ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 149-154

Fast Exact Leave-One-Out Cross-Validation of Least-Squares Support Vector Machines

Kamel Saadi, Gavin C. Cawley and Nicola L. C. Talbot

School of Information Systems
University of East Anglia
Norwich, U.K. NR4 7TJ
gcc@sys.uea.ac.uk

Abstract.
Model selection methods for kernel machines often seek to minimise an upper bound on the leave-one-out cross-validation error. This paper describes an efficient algorithm for *exact* leave-one-out cross-validation of least-squares support vector machines, in both classification and regression settings. The proposed method exploits the considerable redundancy in the family of systems of linear equations to be solved in explicit computation of the leave-one-out error. The efficiency of the proposed approach is demonstrated using real-world and synthetic benchmark datasets.

1 Introduction

The generalisation properties of kernel models are typically governed by the choice of kernel and a small number of kernel and regularisation parameters. Finding the optimal values for these parameters, i.e. those minimising an appropriate loss function on unseen data, is an activity known as *model selection*. If the available data is sufficiently large, it is common practice to partition the data into three sets; the *training* set is used to estimate the model parameters, performance on the *validation* set provides the criterion for model selection, and the generalisation of the model estimated using the statistically pure *test* set. In situations where little data is available cross-validation may be preferable. First the available data is partitioned into S disjoint sets, a model is then trained using each combination of $S-1$ partitions and its performance estimated over the remaining portion of the data. The cross-validation estimate of the generalisation of the model is the mean of the test-partition performance of the S models. The most extreme form of cross-validation, where S is equal to the number of training patterns, is known as *leave-one-out* cross-validation.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 149-154

Leave-one-out cross-validation provides a useful criterion for model selection as it gives an almost unbiased estimate of the expected error on unseen data [1]. In this paper we demonstrate an efficient algorithm for leave-one-out cross-validation of the least-squares support vector machine for use in model selection and comparison.

2 The Least Squares Support Vector Machine

Ridge regression [2] is a method from classical statistics that implements a regularised form of least-squares regression. Given training data,

$$\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_{i=1}^{\ell}, \quad \boldsymbol{x}_i \in \mathcal{X} \subset \mathbb{R}^d, \quad y_i \in \mathcal{Y} \subset \mathbb{R},$$

ridge regression determines the parameter vector, $\boldsymbol{w} \in \mathbb{R}^d$, of a linear model, $f(\boldsymbol{x}) = \boldsymbol{w} \cdot \boldsymbol{x} + b$, by minimising the objective function

$$W_{\mathrm{RR}}(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{\ell}\sum_{i=1}^{\ell}(y_i - \boldsymbol{w} \cdot \boldsymbol{x}_i - b)^2. \tag{1}$$

The objective function used in ridge regression (1) corresponds to a form of Tikhonov regularisation [3] of a sum-of-squares error metric, where γ is a regularisation parameter controlling the bias-variance trade-off [4]. This is equivalent to penalised maximum likelihood estimation of $\boldsymbol{w}$, assuming the targets have been corrupted by an independent and identically distributed (i.i.d.) sample from a Gaussian noise process, with zero mean and variance σ^2, i.e.

$$y_i = \boldsymbol{w} \cdot \boldsymbol{x}_i + b + \epsilon_i, \quad \epsilon \sim \mathcal{N}(0, \sigma^2).$$

A non-linear form of ridge regression [5–7], the least-squares support vector machine, can be obtained via the so-called "kernel trick", whereby a linear ridge regression model is constructed in a high dimensional feature space, $\mathcal{F}$ ($\boldsymbol{\phi} : \mathcal{X} \to \mathcal{F}$), induced by a non-linear kernel function defining the inner product $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x}) \cdot \boldsymbol{\phi}(\boldsymbol{x}')$. The kernel function, $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ may be any positive definite "Mercer" kernel. The objective function minimised in constructing a least-squares support vector machine is given by

$$W_{\mathrm{LS-SVM}}(\boldsymbol{w}, b) = \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{\gamma}{\ell}\sum_{i=1}^{\ell}(y_i - \boldsymbol{w} \cdot \boldsymbol{\phi}(\boldsymbol{x_i}) + b)^2.$$

The representer theorem [8] indicates that the solution of an optimisation problem of this nature can be written in the form of an expansion involving training patterns, i.e. $\boldsymbol{w} = \sum_{i=1}^{\ell} \alpha_i \boldsymbol{\phi}(\boldsymbol{x}_i)$. The output of the least-squares support vector machine is then given by the kernel expansion

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

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 149-154

It can easily be shown [5, 6] that the optimal coefficients of this expansion are given by the solution of a set of linear equations

$$\left[\begin{array}{cc} \boldsymbol{\Omega} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ 0 \end{array}\right], \tag{2}$$

where $\boldsymbol{\Omega} = \boldsymbol{K} + \ell\gamma^{-1}\boldsymbol{I}$, $\boldsymbol{K} = [k_{ij} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]_{i,j=1}^{\ell}$, $\boldsymbol{y} = (y_1, y_2, \ldots, y_\ell)^T$, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_\ell)^T$ and $\mathbf{1} = (1, 1, \ldots, 1)^T$.

3 Fast Exact Leave-One-Out Cross-Validation

In each iteration of the leave-one-out cross-validation procedure, a least squares support vector machine is trained excluding a single training pattern. Clearly this can be implemented by solving the system of linear equations (2), striking out one row and the corresponding column, for example:

$$\left[\begin{array}{ccccc} \omega_{11} & \omega_{12} & \cdots & \omega_{1l} & 1 \\ \omega_{21} & \omega_{22} & \cdots & \omega_{2l} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \omega_{l1} & \omega_{l2} & \cdots & \omega_{ll} & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{array}\right] \left[\begin{array}{c} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_l \\ b \end{array}\right] = \left[\begin{array}{c} y_1 \\ y_2 \\ \vdots \\ y_l \\ 0 \end{array}\right].$$

To perform leave-one-out cross-validation, we must solve ℓ systems of ℓ linear equations in ℓ variables. The solution of a system of linear equations using Gauss-Jordan elimination requires $\mathcal{O}(\ell^3)$ operations, so it would appear at first glance that the leave-one-out cross-validation of the least-squares support vector machine involves $\mathcal{O}(\ell^4)$ operations. Fortunately there is a considerable degree of redundancy between the systems of linear equations, and so faster solutions are possible. Consider the first three steps of the Gauss-Jordan elimination process for a matrix $\boldsymbol{A}$:

$$\boldsymbol{A}^0 = \left[\begin{array}{ccccc} a_{11}^0 & a_{12}^0 & a_{13}^0 & a_{14}^0 & a_{15}^0 \\ a_{21}^0 & a_{22}^0 & a_{23}^0 & a_{24}^0 & a_{25}^0 \\ a_{31}^0 & a_{32}^0 & a_{33}^0 & a_{34}^0 & a_{35}^0 \\ a_{41}^0 & a_{42}^0 & a_{43}^0 & a_{44}^0 & a_{45}^0 \\ a_{51}^0 & a_{52}^0 & a_{53}^0 & a_{54}^0 & a_{55}^0 \end{array}\right] \quad \boldsymbol{A}^1 = \left[\begin{array}{ccccc} a_{11}^1 & a_{12}^1 & a_{13}^1 & a_{14}^1 & a_{15}^1 \\ 0 & a_{22}^1 & a_{23}^1 & a_{24}^1 & a_{25}^1 \\ 0 & a_{32}^1 & a_{33}^1 & a_{34}^1 & a_{35}^1 \\ 0 & a_{42}^1 & a_{43}^1 & a_{44}^1 & a_{45}^1 \\ 0 & a_{52}^1 & a_{53}^1 & a_{54}^1 & a_{55}^1 \end{array}\right]$$

$$\boldsymbol{A}^2 = \left[\begin{array}{ccccc} a_{11}^2 & a_{12}^2 & a_{13}^2 & a_{14}^2 & a_{15}^2 \\ 0 & a_{22}^2 & a_{23}^2 & a_{24}^2 & a_{25}^2 \\ 0 & 0 & a_{33}^2 & a_{34}^2 & a_{35}^2 \\ 0 & 0 & a_{43}^2 & a_{44}^2 & a_{45}^2 \\ 0 & 0 & a_{53}^2 & a_{54}^2 & a_{55}^2 \end{array}\right] \quad \boldsymbol{A}^3 = \left[\begin{array}{ccccc} a_{11}^3 & a_{12}^3 & a_{13}^3 & a_{14}^3 & a_{15}^3 \\ 0 & a_{22}^3 & a_{23}^3 & a_{24}^3 & a_{25}^3 \\ 0 & 0 & a_{33}^3 & a_{34}^3 & a_{35}^3 \\ 0 & 0 & 0 & a_{44}^3 & a_{45}^3 \\ 0 & 0 & 0 & a_{54}^3 & a_{55}^3 \end{array}\right]$$

Clearly the results obtained from the i^{th} step form a useful starting point for solution of the systems of linear equations striking out any row between rows

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 149-154

$i+1$ and ℓ and the corresponding column. One can take advantage of this redundancy in this family of systems of linear equations by caching of these partial solutions from Gauss-Jordan elimination of the original matrix $\mathbf{\Omega}$ (we assume sufficient memory to store all partial solutions).

4 Simulation Results

For simplicity, the efficiency of the proposed fast exact leave-one-out cross-validation algorithm is first assessed via a series of random datasets containing between 20 and 400 training patterns, using a linear kernel $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{x} \cdot \boldsymbol{x}'$, without regularisation ($\gamma \to \infty$). Figure 1 shows a comparison of the computational expense, as measured by the number of floating point operations performed, of leave-one-out validation schemes based on standard and efficient Gauss-Jordan elimination algorithms. The proposed algorithm is demonstrated to be between 2.28 and 5.83 times faster than the standard approach, with the improvement in speed increasing as the training set becomes larger.

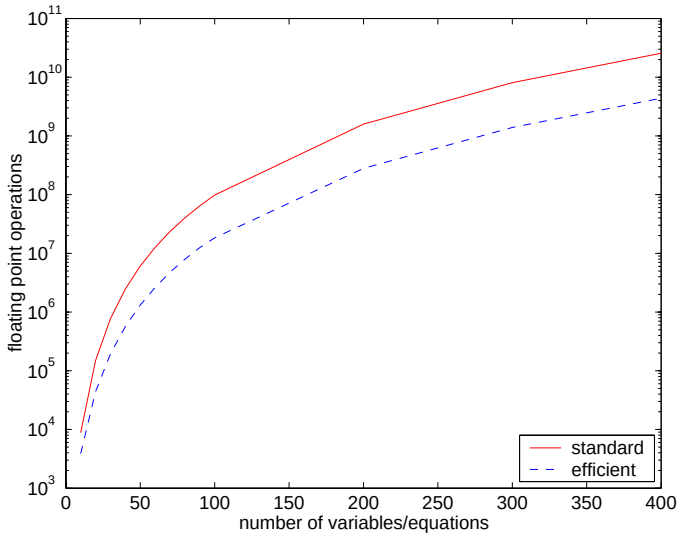

Figure 1: Comparison of the computational expense of leave-one-out validation schemes based on standard and efficient Gauss-Jordan elimination.

The Pima Indian benchmark dataset [9] describes the effect of a variety of physiological attributes on the occurrence of diabetes in a population of women of Pima Indian descent living near Phoenix Arizona. The dataset defines a classification problem, in which individuals are classified as diabetic or non-diabetic given a vector containing 8 attributes thought to be relevant. The data is comprised of a training set of 200 patterns and a test set of 332 patterns. A least-squares support vector classification network was then trained, using an

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 149-154

anisotropic Gaussian radial basis function kernel,

$$\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \exp\left\{\sum_{i=1}^{n} \sigma_i^{-2}(x_i - x_i')^2\right\}.$$

The values of the regularisation, γ, and kernel, $\boldsymbol{\sigma}$, parameters were selected so as to minimise the 10-fold cross-validation error over the training set. The leave-one-out cross-validation error was then found, using both the standard and efficient algorithms, to be 23%. The time taken to compute the leave-one-out error using the standard approach was 63.1 seconds, and using the proposed fast algorithm only 10.16 seconds, a factor of 5.86 times faster.

5 Discussion

In this paper we have introduced a fast exact algorithm for leave-one-out cross-validation of least-squares support vector machines in a non-linear regression setting. A similar approach is equally valid for least-squares support vector machines in a classification setting, where the optimal values of the model parameters are again found by solution of a system of linear equations. This approach could also be adapted to provide an upper bound[1] on the leave-one-out error of the 2-norm formulation of the support vector machine [10], where the optimal model parameters are given by the solution of the Karush-Kuhn-Tucker (KKT) system for the set of support vectors [11]. We are also investigating the redundancy involved in solution of families of positive-definite systems of linear equations via the Cholesky method, for leave-one-out cross-validation of kernel ridge regression models (essentially a least-squares support vector machine without a bias term [5]).

6 Summary

An efficient algorithm for exact leave-one-out cross-validation of least-squares support vector machine in a regression setting is introduced. The efficiency of the proposed method is then demonstrated using real world and synthetic benchmark dataset. The most important feature of the proposed algorithm is that it is numerically equivalent to explicit computation of the leave-one-out error, but is significantly faster.

7 Acknowledgements

The authors would like to thank Rob Foxall and the anonymous reviewers for their helpful comments on previous drafts of this manuscript. This work was supported by Royal Society research grant RSRG-22270.

[1] It is assumed that the set of support vectors is unchanged during the leave-one-out cross-validation procedure.

ESANN'2002 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 24-26 April 2002, d-side publi., ISBN 2-930307-02-1, pp. 149-154

References

[1] A. Luntz and V. Brailovsky. On estimation of characters obtained in statistical procedure of recognition (in russian). *Technicheskaya Kibernetica*, 1969.

[2] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

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

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

[5] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *Proceedings, 15th International Conference on Machine Learning*, pages 515–521, Madison, WI, July 24–27 1998.

[6] J. Suykens, L. Lukas, and J. Vandewalle. Sparse approximation using least-squares support vector machines. In *Proceedings, IEEE International Symposium on Circuits and Systems*, pages 11757–11760, Geneva, Switzerland, May 2000.

[7] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle. Weighted least squares support vector machines : robustness and sparse approximation. *Neurocomputing*, 2001.

[8] G. S. Kimeldorf and G. Wahba. Some results on Tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[9] S. D. Bay. The UCI KDD archive [`http://kdd.ics.uci.edu/`]. University of California, Department of Information and Computer Science, Irvine, CA, 1999.

[10] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[11] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines (and other kernel-based learning methods).* Cambridge University Press, Cambridge, U.K., 2000.