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

Abstract

Choosing optimal hyperparameters 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 measures. In this paper, we empirically study the usefulness of several simple performance measures that are very inexpensive to compute. The results point out which of these performance measures are adequate functionals for tuning SVM hyperparameters. For SVMs with L1 soft-margin formulation, none of the simple measures yields a performance as good as k-fold cross-validation.

Keywords: Support vector machine; Model selection; Generalization error estimate; Performance measure; Hyperparameter tuning.

1 Introduction

Support vector machines (SVMs) [12] 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 done by solving the primal problem:

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

where x_i is the i-th example, 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:

$$\begin{aligned} &\min \quad f(\alpha) = \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j k(x_i, x_j) - \sum_i \alpha_i \\ &\text{s.t.} \quad 0 \leq \alpha_i \leq C \ \forall i; \quad \sum_i y_i \alpha_i = 0 \end{aligned} \qquad \text{(D)}$$

where $k(x, \bar{x}) = \phi(x) \cdot \phi(\bar{x})$ is the kernel function that performs the nonlinear mapping. Popular kernel functions are:

Gaussian kernel: $k(x, \bar{x}) = \exp(-\frac{\|x - \bar{x}\|^2}{2\sigma^2})$

Polynomial kernel: $k(x, \bar{x}) = (1 + x \cdot \bar{x})^d$

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;

- 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 particularly 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. 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 [9] 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. In spite of that, for tuning SVM hyperparameters, LOO is still very expensive.

2.2 Xi-Alpha Bound

In [7], Joachims developed the following estimate, which is an upper bound on the error rate of leave-one-out procedure. 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):

$$Err_{\xi\alpha} = \frac{1}{l} card\{i : (2\alpha_i R_\Delta^2 + \xi_i) \geq 1\} \tag{1}$$

Here *card* denotes cardinality and R_Δ^2 is an upper bound on $c \leq k(x, \bar{x}) \leq c + R_\Delta^2$ for all x, $\bar{x}$ and some constant c. We refer to the estimate in (1) as the *Xi-Alpha bound.*

2.3 Approximate Span Bound

Vapnik et al [13] 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 better reflects the actual error rate. The following upper bound on LOO error was also proposed in [13]:

$$\frac{N_{LOO}}{l} \leq \frac{S \max(D, 1/\sqrt{C}) \sum_{i=1}^{n^*} \alpha_i + m}{l} \tag{2}$$

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 [13] 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 (2) 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 [13] that $S \leq D_{SV}$. Thus, we get

$$\frac{N_{LOO}}{l} \leq \frac{D_{SV} \max(D, 1/\sqrt{C}) \sum_{i=1}^{n^*} \alpha_i + m}{l} \tag{3}$$

The right-hand side of (2) is referred as the *span bound.* Since the bound in (3) is a looser bound than the span bound, we refer to it as the *approximate span bound.*

2.4 VC Bound

SVMs are based on the idea of *structural risk minimization* introduced by *statistical learning theory* [12]. 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 $(x_1, y_1), \cdots, (x_l, y_l)$ are drawn from some unknown probability distribution $P(x, y)$. Then, the expected test error for a trained machine is:

$$R(\alpha) = \int \frac{1}{2} |y - f(x, \alpha)| dP(x, y)$$

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

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

For a particular choice of α, with probability $1-\eta$ $(0 \le \eta \le 1)$, the following bound holds [12]:

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

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 (4) 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* [12] 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 (4) by a loose bound on it:

$$h \le D^2 \|w\|^2 + 1 \tag{5}$$

The right-hand side of (5) is a loose bound on VC-dimension and, if we use this bound to approximate h, sometimes we may get into a situation where l/h is so small that the term inside the square root in (4) 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 $l+1$.

2.5 Radius-Margin Bound

For SVMs with hard-margin formulation, it was shown by Vapnik et al [13] that the following bound holds:

$$LOO\ Err \le \frac{1}{4l} D^2 \|w\|^2 \tag{6}$$

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 (6) is usually referred as the *radius-margin bound*.

The SVM problem with L2 soft-margin formulation can be converted to the hard-margin SVM problem with a slightly modified kernel function [4]. Chapelle et al [3] explored the computation of gradient of D^2 and $\|w\|^2$, and their results make these gradient computation very easy. In their experiment, they minimize 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 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 mainly 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 datasets: Banana, Image, Splice, Waveform and Tree. General information about the datasets is given in Table 1. The detailed information of the first four datasets can be found in [10]. The Tree dataset 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 datasets 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.

Table 1. General information about the datasets

Datasets	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	11692

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, VC bound, approximate span bound and $D^2\|w\|^2$.

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, particularly with Gaussian kernel.

In 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 dataset (Figures 1 – 4). The plots for the other datasets show similar variations with respect to the two hyperparameters. We make the plots of other datasets available at: http://guppy.mpe.nus.edu/~mpessk/ncfigures.pdf. In order to show the variations 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.

Another experiment was set up to see how the size of the training set affects the performance of different estimates. The Waveform dataset 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. Figure 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.

K-fold Cross-Validation:

On each dataset, 5-fold cross-validation produced a curve that not only has a minimum very close to that of the test error curve, but it also has a shape very similar to the curve of the test error. *Of all the estimates, 5-fold cross-validation yielded the best performance.* Even for a small training set with 200 examples, 5-fold cross-validation gave a quite good estimate of generalization error (see Figure 5).

Recently, a lot of research work has been devoted to speeding up the LOO procedure so that it can be used to tune the hyperparameters of SVMs. Some of those speed-up strategies, such as alpha seeding [6] and loose tolerance [8], 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.

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 a curve that has a shape slightly different from that 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, the α_i are small and hence, the Xi-Alpha estimate in (1) is very close to the LOO estimate.

Another nice property of Xi-Alpha bound is that, irrespective of the size of training set, it always gives an estimate reasonably close to the true minimum in terms of test error (see Figure 5).

To see the correlation of the above two estimate (k-fold cross-validation estimate and Xi-Alpha bound) with test error, we tried the combination 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 plot is shown in Figure 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. This plot shows that 5-fold cross-validation estimate has much better correlation with the test error.

Approximate Span Bound:

In [13], Vapnik et al effectively used span-based idea for tuning SVM hyperparameters. In approximate span bound, S is replaced by D_{SV}. The poor behavior of this bound is probably due to the fact that D_{SV} is a poor approximate of S.

VC Bound:

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

$\mathbf{D^2\|w\|^2}$ for L1 Soft-Margin Formulation:

Let us now consider $D^2\|w\|^2$ for L1 soft-margin formulation. Figure 1 and 2 clearly show the inadequacy of this measure for tuning hyperparameters. The plots for the other datasets are also very similar. The inadequacy 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

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

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

$$\begin{aligned} D^2\|w\|^2 &= D^2\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j) \\ &\leq 4\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j) \end{aligned}$$

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,

$$\begin{aligned} &\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j k(x_i,x_j) \\ &= \sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \\ &= (\sum_{j=1}^{l}\alpha_i y_i)^2 = 0 \end{aligned}$$

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

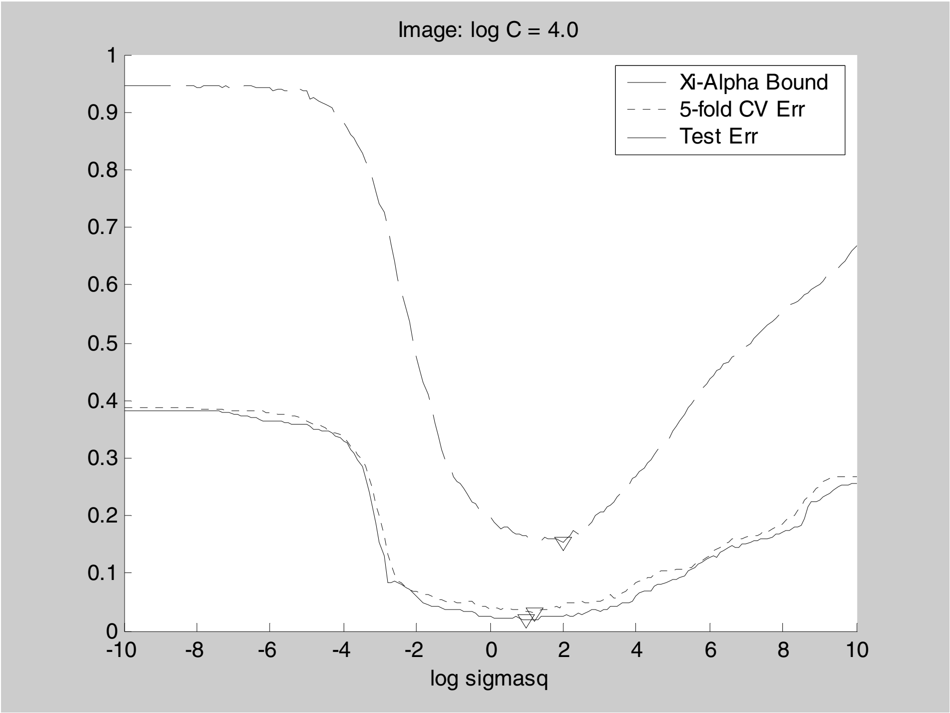
Cristianini et al in [5] showed that $D^2\|w\|^2$ is good for tuning the width of the Gaussian kernel for hard-margin SVM. The asymptotic movement of $D^2\|w\|^2$ to zero as σ^2 goes to infinity that we established above holds only when C is fixed at a finite value. When C is infinity (the hard margin 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 of Cristianini et al.

Schölkopf et al [11] 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.

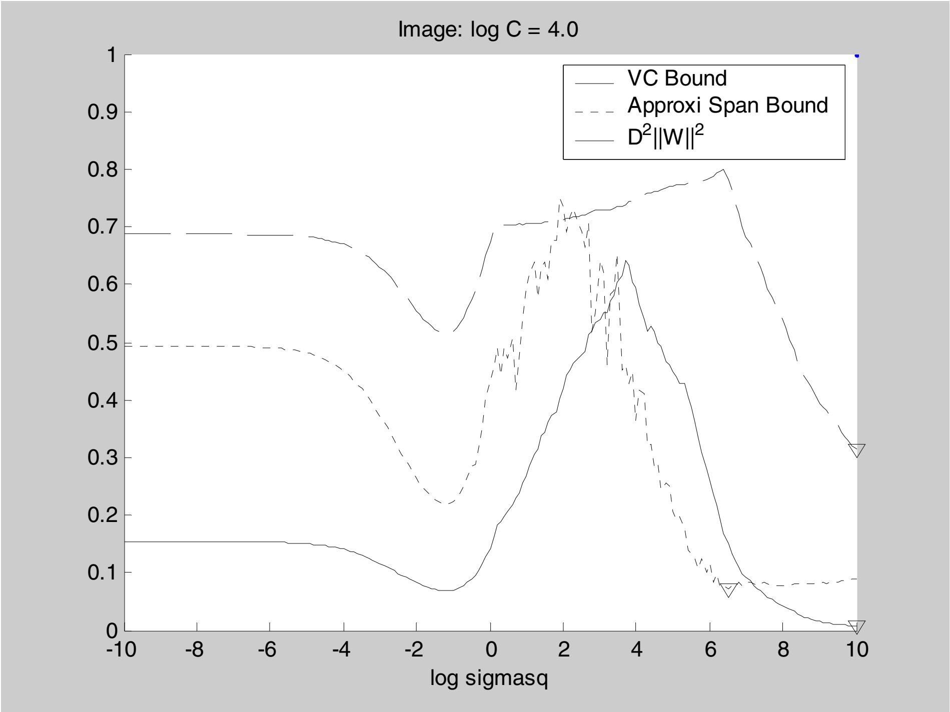
$\mathbf{D^2\|w\|^2}$ for L2 Soft-Margin Formulation:

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, which can be converted to an SVM hard-margin problem, our experiments show that radius-margin bound gives a very good estimate of the optimal hyperparameters. This agrees with the results of Chapelle et al [3], 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 Figure 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.

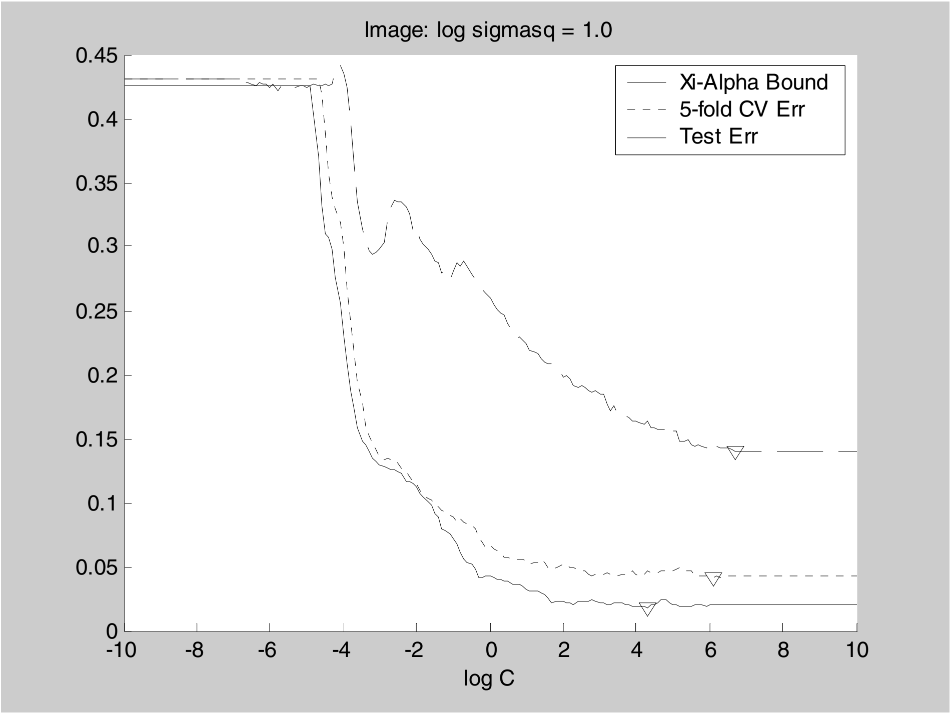


(a)

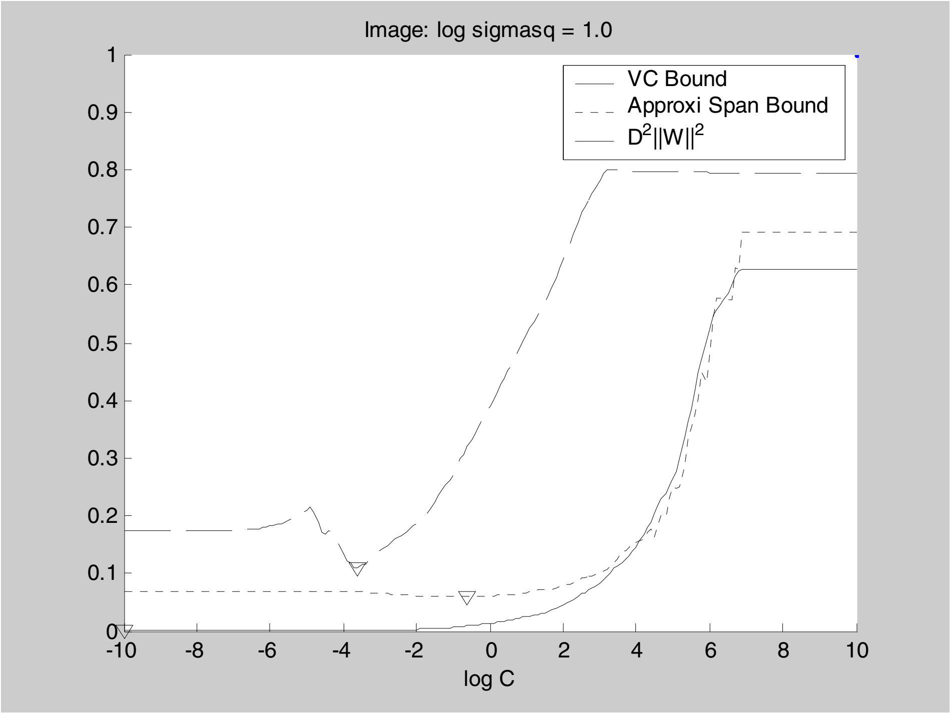


(b)

Figure 1: Variation of Xi-Alpha Bound, 5-fold CV Err, Test Err, 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. In (b), the vertical axis is normalized differently for Xi-Alpha Bound, Approximate Span Bound and $D^2\|w\|^2$. For each curve, ∇ denotes the minimum point.



(a)



(b)

Figure 2: Variation of Xi-Alpha Bound, 5-fold CV Err, Test Err, 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. In (b), the vertical axis is normalized differently for Xi-Alpha Bound, Approximate Span Bound and $D^2\|w\|^2$. For each curve, ∇ denotes the minimum point.

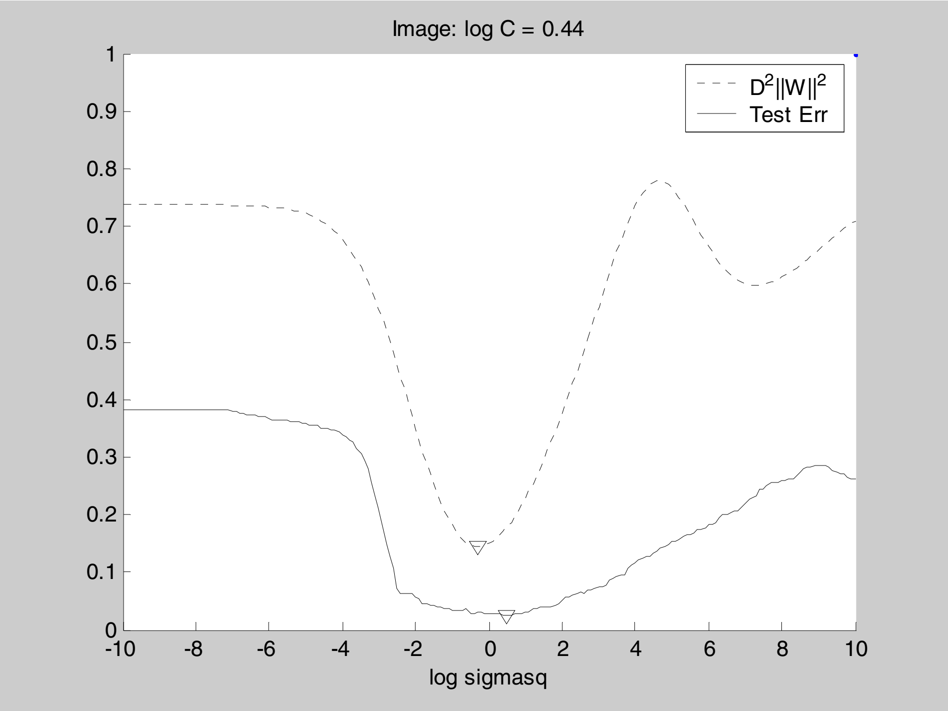


Figure 3: Variation of $D^2\|w\|^2$ and Test Err with respect to σ^2 for fixed C value, for SVM L2 soft-margin formulation. The vertical axis for $D^2\|w\|^2$ is normalized. For each curve, ∇ denotes the minimum point.

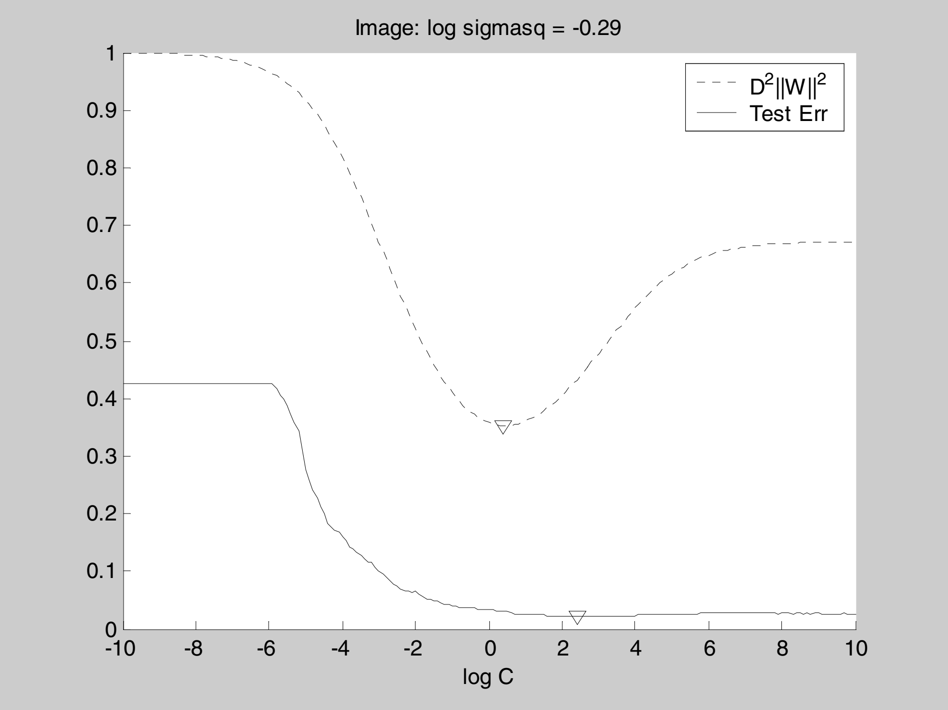


Figure 4: Variation of $D^2\|w\|^2$ and Test Err with respect to C for fixed σ^2 value, for SVM L2 soft-margin formulation. The vertical axis for $D^2\|w\|^2$ is normalized. For each curve, ∇ denotes the minimum point.

Table 2: The value of Test Err at the 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 (0.60)	0.0188 (1.00)	0.0947 (3.40)	0.1022 (3.20)	0.1089 (3.80)
5-fold CV Err	0.1276 (1.30)	0.0198 (1.20)	0.0975 (3.20)	0.1159 (4.40)	0.1144 (5.0)
Xi-Alpha Bound	0.1453 (-2.10)	0.0257 (2.00)	0.0979 (3.80)	0.1035 (3.0)	0.1551 (1.0)
VC Bound	0.4094 (8.90)	0.2564 (10.0)	0.1766 (8.40)	0.3293 (10.0)	0.2609 (-10.0)
Approxi Span Bound	0.3943 (6.60)	0.1436 (6.50)	0.1407 (5.60)	0.1243 (5.20)	0.1356 (9.80)
$D^2\|w\|^2$	0.5594 (10.0)	0.2564 (10.0)	0.4800 (10.0)	0.3293 (10.0)	0.1627 (-2.40)

Table 3: The value of Test Err at the 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.

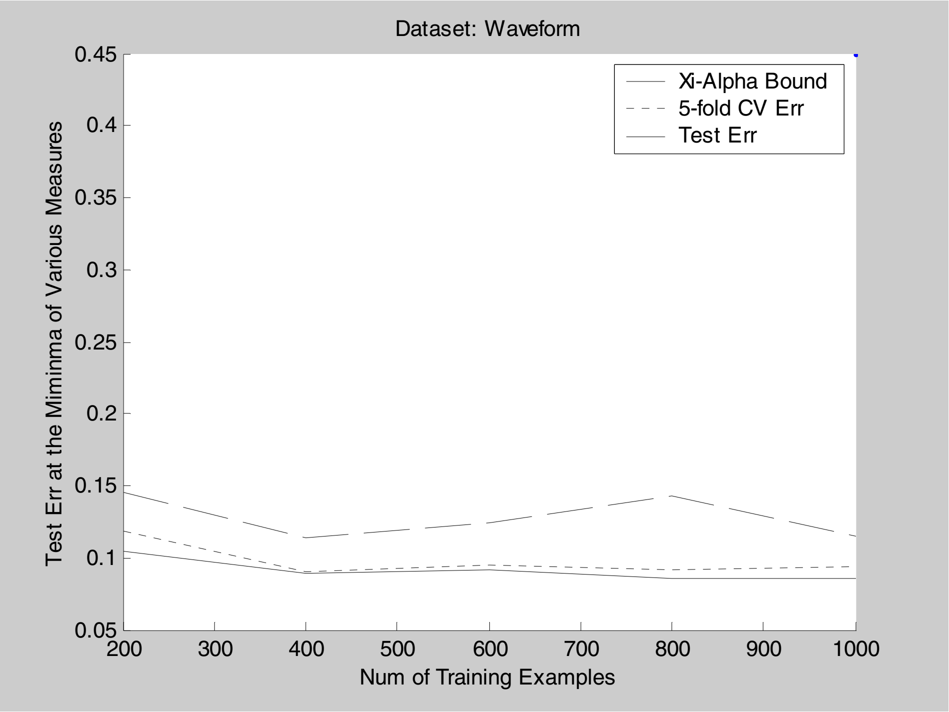
Criterion	Banana log σ^2=0.60	Image log σ^2=1.0	Splice log σ^2=3.40	Waveform log σ^2=3.20	Tree log σ^2=3.80
Test Err	0.1045 (5.20)	0.0178 (4.30)	0.0947 (0.40)	0.1022 (1.40)	0.1089 (8.60)
5-fold CV Err	0.1278 (9.00)	0.0198 (6.10)	0.0947 (0.50)	0.1102 (0.0)	0.1218 (4.80)
Xi-Alpha Bound	0.1286 (9.30)	0.0198 (6.70)	0.3398 (-2.70)	0.1487 (-2.80)	0.1160 (9.60)
VC Bound	0.3987 (-3.0)	0.1584 (-3.6)	0.4800 (-10.0)	0.3293 (-10.0)	0.2609 (-10.0)
Approxi Span Bound	0.1251 (1.80)	0.0535 (-0.60)	0.1136 (-0.90)	0.1102 (0.0)	0.1363 (1.20)
$D^2\|w\|^2$	0.5594 (-10.0)	0.2564 (-10.0)	0.4800 (-10.0)	0.3293 (-10.0)	0.2609 (-10.0)

Table 4: The value of Test Err at the 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.

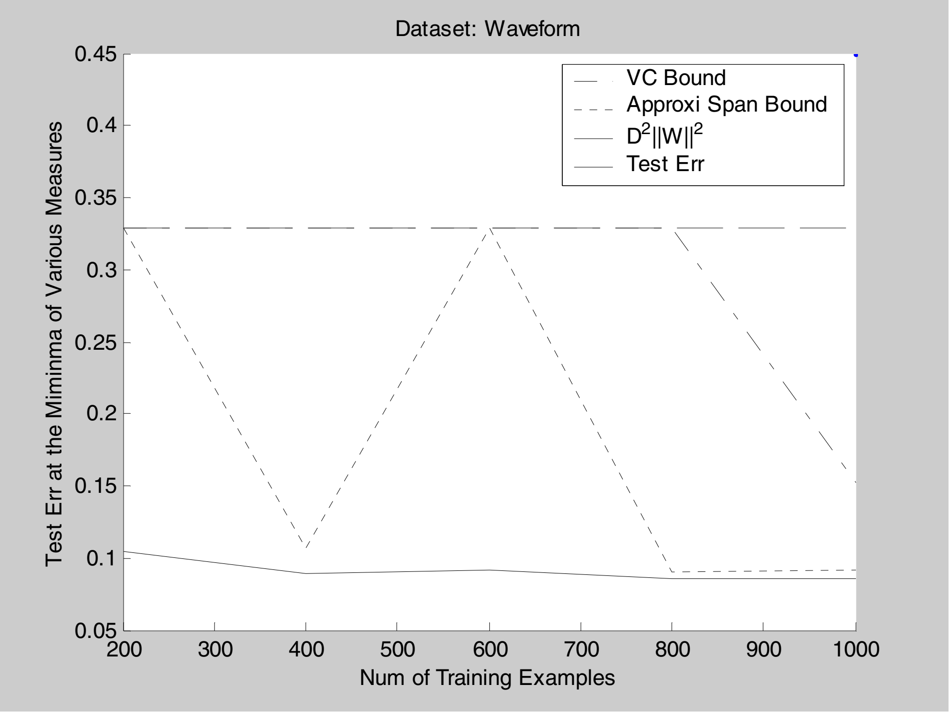
Criterion	Banana log C = -0.90	Image log C = 0.44	Splice log C = 6.91	Waveform log C = 0	Tree log C = 9.80
Test Err	0.1118 (-1.40)	0.0238 (0.50)	0.0947 (3.30)	0.0991 (2.80)	0.1049 (4.60)
$D^2\|w\|^2$	0.1141 (-1.60)	0.0297 (-0.30)	0.1002 (3.10)	0.1011 (2.20)	0.1627 (-2.40)

Table 5: The value of Test Err at the 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 σ^2=-1.39	Image log σ^2=-0.29	Splice log σ^2=3.07	Waveform log σ^2= 2.80	Tree log σ^2=4.60
Test Err	0.1118 (0.0)	0.0218 (2.40)	0.1007 (2.20)	0.0991 (0.0)	0.1049 (9.80)
$D^2\|w\|^2$	0.1127 (-0.90)	0.0297 (0.40)	0.1016 (9.20)	0.1007 (-0.60)	0.1413 (-1.40)

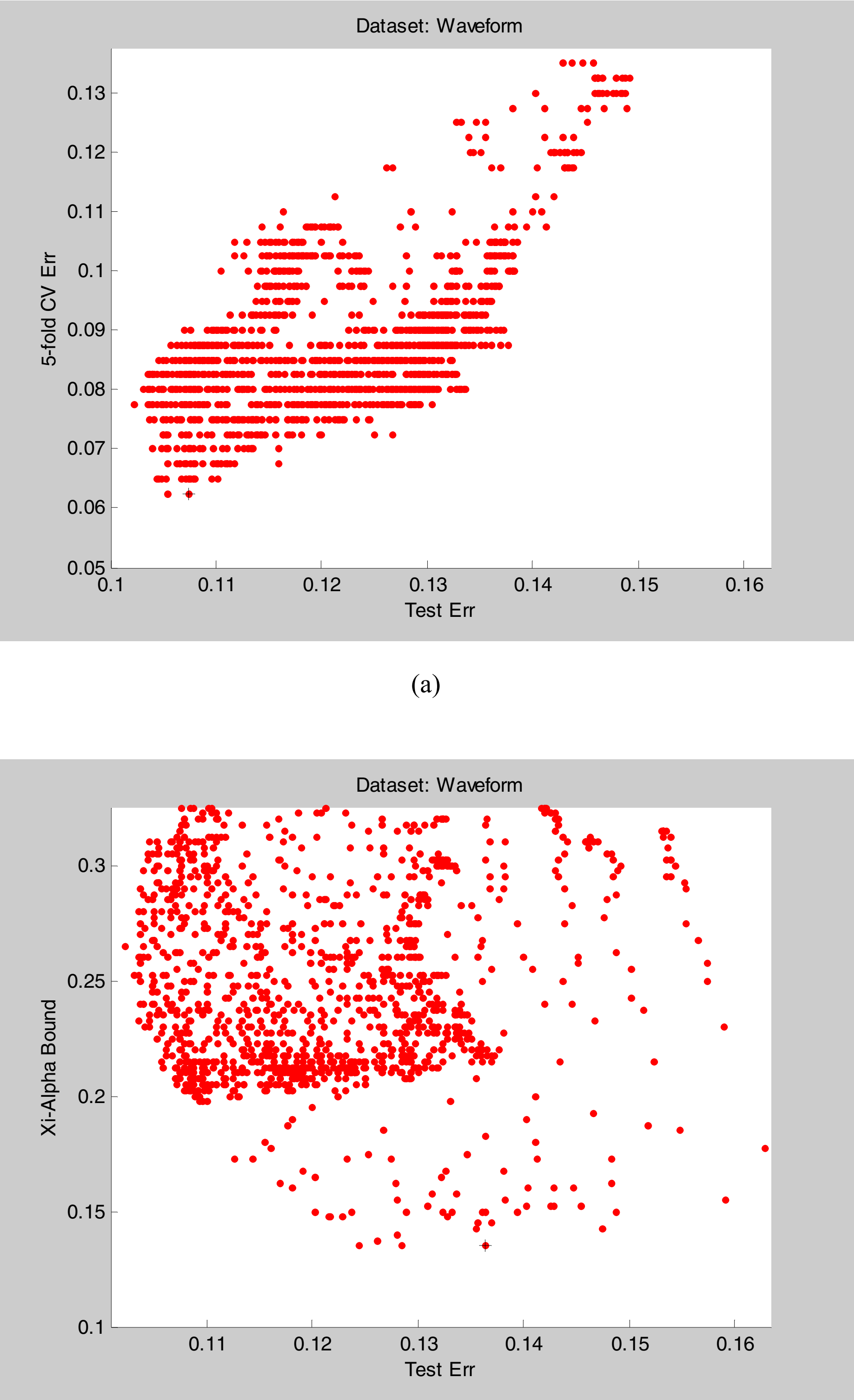


(a)



(b)

Figure 5: Performance of various measures for different training set sizes. The waveform dataset 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.



(a)

(b)

Figure 6: Correlation of 5-fold cross-validation and Xi-Alpha bound 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 +.

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:

- 5-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 5-fold cross validation. It even gives a good estimate on small training set. The 5-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.

- 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 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.

References

[1] R.R. Bailey, E.J. Pettit, R.T. Borochoff, M.T. Manry, and 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*,* Vol. 2, No.2 (1998) 955-975.

[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, Choosing Kernel Parameters for Support Vector Machines, Submitted to Machine Learning, 2000. Available: http://www.ens-lyon.fr/~ochapell/kernel_params.ps.gz

[4] C. Cortes and V. Vapnik, Support Vector Networks, Machine Learning 20 (1995) 273-297.

[5] N. Cristianini, C. Campbell and J. Shawe-Taylor, Dynamically Adapting Kernels in Support Vector Machines, in: M. Kearns, S. Solla and D. Cohn, Ed., Advances in Neural Information Processing Systems, Vol. 11. (MIT Press, 1999) 204-210.

[6] D. DeCoste and K. Wagstaff, Alpha Seeding for Support Vector Machines, In: Proceedings of Inernational Conference on Knowledge Discovery and Data Mining (KDD-2000).

[7] 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.

[8] J. H. Lee and C.J. Lin, Automatic Model Selection for Support Vector Machines. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, 2000.

[9] Luntz and V. Brailovsky, On Estimation of Characters Obtained in Statistical Procedure of Recognition, Technicheskaya Kibernetica, 3 (1969). (in Russian).

[10] G. Rätsch, Benchmark Datasets, 1999.
Available: http://ida.first.gmd.de/~raetsch/data/benchmarks.htm

[11] B. Schölkopf, C. Burges, and V. Vapnik, Extracting Support Data for A Given Task, in: U. M. Fayyad and R. Uthurusamy, Ed.,Proc. First International Conference on Knowledge Discovery & Data Mining (AAAI Press, Menlo Park, 1995).

[12] V. Vapnik, Statistical Learning Theory (John Wiley & Sons, 1998).

[13] V. Vapnik and O. Chapelle, Bounds on Error Expectation for Support Vector Machine, in: Smola, Bartlett, Schölkopf and Schuurmans, Ed., Advences in Large Margin Classifiers (MIT Press, 1999).