Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 462–474
NEUROCOMPUTING
www.elsevier.com/locate/neucom

# Sparse support vector regression based on orthogonal forward selection for the generalised kernel model

X.X. Wang[a], S. Chen[b,*], D. Lowe[a], C.J. Harris[b]

[a]*Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK*
[b]*School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*

Received 4 February 2005; received in revised form 9 December 2005; accepted 17 December 2005
Communicated by J. Zhang
Available online 18 May 2006

**Abstract**

This paper considers sparse regression modelling using a generalised kernel model in which each kernel regressor has its individually tuned centre vector and diagonal covariance matrix. An orthogonal least squares forward selection procedure is employed to select the regressors one by one, so as to determine the model structure. After the regressor selection, the corresponding model weight parameters are calculated from the Lagrange dual problem of the original regression problem with the regularised $\varepsilon$-insensitive loss function. Unlike the support vector regression, this stage of the 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, sparser representation can be obtained. Experiments involving one simulated example and three real data sets are used to demonstrate the effectiveness of the proposed novel regression modelling approach.


*Keywords:* Generalised kernel model; Orthogonal least squares forward selection; Regression; Sparse modelling; Support vector machine

## 1. Introduction

Objective of modelling from data is not that a model should fit well to the training data. Rather, the goodness of a model is characterised by its generalisation capability, and the model should be easy to interpret and to extract knowledge from. All these vital properties depend on crucially the ability of a modelling process to obtain appropriately sparse representations. Forward selection using the orthogonal least squares (OLS) algorithm [7–10,13] is a simple and efficient method that is capable of producing parsimonious linear-in-the-weights nonlinear models with excellent generalisation performance. Alternatively, the state-of-the-art sparse kernel modelling techniques, such as the support vector machine (SVM) [31,18,26–29,14,6,15], have become popular in data modelling applications. Originated from maximum margin linear classification, one of the main features of the SVM is to use hyperplane. Specifically, the training data are mapped to a high dimensional space where they can be approximated by a hyperplane. In classification, this hyperplane is adjusted to obtain the maximum classification margin. In regression, the gradient of this hyperplane is kept as small as possible. More precisely, in a SVM regression problem, the parameter of the hyperplane is obtained by minimising the cost consisting of the linear $\varepsilon$-insensitive loss function and the squared gradient of the hyperplane [18].

With the aid of the reproducing kernel Hilbert space through Mercer theorem [2], Mercer kernel can be used, and the required mapping from the input space to the high dimensional space is given implicitly by this kernel function. A common feature of the SVM regression modelling techniques as well as the OLS kernel modelling methods [7–10,13] is that the kernel centres are 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

*Corresponding author.
*E-mail addresses:* x.wang@aston.ac.uk (X.X. Wang), sqc@ecs.soton.ac.uk (S. Chen), d.lowe@aston.ac.uk (D. Lowe), cjh@ecs.soton.ac.uk (C.J. Harris).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.12.129

on the sparsity and generalisation capability of the resulting model, and it has to be determined via 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 kernel function at a position not occupied by a train data point or giving different kernel regressors at different positions different covariance matrices are not allowed for the SVM methods that use Mercer theorem. Also this "generalised" kernel model will change the "linear" learning problem associated with the SVM-type models to a nonlinear one.

Unlike the SVM formulation, the method proposed in this paper minimises the cost consisting of the linear $\varepsilon$-insensitive loss function and the squared weights of the regressors. This formulation allows the use of non-Mercer kernels. Specifically, the "generalised" kernel function is used in which each kernel regressor has its tunable centre vector and diagonal covariance matrix. To arrive at a sparse representation, the OLS forward selection procedure is adopted to select regressors one by one by incrementally minimising the training mean square error (MSE). Unlike the standard OLS algorithm [7], however, at each stage of selection the optimisation is with respect to the kernel centre vector and diagonal covariance matrix, and the determination of these kernel parameters is performed using a guided random search algorithm called the repeated weighted boosting search (RWBS) algorithm [12], which has its root from boosting optimisation [30,16,5,25]. Thus, regression modelling is carried out by a "kernel hunting". The "support vectors" are selected by the OLS criterion and, unlike the SVM, the number of regressors is not controlled by the $\varepsilon$ value of the $\varepsilon$-insensitive loss function. After the selection of a parsimonious model representation, the kernel weights are then calculated from the Lagrange dual of the original minimisation problem. This proposed generalised kernel regression modelling approach has the potential of improving modelling capacity and producing sparser final models, compared with the standard SVM algorithm. The advantages of the proposed method are illustrated using a simulated example and three real-data sets.

The remaining of the paper is organised as follows. Section 2 reviews the standard kernel regression modelling, which positions the kernel centres at the training input data points and adopts a single common variance for every kernel regressors. The classical SVM formulation is first summarised. An alternative Lagrange dual problem of the general SVM problem is then considered, which does not restrict to the use of Mercer kernels. This method will be referred to as the extended SVM (ESVM). Unlike the standard SVM method, the solution obtained by the ESVM is not sparse. To derive a sparse representation, the standard OLS algorithm [7] is used to select a parsimonious model, and this is followed by solving the corresponding sparse ESVM problem to yield the model weight parameters. This method will be referred to as the sparse extended SVM (SESVM). 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 [12] to determine the kernel centres and diagonal covariance matrices. This guarantees a sparse representation. Again, the kernel weights are solved from a similar ESVM problem after obtaining a sparse representation. This proposed new method will be called the generalised sparse extended SVM (GSESVM) for the purpose of comparison with the methods of Section 2. Section 4 provides the results of our modelling experiments, while Section 5 summarises 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 $i$th 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 SVM method solves this problem by using the following strategy.

### 2.1. Support vector machine regression problem

The minimisation problem of the SVM method using the linear $\varepsilon$-insensitive loss function [18] can be stated as below:

$$\min J(\mathbf{w}, \boldsymbol{\xi}^*, \boldsymbol{\xi}) = \min\left\{\frac{1}{2}\bar{\mathbf{w}}^{\mathrm{T}}\bar{\mathbf{w}} + C\left(\sum_{i=1}^{N}\xi_i^* + \sum_{i=1}^{N}\xi_i\right)\right\}, \tag{1}$$

$$\text{subject to} \quad \begin{cases} y_i - \bar{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x}_i) - b \leqslant \varepsilon + \xi_i^*, & 1 \leqslant i \leqslant N, \\ \bar{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x}_i) + b - y_i \leqslant \varepsilon + \xi_i, & 1 \leqslant i \leqslant N, \\ \xi_i^* \geqslant 0, & 1 \leqslant i \leqslant N, \\ \xi_i \geqslant 0, & 1 \leqslant i \leqslant N, \end{cases} \tag{2}$$

where $\boldsymbol{\varphi}(\mathbf{x})$ is the selected mapping from the input space to the high-dimensional space, $y = \bar{\mathbf{w}}^{\mathrm{T}}\boldsymbol{\varphi}(\mathbf{x}) + b$ is the linear regression function (hyperplane) in the high-dimensional space with $\bar{\mathbf{w}}$ as its gradient, $C$ is a pre-specified value that defines regularisation, $\boldsymbol{\xi} = [\xi_1\ \xi_2\ \cdots\ \xi_N]^{\mathrm{T}}$ and $\boldsymbol{\xi}^* = [\xi_1^*\ \xi_2^*\ \cdots\ \xi_N^*]^{\mathrm{T}}$ are stack variables representing upper and lower constraints on the system outputs, and $\varepsilon$ is a given value that defines the $\varepsilon$-insensitive loss function.

Let us define the Mercer kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \langle \boldsymbol{\varphi}(\mathbf{x}_i), \boldsymbol{\varphi}(\mathbf{x}_j) \rangle \tag{3}$$

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

$$\max \bar{L}(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}) = \max\left\{-\varepsilon\sum_{i=1}^{N}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(\mathbf{x}_i, \mathbf{x}_j)\right\}, \tag{4}$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0, & \\ 0 \leqslant \alpha_i^* \leqslant C, & 1 \leqslant i \leqslant N, \\ 0 \leqslant \alpha_i \leqslant C, & 1 \leqslant i \leqslant N. \end{cases} \tag{5}$$

After obtaining the Lagrange multipliers, $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \cdots\ \alpha_N]^{\mathrm{T}}$ and $\boldsymbol{\alpha}^* = [\alpha_1^*\ \alpha_2^*\ \cdots\ \alpha_N^*]^{\mathrm{T}}$, and the bias term $b$, the regression model is given by

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)k(\mathbf{x}_i, \mathbf{x}) + b. \tag{6}$$

It is well known that the use of the $\varepsilon$-insensitive cost function leads to a more robust parameter estimate, compared with the conventional least squares cost function. The choice of the $\varepsilon$-insensitive loss function is also attractive because many of the "weights" $\alpha_i^* - \alpha_i$ become zero, leading to a sparse solution in (6). One of the most common choices of kernel function is the Gaussian function of the form:

$$k(\mathbf{x}_i, \mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right). \tag{7}$$

The common kernel variance $\sigma^2$ is not provided by the algorithm and has to be determined by other means, such as via cross validation.

### 2.2. Dual of the minimisation problem with $\varepsilon$-insensitive loss function and squared regressor weights

Consider the modelling of the training data set $\{\mathbf{x}_i, y_i\}_{i=1}^N$ with the regression model

$$\hat{\mathbf{y}}(\mathbf{x}) = \sum_{i=1}^{N} w_i h_i(\mathbf{x}) + b, \tag{8}$$

where $w_i$ is the $i$th model weight, and $h_i(\mathbf{x})$ is the $i$th kernel regressor centred at the training input $\mathbf{x}_i$. By adopting the combined cost function of the $\varepsilon$-insensitive loss function and the squared regressor weights, the following minimisation problem can be established:

$$\min J(\mathbf{w}, \boldsymbol{\xi}^*, \boldsymbol{\xi}) = \min\left\{\frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\left(\sum_{i=1}^{N}\xi_i^* + \sum_{i=1}^{N}\xi_i\right)\right\}, \tag{9}$$

$$\text{subject to} \quad \begin{cases} y_i - \sum_{j=1}^{N} w_j h_j(\mathbf{x}_i) - b \leqslant \varepsilon + \xi_i^*, & 1 \leqslant i \leqslant N, \\ \sum_{j=1}^{N} w_j h_j(\mathbf{x}_i) + b - y_i \leqslant \varepsilon + \xi_i, & 1 \leqslant i \leqslant N, \\ \xi_i^* \geqslant 0, & 1 \leqslant i \leqslant N, \\ \xi_i \geqslant 0, & 1 \leqslant i \leqslant N. \end{cases} \tag{10}$$

Although the optimisation problem (9) and (10) appears to have the same form as that of (1) and (2), these two problems are different. Let us define $\mathbf{w} = [w_1\ w_2\ \cdots\ w_N]^{\mathrm{T}}$ and

$$\mathbf{h}(\mathbf{x}_i) = [h_1(\mathbf{x}_i)\ h_2(\mathbf{x}_i)\ \cdots\ h_N(\mathbf{x}_i)]^{\mathrm{T}}. \tag{11}$$

The Lagrangian of the minimisation problem (9) and (10) can be written as

$$\begin{aligned} L(\mathbf{w}, \boldsymbol{\xi}^*, \boldsymbol{\xi}, \boldsymbol{\alpha}^*, \boldsymbol{\alpha}, \boldsymbol{\gamma}^*, \boldsymbol{\gamma}) &= \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}(\xi_i^* + \xi_i) - \sum_{i=1}^{N}(\gamma_i^*\xi_i^* + \gamma_i\xi_i) \\ &\quad - \sum_{i=1}^{N}\alpha_i(y_i - \mathbf{w}^{\mathrm{T}}\mathbf{h}(\mathbf{x}_i) - b + \varepsilon + \xi_i) \\ &\quad - \sum_{i=1}^{N}\alpha_i^*(\mathbf{w}^{\mathrm{T}}\mathbf{h}(\mathbf{x}_i) + b - y_i + \varepsilon + \xi_i^*), \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha} = [\alpha_1\ \alpha_2\ \cdots\ \alpha_N]^{\mathrm{T}}$, $\boldsymbol{\alpha}^* = [\alpha_1^*\ \alpha_2^*\ \cdots\ \alpha_N^*]^{\mathrm{T}}$, $\boldsymbol{\gamma} = [\gamma_1\ \gamma_2 \cdots\ \gamma_N]^{\mathrm{T}}$ and $\boldsymbol{\gamma}^* = [\gamma_1^*\ \gamma_2^*\ \cdots\ \gamma_N^*]^{\mathrm{T}}$ are the Lagrange multipliers. From the Kuhn–Tucker conditions, we have

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\mathbf{h}(\mathbf{x}_i) = \mathbf{0}, \tag{13}$$

$$\frac{\partial L}{\partial \boldsymbol{\xi}^*} = [C\ C\ \cdots\ C]^{\mathrm{T}} - \boldsymbol{\alpha}^* - \boldsymbol{\gamma}^* = \mathbf{0}, \tag{14}$$

$$\frac{\partial L}{\partial \boldsymbol{\xi}} = [C\ C\ \cdots\ C]^{\mathrm{T}} - \boldsymbol{\alpha} - \boldsymbol{\gamma} = \mathbf{0}, \tag{15}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0. \tag{16}$$

Substituting the Kuhn–Tucker conditions into Lagrangian (12) leads to the dual problem of the primal problem (9) and (10):

$$\begin{aligned} \max \bar{L}(\boldsymbol{\alpha}^*, \boldsymbol{\alpha}) = \max\Bigg\{ & -\varepsilon\sum_{i=1}^{N}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i) \\ & - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\mathbf{h}^{\mathrm{T}}(\mathbf{x}_i)\mathbf{h}(\mathbf{x}_j)\Bigg\}, \end{aligned} \tag{17}$$

$$\text{subject to} \quad \begin{cases} \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0, & \\ 0 \leqslant \alpha_i^* \leqslant C, & 1 \leqslant i \leqslant N, \\ 0 \leqslant \alpha_i \leqslant C, & 1 \leqslant i \leqslant N. \end{cases} \tag{18}$$

After obtaining $\boldsymbol{\alpha}^*$ and $\boldsymbol{\alpha}$, we can calculate the model weights from (13) as

$$\mathbf{w} = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)\mathbf{h}(\mathbf{x}_i). \tag{19}$$

The key difference between the minimisation problem (9) and (10) and the SVM one given in (1) and (2) is that here regularisation directly controls the kernel weights, but not the gradient of the unseen hyperplane as is in the case of (1) and (2). Thus, this approach does not impose any restriction on the kernel function used. We refer to this approach as the ESVM method to contrast with the SVM method discussed in Section 2.1.

### 2.3. Construction of sparse ESVM models

One drawback of the aforementioned ESVM method is that solution (19) is generally non-sparse. To obtain a sparse model, we propose first to use the OLS algorithm [7] to select a parsimonious subset model from the full regression model (8). Without the loss of generality, we will assume the bias term $b = 0$ in model (8). In fact, this bias term can be regarded as a constant regressor. The regression model (8) over the training set can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{w} + \mathbf{e}, \tag{20}$$

where $\mathbf{y} = [y_1\ y_2\ \cdots\ y_N]^{\mathrm{T}}$, $\mathbf{e} = [e_1\ e_2\ \cdots\ e_N]^{\mathrm{T}}$ with $e_i = y_i - \mathbf{w}^{\mathrm{T}}\mathbf{h}(\mathbf{x}_i)$ denoting the modelling error at the input $\mathbf{x}_i$, and

$$\mathbf{H} = [\mathbf{h}_1\ \mathbf{h}_2\ \cdots\ \mathbf{h}_N] \tag{21}$$

is the regression matrix with the regressor columns or model bases defined by

$$\mathbf{h}_i = [h_i(\mathbf{x}_1)\ h_i(\mathbf{x}_2)\ \cdots\ h_i(\mathbf{x}_N)]^{\mathrm{T}}, \quad 1 \leqslant i \leqslant N. \tag{22}$$

Let an orthogonal decomposition of $\mathbf{H}$ be

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

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

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

The regression model (20) can alternatively be expressed as

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

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

The sum of squared errors for this $N$-term regression model can be expressed as [7]

$$J_N = \mathbf{e}^{\mathrm{T}}\mathbf{e} = \mathbf{y}^{\mathrm{T}}\mathbf{y} - \sum_{i=1}^{N} \frac{(\mathbf{y}^{\mathrm{T}}\mathbf{p}_i)^2}{\mathbf{p}_i^{\mathrm{T}}\mathbf{p}_i}. \tag{26}$$

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

$$\mathrm{ER}_j = \frac{(\mathbf{y}^{\mathrm{T}}\mathbf{p}_j)^2}{\mathbf{p}_j^{\mathrm{T}}\mathbf{p}_j}. \tag{27}$$

Based on this error reduction criterion, a subset model can be obtained in a forward selection procedure [7]. At the $l$th selection stage, a model term is selected from the remaining candidates $\mathbf{p}_j$, $l \leqslant j \leqslant N$, as the $l$th model term in the subset model, if it maximises the error reduction criterion $\mathrm{ER}_j$. The details of the selection algorithm are readily available in [7–10,13] and is not repeated here. The selection is terminated at the $N_{\mathrm{s}}$ stage if the MSE

$$\frac{1}{N} J_{N_{\mathrm{s}}} \leqslant \zeta, \tag{28}$$

where the small positive tolerance value $\zeta$ controls the sparsity level of the selected subset model. This produces a parsimonious model containing $N_{\mathrm{s}}$ terms. Appropriate value for $\zeta$ is problem dependent and may be learnt via cross validation. Alternatively, the Akaike information criterion [1,23] can be adopted to terminate the subset model selection procedure. Moreover, the optimal experimental design criteria can be combined with the least squares cost (26) to automatically terminate the selection with an appropriate $N_{\mathrm{s}}$-term subset model without the need for the user to specify a tolerance value $\zeta$ [9,19,20]. It should also be pointed out that regularisation can naturally be incorporated into this OLS forward selection procedure [9].

As is in the standard kernel regression modelling, each kernel regressor is positioned at a training input data point and a single common kernel variance $\sigma^2$ is used for every regressors. Using the OLS forward selection procedure described above, we first obtain a sparse representation containing $N_{\mathrm{s}}$ kernel regressors. The corresponding kernel weights are then calculated using the ESVM method of Section 2.2. We will referred to this approach of constructing sparse kernel models as the 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, in which the regressors take the form:

$$h_j(\mathbf{x}) = g\left(\sqrt{(\mathbf{x} - \boldsymbol{\mu}_j)^{\mathrm{T}}\boldsymbol{\Sigma}_j^{-1}(\mathbf{x} - \boldsymbol{\mu}_j)}\right), \tag{29}$$

where $1 \leqslant j \leqslant M$, $\boldsymbol{\mu}_j$ is the mean vector of the $j$th kernel, $\boldsymbol{\Sigma}_j = \mathrm{diag}\{\sigma_{j,1}^2, \sigma_{j,2}^2, \ldots, \sigma_{j,m}^2\}$ its diagonal covariance matrix, $M$ is the number of regressors in the model, and $g(\bullet)$ a chosen kernel function.

### 3.1. Construction of sparse generalised kernel models

We propose a construction procedure for obtaining sparse generalised kernel models by adopting an orthogonal forward selection to append the regressors one by one. At the $l$th stage of model construction, the $l$th kernel regressor is determined by maximising the following error

reduction criterion:

$$\mathrm{ER}_l(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) = \frac{(\mathbf{y}^{\mathrm{T}}\mathbf{p}_l)^2}{\mathbf{p}_l^{\mathrm{T}}\mathbf{p}_l}, \tag{30}$$

where $\mathbf{p}_l$ is obtained by an orthogonal transformation of the $l$th model column $\mathbf{h}_l = [h_l(\mathbf{x}_1)\ h_l(\mathbf{x}_2)\ \cdots\ h_l(\mathbf{x}_N)]^{\mathrm{T}}$ via

$$\mathbf{p}_l = \mathbf{h}_l - \sum_{j=1}^{l-1} d_{j,l}\mathbf{p}_j \tag{31}$$

and $\mathbf{p}_j$, $1 \leqslant j \leqslant l-1$, 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 the MSE

$$\frac{1}{N}J_{M_s} = \frac{1}{N}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \frac{1}{N}\sum_{l=1}^{M_s}\mathrm{ER}_l(\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l) \leqslant \zeta \tag{32}$$

yielding an $M_s$-term generalised kernel model. The corresponding kernel weights can readily be calculated using the ESVM method of Section 2.2. For a comparison purpose, we will call this construction approach the GSESVM method.

### 3.2. Determination of the generalised kernel parameters

It can be seen that at each regression stage, the task is to determine the generalised kernel parameters $\mathbf{u}$ so as to minimise the cost function

$$f(\mathbf{u}) = \frac{1}{\mathrm{ER}_l(\mathbf{u})}, \tag{33}$$

where the parameter vector $\mathbf{u}$ 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 standard global optimisation methods, such as the genetic algorithm [17,24] and adaptive simulated annealing [22,11], can be used. We have developed a simple and effective guided global search method called the RWBS algorithm [12], which is adopted to perform this optimisation task. The algorithm for determining the generalised kernel parameters at each incremental model stage is summarised as follows.

*Repeated weighted boosting search*: Specify the following algorithmic parameters: $P_S$—population size, $N_G$—number of generations in the repeated search, and $\zeta_I$—accuracy for terminating the weighted boosting search.

**Outer loop: generations** For $k = 1 : N_G$

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

*Weighted boosting search initialisation*: Assign the initial distribution weightings $\delta_i(0) = 1/P_S$, $1 \leqslant i \leqslant P_S$, for the population

(1) For $1 \leqslant i \leqslant P_S$, generate $\mathbf{h}_l^{[i]}$ from $\mathbf{u}_i^{(k)}$, the candidates for the $l$th regressor, and orthogonalise them:

$$d_{j,l}^{[i]} = \frac{\mathbf{p}_j^{\mathrm{T}}\mathbf{h}_l^{[i]}}{\mathbf{p}_j^{\mathrm{T}}\mathbf{p}_j}, \quad 1 \leqslant j < l, \tag{34}$$

$$\mathbf{p}_l^{[i]} = \mathbf{h}_l^{[i]} - \sum_{j=1}^{l-1} d_{j,l}^{[i]}\mathbf{p}_j. \tag{35}$$

(2) For $1 \leqslant i \leqslant P_S$, calculate the loss of each population member

$$S_l^{[i]} = f(\mathbf{u}_i^{(k)}) = \frac{(\mathbf{p}_l^{[i]})^{\mathrm{T}}\mathbf{p}_l^{[i]}}{(\mathbf{y}^{\mathrm{T}}\mathbf{p}_l^{[i]})^2}. \tag{36}$$

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

*Step* 1: *Boosting*

(1) Find

$$\mathbf{u}_{\mathrm{best}}^{(k)} = \arg\min\{S_l^{[i]}, 1 \leqslant i \leqslant P_S\},$$

$$\mathbf{u}_{\mathrm{worst}}^{(k)} = \arg\max\{S_l^{[i]}, 1 \leqslant i \leqslant P_S\}.$$

(2) Normalise the loss function values

$$\bar{S}_l^{[i]} = \frac{S_l^{[i]}}{\sum_{j=1}^{P_S} S_l^{[j]}}, \quad 1 \leqslant i \leqslant P_S.$$

(3) Compute a weighting factor $\beta_t$ according to

$$\eta_t = \sum_{i=1}^{P_S} \delta_i(t-1)\bar{S}_l^{[i]}, \quad \beta_t = \frac{\eta_t}{1-\eta_t}.$$

(4) Update the distribution weightings for $1 \leqslant i \leqslant P_S$

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\bar{S}_l^{[i]}} & \text{for } \beta_t \leqslant 1, \\ \delta_i(t-1)\beta_t^{1-\bar{S}_l^{[i]}} & \text{for } \beta_t > 1, \end{cases}$$

and normalise them

$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{j=1}^{P_S}\delta_j(t)}, \quad 1 \leqslant i \leqslant P_S.$$

*Step* 2: *Parameter updating*

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

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

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

$$\mathbf{u}_{P_S+2} = \mathbf{u}_{\mathrm{best}}^{(k)} + (\mathbf{u}_{\mathrm{best}}^{(k)} - \mathbf{u}_{P_S+1}).$$

(3) Calculate $\mathbf{h}_l^{[P_S+1]}$ and $\mathbf{h}_l^{[P_S+2]}$ from $\mathbf{u}_{P_S+1}$ and $\mathbf{u}_{P_S+2}$, orthogonalise these two candidate model columns (as in (34) and (35)), and compute their loss function values (as in (36))
(4) Choose a better point (smaller loss function value) from $\mathbf{u}_{P_S+1}$ and $\mathbf{u}_{P_S+2}$ to replace $\mathbf{u}_{\text{worst}}^{(k)}$
If $\|\mathbf{u}_{P_S+1} - \mathbf{u}_{P_S+2}\| < \zeta_I$, exist **inner loop**

**End of inner loop**
The solution found is $\mathbf{u}_{\text{best}}^{(k)}$
**End of outer loop**

This yields the solution $\mathbf{u} = \mathbf{u}_{\text{best}}^{(N_G)}$ as the parameter vector (mean vector and diagonal covariance matrix) of the $l$th regressor, as well as the corresponding orthogonal model column $\mathbf{p}_l$.

The motivation and analysis of the RWBS algorithm as a general global optimiser are detailed in [12]. The appropriate values for the algorithmic parameters, $P_S$, $N_G$ and $\zeta_I$, 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. 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, $\zeta_I$, can be set to a relatively large value. This makes the search efficient, achieving convergence with a small number of the cost function evaluations. As an alternative to choose $\zeta_I$, one can simply set a maximum number of iterations $M_I$ for the inner-loop optimisation. The population size $P_S$ and the number of generations $N_G$ should be set to sufficiently large values so that the parameter space will be sampled sufficiently. The optimisation experiments reported in [12] suggested that the algorithmic parameters of the RWBS algorithm are not difficult to set.

It should be emphasised that $P_S$, $N_G$ and $\zeta_I$ (or $M_I$) are not the learning hyperparameters of the GSESVM algorithm. Rather they are the optimisation algorithmic parameters. The learning hyperparameters of the GSESVM algorithm are $C$ and $\varepsilon$. It is important to distinguish these two types of algorithmic parameters. Obviously, the optimisation algorithmic parameters need to be set appropriately but they are not as critical as the learning hyperparameters in the influence of the model generalisation capability. When one chooses a particular optimiser to solve the constrained quadratic programming (QP) of the SVM learning problem, for example, one also needs to assign some optimisation algorithmic parameters. These QP optimiser's algorithmic parameters are similar in nature to the algorithmic parameters of the RWBS optimiser, and they are not the learning hyperparameters of the SVM algorithm. The learning hyperparameters of the SVM algorithm are the kernel variance $\sigma^2$, $C$ and $\varepsilon$.

Table 1
Summary of the experimental results for the simulated example

| Algorithm | SVM | SESVM | GSESVM |
|---|---|---|---|
| Kernel type | Gaussian | Gaussian | Generalised Gaussian |
| Error band $\varepsilon$ | 0.2 | 0.3 | 0.2 |
| Regularisation $C$ | 0.5 | 0.3 | 1.0 |
| Model size | 172 | 16 | 9 |
| MSE over noisy training set | 0.9522 | 0.9697 | 0.9658 |
| MSE over noisy test set | 1.2572 | 1.1950 | 1.2285 |
| MSE over noise-free test set | 0.0740 | 0.0353 | 0.0344 |

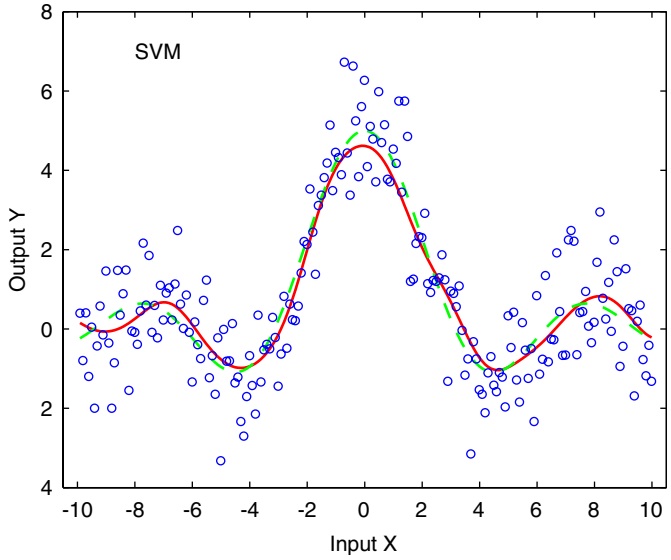



Fig. 1. The experiment result of the SVM method for the simulated example. The circles are the noisy training data, the dashed curve is the sinc function, and the solid curve is the kernel model with 172 support vectors. The kernel variance $\sigma^2 = 1$, regularisation parameter $C = 0.5$ and error band parameter $\varepsilon = 0.2$.

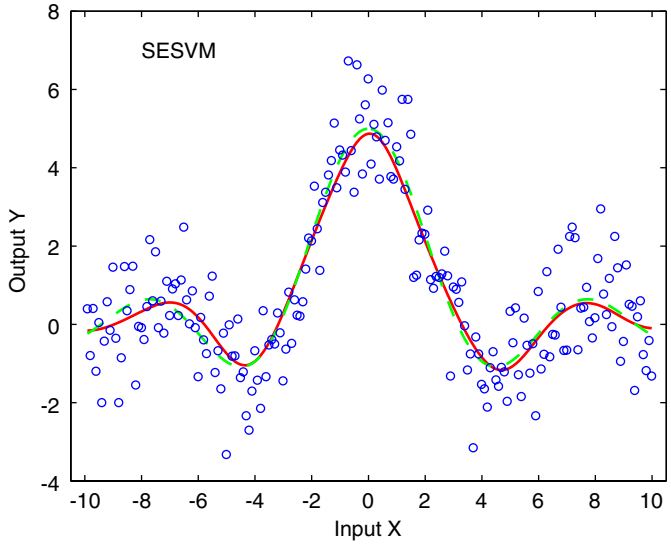



Fig. 2. The experiment result of the SESVM method for the simulated example. The circles are the noisy training data, the dashed curve is the sinc function, and the solid curve is the kernel model with 16 support vectors. The kernel variance $\sigma^2 = 1$, regularisation parameter $C = 0.3$ and error band parameter $\varepsilon = 0.3$.

## 4. Modelling experiments

A one-dimensional simulated example and three real data sets were used in our modelling experiments. For each example, three sets of results were obtained by the SVM, the SESVM and the GSESVM, respectively. The learning hyperparameters, $C$ and $\varepsilon$, were optimised using grid search optimisation based on cross validation for each algorithm. The single common kernel variance $\sigma^2$, required for the SVM and SESVM algorithms, was similarly determined. The optimisation algorithmic parameters of the RWBS, $P_S$, $N_G$ and $M_I$, were chosen empirically.

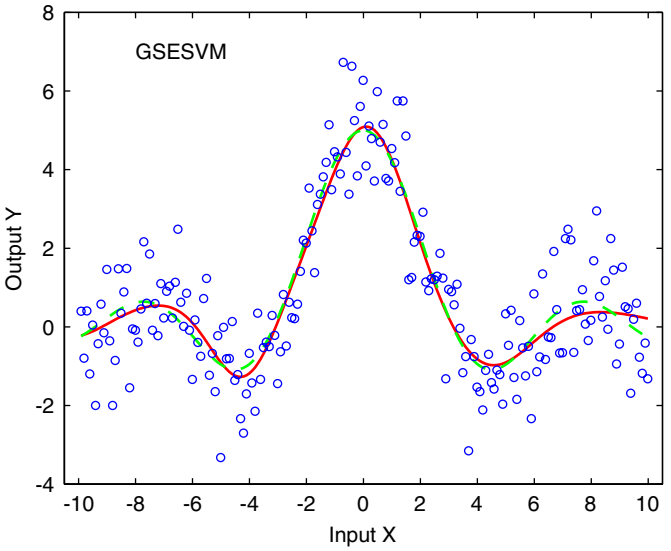



Fig. 3. The experiment result of the GSESVM method for the simulated example. The circles are the noisy training data, the dashed curve is the sinc function, and the solid curve is the generalised kernel model with nine support vectors. The regularisation parameter $C = 1.0$ and error band parameter $\varepsilon = 0.2$.

**Example 1.** Two hundred points of training data $\{x, y\}$ were generated from the scalar sinc function corrupted by an observation noise shown below

$$y = \frac{5 \sin x}{x} + \eta, \tag{37}$$

where the equally spaced input $x \in [-10, 10]$ and $\eta$ denotes the Gaussian white noise process with unit variance. A separate noisy test data containing 200 data samples was provided for model validation purpose. Two hundred points of noise-free data were also generated as the additional test data set. For the Gaussian kernel modelling, the common kernel variance was set to $\sigma^2 = 1$. This value was found empirically to be appropriate. The error band

Table 2
Summary of the experimental results for the engine data set

| Algorithm | SVM | SESVM | GSESVM |
|---|---|---|---|
| Kernel type | Gaussian | Gaussian | Generalised Gaussian |
| Error band $\varepsilon$ | 0.01 | 0.0107 | 0.01 |
| Regularisation $C$ | 14.0 | 1600.0 | 300.0 |
| Model size | 94 | 50 | 15 |
| MSE over training set | 0.0004388 | 0.0004548 | 0.0004586 |
| MSE over test set | 0.0004930 | 0.0004991 | 0.0004894 |

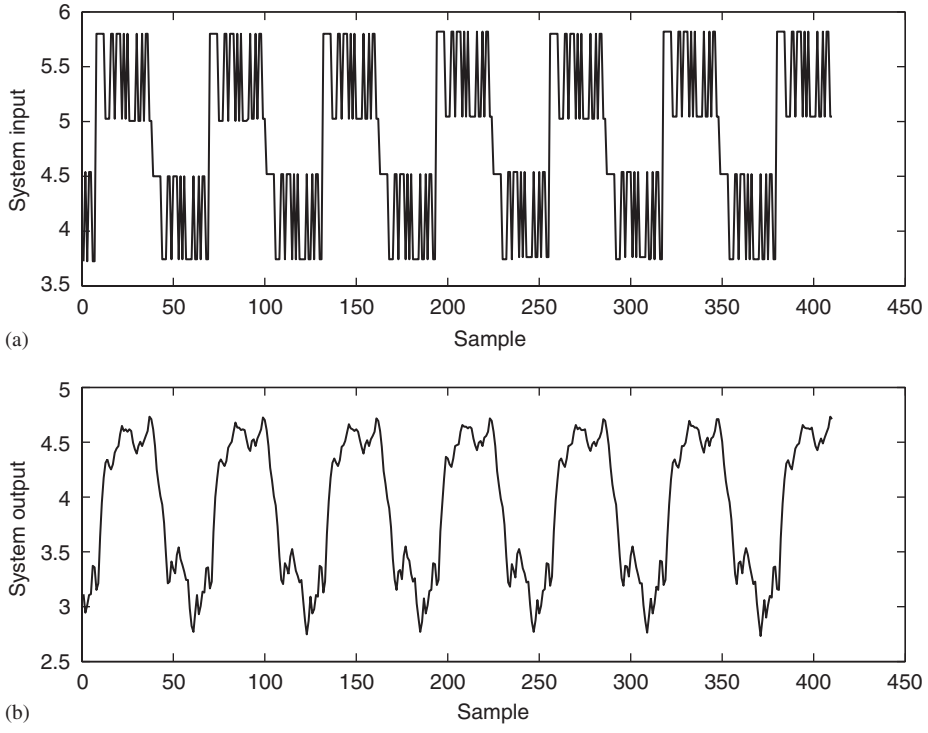



Fig. 4. The engine data set: (a) system input $v(t)$ and (b) system output $y(t)$.

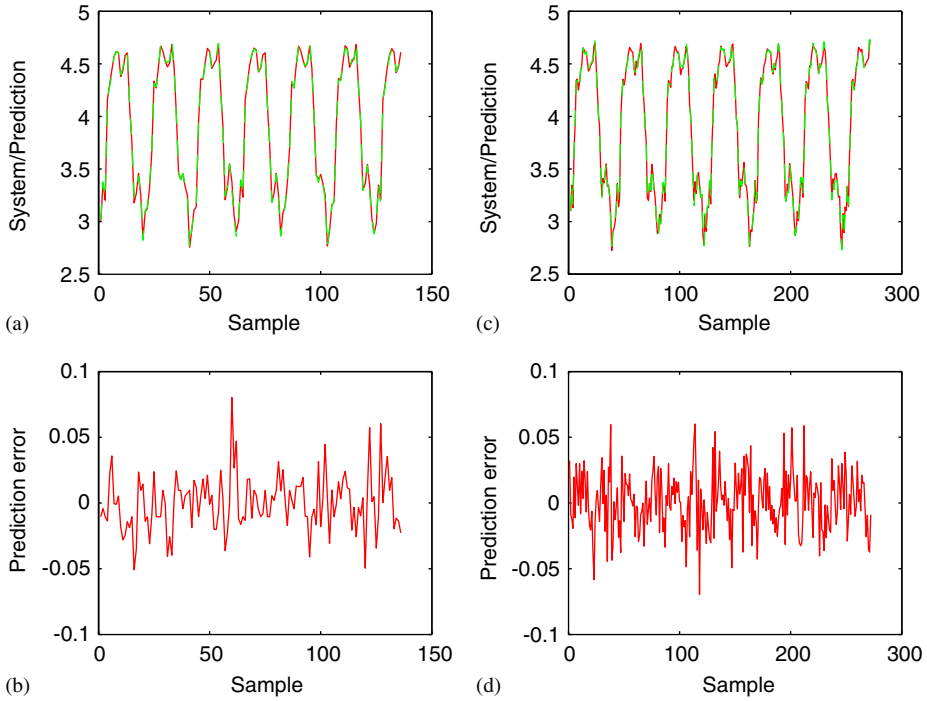



Fig. 5. The experiment result of the SVM method for the engine data set: (a) model prediction (green or light curve) superimposed on system output (red or dark curve); (b) model prediction error over the training set; (c) model prediction (green or light curve) superimposed on system output (red or dark curve); and (d) prediction error over the test set. The regularisation parameter $C = 14.0$, error band parameter $\varepsilon = 0.01$ and the model contains 94 kernels.

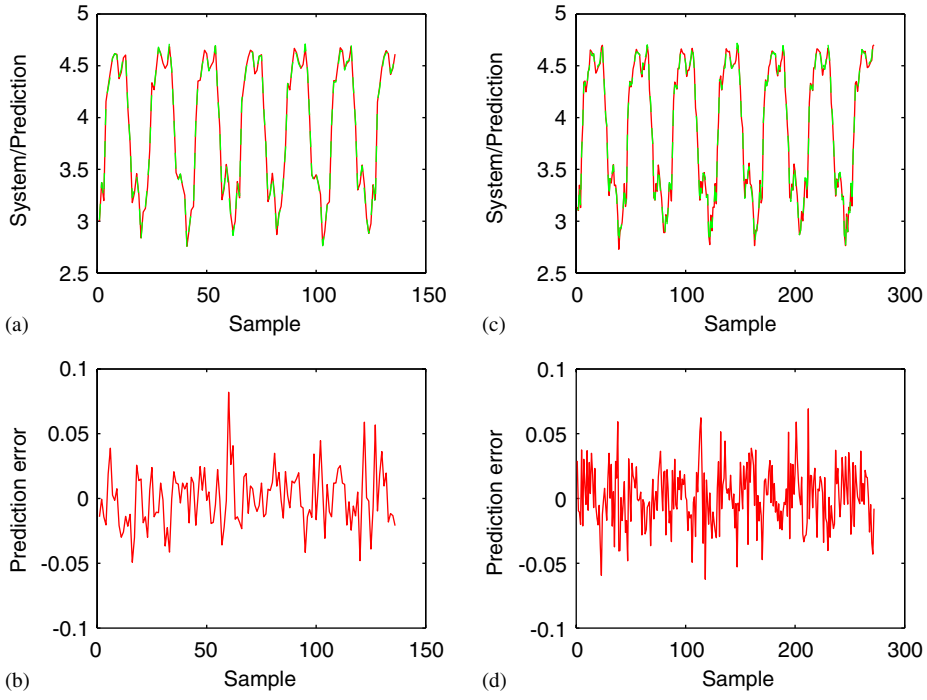



Fig. 6. The experiment result of the SESVM method for the engine data set: (a) model prediction (green or light curve) superimposed on system output (red or dark curve); (b) model prediction error over the training set; (c) model prediction (green or light curve) superimposed on system output (red or dark curve); and (d) prediction error over the test set. The regularisation parameter $C = 1600.0$, error band parameter $\varepsilon = 0.0107$ and the model contains 50 kernels.

parameter $\varepsilon$ and regularisation parameter $C$ for each algorithm were determined by grid search to minimise the MSE over the noisy test data set. The algorithmic parameters of the RWBS were chosen empirically. The experimental results obtained by the SVM, SESVM and GSESVM methods are summarised in Table 1. Judging by

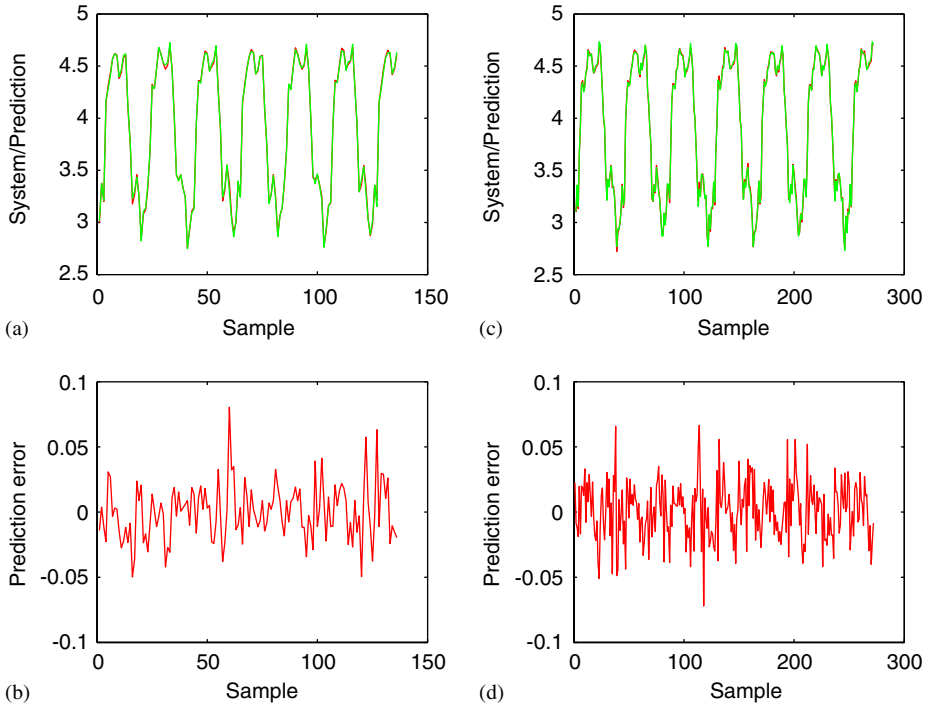



Fig. 7. The experiment result of the GSESVM method for the engine data set: (a) model prediction (green or light curve) superimposed on system output (red or dark curve); (b) model prediction error over the training set; (c) model prediction (green or light curve) superimposed on system output (red or dark curve); and (d) prediction error over the test set. The regularisation parameter $C = 300.0$, error band parameter $\varepsilon = 0.01$ and the model contains 15 kernels.

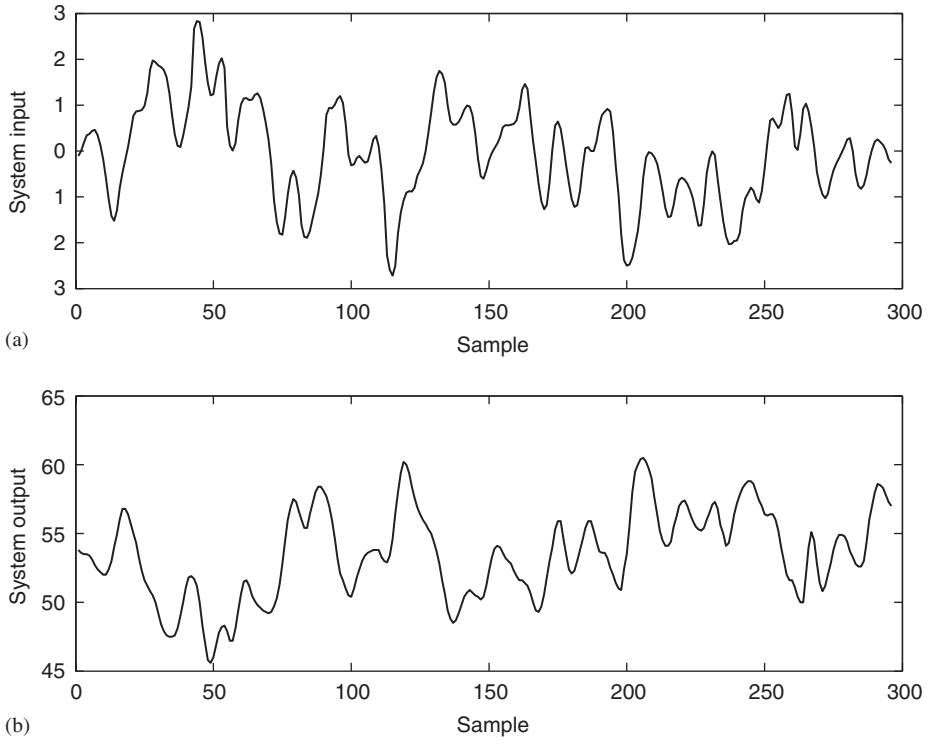



Fig. 8. The gas furnace data set: (a) system input $v(t)$ and (b) system output $y(t)$.

their MSE values over the noisy test data set, the three algorithms had similarly good generalisation capability but the model produced by the GSESVM method was the sparsest containing only 9 kernels. The model maps derived by the three methods are depicted in Figs. 1–3, respectively, in comparison with the underlying sinc function.

**Example 2.** 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. The data set, depicted in Fig. 4, contained 410 samples. The study [3] has shown that this data set can be modelled as

$$y_i = F_S(\mathbf{x}_i) + e_i, \tag{38}$$

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. It is often claimed that the SVM method is capable of constructing sparse models with excellent generalisation performance with a small training set. We constructed the training set by using the data pairs $(\mathbf{x}_i, y_i)$ for $i = 3, 6, 9, 12, \ldots$ and putting rest of the data pairs into the test set. Thus, the training set contained $N = 136$ points, while the test set had 272 samples. The values of the single common kernel variance $\sigma^2$ for the two Gaussian kernel modelling cases were determined using a grid search, and the appropriate values were found to be 1.69 for the SVM algorithm and 2.60 for the SESVM algorithm, respectively.

Again, the error band parameter $\varepsilon$ and regularisation parameter $C$ for each algorithm were found by grid search to minimise the MSE over the test data set. The algorithmic parameters of the RWBS were determined empirically. Table 2 summarises the experimental results obtained by the SVM, SESVM and GSESVM algorithms. It can be seen that the GSESVM method produced the best result, in terms of model generalisation capability and model size. Fig. 5 depicts the model prediction $\hat{y}_i$ and the prediction error $\hat{e}_i = y_i - \hat{y}_i$ obtained by the SVM model over both the training and test sets. Similarly, the modelling results of the SESVM and GSESVM algorithms are shown in Figs. 6 and 7, respectively.

Table 3
Summary of the experimental results for the gas furnace data set

| Algorithm | SVM | SESVM | GSESVM |
|---|---|---|---|
| Kernel type | Gaussian | Gaussian | Generalised Gaussian |
| Error band $\varepsilon$ | 0.15 | 0.05 | 0.05 |
| Regularisation $C$ | 50.0 | 880.0 | 600.0 |
| Model size | 79 | 47 | 5 |
| MSE over training set | 0.0316 | 0.0801 | 0.0603 |
| MSE over test set | 0.1070 | 0.0871 | 0.0760 |

**Example 3.** This example constructed a model for the gas furnace data set (Series J in [4]). The data set contained 296 pairs of input–output points, where the input $v(t)$ was the coded input gas feed rate and the output $y(t)$ represented the $CO_2$ concentration from the gas furnace. Fig. 8 depicts this data set. Let the desired output be $y_i = y(i)$ for the model input vector

$$\mathbf{x}_i = [y(i-1)\ y(i-2)\ y(i-3)\ v(i-1)\ v(i-2)\ v(i-3)]^T \tag{39}$$

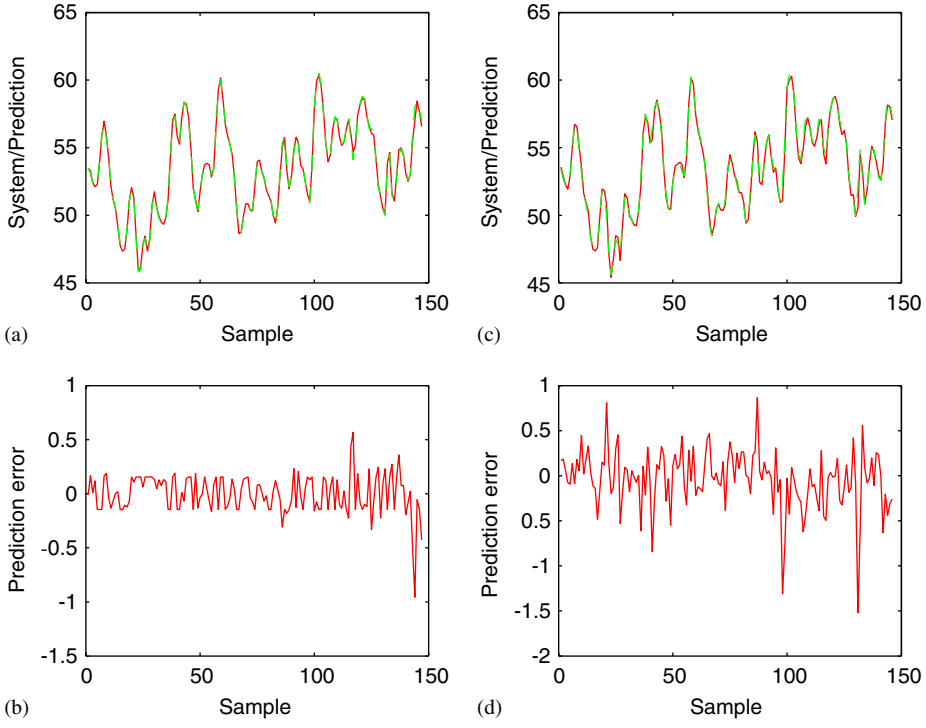



Fig. 9. The experiment result of the SVM method for the gas furnace data set: (a) model prediction (green or light curve) superimposed on system output (red or dark curve); (b) model prediction error over the training set; (c) model prediction (green or light curve) superimposed on system output (red or dark curve); and (d) prediction error over the test set. The regularisation parameter $C = 50.0$, error band parameter $\varepsilon = 0.15$ and the model contains 79 kernels.

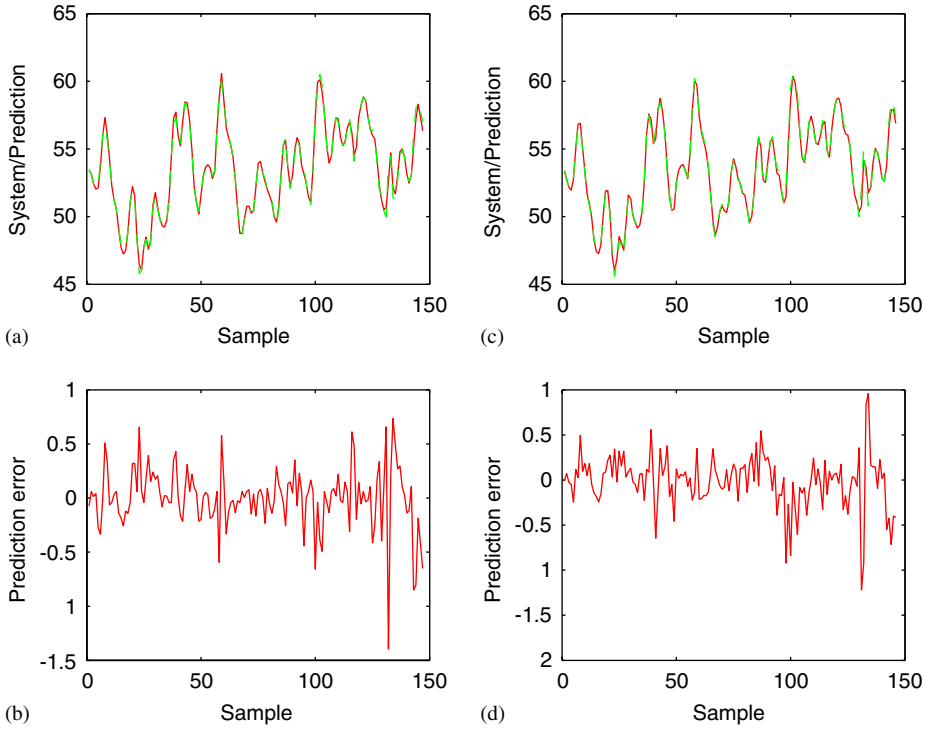



Fig. 10. The experiment result of the SESVM method for the gas furnace data set: (a) model prediction (green or light curve) superimposed on system output (red or dark curve); (b) model prediction error over the training set; (c) model prediction (green or light curve) superimposed on system output (red or dark curve); and (d) prediction error over the test set. The regularisation parameter $C = 880.0$, error band parameter $\varepsilon = 0.05$ and the model contains 47 kernels.

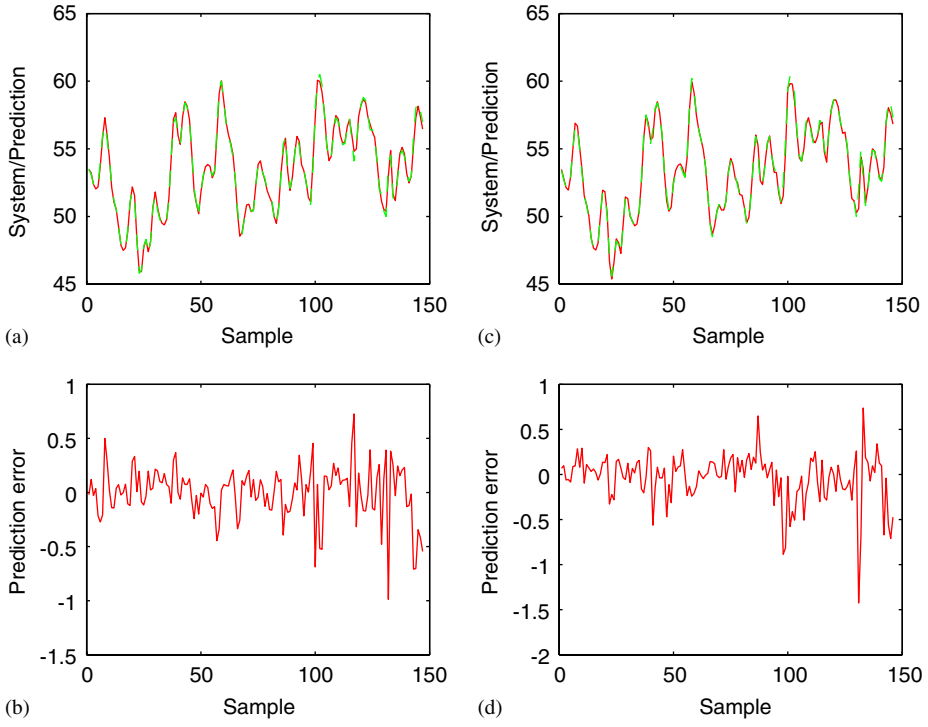



Fig. 11. The experiment result of the GSESVM method for the gas furnace data set: (a) model prediction (green or light curve) superimposed on system output (red or dark curve); (b) model prediction error over the training set; (c) model prediction (green or light curve) superimposed on system output (red or dark curve); and (d) prediction error over the test set. The regularisation parameter $C = 600.0$, error band parameter $\varepsilon = 0.05$ and the model contains 5 kernels.

Table 4
Summary of the experimental results for the Boston housing data set

| Algorithm | SVM | SESVM | GSESVM |
|---|---|---|---|
| Kernel type | Gaussian | Gaussian | Generalised Gaussian |
| Error band $\varepsilon$ | 2.0 | 2.0 | 10.0 |
| Regularisation $C$ | 800.0 | 600000.0 | 800.0 |
| Model size | 250 | 48 | 3 |
| MSE over training set | 6.0166 | 22.9726 | 18.9652 |
| MSE over test set | 22.9816 | 19.1912 | 15.6655 |

for $4 \leqslant i \leqslant 296$. The even points of $(y_i, \mathbf{x}_i)$ were used for the training set while the odd points were selected as the test set. For the Gaussian kernel modelling, an appropriate value for the single common kernel variance was found empirically to be $\sigma^2 = 20$. The error band parameter $\varepsilon$ and regularisation parameter $C$ for each algorithm were determined by grid search to minimise the MSE over the test data set. The algorithmic parameters of the RWBS were set empirically. Table 3 gives the experimental results obtained by the SVM, SESVM and GSESVM algorithms. The modelling results are also plotted in Figs. 9–11, respectively, for the three algorithms. It is clear that for this example the best result was obtained by the GSESVM method.

**Example 4.** This is a popular regression benchmark data set, Boston Housing, available at the UCI repository [21]. The data set comprises 506 data points with 14 variables. The task was to predict the median house value from the remaining 13 attributes. The first 456 data points from the data set were used for training and the remaining 50 data points were used to form the test set. As usual, the appropriate value for the single common kernel variance, required by the Gaussian kernel modelling, was determined via cross validation, yielding $\sigma^2 = 2116.0$ and $\sigma^2 = 2025.0$ for the SVM and SESVM, respectively. The error band parameter $\varepsilon$ and regularisation parameter $C$ for each algorithm were chosen by grid search via cross validation. The algorithmic parameters of the RWBS were set empirically. Table 4 summarises the modelling results for this data set. The results of Table 4 again show that the GSESVM method produced the best model, in terms of model generalisation performance and model size.

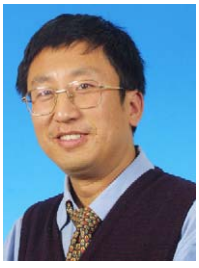
## 5. Conclusions

In this paper, we have first considered an alternative SVM formulation, referred to as the ESVM method, which does not assume the reproducing kernel Hilbert space and is capable of applying to non-Mercer kernels. Secondly, we have proposed a sparse kernel model construction algorithm, called the SESVM. 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 ESVM formulation. Thirdly, which is a major contribution of our work, we have developed the generalised kernel modelling in which each kernel regressor has its tunable centre vector and diagonal covariance matrix. An orthogonal forward selection procedure has been proposed to construct a sparse generalised kernel model representation. At each model construction stage, a kernel regressor is optimised using a global optimisation search algorithm. Again the corresponding model weights are then calculated using the ESVM formulation, and this novel generalised kernel construction algorithm has been referred to as the GSESVM method. Our modelling experimental results have clearly demonstrated that both the SESVM and GSESVM methods compare favourably with the standard SVM formulation in terms of producing sparse models that generalise well. The GSESVM method has been shown to be particularly effective in constructing very sparse models with excellent generalisation capability, and we believe that it offers a state-of-the-art technique for regression modelling.

## References

[1] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control AC-19 (1974) 716–723.
[2] N. Aronszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. 68 (1950) 337–404.
[3] S.A. Billings, S. Chen, R.J. Backhouse, The identification of linear and non-linear models of a turbocharged automotive diesel engine, Mech. Syst. Signal Process. 3 (2) (1989) 123–142.
[4] G.E.P. Box, G.M. Jenkins, Time Series Analysis, Forecasting and Control, Holden Day, San Francisco, CA, 1976.
[5] L. Breiman, Prediction games and arcing algorithms, Neural Comput. 11 (7) (1999) 1493–1518.
[6] O. Chapelle, V. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1–3) (2002) 131–159.
[7] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their application to non-linear system identification, Int. J. Control 50 (5) (1989) 1873–1896.
[8] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (2) (1991) 302–309.
[9] S. Chen, X. Hong, C.J. Harris, Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design, IEEE Trans. Autom. Control 48 (6) (2003) 1029–1036.
[10] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using orthogonal forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. Part B 34 (2) (2004) 898–911.
[11] S. Chen, B.L. Luk, Adaptive simulated annealing for optimization in signal processing applications, Signal Process. 79 (1) (1999) 117–128.
[12] S. Chen, X.X. Wang, C.J. Harris, Experiments with repeating weighted boosting search for optimization in signal processing applications, IEEE Trans. Syst. Man Cybern. Part B 35 (4) (2005) 682–693.
[13] S. Chen, Y. Wu, B.L. Luk, Combined genetic algorithm optimisation and regularised orthogonal least squares learning for radial basis function networks, IEEE Trans. Neural Networks 10 (5) (1999) 1239–1243.
[14] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines: and Other Kernel-Based Learning Methods, Cambridge University Press, Cambridge, UK, 2000.
[15] K. Duan, S.S. Keerthi, A.N. Poo, Evaluation of simple performance measures for tuning SVM hyperparameters, Neurocomputing 51 (2003) 41–59.

[16] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[17] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[18] S. Gunn, Support vector machines for classification and regression, Technical Report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, UK, May 1998.
[19] X. Hong, C.J. Harris, Nonlinear model structure design and construction using orthogonal least squares and D-optimality design, IEEE Trans. Neural Networks 13 (5) (2002) 1245–1250.
[20] X. Hong, C.J. Harris, S. Chen, P.M. Sharkey, Robust nonlinear model identification methods using forward regression, IEEE Trans. Syst. Man Cybern. Part A 33 (4) (2003) 514–523.
[21] ⟨http://www.ics.uci.edu/~mlearn/MLRepository.html⟩.
[22] L. Ingber, Simulated annealing: practice versus theory, Math. Comput. Modeling 18 (11) (1993) 29–57.
[23] I.J. Leontaritis, S.A. Billings, Model selection and validation methods for non-linear systems, Int. J. Control 45 (1) (1987) 311–341.
[24] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms: concepts and Design, Springer, London, 1998.
[25] R. Meir, G. Rätsch, An introduction to boosting and leveraging, in: S. Mendelson, A. Smola (Eds.), Advanced Lectures in Machine Learning, Springer, Berlin, 2003, pp. 119–184.
[26] C.S. Ong, A.J. Smola, R.C. Williamson, Hyperkernels, Neural Information Processing Systems, vol. 15, MIT Press, Cambridge, MA, 2002.
[27] B. Schölkopf, A.J. Smola, Learning with Kernels: support vector machines, regularization, optimization, and beyond, MIT Press, Cambridge, MA, 2002.
[28] B. Schölkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neural Comput. 12 (5) (2000) 1207–1245.
[29] B. Schölkopf, K.K. Sung, C.J.C. Burges, F. Girosi, P. Niyogi, T. Poggio, V. Vapnik, Comparing support vector machines with Gaussian kernels to radial basis function classifiers, IEEE Trans. Signal Process. 45 (11) (1997) 2758–2765.
[30] R.E. Schapire, The strength of weak learnability, Mach. Learn. 5 (2) (1990) 197–227.
[31] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.



**Xunxian Wang** received his Ph.D. degree in the control theory and application field from Tsinghua University, Beijing, China, in July 1999.

From August 1999 to August 2001, he was a postdoctoral researcher in the State Key Laboratory of Intelligent Technology and Systems, Beijing, China. From September 2001 to December 2004, he was a research fellow at the University of Portsmouth, Portsmouth, UK. From January 2005, he has been a research fellow at Neural Computing Research Group, Aston University, Birmingham, UK. Dr. Wang's main interests are in machine learning and neural networks, control theory and systems as well as robotics.

**Sheng Chen** received his Ph.D. degree 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. He previously held research and academic appointments at the University of Sheffield, Sheffield, the University of Edinburgh, Edinburgh, and the University of Portsmouth, Portsmouth, all in UK. Professor Chen's research works include wireless communications, machine learning and neural networks, finite-precision digital controller design, and evolutionary computation methods. He has published over 260 research papers.

In the database of the world's most highly cited researchers, compiled by Institute for Scientific Information (ISI) of the USA, Dr. Chen is on the list of the highly cited researchers in the engineering category.

**David Lowe** has held the Chair of Neural Computing at Aston University, UK, since 1994. He is a co-inventor of the Radial Basis Function neural network architecture, and the NeuroScale model for data visualisation. 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 nonlinear methods for digital steganography.

**Chris Harris** received his Ph.D. degree from the University of Southampton, Southampton, UK.

He previously held appointments at the University of Hull, Hull, the UMIST, Manchester, the University of Oxford, Oxford, and the University of Cranfield, Cranfield, all in UK, as well as being employed by the UK Ministry of Defence. He returned to the University of Southampton as the Lucas Professor of Aerospace Systems Engineering in 1987 to establish the Advanced Systems Research Group and, more recently, Image, Speech and Intelligent Systems Group. His research interests lie in the general area of intelligent and adaptive systems theory and its application to intelligent autonomous systems such as autonomous vehicles, management infrastructures such as command and control, intelligent control, and estimation of dynamic processes, multi-sensor data fusion, and systems integration. He has authored and co-authored 12 research books and over 400 research papers, and he is the associate editor of numerous international journals.

Dr. Harris was elected to the Royal Academy of Engineering in 1996, was awarded the IEE Senior Achievement medal in 1998 for his work in autonomous systems, and the highest international award in IEE, the IEE Faraday medal, in 2001 for his work in intelligent control and neurofuzzy systems.