Parsimonious least squares support vector regression using orthogonal forward selection with the generalised kernel model

Xunxian Wang and David Lowe

Neural Computing Research Group,
Aston University, Birmingham B4 7ET, UK
E-mail: x.wang@aston.ac.uk
E-mail: d.lowe@aston.ac.uk

Sheng Chen* and Chris J. Harris

School of Electronics and Computer Science,
University of Southampton,
Southampton SO17 1BJ, UK
E-mail: sqc@ecs.soton.ac.uk
E-mail: cjh@ecs.soton.ac.uk
*Corresponding author

Abstract: A sparse regression modelling technique is developed using a generalised kernel model in which each kernel regressor has its individually tuned position (centre) vector and diagonal covariance matrix. An orthogonal least squares forward selection procedure is employed to append the regressors one by one. After the determination of the model structure, namely the selection of an appropriate number of regressors, the model weight parameters are calculated from the Lagrange dual problem of the original least squares problem. Different from the least squares support vector regression, this regression modelling procedure involves neither reproducing kernel Hilbert space nor Mercer decomposition concepts. As the regressors used are not restricted to be positioned at training input points and each regressor has its own diagonal covariance matrix, a very sparse representation can be obtained with excellent generalisation capability. Experimental results involving two real data sets demonstrate the effectiveness of the proposed regression modelling approach.

Keywords: generalised kernel model; least squares support vector machine; orthogonal least squares forward selection; regression; sparse modelling.

Reference to this paper should be made as follows: Wang, X., Chen, S., Lowe, D. and Harris, C.J. (2006) ‘Parsimonious least squares support vector regression using orthogonal forward selection with the generalised kernel model’, *Int. J. Modelling, Identification and Control*, Vol. 1, No. 4, pp.245–256.

Biographical notes: Xunxian Wang received his PhD in the control theory and application from Tsinghua University, Beijing, China, in July 1999. From January 2005, he has been a Research Fellow at Neural Computing Research Group, Aston University, UK. His main interests are in machine learning and neural networks, control theory and systems as well as robotics.

Sheng Chen received his PhD in control engineering from the City University, London, UK, in 1986. He joined the School of Electronics and Computer Science, University of Southampton, Southampton, UK, in September 1999. Professor Chen’s research interests include wireless communications, machine learning, finite-precision digital controller design and evolutionary computation.

David Lowe has held the Chair of Neural Computing at Aston University, UK, since 1994. He is a Coinventor of the Radial Basis Function neural network architecture. His current research activities relate to stochastic generative control, biomedical applications of statistical pattern processing focussing on DNA microarrays and EEG/MEG brain signal analysis and non-linear methods for digital steganography.

Chris J. Harris is a Professor of Computational Intelligence at the University of Southampton. His research interests include intelligent autonomous systems, intelligent control, estimation of dynamic processes, and multi-sensor data fusion. He is a follow of the Royal Academy of Engineering. He was awarded the IEE Faraday medal in 2001 for his work in intelligent control and neurofuzzy systems.

1 Introduction

Having a good generalisation capability and a sparse representation are the two key requirements in establishing a learning machine. Forward selection using the Orthogonal Least Squares (OLS) algorithm (Chen et al., 1989, 1991, 1999, 2003 and 2004) is a simple and efficient construction method that is capable of producing parsimonious linear-in-the-weights non-linear models with excellent generalisation performance. Alternatively, the state-of-the-art sparse kernel modelling techniques, such as the Support Vector Machine (SVM) (Chapelle et al., 2002; Cristianini and Shawe-Taylor, 2000; Duan et al., 2003; Ong et al., 2005; Scholkopf et al., 1997; Scholkopf et al.,2000; Scholkopf and Smola, 2000; Vapnik, 1995; Vapnik et al., 1997), have become popular in data modelling applications. Originated from the maximum margin linear classification problem, one of the main features of the SVM approach is to use a hyperplane in a high dimensional space to perform both the classification and regression. In classification, the hyperplane is adjusted to obtain the maximum classification margin. In regression, the gradient of the hyperplane is kept to be as small as possible. The Least Squares Support Vector Machine (LS-SVM) regression (de Kruif and de Vries, 2003; Suykens and Van-dewalle, 1999; Suykens et al., 2002; Van Gestel et al., 2001) is an algorithm for solving the Least Squares (LS) problem in its Lagrange dual space, just as the SVM. In an LS-SVM-type method, the training data are mapped to a high dimensional space where they can be approximated by a hyperplane. The parameter of the hyperplane is obtained by minimising the combined cost function consisting of the least squares error and the squared gradient of the hyperplane.

With the aid of the reproducing kernel Hilbert space through Mercer theorem (Aronszajn, 1950), a Mercer kernel can be used and the required mapping from the input space to the high dimensional space is defined implicitly by this kernel function. How to select an appropriate kernel which realises exactly the required mapping is a key problem and some techniques, such as the hyperkernel method, have been used to determine the kernel type as well as the kernel parameters (Duan et al., 2003; Ong et al., 2005). Lanckriet et al. (2004) describe a method for combining multiple kernel representations in an optimal fashion, by formulating a convex optimisation problem which is solvable by semidefinite programming. By combining several kernels together, the produced system model will have an improved performance. A limitation of the SVM-based regression modelling techniques is the fact that the kernel centres or mean vectors are typically placed at the training input data and a fixed common kernel variance is used for all the regressor kernels. The value of this common kernel variance obviously has a critical influence on the sparsity and generalisation capability of the resulting model and it has to be determined via some sort of cross validation. If the positions of kernel regressors are more flexible and different kernel regressors can have their own diagonal covariance matrices, a better system model can be established. However, putting a kernel function at a position not occupied by a training data point or giving different kernel regressors at different positions different covariance matrices are not allowed for the SVM-based methods due to the use of Mercer theorem. It should also be pointed out that the model representation obtained by the LS-SVM is not sparse. To obtain a sparse model, pruning technique can be applied (de Kruif and de Vries, 2003). By contrast, the SVM approach is capable of producing sparse models. However, we found that the models produced by the SVM method are typically not as sparse as those obtained by the OLS algorithm (Chen et al., 2003, 2004).

In the method proposed in this paper, we determine the kernel parameters based on the given training data before doing a classical LS-SVM procedure. Thus, the proposed method can use non-Mercer kernels. Specifically, the generalised kernel function is used in which each kernel regressor has its tunable centre vector and diagonal covariance matrix. Unlike the LS-SVM formulation, we consider an alternative Lagrange dual problem of the general regularised LS problem, which does not restrict to the use of Mercer kernels. To arrive at a sparse representation, an OLS forward selection procedure is adopted to append regressors one by one by incrementally minimising the regularised training Mean Square Eerror (MSE). Unlike the standard OLS algorithm (Chen et al., 1989), however, at each stage of selection, the optimisation with respect to the kernel centre vector and diagonal covariance matrix is performed using a guided random search algorithm called the Repeated Weighted Boosting Search (RWBS) (Chen et al., 2005). The RWBS algorithm is a global optimisation search method that adopts some ideas from boosting (Breiman, 1999; Freund and Schapire, 1997; Meir and Ratsch, 2003; Schapire, 1990). This optimisation algorithm is simple, robust, easy to implement and can be used in the situation where the cost function is multimodal and/or non-smooth. After the selection of a parsimonious model representation, the associate kernel weights are then calculated from the Lagrange dual problem of the original regularised LS problem. This proposed generalised kernel regression modelling approach has the potential of improving modelling capability and producing sparser final models, compared with the standard approach of restricting the kernel centres to the training inputs and using a single fixed common variance. The advantages of the proposed method are illustrated using two real-life data modelling examples.

The outline of the paper is as follows. Section 2 reviews the standard kernel regression modelling, which positions the kernel regressors at the training input data points and adopts a common variance for every kernel regressor. The classical LS-SVM formulation is first summarised. An alternative Lagrange dual problem of the general regularised LS problem is then considered, which does not restrict to the use of Mercer kernels. This method will be referred to as the Extended LS-SVM (LS-ESVM). Thirdly, to derive a sparse representation, the standard OLS algorithm (Chen et al., 1989) is used to select a parsimonious model and this is followed by solving the corresponding Sparse Extended LS-SVM problem to yield the model weight parameters. This method will be called the Sparse Extended LS-SVM (LS-SESVM). Here it is worth pointing out that many existing kernel selection algorithms, such as the kernel matching pursuit (Vincent and Bengio, 2002), are in fact identical or equivalent to the OLS algorithm (Chen et al., 1989) and therefore, the similarity between the LS-SESVM algorithm and these kernel selection methods should be apparent. The main contribution of this paper is presented in Section 3, where the generalised kernel regression modelling is considered. A new OLS forward selection procedure is proposed which uses the RWBS algorithm (Chen et al., 2005) to determine the kernel centres and diagonal covariance matrices. This guarantees a sparse representation. Again, the associated kernel weights are solved from a similar LS-ESVM problem after obtaining a sparse representation. We will refer to this proposed new method as the Generalised Sparse Extended LS-SVM (LS-GSESVM) for the purpose of comparison with the methods of Section 2. Section 4 describes our modelling experiments, while Section 5 offers our conclusions.

2 Standard kernel regression modelling

The task of kernel regression modelling is to construct a kernel model from the given training data set $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i$ is the ith training input vector of dimension m, y_i is the desired output for the input $\mathbf{x}_i$ and N the number of training data. The LS-SVM method is a standard approach to solve this problem.

2.1 *The least squares support vector machine problem*

The minimisation problem of the LS-SVM can be stated as below:

$$\min J(\mathbf{w}, \mathbf{e}) = \min \left\{ \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \right\} \tag{1}$$

$$\text{s.t. } y_i = \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}_i) + b + e_i,\ i = 1, \ldots, N \tag{2}$$

where $\boldsymbol{\varphi}(\mathbf{x})$ is the selected mapping from the input space to the high-dimensional space, $y = \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b$ is the regression linear function (hyperplane) in the high dimensional space, $\mathbf{w}$ is the gradient of the hyperplane, $\mathbf{e} = [e_1\ e_2 \cdots e_N]^T$ denotes the regression error vector, and C is a constant that determines the trade off between regularisation and training error. Let us define the Mercer kernel matrix

$$\mathbf{K} = \begin{bmatrix} k_{1,1} & k_{1,2} & \cdots & k_{1,N} \\ k_{2,1} & k_{2,2} & \cdots & k_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ k_{N,1} & k_{N,2} & \cdots & k_{N,N} \end{bmatrix} \tag{3}$$

where the Mercer kernel

$$k_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j) \rangle \tag{4}$$

with $\langle \bullet, \bullet \rangle$ denoting the inner product in the high-dimensional space. It is well known that the dual problem of Equations (1) and (2) is:

$$\max \bar{L}(\boldsymbol{\alpha}) = \max \left\{ -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{K}\boldsymbol{\alpha} - \frac{1}{2C}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{y} \right\} \tag{5}$$

$$\text{s.t. } \sum_{i=1}^{N} \alpha_i = 0 \tag{6}$$

where $\boldsymbol{\alpha}^T = [\alpha_1\ \alpha_2 \cdots \alpha_N]$ and $\mathbf{y} = [y_1\ y_2 \cdots y_N]^T$. The regression model is given by

$$\hat{y}(\mathbf{x}) = \mathbf{w}^T\boldsymbol{\varphi}(\mathbf{x}) + b = \sum_{i=1}^{N} \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b \tag{7}$$

The most common choice of kernel function is the Gaussian function of the form:

$$k(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2} \right) \tag{8}$$

The common kernel variance σ^2 is not provided by the algorithm and has to be determined by other means, such as via cross validation. The model obtained by the LS-SVM algorithm is not sparse. In de Kruif and de Vries (2003), a pruning method is used to obtain a sparse representation.

2.2 *The dual of the regularised least squares problem*

Consider the regularised LS regression problem stated as below

$$\min J(\boldsymbol{\alpha}, \mathbf{e}) = \min \left\{ \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \right\} \tag{9}$$

$$\text{s.t. } y_i = \sum_{j=1}^{M} \alpha_j g_j(\mathbf{x}_i) + b + e_i,\ i = 1, \ldots, N \tag{10}$$

where the regression model is defined by

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{M} \alpha_j g_j(\mathbf{x}) + b \tag{11}$$

M is the number of kernel regressors and $g_j(\mathbf{x})$, $1 \le j \le M$, are the regressors, which may take (but not restrict to) the form of $g_j(\mathbf{x}) = k(\mathbf{x}_j, \mathbf{x})$ with $M = N$. Let us introduce the following definitions $\mathbf{b} = [b\ b \cdots b]^T$, $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2 \cdots \alpha_M]^T$ and

$$\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2 \cdots \mathbf{g}_M] \tag{12}$$

with

$$\mathbf{g}_j = [g_j(\mathbf{x}_1)\ g_j(\mathbf{x}_2) \cdots g_j(\mathbf{x}_N)]^T,\ 1 \le j \le M \tag{13}$$

Then the optimisation Problem (9) and (10) can be written in the matrix form as

$$\min J(\boldsymbol{\alpha}, \mathbf{e}) = \min \left\{ \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{C}{2}\mathbf{e}^T\mathbf{e} \right\} \tag{14}$$

$$\text{s.t. } \mathbf{y} = \mathbf{G}\boldsymbol{\alpha} + \mathbf{b} + \mathbf{e} \tag{15}$$

The Lagrangian of this optimisation problem is given by

$$L(\boldsymbol{\gamma}, \boldsymbol{\alpha}, b, \mathbf{e}) = \frac{1}{2}\boldsymbol{\alpha}^T\boldsymbol{\alpha} + \frac{C}{2}\mathbf{e}^T\mathbf{e} - \boldsymbol{\gamma}^T(\mathbf{G}\boldsymbol{\alpha} + \mathbf{b} + \mathbf{e} - \mathbf{y}) \tag{16}$$

where $\boldsymbol{\gamma}^T = [\gamma_1\ \gamma_2 \ldots \gamma_N]$ are the Lagrange multipliers.

From the Kuhn-Tucker conditions, we have

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \boldsymbol{\alpha} - \mathbf{G}^T\boldsymbol{\gamma} = 0 \tag{17}$$

$$\frac{\partial L}{\partial b} = \sum_{j=1}^{N} \gamma_j = 0 \tag{18}$$

$$\frac{\partial L}{\partial \mathbf{e}} = C\mathbf{e} - \boldsymbol{\gamma} = 0 \tag{19}$$

The dual problem of the primal Problem (14) and (15) can be obtained as

$$\max \bar{L}(\boldsymbol{\gamma}) = \max\left\{-\frac{1}{2}\boldsymbol{\gamma}^T\mathbf{G}\mathbf{G}^T\boldsymbol{\gamma} - \frac{1}{2C}\boldsymbol{\gamma}^T\boldsymbol{\gamma} + \boldsymbol{\gamma}^T\mathbf{y}\right\} \tag{20}$$

$$\text{s.t.} \sum_{j=1}^{N} \gamma_j = 0 \tag{21}$$

After $\boldsymbol{\gamma}$ is known, the kernel weight vector $\boldsymbol{\alpha}$ in (11) can be obtained from (17) as follows

$$\boldsymbol{\alpha} = \mathbf{G}^T\boldsymbol{\gamma} \tag{22}$$

The difference between the regularised LS problem of equations (9) and (10) and the one given in Equations (1) and (2) is that in the former problem the regularisation item controls the weight of the kernel function directly while in the latter it controls the gradient of the unseen hyperplane. Note that the kernel function used in this LS-ESVM approach does not restrict to a Mercer kernel.

2.3 *Construction of sparse kernel models*

The LS-ESVM algorithm of Section 2.2 cannot give a sparse system model. To obtain a sparse model, we propose first to use the OLS algorithm (Chen et al., 1989) to select a parsimonious subset model from the full regression model (12). Without the loss of generality, we will assume the bias term $b = 0$ in the model (11). Let an orthogonal decomposition of the regression matrix $\mathbf{G}$ be

$$\mathbf{G} = \mathbf{P}\,\mathbf{D} \tag{23}$$

where $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2 \ldots \mathbf{p}_M]$ with orthogonal columns satisfying $\mathbf{p}_i^T\mathbf{p}_j = 0$ if $i \neq j$ and

$$\mathbf{D} = \begin{bmatrix} 1 & d_{1,2} & \cdots & d_{1,M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & d_{M-1,M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{24}$$

The regression model (15) can alternatively be expressed as

$$\mathbf{y} = \mathbf{P}\,\mathbf{D}\boldsymbol{\alpha} + \mathbf{e} = \mathbf{P}\boldsymbol{\theta} + \mathbf{e} \tag{25}$$

where the weight vector $\boldsymbol{\theta}$ in the orthogonal model space satisfies the triangular system $\boldsymbol{\theta} = \mathbf{D}\boldsymbol{\alpha}$.

Consider the regularised LS cost for this M-term regression model stated as below

$$J_M = \frac{1}{2}\boldsymbol{\theta}^T\boldsymbol{\theta} + \frac{C}{2}\mathbf{e}^T\mathbf{e} \tag{26}$$

By substituting the regularised LS solution for $\boldsymbol{\theta}$ in (26) and using the orthogonal property of $\mathbf{P}$, it can be shown that

$$J_M = C\mathbf{y}^T\mathbf{y} - \sum_{j=1}^{M} \frac{\left(\mathbf{y}^T\mathbf{p}_j\right)^2}{C\mathbf{p}_j^T\mathbf{p}_j + 1} \tag{27}$$

Define the error reduction due to the jth term $\mathbf{p}_j$ as

$$\mathrm{ER}_j = \frac{\left(\mathbf{y}^T\mathbf{p}_j\right)^2}{C\mathbf{p}_j^T\mathbf{p}_j + 1} \tag{28}$$

Based on this error reduction criterion, a subset model can be obtained in a forward selection procedure (Chen et al., 1989). At the lth selection stage, a model term is selected from the remaining candidates $\mathbf{p}_j$, $l \leq j \leq M$, as the lth model term in the subset model, if it maximises the error reduction criterion ER_j. The details of the selection algorithm are readily available in (Chen et al., 1989, 1991, 1999, 2003, 2004) and therefore, will not be repeated here. The selection is terminated at the M_s stage if

$$J_{M_s} \leq \xi \tag{29}$$

where the small positive tolerance value ξ controls the sparsity level of the selected subset model. This produces a parsimonious model containing M_s terms. Appropriate value for ξ is problem dependent and may be learnt via cross validation. Alternatively, the Akaike information criterion (Akaike, 1974; Leonataritis and Billings, 1987) can be adopted to terminate the subset model selection procedure. Moreover, the optimal experimental design criteria can be combined with the regularised LS criterion (26) to automatically terminate the selection with an appropriate M_s-term subset model without the need for the user to specify a tolerance value ξ, see (Chen et al., 2003; Hong and Harris, 2002; Hong et al., 2003).

In the standard kernel regression modelling, each kernel regressor is positioned at a training input data point and a common kernel variance σ^2 is used for every regressor. Using the OLS forward selection procedure described above, we first obtain a sparse representation containing M_s kernel regressors. The corresponding kernel weights are then calculated using the LS-ESVM method of Section 2.2. We will referred to this approach of constructing sparse kernel models as the sparse extended LS-SVM (LS-SESVM) method.

3 Generalised kernel regression modelling

In Section 2.2, the deduction of the dual problem does not assume the concept of reproducing kernel Hilbert space and Mercer theorem. Therefore, we are not restricted to Mercer kernels. For example, we will allow a kernel function to take position other than the training input data points and to have an individually tunable diagonal covariance matrix.

This leads to the generalised kernel regression modelling. Specifically, we consider the regressors which take the form of the generalised Gaussian kernel:

$$g_j(\mathbf{x}) = g(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) = \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)^T \boldsymbol{\Sigma}_j^{-1}\left(\mathbf{x} - \boldsymbol{\mu}_j\right)\right) \quad (30)$$

for $1 \leq j \leq M$, where $\boldsymbol{\mu}_j$ is the mean vector of the jth kernel and $\boldsymbol{\Sigma}_j = \mathrm{diag}\{\sigma_{j,1}^2, \sigma_{j,2}^2, \cdots \sigma_{j,m}^2\}$ its diagonal covariance matrix.

3.1 *Construction of sparse generalised kernel models*

In this section, we develop an incremental construction procedure for obtaining sparse generalised kernel models. We will adopt an orthogonal forward selection to append the kernels one by one. At the lth stage of model construction, the lth kernel regressor is determined by maximising the following error reduction criterion:

$$\mathrm{ER}_l(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) = \frac{\left(\mathbf{y}^T \mathbf{p}_l\right)^2}{C\mathbf{p}_l^T \mathbf{p}_l + 1} \quad (31)$$

where $\mathbf{p}_l$ is obtained by an orthogonal transformation of the lth model column $\mathbf{g}_l$ via

$$\mathbf{p}_l = \mathbf{g}_l - \sum_{j=1}^{l-1} d_{j,l}\mathbf{p}_j \quad (32)$$

and $\mathbf{p}_j$, $1 \leq j \leq l-1$, are the orthogonalised model columns already selected. All the discussions in Section 2.3 regarding the termination of selection apply here. For example, the model appending process can be terminated when

$$J_{M_s} = C\mathbf{y}^T\mathbf{y} - \sum_{l=1}^{M_s} \mathrm{ER}_l(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \leq \xi \quad (33)$$

yielding an M_s-term generalised kernel model. The corresponding kernel weights can readily be calculated using the LS-ESVM method of Section 2.2. For a comparison purpose, we will called this construction approach the generalised sparse extended LS-SVM (LS-GSESVM) method.

3.2 *Determination of the generalised kernel parameters*

It can be seen that at each increment regression stage, the basic task is to minimise the cost function

$$f(\mathbf{u}) = \frac{1}{\mathrm{ER}_l(\mathbf{u})}, \quad \mathbf{u} \in \mathcal{U} \quad (34)$$

where $\mathbf{u}$ contains the regressor mean vector $\boldsymbol{\mu}_l$ and diagonal covariance matrix $\boldsymbol{\Sigma}_l$. This optimisation task may be carried out with a gradient based optimisation method. A gradient method however depends on the initial condition and may be trapped at the local minima. Alternatively, the global optimisation methods, such as the Genetic Algorithm (GA) (Golodberg, 1989; Man et al., 1998) and Adaptive Simulated Annealing (ASA) (Chen and Luk, 1999; Ingber, 1993), can be used. We propose to perform this optimisation task using a guided random search algorithm called the RWBS (Chen et al., 2005). The RWBS algorithm is a simple yet efficient global search algorithm. In several global optimisation applications investigated in (Chen et al., 2005), the RWBS algorithm was shown to achieve a similar convergence speed as the GA and ASA. The RWBS algorithm has additional advantages of requiring minimum programming effort and having very few algorithmic parameters that require to tune.

3.3 *Repeated Weighted Boosting Search*

The procedure of using the RWBS algorithm to determine the generalised kernel parameters at the lth incremental modelling stage can now be summarised as follows. Let P_S be the population size, N_G the number of generations in the repeated search and ξ_B the accuracy for terminating the weighted boosting search.

Outer loop: generations For $n = 1 : N_G$

Generation initialisation: Initialise the population by setting $\mathbf{u}_1^{(n)} = \mathbf{u}_{\mathrm{best}}^{(n-1)}$ and randomly generating rest of the population members $\mathbf{u}_i^{(n)}$, $2 \leq i \leq P_S$, where $\mathbf{u}_{\mathrm{best}}^{(n-1)}$ denotes the solution found in the previous generation. If $n = 1$, $\mathbf{u}_1^{(n)}$ is also randomly chosen.

Weighted boosting search initialisation: Assign the initial distribution weightings $\delta_i(0) = \frac{1}{P_S}$, $1 \leq i \leq P_S$, for the population. Then

1 For $1 \leq i \leq P_S$, generate $\mathbf{g}_l^{(i)}$ from $\mathbf{u}_i^{(n)}$, the candidates for the lth regressor and orthogonalise them:

$$d_{j,l}^{(i)} = \frac{\mathbf{p}_j^T \mathbf{g}_l^{(i)}}{\mathbf{p}_j^T \mathbf{p}_j}, \quad 1 \leq j < l \quad (35)$$

$$\mathbf{p}_l^{(i)} = \mathbf{g}_l^{(i)} - \sum_{j=1}^{l-1} d_{j,l}^{(i)} \mathbf{p}_j \quad (36)$$

2 For $1 \leq i \leq P_S$, calculate the cost function value of each $\mathbf{u}_i^{(n)}$:

$$f_i = f(\mathbf{u}_i^{(n)}) = \frac{C\left(\mathbf{p}_l^{(i)}\right)^T \mathbf{p}_l^{(i)} + 1}{\left(\left(\mathbf{p}_l^{(i)}\right)^T \mathbf{y}\right)^2} \quad (37)$$

Inner loop: weighted boosting search Set $t = 0$; For $t = t + 1$

Step 1: Boosting

1 Find

$$i_{\mathrm{best}} = \arg \min_{1 \leq i \leq P_S} f_i$$

$$i_{\mathrm{worst}} = \arg \max_{1 \leq i \leq P_S} f_i$$

Denote $\mathbf{u}_{\mathrm{best}}^{(n)} = \mathbf{u}_{i_{\mathrm{best}}}^{(n)}$ and $\mathbf{u}_{\mathrm{worst}}^{(n)} = \mathbf{u}_{i_{\mathrm{worst}}}^{(n)}$.

2 Normalise the cost function values

$$\bar{f}_i = \frac{f_i}{\sum_{m=1}^{P_S} f_m}, \; 1 \le i \le P_S$$

3 Compute a weighting factor β_t according to

$$\eta_t = \sum_{i=1}^{P_S} \delta_i(t-1)\bar{f}_i, \; \beta_t = \frac{\eta_t}{1-\eta_t}$$

4 Update the distribution weightings for $1 \le i \le P_S$

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\bar{f}_i}, & \text{for } \beta_t \le 1 \\ \delta_i(t-1)\beta_t^{1-\bar{f}_i}, & \text{for } \beta_t > 1 \end{cases}$$

and normalise them

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{m=1}^{P_S} \delta_m(t)}, \quad 1 \le i \le P_S$$

Step 2: Parameter updating

1 Construct the $(P_S + 1)$th point using the formula

$$\mathbf{u}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i(t)\mathbf{u}_i^{(n)}$$

2 Construct the $(P_S + 2)$th point using the formula

$$\mathbf{u}_{P_S+2} = \mathbf{u}_{\text{best}}^{(n)} + \left(\mathbf{u}_{\text{best}}^{(n)} - \mathbf{u}_{P_S+1}\right)$$

3 Calculate $\mathbf{g}_l^{(P_S+1)}$ and $\mathbf{g}_l^{(P_S+2)}$ from $\mathbf{u}_{P_S+1}$ and $\mathbf{u}_{P_S+2}$, orthogonalise these two candidate model columns (as in (35) and (36)), and compute their corresponding cost function values f_i, $i = P_S + 1, P_S + 2$ (as in (37)). Then find

$$i_* = \arg \min_{i=P_S+1, P_S+2} f_i$$

4 The pair $(\mathbf{u}_{i_*}, f_{i_*})$ then replaces $(\mathbf{u}_{\text{worst}}^{(n)}, f_{i_{\text{worst}}})$ in the population

If $\|\mathbf{u}_{P_S+1} - \mathbf{u}_{P_S+2}\| < \xi_B$, exit **inner loop**

End of inner loop
The solution found in the nth generation is $\mathbf{u} = \mathbf{u}_{\text{best}}^{(n)}$.

End of outer loop
This yields the solution $\mathbf{u} = \mathbf{u}_{\text{best}}^{(N_G)}$, that is, $\boldsymbol{\mu}_l$ and $\boldsymbol{\Sigma}_l$ of the lth regressor, as well as the corresponding orthogonal model column $\mathbf{p}_l$.

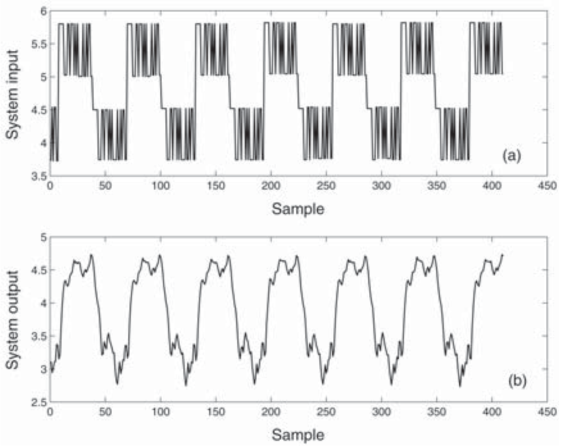
The motivation and analysis of the RWBS algorithm as a global optimiser are detailed in (Chen et al., 2005). To guarantee a global optimal solution as well as to achieve a fast convergence, the algorithmic parameters, P_S, N_G and ξ_B, need to be set carefully. The appropriate values for these algorithmic parameters depends on the dimension of $\mathbf{u}$ and how hard the objective function to be optimised. Generally, these algorithmic parameters have to be found empirically, just as in any global optimisation algorithm. The elitist initialisation adopted in the algorithm is very useful, as it keeps the information obtained by the previous search generation, which otherwise would be lost due to the randomly sampling initialisation. In the inner loop optimisation, there is no need for every members of the population to converge to a (local) minimum and it is sufficient to locate where the minimum lies. Thus, the accuracy for stopping the weighted boosting search, ξ_B, can be set to a relatively large value. This makes the search efficient, achieving convergence with a small number of the cost function evaluations. The number of repeats or generations N_G should be sufficiently large so that the parameter space will be sampled sufficiently.

4 Modelling experiments

Two real data sets were used to demonstrate the effectiveness of the proposed LS-GSESVM construction procedure. The standard SVM algorithm with the ε-insensitive loss function (ε-SVM) (Gunn, 1998) was used as the benchmarker in our modelling experiments.

Example 1 This example constructed a model representing the relationship between the fuel rack position (input $v(t)$) and the engine speed (output $y(t)$) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed. Detailed system description and experimental setup can be found in (Billings et al., 1989). The data set, depicted in Figure 1, contained 410 samples. The first 210 data points were used in training and the last 200 points were used to form the test set. The previous study (Billings et al., 1989) has shown that this data set can be modelled adequately as $y_i = F_S(\mathbf{x}_i) + e_i$, where $y_i = y(i)$ and $\mathbf{x}_i = [y(i-1) \; v(i-1) \; v(i-2)]^T$, $F_S(\bullet)$ describes the unknown underlying system to be identified and e_i denotes the system noise. Five sets of the results were obtained. Firstly, the modelling result of using the standard LS-SVM method of Section 2.1 was obtained. Secondly, the alternative LS-ESVM method of Section 2.2 was used to perform the modelling experiment. In the third experiment, the LS-SESVM method of Section 2.3 was used to obtain a sparse model by applying the standard OLS forward selection to yield a sparse representation and then calculating the kernel weights of the resultant model based on the LS-ESVM method. The fourth modelling result was produced based on the generalised kernel modelling with the generalised Gaussian kernel function of (30), where each regressor had its tunable mean vector and diagonal covariance matrix. The LS-GSESVM algorithm of Section 3 was adopted to construct a sparse model representation. Lastly, the ε-SVM (Gunn, 1998) was employed to produced a spare model and the result obtained was used for comparison with the proposed LS-GSESVM method. For the first three cases and for the ε-SVM, the kernel function, chosen as the Gaussian function of (8), had a common variance σ^2 for every regressor and the regressors were positioned at the training input data points. The previous study (Chen et al., 2003, 2004) has shown that when using the Gaussian kernel model to model this engine data set, the appropriate value of the common kernel variance is $\sigma^2 = 1.69$. This value of σ^2 was thus used in all the four Gaussian kernel modelling cases. For the construction of a generalised Gaussian kernel model, the appropriate values of the RWBS optimisation algorithmic parameters, P_S, N_G and ξ_B, were determined empirically.

Figure 1 The engine data set: (a) system input $v(t)$ and (b) system output $y(t)$



The choices of the regularisation parameter C for the LS-SVM, LS-ESVM, LS-SESVM and LS-GSESVM were first determined. For the two non-sparse modelling methods, the LS-SVM and LS-ESVM, Figure 2 shows the influence of C on modelling performance. A property of the proposed LS-ESVM method can be clearly seen from Figure 2. Both the training and test performance continuously improved as C increased with the rate of improvement slowing down for large C but the ratio of the test MSE over the training MSE was approximately constant over a large range of C values. This was not the case for the standard LS-SVM, whose test performance deteriorated rapidly for $C > 10^4$. From Figure 2, it is clear that $C = 10^4$ is appropriate for the LS-SVM. By contrast, a much larger C, up to 5×10^8, can be used for the LS-ESVM. We chose to use the same $C = 10^4$ for the two non-sparse LS-SVM and LS-ESVM methods and the resulting two models are summarised in Table 1. With $C = 5 \times 10^8$, the LS-SESVM and LS-GSESVM algorithms were used to construct a sparse Gaussian kernel model and a sparse generalised Gaussian kernel model, respectively. Figure 3 depicts the modelling performance as a function of the subset model size for the LS-SESVM method. It can be seen that the training MSE stopped improving after 18 model terms had been selected and at the model size of 22, the training and test MSE values were approximately equal. This suggested that a 22-term model was appropriate. For the LS-GSESVM, Figure 4 shows the modelling performance as a function of the subset model size. The result of Figure. 4 suggested that a 12-term generalised Gaussian kernel model was adequate. The results of using the LS-SESVM and LS-GSESVM are also summarised in Table 1.

Apart from σ^2, the ε-SVM algorithm requires two other learning parameters, the regularisation parameter C and ε value. We first fixed ε to a value of 0.02 and investigated the influence of C to the modelling performance. The results obtained are depicted in Figure 5, where it can be seen that the value $C = 40$ is appropriate. We next used $C = 40$ and examined the influence of ε. The results obtained are shown in Figure 6, which confirms that the appropriate ε value is around 0.02. Given the C value, the model size as well as the modelling performance are obviously functions of ε. With $C = 40$, Figure 7 illustrates the relationship between the model size and ε value, while Figure 8 shows the relationship between the modelling performance and model size. In particular, with $C = 40$ and $\varepsilon = 0.023$, the ε-SVM algorithm constructed a model consisting of 75 Support Vectors (SVs). The performance of this SVM model is compared with those of the other four methods in Table 1, where it is clear that the models produced by all the five methods had similarly good generalisation capabilities. The LS-SESVM method is seen to be capable of producing a sparser model in comparison to the ε-SVM method.

Figure 2 Influence of the regularisation parameter C to the performance of the LS-SVM and LS-ESVM for the engine data set: (a) MSE values over the training and test sets and (b) ratio of the test MSE over the training MSE. The kernel variance is $\sigma^2 = 1.69$

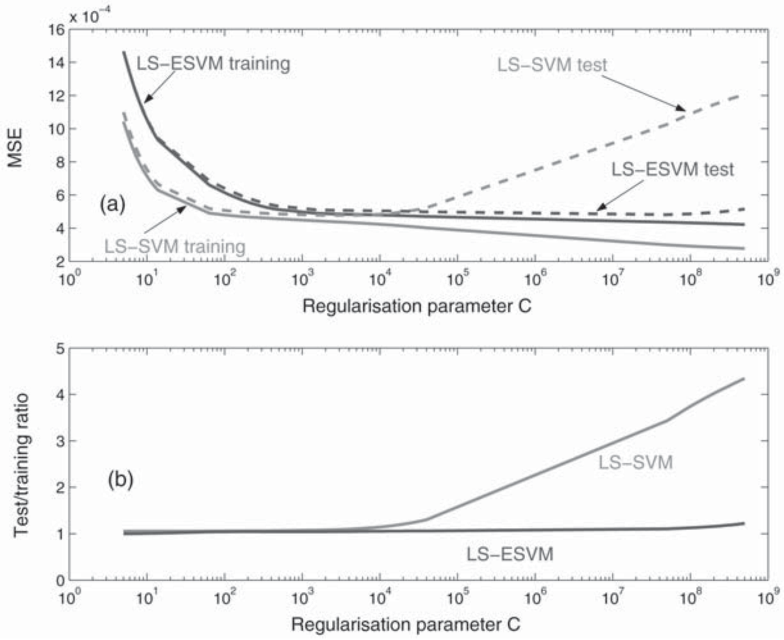
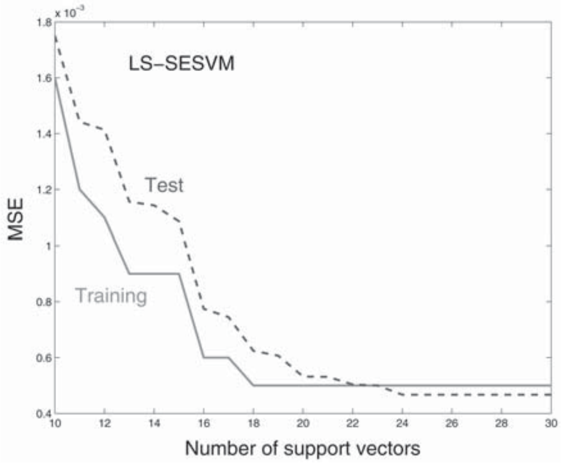


Table 1 Summary of the experimental results for the engine data set

Algorithm	*LS-SVM*	*LS-ESVM*	*LS-SESVM*	*LS-GSESVM*	*ε-SVM*
kernel type	Gaussian	Gaussian	Gaussian	Generalised Gaussian	Gaussian
C value	10^4	10^4	5×10^8	5×10^8	40
ε value	NA	NA	NA	NA	0.023
Kernel variance	1.69	1.69	1.69	NA	1.69
Sparse	No	No	Yes	Yes	Yes
Model size	208	208	22	12	75
Training MSE	4.23×10^{-4}	4.79×10^{-4}	4.71×10^{-4}	5.00×10^{-4}	4.73×10^{-4}
Test MSE	4.85×10^{-4}	5.05×10^{-4}	5.04×10^{-4}	5.08×10^{-4}	4.95×10^{-4}
Training time	2 sec	6 sec	6 sec	6 sec	8 min 33 sec

This confirms our previous observations that the OLS algorithm and the SVM method both have the same excellent generalisation capability but the former is able to produce sparser models than the latter (Chen et al., 2003, 2004). The proposed generalised sparse modelling method is seen to be able to construct a much sparser model than the standard OLS algorithm. The model prediction $\hat{y}_i$ and prediction error $\hat{e}_i = y_i - \hat{y}_i$ for the 12-term generalised Gaussian kernel model constructed by the LS-GSESVM method are illustrated in Figure 9. The performance of the other four models, not shown, are similar to those shown in Figure 9. The generalised kernel modelling approach based on the LS-GSESVM has a clear advantage of producing the sparsest model.

Figure 3 Modelling performance as function of the selected model size for the engine data set using the LS-SESVM. The kernel variance is $\sigma^2 = 1.69$ and the regularisation parameter $C = 5 \times 10^8$



The computational complexity of each algorithm was investigated by measuring the training time that is required for the algorithm to construct a model. The modelling experiment was performed on a low-cost PC with the operating system Windows2000 and the simulation was run using MATLAB 6.1 with Optimisation Tool Boxes. The CPU time was read as the programme executive time recorded by the Windows task manager. The run time measured did not include the tuning time for finding the learning parameters of ε and/or C. With the training conditions as set in Table 1, the training times for all the five algorithms are compared in the same table. It is seen that for this example of a small-size training set (208 points) and a small input dimension ($m = 3$) the proposed LS-GSESVM method compared favourably with the standard SVM method. The LS-GSESVM was about 80 times faster than the ε-SVM under the given computational platform.

Figure 4 Modelling performance as function of the selected model size for the engine data set using the LS-GSESVM. The regularisation parameter is $C = 5 \times 10^8$

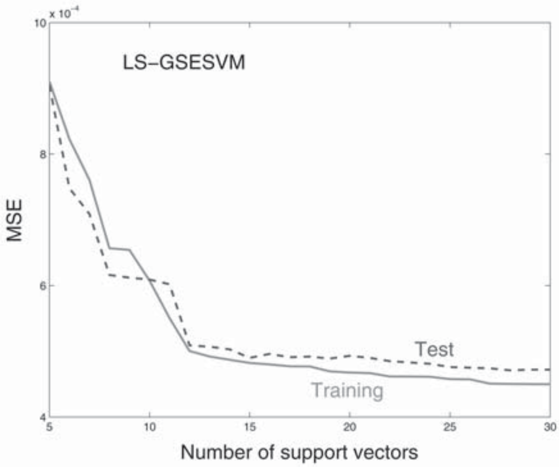


Figure 5 Influence of the regularisation parameter C to the performance of the ε-SVM for the engine data set, given $\varepsilon = 0.02$ and $\sigma^2 = 1.69$

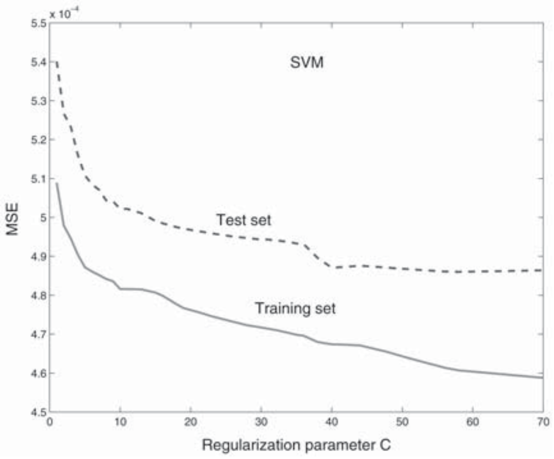


Figure 6 Influence of the ε value to the performance of the ε-SVM for the engine data set, given $C = 40$ and $\sigma^2 = 1.69$

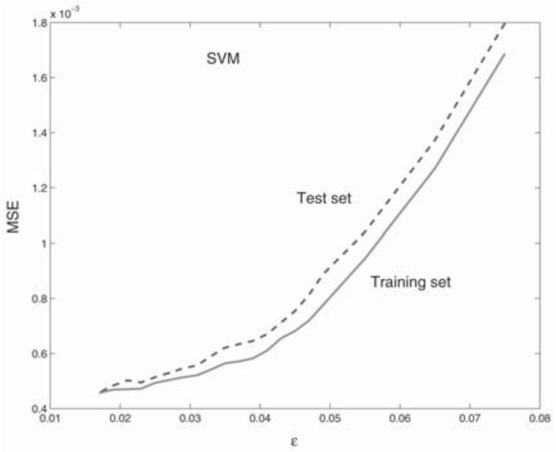


Figure 7 Relationship between the model size and ε value for the engine data set using the ε-SVM. The regularisation parameter is $C = 40$ and kernel variance $\sigma^2 = 1.69$

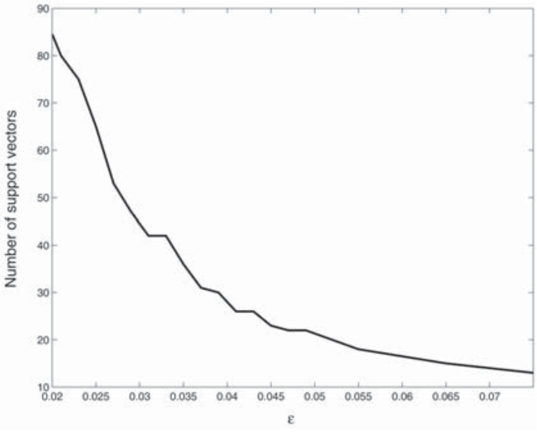
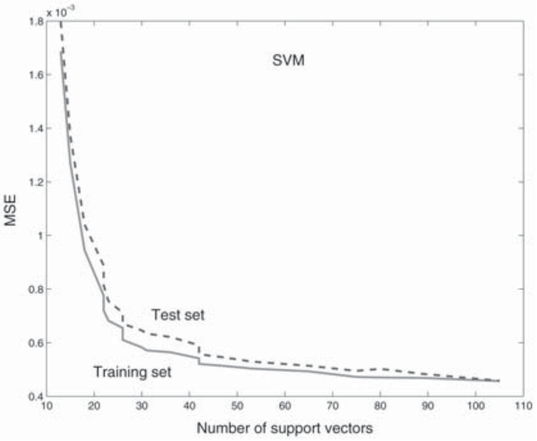
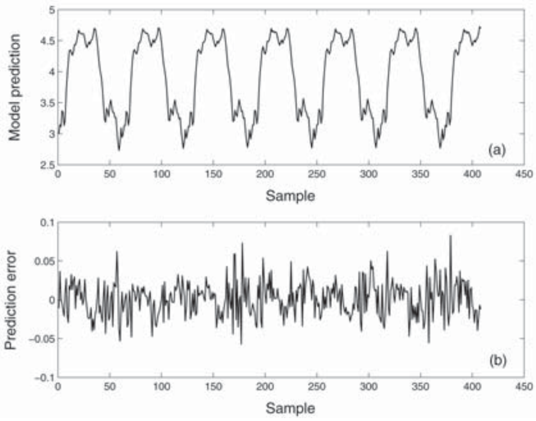


Figure 8 Relationship between the modelling performance and model size for the engine data set using the ε-SVM. The regularisation parameter is $C = 40$ and and kernel variance $\sigma^2 = 1.69$



Example 2. This is a popular regression benchmark data set, Boston Housing, available at the UCI repository (Murphy and Aha; 1992). The data set comprises 506 data points with 14 variables. We performed the task of predicting the median house value from the remaining 13 attributes. We randomly selected 456 data points from the data set for training and used the remaining 50 data points to form the test set. Average results were given over 100 repetitions. The three sparse construction methods, the LS-SESVM, the LS-GSESVM and the ε-SVM, were compared. The learning parameters, σ^2, ε and C for the SVM algorithm, σ^2 and C for the LS-SESVM algorithm and C for the LS-GSESVM algorithm, were determining empirically through cross validation.

Figure 9 The model prediction (a) and prediction error (b) of the 12-term generalised Gaussian kernel model constructed by the LS-GSESVM for the engine data set



Specifically, given one particular set of the data partition, we searched the appropriate values of σ^2, ε and C for the ε-SVM so as to minimise the test MSE. The obtained σ^2, ε and C were then used in the other 99 repetitions. It was found by this cross validation process that $\sigma^2 = 2025$, $\varepsilon = 2$ and $C = 750$ were appropriate for the ε-SVM. Figure 10 depicts the MSE versus ε plot obtained by the SVM algorithm, given $\sigma^2 = 2025$ and $C = 750$. The results of Figure 10 confirm that, with $\sigma^2 = 2025$ and $C = 750$, the appropriate value for ε was 2. By setting $\sigma^2 = 2025$ and $\varepsilon = 2$, the influence of C on the modelling performance was shown in Figure 11, where it can be seen that the appropriate value for C in this case was 750. Similarly, given $C = 750$ and $\varepsilon = 2$, Figure 12 shows the MSE versus σ^2 plot. It can be seen that the appropriate value for the kernel variance was $\sigma^2 = 2025$. Since the extensive search for the single common Gaussian kernel variance had been performed for the SVM method and the appropriate value found was $\sigma^2 = 2025$, we also chose to use this value for the LS-SESVM method. The search for an appropriate value of the regularisation parameter C for the LS-SESVM was then carried out and the Figure 13 depicts the modelling performance as a function of C using the LS-SESVM in one run, given $\sigma^2 = 2025$. It can be seen from Figure 13 that a very large value of $C = 1120$ could be used for the LS-SESVM. A similar search process was performed for the LS-GSESVM and it was found that the value of $C = 1120$ was also adequate for the LS-GSESVM.

Table 2 compares the mean modelling performance as well as the mean model sizes averaged over 100 runs together with their associated standard deviations, obtained by the three sparse modelling methods. It can be seen that the ε-SVM algorithm achieved a slightly better generalisation

Figure 10 Influence of the ε value to the performance of the ε-SVM for the Boston Housing data set, given $\sigma^2 = 2025$ and $C = 750$, in one run

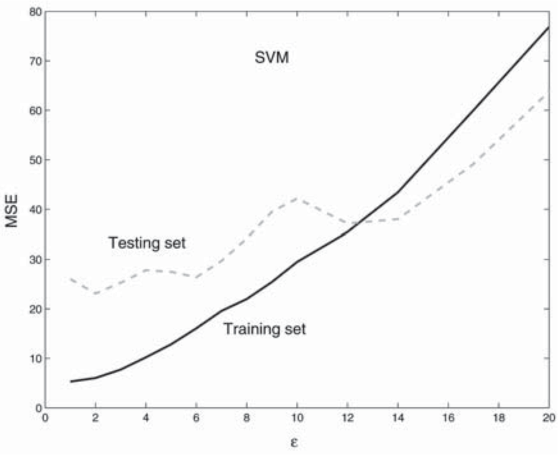


Figure 11 Influence of the regularisation parameter C to the performance of the ε-SVM for the Boston Housing data set, given $\sigma^2 = 2025$ and $\varepsilon = 2$, in one run

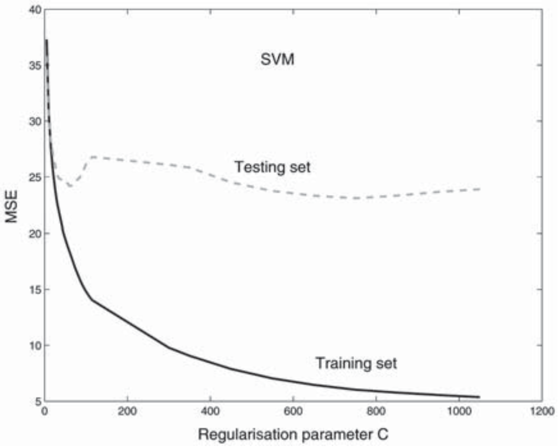


Figure 12 Influence of the kernel variance σ^2 to the performance of the ε-SVM for the Boston Housing data set, given $C = 750$ and $\varepsilon = 2$, in one run

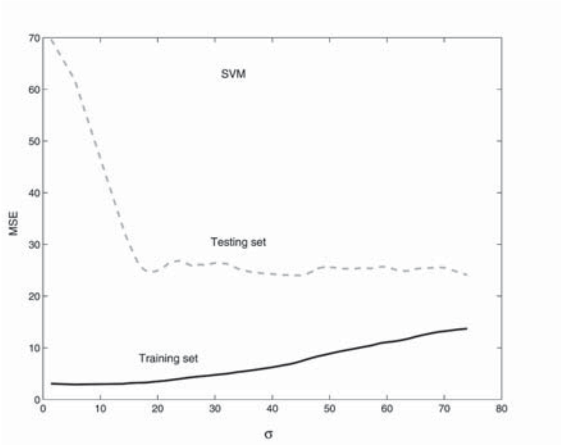
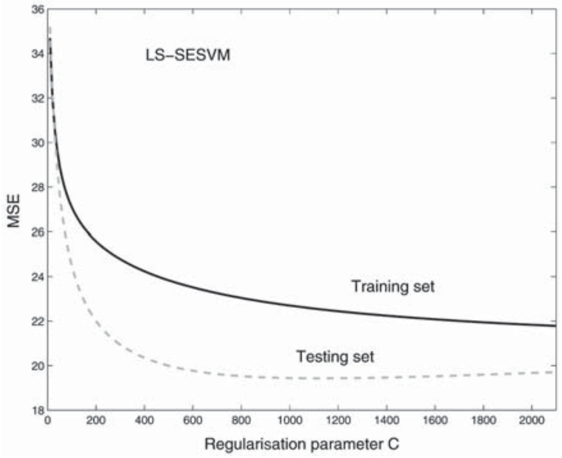


Figure 13 Influence of the regularisation parameter C to the performance of the LS-SESVM for the Boston Housing data set, given $\sigma^2 = 2025$, in one run



performance than the LS-SESVM method for this example and it also appeared to be more robust than the latter as shown by their associated estimation standard deviations. However, the LS-SESVM method arrived at a much sparser model than the SVM method. The proposed LS-GSESVM approach is seen not only to produce the sparsest model but also to have the best model generalisation performance. The average model size obtained by the LS-GSESVM method is less than 4% of the SVM model size and its test MSE is less than 72% of the test MSE achieved by the SVM method. We also compared the computational complexity of the three construction algorithms by recording the training times required in typical one-run using the three methods. That is, the CPU times given in Table 2 were corresponding to the training time for the LS-SESVM to obtain a model of 143 terms, the training time for the LS-GSESVM to derive a model of 10 terms and the training time for the ε-SVM to result in a model of 244 terms. It can be seen that for

Table 2 Summary of the experimental results for the Boston Housing data set

algorithm	*LS-SESVM*	*LS-GSESVM*	*ε-SVM*
Kernel type	Gaussian	Generalised Gaussian	Gaussian
Regularisation C	1120	1120	750
ε value	NA	NA	2
Kernel variance	2025	NA	2025
Model size (mean±std)	142.5 ± 52.4	9.5 ± 1.7	243.2 ± 5.3
Training MSE (mean±std)	14.3628 ± 2.8242	13.0098 ± 3.1448	6.7986 ± 0.4444
Test MSE (mean±std)	24.8421 ± 11.3903	16.6487 ± 7.0692	23.1750 ± 9.0459
Typical training time	1 min 35 sec	1 min 56 sec	2 hr 13 min 20 sec

this benchmark example of a modest-size training set (456 points) and a relatively large input dimension ($m = 13$) our novel LS-GSESVM algorithm was about 70 times faster than the standard SVM method under the specific computational environment.

5 Conclusion

The contributions of this paper are threefold. Firstly, we have considered an alternative LS-SVM formulation, referred to as the LS-ESVM, which does not assume the reproducing kernel Hilbert space and can be applied to non-Mercer kernels. Secondly, a sparse kernel model construction algorithm, called the LS-SESVM, has been proposed. In this approach a parsimonious representation is selected using the standard OLS forward selection procedure and the corresponding model weights are then computed using the LS-ESVM formulation. In our modelling experiments, the LS-SESVM method has been shown to have a similarly good generalisation capability as the standard SVM algorithm but it is able to produce sparser model representations than the SVM method. Thirdly, which is a major contribution of our work, the generalised kernel modelling has been derived where each kernel regressor has its tunable centre vector and diagonal covariance matrix. An orthogonal forward selection procedure has been proposed to incrementally construct a sparse generalised kernel model representation. At each model construction stage, a kernel regressor is optimised using a guided random search optimisation algorithm. Again the corresponding model weights are then calculated using the LS-ESVM formulation. Our modelling experimental results have clearly demonstrated the advantage of this proposed novel modelling technique to produce very sparse models that generalise well. The proposed LS-GSESVM approach has also been shown to be much faster in constructing a model than the standard SVM approach for the two real-data sets.

References

Akaike, H. (1974) 'A new look at the statistical model identification', *IEEE Transactions on Automatic Control*, Vol. AC-19, pp.716–723.

Aronszajn, N. (1950) 'Theory of reproducing kernels', *Transactions of the American Mathematical Society*, Vol. 68, pp.337–404.

Billings, S.A., Chen, S. and Backhouse, R.J. (1989) 'The identification of linear and non-linear models of a turbocharged automotive diesel engine', *Mechanical Systems and Signal Processing*, Vol. 3, No. 2, pp.123–142.

Breiman, L. (1999) 'Prediction games and arcing algorithms', *Neural Computation*, Vol. 11, No. 7, pp.1493–1518.

Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S. (2002) 'Choosing multiple parameters for support vector machines', *Machine Learning*, Vol. 46, Nos. 1–3, pp.131–159.

Chen, S., Billings, S.A. and Luo, W. (1989) 'Orthogonal least squares methods and their application to non-linear system identification', *International Journal of Control*, Vol. 50, No. 5, pp.1873–1896.

Chen, S., Cowan, C.F.N. and Grant, P.M. (1991) 'Orthogonal least squares learning algorithm for radial basis function networks', *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp.302–309.

Chen, S. and Luk, B.L. (1999) 'Adaptive simulated annealing for optimization in signal processing applications', *Signal Processing*, Vol. 79, No. 1, pp.117–128.

Chen, S., Wu, Y. and Luk, B.L. (1999) 'Combined genetic algorithm optimisation and regularised orthogonal least squares learning for radial basis function networks', *IEEE Transactions on Neural Networks*, Vol. 10, No. 5, pp.1239–1243.

Chen, S., Hong, X. and Harris, C.J. (2003) 'Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design', *IEEE Transactions on Automatic Control*, Vol. 48, No. 6, pp.1029–1036.

Chen, S., Hong, X., Harris, C.J. and Sharkey, P.M. (2004) 'Sparse modelling using orthogonal forward regression with PRESS statistic and regularization', *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol. 34, No. 2, pp.898–911.

Chen, S., Wang, X.X. and Harris, C.J. (2005) 'Experiments with repeating weighted boosting search for optimization in signal processing applications', *IEEE Transactions on Systems, Man and Cybernetics, Part B*, Vol. 35, No. 4, pp.682–693.

Cristianini, N. and Shawe-Taylor, J. (2000) *An Introduction to Support Vector Machines: And Other Kernel-Based Learning Methods*, Cambridge, UK: Cambridge University Press.

de Kruif, B.J. and de Vries, T.J.A. (2003) 'Pruning error minimization in least squares support vector machines', *IEEE Transactions on Neural Networks*, Vol. 14, No. 3, pp.696–702.

Duan, K., Keerthi, S.S. and Poo, A.N. (2003) 'Evaluation of simple performance measures for tuning SVM hyperparameters', *Neurocomputing*, Vol. 51, pp.41–59.

Freund, Y. and Schapire, R.E. (1997) 'A decision-theoretic generalization of on-line learning and an application to boosting', *Journal of Computer and System Sciences*, Vol. 55, No. 1, pp.119–139.

Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, MA: Addison Wesley.

Gunn, S. (1998) 'Support vector machines for classification and regression', *Technical Report*, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, UK, May 1998.

Hong, X. and Harris, C.J. (2002) 'Nonlinear model structure design and construction using orthogonal least squares and D-optimality design', *IEEE Transactions on Neural Networks*, Vol. 13, No. 5, pp.1245–1250.

Hong, X., Harris, C.J., Chen, S. and Sharkey, P.M. (2003) 'Robust nonlinear model identification methods using forward regression', *IEEE Transactions on Systems, Man and Cybernetics, Part A*, Vol. 33, No. 4, pp.514–523.

Ingber, L. (1993) 'Simulated annealing: practice versus theory', *Mathematical and Computer Modeling*, Vol. 18, No. 11, pp.29–57.

Lanckriet, R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E. and Jordan, M.I. (2004) 'Learning the kernel matrix with semidefinite programming', *Journal of Machine Learning Research*, Vol. 5, pp.27–72.

Leontaritis, I.J. and Billings, S.A. (1987) 'Model selection and validation methods for non-linear systems', *International Journal of Control*, Vol. 45, No. 1, pp.311–341.

Man, K.F., Tang, K.S. and Kwong, S. (1998) *Genetic Algorithms: Concepts and Design*, London: Springer-Verlag.

Meir, R. and Rätsch, G. (2003) 'An introduction to boosting and leveraging', in S. Mendelson and A. Smola (Eds). *Advanced Lectures in Machine Learning*. Springer Verlag, pp.119–184.

Murphy, P.M. and Aha, D.W. (1992) UCI Repository of Machine Learning Databases Available at: http:// www. ics.uci.edu/~mlearn/MLRepository.html.

Ong, C.S., Smola, A.J. and Williamson, R.C. (2005) 'Learning the kernel with hyperkernels', *Journal of Machine Learning Research*, Vol. 6, pp.1043–1071.

Schapire, R.E. (1990) 'The strength of weak learnability', *Machine Learning*, Vol. 5, No. 2, pp.197–227.

Schölkopf, B., Sung, K.K., Burges, C.J.C., Girosi, F., Niyogi, P., Poggio, T. and Vapnik, V. (1997) 'Comparing support vector machines with Gaussian kernels to radial basis function classifiers', *IEEE Transactions on Signal Processing*, Vol. 45, No. 11, pp.2758–2765.

Schölkopf, B., Smola, A.J., Williamson, R.C. and Bartlett, P.L. (2000) 'New support vector algorithms', *Neural Computation*, Vol. 12, No. 5, pp.1207–1245.

Schölkopf, B. and Smola, A.J. (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, Cambridge, MA: MIT Press.

Suykens, J.A.K. and Vandewalle, J. (1999) 'Least squares support vector machine classifiers', *Neural Processing Letters*, Vol. 9, No. 3, pp.293–300.

Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J. (2002) *Least Squares Support Vector Machines*, Singapore: World Scientific.

Van Gestel, T., Suykens, J.A.K., Baestaens, D-E., Lambrechts, A., Lanckriet, G., Vandaele, B., De Moor, B. and Vandewalle, J. (2001) 'Financial time series prediction using least squares support vector machines within the evidence framework', *IEEE Transactions on Neural Networks*, Vol. 12, No. 4, pp.809–821.

Vapnik, V. (1995) *The Nature of Statistical Learning Theory*, New York: Springer-Verlag.

Vapnik, V., Golowich, S. and Smola, A. (1997) 'Support vector method for function approximation, regression estimation and signal processing', in M.C. Mozer, M.I. Jordan and T. Petsche (Eds). *Advances in Neural Information Processing Systems 9*, Cambridge, MA: MIT Press, pp.281–287.

Vincent, P. and Bengio, Y. (2002) 'Kernel matching pursuit', *Machine Learning*, Vol. 48, No. 1, pp.165–187.