Sparse Kernel Regression Modelling Using Combined Locally Regularized Orthogonal Least Squares and D-Optimality Experimental Design

S. Chen†, X. Hong‡ and C.J. Harris†

† Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.

‡ Department of Cybernetics
University of Reading, Reading RG6 6AY, U.K.

Abstract

The paper proposes an efficient nonlinear identification algorithm by combining a locally regularized orthogonal least squares (LROLS) model selection with a D-optimality experimental design. The proposed algorithm aims to achieve maximized model robustness and sparsity via two effective and complementary approaches. The LROLS method alone is capable of producing a very parsimonious model with excellent generalization performance. The D-optimality design criterion further enhances the model efficiency and robustness. An added advantage is that the user only needs to specify a weighting for the D-optimality cost in the combined model selecting criterion and the entire model construction procedure becomes automatic. The value of this weighting does not influence the model selection procedure critically and it can be chosen with ease from a wide range of values.

Keywords: Sparse modelling, orthogonal least squares, regularization, Bayesian learning, optimal experimental design, D-optimality.

1 Introduction

A basic principle in practical nonlinear data modelling is the parsimonious principle. A large class of nonlinear models and neural networks can be classified as a kernel regression model [1]–[3]. For this class of nonlinear models, the orthogonal least squares (OLS) algorithm [4],[5] is an efficient learning procedure for constructing sparse regression models. If data are highly noisy, however, the parsimonious principle alone may not be entirely immune to overfitting, and small models constructed may still fit into noise. A useful technique for overcoming overfitting is regularization [6]–[8]. From the powerful Bayesian learning viewpoint, a regularization parameter is equivalent to the ratio of the related hyperparameter to the noise parameter and an effective Bayesian learning method is the evidence procedure which iteratively optimizes model parameters and associated hyperparameters [9]. Adopting this Bayesian learning method to regression models, the LROLS algorithm [10]-[12] has recently been proposed, which introduces individual regularizer for each weight. This LROLS algorithm provides an efficient procedure for constructing sparse models from noisy data that generalize well.

Optimal experimental designs [13] have been used to construct smooth model response surfaces based on the setting of the experimental variables under well controlled experimental conditions. In optimal design, model adequacy is evaluated by design criteria that are statistical measures of goodness of experimental designs by virtue of design efficiency and experimental effort. For kernel regression models, quantitatively, model adequacy is measured as function of the eigenvalues of the design matrix. There are a variety of optimal design criteria based on different aspects of experimental design [13]. The D-optimality criterion is most effective in optimizing the parameter efficiency and model robustness via the maximization of the determinant of the design matrix. The traditional nonlinear model structure determination based on optimal experimental designs is however inherent inefficient and computationally prohibitive, incurring the curse of dimensionality. In a recent work [14],[15], this computational difficulty is overcome and an efficient model construction algorithm has been proposed based on the OLS algorithm coupled with the D-optimality experimental design.

This note shows that further advantages can be gained by combining the LROLS algorithm with the D-optimality experimental design. Computational efficiency of the resulting algorithm as usual is ensured by the orthogonal forward selection procedure. The local regularization enforces model sparsity and avoids overfitting in model parameters while the D-optimality design optimizes model efficiency and parameter robustness. The coupling effects of these two approaches in the combined algorithm further enhance each other. Moreover, the model construction process becomes fully automatic, and there is only one user specified quantity which has no critical influence on the model selection procedure.

2 The kernel regression model

Consider the general discrete-time nonlinear system represented by the nonlinear model [1]:

$$y(k) = f(y(k-1), \cdots, y(k-n_y), u(k-1), \cdots, u(k-n_u)) + e(k) = f(\mathbf{x}(k)) + e(k), \qquad (1)$$

where $u(k)$ and $y(k)$ are the system input and output variables, respectively, n_u and n_y are positive integers representing the lags in $u(k)$ and $y(k)$, respectively, $e(k)$ is the system white noise, $\mathbf{x}(k) = [y(k-1) \cdots y(k-n_y)\ u(k-1) \cdots u(k-n_u)]^T$ denotes the system "input" vector, and $f(\bullet)$ is the unknown system mapping. The system model (1) is to be identified from an N-sample observation data set $\{\mathbf{x}(k), y(k)\}_{k=1}^N$ using some suitable functional which can approximate $f(\bullet)$ with arbitrary accuracy. One class of such functionals is the kernel regression model of the form:

$$y(k) = \hat{y}(k) + e(k) = \sum_{i=1}^{n_M} \theta_i \phi_i(\mathbf{x}(k)) + e(k), \qquad (2)$$

where $\hat{y}(k)$ denotes the model output, θ_i are the model weights, $\phi_i(\mathbf{x}(k))$ are the regressors, and n_M is the total number of candidate regressors or model terms.

By letting $\boldsymbol{\phi}_i = [\phi_i(\mathbf{x}(1)) \cdots \phi_i(\mathbf{x}(N))]^T$, for $1 \leq i \leq n_M$, and defining

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \ \ \boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \cdots \boldsymbol{\phi}_{n_M}], \ \ \boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_{n_M} \end{bmatrix}, \ \ \mathbf{e} = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}, \tag{3}$$

the regression model (2) can be written in the matrix form

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{e}. \tag{4}$$

Let an orthogonal decomposition of the matrix $\boldsymbol{\Phi}$ be

$$\boldsymbol{\Phi} = \mathbf{W}\mathbf{A}, \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,n_M} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n_M-1,n_M} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{6}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \cdots \mathbf{w}_{n_M}] \tag{7}$$

with columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \neq j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\mathbf{g} + \mathbf{e}, \tag{8}$$

where the orthogonal weight vector $\mathbf{g} = [g_1 \cdots g_{n_M}]^T$ satisfy the triangular system $\mathbf{A}\boldsymbol{\theta} = \mathbf{g}$.

3 The locally regularized OLS algorithm with D-optimality design

The LROLS algorithm [10]-[12] adopts the following regularized error criterion:

$$J_R(\mathbf{g}, \boldsymbol{\lambda}) = \mathbf{e}^T\mathbf{e} + \sum_{i=1}^{n_M} \lambda_i g_i^2 = \mathbf{e}^T\mathbf{e} + \mathbf{g}^T \boldsymbol{\Lambda} \mathbf{g}, \tag{9}$$

where $\boldsymbol{\lambda} = [\lambda_1 \cdots \lambda_{n_M}]^T$ is the regularization parameter vector, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \cdots, \lambda_{n_M}\}$. It can readily be shown that the criterion (9) can be expressed as

$$\mathbf{e}^T\mathbf{e} + \mathbf{g}^T \boldsymbol{\Lambda} \mathbf{g} = \mathbf{y}^T\mathbf{y} - \sum_{i=1}^{n_M} \left(\mathbf{w}_i^T \mathbf{w}_i + \lambda_i \right) g_i^2. \tag{10}$$

Normalizing (10) by $\mathbf{y}^T\mathbf{y}$ yields

$$\left(\mathbf{e}^T\mathbf{e} + \mathbf{g}^T\mathbf{\Lambda}\mathbf{g}\right)/\mathbf{y}^T\mathbf{y} = 1 - \sum_{i=1}^{n_M}\left(\mathbf{w}_i^T\mathbf{w}_i + \lambda_i\right)g_i^2/\mathbf{y}^T\mathbf{y}. \tag{11}$$

As in the case of the OLS algorithm [4], the regularized error reduction ratio due to $\mathbf{w}_i$ is defined by

$$[\mathrm{rerr}]_i = \left(\mathbf{w}_i^T\mathbf{w}_i + \lambda_i\right)g_i^2/\mathbf{y}^T\mathbf{y}. \tag{12}$$

Based on this ratio, significant regression can be selected in a forward regression procedure, and the selection process is terminated at the n_s-th stage when

$$1 - \sum_{l=1}^{n_s}[\mathrm{rerr}]_l < \xi \tag{13}$$

is satisfied, where $0 < \xi < 1$ is a chosen tolerance. This produces a sparse model containing n_s $(\ll n_M)$ significant regressors. The Bayesian evidence procedure [9] can readily be used to optimize the regularization parameters. Applying this evidence procedure leads to the updating formulas:

$$\lambda_i^{\mathrm{new}} = \frac{\gamma_i^{\mathrm{old}}}{N - \gamma^{\mathrm{old}}}\frac{\mathbf{e}^T\mathbf{e}}{g_i^2}, \ \ 1 \le i \le n_M, \tag{14}$$

where

$$\gamma_i = \frac{\mathbf{w}_i^T\mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T\mathbf{w}_i} \tag{15}$$

and

$$\gamma = \sum_{i=1}^{n_M}\gamma_i. \tag{16}$$

Usually a few iterations (typically 10 to 20) are sufficient to find an optimal $\boldsymbol{\lambda}$.

It is worth emphasizing that, for the LROLS algorithm, the choice of ξ is less critical than the original OLS algorithm. This is because multiple regularizers enforce sparsity. If, for example, ξ is chosen too small, those unnecessarily added terms will have very large λ_l associated with them, effectively forcing their weights to be zero [10]-[12]. Nevertheless, an appropriate value for ξ is desired. Alternatively, the Akaike information criterion (AIC) [16],[17] can be adopted to terminate the subset model selection process. The AIC can be viewed as a model structure regularization by conditioning the model size using a penalty term to penalize large sized models. However, the use of AIC or other information based criteria in forward regression only affects the stopping point of the model selection, but does not penalizes the regressor that may cause poor model performance (e.g. too large variance of parameter estimate or ill-posedness of the regression matrix), if it is selected. In simple words, the penalty term in AIC does not tell which regressor should be selected. Optimal experimental design criteria offer better solutions as they are directly linked to model efficiency and parameter robustness.

In experimental design, the data covariance matrix $\mathbf{\Phi}^T\mathbf{\Phi}$ is called the design matrix. The D-optimality design criterion maximizes the determinant of the design matrix for the constructed model. Specifically, let $\mathbf{\Phi}_{n_s}$ be a column subset of $\mathbf{\Phi}$ representing a constructed n_s-term subset model. According to the D-optimality criterion, the selected subset model is the one that maximizes $\det(\mathbf{\Phi}_{n_s}^T\mathbf{\Phi}_{n_s})$. It is straightforward to verify that maximizing $\det(\mathbf{\Phi}_{n_s}^T\mathbf{\Phi}_{n_s})$ is identical to maximizing $\det(\mathbf{W}_{n_s}^T\mathbf{W}_{n_s})$ or, equivalently, minimizing $-\log\det(\mathbf{W}_{n_s}^T\mathbf{W}_{n_s})$ [14],[15]. Thus the combined LROLS and D-optimality algorithm can be viewed as based on the combined criterion

$$J_C(\mathbf{g},\boldsymbol{\lambda},\beta) = J_R(\mathbf{g},\boldsymbol{\lambda}) + \beta\sum_{i=1}^{n_M} -\log(\mathbf{w}_i^T\mathbf{w}_i), \tag{17}$$

where β is a fixed small positive weighting for the D-optimality cost. In this combined algorithm, the updating of the model weights and regularization parameters is exactly as in the LROLS algorithm, but the selection is according to the combined error reduction ratio defined as

$$[\text{cerr}]_i = \left((\mathbf{w}_i^T\mathbf{w}_i + \lambda_i)g_i^2 + \beta\log(\mathbf{w}_i^T\mathbf{w}_i)\right)/\mathbf{y}^T\mathbf{y} \tag{18}$$

and the selection is terminated with an n_s-term model when

$$[\text{cerr}]_l \le 0 \;\text{ for }\; n_s + 1 \le l \le n_M. \tag{19}$$

The iterative model selection procedure can now be summarized:

Initialization. Set λ_i, $1 \le i \le n_M$, to the same small positive value (e.g. 0.001), and choose a fixed β

Step 1. Given the current $\boldsymbol{\lambda}$, use the procedure described in Appendix to select a subset model with n_s terms.

Step 2. Update $\boldsymbol{\lambda}$ using (14) with $n_M = n_s$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a pre-set maximum iteration number is reached, stop; otherwise go to *Step 1*.

The introduction of the D-optimality cost into the algorithm further enhances the efficiency and robustness of the selected subset model and, as a consequence, the combined algorithm can often produce sparser models with equally good generalization properties, compared with the LROLS algorithm. An additional advantage is that it simplifies the selection procedure. Notice that it is no longer needed to specify the tolerance ξ and the algorithm will terminate automatically when the condition (19) is reached. Unlike the combined OLS and D-optimality algorithm [14],[15], the value of weighting β does not critically influence the performance of this combined LROLS and D-optimality algorithm. The weighting β can (almost) be chosen arbitrarily from a large range of values. This will be demonstrated in the following modelling examples.

4 Modelling examples

Example 1. This example used a radial basis function (RBF) network to model the scalar function

$$f(x) = \sin(2\pi x), \ \ 0 \le x \le 1. \tag{20}$$

For a detailed description of the RBF network see, for example, [5]. The RBF model employed Gaussian kernel function with a variance of 0.04. One hundred training data were generated from $y = f(x) + e$, where the input x was uniformly distributed in $(0,\ 1)$ and the noise e was Gaussian with zero mean and variance 0.16. The noisy training points y and the underlying function $f(x)$ are plotted in Fig. 1 (a). As each training data x was considered as a candidate RBF center, there were $n_M = 100$ regressors in the regression model (2). The training data were extremely noisy. One hundred noise-free data $f(x)$ with equally spaced x were also generated as the testing data set for model validation. In the previous works [10]-[12], it was shown that without regularization the constructed models suffered from a serious overfitting problem, and the LROLS algorithm was able to overcome this problem and produced a sparse 6-term model, with the mean square error (MSE) values over the noisy training set and the noise-free testing set being 0.15917 and 0.00181, respectively.

Table 1 compares the MSE values over the training and testing sets for the models constructed by the combined LROLS and D-optimality algorithm with those of the combined OLS and D-optimality algorithm [14],[15], given a wide range of β values. It can be seen clearly that using the D-optimality alone without regularization the constructed models can still fit into the noise unless the weighting β is set to some appropriate value. Combining regularization with D-optimality design, the results obtained are consistent over a wide range of β values and, effectively, the value of β has no serious influence on the model construction process. It can also be seen that the combined LROLS and D-optimality algorithm was able to produce a sparser 5-term model with equally good generalization properties, compared with the result using the LROLS algorithm alone [10]-[12]. The model map of the 5-term model produced by the combined LROLS and D-optimality algorithm with $\beta = 10^{-5}$ is shown in Fig. 1 (b).

Example 2. This was a two-dimensional simulated nonlinear time series given by

$$\begin{aligned} y(k) \ &= \ \Big(0.8 - 0.5\exp(-y^2(k-1))\Big)\, y(k-1) - \Big(0.3 + 0.9\exp(-y^2(k-1))\Big)\, y(k-2) \\ &+ \ 0.1\sin(\pi y(k-1)) + e(k) \end{aligned} \tag{21}$$

where the noise $e(k)$ was Gaussian with zero mean and variance 0.09. One thousand noisy samples were generated given $y(0) = y(-1) = 0.0$. The first 500 data points were used for training, and the other 500 samples were used for model validation. The underlying noise-free system

$$y_d(k) \ = \ \Big(0.8 - 0.5\exp(-y_d^2(k-1))\Big)\, y_d(k-1) - \Big(0.3 + 0.9\exp(-y_d^2(k-1))\Big)\, y_d(k-2)$$

$$+ \ 0.1 \sin(\pi y_d(k-1)) \tag{22}$$

was specified by a limit circle, as shown in Fig. 2 (a). A Gaussian RBF model of the form

$$\hat{y}(k) = \hat{f}_{RBF}(\mathbf{x}(k)) \quad \text{with} \quad \mathbf{x}(k) = [y(k-1) \ y(k-2)]^T \tag{23}$$

was constructed using the noisy training data. The Gaussian kernel function had a variance of 0.81. As each data point $\mathbf{x}(k)$ was considered as a candidate RBF center, there were $n_M = 500$ candidate regressors. The previous study [10]-[12] constructed a sparse 18-term model using the LROLS algorithm, with the MSE values over the training and testing sets being 0.09264 and 0.09678, respectively.

The modelling accuracies over both the training and testing sets are compared in Table 2 for the two algorithms, the combined LROLS and D-optimality and the combined OLS and D-optimality, with a range of β values. Again it is seen that, when combining with the D-optimality design, the LROLS was able to produce sparser models with equally good generalization performance, compared with the result obtained using the LROLS algorithm alone. It is also clear that for the combined LROLS and D-optimality algorithm the model construction process is insensitive to the value of β. The RBF model produced by the combined LROLS and D-optimality algorithm with $\beta = 10^{-4}$ was used to iteratively generate the time series according to

$$\hat{y}_d(k) = \hat{f}_{RBF}(\mathbf{x}_d(k)) \quad \text{with} \quad \mathbf{x}_d(k) = [\hat{y}_d(k-1) \ \hat{y}_d(k-2)]^T, \tag{24}$$

given $\mathbf{x}_d(0) = [0.1 \ 0.1]^T$. The resulting phase plot is shown in Fig. 2 (b).

Example 3. This example constructed a model representing the relationship between the fuel rack position (input) and the engine speed (output) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed. It is known that at low engine speed, the relationship between the input and output is nonlinear [18]. Detailed system description and experimental setup can be found in [18]. The data set contained 410 samples. The first 210 data points were used in modelling and the last 200 points in model validation. A RBF model of the form:

$$\hat{y}(k) = \hat{f}_{RBF}(\mathbf{x}(k)) \quad \text{with} \quad \mathbf{x}(k) = [y(k-1) \ u(k-1) \ u(k-2)]^T \tag{25}$$

was used to model the data. The variance of the RBF kernel function was chosen to be 1.69. As each input vector $\mathbf{x}(k)$ was considered as a candidate RBF center, there were $n_M = 210$ candidate regressors. Previously, a 34-term model was constructed using the LROLS algorithm [10]–[12], and the resulting MSE values over the training and testing sets were 0.000435 and 0.000487, respectively.

The MSE values of the models produced by the combined LROLS and D-optimality algorithm and the combined OLS and D-optimality one are compared in Table 3, given a range of β values. It can be seen again that the former is insensitive to the weighting value for the D-optimality cost. This real-data identification example really demonstrates the power of combining regularization with the D-optimality design: the ability to produce a much sparser model with similar good generalization performance, compared with relying on regularization alone. The constructed RBF model by the combined LROLS and D-optimality algorithm with $\beta = 10^{-5}$ was used to generate the one-step prediction $\hat{y}(k)$ of the system output according to (25). The iterative model output $\hat{y}_d(k)$ was also produced using

$$\hat{y}_d(k) = \hat{f}_{RBF}(\mathbf{x}_d(k)) \quad \text{with} \quad \mathbf{x}_d(k) = [\hat{y}_d(k-1)\ u(k-1)\ u(k-2)]^T. \tag{26}$$

The one-step model prediction and iterative model output for this 22-term model selected by the combined LROLS and D-optimality algorithm are shown in Fig. 3, in comparison with the system output.

5 Conclusions

A locally regularized OLS algorithm with the D-optimality design has been proposed for nonlinear system identification using the kernel regression model. It has been demonstrated that combining regularization with the D-optimality experimental design provides a state-of-art procedure for constructing very sparse models with excellent generalization performance. It has been shown that the performance of the algorithm is insensitive to the D-optimality cost weighting, and the model construction process is fully automated. The computational requirements of this iterative model selection procedure are very simple and its implementation straightforward.

Appendix

The modified Gram-Schmidt orthogonalization procedure calculates the $\mathbf{A}$ matrix row by row and orthogonalizes $\mathbf{\Phi}$ as follows: at the l-th stage make the columns $\boldsymbol{\phi}_j$, $l+1 \leq j \leq n_M$, orthogonal to the l-th column and repeat the operation for $1 \leq l \leq n_M - 1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)} = \boldsymbol{\phi}_j$, $1 \leq j \leq n_M$, then

$$\left.\begin{array}{l} \mathbf{w}_l = \boldsymbol{\phi}_l^{(l-1)}, \\ a_{l,j} = \mathbf{w}_l^T \boldsymbol{\phi}_j^{(l-1)} / \left(\mathbf{w}_l^T \mathbf{w}_l\right), \ \ l+1 \leq j \leq n_M, \\ \boldsymbol{\phi}_j^{(l)} = \boldsymbol{\phi}_j^{(l-1)} - a_{l,j}\mathbf{w}_l, \ \ l+1 \leq j \leq n_M, \end{array}\right\} l = 1, 2, \cdots, n_M - 1. \tag{27}$$

The last stage of the procedure is simply $\mathbf{w}_{n_M} = \boldsymbol{\phi}_{n_M}^{(n_M-1)}$. The elements of $\mathbf{g}$ are computed by transforming $\mathbf{y}^{(0)} = \mathbf{y}$ in a similar way:

$$\left.\begin{array}{l} g_l = \mathbf{w}_l^T \mathbf{y}^{(l-1)} / \left(\mathbf{w}_l^T \mathbf{w}_l + \lambda_l\right), \\ \mathbf{y}^{(l)} = \mathbf{y}^{(l-1)} - g_l \mathbf{w}_l, \end{array}\right\} 1 \leq l \leq n_M. \tag{28}$$

This orthogonalization scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner. First define

$$\mathbf{\Phi}^{(l-1)} = \left[\mathbf{w}_1 \cdots \mathbf{w}_{l-1} \, \boldsymbol{\phi}_l^{(l-1)} \cdots \boldsymbol{\phi}_{n_M}^{(l-1)}\right]. \tag{29}$$

If some of the columns $\boldsymbol{\phi}_l^{(l-1)}, \cdots, \boldsymbol{\phi}_{n_M}^{(l-1)}$ in $\mathbf{\Phi}^{(l-1)}$ have been interchanged, this will still be referred to as $\mathbf{\Phi}^{(l-1)}$ for notational convenience. The l-th stage of the selection procedure is given as follows.

Step 1. For $l \le j \le n_M$, compute

$$\left.\begin{aligned} g_l^{(j)} &= \left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \mathbf{y}^{(l-1)} / \left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)} + \lambda_j\right), \\ [\mathrm{cerr}]_l^{(j)} &= \left(\left(g_l^{(j)}\right)^2 \left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)} + \lambda_j\right) + \beta \log\left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)}\right)\right) / \left(\mathbf{y}^T \mathbf{y}\right). \end{aligned}\right\}$$

Step 2. Find

$$[\mathrm{cerr}]_l = [\mathrm{cerr}]_l^{(j_l)} = \max\{[\mathrm{cerr}]_l^{(j)}, \ \ l \le j \le n_M\}.$$

Then the j_l-th column of $\mathbf{\Phi}^{(l-1)}$ is interchanged with the l-th column of $\mathbf{\Phi}^{(l-1)}$, the j_l-th column of $\mathbf{A}$ is interchanged with the l-th column of $\mathbf{A}$ up to the $(l-1)$-th row, and the j_l-th element of $\boldsymbol{\lambda}$ is interchanged with the l-th element of $\boldsymbol{\lambda}$. This effectively selects the j_l-th candidate as the l-th regressor in the subset model.

Step 3. Perform the orthogonalization as indicated in (27) to derive the l-th row of $\mathbf{A}$ and to transform $\mathbf{\Phi}^{(l-1)}$ into $\mathbf{\Phi}^{(l)}$. Calculate g_l and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (28).

The selection is terminated at the n_s stage when the criterion (19) is satisfied and this produces a subset model containing n_s significant regressors. The algorithm described here is in its standard form. A fast implementation can be adopted, as shown in [19], to reduce complexity.

References

[1] S. Chen and S.A. Billings, "Representation of non-linear systems: the NARMAX model," *Int. J. Control*, Vol.49, No.3, pp.1013–1032, 1989.

[2] S.A. Billings and S. Chen, "Extended model set, global data and threshold model identification of severely non-linear systems," *Int. J. Control*, Vol.50, No.5, pp.1897–1923, 1989.

[3] C.J. Harris, X. Hong and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*. Springer Verlag, 2002.

[4] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, Vol.50, No.5, pp.1873–1896, 1989.

[5] S. Chen, C.F.N. Cowan and P.M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.2, No.2, pp.302–309, 1991.

[6] A.E. Hoerl and R.W. Kennard, "Ridge regression: biased estimation for non-orthogonal problems," *Technometrics*, Vol.12, pp.55–67, 1970.

[7] C.M. Bishop, "Improving the generalisation properties of radial basis function neural networks," *Neural Computation*, Vol.3, No.4, pp.579–588, 1991.

[8] S. Chen, E.S. Chng and K. Alkadhimi, "Regularised orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Control*, Vol.64, No.5, pp.829–837 1996.

[9] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, Vol.4, No.3, pp.415–447, 1992.

[10] S. Chen, "Kernel-based data modelling using orthogonal least squares selection with local regularisation," in *Proc. 7th Annual Chinese Automation and Computer Science Conf. in U.K.* (Nottingham, U.K.), Sept.22, 2001, pp.27–30.

[11] S. Chen, "Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models," to be presented at *6th Int. Conf. Signal Processing* (Beijing, China), Aug.26-30, 2002.

[12] S. Chen, "Local regularization assisted orthogonal least squares regression," submitted to *IEEE Trans. Neural Networks*, 2001.

[13] A.C. Atkinson and A.N. Donev, *Optimum Experimental Designs*. Oxford: Clarendon Press, 1992.

[14] X. Hong and C.J. Harris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Trans. Neural Networks*, to appear, 2002.

[15] X. Hong and C.J. Harris, "Experimental design and model construction algorithms for radial basis function networks," submitted to *IEEE Trans. Neural Networks*, 2002

[16] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, Vol.AC-19, pp.716–723.

[17] I.J. Leontaritis and S.A. Billings, "Model selection and validation methods for non-linear systems," *Int. J. Control*, Vol.45, No.1, pp.311–341, 1987.

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

[19] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Processing*, Vol.43, No.7, pp.1713–1715, 1995.

Table 1: Comparison of modelling accuracy for simple scalar function modelling.

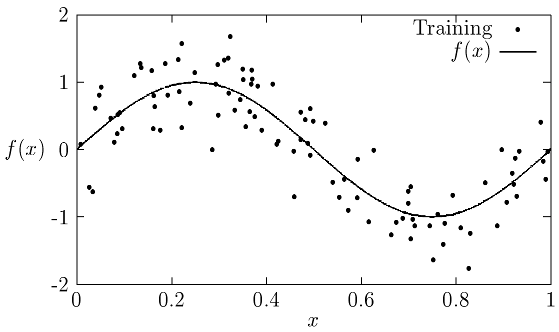
D-optimality	MSE over noise training data		MSE over noise-free testing data		number of terms	
weighting β	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
1e-8	0.15766	0.14743	0.00168	0.02138	6	15
1e-7	0.15766	0.14743	0.00168	0.02138	6	15
1e-6	0.15823	0.14743	0.00202	0.02138	6	15
1e-5	0.15705	0.14743	0.00194	0.02138	5	15
1e-4	0.15826	0.14761	0.00246	0.02068	5	15
1e-3	0.15705	0.14933	0.00194	0.01585	5	12
1e-2	0.15705	0.15560	0.00194	0.00423	5	6
1e-1	0.15911	0.15544	0.00223	0.00427	5	6

Table 2: Comparison of modelling accuracy for 2-dimensional simulated time series modelling.

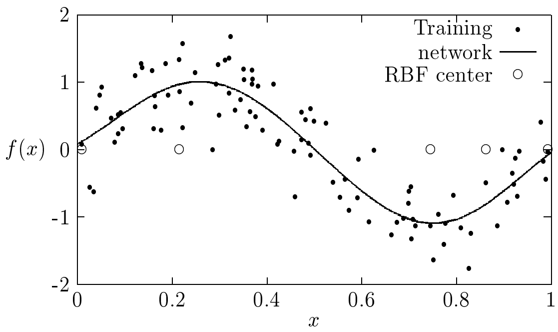
D-optimality weighting β	MSE over training data		MSE over testing data		number of terms	
	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
1e-6	0.09275	0.07764	0.09635	2.53132	19	94
1e-4	0.09311	0.07762	0.09607	0.41540	13	93
1e-2	0.09338	0.08966	0.09750	0.09379	13	25
1e+0	0.09395	0.09360	0.09667	0.09627	13	14

Table 3: Comparison of modelling accuracy for engine data set.

D-optimality weighting β	MSE over training data		MSE over testing data		number of terms	
	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt	LROLS + D-opt	OLS + D-opt
1e-8	0.000459	0.000336	0.000488	0.000872	22	60
1e-7	0.000442	0.000345	0.000484	0.000831	27	58
1e-6	0.000441	0.000345	0.000479	0.000838	25	57
1e-5	0.000452	0.000429	0.000499	0.000517	22	24
1e-4	0.000586	0.000445	0.000606	0.000497	20	22
1e-3	0.000478	0.000503	0.000501	0.000536	20	19
1e-2	0.000884	0.000883	0.000982	0.000987	16	16
1e-1	0.004951	0.004951	0.005050	0.005052	12	12



(a)



(b)

Figure 1: Simple scalar function modelling problem. (a): Noisy training data y (dots) and underlying function $f(x)$ (curve); and (b): model mapping (curve) produced by the LROLS + D-optimality algorithm with $\beta = 10^{-5}$, circles indicate the RBF centers.

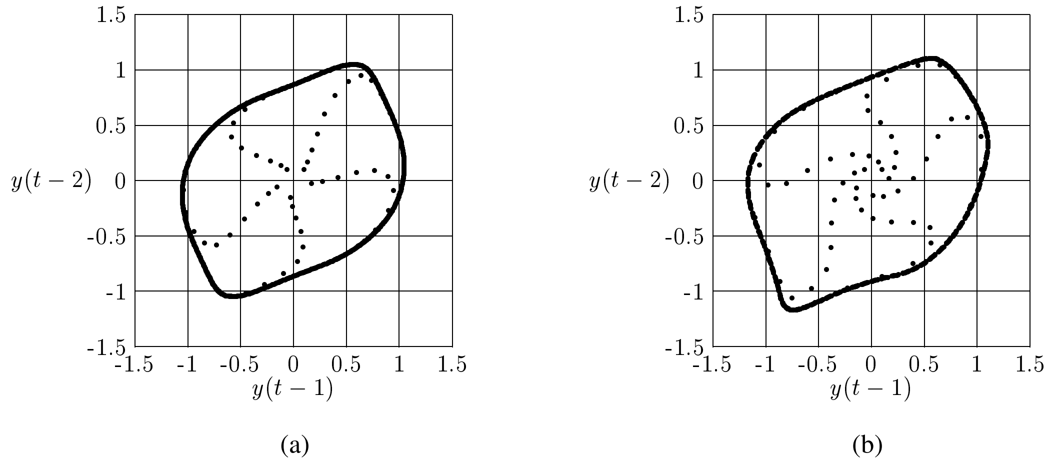


Figure 2: Two-dimensional time series modelling problem. (a): Phase plot of the noise-free time series ($y_d(0) = y_d(-1) = 0.1$); (b): Phase plot of the iterative RBF model output ($\hat{y}_d(0) = \hat{y}_d(-1) = 0.1$), the model was constructed by the LROLS + D-optimality algorithm with $\beta = 10^{-4}$.

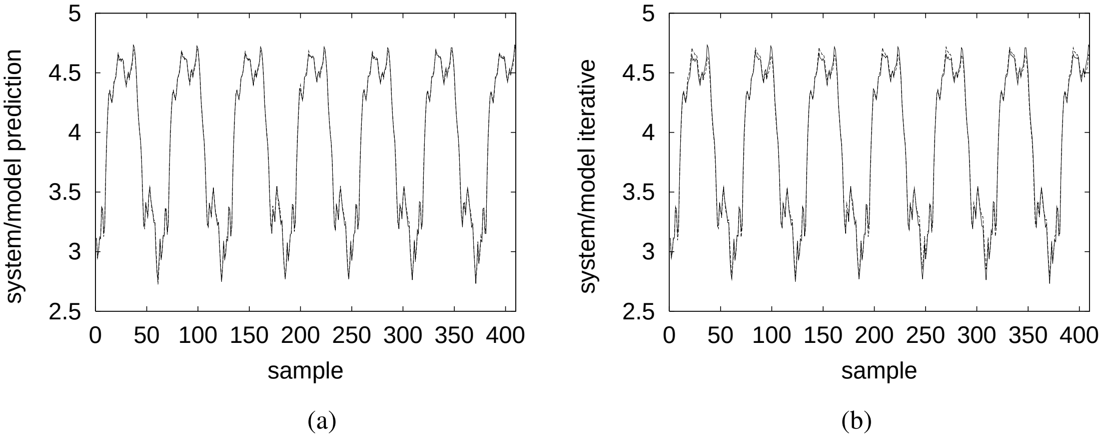


Figure 3: System output $y(k)$ (solid) superimposed on (a) model one-step prediction $\hat{y}(k)$ (dashed) and (b) model iterative output $\hat{y}_d(k)$ (dashed). The model was selected by the LROLS + D-optimality algorithm with $\beta = 10^{-5}$.