Orthogonal Forward Selection for Constructing the Radial Basis Function Network with Tunable Nodes

Sheng Chen[1], Xia Hong[2], and Chris J. Harris[1]

[1] School of Electronics and Computer Science,
University of Southampton, Southampton SO17 1BJ, UK
{sqc, cjh}@ecs.soton.ac.uk
[2] Department of Cybernetics, University of Reading,
Reading RG6 6AY, UK
x.hong@reading.ac.uk

Abstract. An orthogonal forward selection (OFS) algorithm based on the leave-one-out (LOO) criterion is proposed for the construction of radial basis function (RBF) networks with tunable nodes. This OFS-LOO algorithm is computationally efficient and is capable of identifying parsimonious RBF networks that generalise well. Moreover, the proposed algorithm is fully automatic and the user does not need to specify a termination criterion for the construction process.

1 Introduction

The radial basis function (RBF) network is a popular artificial neural network structure that has found wide applications in machine learning and engineering. The parameters of the RBF network include its centre vectors and variances of the basis functions as well as the weights that connect the RBF nodes to its output node. The parameters of a RBF network can be learned via nonlinear optimisation using the gradient based algorithm [1], the evolutionary algorithm [2] or the E-M algorithm [3]. Such a nonlinear learning approach is computationally expensive and may encounter the local minima problem. Additionally, the network structure or the number of RBF nodes has to be determined via other means. Alternatively, clustering algorithms can be applied to find the RBF centre vectors as well as the associated basis function variances [4]-[6]. This leaves the RBF weights to be determined by the usual linear least squares solution. Again, the number of the clusters has to be determined via other means, such as cross validation.

A popular approach for constructing RBF networks is to formulate the problem as a linear learning one by considering the training input data points as candidate RBF centres and employing a common variance for every RBF node. A parsimonious RBF network is then identified using the efficient orthogonal least squares (OLS) algorithm [7]-[10]. Similarly, the support vector machine (SVM) and other sparse kernel modelling methods [11]-[17] also fit the kernel centres to the training input data points and adopt a common variance for every kernels. A sparse kernel representation is then sought. Since the common variance is not provided by the learning algorithm, it has to be determined via cross validation. In a recent work [10], a locally regularised OLS

D.S. Huang, X.-P. Zhang, G.-B. Huang (Eds.): ICIC 2005, Part I, LNCS 3644, pp. 777–786, 2005.

(LROLS) algorithm based on the leave-one-out (LOO) mean square error criterion has been proposed, which compares favourably with other existing state-of-the-art sparse kernel modelling methods, in terms of model sparisty and generalisation performance.

This paper proposes an efficient construction algorithm for the RBF network with tunable nodes. In this approach, each RBF node has a tunable centre vector and a tunable diagonal covariance matrix, and an orthogonal forward selection (OFS) procedure is adopted to append the RBF nodes one by one by incrementally minimising the LOO criterion. Because the RBF centres are not restricted to the training input points and each node has an individually adjusted covariance matrix, the proposed OFS-LOO algorithm can produce sparser representations with excellent generalisation capability, in comparison with the existing sparse RBF or kernel modelling methods. Efficiency of the proposed algorithm is ensured because of the orthogonalisation procedure. Furthermore, the construction process is fully automatic and there is no need for the user to specify any additional termination criterion.

2 Construction of the RBF Network with Tunable Nodes

Consider the regression modelling problem of approximating the N pairs of training data, $\{(\mathbf{x}_k, y_k)\}_{k=1}^N$, with the RBF network defined in

$$y_k = \hat{y}_k + e_k = \sum_{i=1}^{M} w_i g_i(\mathbf{x}_k) + e_k = \mathbf{g}^T(k)\mathbf{w} + e_k \tag{1}$$

where $\mathbf{x}_k \in \mathcal{R}^m$, $\hat{y}_k$ denotes the RBF model output, $e_k = y_k - \hat{y}_k$ is the modelling error, M is the number of RBF nodes, $\mathbf{w} = [w_1\ w_2 \cdots w_M]^T$ is the RBF weight vector, $g_i(\bullet)$ for $1 \le i \le M$ denote the RBF regressors, and $\mathbf{g}(k) = [g_1(\mathbf{x}_k)\ g_2(\mathbf{x}_k) \cdots g_M(\mathbf{x}_k)]^T$. We will consider the general RBF regressor of the form

$$g_i(\mathbf{x}) = K\left(\sqrt{(\mathbf{x}-\boldsymbol{\mu}_i)^T\, \boldsymbol{\Sigma}_i^{-1}\, (\mathbf{x}-\boldsymbol{\mu}_i)}\right) \tag{2}$$

where $\boldsymbol{\mu}_i$ is the centre vector of the ith RBF unit, the diagonal covariance matrix has the form $\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \cdots, \sigma_{i,m}^2\}$, and $K(\bullet)$ is the chosen RBF or kernel function. By defining $\mathbf{y} = [y_1\ y_2 \cdots y_N]^T$, $\mathbf{e} = [e_1\ e_2 \cdots e_N]^T$, and $\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2 \cdots \mathbf{g}_M]$ with

$$\mathbf{g}_k = [g_k(\mathbf{x}_1)\ g_k(\mathbf{x}_2) \cdots g_k(\mathbf{x}_N)]^T,\ 1 \le k \le M \tag{3}$$

the regression model (1) over the training data set can be written in the matrix form

$$\mathbf{y} = \mathbf{G}\mathbf{w} + \mathbf{e} \tag{4}$$

Note that $\mathbf{g}_k$ denotes the kth column of $\mathbf{G}$ while $\mathbf{g}^T(k)$ is the kth row of $\mathbf{G}$.

Let an orthogonal decomposition of the regression matrix $\mathbf{G}$ be $\mathbf{G} = \mathbf{P}\mathbf{A}$, where $\mathbf{A}$ is the upper triangular matrix with unity diagonal elements and $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_M]$ with the orthogonal columns that satisfy $\mathbf{p}_i^T \mathbf{p}_j = 0$, if $i \ne j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \mathbf{P}\boldsymbol{\theta} + \mathbf{e} \tag{5}$$

where the weight vector $\boldsymbol{\theta} = [\theta_1\ \theta_2 \cdots \theta_M]^T$ in the orthogonal model space satisfies the triangular system $\mathbf{Aw} = \boldsymbol{\theta}$. Since the space spanned by the original model bases $g_i(\bullet)$, $1 \leq i \leq M$, is identical to the space spanned by the orthogonal model bases, the RBF model output is equivalently expressed by

$$\hat{y}_k = \mathbf{p}^T(k)\boldsymbol{\theta} \tag{6}$$

where $\mathbf{p}^T(k) = [p_1(k)\ p_2(k) \cdots p_M(k)]$ is the kth row of $\mathbf{P}$.

2.1 Orthogonal Forward Selection Based on the Leave-One-Out Criterion

The LOO mean square error is a measure of the model generalisation capability [10]. For the n-term RBF model, the LOO criterion is defined as

$$J_n = \frac{1}{N}\sum_{i=1}^{N}\left(e_i^{(n,-i)}\right)^2 = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{e_i^{(n)}}{\eta_i^{(n)}}\right)^2 \tag{7}$$

where $e_i^{(n,-i)}$ denotes the LOO modelling error of the n-term model, $e_i^{(n)}$ the usual n-term modelling error, and $\eta_i^{(n)}$ the LOO modelling error weighting. Note that $e_k^{(n)}$ and $\eta_k^{(n)}$ can be computed recursively using

$$e_k^{(n)} = y_k - \sum_{i=1}^{n}\theta_i p_i(k) = e_k^{(n-1)} - \theta_n p_n(k) \tag{8}$$

and

$$\eta_k^{(n)} = 1 - \sum_{i=1}^{n}\frac{p_i^2(k)}{\mathbf{p}_i^T\mathbf{p}_i + \lambda} = \eta_k^{(n-1)} - \frac{p_n^2(k)}{\mathbf{p}_n^T\mathbf{p}_n + \lambda} \tag{9}$$

respectively, where $\lambda \geq 0$ is a small regularisation parameter. Therefore, the computation of the LOO criterion J_n is very efficient.

The proposed OFS-LOO algorithm appends the RBF nodes one by one by incrementally minimising the LOO criterion J_n. Specifically, at the nth stage of the construction procedure, the nth RBF node is determined by minimising J_n with respect to the node's centre vector $\boldsymbol{\mu}_n$ and diagonal covariance matrix $\boldsymbol{\Sigma}_n$

$$\min_{\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n} J_n\left(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n\right) \tag{10}$$

The construction procedure is automatically terminated if $J_M \leq J_{M+1}$, yielding an M-term RBF network. Note that the LOO criterion J_n is at least locally convex and such an M exists [10]. After the OFS-LOO model construction, the LROLS-LOO algorithm of [10] can be applied to further reduce the model size and to automatically update regularisation parameters. Note that the refinement involving the LROLS-LOO requires a minimal computation, as the selected model size M is typically very small.

2.2 Positioning and Shaping a RBF Node

The task at the nth stage of the model construction is to position and shape the nth RBF node by solving the optimisation problem (10). Since this optimisation problem is non-convex, a gradient-based algorithm may become trapped at a local minimum. We adopt a global search algorithm called the repeated weighted boosting search (RWBS) [18] to determine $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$. The algorithm is summarised as follows. Let $\mathbf{u}$ be the vector that contains $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$. Give the following initial conditions:

$$e_k^{(0)} = y_k \text{ and } \eta_k^{(0)} = 1,\ 1 \le k \le N, \text{ and } J_0 = \frac{1}{N}\mathbf{y}^T\mathbf{y} = \frac{1}{N}\sum_{k=1}^{N} y_k^2 \tag{11}$$

Specify the following algorithmic parameters: P_S – population size, N_G – number of generations in the repeated search, and ξ_B – accuracy for terminating the weighted boosting search.

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

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

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

1. For $1 \le i \le P_S$, generate $\mathbf{g}_n^{i)}$ from $\mathbf{u}_i^{[l]}$, the candidates for the nth model column, and orthogonalise them:

$$\alpha_{j,n}^{i)} = \frac{\mathbf{p}_j^T\mathbf{g}_n^{i)}}{\mathbf{p}_j^T\mathbf{p}_j},\ 1 \le j < n \tag{12}$$

$$\mathbf{p}_n^{i)} = \mathbf{g}_n^{i)} - \sum_{j=1}^{n-1}\alpha_{j,n}^{i)}\mathbf{p}_j \tag{13}$$

$$\theta_n^{i)} = \frac{\left(\mathbf{p}_n^{i)}\right)^T\mathbf{y}}{\left(\mathbf{p}_n^{i)}\right)^T\mathbf{p}_n^{i)} + \lambda} \tag{14}$$

2. For $1 \le i \le P_S$, calculate the LOO cost function value of each $\mathbf{u}_i^{[l]}$:

$$e_k^{(n)}(i) = e_k^{(n-1)} - p_n^{i)}(k)\theta_n^{i)},\ 1 \le k \le N \tag{15}$$

$$\eta_k^{(n)}(i) = \eta_k^{(n-1)} - \frac{\left(p_n^{i)}(k)\right)^2}{\left(\mathbf{p}_n^{i)}\right)^T\mathbf{p}_n^{i)} + \lambda},\ 1 \le k \le N \tag{16}$$

$$J_n^{i)} = \frac{1}{N}\sum_{k=1}^{N}\left(\frac{e_k^{(n)}(i)}{\eta_k^{(n)}(i)}\right)^2 \tag{17}$$

where $p_n^{i)}(k)$ is the kth element of $\mathbf{p}_n^{i)}$.

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

Step 1: Boosting

1. Find
$$i_{\text{best}} = \arg \min_{1 \le i \le P_S} J_n^{i)} \quad \text{and} \quad i_{\text{worst}} = \arg \max_{1 \le i \le P_S} J_n^{i)}$$
Denote $\mathbf{u}_{\text{best}}^{[l]} = \mathbf{u}_{i_{\text{best}}}^{[l]}$ and $\mathbf{u}_{\text{worst}}^{[l]} = \mathbf{u}_{i_{\text{worst}}}^{[l]}$.
2. Normalise the cost function values
$$\bar{J}_n^{i)} = \frac{J_n^{i)}}{\sum_{m=1}^{P_S} J_n^{m)}}, \ 1 \le i \le P_S$$
3. Compute a weighting factor β_t according to
$$\xi_t = \sum_{i=1}^{P_S} \delta_i(t-1) \bar{J}_n^{i)}, \ \beta_t = \frac{\xi_t}{1 - \xi_t}$$
4. Update the distribution weightings for $1 \le i \le P_S$
$$\delta_i(t) = \begin{cases} \delta_i(t-1) \beta_t^{\bar{J}_n^{i)}}, & \text{for } \beta_t \le 1 \\ \delta_i(t-1) \beta_t^{1 - \bar{J}_n^{i)}}, & \text{for } \beta_t > 1 \end{cases}$$
and normalise them
$$\delta_i(t) = \frac{\delta_i(t)}{\sum_{m=1}^{P_S} \delta_m(t)}, \ 1 \le i \le P_S$$

Step 2: Parameter updating

1. Construct the $(P_S + 1)$th point using the formula
$$\mathbf{u}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i(t) \mathbf{u}_i^{[l]}$$
2. Construct the $(P_S + 2)$th point using the formula
$$\mathbf{u}_{P_S+2} = \mathbf{u}_{\text{best}}^{[l]} + \left(\mathbf{u}_{\text{best}}^{[l]} - \mathbf{u}_{P_S+1} \right)$$
3. Calculate $\mathbf{g}_n^{P_S+1)}$ and $\mathbf{g}_n^{P_S+2)}$ from $\mathbf{u}_{P_S+1}$ and $\mathbf{u}_{P_S+2}$, orthogonalise these two candidate model columns (as in (12) to (14)), and compute their corresponding LOO cost function values $J_n^{i)}$, $i = P_S + 1, P_S + 2$ (as in (15) to (17)). Then find
$$i_* = \arg \min_{i = P_S+1, P_S+2} J_n^{i)}$$
The pair $(\mathbf{u}_{i_*}, J_n^{i_*)})$ then replaces $(\mathbf{u}_{\text{worst}}^{[l]}, J_n^{i_{\text{worst}})})$ in the population

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

End of inner loop

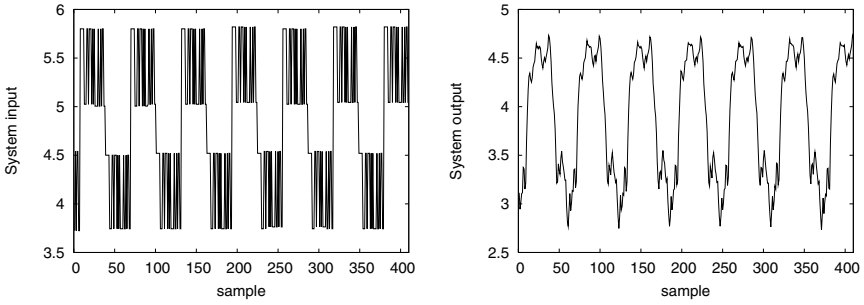
The solution found in the lth generation is $\mathbf{u} = \mathbf{u}_{\text{best}}^{[l]}$.

End of outer loop

This yields the solution $\mathbf{u} = \mathbf{u}_{\text{best}}^{[N_G]}$, i.e. $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ of the nth RBF node, the nth model column $\mathbf{g}_n$, the orthogonalisation coefficients $\alpha_{j,n}$, $1 \leq j < n$, the corresponding orthogonal model column $\mathbf{p}_n$, and the weight θ_n, as well as the n-term modelling errors $e_k^{(n)}$ and associated LOO modelling error weightings $\eta_k^{(n)}$ for $1 \leq k \leq N$.

3 Modelling Examples

Example 1. The engine data set [19] was used to demonstrate the effectiveness of the proposed OFS-LOO algorithm. The data were collected from a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed, where the input $u(t)$ was the fuel rack position and the output $y(t)$ was the engine speed. The input-output data set, depicted in Fig. 1, contained 410 samples. The first 210 data points were used in training and the last 200 points in model validation. The previous study



(a) (b)

Fig. 1. The engine data set: (a) input $u(t)$ and (b) output $y(t)$

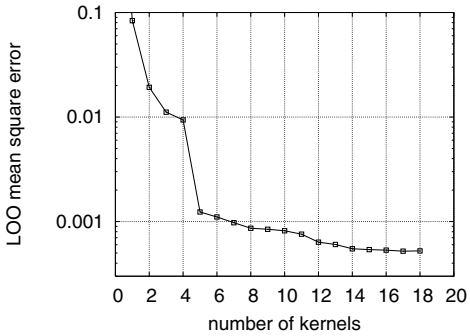
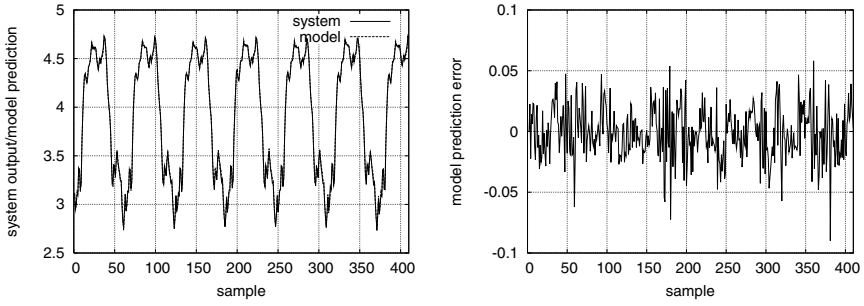


Fig. 2. The LOO mean square error as a function of the model size for the engine data set

Table 1. Comparison of the three models obtained by the SVM, LROLS-LOO and OFS-LOO algorithms for the engine data set

algorithm	RBF type	model size	MSE over training set	MSE over test set
SVM	fixed Gaussian	92	0.000447	0.000498
LROLS-LOO	fixed Gaussian	22	0.000453	0.000490
OFS-LOO	tunable Gaussian	15	0.000466	0.000480



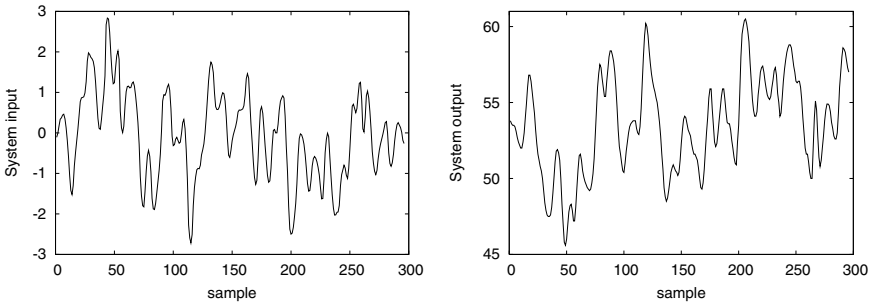
(a) (b)

Fig. 3. Modelling performance for the engine data set by the 15-node RBF network constructed by the OFS-LOO algorithm: (a) the model output $\hat{y}_k$ superimposed on the system output y_k, and (b) the modelling error $e_k = y_k - \hat{y}_k$

[9],[10] has shown that this data set can be modelled adequately as $y_i = f_s(\mathbf{x}_i) + e_i$ with $y_i = y(i)$, $\mathbf{x}_i = [y(i-1)\ u(i-1)\ u(i-2)]^T$, where $f_s(\bullet)$ describes the unknown underlying system to be identified and e_i denotes the system noise.

In the work [10], various state-of-the-art RBF and kernel modelling techniques were applied to construct Gaussian RBF network models for this data set, and the LROLS-LOO algorithm produced the best result. We applied the proposed OFS-LOO technique to this data set. Fig. 2 depicts the LOO mean square error (MSE) as a function of the model size during the modelling process using the OFS-LOO. It can be seen that the algorithm automatically constructed a 17-term RBF model, since $J_{18} > J_{17}$. The LROLS-LOO algorithm was then employed to further simplify this obtained model, yielding a final 15-term RBF network. This 15-term model is compared with the model quoted from [10], which was obtained purely by the LROLS-LOO method, in Table 1. As a comparison, the model obtained by the SVM algorithm is also listed in Table 1. Fig. 3 illustrates the modelling performance of the 15-node RBF network constructed by the OFS-LOO algorithm.

Example 2. This example constructed a model for the gas furnace data set (Series J in [20]). The data set, depicted in Fig. 4, contained 296 pairs of input-output points. 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, and the input vector was defined as $\mathbf{x}_k = [y_{k-1}\ y_{k-2}\ y_{k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}]^T$. In the



(a) (b)

Fig. 4. The gas furnace data set: (a) input $u(t)$ and (b) output $y(t)$

Table 2. Comparison of the three models obtained by the SVM, LROLS-LOO and OFS-LOO algorithms for the gas furnace data set

algorithm	RBF type	model size	training MSE	LOO MSE
SVM	fixed Gaussian	62	0.052416	0.054376
LROLS-LOO	fixed thin-plate-spline	28	0.053306	0.053685
OFS-LOO	tunable Gaussian	15	0.054306	0.054306

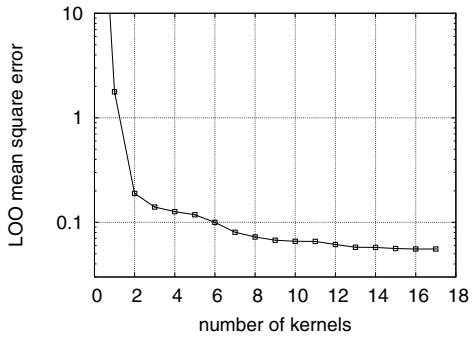
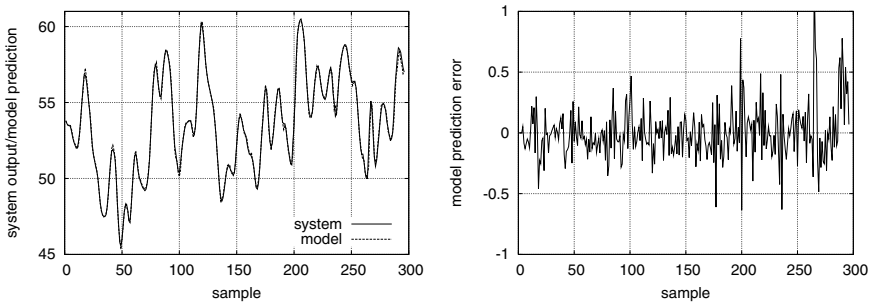


Fig. 5. The LOO mean square error as a function of the model size for the gas furnace data set

study [10], several existing RBF modelling techniques were applied to this data set using the thin-plate-spline basis functions defined by

$$K(\|\mathbf{x}-\mathbf{x}_i\|) = \|\mathbf{x}-\mathbf{x}_i\|^2 \log(\|\mathbf{x}-\mathbf{x}_i\|),\ 1 \le i \le N, \tag{18}$$

and the best result was obtained by the LROLS-LOO algorithm. The RBF network constructed by the LROLS-LOO algorithm is given in Table 2, where the LOO MSE was used to indicate the model generalization performance since there was no test data set. We also applied the SVM algorithm to fit a RBF network with the Gaussian basis function to this data set and the resulting model is also listed in Table 2.



(a) (b)

Fig. 6. Modelling performance for the gas furnace data set by the 15-node RBF network constructed by the OFS-LOO algorithm: (a) the model output $\hat{y}_k$ superimposed on the system output y_k, and (b) the modelling error $e_k = y_k - \hat{y}_k$

We applied the proposed OFS-LOO technique to this data set. Fig. 5 depicts the LOO MSE as a function of the model size during the modelling process using the OFS-LOO. It can be seen that the algorithm automatically constructed a 16-term RBF model, since $J_{17} \geq J_{16}$. The LROLS-LOO algorithm was then employed to further simplify this obtained model, yielding a final 15-term RBF network. This 15-term model is compared with the two models obtained by the SVM and LROLS-LOO algorithms, in Table 2. Fig. 6 illustrates the modelling performance of this 15-node RBF network constructed by the OFS-LOO algorithm.

4 Conclusions

A novel construction algorithm has been proposed for RBF networks with tunable nodes. Unlike most of the sparse RBF or kernel modelling methods, the RBF centres are not restricted to the training input data points and each node has an individually adjusted diagonal covariance matrix. The proposed OFS-LOO method appends the RBF nodes one by one by incrementally minimising the LOO mean square error. This construction process is computationally efficient due to the orthogonalisation procedure employed. Moreover, the model construction is fully automatic and the user does not need to specify a termination criterion.

Acknowledgements

S. Chen wish to thank the support of the United Kingdom Royal Academy of Engineering.

References

1. An, P.E., Brown, M., Chen, S., Harris, C.J.: Comparative Aspects of Neural Network Algorithms for On-Line Modelling of Dynamic Processes. Proc. I. MECH. E., Pt.I, J. Systems and Control Eng. **207** (1993) 223–241

2. Whitehead, B.A., Choate, T.D.: Evolving Space-Filling Curves to Distribute Radial Basis Functions Over an Input Space. IEEE Trans. Neural Networks **5** (1994) 15–23
3. Yang, Z.R., Chen, S.: Robust Maximum Likelihood Training of Heteroscedastic Probabilistic Neural Networks. Neural Networks **11** (1998) 739–747
4. Moody, J., Darken, C.J.: Fast Learning in Networks of Locally-Tuned Processing Units. Neural Computation **1** (1989) 281–294
5. Chen, S., Billings, S.A., Grant, P.M.: Recursive Hybrid Algorithm for Non-linear System Identification Using Radial Basis Function Networks. Int. J. Control **55** (1992) 1051–1070
6. Chen, S.: Nonlinear Time Series Modelling and Prediction Using Gaussian RBF Networks with Enhanced Clustering and RLS Learning. Electronics Letters **31** (1995) 117–118
7. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Trans. Neural Networks **2** (1991) 302–309
8. Chen, S., Wu, Y., Luk, B.L.: Combined Genetic Algorithm Optimisation and Regularised Orthogonal Least Squares Learning for Radial Basis Function Networks. IEEE Trans. Neural Networks **10** (1999) 1239–1243
9. Chen, S., Hong, X., Harris, C.J.: Sparse Kernel Regression Modelling Using Combined Locally Regularized Orthogonal Least Squares and D-Optimality Experimental Design. IEEE Trans. Automatic Control **48** (2003) 1029–1036
10. Chen, S., Hong, X., Harris, C.J., Sharkey, P.M.: Sparse Modelling Using Orthogonal Forward Regression with PRESS Statistic and Regularization. IEEE Trans. Systems, Man and Cybernetics, Part B **34** (2004) 898–911
11. Vapnik, V.: The Nature of Statistical Learning Theory. Springer-Verlag, New York (1995)
12. Gunn, S.: Support Vector Machines for Classification and Regression. Technical Report, ISIS Research Group, Department of Electronics and Computer Science, University of Southampton, UK (1998)
13. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic Decomposition by Basis Pursuit. SIAM Review **43** (2001) 129–159
14. Tipping, M.E.: Sparse Bayesian Learning and the Relevance Vector Machine. J. Machine Learning Research **1** (2001) 211–244
15. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA (2002)
16. Vincent, P., Bengio, Y.: Kernel Matching Pursuit. Machine Learning **48** (2002) 165–187
17. Lanckriet, G.R.G., Cristianini, N., Bartlett, P., Ghaoui, L.E., Jordan, M.I.: Learning the Kernel Matrix with Semidefinite Programming. J. Machine Learning Research **5** (2004) 27–72
18. Chen, S., Wang, X.X., Harris, C.J.,: Experiments with Repeating Weighted Boosting Search for Optimization in Signal Processing Applications. IEEE Trans. Systems, Man and Cybernetics, Part B **35** (2005) 682–693
19. Billings, S.A., Chen, S., Backhouse, R.J.: The Identification of Linear and Non-linear Models of a Turbocharged Automotive Diesel Engine. Mechanical Systems and Signal Processing **3** (1989) 123–142
20. Box, G.E.P., Jenkins, G.M.: Time Series Analysis, Forecasting and Control. Holden Day Inc. (1976)