Sparse Modeling Using Orthogonal Forward Regression With PRESS Statistic and Regularization

Sheng Chen, *Senior Member, IEEE*, Xia Hong, *Senior Member, IEEE*, Chris J. Harris, and Paul M. Sharkey

***Abstract*—The paper introduces an efficient construction algorithm for obtaining sparse linear-in-the-weights regression models based on an approach of directly optimizing model generalization capability. This is achieved by utilizing the delete-1 cross validation concept and the associated leave-one-out test error also known as the predicted residual sums of squares (PRESS) statistic, without resorting to any other validation data set for model evaluation in the model construction process. Computational efficiency is ensured using an orthogonal forward regression, but the algorithm incrementally minimizes the PRESS statistic instead of the usual sum of the squared training errors. A local regularization method can naturally be incorporated into the model selection procedure to further enforce model sparsity. The proposed algorithm is fully automatic, and the user is not required to specify any criterion to terminate the model construction procedure. Comparisons with some of the existing state-of-art modeling methods are given, and several examples are included to demonstrate the ability of the proposed algorithm to effectively construct sparse models that generalize well.**

***Index Terms*—Bayesian learning, cross validation, orthogonal forward regression, predicted residual sums of squares (PRESS) statistic, regularization, sparse data modeling.**

I. Introduction

THE objective of modeling from data is not that the model simply fits the training data well. Rather, the goodness of a model is characterized by its generalization capability, interpretability and ease for knowledge extraction. Note that all these properties depend crucially on the ability to construct appropriate sparse models by the modeling process, and a basic principle in practical nonlinear data modeling is the parsimonious principle that ensures the smallest possible model that explains the training data. There exists a vast amount of works in the area of sparse modeling (e.g., [1]–[9]). Recently, a well-known sparse kernel modeling algorithm has perhaps been the support vector machine (SVM) method [8], which is widely regarded as the state-of-art technique for regression and classification applications. The formulation of SVM embodies the structural risk minimization principle, thus combining excellent generalization properties with a sparse model representation. Despite these attractive features and many good empirical results obtained using the SVM method, data modeling practicians have realized that the ability for the SVM method to produce sparse models has perhaps been overstated. This has motivated Tipping [9] to introduce the relevance vector machine (RVM) method.

The RVM method adopts a Bayesian learning framework [4]. The introduction of individual hyperparameters for every weights of the regression model is the key feature of the RVM method and is ultimately responsible for the sparsity properties of the RVM method [9]. An evidence procedure [4] is used to iteratively optimize kernel weights and the associated hyperparameters. During the optimization process, many of these hyperparameters are driven to large values so that the corresponding model weights are effectively forced to be zero and their associated model terms can then be removed from the trained model. The results given in [9] have demonstrated that the RVM has a comparable generalization performance to the SVM but requires dramatically fewer kernel functions or model terms than the SVM. A drawback of the RVM method is a significant increase in computational complexity, compared with the SVM method. A more serious limitation is however inherent in the evidence framework. The computation of the associated Hessian matrix required for updating hyperparameters is expensive, and this Hessian matrix may be near singular or singular and thus noninvertible. At a local minimum, some eigenvalues of this Hessian matrix may even be negative [10] and thus cause numerical instability for the iterative optimization procedure.

The orthogonal least squares (OLS) algorithm [1], which was developed in the late 1980s for nonlinear system modeling, remains popular for nonlinear data modeling practicians because the algorithm is simple and efficient and is capable of producing parsimonious linear-in-the-weights nonlinear models with good generalization performance. Over time, many "improved" variants of the OLS algorithm have been proposed [11]–[16]. In particular, the locally regularized OLS (LROLS) algorithm [13], [15] has been shown to be capable of producing very sparse regression models that generalize well. The key idea of the LROLS algorithm is in fact adopted from the RVM method, namely using the multiple regularizers to enforce sparsity. The LROLS algorithm is, however, based on the forward selection principle, has the ability to reveal the significance of individual model regressor, and only selects those significant terms, whereas the RVM method starts with the full model set and is effectively based on the backward elimination principle. It is well known that forward selection is computationally more attractive compared with backward elimination. More importantly, in the LROLS algorithm, only a subset matrix of the full Hessian matrix is used. This subset matrix is diagonal and well-conditioned with small eigenvalue spread. Therefore, the inverse of the Hessian is trivial, the

Manuscript received December 5, 2002; revised March 20, 2003. This paper was recommended by Associate Editor M. Berthold

S. Chen and C. J. Harris are with Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K.

X. Hong and P. M. Sharkey are with Department of Cybernetics, University of Reading, Reading RG6 6AY, U.K.

Digital Object Identifier 10.1109/TSMCB.2003.817107

regularization parameter updating is exact and simple, and the iterative procedure converges fast.

As in most model construction algorithms, the criterion used by the OLS algorithm in the model construction process is the training mean square error (MSE). Since the training MSE typically decreases as the model size increases, a separate stopping criterion is needed to stop the selection procedure in order to avoid an over-fitted model. For example, information-based criteria, such as the AIC and the minimum description length [17]–[19], can be adopted to terminate the model selection process. An information based criterion can be viewed as a model structure regularization by using a penalty term to penalize large-sized models. However, the penalty term in an information based criterion does not determine which model term should be selected. Multiple regularizers, i.e., local regularization [9], [13], [15], and optimal experimental design criteria [14] offer better solutions as model structure regularization as they are directly linked to model efficiency and parameter robustness. The underlying "problem," however, remains that the basic criterion for most model construction procedures is the training MSE. Arguably, a better and more natural approach is using a criterion of model generalization capability directly in the model selection procedure rather than only using it as a measure of model complexity.

The evaluation of model generalization capability is directly based on the concept of cross validation [20]. This paper investigates a model construction algorithm using a model selection criterion that is based explicitly on cross validation. This is achieved with a training data set only by utilizing the concept of delete-1 cross validation and the associated leave-one-out test error also known as the predicted residual sums of squares (PRESS) statistic [21]–[23]. The use of the leave-one-out estimate for general nonlinear-in-the-weights models has been studied in [24]–[26]. Even for the class of linear-in-the-weights models, computation of the mean square PRESS error is normally expensive and the use of the PRESS statistic in model selection is generally prohibitive. However, a recent study [27] has shown that, using the OLS algorithm, the calculation of the PRESS statistic becomes efficient and model selection based on the PRESS statistic is computationally affordable. It is well known that local regularization is effective in enforcing model sparsity as well as ensuring excellent generalization performance [9], [13], [15]. We combine the PRESS statistic with local regularization in the orthogonal forward regression procedure. The resulting algorithm selects a sparse model by incrementally minimizing a regularized mean square PRESS error.

Our motivation is twofold. First, we aim to derive a construction algorithm based directly on optimizing model generalization capability, without resorting to use a separate validation data set. We also want the model construction process to be automatic without the need for the user to specify some additional terminating criterion. The usual training MSE cannot achieve these objectives, but the PRESS statistic provides the capability to do so. Second, the level of sparsity and computational efficiency are also critical to the model construction process, and the LROLS algorithm (based on the training MSE) is known to offer considerable advantages in these two aspects. By combining the PRESS statistic with the LROLS algorithm, we obtain a truly automatic and computationally efficient construction algorithm capable of producing very sparse models with excellent generalization performance, using a training data set only. Several illustrative examples are included to illustrate the effectiveness of this approach. Comparisons with some of the existing state-of-art modeling methods are given, and the results demonstrate that our LROLS algorithm based on PRESS statistic compares favorably in terms of achieving the above-stated objectives.

II. Linear-in-the-Weights Regression Model

Consider the general discrete-time nonlinear system represented by the nonlinear model [28]

$$\begin{aligned} y(k) &= f(y(k-1),\ldots,y(k-n_y) \\ &\qquad u(k-1),\ldots,u(k-n_u)) + e(k) \\ &= f(\mathbf{x}(k)) + e(k) \end{aligned} \tag{1}$$

where $u(k)$ and $y(k)$ are the system input and output variables with integers n_u and n_y representing the lags in $u(k)$ and $y(k)$, respectively, $e(k)$ is a white noise, $\mathbf{x}(k) = [y(k-1)\ldots y(k-n_y)u(k-1)\ldots u(k-n_u)]^T$ denotes the system "input" vector, and $f(\cdot)$ is the unknown system mapping. The system (1) is to be identified from an N-sample observation data set $\mathcal{T}_N = \{\mathbf{x}(k), y(k)\}_{k=1}^N$ using some suitable functional that can approximate $f(\cdot)$ with arbitrary accuracy. One class of such functionals is the regression model of the form

$$\begin{aligned} y(k) = \hat{y}(k) + e(k) &= \sum_{i=1}^{n_M} \theta_i \phi_i(\mathbf{x}(k)) + e(k) \\ &= \boldsymbol{\phi}^T(k)\boldsymbol{\theta} + e(k) \end{aligned} \tag{2}$$

where $\hat{y}(k)$ denotes the model output, θ_i are the model weights, $\boldsymbol{\theta} = [\theta_1 \ldots \theta_{n_M}]^T$, $\phi_i(\mathbf{x}(k))$ are the regressors and $\boldsymbol{\phi}(k) = [\phi_1(k)\ldots\phi_{n_M}(k)]^T$ with $\phi_i(k) = \phi_i(\mathbf{x}(k))$, and n_M is the total number of candidate regressors. The model (2) is very general and includes all the kernel-based models, the polynomial-expansion model [1], and the general linear-in-the-weights nonlinear model [29]. In particular, for kernel-based models, the regressor $\phi_i(\mathbf{x}(k))$ takes the form

$$\phi_i(\mathbf{x}(k)) = \phi(\|\mathbf{x}(k) - \mathbf{c}_i\|), \tag{3}$$

where $\mathbf{c}_i$ is the kernel center, and $\phi(\cdot)$ is a given kernel function.

By letting $\boldsymbol{\phi}_i = [\phi_i(1) \; \cdots \; \phi_i(N)]^T$, for $1 \le i \le n_M$, and defining $\mathbf{y} = [y(1) \; \cdots \; y(N)]^T$, $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \; \cdots \; \boldsymbol{\phi}_{n_M}]$ and $\mathbf{e} = [e(1)\ldots e(N)]^T$, the regression model (2) can be written as

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\theta} + \mathbf{e}. \tag{4}$$

Let an orthogonal decomposition of the matrix $\boldsymbol{\Phi}$ be

$$\boldsymbol{\Phi} = \mathbf{W}\mathbf{A} \tag{5}$$

where

$$\mathbf{A} = \begin{bmatrix} 1, & a_{1,2} & \cdots & a_{1,n_M} \\ 0, & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n_M-1,n_M} \\ 0, & \cdots & 0 & 1 \end{bmatrix} \tag{6}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \quad \cdots \quad \mathbf{w}_{n_M}] \tag{7}$$

with columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \neq j$. The regression model (4) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\mathbf{g} + \mathbf{e} \tag{8}$$

where the orthogonal weight vector $\mathbf{g} = [g_1 \ \cdots \ g_{n_M}]^T$ satisfies the triangular system $\mathbf{A}\boldsymbol{\theta} = \mathbf{g}$. The space spanned by the original model bases $\phi_i(k)$, $1 \leq i \leq n_M$,] is identical to the space spanned by the orthogonal model bases $w_i(k)$, $1 \leq i \leq n_M$, and the model is equivalently expressed by

$$\hat{y}(k) = \mathbf{w}^T(k)\mathbf{g} \tag{9}$$

where $\mathbf{w}(k) = [w_1(k) \ \cdots \ w_{n_M}(k)]^T$.

III. OLS Algorithm Based on PRESS Statistic and Local Regularization

A. *Model Selection Using Cross Validation With PRESS Statistic*

Consider the model selection problem where a set of n_s models or predictors have been identified using the training data set $\mathcal{T}_N$. Denote these predictors, identified using all the N data points of $\mathcal{T}_N$, as $\hat{y}_j(k)$ with index $j = 1, 2, \ldots, n_s$. Cross validation using the stacked regression combines these models to achieve better generalization performance [22], [23]. Our aim here is to select a single parsimonious model with good generalization capability. To optimize generalization capability, cross validation is often used for model selection [20]. A commonly used cross validation is the delete-1 cross validation. The idea is as follows. For every predictor, each data point in the training set $\mathcal{T}_N$ is sequentially set aside in turn, a model is estimated using the remaining $N-1$ data points, and the prediction error is derived using only the data point that was removed from training. Specifically, let $\mathcal{T}_{N,-k}$ be the resulting data set by removing the kth data point from $\mathcal{T}_N$, and denote the jth model estimated using $\mathcal{T}_{N,-k}$ as $\hat{y}_{j,-k}(k)$ and the related predicted model residual at k as

$$\epsilon_{j,-k}(k) = y(k) - \hat{y}_{j,-k}(k). \tag{10}$$

The leave-one-out test error or the mean square PRESS error [21], [24] for the jth model $\hat{y}_j(k)$ is obtained by averaging all these prediction errors:

$$E\left[\epsilon_{j,-k}^2(k)\right] = \frac{1}{N}\sum_{k=1}^{N} \epsilon_{j,-k}^2(k). \tag{11}$$

To select the best predictor from the n_s candidate predictors $\hat{y}_j(k)$, $1 \leq j \leq n_s$, the same modeling procedure is applied to each of the n_s predictors, and the predictor with the minimum PRESS statistic is selected.

For linear-in-the-weights models, the PRESS statistic can be generated, without actually sequentially splitting the training data set and repeatedly estimating the associated models, by using the Sherman–Morrison–Woodbury theorem [21]. Consider that an n_M-term model $\hat{y}_{n_M}(k)$ is identified using $\mathcal{T}_N$ based on the model form of (2). The PRESS errors $\epsilon_{n_M,-k}(k)$ are calculated using [21], [24]

$$\begin{aligned} \epsilon_{n_M,-k}(k) &= y(k) - \hat{y}_{n_M,-k}(k) \\ &= \frac{\epsilon_{n_M}(k)}{1 - \boldsymbol{\phi}^T(k)(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\phi}(k)} \end{aligned} \tag{12}$$

where $\epsilon_{n_M}(k) = y(k) - \hat{y}_{n_M}(k)$. The computation of the PRESS error in (12) relies on the assumption that the regression matrix $\boldsymbol{\Phi}$ has full rank. Obviously, choosing the best subset model that minimizes the PRESS statistic quickly becomes computationally prohibitive even for a modest n_M-term model set. Moreover, the PRESS error (12) itself is computational expensive because the matrix inversion involved. However, if we choose only to incrementally minimize the PRESS statistic in an orthogonal forward regression manner, as presented in [27], the model selection procedure based on the PRESS statistic becomes computationally affordable. Furthermore, due to orthogonalization, the calculation of the PRESS errors becomes very efficient [27]. Note that the orthogonal forward selection procedure will always select a subset model such that the associated Hessian matrix is not only diagonal but well conditioned as well.

B. *LROLS Algorithm Based on PRESS Statistic*

The LROLS algorithm [13], [15] is based on the following regularized training error criterion

$$J_R(\mathbf{g}, \boldsymbol{\lambda}) = \mathbf{e}^T\mathbf{e} + \sum_{i=1}^{n_M} \lambda_i g_i^2 = \mathbf{e}^T\mathbf{e} + \mathbf{g}^T\boldsymbol{\Lambda}\mathbf{g} \tag{13}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \cdots \ \lambda_{n_M}]^T$ is the regularization parameter vector, and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_1, \ldots, \lambda_{n_M}\}$. The algorithm selects a subset model by incrementally minimizing the regularized training MSE. As is defined in [13] and [15], the regularized error reduction ratio due to the regressor $\mathbf{w}_i$ is given by

$$[\mathrm{rerr}]_i = \left(\mathbf{w}_i^T\mathbf{w}_i + \lambda_i\right) g_i^2 / \mathbf{y}^T\mathbf{y}. \tag{14}$$

At the nth stage of selection procedure, a model term is selected if it produces the largest regularized error reduction ratio among the remaining n to n_M candidates. The selection process is terminated at the n_sth stage if

$$1 - \sum_{l=1}^{n_s} [\mathrm{rerr}]_l < \xi \tag{15}$$

where $0 < \xi < 1$ is a user-specified tolerance. This produces a sparse model containing n_s $(\ll n_M)$ significant regressors.

The regularization parameters can be optimized iteratively using an evidence procedure [4], [9]. The following error criterion is obviously equivalent to the criterion (13):

$$J_B(\mathbf{g},\mathbf{h},\beta) = \beta\mathbf{e}^T\mathbf{e} + \sum_{i=1}^{n_M} h_i g_i^2 = \beta\mathbf{e}^T\mathbf{e} + \mathbf{g}^T\mathbf{H}\mathbf{g} \tag{16}$$

where β is the noise parameter (inverse of noise variance), $\mathbf{h} = [h_1 \ldots h_{n_M}]^T$ is the hyperparameter vector, and $\mathbf{H} = \text{diag}\{h_1,\ldots,h_{n_M}\}$. The relationship between a regularization parameter and its corresponding hyperparameter is given by

$$\lambda_i = \frac{h_i}{\beta}. \tag{17}$$

It can be shown that the log evidence for $\mathbf{h}$ and β is [4]

$$\begin{aligned}\log(\text{evidence}) &= \sum_{i=1}^{n_M}\frac{1}{2}\log(h_i) - \frac{N}{2}\log(\beta)\\ &\quad - \sum_{i=1}^{n_M}\frac{1}{2}h_i g_i^2 - \frac{1}{2}\beta\mathbf{e}^T\mathbf{e}\\ &\quad - \frac{1}{2}\log(\det(\mathbf{B})) + \text{constant}.\end{aligned} \tag{18}$$

Because of the orthogonalization, the Hessian matrix $\mathbf{B}$ is diagonal and is given by

$$\begin{aligned}\mathbf{B} &= \mathbf{H} + \beta\mathbf{W}^T\mathbf{W}\\ &= \text{diag}\left\{h_1 + \beta\mathbf{w}_1^T\mathbf{w}_1,\ldots,h_{n_M} + \beta\mathbf{w}_{n_M}^T\mathbf{w}_{n_M}\right\}.\end{aligned} \tag{19}$$

Setting the derivatives of log(evidence) with respect to $\mathbf{h}$ and β to zeros yields the updating formulas for $\mathbf{h}$ and β, respectively. Substituting these updating formulas into (17) results in the updating formulas for the regularization parameters:

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}}\frac{\mathbf{e}^T\mathbf{e}}{g_i^2}, \quad 1 \le i \le n_M \tag{20}$$

where

$$\gamma_i = \frac{\mathbf{w}_i^T\mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T\mathbf{w}_i} \quad \text{and} \quad \gamma = \sum_{i=1}^{n_M}\gamma_i. \tag{21}$$

Usually a few iterations (typically less than 10) are sufficient to find a (near) optimal $\boldsymbol{\lambda}$.

In the orthogonal forward selection procedure, if $\mathbf{w}_i^T\mathbf{w}_i$ is too small (near zero), this term will not be selected. This build-in mechanism automatically avoids any ill-conditioning situations. For the original OLS algorithm [1], the value of the user specified terminating scalar ξ is critical for avoiding an overfitted model. For the above LROLS algorithm based on training MSE, the choice of ξ is less critical, as is shown in [13] and [15]. Nevertheless, the user is still required to specify an appropriate value for ξ. The main component in the model selection criterion for the above algorithm is the training MSE. We can modify the model selection procedure so that the subset model selection is based entirely on the model generalization capability. Specifically, we use the PRESS statistic as the subset model selection criterion, whereas the model parameter estimate and the update of regularization parameters remain the same as the above LROLS algorithm.

Using the equivalent orthogonal model (9) and incorporating parameter regularization, the PRESS error is given by

$$\begin{aligned}\epsilon_{n_M,-k}(k) &= y(k) - \hat{y}_{n_M,-k}(k)\\ &= \frac{\epsilon_{n_M}(k)}{1 - \mathbf{w}^T(k)\left(\mathbf{W}^T\mathbf{W} + \boldsymbol{\Lambda}\right)^{-1}\mathbf{w}(k)} = \frac{\epsilon_{n_M}(k)}{\eta_{n_M}(k)}\end{aligned} \tag{22}$$

where the PRESS error weighting is given by

$$\eta_{n_M}(k) = 1 - \sum_{i=1}^{n_M}\frac{w_i^2(k)}{\mathbf{w}_i^T\mathbf{w}_i + \lambda_i}. \tag{23}$$

The derivation of (22) is similar to the case without regularization given in [27]. The mean square PRESS error for the model with a size n is then given by

$$J_n = E\left[\epsilon_{n,-k}^2(k)\right] = E\left[\left(\frac{\epsilon_n(k)}{\eta_n(k)}\right)^2\right] = \frac{1}{N}\sum_{k=1}^{N}\frac{\epsilon_n^2(k)}{\eta_n^2(k)}. \tag{24}$$

Note that the model residual $\epsilon_n(k)$ for the n-term model can be computed recursively as

$$\epsilon_n(k) = y(k) - \sum_{i=1}^{n} w_i(k)g_i = \epsilon_{n-1}(k) - w_n(k)g_n \tag{25}$$

and similarly, the PRESS error weighting $\eta_n(k)$ can be written in a recursive formula by

$$\eta_n(k) = 1 - \sum_{i=1}^{n}\frac{w_i^2(k)}{\mathbf{w}_i^T\mathbf{w}_i + \lambda_i} = \eta_{n-1}(k) - \frac{w_n^2(k)}{\mathbf{w}_n^T\mathbf{w}_n + \lambda_n}. \tag{26}$$

The recursive formulas (25) and (26) enable an efficient computation of the PRESS statistic (24).

The subset model selection procedure can be modified as follows. At the nth stage of the selection procedure, a model term is selected among the remaining n to n_M candidates if the resulting n-term model produces the smallest mean square PRESS error. It has been shown in [27] that the PRESS statistic J_n is concave with respect to the model size n. That is, there exists an "optimal" model size n_s such that for $n \le n_s$, J_n decreases as n increases, whereas for $n \ge n_s + 1$, J_n increases as n increases. Thus, the selection procedure is automatically terminated with an n_s-term model when $J_{n_s+1} > J_{n_s}$, without the need for the user to specify a separate termination criterion. The iterative model selection procedure based on the LROLS algorithm with PRESS statistic can now be summarized.

1) Initialization: Set λ_i, $1 \le i \le n_M$, to the same small positive value (e.g., 0.000 01). Set iteration $I = 1$.

Step 1) Given the current $\boldsymbol{\lambda}$ and with the following initial conditions:

$$\epsilon_0(k) = y(k) \quad \text{and} \quad \eta_0(k) = 1, \quad k = 1, 2, \ldots, N,$$

$$J_0 = \frac{1}{N}\mathbf{y}^T\mathbf{y} = \frac{1}{N}\sum_{k=1}^{N} y^2(k) \tag{27}$$

use the procedure described in the Appendix to select a subset model with n_I terms.

Step 2) Update $\boldsymbol{\lambda}$ using (20) and (21) with $n_M = n_I$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a preset maximum iteration number is reached, stop; otherwise, set $I+ = 1$, and go to Step 1.

The requirements of computing the PRESS statistic in the selection process represent a considerable complexity increase, compared with the LROLS algorithm using the regularized training MSE. However, the computational complexity of the proposed algorithm is clearly affordable, due to the orthogonal forward regression and, in particular, the efficient recursive calculation of the PRESS errors. The main advantage of the proposed algorithm is that the model selection is directly based on the model generalization capability. Therefore, even without regularization, the algorithm, which is first derived in [27], is capable of producing sparse models with excellent generalization performance. The local regularization can often further enforce sparsity to derive much sparser models in many practical modeling situations. These observations will be illustrated later on by some modeling examples.

IV. Comparison with Some Existing Model Construction Algorithms

Several model construction algorithms are used for a comparison with the proposed LROLS algorithm with PRESS statistic, and they are the OLS algorithm with PRESS statistic [27], the LROLS algorithm with regularized training MSE [13], [15], the RVM algorithm [9], and the enhanced κ-means clustering and least squares (CLS) algorithm [30], [31]. The OLS with PRESS statistic and the LROLS with PRESS statistic are both automatic construction algorithms using only a training data set. It should be pointed out that the computational complexity of the LROLS algorithm is not necessarily significantly more than that of the OLS algorithm, even though the former involves an iterative loop. This is because typically after the first iteration, which has a complexity of the OLS algorithm, the model set contains only n_1 $(\ll n_M)$ terms, and the complexity of the subsequent iteration decreases dramatically. The LROLS algorithm with training MSE has a complexity less than that of the LROLS algorithm with PRESS statistic, and it is also an automatic construction algorithm using only a training data set, provided that an appropriate value for ξ can be specified. When an appropriate ξ cannot be found, it may then be necessary to use other terminating criteria, such as employing an additional validation data set.

The RVM algorithm [9] shares certain common features with the LROLS algorithm, as they both use an approach of multiple regularizers to enforce sparsity and adopt a similar evidence procedure for updating hyperparameters or regularization parameters. It is therefore not surprising that the generalization capabilities and the levels of sparsity provided by the two algorithms are similar. The LROLS algorithm, however, has considerable computational advantages in that it can operate robustly in difficult modeling conditions, and its iterative loop generally converges faster compared with the RVM (see the discussion in [15]). Using the equivalent regularization formula, the RVM for regression [9] can be reformulated to involve an iterative loop of the model weight estimation and regularization parameter updating. With given $\boldsymbol{\lambda}$, the model weight estimate is the regularized least squares (LS) solution:

$$\boldsymbol{\theta} = \tilde{\mathbf{B}}^{-1}\boldsymbol{\Phi}^T\mathbf{y} \tag{28}$$

where the Hessian matrix $\tilde{\mathbf{B}}$ is given by

$$\tilde{\mathbf{B}} = \boldsymbol{\Phi}^T\boldsymbol{\Phi} + \boldsymbol{\Lambda}. \tag{29}$$

The regularization parameters are updated using

$$\lambda_i^{\text{new}} = \frac{\tilde{\gamma}_i^{\text{old}}}{N - \tilde{\gamma}^{\text{old}}}\frac{\mathbf{e}^T\mathbf{e}}{\theta_i^2}, \quad 1 \le i \le n_M \tag{30}$$

where

$$\tilde{\gamma} = \sum_{i=1}^{n_M}\tilde{\gamma}_i \quad \text{with } \tilde{\gamma}_i = 1 - \lambda_i\bar{b}_{i,i} \tag{31}$$

and $\bar{b}_{i,i}$ denotes the ith diagonal element of $\tilde{\mathbf{B}}^{-1}$. The RVM starts with the full model set $\boldsymbol{\Phi}$ and removes those regressors that have large values in their associated regularization parameters. Clearly, the inverse of $\tilde{\mathbf{B}}$ is expensive and this matrix may be ill-conditioned or even singular. Compared with the LROLS algorithm, the regularization parameter updating is much more expensive, and the iterative procedure generally converges with slower rate and may suffer from numerical instability. Model pruning in the RVM method is done by specifying a large threshold $\lambda_{\max}$. If a model term $\boldsymbol{\phi}_i$ with its associated regularization parameter λ_i satisfying $\lambda_i > \lambda_{\max}$, it is removed. During the iterative procedure, $\lambda_{\max}$ may need to be reduced gradually in order to derive an appropriate final sparse model. Provided that $\lambda_{\max}$ can be set appropriately and numerical instability does not occur, the RVM algorithm can provide a very sparse model with excellent generalization performance using only a training data set.

Kernel based models belong to a special case of the general linear-in-the-weights model (2). Typically, each training input data $\mathbf{x}(k)$ is fitted with a kernel function, and a sparse representation is then sought. This is the approach adopted by the OLS, SVM, and RVM methods. An alternative approach is to use a clustering algorithm to partition the training input data $\{\mathbf{x}(k)\}_{k=1}^N$ into clusters and use the cluster centers for the kernel centers. The related model weights can then be solved using the usual LS method. Early works adopting this approach (e.g., [32], [33]) used the κ-means clustering [34] to seek the cluster prototypes. The traditional κ-means clustering algorithm can only achieve a local optimal solution, which depends on the initial locations of cluster centers. A consequence of this local optimality is that some initial centers can become stuck in regions of the input domain with few or no input patterns and never move to where they are needed. An improved κ-means clustering algorithm was proposed in [35], which overcomes the above-mentioned drawback. By using a cluster variation-weighted measure, this enhanced κ-means partitioning process always achieves an optimal center configuration in the sense that after convergence all clusters have an equal cluster variance. The enhanced CLS algorithm [30], [31] adopts this partitioning process to seek the kernel centers $\mathbf{c}_i, 1 \leq i \leq n_s$. The enhanced κ-means clustering algorithm [35] is summarized as follows:

$$\mathbf{c}_i(k+1) = \mathbf{c}_i(k) + M_i(\mathbf{x}(k))(\eta_c(\mathbf{x}(k) - \mathbf{c}_i(k))) \quad (32)$$

where $0 < \eta_c < 1.0$ is a learning rate, the membership function $M_i(\mathbf{x}(k))$ is defined as

$$M_i(\mathbf{x}) = \begin{cases} 1, & \text{if } v_i\|\mathbf{x} - \mathbf{c}_i\|^2 \leq v_l\|\mathbf{x} - \mathbf{c}_l\|^2 \quad \text{for all} l \neq i \\ 0, & \text{otherwise} \end{cases} \quad (33)$$

and v_i is the variation or "variance" of the ith cluster. To estimate variation v_i, the following updating rule is used:

$$v_i(k+1) = \alpha_c v_i(k) + (1 - \alpha_c)(M_i(\mathbf{x}(k))\|\mathbf{x}(k) - \mathbf{c}_i(k)\|^2), \quad (34)$$

where α_c is a constant slightly less than 1.0. The initial variations $v_i(0), 1 \leq i \leq n_s$ are set to the same small number. Note that the learning rate can be self-adjusting based on an "entropy" formula [35]

$$\eta_c = 1 - H(\bar{v}_1, \ldots, \bar{v}_{n_s})/\log(n_s) \quad (35)$$

where

$$H(\bar{v}_1, \ldots, \bar{v}_{n_s}) = \sum_{i=1}^{n_s} -\bar{v}_i \log(\bar{v}_i) \quad \text{with } \bar{v}_i = \frac{v_i}{\sum_{l=1}^{n_s} v_l}. \quad (36)$$

Given the set of kernel centers $\mathbf{c}_i, 1 \leq i \leq n_s$, the kernel model in the form (2) can be formed, and the model weight vector is readily given by the LS solution $\boldsymbol{\theta} = (\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\mathbf{y}$. The enhanced CLS algorithm has a very low complexity. However, the algorithm itself does not provide the required number of cluster prototypes or model terms for adequately modeling the data, i.e., it cannot determine the model structure. We will adopt a practical strategy of having a separate validation data set at a cost of increasing complexity. A range of models with different model sizes n_s are fitted to the training data set, and the MSE values of the fitted models are computed over the validation set. The model with a size n_s^*, which has the smallest test MSE, is selected.

V. Modelling Examples

Example 1: This example used a radial basis function (RBF) network to model the scalar function

$$f(x) = \frac{\sin(x)}{x}, \quad -10 \leq x \leq 10. \quad (37)$$

Four hundred training data were generated from $y = f(x) + e$, where the input x was uniformly distributed in $(-10, 10)$, and the noise e was Gaussian with zero mean and standard deviation 0.2. The first 200 data points were used for training and the other 200 samples for model validation. The RBF model employed the Gaussian kernel function of the form

$$\phi_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma^2}\right) \quad (38)$$

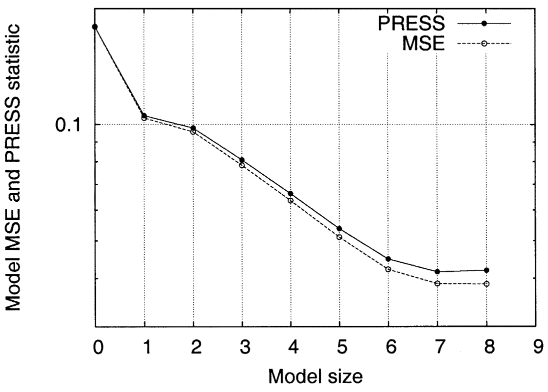


Fig. 1. Evolution of training MSE and PRESS statistic versus model size for simple scalar function modeling problem using the OLS algorithm based on PRESS statistic without the help of a validation set.

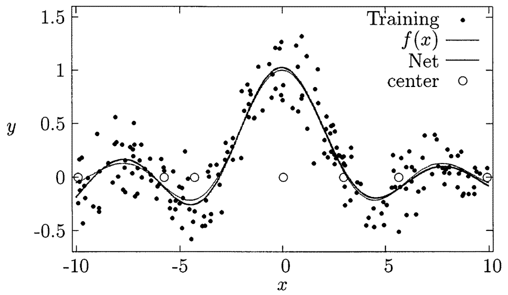


Fig. 2. Simple scalar function modeling problem. (Dots) Noisy training data y. (Thin curve) Underlying function $f(x)$. (Thick curve) Model mapping. (Circles) Selected RBF centers. The seven-term model was identified by the OLS algorithm based on PRESS statistic without the help of a validation set.

where $\mathbf{c}_i$ was the ith RBF center vector and σ^2 the kernel variance. For this example, $\sigma^2 = 10.0$ was found to be optimal empirically and was used for all the models. As each training data x was considered as a candidate RBF center, there were $n_M = 200$ regressors in the regression model (2). The training data were very noisy. Two hundred noise-free data $f(x)$ with equally spaced x in $(-10, 10)$ were also generated as an additional testing data set for evaluating model performance.

We first applied the OLS algorithm based on PRESS statistic. Fig. 1 depicts the evolution of the training MSE and PRESS statistic in log scale during the forward regression procedure with a typical set of noisy training data set. It can be seen that the PRESS statistic continuously decreased until $J_8 = 0.041\,940 > J_7 = 0.041\,590$, and the algorithm terminated with a seven-term model. The training MSE, the mean square PRESS error, and the MSEs over the noisy and noise-free testing sets, respectively, for the constructed seven-term model are summarized in Table I. Fig. 2 shows the noisy training points y and the underlying function $f(x)$ together with the mapping generated using the model identified by the OLS algorithm based on PRESS statistic. It can be seen from the results shown in Table I and Fig. 2 that the OLS algorithm based on PRESS statistic automatically identified a very sparse model from the noisy training data set with excellent generalization capability.

Next, we applied the LROLS algorithm with PRESS statistic to the same noisy training data set. After the first iteration, the model set contained seven candidates, and subsequent iterations did not reduce the model set any further. After ten iterations, the

TABLE I
COMPARISON OF MODELING ACCURACY FOR THE SIMPLE SCALAR FUNCTION MODELING

algorithm	validation set used	model size	training MSE	PRESS statistic	testing MSE	MSE over noise-free testing set
OLS with PRESS	No	7	0.038762	0.041590	0.042097	0.000887
LROLS with PRESS	No	7	0.038792	0.039064	0.042001	0.000736
LROLS with MSE	No	8	0.038879	0.041650	0.042310	0.000829
RVM	No	15	0.038784	0.041565	0.041827	0.000668
CLS	Yes	7	0.039110	0.041968	0.041257	0.000719

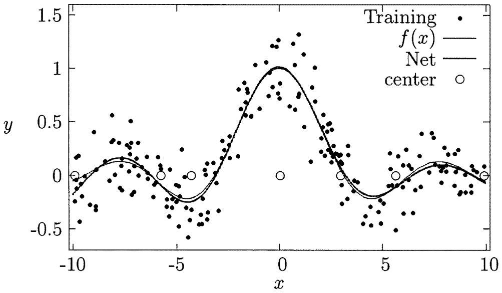


Fig. 3. Simple scalar function modeling problem. (Dots) Noisy training data y. (Thin curve) Underlying function $f(x)$. (Thick curve) Model mapping. (Circles) Selected RBF centers. The seven-term model was identified by the LROLS algorithm based on PRESS statistic without the help of a validation set.

regularization parameters were considered to have converged, and the modeling accuracy of the resulting seven-term model is also summarized in Table I. The corresponding model mapping generated by this seven-term model is depicted in Fig. 3.

It is informative to examine the selection process of the LROLS algorithm based on training MSE. At the first iteration, the model selection procedure stopped at the 18th stage, when it detected that adding one more term would cause the problem to be singular or very ill-conditioned. The model set after the first iteration thus contained 17 terms. The model, after $\boldsymbol{\lambda}$ had converged (ten iterations), is listed in Table II. It can be seen from Table II that the regularization parameters related to the ninth to 17th terms were all very large and the associated model weights were effectively zero. This clearly indicated an eight-term model. The modeling accuracy of this eight-term model is summarized in Table I, and the corresponding model mapping is illustrated in Fig. 4. For this example, the role of terminating threshold ξ was not critical at all, and the local regularization enabled the selection of a very sparse model from the noisy training data set with excellent generalization capability.

For the RVM algorithm, the iterative process was observed to converge slower, and 50 iterations were required. The pruning threshold was initially set to $\lambda_{\max} = 1.0 \times 10^6$, which was subsequently reduced to 500.0 at the tenth iteration, to 4.5 at the 30th iteration, and to 0.05 at the 40th iteration. The construction process produced a 15-term model as listed in Table III, and the modeling accuracy of this model can be seen in Table I. Fig. 5 depicts the model mapping generated by this 15-term model, where it can be observed that some of the selected centers were very close to each other and a seven-term model was in fact possible, but there existed no mechanism within the RVM algorithm to find this very sparse seven-term model that would have the same excellent generalization performance as the 15-term model shown in Table III.

TABLE II
SELECTION PROCEDURE OF THE LROLS ALGORITHM BASED ON TRAINING MSE FOR THE SIMPLE SCALAR FUNCTION MODELING AFTER $\boldsymbol{\lambda}$ HAS CONVERGED (TEN ITERATIONS)

model term l	weight θ_l	regularizer λ_l
1	4.98536e+00	1.44989e-01
2	-6.69768e+00	9.15759e-01
3	-4.95720e+00	1.58502e-01
4	5.71989e+00	2.82009e-02
5	2.57559e+00	3.19889e-02
6	-1.23981e+00	3.52239e-02
7	-3.57695e-01	3.21713e-01
8	3.77955e-02	9.54683e-01
9	-3.45849e-03	8.34312e+01
10	-2.97624e-03	1.85563e+01
11	-4.61672e-04	3.31509e+00
12	5.96767e-19	3.81729e+15
13	1.82911e-20	4.29842e+16
14	-5.76015e-21	4.30654e+14
15	9.18710e-22	2.45111e+14
16	-1.64138e-60	3.08374e+55
17	-2.89962e-62	3.52990e+55

MSE over training set: 0.038879
PRESS statistic: 0.041650
MSE over noisy testing set: 0.042310
MSE over noise-free testing set: 0.000829

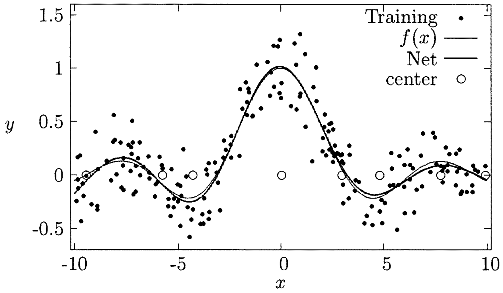


Fig. 4. Simple scalar function modeling problem. (Dots) Noisy training data y. (Thin curve) Underlying function $f(x)$. (Thick curve) Model mapping. (Circles) Selected RBF centers. The eight-term model was identified by the LROLS algorithm based on the training MSE without the help of a validation set.

For the enhanced CLS algorithm, the adaptive constant was set to $\alpha_c = 0.96$, and the clustering algorithm was passed through the training data set 50 times to ensure convergence. A separate validation set was required to help determining the model structure, and Fig. 6 shows the training and testing MSE values over the training and validation sets, respectively, versus the model size n_s. The result shown in Fig. 6 clearly indicated a seven-term model. The model accuracy of the resulting seven-term model is summarized in Table I, and the corresponding model mapping is illustrated in Fig. 7.

TABLE III
MODEL CONSTRUCTED BY THE RVM ALGORITHM FOR THE SIMPLE SCALAR FUNCTION MODELING AFTER $\boldsymbol{\lambda}$ HAS CONVERGED (50 ITERATIONS)

center	weight θ_l	regularizer λ_l
4.07517e+00	-2.26930e+00	9.12305e-03
-6.12343e+00	1.66785e+00	7.35661e-03
-3.58204e+00	-1.74771e+00	3.98148e-03
-5.98057e+00	2.49459e-01	4.60483e-02
-3.63497e+00	-1.14174e+00	6.04091e-03
-9.90796e+00	-1.07007e+00	3.44794e-02
9.93232e+00	-9.43063e-01	4.26291e-02
-3.52529e+00	-1.21626e+00	5.74984e-03
-3.51687e+00	-1.07321e+00	6.51856e-03
6.79055e+00	7.50657e-01	2.56213e-02
-3.48645e+00	-5.87322e-01	1.19164e-02
6.89690e+00	4.99379e-01	3.83332e-02
-1.16057e+00	4.66568e+00	1.84250e-03
6.83757e+00	7.61092e-01	2.52520e-02
-6.08119e+00	1.37302e+00	8.78291e-03

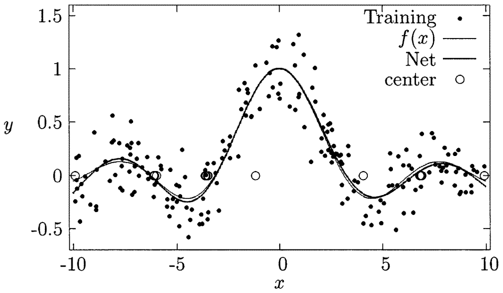


Fig. 5. Simple scalar function modeling problem. (Dots) Noisy training data y. (Thin curve) Underlying function $f(x)$. (Thick curve) Model mapping. (Circles) Selected RBF centers. The 15-term model was identified by the RVM algorithm without the help of a validation set.

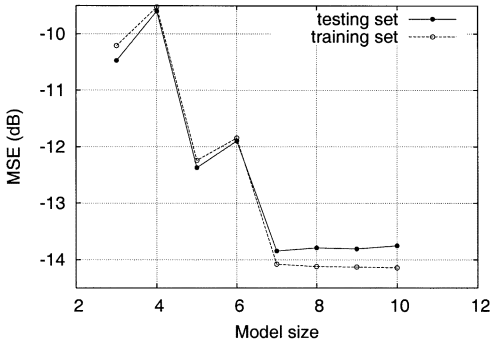


Fig. 6. Training and testing MSE values over the training and validation sets, respectively, versus model size for simple scalar function modeling problem using the enhanced CLS algorithm with the help of a validation set.

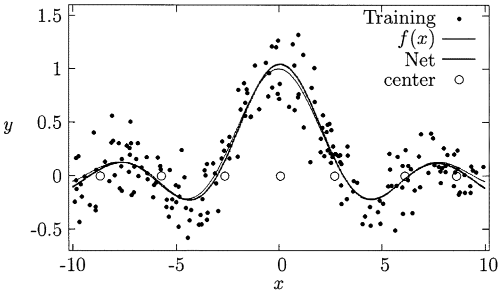


Fig. 7. Simple scalar function modeling problem. (Dots) Noisy training data y. (Thin curve) Underlying function $f(x)$. (Thick curve) Model mapping. (Circles) Selected RBF centers. The seven-term model was identified by the enhanced CLS algorithm with the help of a validation set.

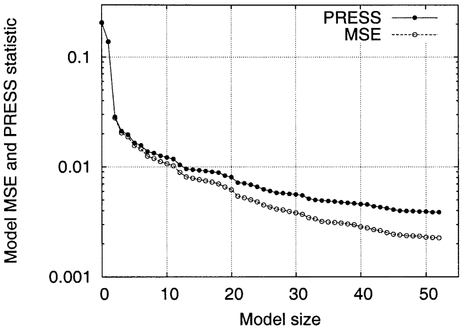


Fig. 8. Evolution of training MSE and PRESS statistic versus model size for simulated control system modeling problem using the OLS algorithm based on PRESS statistic without the help of a validation set.

Example 2: This was a simulated nonlinear dynamic control system considered in [36]. The underlying dynamic system was governed by (39), shown at the bottom of the next page, where the system input $u(k)$ was a random signal uniformly distributed in the interval $[-1, 1]$. The noisy system output was given by $y(k) = z(k)+e(k)$, where the noise $e(k)$ was Gaussian with zero mean and standard deviation 0.05. Four hundred noisy samples were generated. The first 200 data points were used for training, and the other 200 samples were used for model validation. A RBF network with the thin-plate-spline basis function

$$\phi_i(\mathbf{x}(k)) = ||\mathbf{x}(k) - \mathbf{c}_i||^2 \log(||\mathbf{x}(k) - \mathbf{c}_i||) \tag{40}$$

and the input vector

$$\mathbf{x}(k) = [y(k-1)y(k-2)y(k-3)u(k-1)u(k-2)]^T \tag{41}$$

was used to construct a model from the noisy training data set. As each training data point $\mathbf{x}(k)$ was considered as a candidate RBF center, there were $n_M = 200$ candidate regressors.

Fig. 8 illustrates the evolution of the training MSE and PRESS statistic for the OLS algorithm based on PRESS statistic, where it can be seen that the PRESS statistic continuously decreased until $J_{52} = 0.003\,870 > J_{51} = 0.003\,864$, and the algorithm terminated with a 51-term model. For the LROLS algorithm based on PRESS statistic, the model set was reduced to a size of 51 after the first iteration, and a constant size of 31 terms was reached after a few iterations. The final 31-term model was produced after 20 iterations. For the LROLS algorithm based on training MSE, it was found that as

$$z(k) = \frac{z(k-1)z(k-2)z(k-3)u(k-2)(z(k-3)-1) + u(k-1)}{1 + z^2(k-2) + z^2(k-3)} \tag{39}$$

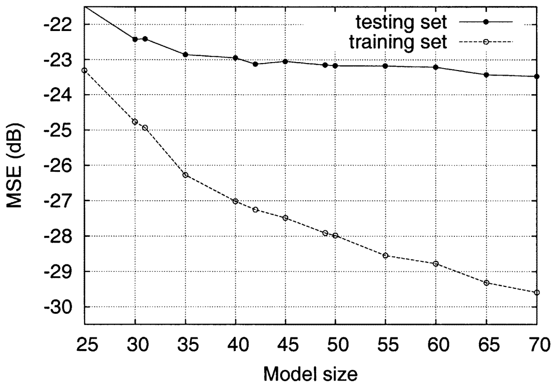


Fig. 9. Training and testing MSE values over the training and validation sets, respectively, versus subset model size for simulated control system modeling problem using the LROLS algorithm based on training MSE with the help of a validation set.

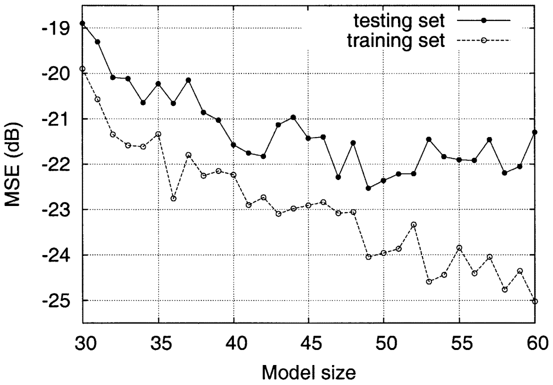


Fig. 10. Training and testing MSE values over the training and validation sets, respectively, versus model size for simulated control system modeling problem using the enhanced CLS algorithm with the help of a validation set.

more terms were added the training MSE kept decreasing and the corresponding regularization parameters all had reasonable small values. Thus, it was difficult to determine an adequate sparse model by specifying an appropriate value for ξ, and it was decided to use the validation data set to help the model construction. Fig. 9 depicts the training and testing MSE values versus the subset model size, and the result appeared to suggest a 42-term subset model. For the RVM algorithm, 60 iterations were used, and the pruning threshold was initially set to $\lambda_{\max} = 1000.0$ which was subsequently reduced to 0.25 at the 50th iteration. With this choice of $\lambda_{\max}$, the RVM method was able to construct a 42-term model without the need of employing the validation set. For the enhanced CLS algorithm, $\alpha_c = 0.96$ was used with 60 passes of the training data set for clustering. The model structure determination with the help of the validation set is depicted in Fig. 10, where a decision was made to choose the 49-term model.

The five models produced by the five algorithms are compared in Table IV. The constructed RBF model $\hat{f}_{\mathrm{RBF}}(\cdot)$ was used to iteratively generate the model output according to

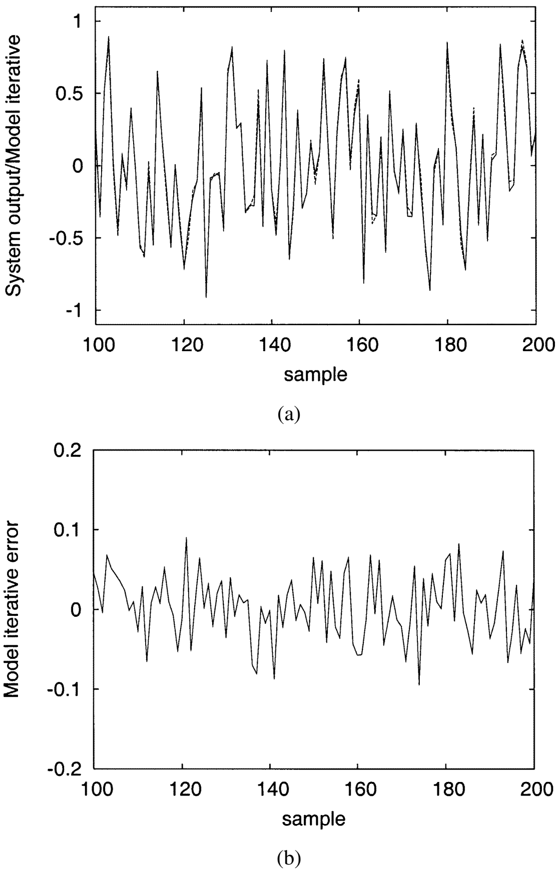
$$\hat{y}_d(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}_d(k)) \tag{42}$$

with the input vector $\mathbf{x}_d(k)$ given by

$$\mathbf{x}_d(k) = [\hat{y}_d(k-1)\hat{y}_d(k-2)\hat{y}_d(k-3)u(k-1)u(k-2)]^T. \tag{43}$$

TABLE IV
COMPARISON OF MODELING ACCURACY FOR THE SIMULATED CONTROL SYSTEM MODELING.

algorithm	validation set used	model size	training MSE	PRESS statistic	testing MSE
OLS with PRESS	No	51	0.002280	0.003864	0.005187
LROLS with PRESS	No	31	0.003192	0.003706	0.005892
LROLS with MSE	Yes	42	0.001883	0.003067	0.004872
RVM	No	42	0.001598	0.002577	0.004935
CLS	Yes	49	0.003940	0.007607	0.005580



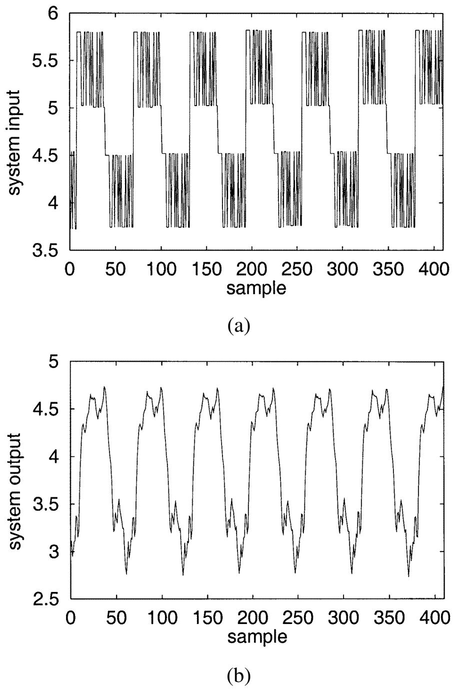
(a)

(b)

Fig. 11. Modeling performance for simulated control system modeling problem. (a) Iterative model output $\hat{y}_d(k)$ (dashed) superimposed on system output $y(k)$ (solid). (b) Iterative model error $\epsilon_d(k)$. The 42-term model was constructed by the RVM algorithm without the help of a validation set.

Fig. 11 plots the iterative model output $\hat{y}_d(k)$ and error $\epsilon_d(k) = y(k) - \hat{y}_d(k)$ for the 42-term model constructed by the RVM algorithm. The other four model responses, which are not shown here, were all similar to the results shown in Fig. 11. The results demonstrate that the five constructed models were adequate and had similar generalization capability. For this example, the OLS with PRESS, the LROLS with PRESS, and the RVM were able to automatically construct sparse models without the help of a validation set, and in particular, the LROLS algorithm based on PRESS statistic resulted in a much sparser model, compared with the other four algorithms.

Example 3: This example constructed a model representing the relationship between the fuel rack position (input $u(k)$) and the engine speed (output $y(k)$) for a Leyland TL11 turbocharged, direct-injection diesel engine operated at low engine speed. It is known that at low engine speed, the relationship between the input and output is nonlinear [37]. Detailed system description and experimental setup can be found in [37]. The



(a)

(b)

Fig. 12. Engine data set (a) input $u(k)$ and (b) output $y(k)$.

data set, which is depicted in Fig. 12, contained 410 samples. The first 210 data points were used in modeling and the last 200 points in model validation. A RBF model with the input vector

$$\mathbf{x}(k) = [y(k-1)u(k-1)u(k-2)]^T \tag{44}$$

and the Gaussian basis function of variance 1.69 was used to model the data. As each $\mathbf{x}(k)$ in the training data set was considered as a candidate RBF center, there were $n_M = 210$ candidate regressors. From Fig. 12, it is seen that a strong periodic component was presented in the data, and this was believed to have caused numerical problem for the RVM method during the modeling construction.

Fig. 13 shows the evolution of the training MSE and PRESS statistic for the OLS algorithm based on PRESS statistic. For this example, the algorithm resulted in a sparse 22-term model. For the LROLS algorithm based on PRESS statistic, the model set contained 22 terms after the first iteration, and the subsequent iterations did not reduce the model set any further. The final 22-term model was obtained after ten iterations. For the LROLS algorithm based on training MSE, during the first iteration, the model selection procedure stopped at the 55th stage to avoid the singular or very ill-conditioning problem. By examining the 54-term model obtained after ten iterations, it was seen that the 35th to 54th terms had large values of λ_i associated with them. Thus, the algorithm was able to produce a sparse 34-term model using only the training data set and without the need to specify an appropriate value for ξ. Note that if the validation set was used to help determining the model structure, a sparser model with 23 terms was obtained with the same generalization performance as the 34-term model. For the enhanced CLS algorithm, the clustering algorithm passed through the training data set 50 times with $\alpha_c = 0.96$. The model structure determination with the aid of the validation set is illustrated in Fig. 14, and the result suggested a 23-term model.

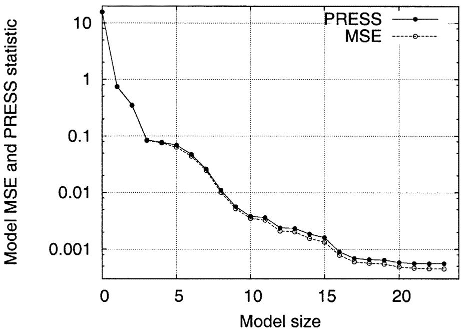


Fig. 13. Evolution of training MSE and PRESS statistic versus model size for engine data set modeling problem using the OLS algorithm based on PRESS statistic without the help of a validation set.

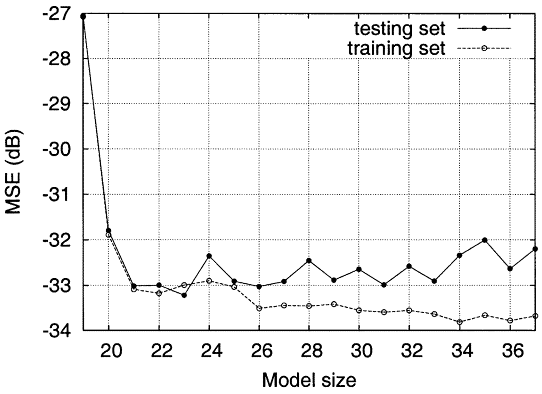


Fig. 14. Training and testing MSE values over the training and validation sets, respectively, versus model size for engine data set modeling problem using the enhanced CLS algorithm with the help of a validation set.

The modeling accuracies of the four resulting models are compared in Table V, where it can be seen that the four models have similarly good generalization performance. The constructed RBF model $\hat{f}_{\mathrm{RBF}}(\cdot)$ was used to generate the model prediction according to

$$\hat{y}(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}(k)) \tag{45}$$

with the input vector $\mathbf{x}(k)$ given by (44). Fig. 15 depicts the model prediction $\hat{y}(k)$ and the prediction error $\epsilon(k) = y(k) - \hat{y}(k)$ for the 22-term model constructed by the LROLS algorithm based on PRESS statistic. The other three models have similar prediction performance to the results shown in Fig. 15.

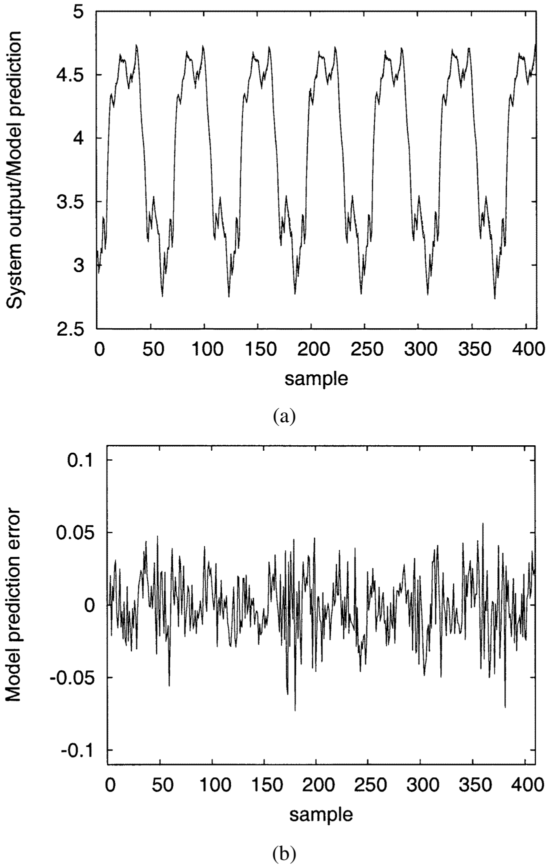
For this example, the RVM algorithm as implemented in the form given in Section IV failed to work due to numerical instability of the iterative loop for updating regularization parameters. Various initial values for $\boldsymbol{\lambda}$ were tried and a more stable updating formula for $\boldsymbol{\lambda}$

$$\lambda_i^{\mathrm{new}} = (1-\eta)\lambda_i^{\mathrm{old}} + \eta \cdot \frac{\tilde{\gamma}_i^{\mathrm{old}}}{N - \tilde{\gamma}^{\mathrm{old}}} \frac{\mathbf{e}^T\mathbf{e}}{\theta_i^2}, \quad 1 \le i \le n_M \tag{46}$$

was also used, but the iterative loop for updating $\boldsymbol{\lambda}$ was unstable. This numerical instability caused the algorithm to force every regularization parameters to take very large values, which was the root of failure. It is conceivable that the RVM method implemented with some other more robust form may still work well in this situation. However, the results shown here serve to highlight a potentially inherent instability of the RVM method,

TABLE V
COMPARISON OF MODELING ACCURACY FOR THE ENGINE DATA SET

algorithm	validation set used	model size	training MSE	PRESS statistic	testing MSE
OLS with PRESS	No	22	0.000452	0.000555	0.000484
LROLS with PRESS	No	22	0.000453	0.000457	0.000490
LROLS with MSE	No	34	0.000439	0.000643	0.000485
CLS	Yes	23	0.000502	0.000629	0.000477



(a)

(b)

Fig. 15. Modeling performance for engine data set modeling problem. (a) Model prediction $\hat{y}(k)$ (dashed) superimposed on system output $y(k)$ (solid). (b) Model prediction error $\epsilon(k)$. The 22-term model was constructed by the LROLS algorithm based on PRESS statistic without the help of a validation set.

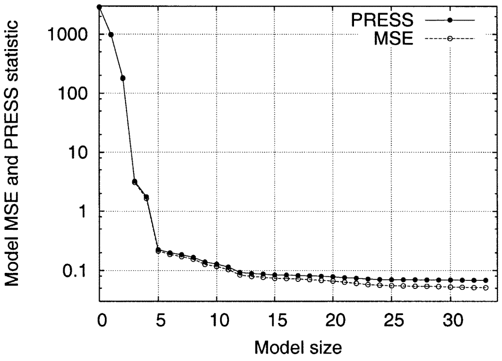


Fig. 16. Evolution of training MSE and PRESS statistic versus model size for gas furnace data set modeling problem by the OLS algorithm based on PRESS statistic using the training data set.

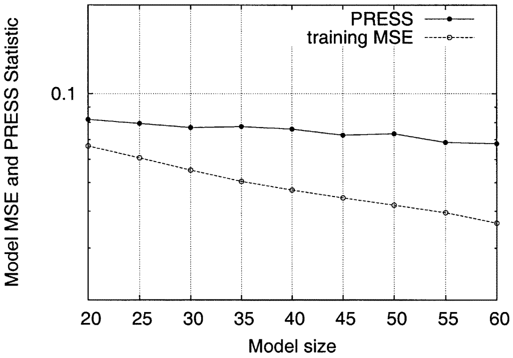


Fig. 17. Evolution of training MSE and PRESS statistic versus model size for gas furnace data set modeling problem by the LROLS algorithm based on training MSE using the training data set.

which can affect the algorithm's performance in adverse modeling environments.

Example 4: This example constructed a model for the gas furnace data set [38, ser. J]. The data set contained 296 pairs of input–output points, where 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. A RBF network with the input vector

$$\mathbf{x}(k) = [y(k-1)y(k-2)y(k-3) \\ \times u(k-1)u(k-2)u(k-3)]^T \quad (47)$$

and the thin-plate-spline basis function was used to fit the data set. The number of candidate regressors in the regression model (2) was $n_M = 296$.

The OLS algorithm based on PRESS statistic terminated the model construction with a sparse 32-term model when the condition $J_{33} = 0.068\,218 > J_{32} = 0.068\,215$ was reached, and Fig. 16 shows the evolution of the training MSE and PRESS statistic during the orthogonal forward regression procedure. The LROLS algorithm based on PRESS statistic with 20 iterations was able to obtain a sparser model of 28 terms. The LROLS algorithm based on training MSE was unable to automatically determine an adequate sparse model without being given an appropriate stopping threshold ξ. Since there was no validation set, an attempt was made using the PRESS statistic to help determining the model structure, and Fig. 17 illustrates the evolution of the training MSE and PRESS statistic for the LROLS algorithm based on training MSE. From Fig. 17, it was difficult to decide how many terms should be included in the constructed model but a decision was made nevertheless to use the 40-term model. For the RVM algorithm, 60 iterations were involved. By choosing the pruning threshold $\lambda_{\max} = 1000.0$ initially and subsequently reducing it to $\lambda_{\max} = 10.0$ at the 50th iteration, the RVM algorithm was able to automatically construct a 40-term model using only the training set. For the enhanced CLS algorithm, the clustering algorithm passed through

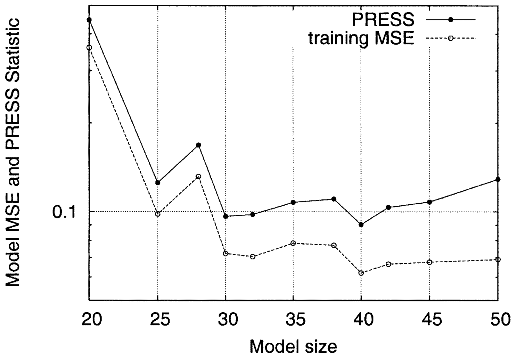


Fig. 18. Evolution of training MSE and PRESS statistic versus model size for gas furnace data set modeling problem by the enhanced CLS algorithm using the training data set.

TABLE VI
COMPARISON OF MODELING ACCURACY FOR THE GAS FURNACE DATA SET

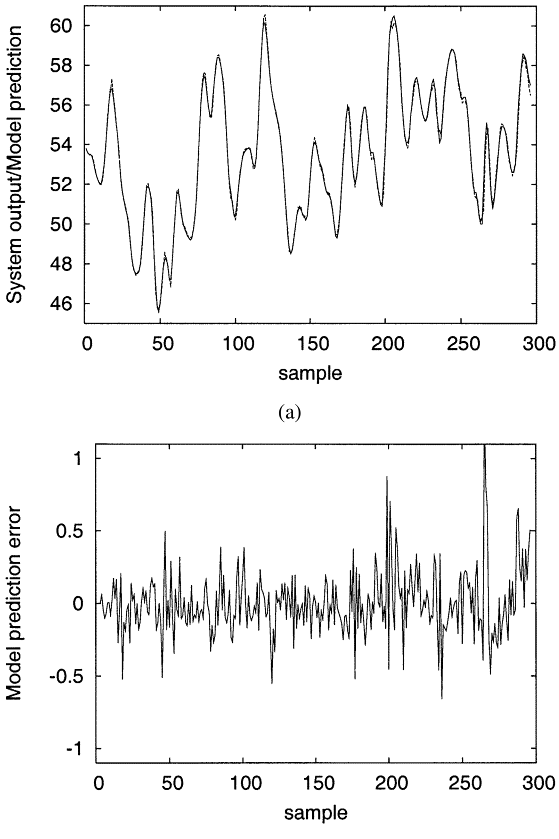
algorithm	model size	training MSE	PRESS statistic
OLS with PRESS	32	0.051273	0.068215
LROLS with PRESS	28	0.053306	0.053685
LROLS with MSE	40	0.047240	0.075832
RVM	40	0.035090	0.053345
clustering and LS	40	0.062032	0.090396

the training set 100 times with $\alpha_c = 0.99$. The model structure determination, which is illustrated in Fig. 18, could only be carried out with the help of PRESS statistic since there was no validation data set. The results shown in Fig. 18 suggested a 40-term model.

The five resulting models are compared in Table VI, where it can be seen that the model constructed by the enhanced CLS algorithm is slightly inferior to the other four models. The constructed RBF model $\hat{f}_{\mathrm{RBF}}(\cdot)$ was used to generate the model prediction according to $\hat{y}(k) = \hat{f}_{\mathrm{RBF}}(\mathbf{x}(k))$ with the input vector $\mathbf{x}(k)$ given by (47). Fig. 19 shows the model prediction $\hat{y}(k)$ and the prediction error $\epsilon(k) = y(k) - \hat{y}(k)$ for the 28-term model constructed by the LROLS algorithm based on PRESS statistic.

VI. CONCLUSION

A novel approach has been considered for sparse data modeling using linear-in-the-weights nonlinear models based directly on optimizing the model generalization capability. This has been achieved by adopting a delete-1 cross validation method and utilizing an efficient computation of the associated leave-one-out test error also known as the PRESS statistic based on an orthogonal forward regression procedure. It has been shown that incorporating a local regularization into the model selection procedure can often further enforce sparsity. The model construction process is fully automated, and the user does not need to specify any stopping criterion for terminating the model selection process. Several modeling examples have been included to demonstrate the ability of the proposed approach to construct sparse models that generalize well. A comparison with some of the existing state-of-art linear-in-the-weights modeling methods has been given.



(a)

(b)

Fig. 19. Modeling performance for gas furnace data set modeling problem. (a) Model prediction $\hat{y}(k)$ (dashed) superimposed on system output $y(k)$ (solid). (b) Model prediction error $\epsilon(k)$. The 28-term model was constructed by the LROLS algorithm based on PRESS statistic using only the training data set.

APPENDIX

The modified Gram–Schmidt orthogonalization procedure [1] calculates the $\mathbf{A}$ matrix row by row and orthogonalizes $\boldsymbol{\Phi}$ as follows: At the lth stage, make the columns $\boldsymbol{\phi}_j$, $l+1 \leq j \leq n_M$, orthogonal to the lth column, and repeat the operation for $1 \leq l \leq n_M - 1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)} = \boldsymbol{\phi}_j$, $1 \leq j \leq n_M$, then for $l = 1, 2, \ldots, n_M - 1$

$$\left.\begin{aligned} \mathbf{w}_l &= \boldsymbol{\phi}_l^{(l-1)} \\ a_{l,j} &= \mathbf{w}_l^T \boldsymbol{\phi}_j^{(l-1)} \Big/ \left(\mathbf{w}_l^T \mathbf{w}_l\right), \quad l+1 \leq j \leq n_M \\ \boldsymbol{\phi}_j^{(l)} &= \boldsymbol{\phi}_j^{(l-1)} - a_{l,j}\mathbf{w}_l, \quad l+1 \leq j \leq n_M. \end{aligned}\right\} \tag{48}$$

The last stage of the procedure is simply $\mathbf{w}_{n_M} = \boldsymbol{\phi}_{n_M}^{(n_M-1)}$. The elements of $\mathbf{g}$ are computed by transforming $\mathbf{y}^{(0)} = \mathbf{y}$ in a similar way:

$$\left.\begin{aligned} g_l &= \mathbf{w}_l^T \mathbf{y}^{(l-1)} \Big/ \left(\mathbf{w}_l^T \mathbf{w}_l + \lambda_l\right), \\ \mathbf{y}^{(l)} &= \mathbf{y}^{(l-1)} - g_l \mathbf{w}_l, \end{aligned}\right\} \quad 1 \leq l \leq n_M. \tag{49}$$

This orthogonalization scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner. First, define

$$\boldsymbol{\Phi}^{(l-1)} = \left[\mathbf{w}_1 \ldots \mathbf{w}_{l-1} \boldsymbol{\phi}_l^{(l-1)} \ldots \boldsymbol{\phi}_{n_M}^{(l-1)}\right]. \tag{50}$$

If some of the columns $\boldsymbol{\phi}_l^{(l-1)}, \ldots, \boldsymbol{\phi}_{n_M}^{(l-1)}$ in $\boldsymbol{\Phi}^{(l-1)}$ have been interchanged, this will still be referred to as $\boldsymbol{\Phi}^{(l-1)}$ for notational convenience. With the initial conditions as specified in (27), the lth stage of the selection procedure is given as follows.

1) For $l \leq j \leq n_M$, compute

$$g_l^{(j)} = \left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \mathbf{y}^{(l-1)} \Big/ \left(\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)} + \lambda_j\right)$$

$$\left.\begin{aligned} \epsilon_l^{(j)}(k) &= y^{(l-1)}(k) - \phi_j^{(l-1)}(k) g_l^{(j)} \\ \eta_l^{(j)}(k) &= \eta_{l-1}(k) - \frac{\left(\phi_j^{(l-1)}(k)\right)^2}{\left(\boldsymbol{\phi}_j^{(l-1)}\right)^T \boldsymbol{\phi}_j^{(l-1)} + \lambda_j}, \end{aligned}\right\} \quad k = 1, \ldots, N$$

$$J_l^{(j)} = \frac{1}{N} \sum_{k=1}^{N} \left(\frac{\epsilon_l^{(j)}(k)}{\eta_l^{(j)}(k)}\right)^2$$

where $y^{(l-1)}(k)$ and $\phi_j^{(l-1)}(k)$ are the kth elements of $\mathbf{y}^{(l-1)}$ and $\boldsymbol{\phi}_j^{(l-1)}$, respectively.

2) Find

$$J_l = J_l^{(j_l)} = \min\left\{J_l^{(j)}, l \leq j \leq n_M\right\}.$$

Then, the j_lth column of $\mathbf{\Phi}^{(l-1)}$ is interchanged with the lth column of $\mathbf{\Phi}^{(l-1)}$, the j_lth column of $\mathbf{A}$ is interchanged with the lth column of $\mathbf{A}$ up to the $(l-1)$th row, and the j_lth element of $\boldsymbol{\lambda}$ is interchanged with the lth element of $\boldsymbol{\lambda}$. This effectively selects the j_lth candidate as the lth regressor in the subset model.

3) The selection procedure is terminated with a $(l-1)$-term model if $J_l > J_{l-1}$. Otherwise, perform the orthogonalization as indicated in (48) to derive the lth row of $\mathbf{A}$ and to transform $\mathbf{\Phi}^{(l-1)}$ into $\mathbf{\Phi}^{(l)}$; calculate g_l, and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (49); update the PRESS error weightings

$$\eta_l(k) = \eta_{l-1}(k) - \frac{w_l^2(k)}{\mathbf{w}_l^T \mathbf{w}_l + \lambda_l}, \quad k = 1, 2, \ldots, N$$

and go to Step 1.

References

[1] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their application to nonlinear system identification," *Int. J. Contr.*, 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, pp. 302–309, May 1991.

[3] J. H. Friedman, "Multivariate adaptive regression splines," *Ann. Statist.*, vol. 19, no. 1, pp. 1–141, 1991.

[4] D. J. C. MacKay, "Bayesian interpolation," *Neural Comput.*, vol. 4, no. 3, pp. 415–447, 1992.

[5] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. London, U.K.: Chapman and Hall, 1993.

[6] T. Kavli, "ASMOD: An algorithm for adaptive spline modeling of observation data," *Int. J. Contr.*, vol. 58, no. 4, pp. 947–968, 1993.

[7] M. Brown and C. J. Harris, *Neurofuzzy Adaptive Modeling and Control*. Englewood Cliffs, NJ: Prentice-Hall, 1994.

[8] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[9] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Res.*, vol. 1, pp. 211–244, 2001.

[10] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford, U. K.: Oxford Univ. Press, 1995.

[11] S. Chen, E. S. Chng, and K. Alkadhimi, "Regularised orthogonal least squares algorithm for constructing radial basis function networks," *Int. J. Contr.*, vol. 64, pp. 829–837, Sept. 1996.

[12] 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, pp. 1239–1243, Sept. 1999.

[13] S. Chen, "Locally regularised orthogonal least squares algorithm for the construction of sparse kernel regression models," in *Proc. 6th Int. Conf. Signal Processing*, vol. 2, Beijing, China, Aug. 26–30, 2002, pp. 1229–1232.

[14] X. Hong and C. J. Harris, "Nonlinear model structure design and construction using orthogonal least squares and D-optimality design," *IEEE Trans. Neural Networks*, vol. 13, pp. 1245–1250, Sept. 2002.

[15] S. Chen, "Local regularization assisted orthogonal least squares regression," , submitted for publication.

[16] X. Hong and C. J. Harris, "A neurofuzzy network knowledge extraction and extended Gram-Schmidt algorithm for model subspace decomposition," IEEE Trans. Fuzzy Syst., vol. 11, pp. 528–541, Aug. 2003, to be published.

[17] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. AC-19, pp. 716–723, 1974.

[18] I. J. Leontaritis and S. A. Billings, "Model selection and validation methods for nonlinear systems," *Int. J. Contr.*, vol. 45, no. 1, pp. 311–341, 1987.

[19] M. H. Hansen and B. Yu, "Model selection and the principle of minimum description length," *J. Amer. Statist. Assoc.*, vol. 96, no. 454, pp. 746–774, 2001.

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

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

[22] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[23] L. Breiman, "Stacked regressions," *Machine Learning*, vol. 24, pp. 49–64, 1996.

[24] L. K. Hansen and J. Larsen, "Linear unlearning for cross-validation," *Adv. Comput. Math.*, vol. 5, pp. 269–280, 1996.

[25] G. Monari and G. Dreyfus, "Withdrawing an example from the training set: An analytic estimation of its effect on a nonlinear parameterised model," *Neurocomput.*, vol. 35, pp. 195–201, 2000.

[26] ——, "Local overfitting control via leverages," *Neural Comput.*, vol. 14, pp. 1481–1506, 2002.

[27] X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic," *Proc. Inst. Elect. Eng. Contr. Theory Applicat.*, vol. 150, no. 3, pp. 245–254, 2003.

[28] S. Chen and S. A. Billings, "Representation of nonlinear systems: The NARMAX model," *Int. J. Contr.*, vol. 49, no. 3, pp. 1013–1032, 1989.

[29] S. A. Billings and S. Chen, "Extended model set, global data and threshold model identification of severely nonlinear systems," *Int. J. Contr.*, vol. 50, no. 5, pp. 1897–1923, 1989.

[30] 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.

[31] S. A. Billings and S. Chen, "The determination of multivariable nonlinear models for dynamic systems," in *Control Dynamic Systems, Neural Network Systems Techniques and Applications*, C. T. Leondes, Ed. San Diego, CA: Academic, 1998, vol. 7, pp. 231–278.

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

[33] S. Chen, S. A. Billings, and P. M. Grant, "Recursive hybrid algorithm for nonlinear system identification using radial basis function networks," *Int. J. Contr.*, vol. 55, pp. 1051–1070, 1992.

[34] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[35] C. Chinrungrueng and C. H. Séquin, "Optimal adaptive κ-means algorithm with dynamic adjustment of learning rate," *IEEE Trans. Neural Networks*, vol. 6, pp. 157–169, Jan. 1995.

[36] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, Jan. 1990.

[37] 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.

[38] G. E. P. Box and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*. San Francisco, CA: Holden Day, 1976.

Sheng Chen (SM'97) received the B.Eng. degree in control engineering from the East China Petroleum Institute, Dongying, China, in 1982 and the Ph.D. degree in control engineering from the City University, London, U.K., in 1986.

He joined the Department of Electronics and Computer Science, University of Southampton, Southampton, U.K., in September 1999. He previously held research and academic appointments at the Universities of Sheffield, Sheffield,, U.K., Edinburgh, Edinburgh, U.K., and Portsmouth, Portsmouth, U.K. His recent research works include adaptive nonlinear signal processing, modeling and identification of nonlinear systems, neural network research, finite-precision digital controller design, evolutionary computation methods, and optimization. He has published over 200 research papers.

Xia Hong (SM'02) received the B.Sc. and M.Sc. degrees from National University of Defense Technology, Changsha, China, in 1984 and 1987, respectively, and the Ph.D. degree from the University of Sheffield, Sheffield, U.K., in 1998, all in automatic control.

She worked as a research assistant at the Beijing Institute of Systems Engineering, Beijing, China, from 1987 to 1993. She worked as a research fellow with the Department of Electronics and Computer Science, University of Southampton, Southampton, U.K., from 1997 to 2001. She is currently a lecturer at the Department of Cybernetics, the University of Reading, Reading, U.K. She is actively engaged in research into neurofuzzy systems, data modeling, and learning theory and their applications. Her research interests include system identification, estimation, neural networks, intelligent data modeling, and control. She has published over 30 research papers and co-authored a research book.

Dr. Hong received a Donald Julius Groen Prize from IMechE, U.K., in 1999.

Chris J. Harris receiving the B.Sc. degree from the University of Leicester, Leicester, U.K., the M.A. degree from the University of Oxford, Oxford, U.K., and the Ph.D. degree from the University of Southampton, Southampton, U.K.

He previously held appointments at the Universities of Hull, Hull, U.K., UMIST, Manchester, U.K., Oxford, and Cranfield, Cranfield, U.K., as well as being employed by the U.K. Ministry of Defense, U.K. He returned to the University of Southampton as the Lucas Professor of Aerospace Systems Engineering in 1987 to establish the Advanced Systems Research Group and, more recently, ISIS. His research interests lie in the general area of intelligent and adaptive systems theory and its application to intelligent autonomous systems such as autonomous vehicles, management infrastructures such as command and control, intelligent control, and estimation of dynamic processes, multi-sensor data fusion, and systems integration. He has authored and co-authored 12 research books and over 300 research papers, and he is the associate editor of numerous international journals.

Dr. Harris was elected to the Royal Academy of Engineering in 1996, was awarded the IEE Senior Achievement model in 1998 for his work in autonomous systems, and the IEE Faraday medal in 2001 (the highest international award in the IEE) for his work in intelligent control and neurofuzzy systems.

Paul M. Sharkey received the Hons. Dip.E.E. and the B.Sc. (Eng.) degree from the University of Dublin, Trinity College, Dublin, Ireland, both in 1985, and the Ph.D. degree from the University of Strathclyde, Glasgow, U.K., in 1988.

He was a Research Engineer at the Robotics research group of the University of Oxford, Oxford, U.K., before joining the Department of Cybernetics, University of Reading, Reading, U.K., in 1993. Currently, he is a Professor of cybernetics, Head of Department, and Chair of the Interactive Systems Research Group. He has research interests in control systems, robotics, vision, and virtual reality. Since 1998, he has been the Programme Chair for the biennial international conference series on disability, virtual reality, and associated technologies.