Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neurocomputing 62 (2004) 441–457
NEUROCOMPUTING
www.elsevier.com/locate/neucom

An approach for constructing parsimonious generalized Gaussian kernel regression models

X.X. Wang[a], S. Chen[b,*], D.J. Brown[a]

[a]*Intelligent Systems and Diagnostics Group, Department of Electronic and Computer Engineering, University of Portsmouth, Anglesea Building, Anglesea Road, Portsmouth PO1 3DJ, UK*
[b]*School of Electronics and Computer Science, University of Southampton, Highfield, Southampton SO17 1BJ, UK*

Received 25 March 2004

Abstract

The paper proposes a novel construction algorithm for generalized Gaussian kernel regression models. Each kernel regressor in the generalized Gaussian kernel regression model has an individual diagonal covariance matrix, which is determined by maximizing the correlation between the training data and the regressor using a repeated guided random search based on boosting optimization. The standard orthogonal least squares algorithm is then used to select a sparse generalized kernel regression model from the resulting full regression matrix. Experimental results involving two real data sets demonstrate the effectiveness of the proposed regression modeling approach.
© 2004 Elsevier B.V. All rights reserved.

Keywords: Neural networks; Regression; Orthogonal least squares; Correlation; Boosting

1. Introduction

A fundamental principle in practical nonlinear data modeling is the parsimonious principle of ensuring the smallest possible model that explains the training data.

*Corresponding author.
E-mail addresses: xunxian.wang@port.ac.uk (X.X. Wang), sqc@ecs.soton.ac.uk (S. Chen), david.j.brown@port.ac.uk (D.J. Brown).

0925-2312/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2004.06.003

Forward selection using the orthogonal least squares (OLS) algorithm [3–6,8] is a simple and efficient construction method that is capable of producing parsimonious linear-in-the-weights nonlinear models with excellent generalization performance. Alternatively, the state-of-art sparse kernel modeling techniques, such as the support vector machine and relevant vector machine [19,21–23], have been gaining popularity in data modeling applications. These existing sparse regression modeling techniques typically place the kernel centers or mean vectors at the training input data and use a fixed common kernel variance for all the regressor kernels. The value of this common kernel variance obviously has a critical influence on the sparsity and generalization capability of the resulting model, and it has to be determined via some sort of cross validation. For example, in [8] a genetic algorithm is applied to determine the appropriate common kernel variance through optimizing the model generalization performance.

In this paper, we consider a generalized Gaussian kernel model, in which each kernel regressor has an individually tuned diagonal covariance matrix. Such a generalized kernel regression model has the potential of improving modeling capability and producing sparser final models, compared with the standard approach of single fixed common variance. The difficult issue is then how to determine these kernel covariance matrices. Since the correlation function between a kernel regressor and the training data defines the "similarity" between the regressor and the training data, it can be used to "shape" the regressor by adjusting the associated kernel covariance matrix in order to maximize the absolute value of this correlation function. A weighted optimization algorithm, which has its root from boosting [9,16,18], is considered to perform the associated optimization task. This weighted optimization algorithm is a guided random search method and the solution obtained may depend on the initial choice of population. To provide a robust optimization and guarantee stable solutions regardless of the initial choice of population, the algorithm is augmented into a repeated weighted optimization method.

The determination of kernel covariance matrices essentially provides the full bank of regressors or the full regression matrix, and this allows the application of the standard OLS algorithm [3,4] to select a parsimonious subset model. The outline of the paper is as follows. Section 2 gives the generalized Gaussian kernel regression model to be considered. Section 3 derives the correlation criterion to be used for determining the kernel covariance matrices and presents a repeated boosting search optimization algorithm for performing the corresponding optimization tasks. Section 4 briefly summarizes the standard OLS algorithm used to select a sparse kernel regression model, while Section 5 describes our modeling experiments. Finally, Section 6 offers our conclusions.

2. Generalized Gaussian kernel regression model

Consider a general discrete stochastic nonlinear system represented by

$$y_k = f_s(y_{k-1}, \ldots, y_{k-n_y}, u_{k-1}, \ldots, u_{k-n_u}; \boldsymbol{\theta}) + \varepsilon_k = f_s(\mathbf{x}_k; \boldsymbol{\theta}) + \varepsilon_k \tag{1}$$

where u_k and y_k are the system input and output variables, respectively, n_u and n_y are positive integers representing the known lags in u_k and y_k, respectively, the observation noise ε_k is uncorrelated with zero mean, $\mathbf{x}_k = [y_{k-1} \cdots y_{k-n_y}\ u_{k-1} \cdots u_{k-n_u}]^{\mathrm{T}}$ denotes the system input vector with a known dimension $n = n_y + n_u$, $f_s(\bullet)$ is a priori unknown system mapping, and $\boldsymbol{\theta}$ is an unknown parameter vector associated with the appropriate, but yet to be determined, model structure. The system model (1) is to be identified from an N-sample system observational data set $D_N = \{\mathbf{x}_k, y_k\}_{k=1}^N$, using some suitable functions which can approximate $f_s(\bullet)$ with arbitrary accuracy.

We will model the unknown dynamical process (1) by using the following generalized Gaussian kernel regression model

$$y_k = \hat{y}_k + \varepsilon_k = \sum_{i=1}^{N} \theta_i g_i(\mathbf{x}_k) + \varepsilon_k \tag{2}$$

where $\hat{y}_k$ denotes the model output given the input $\mathbf{x}_k, \theta_i$ are the model weight parameters, and $g_i(\bullet)$ are the kernel regressors. We allow the regressor function to be chosen as the general Gaussian function $g_i(\mathbf{x}) = G(\mathbf{x}; \mathbf{x}_i, \Sigma_i)$ with

$$G(\mathbf{x}; \mathbf{x}_i, \Sigma_i) = \exp\left(-\tfrac{1}{2}(\mathbf{x} - \mathbf{x}_i)^{\mathrm{T}} \Sigma_i^{-1} (\mathbf{x} - \mathbf{x}_i)\right) \tag{3}$$

where the kernel covariance matrix takes the form of $\Sigma_i = \mathrm{diag}\{\sigma_{i,1}^2, \ldots, \sigma_{i,n}^2\}$. As is in a standard kernel regression model, the kernel mean vectors are placed at the training input data points. If all the diagonal covariance matrices are set to the identical form of $\mathrm{diag}\{\sigma^2, \ldots, \sigma^2\}$, we arrive at the standard Gaussian kernel model.

The kernel model (2) over the training set D_N can be written in the matrix form as

$$\mathbf{y} = \mathbf{G}\boldsymbol{\theta} + \boldsymbol{\varepsilon} \tag{4}$$

by defining

$$\mathbf{y} = [y_1\ y_2 \cdots y_N]^{\mathrm{T}} \tag{5}$$

$$\boldsymbol{\theta} = [\theta_1\ \theta_2 \cdots \theta_N]^{\mathrm{T}} \tag{6}$$

$$\boldsymbol{\varepsilon} = [\varepsilon_1\ \varepsilon_2 \cdots \varepsilon_N]^{\mathrm{T}} \tag{7}$$

$$\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2 \cdots \mathbf{g}_N] \tag{8}$$

$$\mathbf{g}_i = [g_i(\mathbf{x}_1)\ g_i(\mathbf{x}_2) \cdots g_i(\mathbf{x}_N)]^{\mathrm{T}},\ 1 \leqslant i \leqslant N. \tag{9}$$

The objective of sparse modeling is to construct a subset model consisting of N_s ($\ll N$) significant regressors only from the full set of regressors defined in (9).

3. Determination of the full regression matrix

To specify the pool of regressors or the full regression matrix $\mathbf{G}$, one needs to determine all the associated diagonal covariance matrices $\Sigma_i, 1 \leqslant i \leqslant N$. Let us start

the discussion by defining the least squares cost or mean square error (MSE) associated with an m-term model as

$$S_m = \frac{1}{N}\sum_{k=1}^{N}(y_k - \hat{y}_k)^2 \tag{10}$$

where for the notational simplicity the same notation $\hat{y}_k$ is also used for representing the m-term model output. Obviously $S_0 = \mathbf{y}^{\mathrm{T}}\mathbf{y} = \|\mathbf{y}\|^2$.

3.1. Correlation criterion

The correlation between a regressor $\mathbf{g}_i$ and the training data is defined by

$$C(\Sigma_i) = \frac{\mathbf{y}^{\mathrm{T}}\mathbf{g}_i}{\sqrt{\mathbf{y}^{\mathrm{T}}\mathbf{y}}\sqrt{\mathbf{g}_i^{\mathrm{T}}\mathbf{g}_i}} \tag{11}$$

This correlation is a function of the regressor's kernel covariance matrix. We propose to use this correlation function as the optimization criterion to determine the regressor's kernel covariance matrix. Specifically, we should choose Σ_i so that $|C(\Sigma_i)|$ is maximized. Why this is a good strategy to specify the pool of regressors can easily be explained. Assuming that $\mathbf{g}_i$ is selected to form a one-term model, the associated reduction in the MSE value can be shown to be

$$\Delta S = S_0 - S_1 = \frac{(\mathbf{y}^{\mathrm{T}}\mathbf{g}_i)^2}{\mathbf{g}_i^{\mathrm{T}}\mathbf{g}_i} \tag{12}$$

which can be rewritten as

$$\Delta S = (\mathbf{y}^{\mathrm{T}}\mathbf{y})\,\frac{(\mathbf{y}^{\mathrm{T}}\mathbf{g}_i)^2}{(\mathbf{y}^{\mathrm{T}}\mathbf{y})(\mathbf{g}_i^{\mathrm{T}}\mathbf{g}_i)} = \|\mathbf{y}\|^2|C(\Sigma_i)|^2 \tag{13}$$

Since $\|\mathbf{y}\|^2$ is a constant, maximizing $|C(\Sigma_i)|$ leads to a maximum reduction in the MSE value.

Having chosen the optimization criterion, we now turn our attention to optimization algorithm. We propose a repeated guided random search method to perform the associated optimization tasks. This algorithm adopts ideas from boosting [9,16,18].

3.2. Weighted optimization algorithm

The task of maximizing $|C(\Sigma_i)|$ with respect to Σ_i can be carried out by various optimization methods. For example, the global optimization methods, such as the genetic algorithm [13,15] and adaptive simulated annealing [7,14], can be used. A global optimization method however is generally computationally very costly and may be overkill, since in this application we only seek to tune a kernel's diagonal covariance matrix. Let us consider the following simple search method to perform this optimization. Given p points of $\Sigma, \Sigma^{(1)}, \ldots, \Sigma^{(p)}$, let $\Sigma^{\mathrm{best}} = \arg\max\{|C(\Sigma^{(i)})|, 1 \leqslant i \leqslant p\}$ and $\Sigma^{\mathrm{worst}} = \arg\min\{|C(\Sigma^{(i)})|, 1 \leqslant i \leqslant p\}$. A $(p+1)$th point

is generated by a weighted combination of $\Sigma^{(i)}, 1 \leqslant i \leqslant p$. Because this weighted combination is a convex combination, the point $\Sigma^{(p+1)}$ is always within the convex hull defined by the p values. A $(p+2)$th point is then generated as the mirror image of $\Sigma^{(p+1)}$, with respect to Σ^{best}, along the direction defined by $\Sigma^{\text{best}} - \Sigma^{(p+1)}$. The best of $\Sigma^{(p+1)}$ and $\Sigma^{(p+2)}$ then replaces Σ^{worst}. The process is repeated until it converges.

A simple illustration is depicted in Fig. 1 for a one-dimensional case, where there are $p=3$ points, $\Sigma^{(1)}$, $\Sigma^{(2)}$ and $\Sigma^{(3)}$, and $\Sigma^{\text{best}} = \Sigma^{(2)}$ and $\Sigma^{\text{worst}} = \Sigma^{(3)}$. The 4th value $\Sigma^{(4)}$ is a weighted combination of $\Sigma^{(1)}$, $\Sigma^{(2)}$ and $\Sigma^{(3)}$, and $\Sigma^{(5)}$ is the mirror image of $\Sigma^{(4)}$ with respect to $\Sigma^{(2)}$. As $\Sigma^{(4)}$ is better than $\Sigma^{(5)}$ in this case, it replaces $\Sigma^{(3)}$. Clearly, how the weighted combination is performed is critical. The weightings for $\Sigma^{(i)}, 1 \leqslant i \leqslant p$, should reflect the "goodness" of $\Sigma^{(i)}$, and the process should be capable of self-learning or adapting these weightings. This is exactly the basic idea of boosting [9,16,18]. Specifically, by combining the AdaBoost algorithm of [9] with the above-mentioned simple search strategy, we arrive at the weighted optimization algorithm. Given the training data D_N and for fitting the lth regressor's covariance matrix, the algorithm is summarized as follows.

Initialization: Set iteration index $t=0$, give the p randomly chosen initial values for $\Sigma_l, \Sigma_l^{(1)}(t), \Sigma_l^{(2)}(t), \ldots, \Sigma_l^{(p)}(t)$, with the associated weightings $\delta_i^{(t)} = 1/p$ for $1 \leqslant i \leqslant p$, and specify a small positive value ξ for terminating the search.

Step 1: *Boosting*

1. Calculate the loss of each point in the population, namely

$$\text{cost}_i = 1 - |C(\Sigma_l^{(i)}(t))|, \ 1 \leqslant i \leqslant p$$

2. Find

$$\Sigma_l^{\text{best}}(t) = \arg\min\{\text{cost}_i, \ 1 \leqslant i \leqslant p\}$$

and

$$\Sigma_l^{\text{worst}}(t) = \arg\max\{\text{cost}_i, \ 1 \leqslant i \leqslant p\}$$

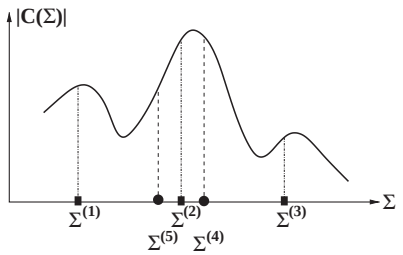


Fig. 1. Illustration of a simple weighted search optimization process.

3. Normalize the loss

$$\text{loss}_i = \frac{\text{cost}_i}{\sum_{j=1}^{p} \text{cost}_j}, \ 1 \leqslant i \leqslant p$$

4. Compute a weighting factor β_t according to

$$\eta_t = \sum_{i=1}^{p} \delta_i^{(t)} \text{loss}_i, \ \beta_t = \frac{\eta_t}{1-\eta_t}$$

5. Update the weighting vector

$$\delta_i^{(t+1)} = \begin{cases} \delta_i^{(t)} \beta_t^{\text{loss}_i} & \text{for } \beta_t \leqslant 1, \\ \delta_i^{(t)} \beta_t^{1-\text{loss}_i} & \text{for } \beta_t > 1, \end{cases} \quad 1 \leqslant i \leqslant p$$

6. Normalize the weighting vector

$$\delta_i^{(t+1)} = \frac{\delta_i^{(t+1)}}{\sum_{j=1}^{p} \delta_j^{(t+1)}}, \ 1 \leqslant i \leqslant p$$

Step 2: *Parameter updating*

1. Construct the $(p+1)$th point using the formula

$$\Sigma_l^{(p+1)}(t) = \sum_{i=1}^{p} \delta_i^{(t+1)} \Sigma_l^{(i)}(t)$$

2. Construct the $(p+2)$th point using the formula

$$\Sigma_l^{(p+2)}(t) = \Sigma_l^{\text{best}}(t) + (\Sigma_l^{\text{best}}(t) - \Sigma_l^{(p+1)}(t))$$

3. Choose a better point (smaller loss value) from $\Sigma_l^{(p+1)}(t)$ and $\Sigma_l^{(p+2)}(t)$ to replace $\Sigma_l^{\text{worst}}(t)$, which will inherit the weighting δ value from $\Sigma_l^{\text{worst}}(t)$.[1]

Set $t = t + 1$ and repeat from Step 1 until

$$\|\Sigma_l^{(p+1)}(t) - \Sigma_l^{(p+1)}(t-1)\| < \xi$$

Then choose the lth regressor covariance matrix as $\Sigma_l = \Sigma_l^{\text{best}}(t)$.

The algorithmic parameter that needs to be set appropriately is the population size p. The population size depends on the dimension of Σ and the objective function to be optimized. Generally, an appropriate value of p has to be found empirically. This is very similar to for example the choice of population size in the genetic algorithm.

[1]Each $\Sigma_l^{(i)}$, $1 \leqslant i \leqslant p$, has an associated weighting value δ_i. When $\Sigma_l^{(p+1)}$ or $\Sigma_l^{(p+2)}$ replaces Σ_l^{worst}, it will keep the weighting value of Σ_l^{worst}.

3.3. Repeated weighted optimization algorithm

The above weighted optimization algorithm performs a guided random search and the solution obtained may depend on the initial choice of population. To derive a robust algorithm that guarantees a global optimal solution, one may incorporate the full idea of the scatter search [10–12] with this weighted optimization algorithm. However, to avoid an overly complicated algorithm, we simply augment the algorithm into the following repeated weighted optimization algorithm. The aim is not to guarantee a global optimal solution. Rather it is to make sure that the algorithm will arrive at similar solutions regardless of the initial choices of population.

Initialization: Give a positive integer number M for controlling the maximum repeating times, and choose a small positive number ξ_1 for terminating the search.

First generation: Randomly choose the p number of the initial population $\Sigma_l^{(1)}, \ldots, \Sigma_l^{(p)}$, and call the weighted optimization algorithm to obtain a solution Σ_l^{best}.

Repeat loop: For $i = 1{:}M$

Set $\Sigma_l^{(1)} = \Sigma_l^{\text{best}}$, and randomly generate the other $p-1$ points $\Sigma_l^{(i)}$ for $2 \leqslant i \leqslant p$.
Call the weighted optimization algorithm to obtain a solution Σ_l^{best}.
If $\|\Sigma_l^{(1)} - \Sigma_l^{\text{best}}\| < \xi_1$
Exit loop;
End if
End for
Choose the lth regressor's covariance matrix as $\Sigma_l = \Sigma_l^{\text{best}}$.

The important algorithmic parameter that need to be chosen appropriately is the termination criterion ξ_1. Basically, ξ_1 determines whether the solutions obtained in different runs of the weighted optimization are closely enough to be regarded as the same solution. If a too small ξ_1 is chosen, the loop may keep going for long time. To safeguard against this, we also specify the maximum repeating times M. Again, appropriate values for M and ξ_1 depends on the dimension of Σ and how hard the objective function to be optimized. Also the choice of p has some influence on the choice of M and ξ_1. Generally, these algorithmic parameters have to be found empirically.

4. OLS algorithm for subset model selection

Once the full regression matrix $\mathbf{G}$ has been designed, the standard OLS algorithm [3,4] can be used to select a subset model. Let an orthogonal decomposition of the regression matrix be

$$\mathbf{G} = \mathbf{\Phi A} \tag{14}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{15}$$

and

$$\mathbf{\Phi} = [\boldsymbol{\varphi}_1 \ \boldsymbol{\varphi}_2 \cdots \boldsymbol{\varphi}_N] \tag{16}$$

with orthogonal columns that satisfy $\boldsymbol{\varphi}_i^{\mathrm{T}}\boldsymbol{\varphi}_j = 0$, if $i \neq j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \mathbf{\Phi}\mathbf{w} + \boldsymbol{\varepsilon} \tag{17}$$

where the orthogonal weight vector $\mathbf{w} = [w_1 \ w_2 \cdots w_N]^{\mathrm{T}}$ satisfy the triangular system

$$\mathbf{A}\boldsymbol{\theta} = \mathbf{w}. \tag{18}$$

Knowing $\mathbf{A}$ and $\mathbf{w}$, $\boldsymbol{\theta}$ can readily be solved from (18).

For the orthogonal regression model (17), the MSE

$$S_N = \frac{1}{N}\boldsymbol{\varepsilon}^{\mathrm{T}}\boldsymbol{\varepsilon} \tag{19}$$

can be expressed as

$$S_N = \frac{1}{N}\mathbf{y}^{\mathrm{T}}\mathbf{y} - \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\phi}_i^{\mathrm{T}}\boldsymbol{\phi}_i w_i^2. \tag{20}$$

Thus the MSE for the l-term subset model can be expressed recursively as

$$S_l = S_{l-1} - \frac{1}{N}\boldsymbol{\phi}_l^{\mathrm{T}}\boldsymbol{\phi}_l w_l^2. \tag{21}$$

At the lth stage of regression, the lth term is selected to maximize the error reduction criterion [3,4]

$$\Delta S_l = \frac{1}{N}\boldsymbol{\phi}_l^{\mathrm{T}}\boldsymbol{\phi}_l w_l^2. \tag{22}$$

The forward selection procedure is terminated at the N_sth stage if

$$S_{N_s} < \zeta \tag{23}$$

is satisfied, where the small positive scalar ζ is a chosen tolerance. This produces a parsimonious model containing N_s significant regressors.

In this study, we should assume that an appropriate tolerance value ζ can be chosen. It is worth emphasizing that the termination of the model construction process can alternatively be decided using cross validation [17,20]. A simple method is to have a separate validation data set. The model construction is based on the training data set, while the performance of the selected model, the MSE (20), is monitored over the validation data set. The construction process is terminated when

the MSE over the validation data set stops improving. Instead of using the pure least squares cost (20), it is also worth pointing out that other criteria can alternatively be adopted for the orthogonal forward selection, and these include regularization, optimal experimental design, and leave-one-out cross validation criterion [5,6].

5. Modeling examples

Two real-data sets were used to demonstrate the effectiveness of the proposed sparse model construction algorithm. The population size p, the maximum repeating times M and the termination criterion ξ_1 were chosen empirically to ensure that the OLS subset selection procedure could produce consistent final models with the same levels of modeling accuracy and model sparsity for repeating runs.

Example 1. This example constructed a model representing the relationship between the fuel rack position (input u_k) and the engine speed (output y_k) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed. Detailed system description and experimental setup can be found in [1]. The data set, depicted in Fig. 2, contained 410 samples. The first 210 data points were used in training and

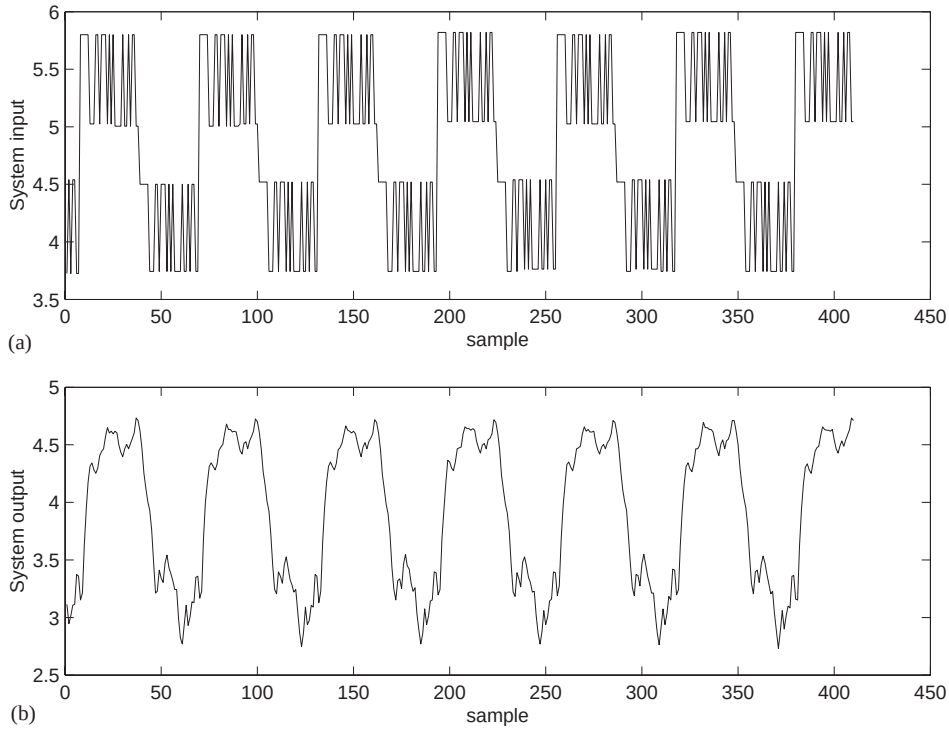


Fig. 2. The engine data set: (a) system input u_k, and (b) system output y_k.

the last 200 points in model validation. The previous study [1] has shown that this data set can be modeled adequately as

$$y_k = f_s(\mathbf{x}_k) + \varepsilon_k \tag{24}$$

with $f_s(\bullet)$ describing the unknown underlying system and the system input vector defining by

$$\mathbf{x}_k = [y_{k-1}\ u_{k-1}\ u_{k-2}]^{\mathrm{T}} \tag{25}$$

The previous results [5,6] have shown that when fitting a Gaussian kernel model with a single common variance, $\sigma^2 = 1.69$ is the optimal value for this kernel variance. Since every training input data points were considered as a candidate regressor's center, there were 210 regressors for the full Gaussian kernel model. With the tolerance level set to $\zeta = 5.5 \times 10^{-4}$, the OLS algorithm selected a 19-term subset model from the full regression model, and the resulting subset model is listed in Table 1. The MSE values of the resulting model were 5.28×10^{-4} for the training set and 6.72×10^{-4} for the validation set, respectively. Fig. 3 shows the corresponding model prediction $\hat{y}_k$ and the model prediction error $\varepsilon_k = y_k - \hat{y}_k$.

The proposed sparse model construction algorithm was then applied to construct a generalized Gaussian kernel model. The algorithmic parameters of the repeated

Table 1
Subset model generated for the engine data set by the OLS algorithm with a Gaussian kernel model of a single common variance

Step l	Mean vector $\mathbf{x}_l$			Diagonal covariance matrix Σ_l			Weight θ_l	MSE $S_l \times 100$
0								1558.9
1	4.2823	5.0245	5.0245	1.69	1.69	1.69	−109.2247	73.9841
2	2.8236	3.7439	3.7439	1.69	1.69	1.69	2.4249	34.7312
3	4.5954	5.8200	5.8200	1.69	1.69	1.69	16.0325	8.3802
4	3.1978	5.8200	3.7439	1.69	1.69	1.69	5.0481	7.5403
5	3.9310	3.7439	4.5006	1.69	1.69	1.69	−2.0419	4.6502
6	4.2976	5.0439	5.0439	1.69	1.69	1.69	106.5281	2.9565
7	4.6183	4.5006	5.0051	1.69	1.69	1.69	0.1787	2.4999
8	3.2131	5.8006	5.8006	1.69	1.69	1.69	−58.8794	1.5953
9	4.5725	5.8006	5.8006	1.69	1.69	1.69	−17.0584	0.7767
10	3.9844	4.5200	4.5200	1.69	1.69	1.69	4.3978	0.5986
11	2.8618	3.7439	4.5200	1.69	1.69	1.69	25.1798	0.4682
12	3.4498	4.5200	3.7439	1.69	1.69	1.69	−0.8959	0.3327
13	3.2284	5.8006	5.8006	1.69	1.69	1.69	61.2593	0.2065
14	2.9381	3.7439	4.5006	1.69	1.69	1.69	−110.8486	0.1589
15	3.1520	5.8006	3.7245	1.69	1.69	1.69	−4.5398	0.1292
16	3.6866	5.8200	5.8200	1.69	1.69	1.69	−2.1195	0.1032
17	2.9763	3.7439	4.5200	1.69	1.69	1.69	91.5013	0.0758
18	3.3735	3.7245	4.5394	1.69	1.69	1.69	−22.2389	0.0579
19	3.5491	3.7439	4.5200	1.69	1.69	1.69	16.7227	0.0528

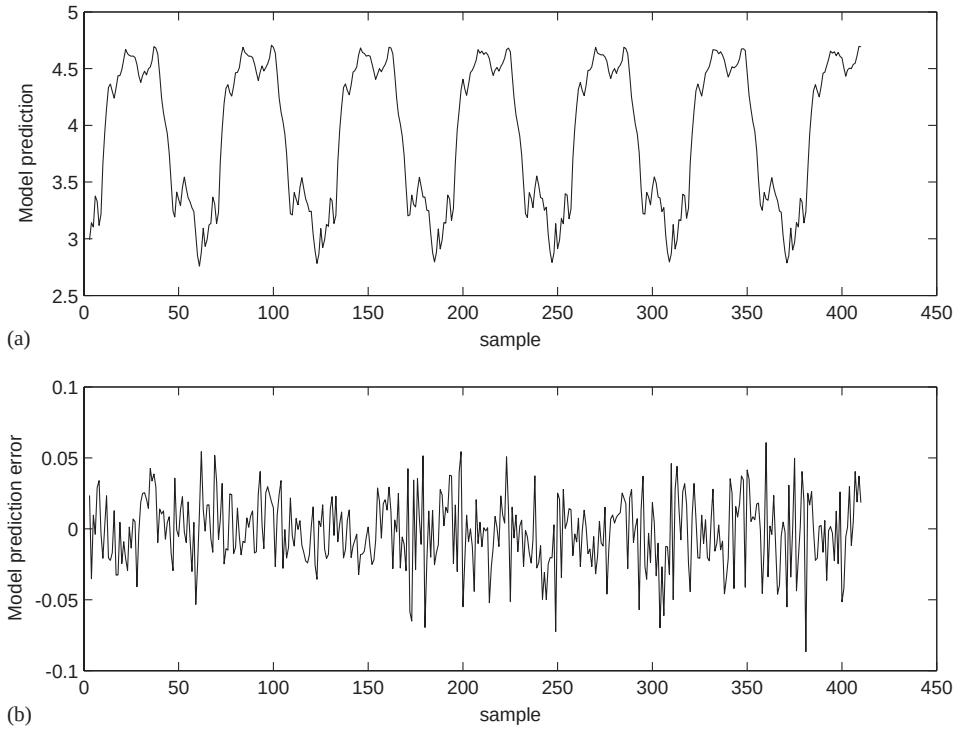


Fig. 3. The engine data set: (a) model prediction $\hat{y}_k$, and (b) model prediction error $\varepsilon_k = y_k - \hat{y}_k$ using the 19-term Gaussian kernel model with a single common kernel variance.

weighted optimization for kernel covariance fitting were chosen to be $p = 37$, $M = 60$ and $\xi_1 = 0.0002$. Using the same tolerance level of $\zeta = 5.5 \times 10^{-4}$, the OLS algorithm selected a 11-term subset model from the full generalized Gaussian kernel model, and the obtained model is listed in Table 2. The MSE values of this model were 5.09×10^{-4} over the training set and 5.19×10^{-4} over the validation set, respectively. The model prediction and prediction error generated by this model are illustrated in Fig. 4.

Example 2. This example constructed a model for the gas furnace data set (Series J in [2]). The data set, illustrated in Fig. 5, contained 296 pairs of input–output points, where the input u_k was the coded input gas feed rate and the output y_k represented CO_2 concentration from the gas furnace. All the 296 data points were used in training, with the model input vector defined by

$$\mathbf{x}_k = [y_{k-1} \;\; y_{k-2} \;\; y_{k-3} \;\; u_{k-1} \;\; u_{k-2} \;\; u_{k-3}]^{\mathrm{T}}. \tag{26}$$

For this data set, the previous experiments have found out that it was difficult for the existing state-of-art kernel regression techniques to fit a Gaussian kernel regression model using a common kernel variance [6]. Various existing state-of-art kernel regression techniques were then used in [6] to fit a thin-plate-spline regression model

Table 2
Subset model generated for the engine data set by the OLS algorithm with a generalized Gaussian kernel model

Step l	Mean vector $\mathbf{x}_l$			Diagonal covariance matrix Σ_l			Weight θ_l	MSE $S_l \times 100$
0								1558.9
1	4.6030	5.8006	5.8006	4.6610	23.2494	18.7487	−52.9824	0.9292
2	4.5114	5.8006	5.8006	4.2126	22.5550	18.0605	53.9543	0.1655
3	4.4579	5.0245	5.8006	2.7926	14.5527	33.8069	−74.9670	0.1202
4	4.4503	5.0051	5.8006	3.5534	360.546	12.8974	−74.5696	0.1134
5	3.2284	5.8006	5.8006	311.554	12.6886	7.5157	−246.1931	0.1129
6	4.6183	5.0051	5.8006	4.8006	48.6543	12.6258	96.1724	0.1007
7	3.6637	5.8006	5.8006	190.214	12.6563	7.5715	245.7579	0.0898
8	4.3510	5.0245	5.0245	2.8708	6.8213	253.1952	13.8707	0.0813
9	3.1062	4.5394	3.7245	400.00	400.00	400.000	−2.5807	0.0642
10	4.3663	5.0439	5.8200	2.2056	40.4580	75.2890	50.1908	0.0592
11	3.9233	3.7439	4.5200	2.0241	327.7485	263.2715	−4.3783	0.0509

The kernel covariance matrices are determined by maximizing the correlation criterion using the repeated weighted optimization algorithm.

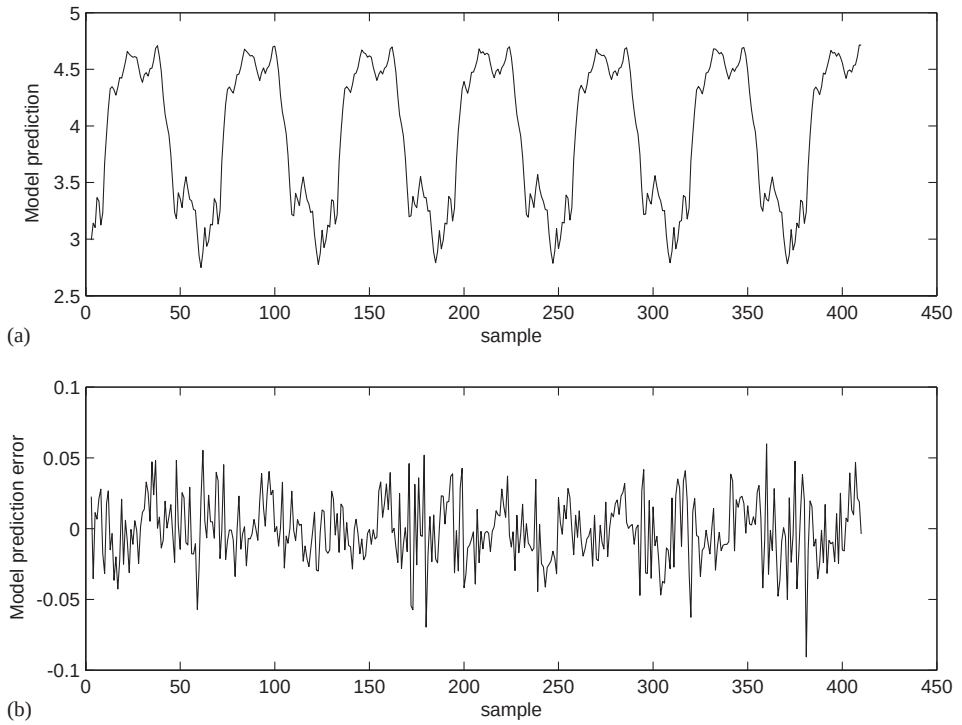


Fig. 4. The engine data set: (a) model prediction $\hat{y}_k$, and (b) model prediction error $\varepsilon_k = y_k - \hat{y}_k$ using the 11-term generalized Gaussian kernel model, each kernel having an individually tuned diagonal covariance matrix.

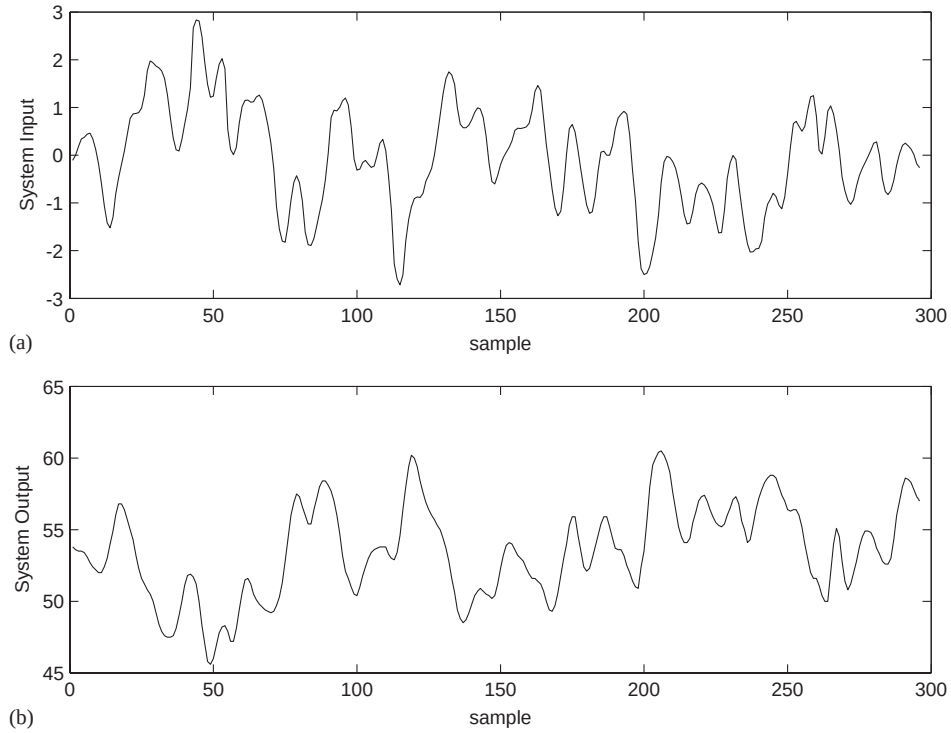


Fig. 5. The gas furnace data set: (a) system input u_k, and (b) system output y_k.

for this data set and the best result obtained required at least 30 model terms to achieve a modeling accuracy of $\zeta = 0.054$.

The proposed sparse model algorithm was employed to construct a generalized Gaussian kernel model for this data set. The kernel covariance matrices were first determined using the repeated weighted optimization with the following algorithmic parameters: $p = 100$, $M = 60$ and $\xi_1 = 0.0001$. With the modeling accuracy of $\zeta = 0.054$, the OLS algorithm constructed a 20-term subset model from the full generalized Gaussian kernel model, as is listed in Table 3. The model prediction and prediction error generated by this model are shown in Fig. 6.

6. Conclusions

A novel construction algorithm has been developed for the generalized Gaussian kernel model. Each kernel regressor in the pool of candidate regressors has an individual diagonal covariance matrix, which is determined by maximizing the absolute value of the correlation between the regressor and the training data using a repeated weighted search optimization. The standard orthogonal least squares

Table 3
Subset model generated for the gas furnace data set by the OLS algorithm with a generalized Gaussian kernel model

Step l	Mean vector $\mathbf{x}_l$/diagonal covariance matrix $0.01 \times \Sigma_l$						Weight θ_l	MSE S_l
0								2844.3
1	59.5	58.0	55.6	−2.053	−2.330	−2.4730	−267.5601	1.2067
	5.0681	94.9960	58.5221	272.6948	158.8822	1.3383		
2	57.2	56.4	55.4	−1.474	−1.746	−1.8910	221.7796	0.1549
	3.7912	16.1987	213.2852	238.2681	51.3793	1.0180		
3	55.0	55.6	56.8	−1.525	−1.086	−0.6200	95.2771	0.1307
	57.7623	270.1883	3.8217	243.6048	0.4934	261.9284		
4	56.0	54.3	53.0	−0.204	−0.528	−0.7400	−40.7100	0.1177
	2.1951	158.9803	343.7560	223.1368	1.3853	357.1565		
5	56.4	57.0	57.4	−0.848	−0.713	−0.6250	−349.0073	0.1061
	9.6873	6.0958	85.0066	328.2904	0.9786	295.3635		
6	51.4	52.8	54.5	−0.748	−0.458	0.0930	−81.1253	0.0974
	49.5275	396.8625	2.7015	47.2399	0.4224	61.6835		
7	57.3	57.0	56.2	−0.582	−0.634	−0.8130	−281.8031	0.0828
	2.5829	67.9224	139.6028	9.7573	91.9482	92.2856		
8	60.4	60.0	59.5	−1.261	−1.739	−2.0530	564.5599	0.0755
	4.7792	246.5125	215.1170	43.7362	4.4022	208.8816		
9	51.6	52.8	53.7	1.683	1.746	1.6070	−375.1686	0.0716
	320.0416	393.5093	7.9007	64.7479	302.4567	212.9779		
10	53.2	53.6	53.6	0.918	0.858	0.7820	355.5893	0.0639
	400.0000	12.1366	32.3310	318.6978	400.0000	86.1125		
11	53.8	53.7	53.6	0.254	−0.007	−0.2290	−150.5212	0.0596
	32.2065	49.2961	223.7835	19.9128	232.0460	0.4760		
12	54.0	54.1	53.9	0.301	0.161	0.0600	−57.9844	0.0571
	275.6604	3.0180	107.5660	313.2379	94.6038	32.9662		
13	50.6	49.7	49.3	−1.269	−1.099	−0.7140	−153.0328	0.0569
	385.9776	71.0925	94.5789	239.7465	0.2531	23.1095		
14	54.4	52.8	51.3	−1.456	−1.825	−1.7990	−17.1629	0.0561
	34.9812	236.0426	291.7318	50.2033	46.0414	0.4039		
15	56.0	56.4	56.4	0.605	0.709	0.6620	−116.3539	0.0557
	3.3890	12.3638	5.9307	337.1134	15.9024	196.7020		
16	52.3	51.2	50.4	−0.194	−0.424	−0.6030	−111.9460	0.0555
	257.1146	84.5112	387.4997	274.5492	181.3500	0.3622		
17	52.6	52.8	53.3	−0.759	−0.493	0.0	196.3985	0.0553
	325.0471	315.4355	32.2995	145.7428	0.3241	271.2281		
18	53.6	53.7	54.4	0.782	0.556	0.2090	85.7510	0.0550
	265.5988	115.8858	1.9625	164.8533	184.6377	133.1301		
19	54.6	55.9	55.9	0.109	0.484	0.6430	171.1110	0.0545
	121.0842	2.2050	74.6179	23.9226	201.9486	215.9408		
20	54.3	53.0	52.6	−0.528	−0.740	−0.8240	174.7513	0.0540
	105.1096	102.6449	14.6421	47.2963	80.0091	0.2617		

The kernel covariance matrices are determined by maximizing the correlation criterion using the repeated weighted optimization algorithm.

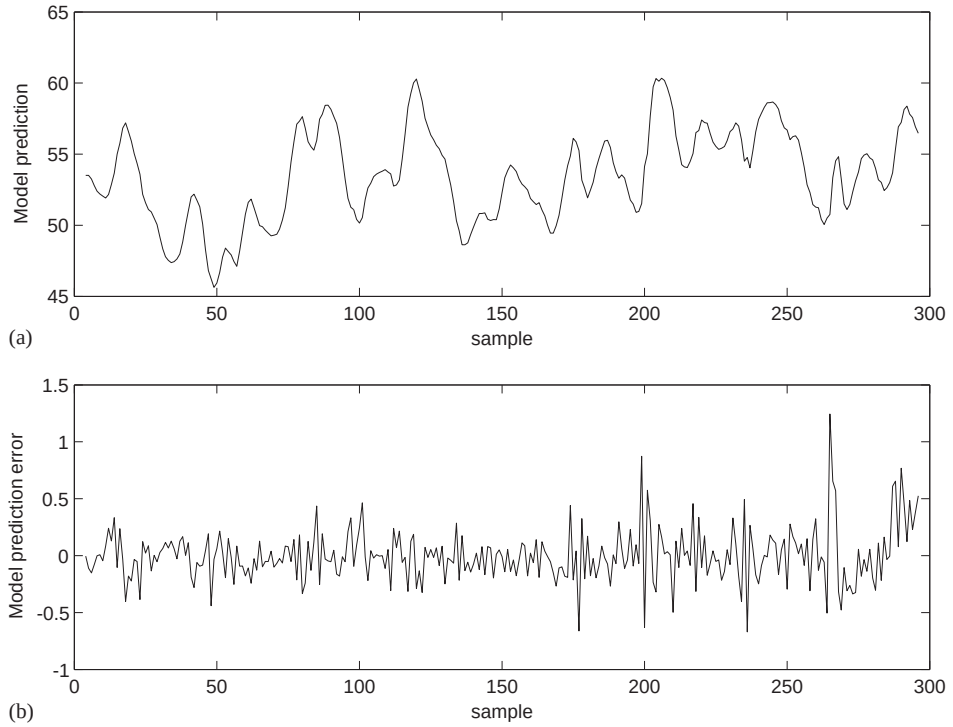


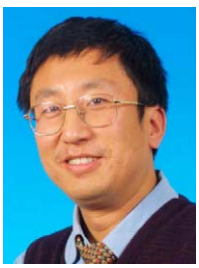
Fig. 6. The gas furnace data set: (a) model prediction $\hat{y}_k$, and (b) model prediction error $\varepsilon_k = y_k - \hat{y}_k$ using the 20-term generalized Gaussian kernel model, each kernel having an individually tuned diagonal covariance matrix.

algorithm is then applied to select a parsimonious model from the full regression matrix. Compared with the existing kernel regression modeling approaches which adopt a single common kernel variance for all the regressors, the proposed method has the advantages of improving modeling capability and producing sparser models. These advantages have been demonstrated by the experimental results involving two real data sets.

References

[1] 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.
[2] G.E.P. Box, G.M. Jenkins, Time Series Analysis, Forecasting and Control, Holden-Day Inc., San Francisco, 1976.
[3] 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.
[4] 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.

[5] 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. Automat. Control 48 (6) (2003) 1029–1036.
[6] 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 Cybernet. Part B 34 (2) (2004) 898–911.
[7] S. Chen, B.L. Luk, Adaptive simulated annealing for optimization in signal processing applications, Signal Process 79 (1) (1999) 117–128.
[8] 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.
[9] 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.
[10] F. Glover, A template for scatter search and path relinking, in: J.-K. Hao, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (Eds.), Artificial Evolution, Lecture Notes in Computer Science, vol. 1363, Springer, Berlin, 1998, pp. 13–54.
[11] F. Glover, Scatter search and path relinking, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw-Hill, New York, 1999, pp. 297–316.
[12] F. Glover, M. Laguna, R. Marti, Scatter search and path relinking: foundations and advanced designs, 2002, submitted for publication.
[13] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
[14] L. Ingber, Simulated annealing: practice versus theory, Math. Comput. Model. 18 (11) (1993) 29–57.
[15] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms: Concepts and Design, Springer, London, 1998.
[16] 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.
[17] R.H. Myers, Classical and Modern Regression with Applications, second ed., PWS-KENT, Boston, 1990.
[18] R.E. Schapire, The strength of weak learnability, Mach. Learn. 5 (2) (1990) 197–227.
[19] B. Schlkopf, A.J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, 2002.
[20] M. Stone, Cross validation choice and assessment of statistical predictions, J. R. Stat. Soc. Ser. B 36 (1974) 117–147.
[21] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[22] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[23] V. Vapnik, S. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, MA, 1997, pp. 281–287.



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 2001, he was a postdoctoral researcher in the State Key laboratory of Intelligent Technology and Systems, Beijing, China. From September 2001, he has been a research associate and now research fellow at the University of Portsmouth, Portsmouth, UK. Dr. Wang's main interests are in machine learning and neural networks, control theory and systems as well as robotics.t

Sheng Chen obtained a B.Eng. degree in control engineering from the East China Petroleum Institute, Dongying, China, in 1982, and a Ph.D. degree in control engineering from the City University at London in 1986. He joined the School of Electronics and Computer Science at the University of Southampton, UK, in September 1999. He previously held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, UK. Dr. Chen is a Senior Member of IEEE. His recent research works include adaptive nonlinear signal processing, modeling and identification of nonlinear systems, machine learning and neural network research, finite-precision digital controller design, evolutionary computation methods and optimization. He has published over 200 research papers.

David J. Brown received his Ph.D. degree in manipulator control from the University of Southampton, Southampton, UK, in April 1983. From August 1980 to 1985, he was a lecturer at the same university directing a research group specializing in digital motion control. He went on to be a technical director of an international electronics company based in Cambridge, UK, from 1985 to 1988. He then formed his own company, which specialized in motion control and diagnostics until 1998. He is now the director of the Computer Intelligence and Applications Research Group based at the University of Portsmouth, Portsmouth, UK. The group's main interests are in machine learning and neural networks.