A Nonconformity Approach to Model Selection for SVMs

David R. Hardoon, Zakria Hussain and John Shawe-Taylor

Telephone: +44 (0)20 7679 0425
Fax: +44 (0)20 7679 1397
Electronic Mail: {D.Hardoon, Z.Hussain, jst}@cs.ucl.ac.uk
URL: http://www.cs.ucl.ac.uk/staff/{D.Hardoon, Z.Hussain, J.Shawe-Taylor}/

Abstract

We investigate the issue of model selection and the use of the nonconformity (strangeness) measure in batch learning. Using the nonconformity measure we propose a new training algorithm that helps avoid the need for Cross-Validation or Leave-One-Out model selection strategies. We provide a new generalisation error bound using the notion of nonconformity to upper bound the loss of each test example and show that our proposed approach is comparable to standard model selection methods, but with theoretical guarantees of success and faster convergence. We demonstrate our novel model selection technique using the Support Vector Machine.

Keywords

Nonconformity, Cross Validation, Support Vector Machines

Department of Computer Science
University College London
Gower Street
London WC1E 6BT, UK

1 Introduction

Model Selection is the task of choosing the best model for a particular data analysis task. It generally makes a compromise between fit with the data and the complexity of the model. Furthermore, the chosen model is used in subsequent analysis of test data. Currently the most popular techniques used by practitioners are Cross-Validation (CV) and Leave-One-Out (LOO).

In this paper the model we concentrate on is the Support Vector Machine (SVM) (Boser et al., 1992). CV and LOO are the modus operandi despite there being a number of alternative approaches proposed in the SVM literature. For instance, Chapelle and Vapnik (1999) explore model selection using the span of the support vectors and re-scaling of the feature space, whereas, Momma and Bennett (2002), motivated by an application in drug design, propose a fully-automated search methodology for model selection in SVMs for regression and classification. Gold and Sollich (2003) give an in depth review of a number of model selection alternatives for tuning the kernel parameters and penalty coefficient C for SVMs, and although they find a model selection technique that performs well (at high computational cost), the authors conclude that "*the hunt is still on for a model selection criterion for SVM classification which is both simple and gives consistent generalisation performance*". More recent attempts at model selection have been given by Hastie et al. (2004) who derive an algorithm that fits the entire path of SVM solutions for every value of the cost parameter, while Li et al. (2005) propose to use the Vapnik-Chervonenkis (VC) bound; they put forward an algorithm that employs a coarse-to-fine search strategy to obtain the best parameters in some predefined ranges for a given problem. Furthermore, Ambroladze et al. (2006) propose a tighter PAC-Bayes bound to measure the performance of SVM classifiers which in turn can be used as a way of estimating the hyperparameters. Finally, de Souza et al. (2006) have addressed model selection for multi-class SVMs using Particle Swarm Optimisation.

Recently, Özöğür-Akyüz et al. (In Press), following on work by Özöğür et al. (2008), show that selecting a model whose hyperplane achieves the maximum separation from a test point obtains comparable error rates to those found by selecting the SVM model through CV. In other words, while methods such as CV involve finding one SVM model (together with its optimal parameters) that minimises the CV error, Özöğür-Akyüz et al. (In Press) keep all of the models generated during the model selection stage and make predictions according to the model whose hyperplane achieves the maximum separation from a test point. The main advantage of this approach is the computational saving when compared to CV or LOO. However, their method is only applicable to large margin classifiers like SVMs.

We continue this line of research, but rather than using the distance of each test point from the hyperplane we explore the idea of using the *nonconformity measure* (Vovk et al., 2005; Shafer & Vovk, 2008) of a test sample to a particular label set. The nonconformity measure is a function that evaluates how 'strange' a prediction is according to the different possibilities available. The notion of nonconformity has been proposed in the on-line learning framework of conformal prediction (Shafer & Vovk, 2008), and is a way of scoring how different a new sample is from a bag[1] of old samples. The premise is that if the observed samples are well-sampled then we should have high confidence on correct prediction of new samples, given that they *conform* to the observations.

We take the nonconformity measure and apply it to the SVM algorithm during testing in order to gain a time advantage over CV and to generalise the algorithm of Özöğür-Akyüz et al. (In Press). Hence we are not restricted to SVMs (or indeed a measure of the margin for prediction) and can apply our method to a broader class of learning algorithms. However, due to space constraints we only address the SVM technique and leave the application to other algorithms (and other nonconformity measures not using the margin) as a future research study. Furthermore we also derive a novel learning theory bound that uses nonconformity as a measure of complexity. To our knowledge this is the first attempt at using this type of measure to upper bound the loss of learning algorithms.

The paper is laid out as follows. In Section 2 we present the definitions used throughout the paper. Our main algorithmic contributions are given in Section 3 where we present our nonconformity measure and its novel use in prediction. Section 4 presents a novel generalisation error bound for our proposed algorithm. Finally, we present experiments in Section 5 and conclude in Section 6.

2 Definitions

The definitions are mainly taken from Shafer and Vovk (2008).

Let (x_i, y_i) be the ith input-output pair from an input space $\mathbf{X}$ and output space $\mathbf{Y}$. Let $z_i = (x_i, y_i)$ denote short hand notation for each pair taken from the joint space $\mathbf{Z} := \mathbf{X} \times \mathbf{Y}$.

We define a *nonconformity measure* as a real valued function $A(S, z)$ that measures how different a sample z is from

---

[1]A *bag* is a more general formalism of a mathematical *set* that allows repeated elements.

a set of observed samples $S = \{z_1, \ldots, z_m\}$. A nonconformity measure must be fixed *a priori* before any data has been observed.

Conformal predictions work by making predictions according to the nonconformity measure outlined above. Given a set $S = \{z_1, \ldots, z_m\}$ of training samples observed over $t = 1, \ldots, m$ time steps and a new sample x, a conformal prediction algorithm will predict y from a set containing the correct output with probability $1 - \epsilon$. For example, if $\epsilon = 0.05$ then the prediction is within the so-called *prediction region* – a set containing the correct y, with 95% probability. In this paper, we extend this framework to the batch learning model to make predictions using confidence estimates, where for example we are 95% confident that our prediction is correct.

In the batch learning setting, rather than observing samples incrementally such as $x_1, y_1, \ldots, x_m, y_m$ we have a training set $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ containing all the samples for training that are assumed to be distributed i.i.d. from a fixed (but unknown) distribution $\mathcal{D}$. Given a function (hypothesis) space $\mathcal{H}$ the batch algorithm takes training sample S and outputs a hypothesis $f : \mathbf{X} \mapsto \mathbf{Y}$ that maps samples to labels.

For the SVM notation let $\phi : \mathbf{X} \mapsto \mathbf{F}$ map the training samples to a higher dimensional feature space $\mathbf{F}$. The primal SVM optimisation problem can be defined like so:

$$\begin{aligned}
\min_{w,b} \quad & \|w\|_2^2 + C \sum_{i=1}^n \xi_i \\
\text{subject to} \quad & y_i \left(\langle w, \phi(x_i) \rangle + b \right) \geq 1 - \xi_i \\
& i = 1, \ldots, n.
\end{aligned}$$

where b is the bias term, $\xi \in \mathbb{R}^n$ is the vector of slack variables and $w \in \mathbb{R}^n$ is the primal weight vector, whose 2-norm minimisation corresponds to the maximisation of the margin between the set of positive and negative samples. The notation $\langle \cdot, \cdot \rangle$ denotes the inner product. The dual optimisation problem gives us the flexibility of using *kernels* to solve nonlinear problems (Schölkopf & Smola, 2002; Shawe-Taylor & Cristianini, 2004). The dual SVM optimisation problem can be formulated like so:

$$\begin{aligned}
\max_\alpha \quad & \sum_i^m \alpha_i - \frac{1}{2} \sum_{i=1}^m y_i y_j \alpha_i \alpha_j \kappa(x_i, x_j), \\
\text{subject to} \quad & \sum_{i=1}^m y_i \alpha_i = 0, \\
& 0 \leq \alpha_i \leq C,
\end{aligned}$$

where $\kappa(\cdot, \cdot)$ is the kernel function and $\alpha \in \mathbb{R}^m$ is the dual (Lagrangian) variables. Throughout the paper we will use the dual optimisation formulation of the SVM as we attempt to find the optimal regularisation parameter for the SVM together with the optimal kernel parameters.

3 Nonconformity Measure

We now discuss the main focus of the paper. Let $S = S_{\text{trn}} \cup S_{\text{val}}$ be composed of a training set S_{trn} and a validation set S_{val}. We assume without loss of generality that,

$$S = \{z_1^t, \ldots, z_m^t, z_1^v, \ldots, z_n^v\}$$

where $S_{\text{trn}} = \{z_1^t, \ldots, z_m^t\}$ and $S_{\text{val}} = \{z_1^v, \ldots, z_n^v\}$.

We start by defining our nonconformity measure $A(S_{\text{val}}, z)$ for a function f over the validation set S_{val} and $j = 1, \ldots, n$ as,

$$A(S_{\text{val}}, z) = yf(x). \tag{1}$$

Note that this does not depend on the whole sample but just the test point. In itself it does not characterise how different the point is. To do this we need the so called *p-value* $p_A(S_{\text{val}}, z)$ that computes the fraction of points in S_{val} with 'stranger' values:

$$p_A(S_{\text{val}}, z) = \frac{\left| \left\{ 1 \leq j \leq n : A(S_{\text{val}}, z_j^v) \leq A(S_{\text{val}}, z) \right\} \right|}{n},$$

which, in this case, measures the number of samples from the validation set that have smaller functional margin than the test point functional margin. The larger the margin obtained the more confidence we have in our prediction. The nonconformity p-value of z is between 1 and $1/n$. If it is small (tends to $1/n$) then sample z is non-conforming and if it is large (tends to 1) then it is conforming.

In order to better illustrate this idea we show a simple pictorial example in Figure 1. We are given six validation samples ordered around 0 (solid line) in terms of their correct/incorrect classification *i.e.*, the value $y^v f(x^v)$ for an

$(x^v, y^v) = z^v$ pair will be correctly classified by f iff $y^v f(x^v) > 0$. In our example two are incorrectly classified (below the threshold) and four are correct. The picture on the left also includes $yf(x)$ for a test sample x when its label is considered to be positive *i.e.*, $y = +1$. In this case there remain 3 validation samples below its value of $yf(x)$ giving us a nonconformity measure p-value using Equation (2) as $p_A(S_{\text{val}}, (x, y = +1)) = \frac{3}{6}$. A similar calculation can be made for the picture on the right when we consider the label $y = -1$ for test point x *i.e.*, $(x, y = -1)$. We are able to conclude, for this sample, that assigning x a label of $y = +1$ gives a nonconformity p-value of $p_A(S_{\text{val}}, (x, +1)) = \frac{1}{2}$ while assigning a label of $y = -1$ gives a p-value of $p_A(S_{\text{val}}, (x, -1)) = \frac{1}{6}$. Therefore, with a higher probability, our test sample x is conforming to $+1$ (or equally non-conforming to -1) and should be predicted positive. We state the standard result for nonconformity measures, but first define a nonconformity
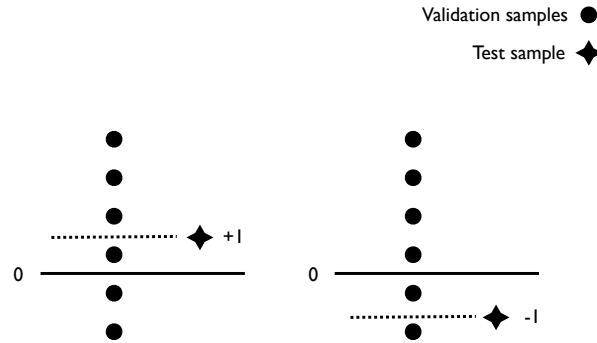

Figure 1: A simple illustrative example of non-conformal prediction using a validation set of six samples (2 are misclassifications, 4 are correctly classified) on a single test sample with a positive functional value and its two label possibilities of $+1$ (left) and -1 (right).

prediction scheme and its associated error.

Definition 3.1. *For a fixed nonconformity measure $A(S, z)$, its associated p-value, and $\epsilon > 0$, the confidence predictor Γ^ϵ predicts the label set*

$$\Gamma^\epsilon(S, x) = \{y : p_A(S, (x, y)) \geq \epsilon\}$$

The confidence predictor Γ^ϵ makes an error on sample $z = (x, y)$ if $y \notin \Gamma^\epsilon(S, x)$.

Proposition 3.2. *For exchangeable distributions we have that*

$$P^{n+1} \{(S, z) : y \notin \Gamma^\epsilon(S, x)\} \leq \epsilon.$$

Proof. By exchangeability all permutations of a training set are equally likely. Denote with $\tilde{S}$ the set S extended with the sample z_{n+1} and for σ a permutation of $n + 1$ objects. Let $\tilde{S}_\sigma$ be the sequence of samples permuted by σ. Consider the permutations for which the corresponding prediction of the final element of the sequence is not an error. This implies that the value $A(\tilde{S}_\sigma, z_{\sigma(n+1)})$ is in the upper $1 - \epsilon$ fraction of the values $A(\tilde{S}_\sigma, z_{\sigma(i)}), i = 1, \ldots, n+1$. This will happen at least $1 - \epsilon$ of the time under the permutations, hence upper bounding the probability of error over all possible sequences by ϵ as required. $\qquad\square$

Following the theoretical motivation from Shawe-Taylor (1998) we proceed by computing all the SVM models and applying them throughout the prediction stage. A fixed validation set, withheld from training, is used to calculate the nonconformity measures. We start by constructing K SVM models so that each decision function $f_k \in F$ is in the set F of decision functions with $k = 1, \ldots, K$. The different set of SVM models can be characterised by different regularisation parameters for C (or ν in ν-SVM) and the width parameter γ in the Gaussian kernel case. For instance, given 10 $C = \{C_1, \ldots, C_{10}\}$ values and 10 $\gamma = \{\gamma_1, \ldots, \gamma_{10}\}$ values for a Gaussian kernel we would have a total of $|C| \times |\gamma| = 100$ SVM models, where $|\cdot|$ denotes the cardinality of a set.

We now describe our new model selection algorithm for the SVM using nonconformity. If the following

$$\frac{|\{\forall j : y_j f_k(x_j) \leq yf_k(x)\}|}{n} > \epsilon$$

statement holds, then we include $y \in \Gamma_k^\epsilon$ where Γ is the prediction region (set of labels conforming). For classification, the set Γ can take the following values:

$$\{\emptyset\}, \{-1\}, \{+1\}, \{-1, +1\}.$$

Clearly finding the prediction region $\Gamma = \{-1\}$ or $\Gamma = \{+1\}$ is useful in the classification scenario as it gives higher confidence of the prediction being correct, while the sets $\Gamma = \{\emptyset\}$ and $\Gamma = \{-1, +1\}$ are useless as the first abstains from making a prediction whilst the second is unbiased towards a label.

Let ϵ_{crit} be the critical ϵ that creates one label in the set Γ_k^ϵ for at least one of the K models:

$$\epsilon_{crit} = \min_{k \in K} \min_{y \in \{-1, +1\}} \frac{\left| \{\forall j : y_j^v f_k(x_j^v) \le y f_k(x)\} \right|}{n}. \tag{2}$$

Furthermore, let k_{crit}, y_{crit} be arguments that realise the minimum ϵ_{crit}, chosen randomly in the event of a tie. This now gives the prediction of x as $y = -y_{crit}$. This is because y_{crit} is non-conforming (strange) and we wish to select the *opposite* (conforming) label. In the experiments section we refer to the prediction strategy outlined above and the model selection strategy given by equation (2) as the nonconformity model selection strategy. We set out the pseudo-code for this procedure in Algorithm 1.

Algorithm 1 Nonconformity model selection.

Input: Sample $S = \{(x_i, y_i)\}_{i=1}^\ell$, SVM parameters C and γ (for Gaussian kernel) where $K = |C| \times |\gamma|$

Output: Predictions of test points $x_{\ell+1}, x_{\ell+2}, \ldots$

1: Take training data S and randomly split into training set $S_{\text{trn}} = \{(x_1^t, y_1^t), \ldots (x_m^t, y_m^t)\}$ and validation set $S_{\text{val}} = \{(x_1^v, y_1^v), \ldots, (x_n^v, y_n^v)\}$ where $m + n = \ell$ {This split is only done *once*}.

2: Train K SVM models on training data S_{trn} to find $f_1(\cdot), \ldots, f_K(\cdot)$.

3: **Prediction Procedure**: For a test point x compute:

$$\epsilon_{crit} = \min_{k \in K} \min_{y \in \{-1, +1\}} \left\{ \frac{\left| \{\forall j : y_j^v f_k(x_j^v) \le y f_k(x)\} \right|}{n} \right\},$$

realised by $k = k_{crit}$ and $y = y_{crit}$.

4: Predict label $-y_{crit}$ for x.

Before proceeding we would like to clarify some aspects of the Algorithm. The data is split into a training and validation set *once* and therefore all K models are computed on the training data – after this procedure we only require to calculate the nonconformity measure p-value for all test points in order to make predictions. However, in b-fold Cross-Validation we require to train, for each C and γ parameter, a further b times. Hence CV will be at most b times more computationally expensive.

4   Nonconformity Generalisation Error Bound

The problem with Proposition 3.2 is that it requires the validation set to be generated afresh for each test point, specifies just one value of ϵ, and only applies to a single test function. In our application we would like to reuse the validation set for all of our test data and use an empirically determined value of ϵ. Furthermore we would like to use the computed errors for different functions in order to select one for classifying the test point.

We therefore need to have uniform convergence of empirical estimates to true values for all values of ϵ and all functions K. We first consider the question of uniform convergence for all values of ϵ.

If we consider the cumulative distribution function $F(\gamma)$ defined by

$$F(\gamma) = P\left((x, y) : y f(x) \le \gamma\right),$$

we need to bound the difference between empirical estimates of this function and its true value. This corresponds to bounding the difference between true and empirical probabilities over the sets

$$\mathcal{A} = \{(-\infty, a] : a \in \mathbb{R}\}.$$

Observe that we cannot shatter two points of the real line with this set system as the larger cannot be included in a set without the smaller. It follows that this class of functions has Vapnik-Chervonenkis (VC) dimension 1. We can therefore apply the following standard result, see for example Devroye et al. (1996).

Theorem 4.1. *Let $\mathcal{X}$ be a measurable space with a fixed but unknown probability distribution P. Let $\mathcal{A}$ be a set system over $\mathcal{X}$ with VC dimension d and fix $\delta > 0$. With probability at least $1 - \delta$ over the generation of an i.i.d. m-sample $S \subset \mathcal{X}$,*

$$\left| \frac{|S \cap \mathcal{A}|}{m} - P(\mathcal{A}) \right| \leq 5.66 \sqrt{\frac{d \ln\left(\frac{em}{d}\right) + \ln \frac{8}{\delta}}{m}}.$$

We now apply this result to the error estimations derived by our algorithm for the K possible choices of model.

Proposition 4.2. *Fix $\delta > 0$. Suppose that the validation set S_{val} of size n in Algorithm 1 has been chosen i.i.d. according to a fixed but unknown distribution that is also used to generate the test data. Then with probability at least $1 - \delta$ over the generation of S_{val}, if for a test point x the algorithm returns a classification $y^v = -y_{crit}$, using function $f_{k_{crit}}, 1 \leq k_{crit} \leq K$, realising a minimum value of ϵ_{crit}, then the probability of misclassification satisfies*

$$P\left((x,y) : y \neq y^v\right) \leq \epsilon_{crit} + 5.66 \sqrt{\frac{\ln(en) + \ln \frac{8K}{\delta}}{n}}.$$

Proof. We apply Theorem 4.1 once for each function f_k, $1 \leq k \leq K$ with δ replaced by δ/K. This implies that with probability $1 - \delta$ the bound holds for all of the functions f_k, including the chosen $f_{k_{crit}}$. For this function the empirical probability of the label y_{crit} being observed is ϵ_{crit}, hence the true probability of this opposite label is bounded as required. $\qquad\qquad\square$

Remark 4.3. *The bound in Proposition 4.2 is applied* using *each test sample which in turn gives a different bound value for each test point (e.g., see Shawe-Taylor (1998)). Therefore, we are unable to compare this bound with existing training set CV bounds (Kearns & Ron, 1999; Zhang, 2001) as they are traditional a* priori *bounds computed over the training data, and which give a uniform value for all test points (i.e., training set bounds (Langford, 2005)).*

5 Experiments

In the following experiments we compare SVM model selection using traditional CV to our proposed nonconformity strategy as well as to the model selection using the maximum margin (Özöğür-Akyüz et al., In Press) from a test sample.

We make use of the Votes, Glass, Haberman, Bupa, Credit, Pima, BreastW and Ionosphere data sets acquired from the UCI machine learning repository.[2] The data sets were pre-processed such that samples containing unknown values and contradictory labels were removed. Table 1 lists the various attributes of each data set. The LibSVM package 2.85 (Chang & Lin, 2001) and the Gaussian kernel were used throughout the experiments. Model selection was

Table 1: Description of data sets: Each row contains the name of the data set, the number of samples and features (i.e. attributes) as well as the total number of positive and negative samples.

Data set	# Samples	# Features	# Positive Samples	# Negative Samples
Votes	52	16	18	34
Glass	163	9	87	76
Haberman	294	3	219	75
Bupa	345	6	145	200
Credit	653	15	296	357
Pima	768	8	269	499
BreastW	683	9	239	444
Ionosphere	351	34	225	126

carried out for the values listed in Table 2.

In the experiments we apply a 10-fold CV routine where the data is split into 10 separate folds, with 1 used for testing and the remaining 9 split into a training and validation set. We then use the following procedures for each of the two model selection strategies:

---

[2]http://archive.ics.uci.edu/ml/

Table 2: Model selection values for γ and C for both cross-validation and nonconformity measure.

$$\gamma \;=\; \{2^{-15}, 2^{-13}, 2^{-11}, 2^{-9}, 2^{-7}, 2^{-5}, 2^{-3}, 2^{-1}, 2^{1}, 2^{3}\}$$
$$C \;=\; \{2^{-5}, 2^{-3}, 2^{-1}, 2^{1}, 2^{3}, 2^{5}, 2^{7}, 2^{9}, 2^{11}, 2^{13}, 2^{15}\}$$

- *Nonconformity*: split the samples into a training and validation set of size $\min(\frac{1}{5}\ell, 50)$ where ℓ is the number of samples.[3] Using the training data we learn all models using C and γ from Table 2.

- *Cross-Validation*: carry out a 10-fold CV *only* on the training data used in the Nonconformity procedure to find the optimal C and γ from Table 2.

The validation set is excluded from training in both methods, but used for prediction in the nonconformity method. Hence, the samples used for training and testing were identical for both CV and the nonconformity model selection strategy. We feel that this was a fair comparison as both methods were given the same data samples from which to train the models.

Table 3 presents the results where we report the average error and standard deviation for Cross-Validation and the nonconformity strategy. We are immediately able to observe that carrying out model selection using the nonconformity measure is, on average, a factor of 7.3 times faster than using CV. The results show that (excluding the Haberman data set) nonconformity seems to perform similarly to CV in terms of generalisation error. However, lower values for the standard deviation on Votes, Glass, Bupa and Credit suggest that on these data sets nonconformity gives more consistent results than CV. Furthermore, when excluding the Haberman data set, the overall error for the model selection using nonconformity is 0.1730 ± 0.0659 and CV is 0.1686 ± 0.0886, constituting a difference of only 0.0044 (less than half a percent) in favour of CV and a standard deviation of 0.0227 in favour of the nonconformity approach. We hypothesise that the inferior results for Haberman are due to the very small numbers of features (only 3).

We also compare the nonconformity strategy to the SVM L_{∞} maximum margin approach (Özöğür-Akyüz et al., In Press). The SVM L_{∞} selects the model with the maximum margin from the test sample in order to make predictions. Once again, the training and testing sets were identical for both methods. Observe that despite the L_{∞} being approximately 7s faster (on average) than our proposed method, we obtain an improvement of 0.0251 ± 0.0108. Hence, bringing us closer to the CV error rate (nonconformity is overall only 1.17% worse than CV when including the Haberman dataset and 0.44% worse when excluding). In fact we obtain lower error rates, than SVM L_{∞}, on all datasets except for Credit (but with a smaller standard deviation).

Since we do not have a single number for the bound on generalisation (as traditional bounds) but rather individual values for each test sample, it is not possible to simply compare the bound with the test error. In order to show how the bound performs we plot the generalisation error as a function of the bound value.

For each value of the bound we take the average error of all test points with predicted error less than or equal to that value. In other words, we create a set[4] B containing the various bound values computed on the test samples. Subsequently, for each element in the set *i.e.*, $\forall i, b_i \in B$ we compute the average error value for the test samples that have a bound value that is smaller or equal to b_i.

Figure 2 shows a plot of this error rate as a function of the bound value. The final value of the function is the overall generalisation error, while the lower error rates earlier in the curve are those attainable by filtering at different bound values. As expected the error increases monotonically as a function of the bound value. Clearly there is considerable weakness in the bound, but this is partly a result of our using a quite conservative VC bound – our main aim here is to show that the predictions are correlated with the actual error rates.

We believe these results to be encouraging as our theoretically motivated model selection technique is faster and achieves similar error rates to Cross-Validation, which is generally considered to be the gold standard. We also find that the nonconformity strategy is slightly slower than the maximum margin approach but performs better in terms of generalisation error.

---

[3]The size of the validation set was varied without much difference in generalisation error.
[4]Hence, no repetition of identical bound values are allowed.

Table 3: Model selection results: Average error and standard deviation as well as the run-time (in seconds) for model selection using nonconformity measure, 10-fold Cross Validation and the SVM-L_∞ margin distance .

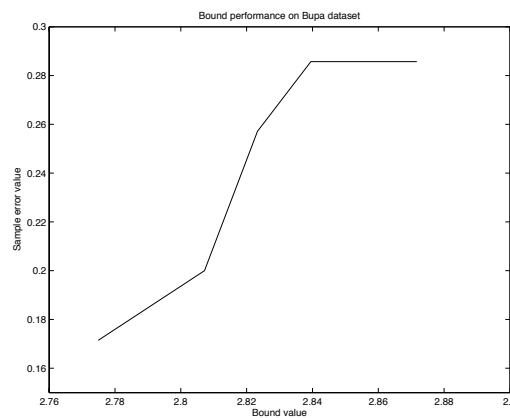| Data set	Nonconformity	Run-Time	Cross-Validation	Run Time	SVM-L_∞	Run-Time
Votes	0.0700 ± 0.1201	0.72s	0.0833 ± 0.2115	5.74s	0.0933 ± 0.0991	0.43s
Glass	0.2167 ± 0.0932	5.25s	0.2085 ± 0.1291	32.58s	0.2328 ± 0.1263	3.08s
Haberman	0.3133 ± 0.0680	58.23s	0.2518 ± 0.0397	455.77s	0.3300 ± 0.0524	49.23s
Bupa	0.2753 ± 0.0620	44.85s	0.2840 ± 0.0604	329.96s	0.3192 ± 0.1085	38.81s
Credit	0.2990 ± 0.0468	86.85s	0.2745 ± 0.1111	592.42s	0.2914 ± 0.0850	72.31s
Pima	0.2562 ± 0.0554	169.05s	0.2473 ± 0.0361	1305.29s	0.3019 ± 0.0516	155.65s
BreastW	0.0378 ± 0.0350	24.80s	0.0335 ± 0.0282	150.36s	0.0408 ± 0.0367	18.29s
Ionosphere	0.0562 ± 0.0493	17.63s	0.0479 ± 0.0440	103.45s	0.1158 ± 0.0565	11.85s
Overall	0.1905 ± 0.0662	50.92s	0.1788 ± 0.0825	371.94s	0.2156 ± 0.0770	43.71s
Overall ex. Haberman	0.1730 ± 0.0659	49.87s	0.1686 ± 0.0886	359.97s	0.1993 ± 0.0805	42.91s

Figure 2: The generalisation error as a function of the bound value for a single train-test split of the Bupa data set. The final value of the function is the overall generalisation error.

6   Discussion

We have presented a novel approach for model-selection and test sample prediction using a nonconformity (strangeness) measure. Furthermore we have given a novel generalisation error bound on the loss of the learning method. The proposed model selection approach is both simple and gives consistent generalisation performance (Gold & Sollich, 2003).

We find these results encouraging as it constitutes a much needed shift from costly model selection based approaches to a faster method that is competitive in terms of generalisation error. Furthermore, in relation to the work of Özöğür-Akyüz et al. (In Press) we have presented a method that is 1) not restricted to SVMs and 2) can use measures other than the margin to make predictions. Therefore the nonconformity measure approach gives us a general way of choosing to make predictions, allowing us the flexibility to apply it to algorithms that are not based on large margins. In future work we aim to investigate the applicability of our proposed model selection technique to other learning methods. Another future research direction is to apply different nonconformity measures to the SVM algorithm presented in this paper such as, for example, a nearest neighbour nonconformity measure (Shafer & Vovk, 2008).

References

Ambroladze, A., Parrado-Hernández, E., & Shawe-Taylor, J. (2006). Tighter PAC-Bayes bounds. *Proceedings of Advances in Neural Information Processing Systems*.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (pp. 144–152). Pittsburgh ACM.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chapelle, O., & Vapnik, V. N. (1999). Model selection for support vector machines. *Proceedings of Advances in Neural Information Processing Systems 12* (pp. 230–237).

de Souza, B. F., de Carvalho, A. C. P. L. F., Calvo, R., & Ishii, R. P. (2006). Multiclass SVM model selection using particle swarm optimization. *Proceedings of the Sixth International Conference on Hybrid Intelligent Systems*.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. No. 31 in Applications of Mathematics. New York: Springer.

Gold, C., & Sollich, P. (2003). Model selection for support vector machine classification. *Neurocomputing*, *55*, 221–249.

Hastie, T., Rosset, S., Tibshirani, R., & Zhu, J. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, *5*, 1391–1415.

Kearns, M., & Ron, D. (1999). Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. *Neural Computation*, *11(6)*, 1427–1453.

Langford, J. (2005). Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, *6*, 273–306.

Li, H., Wang, S., & Qi, F. (2005). SVM model selection with the VC bound. *Computational and Information Science*, *3314*, 1067–1071.

Momma, M., & Bennett, K. P. (2002). Pattern search method for model selection of support vector regression. *Proceedings of the Second SIAM International Conference on Data Mining*.

Özöğür, S., Shawe-Taylor, J., Weber, G. W., & Ögel, Z. B. (2008). Pattern analysis for the prediction of fungal pro-peptide cleavage sites. *Discrete Applied Mathematics, Special Issue on Networks in Computational Biology*, *doi:10.1016/j.dam.2008.06.043*.

Özöğür-Akyüz, S., Hussain, Z., & Shawe-Taylor, J. (In Press). Prediction with the SVM using test point margins. *Annals of Information Systems, Special Issue on Optimization methods in Machine Learning*.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *Journal of Machine Learning Research*, *9*, 371–421.

Shawe-Taylor, J. (1998). Classification accuracy based on observed margin. *Algorithmica*, *22*, 157–172.

Shawe-Taylor, J., & Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge, U.K.: Cambridge University Press.

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world*. New York: Springer.

Zhang, T. (2001). A leave-one-out cross validation bound for kernel methods with application in learning. *Lecture Notes in Computer Science: 14th Annual Conference on Computation Learning Theory*, *2111*, 427–443.