International Journal of Control
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713393989

Regularized orthogonal least squares algorithm for constructing radial basis function networks

S. Chen [a]; E. S. Chng [b]; K. Alkadhimi [a]
[a] Department of Electrical and Electronic Engineering, University of Portsmouth, Portsmouth, U.K.
[b] Department of Electrical Engineering, University of Edinburgh, Edinburgh, U.K.

Online Publication Date: 01 July 1996
To cite this Article: Chen, S., Chng, E. S. and Alkadhimi, K. (1996) 'Regularized orthogonal least squares algorithm for constructing radial basis function networks', International Journal of Control, 64:5, 829 - 837
To link to this article: DOI: 10.1080/00207179608921659
URL: http://dx.doi.org/10.1080/00207179608921659

INT. J. CONTROL, 1996, VOL. 64, NO. 5, 829–837

Regularized orthogonal least squares algorithm for constructing radial basis function networks

S. CHEN†, E. S. CHNG‡ and K. ALKADHIMI†

The paper presents a regularized orthogonal least squares learning algorithm for radial basis function networks. The proposed algorithm combines the advantages of both the orthogonal forward regression and regularization methods to provide an efficient and powerful procedure for constructing parsimonious network models that generalize well. Examples of nonlinear modelling and prediction are used to demonstrate better generalization performance of this regularized orthogonal least squares algorithm over the unregularized one.

1. Introduction

For practical purposes, it is desired to construct a small neural network. Apart from some obvious advantages, small models often generalize better. The orthogonal least squares (OLS) algorithm (Chen *et al.* 1991) is an efficient procedure for learning a parsimonious radial basis function (RBF) network. A simple mechanism can be built into the algorithm to avoid automatically any ill-conditioning of learning problems. For B-splines neural networks (Brown and Harris 1994), a learning procedure called ASMOD (Kavli 1993) has been developed for constructing parsimonious models. The parsimonious principle alone, however, is not entirely immune to overfitting. If data are highly noisy, small models constructed may still fit into noise. A technique for overcoming overfitting is regularization. This technique is usually applied to large full-size neural networks (Poggio and Girosi 1990, Bishop 1991).

Some researchers have combined regularization techniques with the parsimonious principle. For example, Barron and Xiao (1991) proposed a first-order regularized stepwise selection of subset regression models. A recent study (Orr 1993) has applied both the forward regression and zero-order regularization techniques to construct parsimonious RBF networks with improved generalization properties. The zero-order regularization is a technique equivalent to simple weight-decaying in gradient descent methods for multilayer perceptron neutral networks (Hertz *et al.* 1991). It is also known as the ridge regression in the statistical literature (Hoerl and Kennard 1970). Although these regularized subset selection algorithms are capable of choosing a small model with improved generalization properties, they require considerably more computation than the OLS algorithm.

This paper combines the zero-order regularization with the OLS algorithm to derive a regularized OLS (ROLS) algorithm for RBF networks. This new forward selection algorithm is capable of constructing small RBF networks which generalize

Received 18 December 1995. Revised 19 July 1995.

† Department of Electrical and Electronic Engineering, University of Portsmouth, Anglesea Building, Portsmouth PO1 3DJ, U.K.

‡ Department of Electrical Engineering, University of Edinburgh, King's Buildings, Edinburgh EH9 3JL, U.K.

0020-7179/96 $12·00

well. Furthermore, it has a similar computational requirement to that of the OLS algorithm and is, therefore, computationally very efficient. For the notational simplicity, RBF networks with a single output node are considered in this paper. However, the results can readily be applied to multi-output RBF networks (Chen *et al.* 1992). The effectiveness of the ROLS algorithm is demonstrated using a modelling and prediction application.

2. Formulation of linear regression model

Before describing the ROLS algorithm, we formulate the RBF network as a linear regression model. The RBF network with m inputs, n_H hidden nodes and a scalar output is defined by

$$f_r(\boldsymbol{x}) = \sum_{i=1}^{n_H} \theta_i \phi(\|\boldsymbol{x}-\boldsymbol{c}_i\|) \tag{1}$$

where $\boldsymbol{x} = [x_1 \dots x_m]^T$ is the input vector, θ_i are the weights, $\boldsymbol{c}_i = [c_{1,i} \dots c_{m,i}]^T$ are the RBF centres, $\|\cdot\|$ denotes the euclidean norm and $\phi(\cdot)$ is known as the nonlinearity of hidden nodes. Two examples of $\phi(\cdot)$ are the thin-plate-spline function $\phi(r) = r^2 \log(r)$ and the gaussian function $\phi(r) = \exp(-r^2/\rho^2)$, where $\rho > 0$ is a width parameter.

Assume that we have a training set of N samples $\{d(t), \boldsymbol{x}(t)\}_{t=1}^N$, where $d(t)$ is the target or desired network output corresponding to the network input vector $\boldsymbol{x}(t)$. Assume for the time being that we use every $\boldsymbol{x}(t)$ as a centre, that is, $\boldsymbol{c}_i = \boldsymbol{x}(i)$, for $1 \leqslant i \leqslant N$. Then the actual network outputs are

$$f_r(\boldsymbol{x}(t)) = \sum_{i=1}^{N} \theta_i \phi(\|\boldsymbol{x}(t)-\boldsymbol{c}_i\|), 1 \leqslant t \leqslant N \tag{2}$$

The model (2) may be referred to as the 'full' network model. By introducing the notation $\phi_i(t) = \phi(\|\boldsymbol{x}(t)-\boldsymbol{c}_i\|)$, we can express the desired output $d(t)$ as

$$d(t) = \sum_{i=1}^{N} \theta_i \phi_i(t) + e(t), 1 \leqslant t \leqslant N \tag{3}$$

where $e(t)$ is the error between $d(t)$ and $f_r(\boldsymbol{x}(t))$. By defining

$$\boldsymbol{d} = [d(1) \dots d(N)]^T \tag{4}$$

$$\boldsymbol{e} = [e(1) \dots e(N)]^T \tag{5}$$

$$\boldsymbol{\Theta} = [\theta_1 \dots \theta_N]^T \tag{6}$$

$$\boldsymbol{\Phi}_i = [\phi_i(1) \dots \phi_i(N)]^T \tag{7}$$

$$\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1 \dots \boldsymbol{\Phi}_N], \tag{8}$$

we can collect (3) for $1 \leqslant t \leqslant N$ together as

$$\boldsymbol{d} = \boldsymbol{\Phi}\boldsymbol{\Theta} + \boldsymbol{e} \tag{9}$$

Equation (9) has the form of a linear regression model, and the basis vectors $\boldsymbol{\Phi}_i$,

$1 \leqslant i \leqslant N$, can be referred to as regressors. In practice, it is often necessary to select a smaller subset of n_H centres or regressors from the full model (2).

3. The regularized OLS algorithm

The error criterion used in deriving the OLS algorithm (Chen *et al.* 1991) is the total squared error $e^T e$. The least squares criterion in certain circumstances is prone to overfitting. To prevent overfitting, regularization techniques can be applied. In the study of Orr (1993), a regularized forward selection (RFS) algorithm was derived by considering the zero-order regularized error criterion

$$e^T e + \lambda \Theta^T \Theta \tag{10}$$

where $\lambda \geqslant 0$ is the regularization parameter. The RFS algorithm selects one centre from the full model (9) at a time. Each selection is chosen to decrease maximally the regularized squared error (10). A drawback of this algorithm is that it cannot utilize an orthogonalization scheme and therefore requires considerably more computation than the OLS algorithm.

We can actually combine the zero-order regularization with the OLS algorithm to form an efficient procedure for subset selection. Let an orthogonal decomposition of the regression matrix Φ be

$$\Phi = WA \tag{11}$$

where

$$A = \begin{bmatrix} 1 & \alpha_{1,2} & \cdots & \alpha_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \alpha_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{12}$$

and

$$W = [w_1 \ldots w_N] \tag{13}$$

with orthogonal columns that satisfy

$$w_i^T w_j = 0, \quad \text{if } i \neq j \tag{14}$$

The full model (9) can be rewritten as

$$d = Wg + e \tag{15}$$

The orthogonal weight vector $g = [g_1 \ldots g_N]^T$ and the original weight vector Θ satisfy the triangular system

$$A\Theta = g \tag{16}$$

Knowing A and g, Θ can readily be solved from (16). The orthogonalization can be performed, for example, using the Gram–Schmidt or Householder transformation schemes.

The key to derive a computationally efficient ROLS scheme is to consider the following zero-order regularized error criterion

$$e^{\mathrm{T}}e+\lambda g^{\mathrm{T}}g \tag{17}$$

It is obvious that the criterion (17) is similar to the criterion (10) due to the relationship (16). In fact, the term $\lambda g^{\mathrm{T}}g$ penalizes large g_i, which is equivalent to penalizing large θ_i.

After some simple calculations, it can be shown that the regularized error criterion (17) can be expressed as

$$e^{\mathrm{T}}e+\lambda g^{\mathrm{T}}g = d^{\mathrm{T}}d-\sum_{i=1}^{N}(w_i^{\mathrm{T}}w_i+\lambda)g_i^2 \tag{18}$$

Normalizing (18) by $d^{\mathrm{T}}d$ yields

$$(e^{\mathrm{T}}e+\lambda g^{\mathrm{T}}g)/d^{\mathrm{T}}d = 1-\sum_{i=1}^{N}(w_i^{\mathrm{T}}w_i+\lambda)g_i^2/d^{\mathrm{T}}d \tag{19}$$

Similar to the case of the OLS algorithm (Chen *et al.* 1991), we can define the regularized error reduction ratio due to w_i as

$$[rerr]_i \equiv (w_i^{\mathrm{T}}w_i+\lambda)g_i^2/d^{\mathrm{T}}d \tag{20}$$

Based on this ratio, significant regressors can be selected in a forward-regression procedure exactly as in the case of the OLS algorithm (Chen *et al.* 1991). The selection is terminated at the n_{H}th stage when

$$1-\sum_{k=1}^{n_{\mathrm{H}}}[\mathrm{rerr}]_k < \xi \tag{21}$$

is satisfied, where $0<\xi<1$ is a chosen tolerance. This produces a subset network containing n_{H} significant regressors. The ROLS algorithm based on the modified Gram-Schmidt scheme is given in the Appendix.

It should be emphasized that the solution found by the ROLS algorithm is identical to that found by the RFS algorithm of Orr (1993). Both algorithms perform a subset model selection based on the forward search technique. The ROLS algorithm, however, requires considerably less computation than the RFS by exploiting some orthogonal properties. The forward selection is a suboptimal method and does not guarantee to find the optimal solution. To find the optimal n_{H}-term subset model from an N-term full model, it is required to calculate the performance of all possible n_{H}-term subset models and to choose the best one. This is computationally prohibitive even for a modest N and thus impractical. A subset model found using the forward selection technique is generally good enough for many practical applications.

4. Choice of regularization parameter

The appropriate value of λ is problem dependent (dependent on the underlying system that generates the training data and the choice of basis function $\phi(\cdot)$). How to choose a good value of λ has been addressed in the statistical literature (Hoerl and

Kennard 1970, Golub *et al.* 1979). A previous study using the second-order regularization (Bishop 1991) has suggested that the performance of the RBF network may be fairly insensitive to the precise value of λ.

An elegant approach to the selection of the regularization parameter is to adopt a Bayesian interpretation and to calculate the best value of regularization parameter using the evidence procedure (Mackay 1992). Applying this Bayesian approach to the ROLS algorithm results in the following iterative procedure for estimating λ. Given an initial guess of λ, the algorithm constructs a network model. This in turn allows an uptading of λ by the formula

$$\lambda = \frac{\gamma}{N-\gamma}\frac{e^{\mathrm{T}}e}{g^{\mathrm{T}}g} \tag{22}$$

where

$$\gamma = \sum_{i=1}^{n_{\mathrm{H}}} \frac{w_i^{\mathrm{T}} w_i}{w_i^{\mathrm{T}} w_i + \lambda} \tag{23}$$

is the number of good parameter measurements (Mackay 1992). After a few iterations, an appropriate λ value can be found.

5. Examples

In the first example, the RBF network with a gaussian basis function and a width $\rho = 0{\cdot}2$ is used to approximate the scalar function

$$f(x) = \sin(2\pi x), 0 \leqslant x \leqslant 1 \tag{24}$$

One hundred training data were generated from $f(x)+\varepsilon$, where x was taken from the uniform distribution in (0 1) and the noise ε had a gaussian distribution with zero mean and standard deviation 0·4. A separated test data set was also generated for $x = 0, 0{\cdot}01, \ldots, 0{\cdot}99, 1{\cdot}00$. The training data and the function $f(x)$ are plotted in Fig. 1. The training data set is highly ill-conditioned. The ROLS algorithm selected 15 centres from the training set.

Figure 2 depicts the mean square error (MSE) as a function of $\log_{10}(\lambda)$ for both the training and testing data sets. The optimal value of λ for this example is approximately 1·0. However, for a large range of λ values, the MSE over the testing set is quite flat, indicating that the performance of the ROLS algorithm is fairly insensitive to the precise value of λ in this large region. Figure 3 shows the network mapping constructed by the ROLS algorithm with $\lambda = 1{\cdot}0$. As a comparison, the network mapping constructed by the OLS algorithm is given in Fig. 4, where overfitting can be clearly seen.

In the second example, the RBF network with a thin-plate-spline basis function is used to predict the sunspot time series. The sunspot time series over the years 1700–1979 can be found in the Appendix A1 of Tong (1983). The data from 1700 to 1920 were used for training, and the multi-step predictions were then computed over the years 1921–1955 and the years 1921–1979 respectively. A RBF network of 25 centres was constructed using the OLS algorithm in a previous study (Chen 1994), and the predictive accuracy of the resulting RBF model was shown to be better than some

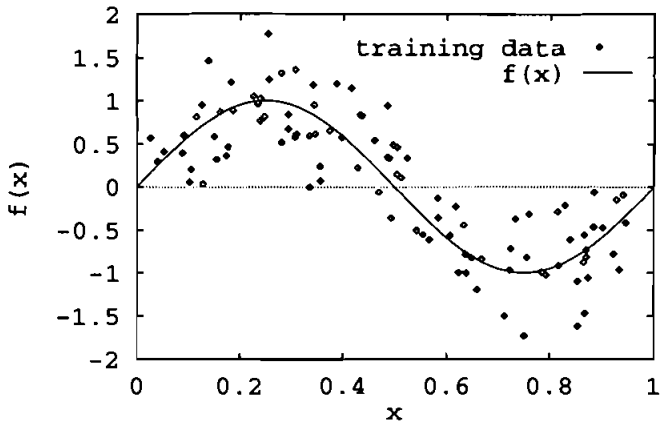

Figure 1. Noisy training data (points) and underlying function (curve).

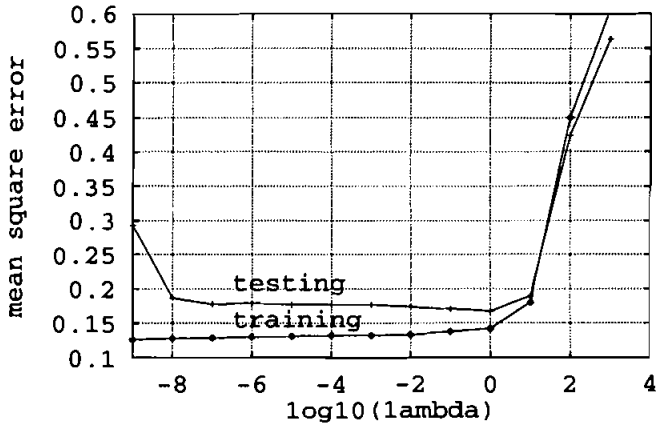

Figure 2. Mean square error as a function of the regularization parameter.

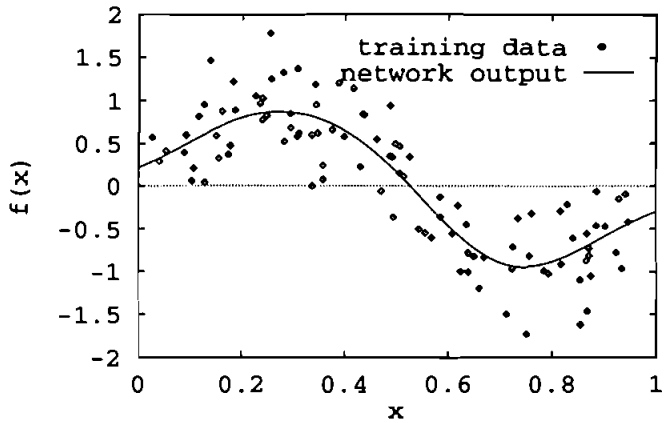

Figure 3. Network mapping constructed by the regularized orthogonal least squares algorithm.

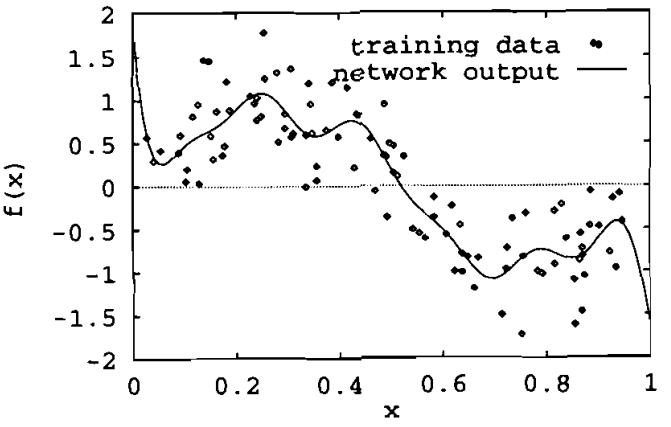

Figure 4. Network mapping constructed by the orthogonal least squares algorithm.

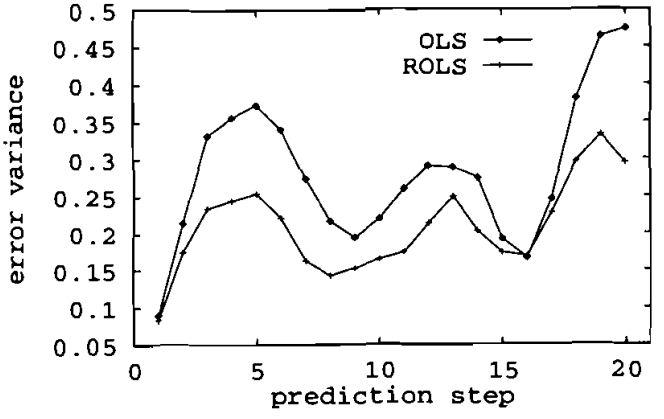

Figure 5. Normalized variances of multi-step prediction errors for the sunspot time series over years 1921–1955.

other nonlinear models fitted to the time series (Weigend *et al.* 1990, Chen and Billings 1989).

As a comparison, the ROLS algorithm was used to construct a RBF network of 25 centres based on the same full network model with $\lambda = 10^7$. Figure 5 compares the predictive performance of this model over the period 1921–1955 with that of the network constructed using the OLS algorithm. Predictive accuracies of the two network models obtained using the ROLS and OLS algorithms respectively over the period 1921–1979 are plotted in Fig. 6. The results shown in Figs 5 and 6 clearly demonstrate that the ROLS algorithm has better generalization properties.

The choices of the λ value were very different for the two examples. This is because the underlying data generating mechanisms were very different and different basis functions were used for the two examples. In fact, when the Bayesian approach mentioned in the previous section was used to estimate λ, the procedure converged approximately to 1·0 for the first example and very close to 10^7 for the second example.

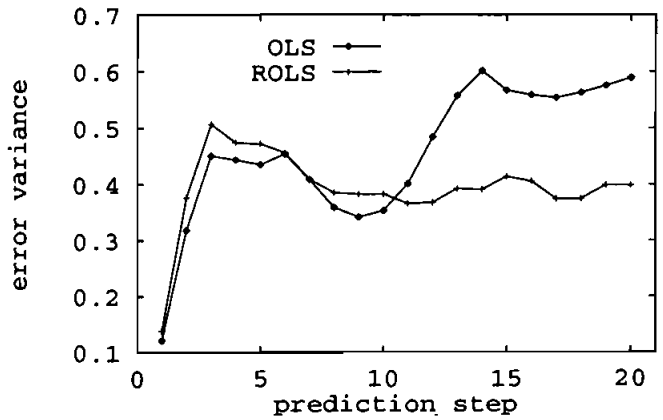

Figure 6. Normalized variances of multi-step prediction errors for the sunspot time series over years 1921–1979.

6. Conclusions

A very efficient learning algorithm for radial basis function networks has been derived by combining the orthogonal-least-squares forward selection and the zero-order regularization technique. This algorithm is capable of constructing parsimonious radial basis function networks which generalize well under severely noisy conditions. Although the method has been presented in the context of radial basis function networks, it can actually be applied to all the nonlinear models that have a linear-in-the-parameters structure, such as the fuzzy basis function network and the Volterra series model.

Appendix

The modified Gram–Schmidt orthogonal procedure calculates the A matrix row by row and orthogonalizes Φ as follows: at the kth stage make the columns Φ_j, $k+1 \leqslant j \leqslant N$, orthogonal to the kth column and repeat the operation for $1 \leqslant k \leqslant N-1$. Specifically, denoting $\Phi_j^{(0)} = \Phi_j$, $1 \leqslant j \leqslant N$, then

$$\left.\begin{aligned} w_k &= \Phi_k^{(k-1)} \\ \alpha_{k,j} &= w_k^{\mathrm{T}} \Phi_j^{(k-1)} / (w_k^{\mathrm{T}} w_k), k+1 \leqslant j \leqslant N \\ \Phi_j^{(k)} &= \Phi_j^{(k-1)} - \alpha_{k,j} w_k, k+1 \leqslant j \leqslant N \end{aligned}\right\} k = 1, 2, \ldots, N-1 \qquad \text{(A 1)}$$

The last stage of the procedure is simply $w_N = \Phi_N^{(N-1)}$. The elements of g are computed by transforming $d^{(0)} = d$ in a similar way

$$\left.\begin{aligned} g_k &= w_k^{\mathrm{T}} d^{(k-1)} / (w_k^{\mathrm{T}} w_k + \lambda) \\ d^{(k)} &= d^{(k-1)} - g_k w_k \end{aligned}\right\} 1 \leqslant k \leqslant N \qquad \text{(A 2)}$$

where $\lambda \geqslant 0$ is the regularization parameter.

This orthogonalization scheme can be used to derive a simple and efficient algorithm for selecting subset models. First introduce the definition of $\Phi^{(k-1)}$ as

$$\Phi^{(k-1)} = [w_1 \ldots w_{k-1} \Phi_k^{(k-1)} \ldots \Phi_N^{(k-1)}] \qquad \text{(A 3)}$$

If some of the columns $\Phi_k^{(k-1)}, \ldots, \Phi_N^{(k-1)}$ in $\Phi^{(k-1)}$ have been interchanged, this will still be referred to as $\Phi^{(k-1)}$ for notational convenience. The kth stage of the selection procedure is given as follows.

Step 1. For $k \leqslant j \leqslant N$, compute

$$\left.\begin{aligned} g_k^{(j)} &= (\boldsymbol{\Phi}_j^{(k-1)})^{\mathrm{T}} \boldsymbol{d}^{(k-1)} / ((\boldsymbol{\Phi}_j^{(k-1)})^{\mathrm{T}} \boldsymbol{\Phi}_j^{(k-1)} + \lambda) \\ [\mathrm{rerr}]_k^{(j)} &= (g_k^{(j)})^2 ((\boldsymbol{\Phi}_j^{(k-1)})^{\mathrm{T}} \boldsymbol{\Phi}_j^{(k-1)} + \lambda) / \boldsymbol{d}^{\mathrm{T}} \boldsymbol{d} \end{aligned}\right\}$$

Step 2. Find

$$[\mathrm{rerr}]_k = [\mathrm{rerr}]_k^{(j_k)} = \max\{[\mathrm{rerr}]_k^{(j)}, k \leqslant j \leqslant N\}$$

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

Step 3. Perform the orthogonalization as indicated in (A 1) to derive the kth row of $\boldsymbol{A}$ and to transform $\boldsymbol{\Phi}^{(k-1)}$ into $\boldsymbol{\Phi}^{(k)}$. $\boldsymbol{d}^{(k-1)}$ is then updated into $\boldsymbol{d}^{(k)}$ in the way shown in (A 2).

The selection is terminated at the n_{H}th stage when the criterion (21) is satisfied and this produces a subset model containing n_{H} significant regressors.

References

Barron, A. R., and Xiao, X., 1991, Discussion of multivariate adaptive regression splines. *Annals of Statistics*, **19**, 67–82.

Bishop, C., 1991, Improving the generalization properties of radial basis function neural networks. *Neural Computation*, **3**, 579–588.

Brown, M., and Harris, C. J., 1994, *Neurofuzzy Adaptive Modelling and Control* (Hemel Hempsted, U.K.: Prentice Hall).

Chen, S., 1994, Radial basis functions for signal prediction and system modelling. *Journal of Applied Science and Computations*, **1**.

Chen, S., and Billings, S. A., 1989, Modelling and analysis of non-linear time series. *International Journal of Control*, **50**, 2151–2171.

Chen, S., Cowan, C. F. N., and Grant, P. M., 1991, Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, **2**, 302–309.

Chen, S., Grant, P. M., and Cowan, C. F. N., 1992, Orthogonal least squares algorithm for training multi-output radial basis function networks. *Proceedings of the Institution of Electrical Engineers* Pt F, **139**, 378–384.

Golub, G. H., Heath, M., and Wahba, G., 1979, Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **1**, 215–223.

Hertz, J., Krough, A., and Palmer, R., 1991, *Introduction to the Theory of Neural Computation* (Redwood City, California, U.S.A.: Addison-Wesley).

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

Kavli, T., 1993, ASMOD: an algorithm for adaptive spline modelling of observation data. *International Journal of Control*, **58**, 947–968.

MacKay, D. J. C., 1992, Bayesian interpolation. *Neural Computation*, **4**, 415–447.

Orr, M. J. L., 1993, Regularised centre recruitment in radial basis function networks. Research Report, No. 59, Centre for Cognitive Science, University of Edinburgh, U.K.

Poggio, T., and Girosi, F., 1990, Networks for approximation and learning. *Proceedings of the Institution of Electrical and Electronics Engineers*, **78**, 1481–1497.

Tong, H., 1983, *Threshold Models in Non-linear Time Series Analysis* (New York: Springer-Verlag).

Weigend, A. S., Huberman, B. A., and Rumelhart, D. E., 1990, Predicting the future: a connectionist approach. *International Journal of Neural Systems*, **1**, 193–209.