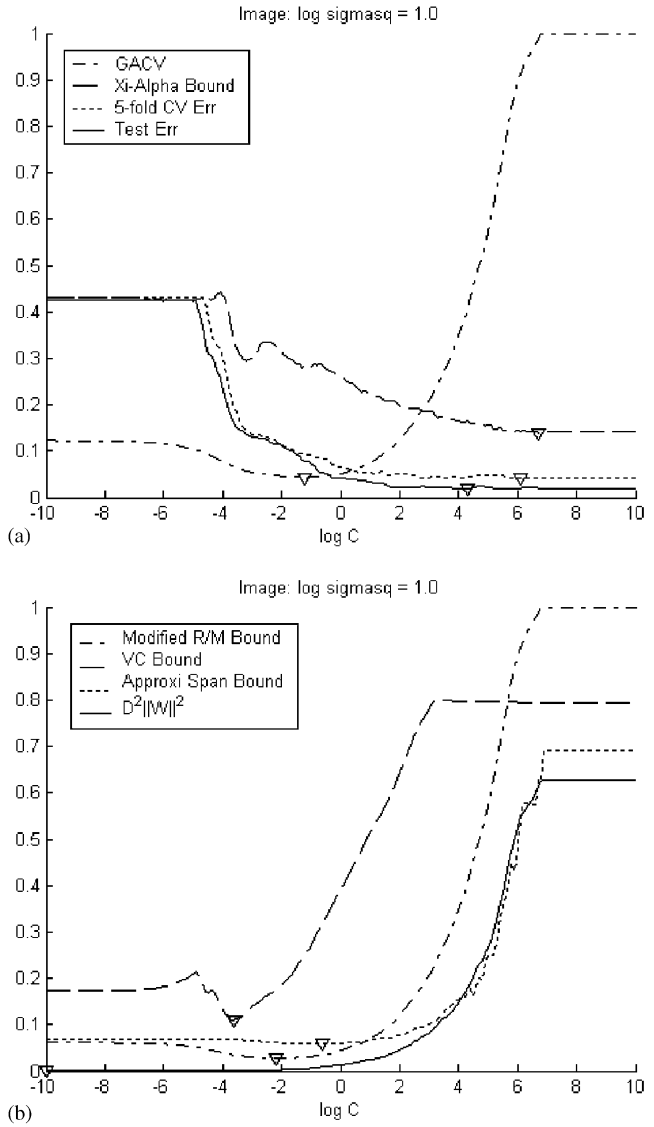

Fig. 2. Variation of GACV, Xi-Alpha bound, 5-fold CV err, test err, modified radius-margin bound, VC bound, approximate span bound, and $D^2\|w\|^2$ with respect to C for fixed σ^2 value, for SVM L1 soft-margin formulation. The vertical axis is normalized differently for GACV, VC bound, approximate span bound and $D^2\|w\|^2$. For each curve, ∇ denotes the minimum point.

4.2. Xi-Alpha bound

Xi-Alpha bound is a very simple bound, which can be computed without any extra work after the SVM is trained on the whole training data. Although it produced curves

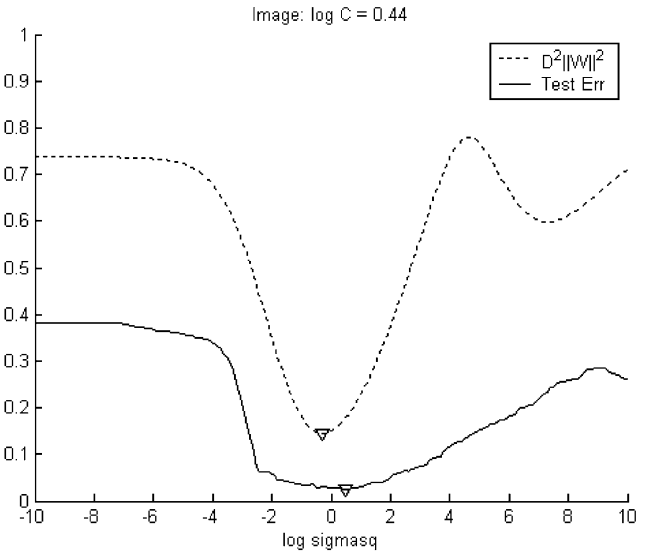

Fig. 3. Variation of $D^2\|w\|^2$ and test err with respect to σ^2 for fixed C value, for SVM L2 soft-margin formulation. The vertical axis for $D^2\|w\|^2$ is normalized. For each curve, ∇ denotes the minimum point.

that have shapes slightly different from those of the test error, in most of the cases, the predicted hyperparameters gave performance reasonably close to the best one in terms of test error.

We also notice that, at low C values, Xi-Alpha bound gives an estimate that is very close to the test error. This is because, at low C values, α_i are small and hence, the Xi-Alpha estimate in (1) is very close to the LOO estimate. However, the situation is very different when C takes large values and so the estimate differs from the test error a lot.

4.3. Generalized approximate cross-validation

For two data sets, Splice and Waveform, minimizing GACV gave a very good estimate of the optimal hyperparameters for SVM; its performance was equal to that of k-fold cross-validation and somewhat better than that of Xi-Alpha bound. For other data sets, however, GACV did not do very well. The minimum of GACV is usually located in a region where the curves are so flat that the minimum is not so apparent. For all the data sets, we noticed that the minimizer of GACV tends to be biased towards smaller C and σ. This agrees with the observation of Wahba et al. in [20].

On Waveform data set, the training-size-varying experiment (see Fig. 5) shows that GACV gives a good performance for various size of training data. For this data set, GACV also shows a better correlation with the test error than Xi-Alpha bound.

Another observation is worth mentioning. It can be seen from the curves that, compared to other estimates, such as k-fold cross-validation and Xi-Alpha bound, GACV

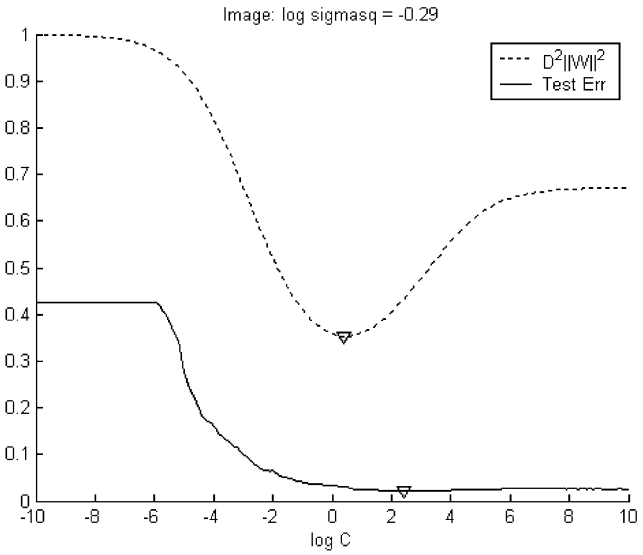

Fig. 4. Variation of $D^2\|w\|^2$ and test err with respect to C for fixed σ^2 value, for SVM L2 soft-margin formulation. The vertical axis for $D^2\|w\|^2$ is normalized. For each curve, ∇ denotes the minimum point.

has a much smoother variation with respect to the hyperparameters. This property can be useful if gradient-based techniques are to be employed for tuning the hyperparameters.

To see the correlation of the above three estimate (k-fold cross-validation estimate, Xi-Alpha bound and GACV) with test error, we tried many combinations of C and σ^2 in a very large range and generated a plot that takes the test error as one coordinate and the estimate as another coordinate. Each point on the plot corresponds to one combination of C and σ^2. The plots are shown in Fig. 6. Since we are especially interested in points at which the estimate and the test error take small values, the figure is magnified to focus only on this particular area. These plots show that k-fold cross-validation estimate has much sharper correlation with the test error.

4.4. Approximate span bound

Approximate span bound performs poorly. In [18], Vapnik et al. effectively used span-based idea for tuning SVM hyperparameters. In approximate span bound, we replaced S by D_{SV}. The poor behavior of this bound is probably due to the fact that D_{SV} is a poor approximate of S.

4.5. VC bound

The experiments show that VC bound is not good for tuning SVM hyperparameters, at least for the data sets used by us. However, for another data set, Burges [2] found

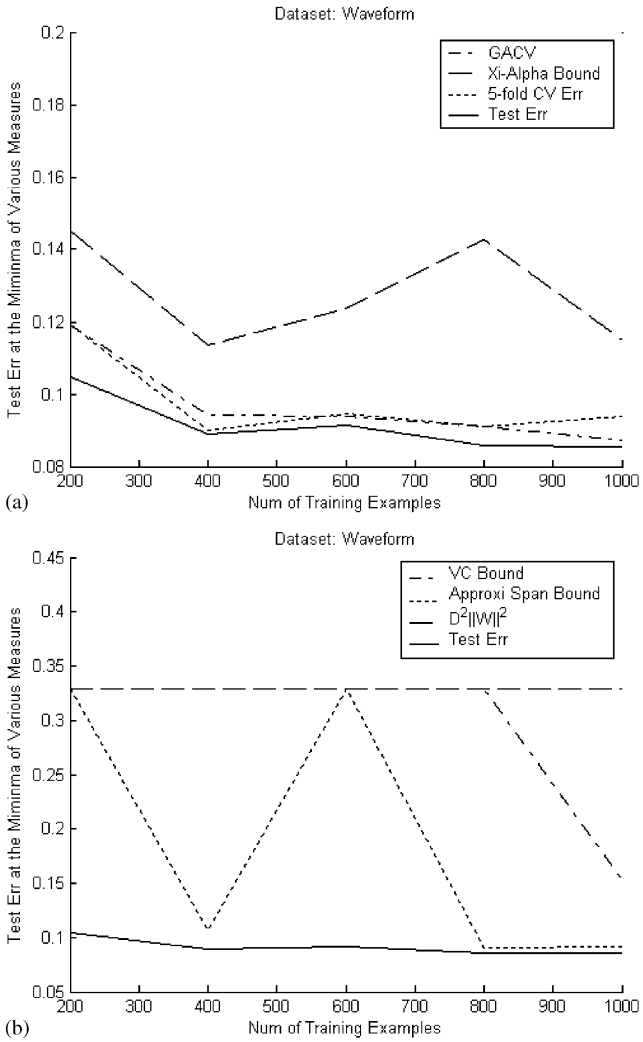

Fig. 5. Performance of various measures for different training set sizes. The waveform data set has been used in this experiment. The following values were tried for the number of training examples: 200, 400, 600, 800, and 1000. The number of the test examples is 4000.

this bound to be useful for determining a good value for σ^2. Therefore, it is not clear how useful this bound is. It is quite possible that the goodness of the VC bound depends on how well $D^2\|w\|^2 + 1$ approximates the VC dimension h.

4.6. $D^2\|w\|^2$ for L1 soft-margin formulation

Let us now consider $D^2\|w\|^2$ for L1 soft-margin formulation. Figs. 1 and 2 clearly show the inadequacy of this measure for tuning hyperparameters. The plots for the

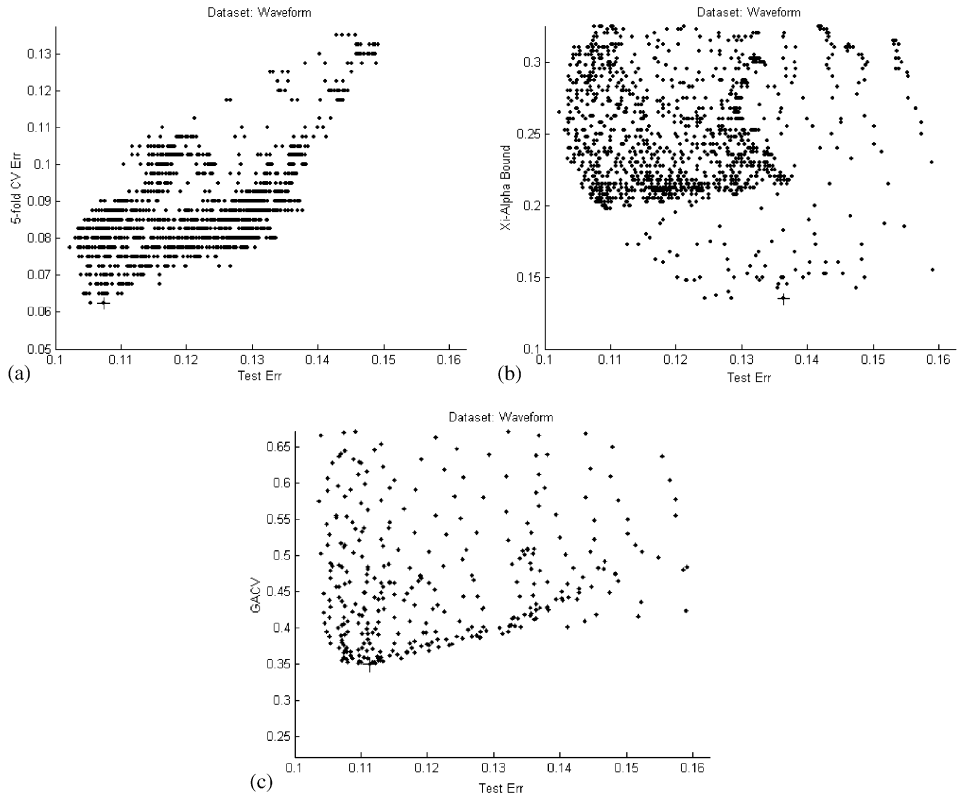

Fig. 6. Correlation of 5-fold cross-validation, Xi-Alpha bound and GACV with test error. Each point corresponds to one combination of C and σ^2. Each figure has been magnified to show only points where test error and the estimate take small values. The points with least value of the estimate are marked by $+$.

other data sets are also very similar. The inadequacy is quite obvious and can be easily explained. We can prove that, for an SVM with Gaussian kernel, $D^2\|w\|^2$ goes to zero as C goes to zero or as σ^2 goes to infinity.

First, let us fix σ^2 and consider the variation of $D^2\|w\|^2$ as C goes to zero. We have

$$\begin{aligned}\|w\|^2 &= \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j)\\ &\leqslant \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j k(x_i,x_j)\\ &\leqslant \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j\\ &\leqslant l^2C^2.\end{aligned}$$

Since D^2 is independent of C and upper-bounded by 4, it easily follows that, as C goes to zero, C goes to zero and so does $D^2\|w\|^2$.

Now let us fix C at a finite value and consider the variation of $D^2\|w\|^2$ as σ^2 goes to infinity. We have

$$D\|w\|^2 = D^2 \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$\leqslant 4 \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i, x_j).$$

As σ^2 goes to infinity, $k(x, \bar{x})$ goes to 1 and, since the alpha variables are bounded by C, we have, in the limit,

$$\sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$= \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j$$

$$= \left(\sum_{i=1}^{l} \alpha_i y_i \right)^2 = 0.$$

Thus, as σ^2 goes to infinity, $D^2\|w\|^2$ goes to zero.

Cristianini et al. in [7] showed that $D^2\|w\|^2$ is good for tuning the width of the Gaussian kernel for hard-margin SVM. The asymptotic movement of $D^2\|w\|^2$ to zero as σ^2 goes to infinity, that we established above, holds only when C is fixed at a finite value. When C is infinity (the hard margin case), the alpha variables are unbounded and hence our proof will not hold. Thus, what we have shown is not in any way inconsistent with the results in [7].

Remark. The modified radius-margin bound as suggested by Chapelle [4] performs much better than $D^2\|w\|^2$. For fixed σ^2, the variation of the bound with respect to C is very similar to that of GACV. This is understandable given the closeness of the expressions in these two bounds. The variations with respect to σ^2 (for fixed C), however, have some differences, particularly at large σ^2 values. Overall, in terms of the test set accuracies attained by tuning C and σ^2, GACV is better.

Schölkopf et al. [15] showed that $D^2\|w\|^2$ is good for tuning the degree of polynomial kernel for SVMs with L1 soft-margin formulation. Our experiments and analysis on $D^2\|w\|^2$ are only limited to SVM with Gaussian kernel. Although $D^2\|w\|^2$ is inadequate for tuning hyperparameters for SVM with Gaussian kernel, possibly it still can be used to tune the degree of polynomial kernel, as Schölkopf et al. did.

Remark. Schölkopf [14] has pointed out to us that the good performance of $D^2\|w\|^2$ on the USPS data set which was reported in their work [15] was probably due to the fact that there is little noise in that data set and hence the "soft-margin aspect" was probably not an important factor.

4.7. $D^2\|w\|^2$ for L2 soft-margin formulation

Earlier, we pointed out that $D^2\|w\|^2$ is inadequate for tuning hyperparameters for the SVM L1 soft-margin formulation with Gaussian kernel. However, For SVMs with L2 soft-margin formulation, our experiments show that radius-margin bound gave a very good estimate of the optimal hyperparameters. This agrees with the results of Chapelle et al. [5], where the radius-margin bound is chosen as the functional that is minimized using gradient descent.

However, we notice that the radius-margin bound may have more than one minimum (see Fig. 3). Typically, there is one local minimum whose value of radius-margin bound is higher than the least radius-margin bound value. This local minimum is usually located at a very large σ^2 value. Thus, minimizing the radius-margin bound using gradient descent technique, as Chapelle et al. did, can get stuck at a local minimum of the radius-margin bound. So, choosing a proper starting point for gradient descent search is important.

5. Conclusions

We have tested several easy-to-compute performance measures for SVMs with L1 soft-margin formulation and SVMs with L2 soft-margin formulation. The conclusions are:

- k-fold cross-validation gives an excellent estimate of the generalization error. For the L1 soft-margin SVM formulation, none of the other measures yields a performance as good as k-fold cross-validation. It even gives a good estimate on small training set. The k-fold cross-validation estimate also has a very good correlation with the test error.
- Xi-Alpha bound can find a reasonably good hyperparameter set for SVM, at which the test error is close to the true minimum of the test error. But the hyperparameters sometimes may not be close to the optimal ones. A nice property of this estimate is that it performs well over a range of training set sizes.
- Compared with k-fold cross-validation and Xi-Alpha bound, GACV has a smoother variation with respect to the hyperparameters. On Waveform and Splice data sets, GACV shows better correlation with test error than Xi-Alpha bound. However, the performance of GACV is worse on the other data sets.
- The approximate span bound and VC bound cannot give a useful prediction of the optimal hyperparameters. This is probably because the approximations introduced into these bounds are too loose.

- For the SVM L1 soft-margin formulation, $D^2\|w\|^2$ is inadequate for tuning the hyperparameters. The modified radius-margin bound performs much better than $D^2\|w\|^2$ though it is somewhat inferior to GACV.
- The radius-margin bound gives a very good prediction of the optimal hyperparameters for SVM L2 soft-margin formulation. However, the possibility of local minima should be taken into consideration when this bound is minimized using gradient descent method.

Acknowledgements

The authors would like to thank Olivier Chapelle and Bernhard Schölkopf for valuable comments through emails. Thanks also go to reviewers for careful readings and helpful comments. Kaibo Duan would like to thank National University of Singapore for financial support through Research Scholarship.

References

[1] R.R. Bailey, E.J. Pettit, R.T. Borochoff, M.T. Manry, X. Jiang, Automatic recognition of USGS land use/cover categories using statistical and neural networks classifiers, in: Proceedings of SPIE OE/Aerospace and Remote Sensing, SPIE 1993.
[2] C.J.C. Burges, A turtorial on support vector machines for pattern recognition, Data Mining Knowledge Discovery 2 (2) (1998) 955–975.
[3] G. Cauwenberghs, T. Poggio. Incremental and decremental support vector machine learning, in: Advances in Neural Information Processing Systems (NIPS'2000), Vol. 13, MIT Press, Cambridge, MA, 2001, pp. 409–415.
[4] O. Chapelle, private communication.
[5] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing kernel parameters for support vector machines, Machine Learning, 46 (2002) 131–160. Available: http://www.ens-lyon.fr/ ochapell/kernel_params.ps.gz
[6] C. Cortes, V. Vapnik, Support vector networks, Machine Learning 20 (1995) 273–297.
[7] N. Cristianini, C. Campbell, J. Shawe-Taylor, Dynamically adapting kernels in support vector machines, in: M. Kearns, S. Solla, D. Cohn (Eds.), Advances in Neural Information Processing Systems, Vol. 11, MIT Press, Cambridge, MA, 1999, pp. 204–210.
[8] D. DeCoste, K. Wagstaff, Alpha seeding for support vector machines, in: Proceedings of Inernational Conference on Knowledge Discovery and Data Mining (KDD-2000), 2000.
[9] T. Joachims, The maximum-margin approach to learning text classifiers: method, theory and algorithms, Ph.D. Thesis, Department of Computer Science, University of Dortmund, 2000.
[10] J.H. Lee, C.J. Lin, Automatic model selection for support vector machines. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, 2000.
[11] M.M.S. Lee, S.S. Keerthi, C.J. Ong, D. DeCoste, An efficient method for computing leave-one-out error in support vector machines, Technical Report, 2001. Available: http://guppy.mpe.nus.edu.sg/ mpessk/papers/loo_new.ps.gz
[12] Luntz, V. Brailovsky, On estimation of characters obtained in statistical procedure of recognition, Technicheskaya Kibernetica 3 (1969) (in Russian).
[13] G. Rätsch, Benchmark data sets, 1999. Available: http://ida.first.gmd.de/∼raetsch/data/benchmarks.htm
[14] B. Schölkopf, private communication.
[15] B. Schölkopf, C. Burges, V. Vapnik, Extracting support data for a given task, in: U.M. Fayyad, R. Uthurusamy (Ed.), Proceedings of the First International Conference on Knowledge Discovery & Data Mining, AAAI Press, Menlo Park, 1995.

[16] K. Tsuda, G. Rätsch, S. Mika, K.-R. Müller, Learning to predict the leave-one-out error of kernel based classifiers, in: A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Proceedings of the ICANN'01, 2001, pp. 331–338.
[17] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[18] V. Vapnik, O. Chapelle, Bounds on error expectation for support vector machine, in: A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 1999.
[19] G. Wahba, Support vector machine, reproducing kernel Hilbert spaces and the randomized GACV, in: B. Scholkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods-Support Vector Learning, MIT press, Cambridge, MA, 1999.
[20] G. Wahba, Y. Lin, Y. Lee, H. Zhang, On the relation between the GACV and Joachims' xi-alpha method for tuning support vector machines, with extension to the nonstandard case, Technical Report 1039, Department of Statistics, University of Wisconsin-Madison, June 2001.
[21] G. Wahba, Y. Lin, H. Zhang, GACV for support vector machines, in: Smola, Bartlett, Schölkopf, Schuurmans (Eds.), Advances in Large Margin Classifiers, MIT Press, Cambridge, MA, 1999.

Kaibo Duan received his B.Eng. degree in Power Engineering in 1996 and his M.Eng. degree in Mechanical Engineering in 1999, both from Nanjing University of Aeronautics and Astronautics (NUAA), China. Currently, he is a Ph.D. student in National University of Singapore, working on kernel methods for classification. His research interests include machine learning and kernel methods.

S. Sathiya Keerthi obtained his Bachelors degree in Mechanical Engineering from REC Trichy, University of Madras in 1980, Masters degree in Mechanical Engineering from University of Missouri-Rolla, in 1982, and Ph.D. in Control Engineering from University of Michigan, Ann Arbor, in 1986. After working for about one year with Applied Dynamics International, Ann Arbor doing R&D in real time simulation, he joined the faculty of the Department of Computer Science and Automation, Indian Institute of Science, Bangalore in April 1987. His academic research covers the following areas: Support Vector Machines, Neural Networks, and Geometric problems in Robotics. He joined the Control Division of the Department of Mechanical Engineering, National University of Singapore, in May 1999, as Associate Professor. Dr. Keerthi has published over 60 papers in leading international journals and conferences.

Aun-Neow Poo received his B.Eng. degree with first class honors from the National University of Singapore and proceeded to the University of Wisconsin as a Ford Foundation Fellow where he received his M.Sc. and Ph.D. degrees in 1970 and 1973, respectively. He is currently Professor in the Department of Mechanical Engineering at the National University of Singapore.

His research interest is in intelligent automation and control in which he has worked for more than 30 years. Among the awards he has received are the Public Administration Medal (silver) from the Government of Singapore, the Chevalier dans l'Ordre des palmes Academiques from the French Government and the Gold Medal from the Institution of Engineers, Singapore.