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 an individual Gaussian regressor to best fit the training data, based on a correlation criterion. It is shown that this is identical to incrementally minimizing the modeling mean square error (MSE). 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-the-art kernel modeling methods for constructing parsimonious models.**

***Index Terms*—Boosting, correlation, Gaussian kernel model, incremental modeling, regression.**

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-the-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 regressors 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 (MSE). 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-the-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 learning 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.

Manuscript received April 30, 2004; revised September 10, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marcelo G. S. Bruno.

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

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

Digital Object Identifier 10.1109/LSP.2004.842250

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

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

where $\mathbf{x}$ is the m-dimensional input variable, w_i, $1 \le i \le M$ denote the model weights, M is the number of regressors, and $g_i(\bullet)$, $1 \le i \le M$ 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

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

where $\boldsymbol{\mu}_i$ is the ith 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:

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

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

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

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 kth 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 as given by

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

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 kth 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

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

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 overfitting 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 adding a small positive regularization parameter to the denominator of the least squares solution (6).

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 below.

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

Step 1: Boosting

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

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

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 $j = 1, \cdots, s$, update the distribution weightings

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

6. Normalize the weighting vector

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

Step 2: Parameter updating

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

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

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

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 kth regressor $\mathbf{u}_k = \mathbf{u}_k^{best}(t)$.

The above basic *boosting search* algorithm performs a guided random search, and the 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 $\mathbf{u}_k^{(1)}, \mathbf{u}_k^{(2)}, \cdots, \mathbf{u}_k^{(s)}$, and call the *boosting search* algorithm to obtain a solution $\mathbf{u}_k^{best}(0)$.

Repeat loop: For $l = 1 : M_R$.

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

```
  Call the boosting search algorithm to
  obtain a solution u_k^best(l).
  If ||u_k^best(l-1) - u_k^best(l)|| < ξ_r, exit the loop.
End for
Choose the kth regressor as u_k = u_k^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 values 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 ideally should 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. Alternatively, 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.

III. Experimental Results

Two examples were used to illustrate the proposed sparse modeling approach. The first example was a one-dimensional simulated data set that 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.

1) 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 six Gaussian regressors with the same $\text{MSE}_6 = 0.011$ 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 six-term model is superimposed on the noisy training data, and the final modeling errors are shown in Fig. 2(b).

2) 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. A detailed system description and a 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 were used 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 $i = 3, 4, \cdots, 210$. 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 $\xi = 0.000\,55$, the incremental modeling produced in repeated runs consistent models of nine Gaussian regressors with the MSE values of 0.000 53 and 0.000 55 over the training and testing sets, respectively. Fig. 3(a) depicts the model prediction $\hat{y}(t)$ for

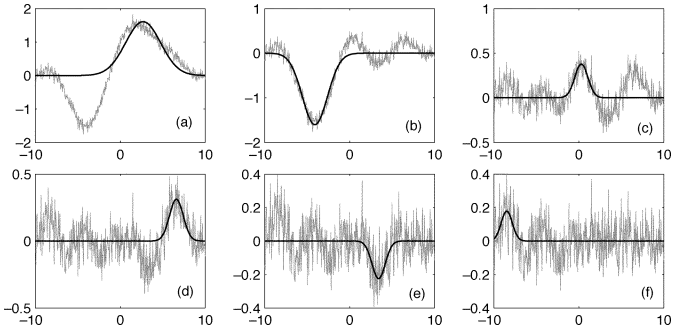

Fig. 1. Incremental modeling procedure for the simple function fitting problem: In (a)–(f), the light curves are the modeling errors of the previous stage $y_i^{(k-1)}$, and the dark curves are the fitted current regressors $w_k g_k(x_i)$ for $1 \le k \le 6$, respectively.

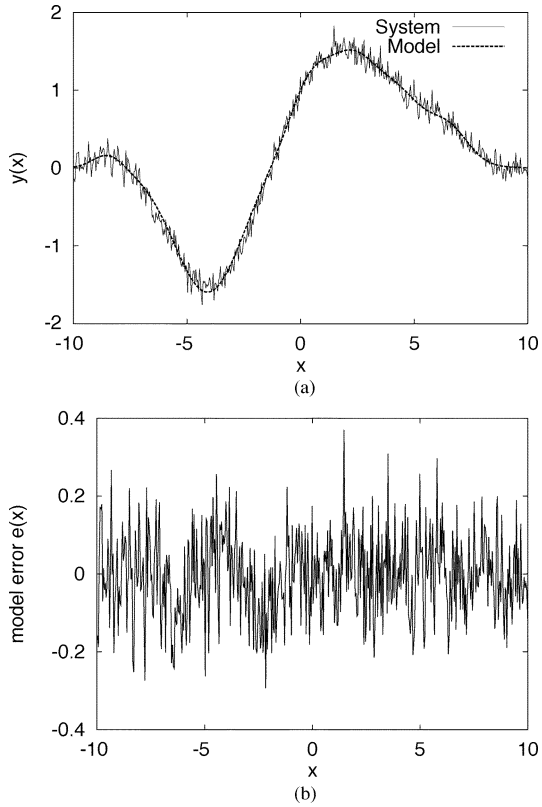

Fig. 2. Incremental modeling results for the simple function fitting problem: (a) Outputs $\hat{y}_i$ of the final six-term model are superimposed on the noisy training data y_i. (b) Final modeling errors.

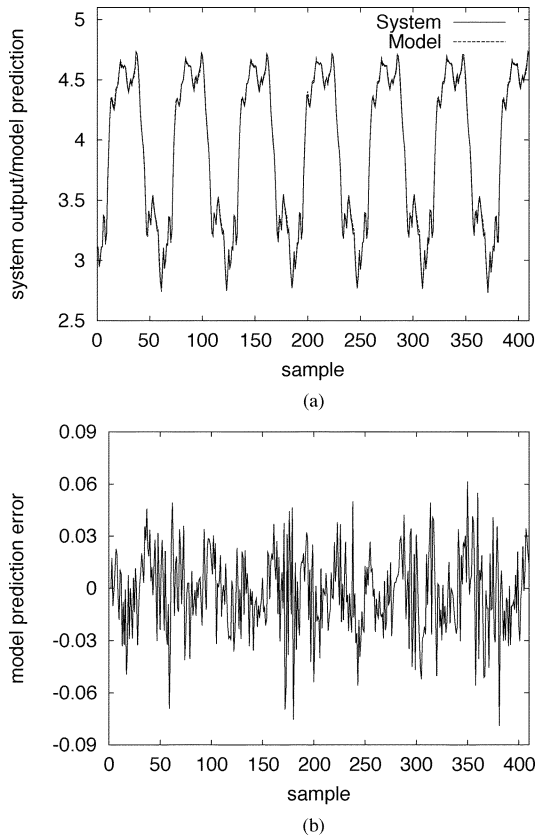

Fig. 3. Engine data set. (a) Model output $\hat{y}(t)$ (dashed) superimposed on system output $y(t)$ (solid). (b) Model prediction error $\epsilon(t) = y(t) - \hat{y}(t)$.

a typical nine-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 given the modeling accuracy of $\xi = 0.000\,55$.

Various existing state-of-the-art kernel modeling techniques had been used to fit this data set in [9] and [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-the-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 nine-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 regressors 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-the-art kernel modeling methods for constructing parsimonious regression models.

References

[1] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to nonlinear 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 Netw.*, vol. 2, no. 2, pp. 302–309, Mar. 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 *Advances in Neural Information Processing Systems 9*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. 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 Process.*, vol. 45, no. 11, pp. 2758–2765, Nov. 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 Netw.*, vol. 10, no. 5, pp. 1239–1243, Sep. 1999.

[7] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learning Res.*, 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 modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 1029–1036, Jun. 2003.

[10] S. Chen, X. Hong, C. J. Harris, and P. M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Syst., Man, Cybern. B*, vol. 34, no. 2, pp. 898–911, Apr. 2004.

[11] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Neural Information Processing Systems 2*, D. S. Touretzky, Ed. San Mateo, CA: Morgan-Kaufmann, 1990, pp. 524–532.

[12] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[13] S. Chen, S. A. Billings, and P. M. Grant, "Recursive hybrid algorithm for nonlinear system identification using radial basis function networks," *Int. J. Control*, vol. 55, no. 5, pp. 1051–1070, 1992.

[14] S. Chen, "Nonlinear time series modeling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning," *Electron. Lett.*, vol. 31, no. 2, pp. 117–118, 1995.

[15] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. R. Statist. Soc. B*, vol. 36, pp. 117–147, 1974.

[16] R. H. Myers, *Classical and Modern Regression With Applications*, 2nd ed. Boston, MA: PWS-KENT, 1990.

[17] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. AC-19, no. 6, pp. 716–723, Dec. 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, U.K.: Springer-Verlag, 1998.

[20] L. Ingber, "Simulated annealing: practice versus theory," *Math. Comput. 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 Process.*, 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. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, 1997.

[23] R. E. Schapire, "The strength of weak learnability," *Mach. Learning*, vol. 5, no. 2, pp. 197–227, 1990.

[24] R. Meir and G. Rätsch, "An introduction to boosting and leveraging," in *Advanced Lectures in Machine Learning*, S. Mendelson and A. Smola, Eds. New York: Springer-Verlag, 2003, pp. 119–184.

[25] S. A. Billings, S. Chen, and R. J. Backhouse, "The identification of linear and nonlinear models of a turbocharged automotive diesel engine," *Mech. Syst. Signal Process.*, vol. 3, no. 2, pp. 123–142, 1989.