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

Sheng Chen[†], Xia Hong[‡] and Chris J. Harris[†]

[†] School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.
E-mails: {sqc,cjh}@ecs.soton.ac.uk

[‡] Department of Cybernetics
University of Reading, Reading RG6 6AY, U.K.
E-mail: x.hong@reading.ac.uk

Presented at 2005 International Conference on Intelligent Computing
August 23-26, 2005, Hefei, China

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

Overview

RBF network has found wide applications in machine learning and engineering

○ Nonlinear optimisation to determine all basis centres, variances and weights

Local minimum and structure determination problems

○ Clustering to determine basis centres and variances

Structure determination problem

○ Orthogonal least squares and sparse kernel modelling

Select centres from data points, cross validation for single common basis variance

What's new. Combining OLS / nonlinear optimisation: OFS to construct RBF nodes one by one, each selected node is determined by nonlinear optimisation

RBF Network

- RBF network modelling of training data $\{(\mathbf{x}_k, y_k)\}_{k=1}^N$

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

M: number of RBF nodes,
$\mathbf{w} = [w_1 \; w_2 \cdots w_M]^T$: RBF weights
$\mathbf{g}(k) = [g_1(\mathbf{x}_k) \; g_2(\mathbf{x}_k) \cdots g_M(\mathbf{x}_k)]^T$: RBF nodes or regressors

- Generic RBF node

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

$\boldsymbol{\mu}_i$: ith RBF centre
$\boldsymbol{\Sigma}_i = \text{diag}\{\sigma_{i,1}^2, \cdots, \sigma_{i,m}^2\}$: diagonal covariance matrix of ith node
$K(\bullet)$: RBF or kernel function.

Learning

- *Learning*: determining number of nodes M, values of all $\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$ and w_i

- *Criterion*: should be model generalisation capability rather than training performance

  Leave-one-out (LOO) criterion is a measure of generalisation

- *State-of-the-art*: locally regularised orthogonal least squares with leave-one-out (LROLS-LOO)

  S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, 34 (2), 898–911, 2004

  Select centres from training input points and adopt a single common variance for every node

- *What's new*: extend to tunable nodes

  Centres not restricted to training input points and each node has a diagonal covariance matrix

  Orthogonal forward selection with leave-one-out (OFS-LOO)

Orthogonal Decomposition

○ RBF model over training set

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

where $\mathbf{G} = [\mathbf{g}_1\ \mathbf{g}_2 \cdots \mathbf{g}_M]$ is regression matrix

○ Orthogonal decomposition

$$\mathbf{G} = \mathbf{P}\mathbf{A}$$

where orthogonal matrix $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_M]$ has orthogonal columns

○ Regression model becomes

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

with $\boldsymbol{\theta} = [\theta_1\ \theta_2 \cdots \theta_M]^T = \mathbf{A}\,\mathbf{w}$

○ Space spanned by original model bases is identical to space spanned by orthogonal model bases

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

Notations: $\mathbf{g}_k$ is kth column of $\mathbf{G}$ while $\mathbf{g}^T(k)$ is kth row of $\mathbf{G}$; $\mathbf{p}_k$ is kth column of $\mathbf{P}$ while $\mathbf{p}^T(k)$ is kth row of $\mathbf{P}$

Leave-One-Out Criterion

○ LOO mean square error for n-term RBF model

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

$e_i^{(n,-i)}$: LOO modelling error, $e_i^{(n)}$: usual modelling error, $\eta_i^{(n)}$: LOO weighting

○ Computation of LOO criterion J_n is very efficient, since

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

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

where $\lambda \geq 0$ is a small regularisation parameter

OFS with LOO Criterion

○ OFS-LOO algorithm constructs RBF nodes one by one: at nth stage determine nth RBF node by minimising J_n

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

○ J_n is at least locally convex:

There exists M such that $J_{n-1} > J_n$ if $n \leq M$ and $J_M \leq J_{M+1}$

Construction procedure is automatically terminated, and user does not need to specify any learning algorithmic parameter

○ After OFS-LOO construction, LROLS-LOO algorithm is used to automatically optimise regularisation parameters and to further reduce model size M

Positioning and shaping a RBF node

○ Determine nth RBF centre $\boldsymbol{\mu}_n$ and covariance matrix $\boldsymbol{\Sigma}_n$ by minimising LOO criterion $J_n(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$ is a non-convex nonlinear optimisation problem

Gradient-based techniques may become trapped at a local minimum

Global optimisation techniques are preferred, e.g. genetic algorithms

○ We adopt a global search algorithm called the repeated weighted boosting search (RWBS)

S. Chen, X.X. Wang and C.J. Harris, "Experiments with repeating weighted boosting search for optimization in signal processing applications," *IEEE Trans. Systems, Man and Cybernetics, Part B*, 35 (4), 682–693, 2005

RWBS is a very simple but effective global optimisation search algorithm

Repeated Weighted Boosting Search

Consider task of minimising $J(\mathbf{u})$

Outer Loop: N_G number of generations

Initialisation: keep best solution found in previous generation as $\mathbf{u}_1$ and randomly choose rest of population $\mathbf{u}_2, \cdots, \mathbf{u}_{P_S}$

Inner Loop: N_I iterations

- Perform a convex combination

$$\mathbf{u}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i \mathbf{u}_i$$

- Weightings

$$\delta_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{P_S} \delta_i = 1$$

are adopted (boosting) to reflect goodness of $\mathbf{u}_i$

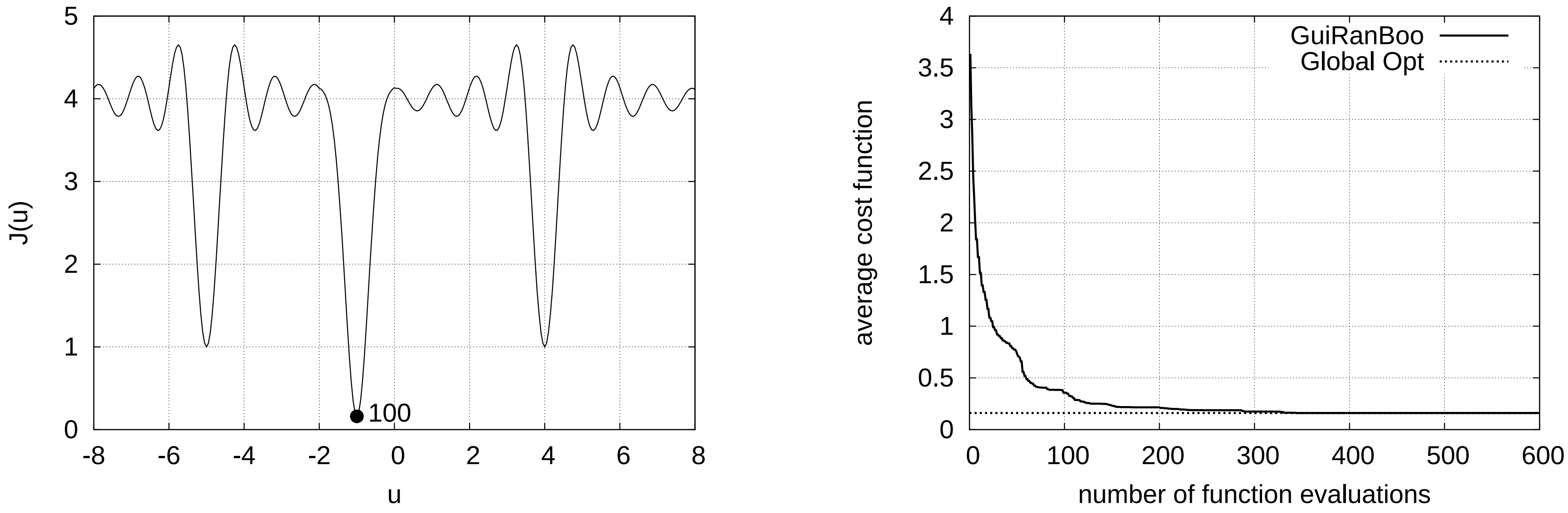
- $\mathbf{u}_{P_S+1}$ or its mirror image $\mathbf{u}_{P_S+2}$ replaces worst member in population $\mathbf{u}_i$, $1 \leq i \leq P_S$

End of *Inner Loop*

End of *Outer Loop*

Optimisation Example

- ◯ Population size $P_S = 6$, number of Inner iterations $N_I = 20$ and number of generations $N_G = 12$
- ◯ 100 random experiments, populations of all 100 runs converge to global minimum



OFS-LOO Algorithm

Give population size P_S, number of generations N_G, accuracy for terminating weighted boosting search ξ_B, and initial conditions

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

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

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

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

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

$$\alpha_{j,n}^{(i)} = \mathbf{p}_j^T \mathbf{g}_n^{(i)} / \mathbf{p}_j^T \mathbf{p}_j, \ 1 \le j < n, \quad \mathbf{p}_n^{(i)} = \mathbf{g}_n^{(i)} - \sum_{j=1}^{n-1} \alpha_{j,n}^{(i)} \mathbf{p}_j, \quad \theta_n^{(i)} = \left(\mathbf{p}_n^{(i)}\right)^T \mathbf{y} / \left(\left(\mathbf{p}_n^{(i)}\right)^T \mathbf{p}_n^{(i)} + \lambda\right) \tag{1}$$

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

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

$$J_n^{(i)} = \frac{1}{N}\sum_{k=1}^{N} \left(e_k^{(n)}(i)/\eta_k^{(n)}(i)\right)^2 \tag{3}$$

where $p_n^{(i)}(k)$ is 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 distribution weightings for $1 \le i \le P_S$ and then normalise them

$$\delta_i(t) = \begin{cases} \delta_i(t-1)\beta_t^{\bar{J}_n^{i)}}, & \text{for } \beta_t \le 1 \\ \delta_i(t-1)\beta_t^{1-\bar{J}_n^{i)}}, & \text{for } \beta_t > 1 \end{cases} \qquad \delta_i(t) = \frac{\delta_i(t)}{\sum_{m=1}^{P_S} \delta_m(t)}$$

Step 2: Parameter updating

1. Construct $(P_S + 1)$th and $(P_S + 2)$th points using

$$\mathbf{u}_{P_S+1} = \sum_{i=1}^{P_S} \delta_i(t)\mathbf{u}_i^{[l]} \qquad \mathbf{u}_{P_S+2} = \mathbf{u}_{\text{best}}^{[l]} + \left(\mathbf{u}_{\text{best}}^{[l]} - \mathbf{u}_{P_S+1}\right)$$

2. 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 (1)), and compute their corresponding LOO cost function values $J_n^{(i)}$, $i = P_S+1, P_S+2$ (as in (2) and (3)). 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 population

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

End of inner loop

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

End of outer loop

This yields:

solution $\mathbf{u} = \mathbf{u}_{\text{best}}^{[N_G]}$, i.e. $\boldsymbol{\mu}_n$ and $\boldsymbol{\Sigma}_n$ of nth RBF node

nth model column $\mathbf{g}_n$

orthogonalisation coefficients $\alpha_{j,n}$, $1 \le j < n$

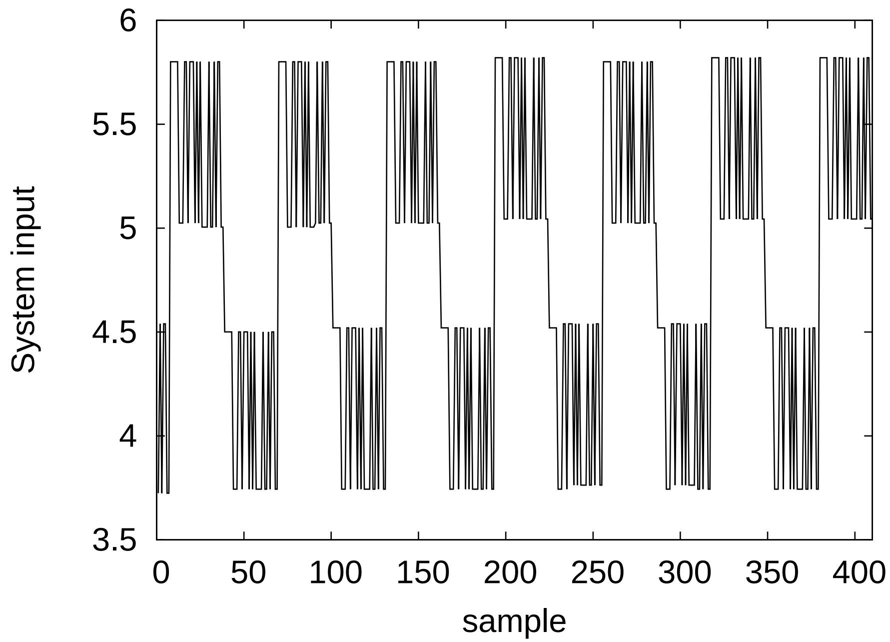
corresponding orthogonal model column $\mathbf{p}_n$ and weight θ_n

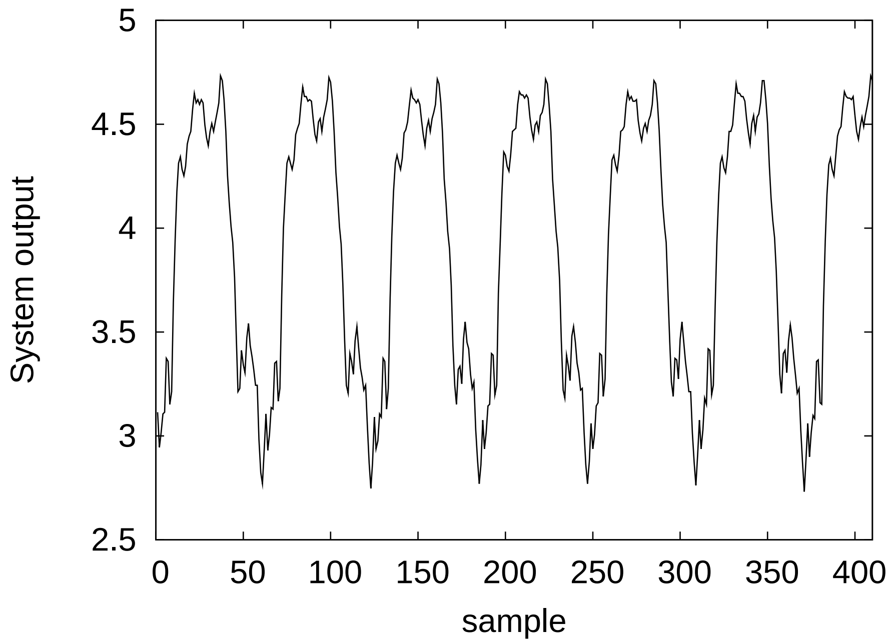
n-term modelling errors $e_k^{(n)}$ and associated LOO modelling error weightings $\eta_k^{(n)}$ for $1 \le k \le N$

Engine Data Modelling

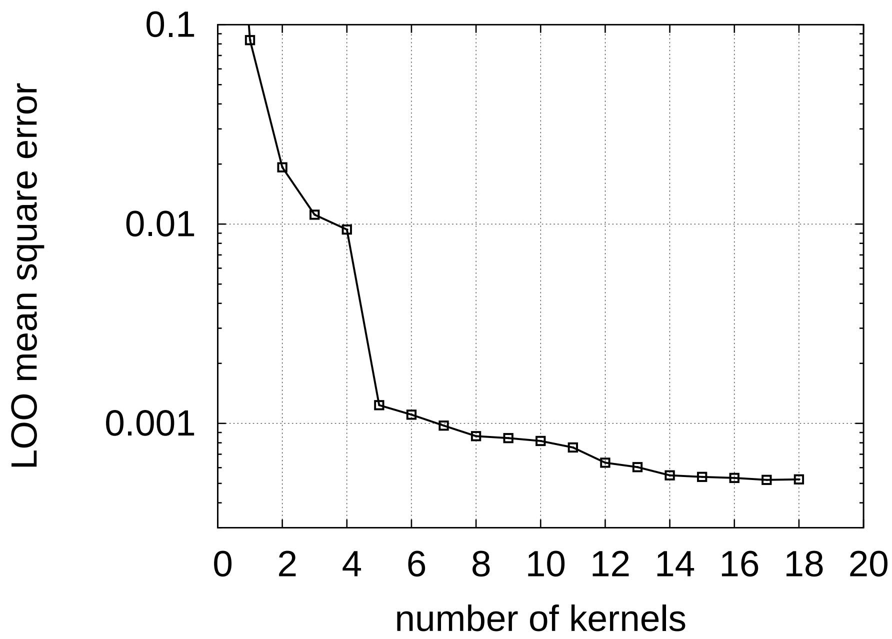
◯ Modelling relationship between fuel rack position (input u_k) and engine speed (output y_k) for a Leyland TL11 turbocharged, direct injection diesel engine operated at low engine speed

◯ Data set contains 410 pairs of input-output samples (u_k, y_k), modelled as $y_k = f_s(\mathbf{x}_k) + e_k$ with $\mathbf{x}_k = [y_{k-1}\ u_{k-1}\ u_{k-2}]^T$, first 210 data points for training and last 200 points for testing





○ LOO mean square error as function of model size for engine data set



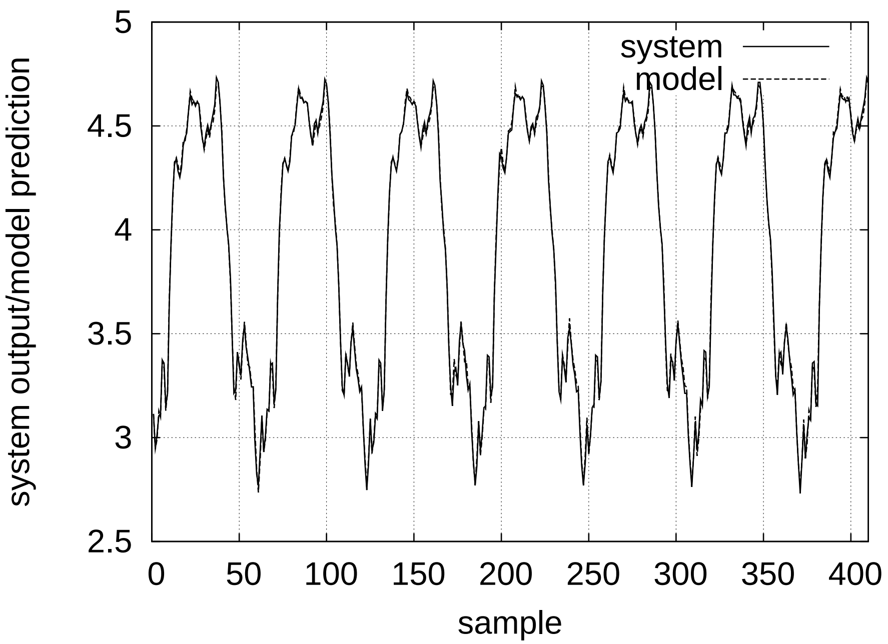
○ OFS-LOO constructed 17 RBF nodes, LROLS-LOO then reduced model to 15 nodes

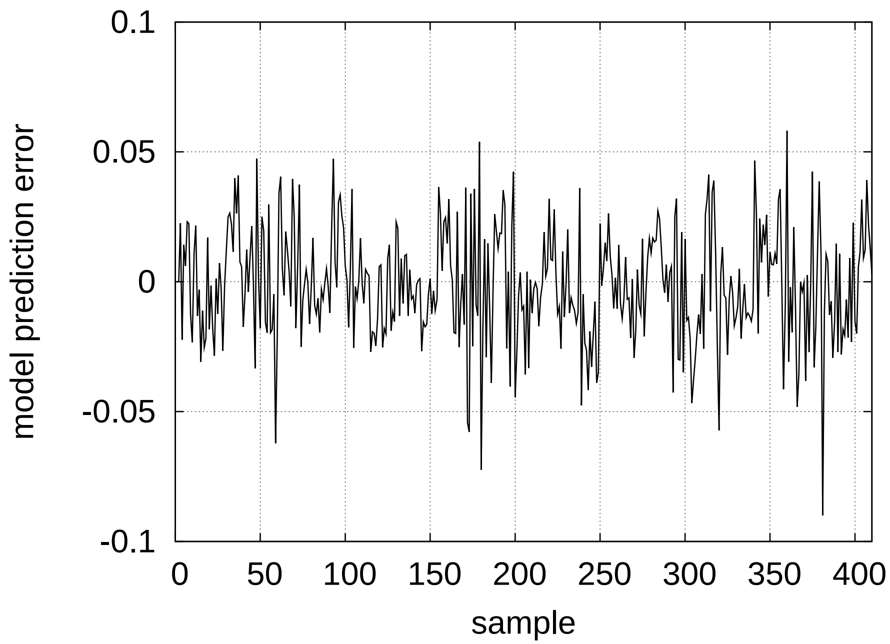
○ Results were compared with those obtained by SVM and LROLS-LOO

◯ Comparison of SVM, LROLS-LOO and OFS-LOO algorithms for 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

◯ Model output $\hat{y}_k$ and error $e_k = y_k - \hat{y}_k$ of 15-node RBF network for engine data set

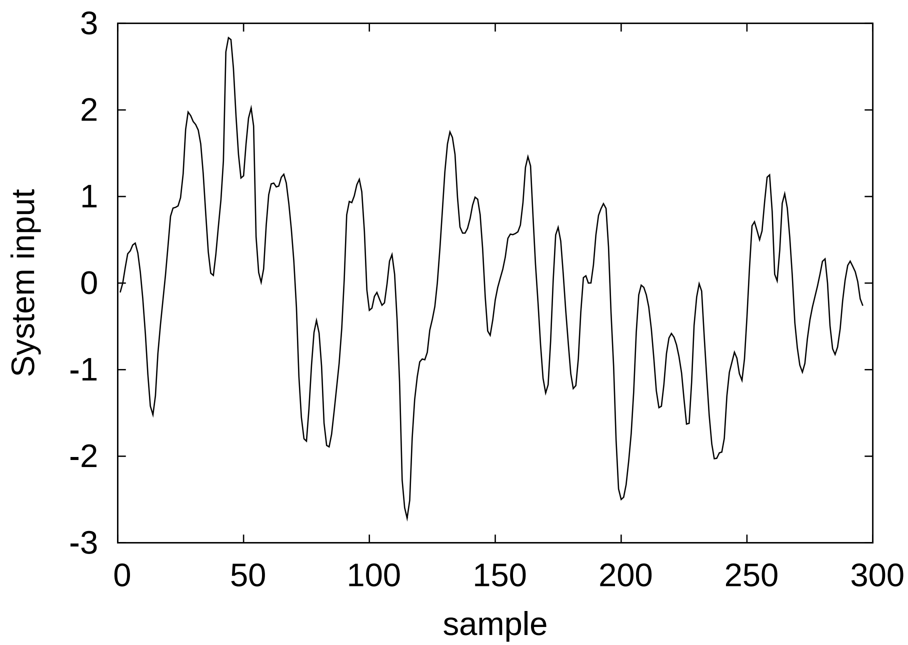


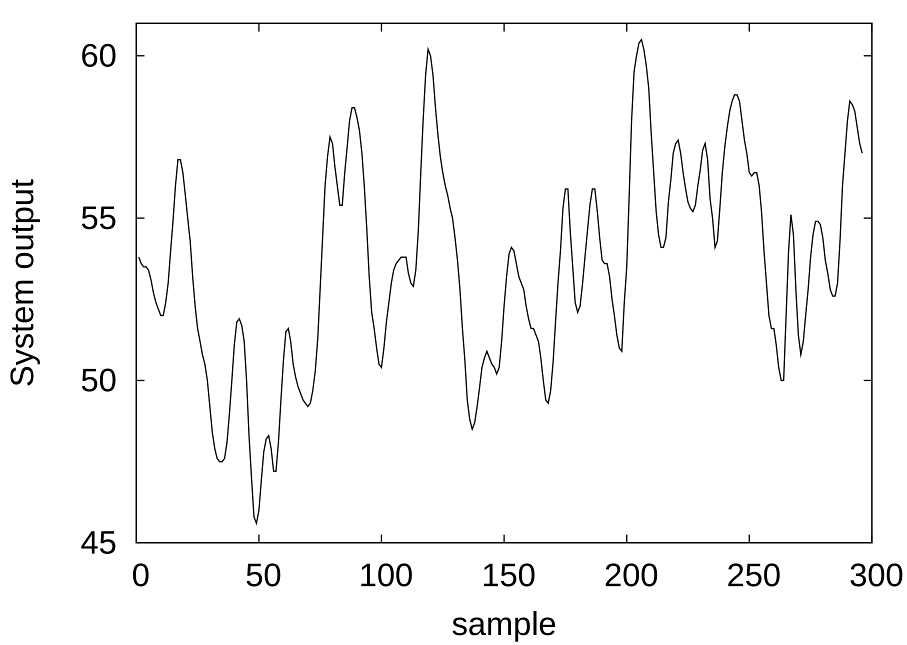


Gas Furnace Data Modelling

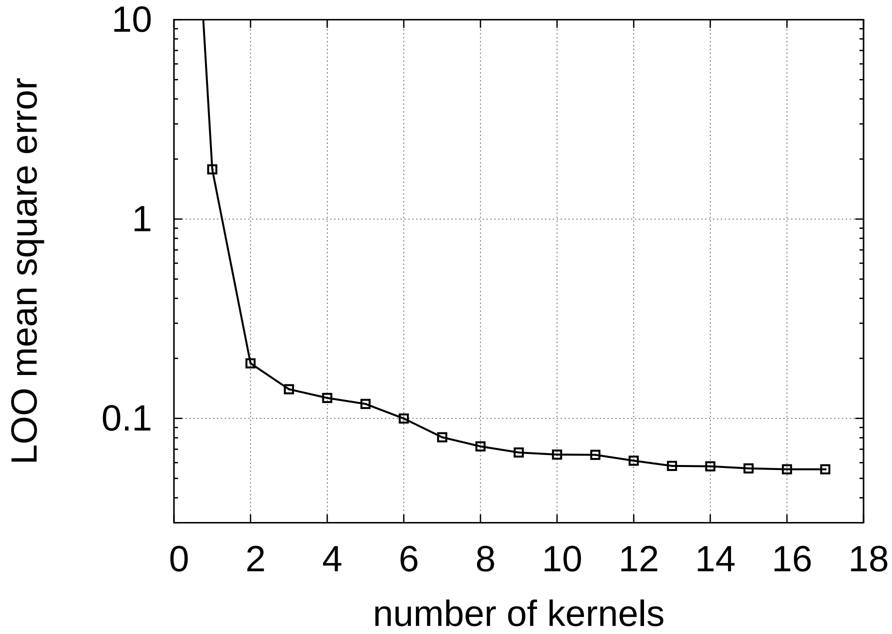
◯ Modelling relationship between coded input gas feed rate (input u_k) and CO_2 concentration (output y_k) for a gas furnace data set

◯ Data set contains 296 pairs of input-output samples (u_k, y_k), modelled as $y_k = f_s(\mathbf{x}_k) + e_k$ with $\mathbf{x}_k = [y_{k-1}\ y_{k-2}\ y_{k-3}\ u_{k-1}\ u_{k-2}\ u_{k-3}]^T$, all the data points were used for training





○ LOO mean square error as function of model size for gas furnace data set



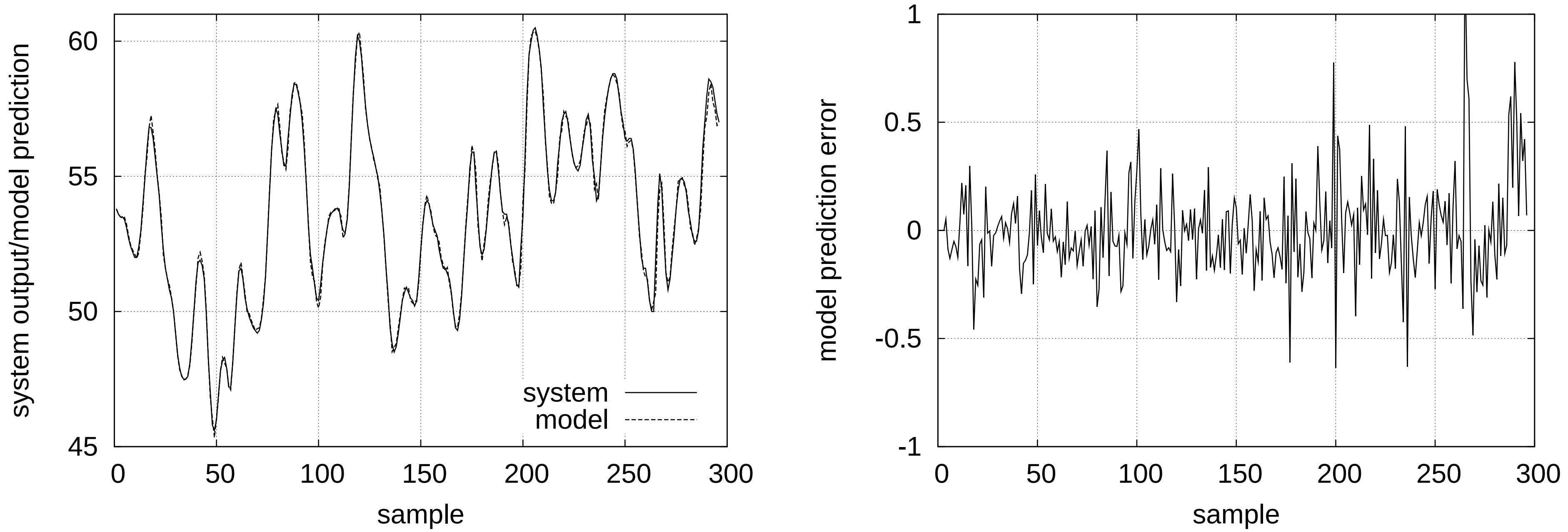
○ OFS-LOO constructed 16 RBF nodes, LROLS-LOO then reduced model to 15 nodes

○ Results were compared with those obtained by SVM and LROLS-LOO

○ Comparison of SVM, LROLS-LOO and OFS-LOO algorithms for 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

○ Model output $\hat{y}_k$ and error $e_k = y_k - \hat{y}_k$ of 15-node RBF network for gas furnace data set



Boston Housing Data Modelling

○ Boston Housing: `http://www.ics.uci.edu/~mlearn/MLRepository.html`

Data set comprises 506 data points with 14 variables

Predicting median house value from remaining 13 attributes

○ Modelling: randomly selected 456 data points from data set for training and used remaining 50 data points to form test set

Average results were given over 100 repetitions

○ Comparison of SVM, LROLS-LOO and OFS-LOO algorithms for Boston Housing data set

algorithm	RBF type	model size	training MSE	test MSE
SVM	fixed Gaussian	243.2 ± 5.3	6.7986 ± 0.4444	23.1750 ± 9.0459
LROLS-LOO	fixed Gaussian	58.6 ± 11.3	12.9690 ± 2.6628	17.4157 ± 4.6670
OFS-LOO	tunable Gaussian	34.6 ± 8.4	10.0997 ± 3.4047	14.0745 ± 3.6178

Conclusions

- A novel construction algorithm has been proposed for regression modelling using the radial basis function network with **tunable nodes**
- Proposed algorithm has ability to tune centre and covariance matrix of individual radial basis function node to minimise leave-one-out mean square error
- A global search algorithm, referred to as **RWBS**, has been adopted to construct radial basis function nodes in an orthogonal forward selection procedure
- Model construction procedure is fully automatic and user does not need to specify any learning algorithmic parameter
- The proposed **OFS-LOO** approach offers enhanced modelling capability with very small radial basis function network models