

# Learning the prior for the PAC-Bayes bound

**A. Ambroladze, E. Parrado-Hernández, J. Shawe-Taylor**

Image, Speech and Intelligent Systems

School of Electronics and Computer Science

University of Southampton

October 15, 2004

## **Abstract**

This paper presents a bound on the performance of a Support Vector Machine obtained within the PAC-Bayes framework. The bound is computed by means of the estimation of a prior of the distribution of SVM classifiers given a particular dataset, and the use of this prior in the PAC-Bayes generalisation bound. The quality of the bound is tested in a model selection task, where it is compared against other procedures to select models based on other PAC-Bayes bounds and ten fold cross-validation. Furthermore, we introduce an algorithm to approximately optimise the new bound and test it against a standard SVM both in terms of bound value and test set error.

## **1 Introduction**

Support vector machines implement linear classifiers in a high-dimensional feature space using the kernel trick to enable a dual representation and efficient computation.

The danger of overfitting in such high-dimensional spaces is countered by maximising the margin of the classifier on the training examples. For this reason there has been considerable interest in bounds on the generalisation in terms of the margin.

Early bounds have relied on covering number computations [5], while later bounds have considered Rademacher complexity. The tightest bounds for practical applications appear to be the PAC-Bayes bound [4] and in particular the form given in [3].

One can view these bounds as a posteriori justifications for the strategy adopted in training an SVM, but as the bounds become more accurate the possibility of using them to perform model selection suggests itself. This could mean that for example kernel parameters such as the radius parameter  $\sigma$  of the Gaussian kernel could be set by choosing the value that optimises the bound.

Unfortunately, so far the more standard (statistically poorly justified and relatively expensive) method of cross-validation has proved more reliable in most experiments. The aim of this paper is to consider a refinement of the PAC-Bayes approach and investigate whether it can improve on the original PAC-Bayes bound and furthermore whether it can deliver reliable model selection.

The standard PAC-Bayes bound uses a Gaussian prior centred at the origin in weight space. The key to the new bound is to use part of the training set to compute a more informative prior and then compute the bound on the remainder of the examples relative to this prior. Subsequently we also consider optimising the new bound by retraining the SVM on the remainder of the training set relative to the prior. The bounds and performance of the new classifiers are tested experimentally.

The rest of the document is organised as follows. Section 2 briefly reviews the PAC-Bayes bound for SVM obtained in [3]. The new bound obtained by means of the refinement of the prior is presented in Section 3; moreover Section 4 describes the algorithm employed to approximately estimate the new bound. The experimental work, included in Section 5, introduces a model selection task to evaluate the usability of the bound and as well as a comparison in terms of classification error of the new algorithm and the standard SVM. Finally, the main conclusions of this work are outlined in Section 6.

## 2 PAC-Bayes bound for SVM

This section is devoted to a brief review of the PAC-Bayes Bound Theorem of [3]. Let us consider a certain distribution  $D$  of patterns  $\mathbf{x}$  lying in a certain input space  $\mathcal{X}$ , with their corresponding output labels  $y$ ,  $y \in \{-1, 1\}$ . In addition, let us also consider a distribution  $Q$  over the classifiers  $c$ . Now we can define two error measures over  $D$ : the true error,  $Q_D \equiv \mathbb{E}_{c \sim Q} c_D$ , as the probability of misclassifying an instance  $\mathbf{x}$  chosen from  $D$  with a classifier  $c$  chosen from  $Q$ ; and the empirical error  $\hat{Q}_S \equiv \mathbb{E}_{c \sim Q} \hat{c}_S$ , as the probability of misclassifying an instance  $\mathbf{x}$  chosen from a certain set of patterns  $S$  of size  $m$  with  $c$ . The terms  $c_D$  and  $\hat{c}_S$  are the true and empirical error of the individual classifier  $c$ , respectively.

From these two quantities we can derive the PAC-Bayes Bound on the true error that holds for any fixed prior  $P(c)$  over the classifiers,  $c$  and for any  $\delta \in (0, 1]$

$$\Pr_{S \sim D^m} \left( \forall Q(c) : \text{KL}(\hat{Q}_S \| Q_D) \leq \frac{\text{KL}(Q \| P) + \ln(\frac{m+1}{\delta})}{m} \right) \geq 1 - \delta, \quad (1)$$

where  $\text{KL}$  is the Kullback-Leibler divergence,  $\text{KL}(p \| q) = q \ln \frac{q}{p} + (1 - q) \ln \frac{1-q}{1-p}$  and  $\text{KL}(Q \| P) = \mathbb{E}_{c \sim Q} \ln \frac{Q(c)}{P(c)}$ .

This bound can be particularised for the case of linear classifiers in the following way. The  $m$  training patterns define a linear classifier that can be represented by the following equation<sup>1</sup>:

$$c(\mathbf{x}) = \text{sign}(\mathbf{w}^T \phi(\mathbf{x})) \quad (2)$$

where  $\phi(\mathbf{x})$  is a nonlinear projection to a certain feature space where a linear classification actually takes place, and  $\mathbf{w}$  is a vector from that feature space determining the separating plane.

For any vector  $\mathbf{w}$  we can define a stochastic classifier in the following way: we choose the distribution  $Q$  to be a spherical Gaussian centered on the direction given by  $\mathbf{w}$  at a distance  $\mu$  of the origin.

Then, according to [3], for classifiers of the form in equation (2) performance can be bounded by

$$\Pr \left( \text{KL}(\hat{Q}_S(\mathbf{w}, \mu) \| Q_D(\mathbf{w}, \mu)) \leq \frac{\frac{\mu^2}{2} + \ln(\frac{m+1}{\delta})}{m} \right) \geq 1 - \delta \quad (3)$$

now  $\hat{Q}_S$  is a stochastic measure of the error of the classifier on the training set. It can be proved (see [3]) that

$$\hat{Q}(\mathbf{w}, \mu)_S = \mathbb{E}_m[\tilde{F}(\mu\gamma(\mathbf{x}, y))] \quad (4)$$

where  $\mathbb{E}_m$  is the mathematical expectation over the  $m$  train examples,  $\gamma(\mathbf{x}, y)$  is the margin of the training patterns

$$\gamma(\mathbf{x}, y) = \frac{y\mathbf{w}^T \phi(\mathbf{x})}{\|\phi(\mathbf{x})\| \|\mathbf{w}\|} \quad (5)$$

and  $\tilde{F} = 1 - F$ , where  $F$  is the cumulative normal distribution

$$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx \quad (6)$$

Bound (3) holds for any (fixed) distribution  $D$  simultaneously for all values of  $\mu$  and  $\mathbf{w}$ .

Moreover, the bound in (3) can be particularised for the SVM classifier, since it obeys expression (2) by means of the kernel trick [2]. In this case, the prior  $P(c)$  is selected to be a spherical Gaussian centered on the origin and with unity standard deviation, while the posterior is determined by the SVM weight vector  $\mathbf{w}_{SVM}$ . The error of  $\mathbf{w}_{SVM}$  can be bound by at most twice this value.

<sup>1</sup>We are considering here unbiased classifiers, i.e., with  $b = 0$ .

### 3 New Prior based bound for SVMs

Our first contribution is motivated by the fact that the PAC-Bayes bound allows us to choose the prior distribution,  $P(c)$ . In the standard application of the bound this is chosen to be a Gaussian centred at the origin. We now consider choosing a different prior based on training an SVM on a subset  $R$  of the training set, of  $r$  training patterns and labels,  $\{\mathbf{x}_k^R, y_k^R\}_{k=1}^r$ . In the experiments this is taken as a random subset but for simplicity of the presentation we will assume these to be the last  $r$  examples  $\{\mathbf{x}_k, y_k\}_{k=m-r+1}^m$ .

With these  $r$  examples we can determine an SVM classifier,  $\mathbf{w}_R$  and form a prior  $P(c)$  consisting in a Gaussian distribution centred at  $\mathbf{w}_R$ . Using this prior, the bound now becomes

$$\Pr \left( \text{KL}(\hat{Q}_S || Q_D) \leq \frac{\frac{\|\mathbf{w}_R - \mathbf{w}\|^2}{2} + \ln\left(\frac{m-r+1}{\delta}\right)}{m-r} \right) \geq 1 - \delta \quad (7)$$

where  $\hat{Q}_S$  is evaluated on the first  $m - r$  examples only.

Note that if we consider  $s$  priors, we should include an extra term  $\ln(s)$  in the numerator. Further note that this bound can be applied to any centre  $\mathbf{w}$  of the posterior distribution  $Q$ , even that determined by the SVM trained on the whole data set, the difference for the original bound is of course that  $Q_S$  is now the average of the  $m - r$  examples not used to determine the prior. In the next section we will consider training an 'SVM' on those examples taking account of the new prior, but our first observation is that we now have a new way of bounding the performance of an SVM. Remarkably, we will see in experiments how this bound can be tighter than the standard PAC-Bayes bound.

The remainder of this section explains the procedure followed to compute the bound. This bound for  $Q_D$  is obtained from the following inequality:

$$Q_D \leq \text{KL}_r^{-1}\{\hat{Q}_S(\mathbf{w}, \mu), \Delta(\mu)\} \quad (8)$$

where

- $\text{KL}_r^{-1}\{p, z\}$  is the larger of the inverses of the Kullback-Leibler divergence for the right argument of the divergence. In other words,

$$q = \text{KL}_r^{-1}\{p, z\} \Rightarrow z = \text{KL}(p||q) \quad \text{and } q \geq p \quad (9)$$

- 

$$\Delta(\mu) = \frac{\frac{\mu^2 \|\mathbf{w} - \mathbf{w}_R\|^2}{2} + \ln\left(\frac{m-r+1}{\delta}\right)}{m-r} \quad (10)$$

- and  $\hat{Q}(\mathbf{w}, \mu)_S$  is the stochastic measure of the error of the classifier on the training set. Note that in the present case, this empirical error has to be averaged only on the  $m - r$  examples not used to build the prior.

In Section 5 we present experimental results comparing this bound to the standard PAC-Bayes bound and using it to guide model selection.

## 4 Optimising the new bound: p-SVM

In this section, we consider a modified SVM algorithm that aims to approximately optimise the bound obtained in the last section.

After obtaining the prior,  $\mathbf{w}_R = \sum_{k=m-r+1}^m \hat{\alpha}_k y_k \phi(\mathbf{x}_k)$ , the optimisation problem to determine the p-SVM (p stands for prior) classifier becomes

$$\min_{\mathbf{w}, \xi_i} \left[ \frac{1}{2} \|\mathbf{w} - \mathbf{w}_R\|^2 + C \sum_{i=1}^{m-r} \xi_i \right] \quad (11)$$

s. t.

$$y_i \mathbf{w}^T \phi(\mathbf{x}_i) \geq 1 - \xi_i \quad i = 1, \dots, m - r \quad (12)$$

$$\xi_i \geq 0 \quad i = 1, \dots, m - r \quad (13)$$

We can build a Lagrangian functional to be optimised by the introduction of the constraints with multipliers  $\alpha_i, i = 1, \dots, m - r$ .

$$L_P = \frac{1}{2} \|\mathbf{w} - \mathbf{w}_R\|^2 + C \sum_{i=1}^{m-r} \xi_i - \sum_{i=1}^{m-r} \alpha_i (y_i \mathbf{w}^T \phi(\mathbf{x}_i) - 1 + \xi_i) - \sum_{i=1}^{m-r} \eta_i \xi_i, \quad \eta_i, \alpha_i \geq 0 \quad (14)$$

Taking the gradient of (14) with respect to  $\mathbf{w}$  and derivatives with respect to  $\xi_i$  we obtain the optimality conditions:

$$\mathbf{w} - \mathbf{w}_R = \sum_{i=1}^{m-r} \alpha_i y_i \phi(\mathbf{x}_i) \quad (15)$$

$$C - \alpha_i - \eta_i = 0 \Rightarrow 0 \leq \alpha_i \leq C \quad i = 1, \dots, m - r \quad (16)$$

Substituting equation (15) in functional (14) and applying the optimality condition (16) we arrive at

$$L_D = \frac{1}{2} \left\| \sum_{i=1}^{m-r} \alpha_i y_i \phi(\mathbf{x}_i) \right\|^2 - \sum_{i=1}^{m-r} \alpha_i (y_i (\mathbf{w}_R^T + \sum_{i=1}^{m-r} \alpha_i y_i \phi^T(\mathbf{x}_i)) \phi(\mathbf{x}_i) - 1) \quad (17)$$

Now we can replace the prior  $\mathbf{w}_R$  by its combination of projected input vectors and substitute the inner products by kernel functions to arrive at

$$L_D = \frac{1}{2} \sum_{i,j=1}^{m-r} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{m-r} \alpha_i - \sum_{i=1}^{m-r} \sum_{k=m-r+1}^m \alpha_i y_i \hat{\alpha}_k y_k \kappa(\mathbf{x}_i, \mathbf{x}_k) - \sum_{i,j=1}^{m-r} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad (18)$$

Grouping terms we have

$$L_D = \sum_{i=1}^{m-r} \alpha_i (1 - y_i \sum_{k=m-r+1}^m \hat{\alpha}_k y_k \kappa(\mathbf{x}_i, \mathbf{x}_k)) - \frac{1}{2} \sum_{i,j=1}^{m-r} \alpha_i \alpha_j y_i y_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \quad (19)$$

Optimising functional (19) subject to constraint (16) we obtain coefficients  $\alpha_i$ ,  $i = 1, \dots, m - r$  that are plugged into equation (15) to obtain  $\Delta(\mu)$  using equation (10). Thus we have computed the second ingredient for determining the bound. Let us address the stochastic error  $\hat{Q}(\mathbf{w}, \mu)_S$  in equation (8).

With respect to the stochastic measure of error,  $\hat{Q}_S$ , in this case, the margin has to be computed according to an overall classifier resulting from  $\mathbf{w}$  that can be expressed with dual variable  $\alpha + \hat{\alpha}$ . Therefore the margin can be computed as:

$$\gamma(\mathbf{x}, y) = \frac{y \sum_{i=1}^m (\hat{\alpha}_i + \alpha_i) y_i \kappa(\mathbf{x}_i, \mathbf{x})}{\sqrt{\kappa(\mathbf{x}, \mathbf{x})}} \quad j = 1, \dots, N - r \quad (20)$$

Up to this point, we have determined all the terms in equation (8) with the exception of parameter  $\mu$ . Varying  $\mu$  now implies varying the prior and so we must add a  $\ln(s)/(m - r)$  term to  $\Delta(\mu)$  if we consider  $s$  different values, however, in the experimental section we decided not to take this extra term into account since its value was found to be insignificant. We have implemented a linear search to obtain the value of  $\mu$  that yields the tightest value for the bound  $Q_D$ . This search as well as the calculation described in the previous paragraphs are summarised in Table 1. The algorithm detailed in Table 1 can be used to either compute the bound on the SVM presented in Section 3 or the one on the p-SVM, depending on which classifier we plug into step 3.

## 5 Experiments

In this section we include two experiments that illustrate the quality of the bound on the performance of the SVM and the capability of the p-SVM to improve the original SVM accuracy.

Table 1: Algorithm to determine the bound

<p><b>Input:</b> <math>\{\mathbf{x}_i, y_i\}_{i=1}^m, \sigma, r</math></p> <p><b>Output:</b> <math>Q_D</math></p>
<ol style="list-style-type: none"> <li>1.- Determine <math>R = \{\mathbf{x}_j, y_j\}_{j=i_1}^{i_r}</math> and <math>M = X - R</math></li> <li>2.- <math>\mathbf{w}_R = \text{SVM}(\{\mathbf{x}_j, y_j\}_{j=i_1}^{j_r}, \sigma)</math></li> <li>3.- Compute <math>W = \ \mathbf{w} - \mathbf{w}_R\ ^2</math>; (<math>\mathbf{w} = \mathbf{w}_{p\text{-SVM}}^{m-r}</math> or <math>\mathbf{w}_{\text{SVM}}^m</math>)</li> <li>4.- Compute <math>L = \ln(\frac{m-r+1}{\delta})</math></li> <li>5.- Compute <math>\gamma(\mathbf{x}, y)</math> for test examples</li> <li>6.- Linear Search for <math>\mu</math> <ol style="list-style-type: none"> <li>6.1.- Fix 4 values of <math>\mu : \{\mu_1, \mu_2, \mu_3, \mu_4\}</math></li> <li>6.2.- For every value of <math>\mu</math> <ol style="list-style-type: none"> <li>6.2.1.- Compute <math>\mathbb{E}_{m-r}[\mu\gamma(\mathbf{x}, y)]</math></li> <li>6.2.2.- Compute <math>\Delta(\mu)</math></li> <li>6.2.3.- Compute <math>Q_{D_i}</math> for every value of <math>\mu</math></li> </ol> </li> <li>6.3.- Determine the <math>\mu</math> that produces the minimum <math>Q_{D_i}</math></li> <li>6.4.- Refine the values of <math>\mu</math></li> <li>6.5.- Go back to 6.2 until convergence</li> </ol> </li> </ol>

Table 2: Description of the datasets.

<b>Problem</b>	<b># Samples</b>	<b># Positives</b>	<b># Negatives</b>	<b>Dimension</b>
Boston	506	355	151	13
Pima	768	538	230	8
Ionosphere	351	246	105	34
Liver	345	242	103	6

## 5.1 Bound guided Model Selection

This experiment involves a model selection task in which we compare the bound introduced in this paper (equation (7)) with that of [3] and a classic ten fold cross-validation procedure. For this task we have selected four UCI [1] datasets, whose descriptions are included in Table 2. Every dataset was split in two subsets, a training set containing 80% of the total amount of patterns and a test set containing the remaining 20%. These subsets maintain the proportion of positive and negative instances of the original datasets.

The task actually consists in using the training set to determine the optimal value of  $\sigma$  with every algorithm and then check which algorithm found out the correct  $\sigma$ , i.e., the one that produces the best classification of the test set. For the procedures based on bounds, the selected  $\sigma$  is the one that produces a tighter bound on the test error of the classifier, while for the cross-validation procedure, the selected  $\sigma$  is the one that minimises the classification error in the left-out folds.

A set of five values of  $\sigma$  is explored for each problem. These values considered are multiples of the square root of the input dimension of the problem,  $d$ :  $\sigma = \{0.25\sqrt{d}, 0.5\sqrt{d}, \sqrt{d}, 2\sqrt{d}, 4\sqrt{d}\}$ .

Tables 3 to 6 show the results of the simulations for the four problems. The column labelled as CE shows the real classification error for the test set. The column labelled 10 F displays the estimation of the test error carried out by means of the ten fold cross-validation; column LANG contains the PAC-Bayes bound on the real error of the classifier computed according to [3] and columns under label APS show the bound presented in this article for several sizes of the reduced training set  $r$ . The values presented in columns APS correspond to the average on 100 different selections of the reduced training set,  $R$ ; the value in brackets indicates the number of times that the bound was the tightest for that particular value of  $\sigma$  (all the bracketed numbers in the same column sum 100).

According to the numbers displayed in the Tables, the bound presented here is able to select the

Table 3: Model selection for Boston

Value of $\sigma$	CE	10 F	LANG	APS		
				10%	20%	30%
$0.25\sqrt{d}$	<b>0.129</b>	<b>0.119</b>	0.481	0.468 (0)	0.462 (0)	0.460 (0)
$0.5\sqrt{d}$	0.158	0.123	0.402	0.376 (0)	0.371 (0)	0.368 (0)
$\sqrt{d}$	0.178	0.128	<b>0.383</b>	0.349 ( <b>100</b> )	0.343 ( <b>100</b> )	0.339 ( <b>100</b> )
$2\sqrt{d}$	0.198	0.136	0.425	0.383 (0)	0.371 (0)	0.364 (0)
$4\sqrt{d}$	0.238	0.165	0.498	0.449 (0)	0.433 (0)	0.424 (0)

Table 4: Model selection for Pima

Value of $\sigma$	CE	10 F	LANG	APS		
				10%	20%	30%
$0.25\sqrt{d}$	0.286	0.257	0.529	0.529 (0)	0.530 (0)	0.532 (0)
$0.5\sqrt{d}$	0.260	0.249	0.475	0.472 (0)	0.473 (0)	0.475 (0)
$\sqrt{d}$	0.260	0.229	<b>0.432</b>	0.420 (49)	0.418 (4)	0.419 (1)
$2\sqrt{d}$	<b>0.240</b>	<b>0.221</b>	0.445	0.420 ( <b>51</b> )	0.406 ( <b>96</b> )	0.403 ( <b>99</b> )
$4\sqrt{d}$	0.253	0.231	0.497	0.485 (0)	0.473 (0)	0.462 (0)

Table 5: Model selection for Ionosphere

Value of $\sigma$	CE	10 F	LANG	APS		
				10%	20%	30%
$0.25\sqrt{d}$	0.129	0.113	0.492	0.486 (0)	0.485 (0)	0.487 (0)
$0.5\sqrt{d}$	0.100	<b>0.085</b>	<b>0.440</b>	0.430 (0)	0.428 (0)	0.429 (0)
$\sqrt{d}$	<b>0.086</b>	0.096	<b>0.440</b>	0.413 ( <b>100</b> )	0.401 ( <b>100</b> )	0.395 ( <b>100</b> )
$2\sqrt{d}$	0.100	0.142	0.502	0.478 (0)	0.457 (0)	0.440 (0)
$4\sqrt{d}$	0.171	0.217	0.569	0.565 (0)	0.562 (0)	0.558 (0)

Table 6: Model selection for Liver

Value of $\sigma$	CE	10 F	LANG	APS		
				10%	20%	30%
$0.25\sqrt{d}$	0.362	0.341	0.579	0.583 (0)	0.587 (0)	0.592 (0)
$0.5\sqrt{d}$	<b>0.261</b>	<b>0.293</b>	<b>0.576</b>	0.577 ( <b>100</b> )	0.578 ( <b>90</b> )	0.582 ( <b>69</b> )
$\sqrt{d}$	0.275	0.304	0.584	0.583 (0)	0.583 (10)	0.585 (31)
$2\sqrt{d}$	0.401	0.402	0.600	0.601 (0)	0.604 (0)	0.608 (0)
$4\sqrt{d}$	0.420	0.420	0.593	0.592 (0)	0.598 (0)	0.603 (0)

parameter correctly in three out of the four cases. This result is equivalent to the one achieved by the cross-validation procedure. However, the computational cost of computing the bound is dramatically smaller than the cost of the cross-validation: the former involves two optimisation problems with a smaller number of patterns than any of the ten problems that need to be solved by the latter. On the other hand, Langford’s bound only selects the correct model in two of the three problems. Nevertheless, this is the least costly method of the three.

Furthermore, the new bound has been shown to be robust with respect to the size of the reduced training set in the sense that except for the case of `pima` all the three sizes are consistent with the selection of the model.

## 5.2 Classification with p-SVM

The second experiment is devoted to evaluate the classification capabilities of p-SVM in comparison with those of the standard SVM. We have selected the same datasets of the previous experiment with the same proportion of training/test set sizes. In order to skip the model selection previous step, we have fixed the value of the kernel spread parameter to  $\sigma = \sqrt{d}$ . Table 7 shows the results of the comparison. The numbers in Table 7 correspond to the average over ten different splits in training and test sets.

The results show that in three out of the four problems tried, the p-SVM bound turns out to be tighter than the SVM bound. This effect is more noticeable in problems `Pima` and `Ionosphere`. With respect to the classification error, it should be pointed out that p-SVM performs clearly better than the standard SVM in two of the proposed datasets (`Boston` and `Pima`) and comparably in `Liver`.

Table 7: Comparison of p-SVM and SVM on the four UCI datasets. For the Bound rows, numbers in brackets stand for the number of times that this entry was the minimum in the column out of a total of ten different training/test sets splits. On the other hand, for the Test error rows, the number in brackets show the number of times that the algorithm achieved the smallest error. Note that the column labelled p-SVM 0% actually shows the standard SVM test error and Langford’s PAC-Bayes Bound.

Boston				
	p-SVM 0%	p-SVM 10%	p-SVM 20%	p-SVM 30%
Bound SVM	0.393	0.358 (4)	0.353 (8)	0.348 (7)
Bound p-SVM	0.393	0.357 (6)	0.354 (2)	0.350 (3)
Test Error	0.133 (0.76)	0.133 (1.08)	0.128 (4.08)	0.130 (4.08)
Pima				
	p-SVM 0%	p-SVM 10%	p-SVM 20%	p-SVM 30%
Bound SVM	0.439	0.431 (2)	0.430 (3)	0.434 (3)
Bound p-SVM	0.439	0.428 (8)	0.427 (7)	0.433 (7)
Test Error	0.214 (1.67)	0.212 (3.67)	0.212 (2.67)	0.219 (2)
Ionosphere				
	p-SVM 0%	p-SVM 10%	p-SVM 20%	p-SVM 30%
Bound SVM	0.441	0.414 (0)	0.403 (1)	0.397 (0)
Bound p-SVM	0.441	0.406 (10)	0.393 (9)	0.386 (10)
Test Error	0.090 (4.42)	0.094 (2.08)	0.094 (1.92)	0.096 (1.58)
Liver				
	p-SVM 0%	p-SVM 10%	p-SVM 20%	p-SVM 30%
Bound SVM	0.585	0.585 (4)	0.585 (3)	0.585 (4.5)
Bound p-SVM	0.585	0.585 (6)	0.585 (7)	0.585 (5.5)
Test Error	0.288 (2.5)	0.288 (2.5)	0.286 (2.5)	0.288 (2.5)

## 6 Concluding remarks

We have presented a new bound on the performance of a SVM classifiers based on the estimation of a prior of the distribution of SVM classifiers given a particular dataset, and the use of this prior in the PAC-Bayes generalisation bound. The experimental work has shown that this bound is tighter than previous state-of-the-art PAC-Bayes bounds for the same classifiers. Moreover, the bound has been found to perform with high reliability in a model selection task, achieving results comparable to classic ten fold cross validation, but a much lower computational burden.

In addition we have come up with a new algorithm to approximately optimise the bound. The classifiers built according to this new algorithm show globally better generalization capabilities than the original SVM.

This positive results open lines for immediate research that explore strategies for further refinements of the prior on the distribution of classifiers to eventually arrive at both tighter bounds on the performance and smaller classification error rates.

## References

- [1] C L Blake and C J Merz. *UCI Repository of machine learning databases*. University of California, Irvine, Dept. of Information and Computer Sciences, [<http://www.ics.uci.edu/~mlern/MLRepository.html>], 1998.
- [2] Bernhard E. Boser, Isabelle Guyon, and Vladimir Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learning Theory*, pages 144–152, 1992.
- [3] J Langford. Tutorial on practical prediction theory for classification. Technical report, IBM Research, 2002.
- [4] J Langford and J Shawe-Taylor. PAC-Bayes & Margins. In *Advances in Neural Information Processing Systems*, volume 14, Cambridge MA, 2002. MIT Press.
- [5] J Shawe-Taylor, P L Bartlett, R C Williamson, and M Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Trans. Information Theory*, 44(5):1926 – 1940, 1998.