Leave-One-Out Cross-Validation Based Model Selection Criteria for Weighted LS-SVMs

Gavin C. Cawley
School of Computing Sciences
University of East Anglia
Norwich NR4 7TJ
United Kingdom
E-mail: gcc@cmp.uea.ac.uk

***Abstract*— While the model parameters of many kernel learning methods are given by the solution of a convex optimisation problem, the selection of good values for the kernel and regularisation parameters, i.e. model selection, is much less straight-forward. This paper describes a simple and efficient approach to model selection for weighted least-squares support vector machines, and compares a variety of model selection criteria based on leave-one-out cross-validation. An external cross-validation procedure is used for performance estimation, with model selection performed independently in each fold to avoid selection bias. The best entry based on these methods was ranked in joint first place in the WCCI-2006 performance prediction challenge, demonstrating the effectiveness of this approach.**

I. Introduction

Kernel learning methods, such as the least-squares support vector machine (LS-SVM) [12] are attractive because they allow the construction of powerful non-linear classifiers, using only relatively simple mathematical and computational techniques. The model parameters of an LS-SVM are given by the solution of a system of linear equations, which can be found efficiently via Cholesky factorisation. The generalisation performance of the LS-SVM is however, heavily dependent on the *model selection* process, in this case the careful selection of an appropriate kernel function and good values for the regularisation and kernel parameters. This paper is concerned with model selection strategies based on minimisation of the leave-one-out cross-validation estimate of a range of model selection criteria, which can be performed very efficiently for this class of kernel learning methods. The aim of the WCCI-2006 Performance Prediction Challenge (PPC) is to identify accurate methods for predicting the performance of statistical classifiers on unseen test data, for use in model selection *and* model evaluation. The challenge takes place over a suite of five benchmark datasets, ADA, GINA, HIVA, NOVA and SYLVA, each having pre-defined training, validation and test partitions. The final performance assessment is based on a combination of the Balanced Error Rate (BER) of the classifier over the test partition and the accuracy of the predicted balanced error rate generated by the model selection procedure. A number of features of the performance prediction challenge warrant serious consideration, and are discussed in the remainder of this section.

A. The Balanced Error Rate (BER) Criterion

TABLE I
Confusion matrix for two-class pattern recognition.

		Prediction	
		$\mathcal{C}^-$	$\mathcal{C}^+$
Truth	$\mathcal{C}^-$	a	b
	$\mathcal{C}^+$	c	d

The Balanced Error Rate (BER) statistic is the average of the misclassification rates on examples drawn from positive and negative classes (denoted by $\mathcal{C}^+$ and $\mathcal{C}^-$ respectively), i.e.

$$BER = \frac{1}{2}\left[\frac{b}{a+b} + \frac{c}{c+d}\right],$$

where a, b, c and d are entries in the *confusion matrix* for a two-class pattern recognition problem, shown in Table I. Clearly the balanced error rate only coincides with the more traditional misclassification rate if there are an equal number of positive and negative examples, in which case $a+b = c+d$. However, the relative class frequencies in the performance prediction challenge benchmarks are skewed in favour of the negative class; in the case of the HIVA and SYLVA benchmarks the ratios are skewed rather heavily in favour of the negative class. We must therefore tailor our approach to account for the unequal weight assigned to false-negative and false-positive errors in the performance assessment criterion. This can be accomplished via a number of means, including altering the bias parameters of the classifier or differentially weighting positive and negative examples during the training procedure. Both of these approaches are investigated in this study.

B. Over-Fitting in the Model Selection Process

Let $G(\boldsymbol{\theta})$ represent the true test error of a classifier with parameters $\boldsymbol{\theta}$, and $g(\boldsymbol{\theta}|\mathcal{D})$ an estimate of the true test error based on a sample of data $\mathcal{D}$. In the context of model selection, $\mathcal{D}$, might refer to an independent *validation* set, or the set of validation partitions arising in (leave-one-out) cross-validation. The expected error of the estimator can be

broken down into *bias* and *variance* components [6],

$$\mathcal{E}_{\mathcal{D}}\left\{\left[g(\boldsymbol{\theta};\mathcal{D})-G(\boldsymbol{\theta})\right]^2\right\} = \mathcal{E}_{\mathcal{D}}\left\{g(\boldsymbol{\theta};\mathcal{D})-G(\boldsymbol{\theta})\right\}^2 + \mathcal{E}_{\mathcal{D}}\left\{\left[g(\boldsymbol{\theta};\mathcal{D}) - \mathcal{E}_{\mathcal{D}}\left\{g(\boldsymbol{\theta};\mathcal{D})\right\}\right]^2\right\}$$

where the expectations are taken over all data sets, $\mathcal{D}$, of fixed size. The first term, the squared *bias*, is low if on average the difference between the true test error and estimated error is small, i.e. the bias represents degree to which the estimated error *systematically* differs from the true test error. The second component, the *variance*, essentially reflects the sensitivity of the estimator to the particular choice of data over which it is evaluated. Note that the *variance* can normally be expected to fall as the size of the sample of data, $\mathcal{D}$, increases. The leave-one-out cross-validation estimator is known to be approximately unbiased [8]. This is a reassuring, but not essential, property for a model selection criterion as it suggests that *on average* the vector of model parameters minimising the model selection criterion are approximately the same as those minimising the true test error. However, the leave-one-out estimator generally exhibits a higher variance than, for example, the k-fold cross-validation estimator e.g. [7]. This is an undesirable property of a model selection criterion as the "optimal" parameters, $\boldsymbol{\theta}$, will be sensitive to the sample of data used. The relatively high variance of the leave-one-out estimator has a number of significant implications for the model selection process:

1) Over-fitting of the Model Selection Criterion: During the model selection process, the parameters $\boldsymbol{\theta}$ are iteratively modified so as to minimise the value of the model selection criterion. However, the value of the model selection criterion can be considered to comprise of two components, a component that is closely related to generalisation performance, and a component that is sensitive to the characteristics specific to the particular sample of data on which it is evaluated. It seems reasonable to expect that the largest reductions in the model selection criterion should come from changes in the parameters that result in a reduction in the true test error. If the number of parameters to be determined during model selection is relatively small, the model selection process is likely to be dominated by changes that genuinely improve generalisation. On the other hand if the number of parameters is relatively large, there may be sufficient degrees of freedom that the model selection becomes sensitive to the particular sample of data used, i.e. *over-fitting* will occur. It is therefore prudent to avoid the selection of a large number of parameters on the basis of a model selection criterion, unless it is known to have a low variance.

2) Feature Selection: Many of the challenge datasets are characterised by a large number of features relative to the number of training patterns. If feature selection were performed as part of the model selection process, this would vastly inflate the degrees of freedom available for over-fitting the model selection criterion, as there is essentially an extra degree of freedom associated with each feature. In practice, performing feature selection on the basis of leave-one-out cross-validation criteria often *degrades* generalisation performance in the presence of a large number of features due to over-fitting of the model selection criterion. Fortunately, least-squares support vector machines generally perform well in high-dimensional spaces, due to the use of formal regularisation [13], and so we are able to neglect a feature selection stage in this study.

3) Model Selection versus Performance Prediction: For performance evaluation purposes, we require a criterion that is both unbiased and also exhibits a low variance. Leave-one-out cross-validation based estimators, while approximately unbiased, are likely to be sub-optimal performance evaluation criteria due to their high variance[1]. However, in general, one should be cautious in using the same criterion for model selection and performance evaluation; any criterion that has been directly optimised during the model selection process is likely to result in a significantly optimistic estimate of the true generalisation performance, due to the variance of the estimator.

C. Reliability of Validation Set Statistics

The design of the performance prediction challenge benchmarks is such that the test set is approximately ten times larger than the training set, which in turn is approximately ten times larger than the validation set, with the ratio of positive and negative examples being closely matched in all three partitions of the available data. However, optimising the balanced error rate on the validation set, so as to achieve a good ranking in the *model development* stage of the challenge, is a risky strategy as the variance of the validation set *estimate* of the test BER is likely to be high, due to the small size of the validation set. In essence, this means it might be possible to "over-fit" the hyper-parameters of the model to the validation set. This is especially true for the `HIVA` benchmark, where the validation set contains only 14 positive examples in addition to the 370 negative examples. As the balanced error rate is very sensitive to errors in the minority class, the validation set BER will be very sensitive to the sampling of these positive examples. If fourteen "clearly positive" examples were chosen, the BER will be unrealistically low, if fourteen "difficult" examples were selected the BER will be unduly high. As there are only fourteen positive patterns, either of these scenarios could easily occur, in which case the validation set BER would be a poor predictor of the test set BER. In this study, we have therefore chosen to largely ignore the validation set performance, available from the challenge web-site, in favour of estimators likely to have a lower variance. It will be interesting to see, in hindsight, whether this was a good strategy at the conclusion of the challenge!

[1]Leave-one-out cross-validation is typically used in the analysis of very small datasets, where the relatively high variance of the leave-one-out estimator is offset by the stability resulting from the greater size of the training partition than is possible using conventional k-fold cross-validation.

II. METHOD

In this section, we give a brief overview of the Least-Squares Support Vector Machine (LS-SVM), including a weighted variant suitable for the performance prediction challenge, before going on to describe an efficient closed-form implementation of the leave-one-out cross-validation method for least-squares kernel learning methods. This forms the basis of a family of model selection procedures, based on the leave-one-out cross-validation estimates of a variety of model selection criteria.

A. Least-Squares Support Vector Machines

Assume we are given labelled training data, $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\ell}$, where $\boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is a vector of input features describing the i^{th} example and $y_i \in \{-1, +1\}$ is an indicator variable such that $y_i = -1$ if the i^{th} example is drawn from class $\mathcal{C}^-$ and $y_i = +1$ is drawn from class $\mathcal{C}^+$. The Least-Squares Support Vector Machine (LS-SVM) aims to construct a linear model $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}) + b$ in a fixed feature space, $\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F}$, that is able to distinguish between examples drawn from $\mathcal{C}^-$ and $\mathcal{C}^+$, such that

$$\boldsymbol{x} \in \begin{cases} \mathcal{C}^+ & \text{if } f(\boldsymbol{x}) \geq 0 \\ \mathcal{C}^- & \text{otherwise} \end{cases}.$$

However, rather than specifying the feature space, $\mathcal{F}$ directly, it is implied by a kernel function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, giving the inner product between the images of vectors in the feature space, $\mathcal{F}$, i.e. $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$. A common kernel function is the isotropic Radial Basis Function (RBF) kernel

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{-\eta \|\boldsymbol{x} - \boldsymbol{x}'\|^2\right\}, \tag{1}$$

where η is a kernel parameter controlling the sensitivity of the kernel function. Other useful kernels include the linear,

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}' \tag{2}$$

and polynomial kernels

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}' + c)^d \tag{3}$$

where c and d are kernel parameters ($d = 2$ gives the quadratic kernel and $d = 3$ the cubic kernel) in addition to the Boolean kernel

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = (1 + \eta)^{\boldsymbol{x} \cdot \boldsymbol{x}'}.$$

The model parameters $(\boldsymbol{w}, b)$ are given by the minimum of a regularised [13] least-squares loss function,

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\ell\mu}\sum_{i=1}^{\ell}\left[y_i - \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) - b\right]^2, \tag{4}$$

where μ is a regularisation parameter controlling the bias-variance trade-off [6]. The accuracy of an LS-SVM on test data is critically dependent on the choice of good values for the *hyper-parameters*, in this case μ and η. The search for the optimal values for such hyper-parameters is a process known as *model selection*.

B. Training Algorithm

The regularised least-squares problem (4) can be solved via a system of linear equations, with a computational complexity of $\mathcal{O}(\ell^3)$ operations, as follows: Minimising (4) can be recast in the form of a constrained optimisation problem,

$$\min \mathcal{J} = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\ell\mu}\sum_{i=1}^{\ell}\varepsilon_i^2 \tag{5}$$

subject to

$$y_i = \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) + b + \varepsilon_i, \qquad \forall\ i \in \{1, 2, \ldots, \ell\}. \tag{6}$$

The primal Lagrangian for this optimisation problem gives the unconstrained minimisation problem,

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\ell\mu}\sum_{i=1}^{\ell}\varepsilon_i^2 - \sum_{i=1}^{\ell}\alpha_i\left\{\boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) + b + \varepsilon_i - y_i\right\},$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_\ell) \in \mathbb{R}^\ell$ is a vector of Lagrange multipliers. The optimality conditions for this problem can be expressed as follows:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \boldsymbol{0} \implies \boldsymbol{w} = \sum_{i=1}^{\ell}\alpha_i\boldsymbol{\phi}(\boldsymbol{x}_i) \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \sum_{i=1}^{\ell}\alpha_i = 0 \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial \varepsilon_i} = 0 \implies \alpha_i = \frac{\varepsilon_i}{\ell\mu}, \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 \implies \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) + b + \varepsilon_i - y_i = 0. \tag{10}$$

Using (7) and (9) to eliminate $\boldsymbol{w}$ and $\boldsymbol{\varepsilon} = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_\ell)$, from (10), we find that

$$\sum_{j=1}^{\ell}\alpha_j\boldsymbol{\phi}(\boldsymbol{x}_j) \cdot \boldsymbol{\phi}(\boldsymbol{x}_i) + b + \ell\mu\alpha_i = y_i. \tag{11}$$

Noting that $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$, the system of linear equations can be written more concisely in matrix form as

$$\begin{bmatrix} \boldsymbol{K} + \mu\ell\boldsymbol{I} & \boldsymbol{1} \\ \boldsymbol{1}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{t} \\ 0 \end{bmatrix}. \tag{12}$$

From (7) and noting that $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$, the output of the LS-SVM can be written in terms of the *dual* model parameters, $(\boldsymbol{\alpha}, b)$, as

$$f(\boldsymbol{x}) = \sum_{i=1}^{\ell}\alpha_i\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

C. Efficient Implementation Via Cholesky Factorisation

A more efficient training algorithm can be obtained, taking advantage of the special structure of the system of linear equations [12]. The system of linear equations (12) to be solved in fitting a least-squares support vector machine can be written as

$$\begin{bmatrix} \boldsymbol{M} & \boldsymbol{1} \\ \boldsymbol{1}^T & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{t} \\ 0 \end{bmatrix}, \tag{13}$$

where $\boldsymbol{M} = \boldsymbol{K} + \mu\ell\boldsymbol{I}$. Unfortunately the matrix on the left-hand side is not positive definite, and so we cannot solve this system of linear equations directly using the Cholesky factorisation. However, the first row of (13) can be re-written as

$$\boldsymbol{M}\left(\boldsymbol{\alpha} + \boldsymbol{M}^{-1}\mathbf{1}b\right) = \boldsymbol{y} \tag{14}$$

Rearranging (14), we see that $\boldsymbol{\alpha} = \boldsymbol{M}^{-1}\left(\boldsymbol{y} - \mathbf{1}b\right)$, using this result to eliminate α, the second row of (13) can be written as,

$$\mathbf{1}^T\boldsymbol{M}^{-1}\mathbf{1}b = \mathbf{1}^T\boldsymbol{M}^{-1}\boldsymbol{y} \tag{15}$$

The system of linear equations can then be re-written as

$$\left[\begin{array}{cc} \boldsymbol{M} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{1}^T\boldsymbol{M}^{-1}\mathbf{1} \end{array}\right]\left[\begin{array}{c} \boldsymbol{\alpha} + \boldsymbol{M}^{-1}\mathbf{1}b \\ b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ \mathbf{1}^T\boldsymbol{M}^{-1}\boldsymbol{y} \end{array}\right].$$

In this case, the matrix on the left hand side is positive-definite, as $\boldsymbol{M} = \boldsymbol{K} + \mu\ell\boldsymbol{I}$ is positive-definite and $\mathbf{1}^T\boldsymbol{M}^{-1}\mathbf{1}$ is positive since the inverse of a positive definite matrix is also positive definite. The revised system of linear equations can then be solved as follows: First solve

$$\boldsymbol{M}\boldsymbol{\eta} = \mathbf{1} \qquad \text{and} \qquad \boldsymbol{M}\boldsymbol{\nu} = \boldsymbol{y}, \tag{16}$$

The model parameters of the least-squares support vector machine are then given by

$$b = \frac{\mathbf{1}^T\boldsymbol{\nu}}{\mathbf{1}^T\boldsymbol{\eta}} \qquad \text{and} \qquad \boldsymbol{\alpha} = \boldsymbol{\nu} - \boldsymbol{\eta}b.$$

The two systems of linear equations (16) can be solved efficiently using the Cholesky decomposition of $\boldsymbol{M} = \boldsymbol{R}^T\boldsymbol{R}$, where $\boldsymbol{R}$ is the upper triangular Cholesky factor of $\boldsymbol{M}$.

D. Weighted Least-Squares Support Vector Machines

For some applications, it may be preferable find the model parameters $(\boldsymbol{w}, b)$ via minimisation of a regularised *weighted* least-squares loss function [12],

$$\mathcal{L} = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{2\mu\ell}\sum_{i=1}^{\ell}\zeta_i\left[y_i - \boldsymbol{w}\cdot\boldsymbol{\phi}(\boldsymbol{x}_i) - b\right]^2,$$

where $\boldsymbol{\zeta} = \{\zeta_1, \zeta_2, \ldots, \zeta_\ell\}$ is a vector of weights associated with each pattern. The optimal dual model parameters, $(\boldsymbol{\alpha},\ b)$ are then given by the solution of a modified system of linear equations,

$$\left[\begin{array}{cc} \boldsymbol{K} + \mu\ell\boldsymbol{W} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{array}\right]\left[\begin{array}{c} \boldsymbol{\alpha} \\ b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ 0 \end{array}\right], \tag{17}$$

where $\boldsymbol{W} = \operatorname{diag}\left\{\left(\zeta_1^{-1}, \zeta_2^{-1}, \ldots, \zeta_\ell^{-1}\right)\right\}$. The most common situation in which a weighted loss function is used is where the proportions of positive and negative examples in the training data are known not to be representative of the operational class frequencies. A weighted loss function is also appropriate if we wish to minimise the balanced error rate, in order to balance the contribution of the sets of positive and negative examples to the data misfit term of the regularised loss function. In this case, the weighting factors should be chosen according to

$$\zeta_i = \left\{\begin{array}{ll} \frac{\ell}{2\ell^+} & \text{if } t_i = +1 \\ \frac{\ell}{2\ell^-} & \text{otherwise} \end{array}\right. \tag{18}$$

where ℓ^+ and ℓ^- represent the number of positive and negative examples respectively. Note that this is asymptotically equivalent to re-sampling the data so that there are an equal number of positive and negative examples (c.f. [4]).

E. Efficient Leave-One-Out Cross-Validation

The optimal values of the parameters of a Least-Squares Support Vector Machine are given by the solution of a system of linear equations (12), the matrix on the left-hand side of which can be decomposed into block-matrix representation, as follows:

$$\left[\begin{array}{cc} \boldsymbol{K} + \mu\ell\boldsymbol{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{array}\right] = \left[\begin{array}{cc} c_{11} & \boldsymbol{c}_1^T \\ \boldsymbol{c}_1 & \boldsymbol{C}_1 \end{array}\right] = \boldsymbol{C}.$$

Let $[\boldsymbol{\alpha}^{(-i)}; b^{(-i)}]$ represent the parameters of the least-squares support vector machine during the i^{th} iteration of the leave-one-out cross-validation procedure, then in the first iteration, in which the first training pattern is excluded,

$$\left[\begin{array}{c} \boldsymbol{\alpha}^{(-1)} \\ b^{(-1)} \end{array}\right] = \boldsymbol{C}_1^{-1}\left[y_2, \ldots, y_\ell, 0\right]^T.$$

The leave-one-out prediction for the first training pattern is then given by,

$$\hat{y}_1^{(-1)} = \boldsymbol{c}_1^T\left[\begin{array}{c} \boldsymbol{\alpha}^{(-1)} \\ b^{(-1)} \end{array}\right] = \boldsymbol{c}_1^T\boldsymbol{C}_1^{-1}\left[y_2, \ldots, y_\ell, 0\right]^T$$

Considering the last ℓ equations in the system of linear equations (12), it is clear that $\left[\boldsymbol{c}_1\ \boldsymbol{C}_1\right]\left[\alpha_2, \ldots, \alpha_\ell, b\right]^T = \left[y_2, \ldots, y_\ell, 0\right]^T$, and so

$$\begin{aligned} \hat{y}_1^{(-1)} &= \boldsymbol{c}_1^T\boldsymbol{C}_1^{-1}\left[\boldsymbol{c}_1\ \boldsymbol{C}_1\right]\left[\boldsymbol{\alpha}^T, b\right]^T \\ &= \boldsymbol{c}_1^T\boldsymbol{C}_1^{-1}\boldsymbol{c}_1\alpha_1 + \boldsymbol{c}_1\left[\alpha_2, \ldots, \alpha_\ell, b\right]^T. \end{aligned}$$

Noting, from the first equation in the system of linear equations (12), that $y_1 = c_{11}\alpha_1 + \boldsymbol{c}_1^T\left[\alpha_2, \ldots, \alpha_\ell, b\right]^T$, thus

$$\hat{y}_1^{(-1)} = y_1 - \alpha_1\left(c_{11} - \boldsymbol{c}_1^T\boldsymbol{C}_1^{-1}\boldsymbol{c}_1\right)$$

Finally, via the block matrix inversion lemma,

$$\left[\begin{array}{cc} c_{11} & \boldsymbol{c}_1^T \\ \boldsymbol{c}_1 & \boldsymbol{C}_1 \end{array}\right]^{-1} = \left[\begin{array}{cc} \kappa^{-1} & -\kappa^{-1}\boldsymbol{c}_1\boldsymbol{C}_1^{-1} \\ \boldsymbol{C}_1^{-1} + \kappa^{-1}\boldsymbol{C}_1^{-1}\boldsymbol{c}_1^T\boldsymbol{c}_1\boldsymbol{C}_1^{-1} & -\kappa^{-1}\boldsymbol{C}_1^{-1}\boldsymbol{c}_1^T \end{array}\right],$$

where $\kappa = c_{11} - \boldsymbol{c}_1^T\boldsymbol{C}_1^{-1}\boldsymbol{c}$, and noting that the system of linear equations (12) is insensitive to permutations of the ordering of the equations and of the unknowns, we have that,

$$r_i^{(-i)} = y_i - \hat{y}_i^{(-i)} = \frac{\alpha_i}{\boldsymbol{C}_{ii}^{-1}}. \tag{19}$$

This means that, assuming the system of linear equations is solved via explicit inversion of $\boldsymbol{C}$, a leave-one-out cross-validation estimate of an appropriate model selection criterion can be evaluated using information already available as a by-product of training the least-squares support vector machine on the entire dataset, with only a negligible additional computational expense.

F. Efficient Implementation via Cholesky Factorisation

The coefficients of the kernel expansion, $\boldsymbol{\alpha}$, can be found efficiently, via Cholesky factorisation, as described in Section II-C. However, in order to perform the efficient leave-one-out cross-validation procedure, we must also determine the diagonal elements of $\boldsymbol{C}^{-1}$ in an efficient manner. Using the block matrix inversion formula, we obtain

$$\boldsymbol{C}^{-1} = \begin{bmatrix} \boldsymbol{M}^{-1} + \boldsymbol{M}^{-1}\mathbf{1}\boldsymbol{S}_M^{-1}\mathbf{1}^T\boldsymbol{M}^{-1} & -\boldsymbol{M}^{-1}\mathbf{1}\boldsymbol{S}_M^{-1} \\ -\boldsymbol{S}_M^{-1}\mathbf{1}^T\boldsymbol{M}^{-1} & \boldsymbol{S}_M^{-1} \end{bmatrix},$$

where $\boldsymbol{M} = \boldsymbol{K} + \mu\ell\boldsymbol{I}$ and $\boldsymbol{S}_M = -\mathbf{1}^T\boldsymbol{M}^{-1}\mathbf{1} = -\mathbf{1}^T\boldsymbol{\eta}$ is the Schur complement of $\boldsymbol{M}$. The inverse of the positive definite matrix, $\boldsymbol{M}$, can be computed efficiently from its Cholesky factorisation, via the `SYMINV` algorithm [11], for example using the LAPACK routine `DTRTRI` [2]. Let $\boldsymbol{R} = [r_{ij}]_{i,j=1}^n$ be the lower triangular Cholesky factor of the positive definite matrix $\boldsymbol{M}$, such that $\boldsymbol{M} = \boldsymbol{R}\boldsymbol{R}^T$. Furthermore, let

$$\boldsymbol{S} = [s_{ij}]_{i,j=1}^n = \boldsymbol{R}^{-1},$$

where

$$s_{ii} = \frac{1}{r_{ii}} \qquad \text{and} \qquad s_{ij} = -s_{ii}\sum_{k=1}^{i-1} r_{ik}s_{kj},$$

represent the (lower triangular) inverse of the Cholesky factor. The inverse of $\boldsymbol{M}$ is then given by $\boldsymbol{M}^{-1} = \boldsymbol{S}^T\boldsymbol{S}$. In the case of efficient leave-one-out cross-validation of least-squares support vector machines, we are principally concerned only with the diagonal elements of $\boldsymbol{M}^{-1}$, given by

$$M_{ii}^{-1} = \sum_{j=1}^{i} s_{ij}^2 \qquad \Longrightarrow \qquad C_{ii}^{-1} = \sum_{j=1}^{i} s_{ij}^2 + \frac{\eta_i^2}{S_M}$$

The computational complexity of the basic training algorithm is $\mathcal{O}(\ell^3)$ operations, being dominated by the evaluation of the Cholesky factor. However, the computational complexity of the analytic leave-one-out cross-validation procedure, when performed as a by-product of the training algorithm, is only $\mathcal{O}(\ell)$ operations. The computational expense of the leave-one-out cross-validation procedure therefore becomes increasingly negligible as the training set becomes larger.

G. Model Selection Criteria

While the optimal model parameters of the LS-SVM are given by the solution of a simple system of linear equations, (12) or (17), some form of model selection is required to determine good values for the *hyper-parameters*, $\boldsymbol{\theta} = (\mu, \boldsymbol{\eta})$ in order to maximise generalisation performance. The analytic leave-one-out cross-validation procedure described in the previous section can easily form the basis of an efficient model selection strategy [5] based on a weighted version of Allen's predicted residual sum-of-squares (PRESS) statistic [1],

$$\mathrm{PRESS}(\boldsymbol{\theta}) = \sum_{i=1}^{\ell} \zeta_i \left\{ r_i^{(-i)} \right\}^2.$$

However the PRESS statistic is best suited to regression problems, and more sophisticated model selection criterion may be preferable in the context of statistical pattern recognition. For instance, the leave-one-out cross-validation estimate of the weighted error rate is given by

$$\mathrm{ERROR}(\boldsymbol{\theta}) = \sum_{i=1}^{\ell} \zeta_i \Psi\left\{ t_i r_i^{(-i)} - 1 \right\}$$

where $\Psi\{\cdot\}$ is the unit step function,

$$\Psi\left\{x\right\} = \begin{cases} 1 & \text{if } x >= 0 \\ 0 & \text{otherwise} \end{cases}.$$

The leave-one-out estimate of the balanced error rate (BER) is obtained by setting the weighting coefficients to give equal weight to the sets of positive and negative examples, that is according to (18). The leave-one-out balanced error rate ought to provide a good model selection criterion for the performance prediction challenge as the balanced error rate over the test set forms the major component of the final ranking criterion. However, while the leave-one-out estimate of the BER provides a reasonable performance estimate for the purposes of the challenge, it is not entirely suitable for model *selection* purposes, as we would prefer a continuous function that is more amenable to numerical optimisation routines. One approach would be to approximate the discontinuous unit step function by a continuous approximation, such as the logistic function [3],

$$\widetilde{\Psi}\{\boldsymbol{x}\} = \frac{1}{1 + \exp\{-\gamma x\}}, \tag{20}$$

where γ is a parameter governing the accuracy of the approximation. Alternatively, we may opt for an upper bound on the balanced error rate, obtained by substituting the weighted hinge loss for the step function,

$$\mathrm{HINGE}(\boldsymbol{\theta}) = \sum_{i=1}^{\ell} \zeta_i \left[t_i r_i^{(-i)} \right]_+$$

or the weighted squared hinge loss,

$$\mathrm{HINGE}^2(\boldsymbol{\theta}) = \sum_{i=1}^{\ell} \zeta_i \left[t_i r_i^{(-i)} \right]_+^2$$

where $[x]_+ = \max\{0, x\}$ (see Figure 1). A final model selection criterion is concerned only with the quality of the relative *ranking* of patterns under leave-one-out cross-validation, via maximising the area under the receiver operating characteristic (AUC). Equivalently, one could instead minimise the *modified* Wilcoxon-Mann-Whitney [9, 14] statistic,

$$\mathrm{WMW}(\boldsymbol{\theta}) = \frac{1}{\ell^+\ell^-} \sum_{i:y_i=+1} \sum_{j:y_j=-1} \Psi\left\{ \hat{y}_i^{(-i)} - \hat{y}_j^{(-j)} \right\},$$

where again, the smooth approximation to the step function (20) can be employed to obtain a continuous selection criterion. The hyper-parameters of the (weighted) LS-SVM, $\boldsymbol{\theta}$, can then be optimised by minimisation of any of these

model selection criteria via, for example, the Nelder-Mead simplex [10] method, as implemented by the `fminsearch` routines of the MATLAB Optimisation Toolbox.

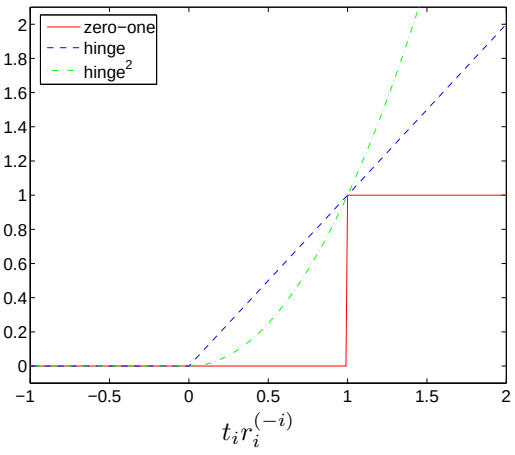

Fig. 1. The hinge and squared hinge loss bounds on the zero-one loss.

III. The Challenge Benchmark Datasets

Table III shows summary information on each of the five challenge benchmark datasets. The class ratios for the HIVA and SYLVA benchmarks are highly skewed, with a very low prior probability for the positive class $P(\mathcal{C}_+)$. The GINA, HIVA and especially the NOVA benchmarks also have a very large number of features, given the number of training examples. Note however that the number of features that have a non-zero variance over the training set, d_{nc}, is significantly less than the total number of features. Obviously features that have zero variance over the training set are uninformative and can safely be omitted from the analysis. Many of the benchmarks also include a large number of binary features, with a high degree of sparsity.

TABLE II
Summary of the dimensions and composition of the five challenge benchmark datasets.

	ADA	**GINA**	**HIVA**	**NOVA**	**SYLVA**
ℓ^+	1029	1550	135	499	805
ℓ^-	3118	1603	3710	1255	12281
$P(\mathcal{C}_+)$	0.248	0.492	0.035	0.285	0.062
d	48	970	1617	16969	216
$d_{\rm nc}$	46	970	1617	12398	211
sparsity	72.1%	0.0%	91.8%	98.1%	76.7%

The following pre-processing steps were taken for each benchmark dataset: ADA - logarithmic transform of features 1 and 3, features 4 and 5 discretized via thresholding at 44 and respectively, standardisation of continuous features. GINA - all features scaled by 255^{-1}. HIVA - no pre-processing required. NOVA - no pre-processing required. SYLVA - standardisation of continuous features, reduction of training set using features never associated with the positive class.

IV. Results

The aim of this study is to evaluate a range of criteria for leave-one-out cross-validation based model selection of weighted least-squares support vector machines. A 100-fold validation approach was used in order to obtain a low-variance estimator of the true test balanced error rate. In each of 100 trials, the data are randomly partitioned into a training set containing approximately 90% of the available data and a test set containing the remaining patterns. Model selection is performed independently in each trial via minimisation of a leave-one-out model selection criterion via the Nelder-Mead simplex optimisation method [10]. A total of 70 experiments were performed, based on different combinations of model selection criteria, kernel function and the use of weighting factors in the training and/or model selection procedures. The results of these experiments are shown in Table III. The best performance on each benchmark are shown in bold.

Table IV shows the weights resulting from a regression analysis of the data given in Table III. The 100-fold validation estimates of the test balanced error rate were standardised to have a zero mean and unit variance. A linear least-squares model was then used to predict the estimate of the test balanced error rate using boolean features representing the choice of model selection criterion (PRESS, HINGE^1, HINGE^2, WMW and ERATE), the use of weighting factors during training and model selection (Training and Selection respectively) and the choice of kernel function (Linear, Quadratic, Cubic, Boolean and RBF). The results suggest that the use of weighting factors in training and/or model selection does not confer a significant advantage and that, unsurprisingly, the choice of kernel is data dependent. The choice of model selection criterion also seems data dependent, but that relatively good performance can be achieved using the simple PRESS statistic, even though this is better suited to regression problems, obviating the need to employ a more complex criteria.

Three final submissions have been made to the WCCI-2006 performance prediction challenge website. The first, shown in Table V, consists of models selected for each benchmark dataset on the basis of the leave-one-out estimate of the balanced error rate, which is also used as the final performance estimate. In this case, it would be reasonable to expect that the predicted balanced error rate will be unduly low as the estimator has also been used as the model selection criterion.

Table VI shows the second final submission. In this case the final model choice is based on the leave-one-out cross-validation estimate of the balanced error rate, but the performance estimate is based on an independent 100-fold validation estimate. This represents, in the author's opinion, the best practice methodology as the performance estimate has not been biased by the model selection process in any

TABLE III

ESTIMATE OF THE TEST BALANCED ERROR RATE BASED ON 100-FOLD VALIDATION FOR THE WCCI-2006 PERFORMANCE PREDICTION CHALLENGE BENCHMARKS FOR A VARIETY OF LEAVE-ONE-OUT CROSS-VALIDATION BASED MODEL SELECTION CRITERIA.

Experiment	Selection Criterion	Weighted Training	Weighted Selection	Kernel	ADA	GINA	HIVA	NOVA	SYLVA
01	PRESS	no	no	Linear	0.1766	0.1366	0.2727	0.0524	0.0156
02	PRESS	no	no	Quadratic	0.1687	0.0562	0.2769	0.0574	0.0098
03	PRESS	no	no	Cubic	0.1701	0.0482	0.2549	0.0653	0.0100
04	PRESS	no	no	Boolean	**0.1649**	0.0550	0.2622	0.0669	0.1155
05	PRESS	no	no	RBF	0.1676	0.0542	**0.2530**	0.0700	0.1160
06	PRESS	yes	no	Linear	0.1740	0.1360	0.2724	0.0532	0.0124
07	PRESS	yes	no	Quadratic	0.1740	0.0562	0.2848	0.0551	0.0101
08	PRESS	yes	no	Cubic	0.1685	**0.0481**	0.2658	0.0643	0.0102
09	PRESS	yes	no	Boolean	0.1763	0.0560	0.2654	0.0686	0.1151
10	PRESS	yes	no	RBF	0.1754	0.0533	0.2754	0.0664	0.1.15
11	PRESS	yes	yes	Linear	0.1757	0.1354	0.2830	0.0506	0.0236
12	PRESS	yes	yes	Quadratic	0.1691	0.0550	0.2965	0.0536	0.0098
13	PRESS	yes	yes	Cubic	0.1681	0.0499	0.2654	0.0627	0.0106
14	PRESS	yes	yes	Boolean	0.1736	0.0528	0.2828	0.0663	0.1153
15	PRESS	yes	yes	RBF	0.1670	0.0533	0.2972	0.0670	0.1148
16	HINGE[1]	no	no	Linear	0.1880	0.1366	0.2906	0.0753	0.0670
17	HINGE[1]	no	no	Quadratic	0.1721	0.0593	0.2741	0.0828	0.0106
18	HINGE[1]	no	no	Cubic	0.1725	0.0515	0.2551	0.0794	0.0099
19	HINGE[1]	no	no	Boolean	0.1723	0.0557	0.2563	0.0938	0.1150
20	HINGE[1]	no	no	RBF	0.1751	0.0560	0.2677	0.0897	0.1153
21	HINGE[1]	yes	no	Linear	0.1876	0.1398	0.2824	0.0688	0.0477
22	HINGE[1]	yes	no	Quadratic	0.1765	0.0547	0.2740	0.0796	0.0115
23	HINGE[1]	yes	no	Cubic	0.1737	0.0494	0.2549	0.0815	0.0099
24	HINGE[1]	yes	no	Boolean	0.1976	0.0540	0.2615	0.0887	0.1159
25	HINGE[1]	yes	no	RBF	0.1972	0.0560	0.2817	0.0954	0.1152
26	HINGE[1]	yes	yes	Linear	0.1974	0.1373	0.2699	0.0567	0.0721
27	HINGE[1]	yes	yes	Quadratic	0.1699	0.0550	0.2757	0.0706	0.0100
28	HINGE[1]	yes	yes	Cubic	0.1738	0.0497	0.2784	0.0820	0.0107
29	HINGE[1]	yes	yes	Boolean	0.1733	0.0527	0.2753	0.0835	0.1155
30	HINGE[1]	yes	yes	RBF	0.1716	0.0548	0.2693	0.0829	0.1153
31	HINGE[2]	no	no	Linear	0.1783	0.1347	0.2670	0.0546	0.0155
32	HINGE[2]	no	no	Quadratic	0.1667	0.0562	0.2628	0.0600	0.0114
33	HINGE[2]	no	no	Cubic	0.1701	0.0491	0.2581	0.0663	0.0094
34	HINGE[2]	no	no	Boolean	0.1689	0.0562	0.2603	0.0738	0.1152
35	HINGE[2]	no	no	RBF	0.1686	0.0517	0.2613	0.0720	0.1150
36	HINGE[2]	yes	no	Linear	0.1801	0.1343	0.2723	0.0557	0.0130
37	HINGE[2]	yes	no	Quadratic	0.1667	0.0538	0.2689	0.0559	0.0107
38	HINGE[2]	yes	no	Cubic	0.1661	0.0504	0.2639	0.0645	0.0103
39	HINGE[2]	yes	no	Boolean	0.1929	0.0538	0.2649	0.0873	0.1153
40	HINGE[2]	yes	no	RBF	0.1818	0.0539	0.2767	0.0677	0.1152
41	HINGE[2]	yes	yes	Linear	0.1747	0.1362	0.2797	0.0488	0.0275
42	HINGE[2]	yes	yes	Quadratic	0.1694	0.0556	0.2863	0.0524	0.0101
43	HINGE[2]	yes	yes	Cubic	0.1670	0.0522	0.2588	0.0626	0.0096
44	HINGE[2]	yes	yes	Boolean	0.1717	0.0523	0.2936	0.0680	0.1149
45	HINGE[2]	yes	yes	RBF	0.1681	0.0531	0.2860	0.0667	0.1153
46	WMW	no	no	Linear	0.1691	0.1366	0.2749	0.0502	0.0109
47	WMW	no	no	Quadratic	0.1700	0.0549	0.2621	0.0485	**0.0091**
48	WMW	no	no	Cubic	0.1688	0.0517	0.2786	0.0584	0.0102
49	WMW	no	no	Boolean	0.1672	0.0526	0.2660	0.0633	0.1167
50	WMW	no	no	RBF	0.1692	0.0544	0.2675	0.0666	0.1152
51	WMW	yes	no	Linear	0.1769	0.1384	0.2740	0.0509	0.0122
52	WMW	yes	no	Quadratic	0.1727	0.0543	0.2632	**0.0447**	0.0095
53	WMW	yes	no	Cubic	0.1675	0.0497	0.2784	0.0606	0.0097
54	WMW	yes	no	Boolean	0.1746	0.0543	0.2638	0.0612	0.1159
55	WMW	yes	no	RBF	0.1714	0.0536	0.2735	0.0614	0.1160
56	ERATE	no	no	Linear	0.1748	0.1346	0.2745	0.0537	0.0112
57	ERATE	no	no	Quadratic	0.1681	0.0542	0.2627	0.0522	0.0096
58	ERATE	no	no	Cubic	0.1682	0.0503	0.2665	0.0590	0.0107
59	ERATE	no	no	Boolean	0.1709	0.0529	0.2571	0.0701	0.1159
60	ERATE	no	no	RBF	0.1685	0.0525	0.2711	0.0723	0.1157
61	ERATE	yes	no	Linear	0.1821	0.1339	0.2735	0.0534	0.0844
62	ERATE	yes	no	Quadratic	0.1689	0.0560	0.2839	0.0514	0.0115
63	ERATE	yes	no	Cubic	0.1716	0.0484	0.2624	0.0640	0.0106
64	ERATE	yes	no	Boolean	0.1827	0.0540	0.2628	0.0640	0.1160
65	ERATE	yes	no	RBF	0.1824	0.0527	0.2797	0.0657	0.1155
66	ERATE	yes	yes	Linear	0.1765	0.1329	0.2745	0.0515	0.0116
67	ERATE	yes	yes	Quadratic	0.1728	0.0529	0.2742	0.0542	0.0095
68	ERATE	yes	yes	Cubic	0.1684	0.0495	0.2740	0.0635	0.0101
69	ERATE	yes	yes	Boolean	0.1721	0.0532	0.2807	0.0639	0.1148
70	ERATE	yes	yes	RBF	0.1710	0.0525	0.2787	0.0652	0.1159

TABLE IV

WEIGHTS OBTAINED BY REGRESSION ANALYSIS OF 100-FOLD VALIDATION ESTIMATE OF THE TEST BALANCED ERROR RATE.

Factor	ADA	GINA	HIVA	NOVA	SYLVA
PRESS	-0.4729	+0.0049	-0.1077	-0.2036	-0.0615
HINGE^1	+0.6871	+0.0375	-0.3774	+1.4446	+0.1203
HINGE^2	-0.2796	-0.0005	-0.4189	+0.0037	-0.0554
WMW	-0.6645	+0.0082	-0.1184	-0.7283	-0.0830
ERATE	-0.2087	-0.0265	-0.3169	-0.2913	+0.0165
Training	+0.8832	-0.0085	+0.4943	-0.0806	+0.0420
Selection	-0.7922	-0.0132	+0.8001	-0.3271	-0.0236
Linear	+0.5679	+1.9856	+0.1422	-0.7780	-0.5275
Quadratic	-0.6471	-0.4272	+0.0343	-0.5183	-0.9257
Cubic	-0.7513	-0.5911	-0.8821	+0.1824	-0.9274
Boolean	+0.0629	-0.4682	-0.6189	+0.7011	+1.1598
RBF	-0.1711	-0.4754	-0.0149	+0.6379	+1.1577

TABLE V

PERFORMANCE OF THE FIRST FINAL SUBMISSION, MODEL CHOICE AND PERFORMANCE ESTIMATION BASED ON LEAVE-ONE-OUT BER.

Dataset	Balanced Error			Guess	Guess	Test
	Train	Valid	Test		Error	Score
ADA	0.1490	0.1542	0.1845	0.1683	0.0162	0.2007
GINA	0.0000	0.0000	0.0461	0.0434	0.0027	0.0485
HIVA	0.0180	0.0216	0.2804	0.2475	0.0329	0.3131
NOVA	0.0000	0.0000	0.0445	0.0436	0.0009	0.0448
SYLVA	0.0028	0.0029	0.0067	0.0048	0.0018	0.0085
Overall	0.0340	0.0357	0.1124	0.1105	0.0034	0.1152

way. Note that the guess error for this method is very much lower.

TABLE VI

SECOND FINAL SUBMISSION, MODEL CHOICE VIA LEAVE-ONE-OUT BER, PERFORMANCE ESTIMATION VIA 100-FOLD VALIDATION BER.

Dataset	Balanced Error			Guess	Guess	Test
	Train	Valid	Test		Error	Score
ADA	0.1490	0.1542	0.1845	0.1742	0.0103	0.1947
GINA	0.0000	0.0000	0.0461	0.0470	0.0009	0.0466
HIVA	0.0180	0.0216	0.2804	0.2776	0.0028	0.2814
NOVA	0.0000	0.0000	0.0445	0.0470	0.0025	0.0464
SYLVA	0.0028	0.0029	0.0067	0.0065	0.0002	0.0067
Overall	0.0340	0.0357	0.1124	0.1105	0.0034	0.1152

V. CONCLUSIONS

In this study, we have investigated a variety of model selection criteria for (weighted) least-squares support vector machines, based on leave-one-out cross-validation estimators. A useful conclusion that may be drawn from the results obtained suggests that the optimal choice of model selection criterion is data dependent (and we cannot know *a-priori* which will perform best) and so it is reasonable to use a simple, mathematically tractable criterion, such as Allen's PRESS statistic. This study generated the joint winning entry in the challenge, finishing first in terms of average score and second in terms of average ranking. The best model also exhibited the second highest area under the receiver operating characteristic on the test set. The study also generated two individual data set winners (HIVA and NOVA). This demonstrates that leave-one-out cross-validation provides an effective means of model selection for least-squares support vector machines, but that an external means of performance estimation is required. If performance evaluation is performed using cross-validation, it is important that the model selection process is performed separately in each trial in order to avoid selection bias.

ACKNOWLEDGEMENTS

The author would like to thank the organisers of the WCCI-2006 performance prediction challenge, the anonymous reviewers for their helpful comments and Nicola Talbot for her help in typesetting the manuscript.

REFERENCES

[1] D. M. Allen. The relationship between variable selection and prediction. *Technometrics*, 16:125–127, 1974.

[2] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. W. Demmel, J. J. Dongarra, J. du Croz, A. Greenbaum, S. Hammarling, and A. McKenney. *LAPACK users' guide (software, environments, tools)*. SIAM Press, 1999.

[3] L. Bo, L. Wang, and L. Jiao. Multiple parameter selection for LS-SVM using smooth leave-one-out error. In *Proceedings of the International Symposium on Neural Networks*, volume 1 of *Lecture Notes in Computer Science*, pages 851–856, Chongquing, China, May 30 – June 1 2005. Springer.

[4] G. C. Cawley and N. L. C. Talbot. Manipulation of prior probabilities in support vector classification. In *Proceedings of the IEEE/INNS International Joint Conference on Neural Networks (IJCNN-2001)*, pages 2433–2438, Washington, DC, USA, July 15–19 2001.

[5] G. C. Cawley and N. L. C. Talbot. Fast leave-one-out cross-validation of sparse least-squares support vector machines. *Neural Networks*, 17(10):1467–1475, December 2004.

[6] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[7] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence (IJCAI)*, pages 1137–1143, San Mateo, CA, 1995. Morgan Kaufmann.

[8] A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition (in Russian). *Techicheskaya Kibernetica*, 3, 1969.

[9] H. B. Mann and D. R. Whitney. On a test whether one of two random variables is stochastically larger than the other. *Ann. Math. Statist.*, 18:50–60, 1947.

[10] J. A. Nelder and R. Mead. A simplex method for function minimisation. *Computer Journal*, 7:308–313, 1965.

[11] T. Seaks. SYMINV : An algorithm for the inversion of a positive definite matrix by the Cholesky decomposition. *Econometrica*, 40(5):961–962, September 1972.

[12] J. A. K. Suykens, T. Van Gestel, J. De Brabanter, B. De Moor, and J. Vanderwalle. *Least squares support vector machines*. World Scientific, 2002.

[13] A. N. Tikhonov and V. Y. Arsenin. *Solutions of ill-posed problems*. John Wiley, New York, 1977.

[14] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics*, 1:80–83, 1945.