A Greedy Training Algorithm for Sparse Least-Squares Support Vector Machines

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. Suykens *et al.* [1] describes a form of kernel ridge regression known as the least-squares support vector machine (LS-SVM). In this paper, we present a simple, but efficient, greedy algorithm for constructing near optimal sparse approximations of least-squares support vector machines, in which at each iteration the training pattern minimising the regularised empirical risk is introduced into the kernel expansion. The proposed method demonstrates superior performance when compared with the pruning technique described by Suykens *et al.* [1], over the motorcycle and Boston housing datasets.

1 Introduction

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

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

The objective function used in ridge regression (1) implements 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 corresponds 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 [1, 5, 6], 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,

$\mathcal{F}$ ($\boldsymbol{\phi} : \mathcal{X} \rightarrow \mathcal{F}$), 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, $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ may be any positive definite "Mercer" kernel. The objective function minimised in constructing a least-squares support vector machine is given by

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

The representer theorem [7] 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

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

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

$$\begin{bmatrix} \boldsymbol{\Omega} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{y} \\ 0 \end{bmatrix},$$

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 $\mathbf{1} = (1, 1, \ldots, 1)^T$.

1.1 Imposing Sparsity

Unfortunately, unlike the support vector machine, the kernel expansion implementing a least-squares support vector machine is in general fully dense, i.e. $\alpha_i \neq 0$, $\forall\ i \in \{1, 2, \ldots, \ell\}$; this, along with the $\mathcal{O}(\ell^2)$ space and $\mathcal{O}(\ell^3)$ time complexities of the training algorithm make this approach impractical for very large-scale applications. Suykens *et al* [6] propose an iterative pruning procedure to obtain a sparse approximation of the full kernel expansion: A LS-SVM is trained on the entire dataset, yielding a vector of coefficients, $\boldsymbol{\alpha}$. A small fraction of the data (say 5%), associated with coefficients having the smallest magnitudes, is discarded and the LS-SVM retrained on the remaining data. This process is repeated until a sufficiently small kernel expansion is obtained. Model selection is performed at each iteration to refine values for the regularisation parameter, γ and any kernel parameters, in order to obtain adequate generalisation. In this paper we propose a constructive training algorithm for sparse approximation of least-squares support vector machines, adding terms to the kernel expansion in a greedy manner. The proposed algorithm also takes into account the residuals of all training patterns, rather than just those included in the kernel expansion, eliminating the need for further model selection.

2 Method

We begin by introducing an improved formulation of the objective function that includes the residuals for all training patterns, rather than just those patterns currently included in the kernel expansion. If the weight vector, $\boldsymbol{w}$, can be closely approximated by a weighted sum of a limited subset of the training vectors, i.e., $\boldsymbol{w} \approx \sum_{i \in \mathcal{S}} \beta_i \boldsymbol{\phi}(\boldsymbol{x}_i)$, $\mathcal{S} \subset \{1, 2, \ldots \ell\}$, then we obtain the objective function minimised by the greedy sparse least-squares support vector machine

$$\mathcal{L}(\boldsymbol{\beta}, b) = \frac{1}{2} \sum_{i,j \in \mathcal{S}} \beta_i \beta_j k_{ij} + \frac{\gamma}{\ell} \sum_{i=1}^{\ell} (y_i - \sum_{j \in \mathcal{S}} \beta_j k_{ij} - b)^2. \tag{2}$$

Setting the partial derivatives of $\mathcal{L}$ with respect to $\boldsymbol{\beta}$ and b to zero, and dividing through by $2\gamma/\ell$, yields:

$$\sum_{i \in \mathcal{S}} \beta_i \sum_{j=1}^{\ell} k_{ij} + \ell b = \sum_{j=1}^{\ell} y_j$$

and

$$\sum_{i \in \mathcal{S}} \beta_i \left\{ \frac{\ell}{2\gamma} k_{ir} + \sum_{j=1}^{\ell} k_{jr} k_{ji} \right\} + b \sum_{i=1}^{\ell} k_{ir} = \sum_{i=1}^{\ell} y_i k_{ir}, \qquad \forall\ r \in \mathcal{S}$$

These equations can be expressed as a system of $|\mathcal{S}|+1$ linear equations in $|\mathcal{S}|+1$ unknowns,

$$\boldsymbol{H} \begin{bmatrix} \boldsymbol{\beta} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega} & \boldsymbol{\Phi} \\ \boldsymbol{\Phi}^T & \ell \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{c} \\ \sum_{k=1}^{\ell} y_k \end{bmatrix},$$

where $\boldsymbol{\Omega} = [\frac{\ell}{2\gamma} k_{ij} + \sum_{r=1}^{\ell} k_{rj} k_{ri}]_{i,j \in \mathcal{S}}$, $\boldsymbol{\Phi} = (\sum_{j=1}^{\ell} k_{ij})_{i \in \mathcal{S}}$, $\boldsymbol{c} = (\sum_{j=1}^{\ell} y_j k_{ij})_{i \in \mathcal{S}}$. Starting with only a bias term, b, a sparse kernel machine is iteratively constructed in a greedy manner. During each iteration, the training pattern minimising the objective function (2) is incorporated into the kernel expansion. Training can be terminated once the kernel expansion has reached a pre-determined size, or if the reduction in the objective function falls below some threshold value. Note further model selection is not generally necessary as the second summation of (2) is over all training patterns.

2.1 Efficient Implementation

At each iteration, $\boldsymbol{H}$ is extended by additional row and column. The inversion of $\boldsymbol{H}_i$ at the i^{th} iteration can be performed efficiently given $\boldsymbol{H}_{i-1}^{-1}$ computed during the previous iteration, via the block matrix inversion identity

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{S}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & -\boldsymbol{A}^{-1}\boldsymbol{B}\boldsymbol{S}^{-1}1 \\ -\boldsymbol{S}^{-1}\boldsymbol{C}\boldsymbol{A}^{-1} & \boldsymbol{S}^{-1} \end{bmatrix}, \tag{3}$$

where $\boldsymbol{S} = (\boldsymbol{D} - \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{B})$. In this case, $\boldsymbol{C}$ and $\boldsymbol{B}$ are row and column vectors respectively and $\boldsymbol{D}$ is a scalar, and so $\boldsymbol{S}$ is also a scalar, giving

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{b} \\ \boldsymbol{b}^T & c \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{A}^{-1} + \frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b}\boldsymbol{b}^T\boldsymbol{A}^{-1} & -\frac{1}{k}\boldsymbol{A}^{-1}\boldsymbol{b} \\ -\frac{1}{k}\boldsymbol{b}^T\boldsymbol{A}^{-1} & \frac{1}{k} \end{bmatrix}, \tag{4}$$

where $k = c - \boldsymbol{b}^T\boldsymbol{A}^{-1}\boldsymbol{b}$. This allows the inversion of $\boldsymbol{H}_i$ with a complexity of only $\mathcal{O}(n^2)$ operations.

3 Results

The Motorcycle benchmark consists of a sequence of accelerometer readings through time following a simulated motor-cycle crash during an experiment to determine the efficacy of crash-helmets (Silverman [8]). Figure 1 shows conventional and greedy sparse support vector machine models of the motorcycle dataset, using a Gaussian radial basis function kernel,

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

The greedy sparse model is functionally identical to the full least-squares support vector machine model with only 15 basis vectors comprising the sparse kernel expansion. Figure 2 compares the 10-fold root-mean-square (RMS) cross-validation error of greedy sparse and pruned least-squares support vector machines as a function of the number of training patterns included in the resulting kernel expansions. The regularisation and kernel parameters for the pruned model were determined in each trial via minimisation of the 10-fold cross-validation error. The cross-validation error is consistently lower for the greedy sparse model regardless of the number of patterns forming the kernel expansion, without the need for further model selection.

The Boston housing dataset describes the relationship between the median value of owner occupied homes in the suburbs of Boston and thirteen attributes representing environmental and social factors believed to be relevant [9]. Figure 3 compares the 10-fold root-mean-square (RMS) cross-validation error of greedy sparse and pruned least-squares support vector machines. Again the error for the greedy sparse method is consistently lower.

4 Summary

This paper presents a simple but efficient greedy training algorithm for constructing sparse approximations of least-squares support vector machines. The proposed algorithm demonstrates performance superior to that of the pruning algorithm of Suykens *et al.* [1] on two real-world benchmark tasks. The new algorithm is also considerably faster as the need for model selection in each iteration is eliminated. The method also provides a plausible approach for large-scale regression problems as it is no longer necessary to store the entire kernel matrix at any stage during training.

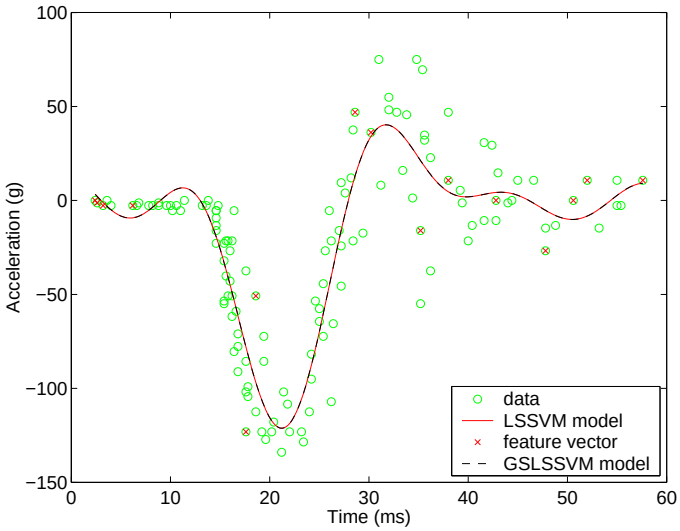

Fig. 1. Least-squares support vector machine (LS-SVM) and greedy sparse least-squares support vector machine (GSLSSVM) models of the motorcycle data set; note the standard and sparse models are essentially identical.

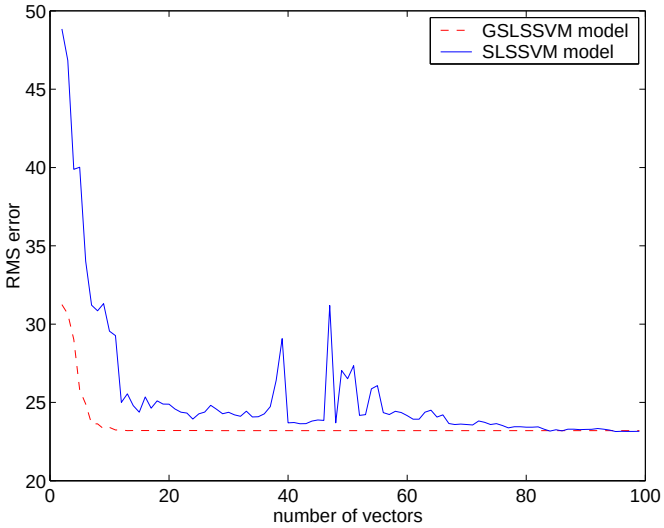

Fig. 2. Cross-validation error of greedy sparse (GSLSSVM) and sparse (SLSSVM) least-squares support vector machine models, over the motorcycle dataset, as a function of the number of training patterns included in the resulting kernel expansions.

5 Acknowledgements

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

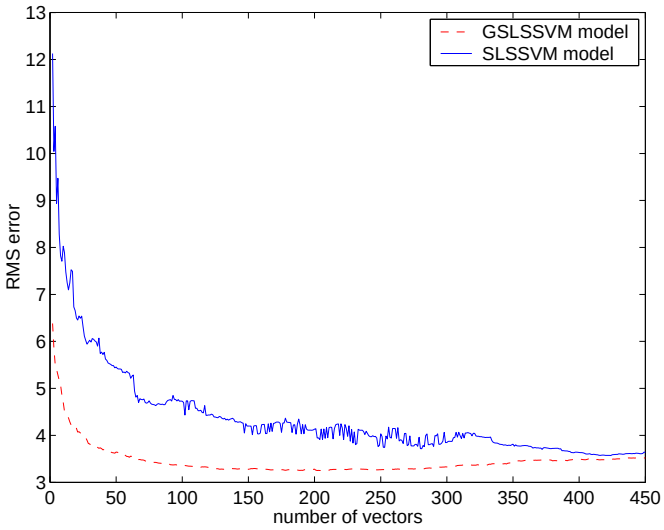

Fig. 3. Cross-validation error of greedy sparse (GSLSSVM) and sparse (SLSSVM) least-squares support vector machine models, over the Boston housing dataset, as a function of the number of training patterns included in the resulting kernel expansions.

References

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

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

[8] B. W. Silverman. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society, B*, 47(1):1–52, 1985.

[9] D. Harrison and D. L. Rubinfeld. Hedonic prices and the demand for clean air. *Journal Environmental Economics and Management*, 5:81–102, 1978.