

Sparse Incremental Regression Modeling Using Correlation Criterion with Boosting Search

S. Chen, X.X. Wang and D.J. Brown

Abstract—A novel technique is presented to construct sparse generalized Gaussian kernel regression models. The proposed method appends regressors in an incremental modeling by tuning the mean vector and diagonal covariance matrix of individual Gaussian regressor to best fit the training data based on a correlation criterion. It is shown that this is identical to incrementally minimize the modeling mean square error. The optimization at each regression stage is carried out with a simple search algorithm re-enforced by boosting. Experimental results obtained using this technique demonstrate that it offers a viable alternative to the existing state-of-art kernel modeling methods for constructing parsimonious models.

Keywords—Regression, Gaussian kernel model, incremental modeling, correlation, boosting

I. INTRODUCTION

A basic principle in nonlinear data modeling is the parsimonious principle of ensuring the smallest possible model that explains the training data. The state-of-art sparse kernel modeling techniques [1]–[10] have widely been adopted in data modeling applications. These existing sparse modeling techniques typically use a fixed common variance for all the regressors and select the kernel centers from the training input data. We present a flexible construction method for generalized Gaussian kernel models by appending regressor one by one in an incremental modeling. The correlation between a Gaussian regressor and the training data is used as the criterion to optimize the mean vector and diagonal covariance matrix of the regressor. This approach is equivalent to incrementally minimizing the modeling mean square error. The optimization is carried out with a simple boosting search. Because kernel means are not restricted to the training input data and each regressor has an individually tuned diagonal covariance matrix, our method can produce very sparse models that generalize well and it offers a viable alternative to the existing state-of-art sparse kernel modeling methods.

Our proposed incremental modeling method is very different from the cascade-correlation incremental learning [11]. In the cascade-correlation method, regression units are constructed on a variable space of increasing dimension, namely, the inputs to a unit being the original inputs and the outputs of the previously selected units. Our proposed method is a truly incremental modeling from the input space to the output space. It has a desired geometric property that a regressor is constructed to fit the peak (in the sense of magnitude) of the current modeling residual at each stage. This geometric property is graphically illustrated in a simple one-dimensional modeling problem. Our method also has advantages over the radial basis function network training methods based on clustering (e.g. [12]–[14]). In these clustering based learn-

ing methods, the number of clusters or the model size must be learned by other means, for example, via cross-validation [15],[16]. Moreover, the regressor kernel variances also need to be decided using some other appropriate techniques.

II. METHOD

Consider the problem of fitting the N pairs of training data $\{\mathbf{x}_l, y_l\}_{l=1}^N$ with the regression model

$$\hat{y}(\mathbf{x}) = \sum_{i=1}^M w_i g_i(\mathbf{x}) \quad (1)$$

where \mathbf{x} is the m -dimensional input variable; w_i , $1 \leq i \leq M$, denote the model weights; M is the number of regressors; and $g_i(\bullet)$, $1 \leq i \leq M$, denote the regressors. We allow the regressor to be chosen as the generalized Gaussian kernel function $g_i(\mathbf{x}) = G(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ with

$$G(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \quad (2)$$

where $\boldsymbol{\mu}_i$ is the i th kernel center or mean vector and the covariance matrix $\boldsymbol{\Sigma}_i$ is diagonal. We will adopt an incremental approach to build up the regression model (1) by appending regressors one by one. Let us first introduce the following notation

$$\left. \begin{aligned} y_i^{(0)} &= y_i \\ y_i^{(k)} &= y_i^{(k-1)} - w_k g_k(\mathbf{x}_i) \end{aligned} \right\} 1 \leq i \leq N \quad (3)$$

Obviously, $y_i^{(k)}$ is the modeling error at \mathbf{x}_i after the k th regressor has been fitted and $y_i^{(0)}$ is simply the desired output for the input \mathbf{x}_i . Next define the mean square error (MSE) for the k -term regression model over the training data as

$$\text{MSE}_k = \frac{1}{N} \sum_{i=1}^N \left(y_i^{(k)} \right)^2 = \frac{1}{N} \sum_{i=1}^N \left(y_i - \sum_{j=1}^k w_j g_j(\mathbf{x}_i) \right)^2 \quad (4)$$

The incremental modeling process is terminated when $\text{MSE}_k < \xi$, where ξ is a preset modeling accuracy. The termination of the model construction process can alternatively be decided by cross-validation [15],[16], and other termination criteria include the Akaike information criterion [17], the optimal experimental design criteria [9] and the leave-one-out generalization criterion [10].

At the k th stage of modeling, the regressor $g_k(\mathbf{x})$ is fitted to the training data set $\{\mathbf{x}_i, y_i^{(k-1)}\}_{i=1}^N$ by tuning its mean vector $\boldsymbol{\mu}_k$ and diagonal covariance matrix $\boldsymbol{\Sigma}_k$. The correlation function between the regressor and the training data set

S. Chen is with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

X.X. Wang and D.J. Brown are with Department of Creative Technologies, University of Portsmouth, Portsmouth PO1 3HE, U.K.

as given by

$$C_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) = \frac{\sum_{i=1}^N g_k(\mathbf{x}_i) y_i^{(k-1)}}{\sqrt{\sum_{i=1}^N g_k^2(\mathbf{x}_i)} \sqrt{\sum_{i=1}^N (y_i^{(k-1)})^2}} \quad (5)$$

defines the similarity between $g_k(\mathbf{x})$ and $\{\mathbf{x}_i, y_i^{(k-1)}\}_{i=1}^N$. This correlation criterion can be used to position and shape a regressor. That is, $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ of the k th regressor are chosen to maximize $|C_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)|$. After the regressor positioning and shaping, the corresponding weight is calculated by the usual least squares solution

$$w_k = \frac{\sum_{i=1}^N y_i^{(k-1)} g_k(\mathbf{x}_i)}{\sum_{i=1}^N g_k^2(\mathbf{x}_i)} \quad (6)$$

Selecting regressors by maximizing $|C_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)|$ is identical to incrementally minimizing the modeling MSE (4). Substituting (3) into (4) with w_k given by (6) yields

$$\text{MSE}_k = \left(\frac{1}{N} \sum_{i=1}^N (y_i^{(k-1)})^2 \right) (1 - C_k^2(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)) \quad (7)$$

Clearly maximizing $|C_k(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)|$ is equivalent to minimizing MSE_k with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. An important technique to alleviate over-fitting and improve robustness of the solution is to apply regularization [6]-[10]. The zero-order regularization can readily be incorporated with our proposed method by changing the least squares solution (6) into

$$w_k = \frac{\sum_{i=1}^N y_i^{(k-1)} g_k(\mathbf{x}_i)}{\lambda + \sum_{i=1}^N g_k^2(\mathbf{x}_i)} \quad (8)$$

where λ is a small positive regularization parameter.

The optimization for determining $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ can be performed with guided random search methods, such as the genetic algorithm [18],[19] and adaptive simulated annealing [20],[21]. However, we perform this optimization by a simple search which is re-enforced by boosting [22]–[24]. Let the vector \mathbf{u}_k contain the mean vector $\boldsymbol{\mu}_k$ and the diagonal covariance matrix $\boldsymbol{\Sigma}_k$. Given the training data $\{\mathbf{x}_i, y_i^{(k-1)}\}_{i=1}^N$, the basic *boosting search* algorithm is summarized:

Initialization: Set iteration index $t = 0$, give the s randomly chosen initial values for \mathbf{u}_k , $\mathbf{u}_k^{(1)}(t)$, $\mathbf{u}_k^{(2)}(t)$, \dots , $\mathbf{u}_k^{(s)}(t)$, with the associated weighting $\delta_i(t) = \frac{1}{s}$ for $1 \leq i \leq s$, and specify a small positive value ξ_b for terminating the search and a maximum number of iterations M_I .

Step 1: Boosting

1. Calculate the loss of each point, namely $\text{cost}_j = 1 - |C_k(\mathbf{u}_k^{(j)}(t))|$, $1 \leq j \leq s$
2. Find $\mathbf{u}_k^{\text{best}}(t) = \arg \min\{\text{cost}_j, 1 \leq j \leq s\}$ and $\mathbf{u}_k^{\text{worst}}(t) = \arg \max\{\text{cost}_j, 1 \leq j \leq s\}$
3. Normalize the loss

$$\text{loss}_j = \frac{\text{cost}_j}{\sum_{l=1}^s \text{cost}_l}, \quad 1 \leq j \leq s$$

4. Compute a weighting factor β_t according to

$$\epsilon_t = \sum_{j=1}^s \delta_j(t) \text{loss}_j, \quad \beta_t = \frac{\epsilon_t}{1 - \epsilon_t}$$

5. For $j = 1, \dots, s$, update the distribution weightings

$$\delta_j(t+1) = \begin{cases} \delta_j(t) \beta_t^{\text{loss}_j} & \text{for } \beta_t \leq 1, \\ \delta_j(t) \beta_t^{1-\text{loss}_j} & \text{for } \beta_t > 1, \end{cases}$$

6. Normalize the weighting vector

$$\delta_j(t+1) = \frac{\delta_j(t+1)}{\sum_{l=1}^s \delta_l(t+1)}, \quad 1 \leq j \leq s$$

Step 2: Parameter updating

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

$$\mathbf{u}_k^{(s+1)}(t) = \sum_{i=1}^s \delta_i(t+1) \mathbf{u}_k^{(i)}(t)$$

2. Construct the $(s+2)$ th point using the formula $\mathbf{u}_k^{(s+2)}(t) = \mathbf{u}_k^{\text{best}}(t) + (\mathbf{u}_k^{\text{best}}(t) - \mathbf{u}_k^{(s+1)}(t))$
3. Choose a better point (smaller loss value) from $\mathbf{u}_k^{(s+1)}(t)$ and $\mathbf{u}_k^{(s+2)}(t)$ to replace $\mathbf{u}_k^{\text{worst}}(t)$

Set $t = t + 1$ and repeat from *Step 1* until $\|\mathbf{u}_k^{(s+1)}(t) - \mathbf{u}_k^{(s+1)}(t-1)\| < \xi_b$ or M_I iterations have been reached. Then choose the k th regressor $\mathbf{u}_k = \mathbf{u}_k^{\text{best}}(t)$

The above basic *boosting search* algorithm performs a guided random search and solution obtained may depend on the initial choice of the population. To derive a robust algorithm that ensures a stable solution, we augment it into the following *repeated boosting search* algorithm.

Initialization: Specify a maximum repeating times M_R and a small positive number ξ_r for stopping the search

First generation: Randomly choose the s number of the initial population $\mathbf{u}_k^{(1)}$, $\mathbf{u}_k^{(2)}$, \dots , $\mathbf{u}_k^{(s)}$, and call the *boosting search* algorithm to obtain a solution $\mathbf{u}_k^{\text{best}}(0)$

Repeat loop: For $l = 1 : M_R$

Set $\mathbf{u}_k^{(1)} = \mathbf{u}_k^{\text{best}}(l-1)$, and randomly generate the other $s-1$ points $\mathbf{u}_k^{(i)}$ for $2 \leq i \leq s$

Call the *boosting search* algorithm to obtain a solution $\mathbf{u}_k^{\text{best}}(l)$

If $\|\mathbf{u}_k^{\text{best}}(l-1) - \mathbf{u}_k^{\text{best}}(l)\| < \xi_r$, Exit loop

End for

Choose the k th regressor as $\mathbf{u}_k = \mathbf{u}_k^{\text{best}}(l)$

The algorithmic parameters that need to be chosen appropriately are the population size s , termination criterion ξ_b and maximum number of iterations M_I in the boosting search as well as the maximum number of repeating times M_R and the stopping criterion ξ_r for the repeating loop. To simplify the algorithm tuning, we can simply fix M_I and M_R without the need to specify ξ_b and ξ_r . In the following modeling experiments, the value of s , M_I and M_R were chosen empirically to ensure that the incremental modeling procedure produced consistent final models with the same levels of modeling accuracy and model sparsity for different runs. The stopping threshold ξ for the incremental modeling procedure should ideally be set to a value slightly larger than the system noise variance. Since the system noise level is generally unknown *a priori*, an appropriate value for ξ has to be learned during the modeling process. Alternative, the Akaike information criterion [17] and the optimal experimental design criteria [9] can be employed to terminate the model construction procedure without the need to specify a modeling accuracy ξ .

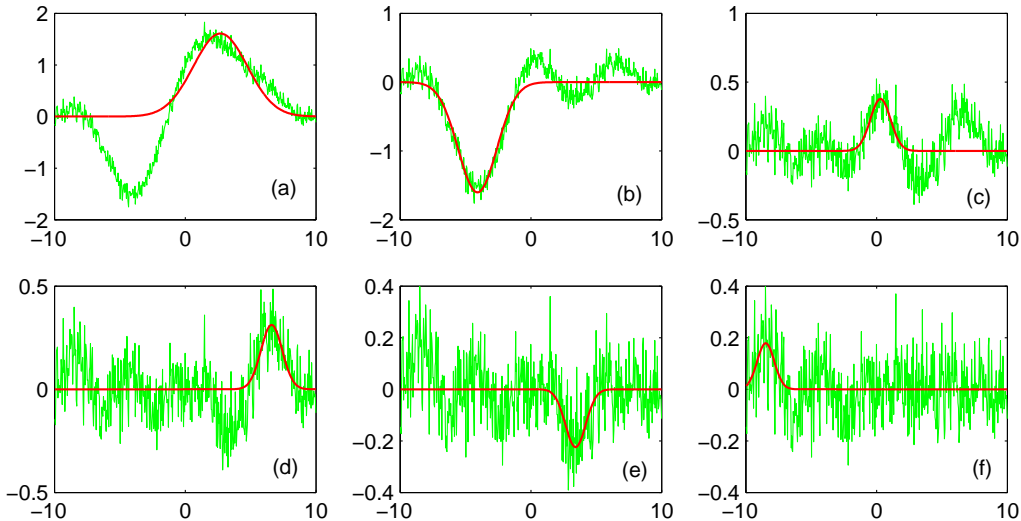


Fig. 1. Incremental modeling procedure for the simple function fitting problem: in (a)–(f), the light curves are the modeling errors of the previous stage, $y_i^{(k-1)}$, and the dark curves are the fitted current regressors, $w_k g_k(x_i)$, for $1 \leq k \leq 6$, respectively.

III. EXPERIMENTAL RESULTS

Two examples were used to illustrate the proposed sparse modeling approach. The first example was a one-dimensional simulated data set and was chosen to demonstrate graphically the motivation and desired property of the incremental regression procedure using the correlation criterion. The second example was a real-data set.

Example 1. The 500 points of training data were generated from

$$y(x) = 0.1x + \frac{\sin x}{x} + \sin 0.5x + e$$

with equal-spaced $x \in [-10, 10]$, where e was a Gaussian white noise with zero mean and variance 0.01. With a population size $s = 5$, the maximum number of iterations $M_I = 20$ and the maximum repeating times $M_R = 10$ together with the modeling accuracy set to $\xi = 0.012$, the incremental modeling consistently produced models of 6 Gaussian regressors with the same $MSE_6 = 0.011$ for a large number of different runs. We also used the Akaike information criterion [17] and the optimal experimental design criteria [9] to stop the selection procedure, rather than specifying the modeling accuracy ξ , and the results obtained are identical. The construction process in a typical run is illustrated graphically in Fig. 1 (a)–(f), where the effectiveness of regressor tuning based on the correlation criterion is clearly demonstrated. In Fig. 2 (a), the model output from the constructed 6-term model is superimposed on the noisy training data, and the final modeling errors are shown in Fig. 2 (b).

Example 2. This example constructed a model representing the relationship between the fuel rack position (input $u(t)$) and the engine speed (output $y(t)$) for a Leyland TL11 turbocharged, direct injection diesel engine operated at a low engine speed. Detailed system description and experimental setup can be found in [25]. The data set contained 410 samples. The first 210 data points were used in training and the last 200 points in model validation. The training data set was constructed with $y_i = y(i)$ and $\mathbf{x}_i = [y(i-1) \ u(i-1) \ u(i-2)]^T$ for $i = 3, 4, \dots, 210$. We used the proposed

approach to fit a generalized Gaussian regression model to this data set. With $s = 37$, $M_I = 60$ and $M_R = 20$ together with $\xi = 0.00055$, the incremental modeling produced in repeated runs consistent models of 9 Gaussian regressors with the MSE values of 0.00053 and 0.00055 over the training and testing sets, respectively. Fig. 3 (a) depicts the model prediction $\hat{y}(t)$ for a typical 9-term model obtained, in comparison with the system output $y(t)$. The corresponding model prediction error $\epsilon(t) = y(t) - \hat{y}(t)$ is shown in Fig. 3 (b). We also ran the experiments using the Akaike information and optimal experimental design criteria to stop the modeling process, and the results obtained were similar to those obtained with given the modeling accuracy of $\xi = 0.00055$.

Various existing state-of-art kernel modeling techniques had been used to fit this data set in [9],[10]. These kernel modeling techniques can only choose the kernel mean vectors from the training input data points and use a single fixed common variance for all the regressors. The best Gaussian kernel model with an optimal single common variance of $\sigma^2 = 1.69$ obtained by one of the existing state-of-art kernel modeling techniques required at least 20 model regressors to achieve the same modeling accuracy (see [10]). In comparison, the proposed modeling approach resulted in a much sparser 9-term generalized Gaussian kernel model.

IV. CONCLUSIONS

An incremental modeling technique has been presented to construct sparse generalized Gaussian regression models. The proposed technique can tune the mean vector and diagonal covariance matrix of individual Gaussian regressor to best fit the training data incrementally based on the correlation between the regressor and the training data. A simple boosting search algorithm has been adopted for regressor tuning at each modeling stage. Experimental results using this construction technique have demonstrated that it offers a viable alternative to the existing state-of-art kernel modeling methods for constructing parsimonious regression models.

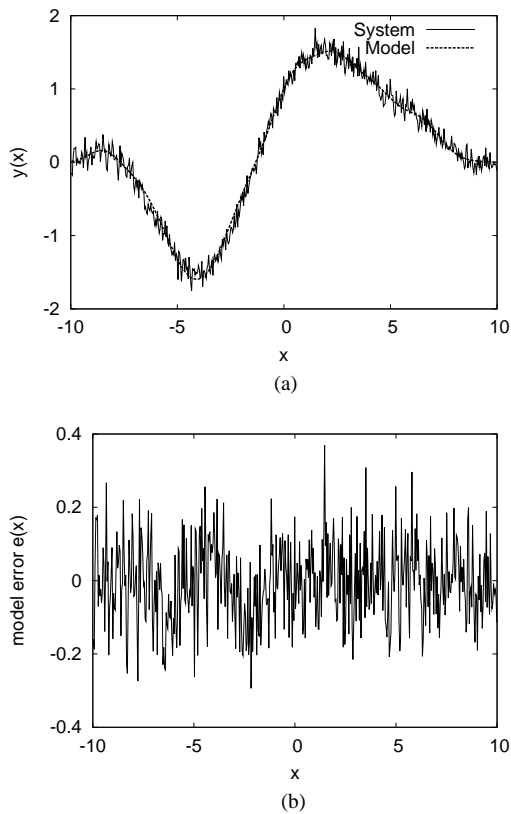


Fig. 2. Incremental modeling results for the simple function fitting problem: in (a) outputs \hat{y}_i of the final 6-term model are superimposed on the noisy training data y_i , and (b) shows the final modeling errors.

REFERENCES

- [1] 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.
- [2] 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.
- [3] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
- [4] V. Vapnik, S. Golowich and A. Smola, "Support vector method for function approximation, regression estimation, and signal processing," in: M.C. Mozer, M.I. Jordan and T. Petsche, Eds., *Advances in Neural Information Processing Systems 9*. Cambridge, MA: MIT Press, 1997, pp.281–287.
- [5] B. Schölkopf, K.K. Sung, C.J.C. Burges, F. Girosi, P. Niyogi, T. Poggio and V. Vapnik, "Comparing support vector machines with Gaussian kernels to radial basis function classifiers," *IEEE Trans. Signal Processing*, Vol.45, No.11, pp.2758–2765, 1997.
- [6] S. Chen, Y. Wu and B.L. Luk, "Combined genetic algorithm optimization and regularised orthogonal least squares learning for radial basis function networks," *IEEE Trans. Neural Networks*, Vol.10, No.5, pp.1239–1243, 1999.
- [7] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, Vol.1, pp.211–244, 2001.
- [8] B. Schölkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
- [9] S. Chen, X. Hong and C.J. Harris, "Sparse kernel regression modelling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, Vol.48, No.6, pp.1029–1036, 2003.
- [10] S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modelling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, Vol.34, No.2, pp.898–911, 2004.
- [11] S.E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in: D.S. Touretzky, Ed., *Neural Information Processing Systems 2*, Morgan-Kaufmann, 1990, pp.524–532.

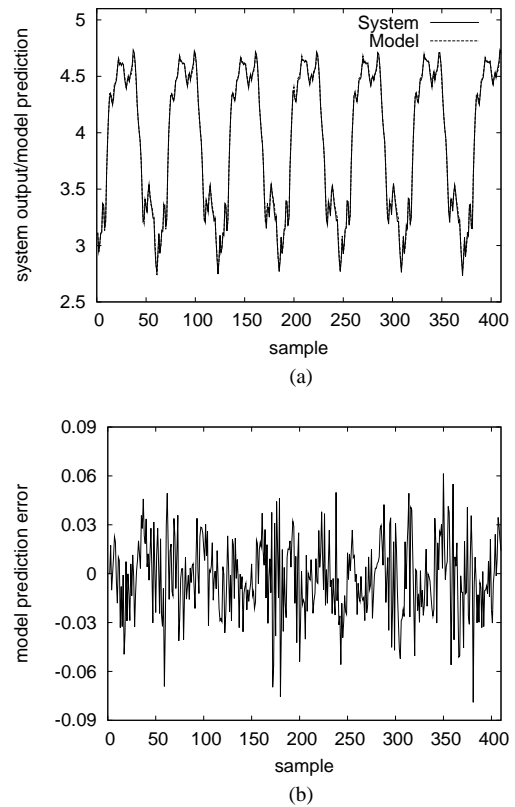


Fig. 3. The engine data set: (a) model output $\hat{y}(t)$ (dashed) superimposed on system output $y(t)$ (solid), and (b) model prediction error $\epsilon(t) = y(t) - \hat{y}(t)$.

- [12] J. Moody and C.J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computation*, Vol.1, pp.281–294, 1989.
- [13] S. Chen, S.A. Billings and P.M. Grant, "Recursive hybrid algorithm for non-linear system identification using radial basis function networks," *Int. J. Control*, Vol.55, No.5, pp.1051–1070, 1992.
- [14] S. Chen, "Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electronics Letters*, Vol.31, No.2, pp.117–118, 1995.
- [15] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. Royal Statistics Society Series B*, Vol.36, pp.117–147, 1974.
- [16] R.H. Myers, *Classical and Modern Regression with Applications*. 2nd Edition, Boston: PWS-KENT, 1990.
- [17] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, Vol.AC-19, pp.716–723, 1974.
- [18] D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison Wesley, 1989.
- [19] K.F. Man, K.S. Tang and S. Kwong, *Genetic Algorithms: Concepts and Design*. London: Springer-Verlag, 1998.
- [20] L. Ingber, "Simulated annealing: practice versus theory," *Mathematical and Computer Modeling*, Vol.18, No.11, pp.29–57, 1993.
- [21] S. Chen and B.L. Luk, "Adaptive simulated annealing for optimization in signal processing applications," *Signal Processing*, Vol.79, No.1, pp.117–128, 1999.
- [22] Y. Freund and R.E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Computer and System Sciences*, Vol.55, No.1, pp.119–139, 1997.
- [23] R.E. Schapire, "The strength of weak learnability," *Machine Learning*, Vol.5, No.2, pp.197–227, 1990.
- [24] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in: S. Mendelson and A. Smola, eds., *Advanced Lectures in Machine Learning*. Springer Verlag, 2003, pp.119–184.
- [25] 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.