Sparse Kernel Density Construction Using Orthogonal Forward Regression With Leave-One-Out Test Score and Local Regularization

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

***Abstract*—This paper presents an efficient construction algorithm for obtaining sparse kernel density estimates based on a regression approach that directly optimizes model generalization capability. Computational efficiency of the density construction is ensured using an orthogonal forward regression, and the algorithm incrementally minimizes the leave-one-out test score. A local regularization method is incorporated naturally into the density construction process to further enforce sparsity. An additional advantage of the proposed algorithm is that it is fully automatic and the user is not required to specify any criterion to terminate the density construction procedure. This is in contrast to an existing state-of-art kernel density estimation method using the support vector machine (SVM), where the user is required to specify some critical algorithm parameter. Several examples are included to demonstrate the ability of the proposed algorithm to effectively construct a very sparse kernel density estimate with comparable accuracy to that of the full sample optimized Parzen window density estimate. Our experimental results also demonstrate that the proposed algorithm compares favorably with the SVM method, in terms of both test accuracy and sparsity, for constructing kernel density estimates.**

***Index Terms*—Cross validation, leave-one-out test score, orthogonal least squares, Parzen window estimate, probability density function, regularization, sparse kernel modeling.**

I. Introduction

ESTIMATION of probability density functions is a recurrent theme in machine learning and many fields of engineering, see for example [1]–[4]. A well-known nonparametric density estimation technique is the classical Parzen window estimate [5], which is remarkably simple and accurate. The particular problem associated with the Parzen window estimate however is the computational cost for testing which scales directly with the sample size, as the Parzen window estimate employs the full data sample set in defining a density estimate for subsequent observations. In today's data-rich environment, this can be a serious problem in practical applications. Recently, the support vector machine (SVM) has been proposed as a promising tool for sparse kernel density estimation [6], [7]. The motivation of the SVM density estimation comes from the claim that the SVM can effectively perform function approximations in high dimensional spaces from finite data with sparse representations. Although this effectiveness has been demonstrated in regression and classification problems, it is known that there are alternative methods for regression and classification [8], [9], which can provide sparser representations than the SVM method. Currently, the machine learning community is actively engaged in the investigation of the SVM density estimation method.

A recent Ph.D. dissertation [10] has proposed an interesting greedy technique for kernel density estimation. This technique constructs sparse kernel density estimates using an orthogonal forward regression (OFR) that incrementally minimizes the training mean square error (MSE) [11]. This sparse density construction algorithm is computationally simple and efficient, and the results given in [10] have demonstrated the potential of this method. One critical aspect of this method, which is less satisfactory, is in when to terminate the density construction procedure. The minimum descriptive length [12] and Akaike's information criterion [13] were first suggested to help terminate the density construction process, but the empirical results showed that models obtained were still often oversized. At the end, a maximum model size was imposed in order to avoid an over-fit model. Motivated by the promising result in [10] and our previous work on sparse data modeling [14]–[16], we propose an efficient construction algorithm for sparse kernel density estimation using the OFR based on the leave-one-out (LOO) test score and local regularization. Specifically, we extend the regression model construction algorithm [16] to the construction of sparse kernel density estimates. We will refer to our proposed algorithm as the sparse density construction (SDC) algorithm.

Our motivation is twofold. Firstly, we aim to derive sparse kernel density estimates based on optimizing model generalization capability or test performance. We also want the kernel density construction process to be automatic without the need for the user to specify some additional termination criterion. The usual training MSE cannot achieve these objectives, but the delete-one cross validation with its associated LOO test score [17]–[20] provides the capability to achieve this aim, without resorting to use a separate validation data set. Secondly, the level of sparsity and computational efficiency are also critical

Manuscript received December 23, 2003; revised March 18, 2004. This paper was recommended by Associate Editor D. D. Nauck.

S. Chen and C. J. Harris are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: sqc@ecs.soton.ac.uk).

X. Hong is with Department of Cybernetics, University of Reading, Reading RG6 6AY, U.K.

Digital Object Identifier 10.1109/TSMCB.2004.828199

to the kernel density construction process. The computational efficiency of using the delete-one cross validation is ensured by using the orthogonal least squares algorithm [21], [22], as is first shown in [20], and multiple-regularizers or local regularization is known to be capable of providing very sparse solutions [8], [14]–[16]. Our previous work on sparse regression modeling [16] has shown that the OFR based on the LOO test score and local regularization offers considerable advantages in realizing these two critical objectives of sparse modeling over several other state-of-art methods. The current investigation shows that the proposed SDC method inherits these crucial advantages. Compared with the SVM method, our SDC algorithm is simpler to implement and has no critical algorithm parameter that needs to be specified by the user. Several examples are used to illustrate the ability of this new SDC algorithm to construct efficiently a sparse density estimate with comparable accuracy to that of the Parzen window estimate. Some examples that have been used in the existing literature to investigate the SVM method are specifically chosen in order to compare the performance of our SDC algorithm with the SVM density estimation method. Our experimental results demonstrate that the SDC algorithm offers a viable alternative to the SVM method for constructing sparse and accurate kernel density estimates.

II. Kernel Density Estimation as Regression

Consider the finite sample set $\mathcal{D} = \{\mathbf{x}_k\}_{k=1}^N$ drawn from a density $p(\mathbf{x})$, where the data samples $\mathbf{x}_k = [x_{1,k}\ x_{2,k}\ \cdots\ x_{m,k}]^T \in \mathcal{R}^m$ are assumed to be independently identically distributed. The task is to estimate the unknown density $p(\mathbf{x})$ using the kernel density estimate of the form

$$\hat{p}(\mathbf{x}) = \sum_{k=1}^{N} \beta_k K(\mathbf{x}, \mathbf{x}_k) \tag{1}$$

with the constraints

$$\beta_k \geq 0, \quad k = 1, 2, \ldots, N \tag{2}$$

and

$$\sum_{k=1}^{N} \beta_k = 1. \tag{3}$$

In this study, the kernel function is assumed to be the Gaussian function of the form

$$K(\mathbf{x}, \mathbf{x}_k) = \frac{1}{(2\pi\rho^2)^{m/2}} \exp\left(-\frac{||\mathbf{x} - \mathbf{x}_k||^2}{2\rho^2}\right) \tag{4}$$

where ρ is a common kernel width. The well-known Parzen window estimate [5] is obtained by setting $\beta_k = 1/N$ for all k. Our aim is to seek a spare representation for $\hat{p}(\mathbf{x})$, i.e., with most of β_k being zero and yet maintaining a comparable test performance or generalization capability to that of the full sample Parzen window estimate having an optimized value for ρ.

A density $p(\mathbf{x})$ is defined as the solution of

$$\int_{-\infty}^{\mathbf{x}} p(\mathbf{u})\, d\mathbf{u} = f_p(\mathbf{x}) \tag{5}$$

subject to the constraints

$$\int_{-\infty}^{\infty} p(\mathbf{u})\, d\mathbf{u} = 1 \tag{6}$$

and

$$p(\mathbf{x}) \geq 0 \tag{7}$$

where $f_p(\cdot)$ is the unknown cumulative distribution function corresponding to the density $p(\cdot)$. Given the data set $\mathcal{D}$, the empirical distribution function $f(\mathbf{x}; N)$ defined by

$$f(\mathbf{x}; N) = \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{m} \theta(x_j - x_{j,k}) \tag{8}$$

with

$$\theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{9}$$

is known to be a good approximation to the true distribution function $f_p(\mathbf{x})$ [6], [7]. Thus, the kernel density estimation problem can be posed as the following regression modeling problem [6], [7], [10]:

$$f(\mathbf{x}; N) = \sum_{k=1}^{N} \beta_k q(\mathbf{x}, \mathbf{x}_k) + \epsilon(\mathbf{x}) \tag{10}$$

subject to the constraints (2) and (3), where the "regressor" $q(\mathbf{x}, \mathbf{x}_k)$ is given by

$$\begin{aligned} q(\mathbf{x}, \mathbf{x}_k) &= \int_{-\infty}^{\mathbf{x}} K(\mathbf{u}, \mathbf{x}_k)\, d\mathbf{u} \\ &= \prod_{j=1}^{m} \left(1 - Q\left(\frac{x_j - x_{j,k}}{\rho}\right)\right) \end{aligned} \tag{11}$$

with

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} \exp\left(-\frac{u^2}{2}\right) du \tag{12}$$

and $\epsilon(\mathbf{x})$ denotes the modeling error at $\mathbf{x}$.

Define $\boldsymbol{\beta} = [\beta_1\ \beta_2\ \cdots\ \beta_N]^T$, $f_k = f(\mathbf{x}_k; N)$, and $\boldsymbol{\phi}(k) = [q_{k,1}\ q_{k,2}\ \cdots\ q_{k,N}]^T$ with $q_{k,i} = q(\mathbf{x}_k, \mathbf{x}_i)$. Then the regression model (10) for the data point $\mathbf{x}_k \in \mathcal{D}$ can be expressed as

$$f_k = \hat{f}_k + \epsilon(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\beta} + \epsilon(k) \tag{13}$$

where $\epsilon(k) = \epsilon(\mathbf{x}_k)$. Furthermore, the regression model (10) over the training data set $\mathcal{D}$ can be written together in the matrix form

$$\mathbf{f} = \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{14}$$

with the following additional notations $\boldsymbol{\Phi} = [q_{i,k}] \in \mathcal{R}^{N \times N}$, with $1 \leq i, k \leq N$, $\boldsymbol{\epsilon} = [\epsilon(1)\ \epsilon(2)\ \ldots\ \epsilon(N)]^T$, and $\mathbf{f} = [f_1\ f_2\ \ldots\ f_N]^T$. For convenience, we will denote the regression matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \ldots\ \boldsymbol{\phi}_N]$ with $\boldsymbol{\phi}_k = [q_{1,k}\ q_{2,k}\ \cdots\ q_{N,k}]^T$. $\boldsymbol{\phi}_k$ should not be confused with $\boldsymbol{\phi}(k)$ (the former is the kth column of $\boldsymbol{\Phi}$, and the latter the kth row of $\boldsymbol{\Phi}$). Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}$ be

$$\boldsymbol{\Phi} = \mathbf{W}\mathbf{A} \tag{15}$$

where

$$\mathbf{A} = \begin{bmatrix} 1, & a_{1,2} & \cdots & a_{1,N} \\ 0, & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0, & \cdots & 0 & 1 \end{bmatrix} \tag{16}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \cdots \quad \mathbf{w}_N] \tag{17}$$

with columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \neq j$. The regression model (14) can alternatively be expressed as

$$\mathbf{f} = \mathbf{W}\mathbf{g} + \boldsymbol{\epsilon} \tag{18}$$

where the weight vector $\mathbf{g} = [g_1\ g_2\ \ldots\ g_N]^T$ associated with the orthogonal space $\mathbf{W}$ satisfies the triangular system $\mathbf{A}\boldsymbol{\beta} = \mathbf{g}$. The space spanned by the original model bases $\boldsymbol{\phi}_i$, $1 \leq i \leq N$, is identical to the space spanned by the orthogonal model bases $\mathbf{w}_i$, $1 \leq i \leq N$, and the model $\hat{f}_k$ is equivalently expressed by

$$\hat{f}_k = \mathbf{w}^T(k)\mathbf{g} \tag{19}$$

where $\mathbf{w}(k) = [w_{k,1}\ w_{k,2}\ \ldots\ w_{k,N}]^T$ is the kth row of $\mathbf{W}$.

In general, the "regression" matrix $\boldsymbol{\Phi}$ in (14) may be ill-conditioned or even noninvertible, particularly for a large data set. This can cause numerical problems for some density construction algorithms, but not the proposed SDC algorithm. This is because the OFR automatically avoids any ill-conditioning problems and selects a subset matrix of $\boldsymbol{\Phi}$ that is well-conditioned.

III. Sparse Density Construction

In the OFR algorithm based on the LOO test score and local regularization [16], the weight parameter vector $\mathbf{g}$ is the regularized least squares solution obtained by minimizing the following regularized error criterion:

$$J_R(\mathbf{g}, \boldsymbol{\lambda}) = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \sum_{i=1}^{N} \lambda_i g_i^2 \tag{20}$$

where $\boldsymbol{\lambda} = [\lambda_1\ \lambda_2\ \ldots\ \lambda_N]^T$ is the regularization parameter vector, which is optimized based on the evidence procedure [23] with the iterative updating formulas [15], [16]

$$\lambda_i^{\text{new}} = \frac{\gamma_i^{\text{old}}}{N - \gamma^{\text{old}}} \frac{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}{g_i^2}, \quad 1 \leq i \leq N \tag{21}$$

where

$$\gamma_i = \frac{\mathbf{w}_i^T\mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T\mathbf{w}_i} \quad \text{and} \quad \gamma = \sum_{i=1}^{N} \gamma_i. \tag{22}$$

Usually a few iterations (typically less than ten) are sufficient to find a local optimal $\boldsymbol{\lambda}$. The criterion (20) has its root in the Bayesian learning framework. For the completeness, this Bayesian interpretation of $J_R(\mathbf{g}, \boldsymbol{\lambda})$ together with the derivation of the updating formulas (21) and (22) are summarized in Appendix A.

An OFR procedure is used to construct a sparse density estimate by incrementally minimizing the LOO test score. Assume that an n-term model is selected from the full model (18). Then the LOO test error [17]–[20], denoted as $\epsilon_{n,-k}(k)$, for the selected n-term model can be shown to be [16], [20]

$$\epsilon_{n,-k}(k) = \frac{\epsilon_n(k)}{\eta_n(k)} \tag{23}$$

where $\epsilon_n(k)$ is the n-term modeling error and $\eta_n(k)$ is the associated LOO error weighting given by

$$\eta_n(k) = 1 - \sum_{i=1}^{n} \frac{w_{k,i}^2}{\mathbf{w}_i^T\mathbf{w}_i + \lambda_i}. \tag{24}$$

The mean square LOO error for the model with a size n is defined by

$$J_n = E\left[\epsilon_{n,-k}^2(k)\right] = \frac{1}{N}\sum_{k=1}^{N} \frac{\epsilon_n^2(k)}{\eta_n^2(k)}. \tag{25}$$

This LOO test score can be computed efficiently due to the fact that the n-term model error $\epsilon_n(k)$ and the associated LOO error weighting can be calculated recursively according to

$$\epsilon_n(k) = f_k - \sum_{i=1}^{n} w_{k,i} g_i = \epsilon_{n-1}(k) - w_{k,n} g_n \tag{26}$$

and

$$\begin{aligned} \eta_n(k) &= 1 - \sum_{i=1}^{n} \frac{w_{k,i}^2}{\mathbf{w}_i^T\mathbf{w}_i + \lambda_i} \\ &= \eta_{n-1}(k) - \frac{w_{k,n}^2}{\mathbf{w}_n^T\mathbf{w}_n + \lambda_n} \end{aligned} \tag{27}$$

respectively. For the benefits of those readers who are unfamiliar with the LOO statistics, the idea of delete-1 cross validation and the computation of the LOO test error are explained in Appendix B.

The subset model selection procedure can be carried as follows: at the nth stage of the selection procedure, a model term is selected among the remaining n to N candidates if the resulting n-term model produces the smallest LOO test score J_n. It has been shown in [20] that the LOO statistic J_n is convex with respect to the model size n. That is, there exists an "optimal" model size n_s such that for $n \leq n_s$ J_n decreases as n increases while for $n \geq n_s + 1$ J_n increases as n increases. This property is extremely useful, as it enables the selection procedure to be 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 SDC procedure based on this OFR with LOO test score and local regularization can now be summarized as follows.

Initialization: Set λ_i, $1 \leq i \leq N$, to the same small positive value (e.g., 0.001). Set iteration index $I = 1$.

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

$$\epsilon_0(k) = f_k \quad \text{and} \quad \eta_0(k) = 1, \quad 1 \leq k \leq N$$
$$J_0 = \frac{1}{N}\mathbf{f}^T\mathbf{f} = \frac{1}{N}\sum_{k=1}^{N} f_k^2 \tag{28}$$

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

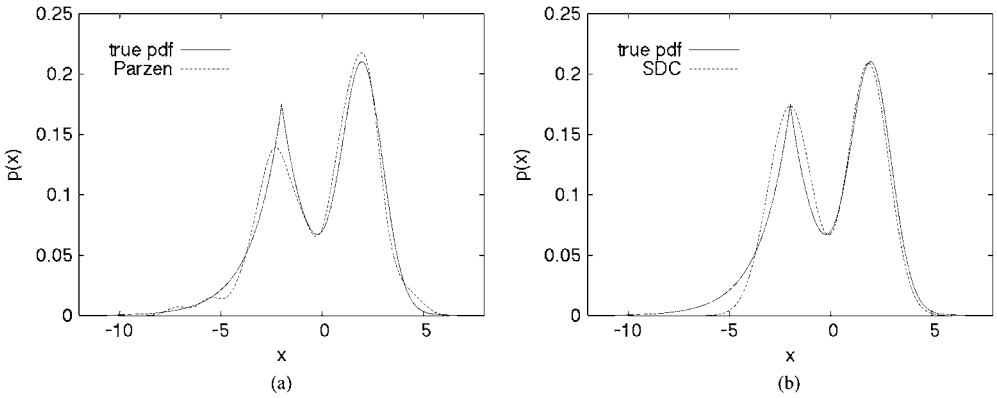

Fig. 1. (a) True density (solid) and a Parzen window estimate (dashed) and (b) true density (solid) and a sparse density construction estimate (dashed), for the 1-D example.

Step 2) Update $\boldsymbol{\lambda}$ using (21) and (22) with $N = n_I$. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a preset maximum iteration number (e.g., 10) is reached, stop; otherwise set $I+=1$ and go to *Step 1*.

The computational complexity of the above algorithm is dominated by the 1st iteration. After the first iteration, the model set contains only n_1 ($\ll N$) terms, and the complexity of the subsequent iteration decreases dramatically. As a probability density, the constraint (2) must be met. In [10], the nonnegative condition (2) is guaranteed by using backward elimination. Let $\mathbf{A}_n$ be the subset matrix of $\mathbf{A}$, corresponding to the n-term model, and $\mathbf{g}_n$ and $\boldsymbol{\beta}_n$ the associated orthogonal and original weight vectors, respectively, linked by $\mathbf{A}_n\boldsymbol{\beta}_n = \mathbf{g}_n$. If adding the nth term causes some of the elements in $\boldsymbol{\beta}_n$ to become negative, the associated previously selected model terms are removed. This strategy requires to carry out re-orthogonalization and in particular re-calculation of the LOO test score, which are computationally expensive. We adopt a much simple method to guarantee the nonnegative condition (2). In the nth stage, a candidate that causes $\boldsymbol{\beta}_n$ to have negative elements, if included, will not be considered at all. The unit length condition (3) can easily be met by normalizing the final n_s-term model weights with

$$\bar{\beta}_i = \frac{\beta_i}{\sum_{l=1}^{n_s}\beta_l}, \quad 1 \le i \le n_s. \tag{29}$$

IV. Numerical Examples

Four examples were used in simulation to test the proposed SDC algorithm and to compare its performance with the Parzen window estimate. Comparison with SVM kernel density estimation was also given by quoting the results of [7]. In order to remove the influence of different ρ values to the quality of the resulting density estimate, the optimal value for ρ, found empirically by cross validation, was used. That is, the value of ρ used was determined by testing performance. For the first three examples, in each case, a data set of N randomly drawn samples was used to construct kernel density estimates, and a separate test data set of $N_{\text{test}} = 10\,000$ samples was used to calculate the L_1 test error for the resulting estimate according to

$$L_1 = \frac{1}{N_{\text{test}}}\sum_{k=1}^{N_{\text{test}}} |p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k)|. \tag{30}$$

TABLE I
Performance of the Parzen Window (PW) Estimate and the Proposed Sparse Density Construction (SDC) Algorithm for the 1-D Example. STD: Standard Deviation

method	L_1 test error (mean $\pm$ STD)	kernel number (mean $\pm$ STD)
PW	$(2.063 \pm 0.626) \times 10^{-2}$	100 ± 0
SDC	$(2.177 \pm 0.737) \times 10^{-2}$	4.7 ± 0.8

The experiment was repeated by 100 different random runs for each example. The fourth example was a two-class two-dimensional (2-D) classification problem taken from [24].

Example 1: This was a one-dimensional (1-D) example, and the density to be estimated was given by

$$p(x) = 0.5\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-2)^2}{2}\right) + 0.5\frac{0.7}{2}\exp(-0.7|x+2|). \tag{31}$$

The number of data points for density estimation was $N = 100$. The optimal kernel widths were found to be $\rho = 0.55$ and $\rho = 1.0$ empirically with cross validation for the Parzen window estimate and the SDC estimate, respectively. Table I compares the performance of the two kernel density construction methods, in terms of the L_1 test error and the number of kernels required. Fig. 1(a) depicts the Parzen window estimated obtained in a run while Fig. 1(b) shows the density obtained by the SDC algorithm in a run, in comparison with the true distribution. For this 1-D example, it can be seen that the accuracy of the proposed SDC algorithm was comparable to that of the Parzen window estimate, and the algorithm realized very sparse estimates with an average kernel number less than 5% of the data samples.

This example was considered in [7], where a SVM Gaussian kernel density estimate of five terms was identified from a single set of 100 training data with an L_1 test error of 2.165×10^{-2} over a test set of 10 000 samples. It can be seen that the result

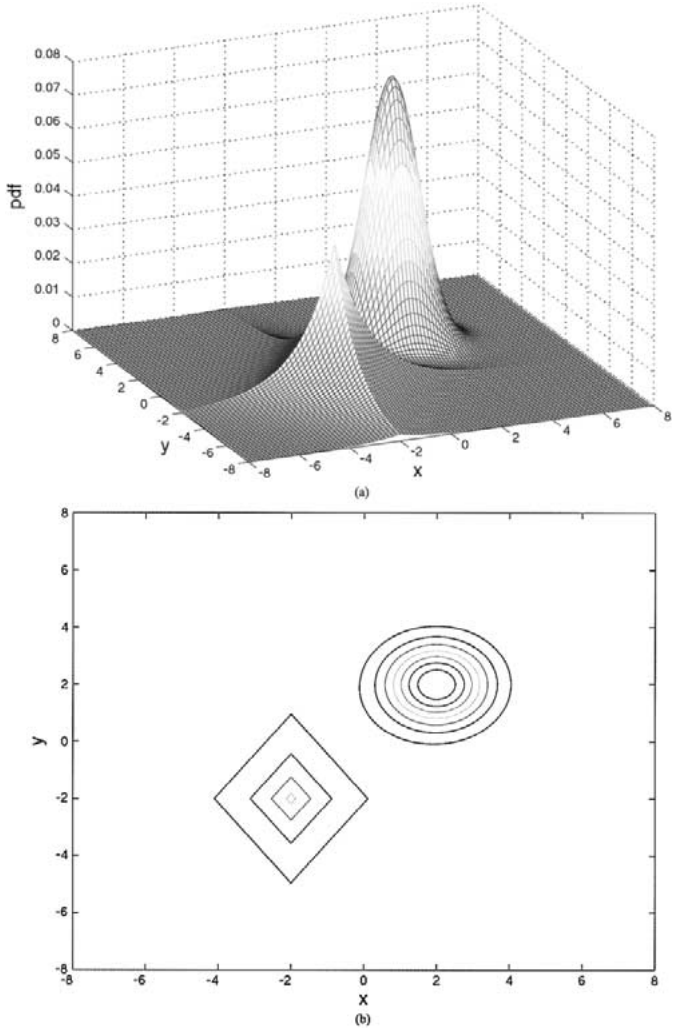

Fig. 2. (a) True density and (b) contour plot for the 2-D example.

obtained by the SDC method compares favorably with that of SVM method.

Example 2: The density to be estimated for this 2-D example was defined by

$$p(x,y) = 0.5\frac{1}{2\pi}\exp\left(-\frac{(x-2)^2}{2}\right)\exp\left(-\frac{(y-2)^2}{2}\right) + 0.5\frac{0.35}{4}\exp(-0.7|x+2|) \times \exp(-0.5|y+2|). \quad (32)$$

Fig. 2 shows this density distribution and its contour plot. The estimation data set contained $N = 500$ samples, and the empirically found optimal kernel widths were $\rho = 0.4$ for the Parzen window estimate and $\rho = 1.1$ for the SDC estimate, respectively. Table II lists the L_1 test errors and the numbers of kernels required for the two density estimation methods. A typical Parzen window estimate and a typical SDC estimate are depicted in Figs. 3 and 4, respectively. Again, for this example, the two density construction methods had comparable accuracies, but the SDC algorithm achieved very sparse estimates with an average number of required kernels less than 3% of the data samples.

TABLE II
PERFORMANCE OF THE PARZEN WINDOW (PW) ESTIMATE AND THE PROPOSED SPARSE DENSITY CONSTRUCTION (SDC) ALGORITHM FOR THE 2-D EXAMPLE. STD: STANDARD DEVIATION. THE NUMBER OF TRAINING POINTS WAS 500

method	L_1 test error (mean $\pm$ STD)	kernel number (mean $\pm$ STD)
PW	$(4.084 \pm 0.779) \times 10^{-3}$	500 ± 0
SDC	$(3.628 \pm 0.826) \times 10^{-3}$	11.9 ± 2.6

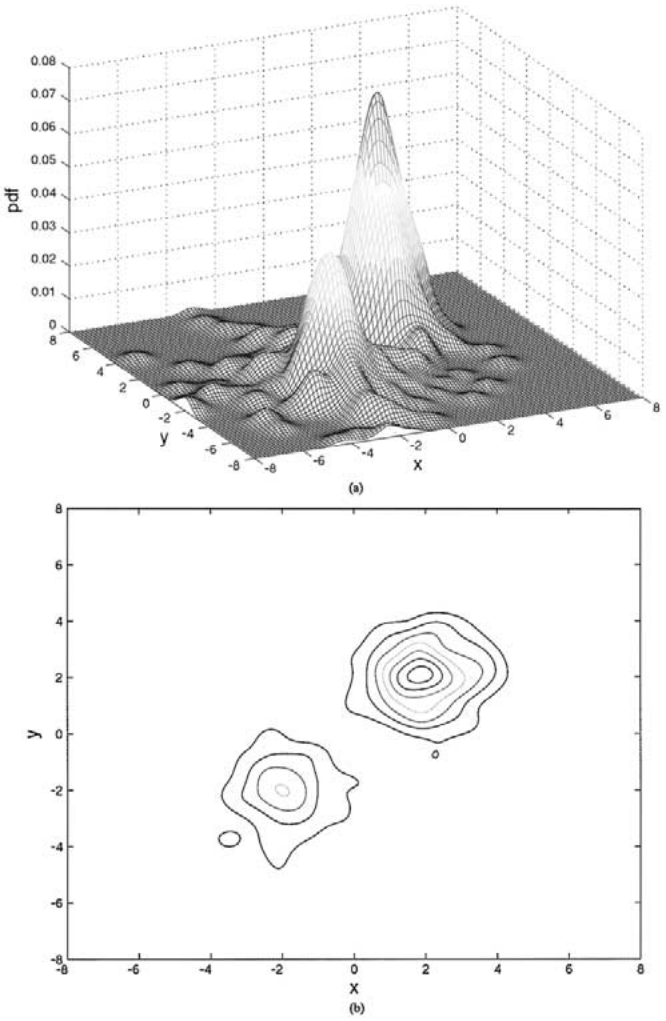

Fig. 3. (a) Parzen window estimate and (b) contour plot (b) for the 2-D example.

This example was also taken from [7], where a SVM Gaussian kernel density estimate of seven terms was identified from a single set of only 60 training data with an L_1 test error of 5.690×10^{-3} over a test set of 10 000 samples. For a comparison, we also performed the experiments over 100 random runs, each with 60 training data points, and the results are listed in Table III. Again, the accuracy of the SDC algorithm is comparable to that of the Parzen window estimate. Obviously, with such a short training data length, the standard deviation of estimate was large. Inspecting the results of the SDC algorithm, it was found that 25% of runs yielded kernel density estimates of less than seven terms with L_1 test errors smaller than 5.600×10^{-3}. This again demonstrates that the SDC method compares favorably with SVM method.

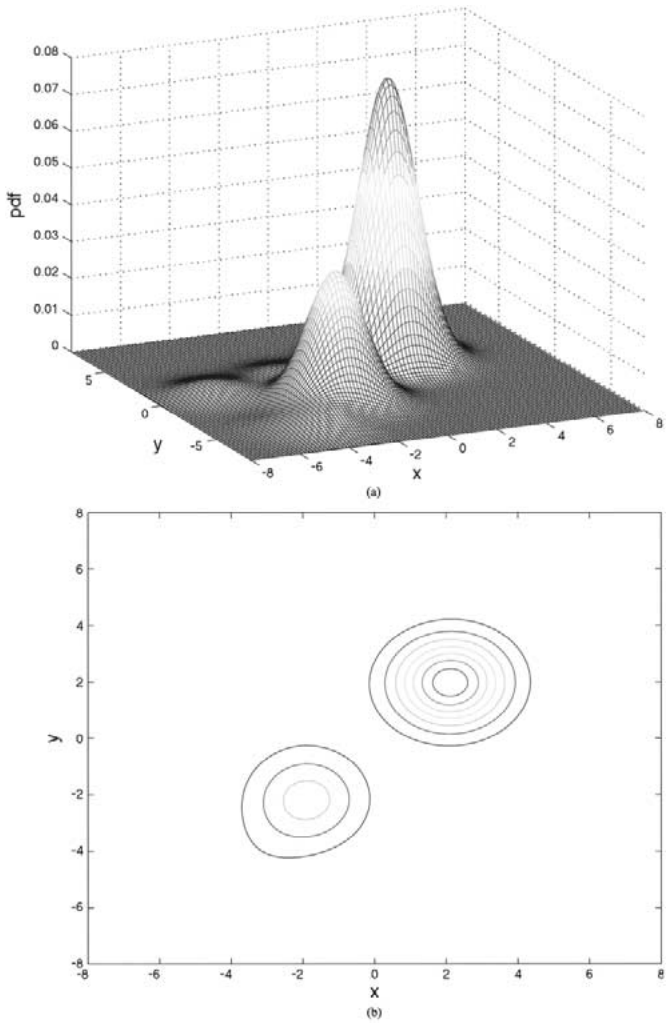

Fig. 4. (a) Sparse density construction estimate (a) and (b) contour plot for the 2-D example.

TABLE III
PERFORMANCE OF THE PARZEN WINDOW (PW) ESTIMATE AND THE PROPOSED SPARSE DENSITY CONSTRUCTION (SDC) ALGORITHM FOR THE 2-D EXAMPLE. STD: STANDARD DEVIATION. THE NUMBER OF TRAINING POINTS WAS 60

method	L_1 test error (mean $\pm$ STD)	kernel number (mean $\pm$ STD)
PW	$(7.842 \pm 1.675) \times 10^{-3}$	60 ± 0
SDC	$(7.456 \pm 2.441) \times 10^{-3}$	6.4 ± 1.6

Example 3: In this six-dimensional (6-D) example, the underlying density to be estimated was given by

$$p(\mathbf{x}) = \frac{1}{3(2\pi)^{6/2}} \times \left\{ \frac{1}{\det|\mathbf{\Gamma}_1|} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \mathbf{\Gamma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right) + \frac{1}{\det|\mathbf{\Gamma}_2|} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T \mathbf{\Gamma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right) + \frac{1}{\det|\mathbf{\Gamma}_3|} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_3)^T \mathbf{\Gamma}_3^{-1}(\mathbf{x}-\boldsymbol{\mu}_3)\right) \right\} \tag{33}$$

TABLE IV
PERFORMANCE OF THE PARZEN WINDOW (PW) ESTIMATE AND THE PROPOSED SPARSE DENSITY CONSTRUCTION (SDC) ALGORITHM FOR THE 6–D EAMPLE: STD STANDARD DEVIATION

method	L_1 test error (mean $\pm$ STD)	kernel number (mean $\pm$ STD)
PW	$(3.636 \pm 0.176) \times 10^{-5}$	600 ± 0
SDC	$(4.478 \pm 1.229) \times 10^{-5}$	14.9 ± 2.1

TABLE V
PERFORMANCE OF THE PARZEN WINDOW (PW) ESTIMATE AND THE PROPOSED SPARSE DENSITY CONSTRUCTION (SDC) ALGORITHM FOR THE TWO-CLASS CLASSIFICATION PROBLEM

method	$\hat{p}(\bullet\|\mathrm{C0})$	kernel width	$\hat{p}(\bullet\|\mathrm{C1})$	kernel width	test error rate
PW	125 kernels	0.24	125 kernels	0.24	8.1%
SDC	5 kernels	0.20	4 kernels	0.20	8.3%

with

$$\boldsymbol{\mu}_1 = [1.0\ 1.0\ 1.0\ 1.0\ 1.0\ 1.0]^T$$
$$\mathbf{\Gamma}_1 = \mathrm{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\} \tag{34}$$
$$\boldsymbol{\mu}_2 = [-1.0\ -1.0\ -1.0\ -1.0\ -1.0\ -1.0]^T$$
$$\mathbf{\Gamma}_2 = \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\} \tag{35}$$
$$\boldsymbol{\mu}_3 = [0.0\ 0.0\ 0.0\ 0.0\ 0.0\ 0.0]^T$$
$$\mathbf{\Gamma}_3 = \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}. \tag{36}$$

The estimation data set contained $N = 600$ samples. The optimal kernel width was found to be $\rho = 0.6$ for the Parzen window estimate and $\rho = 1.1$ for the SDC estimate, respectively, via cross validation using the test data set. The results obtained by the two density construction algorithms are summarized in Table IV. It can be seen that the SDC algorithm achieved a similar accuracy to that of the Parzen window estimate with a much sparser representation. The average number of required kernels for the SDC method was less than 3% of the data samples.

Example 4: The data was obtained from http://www.stats.ox.ac.uk/PRNN/. This was the synthetic data set taken from [24], which was a two-class classification problem in a 2-D feature space. The training set contained 250 samples with 125 points for each class, and the test set had 1000 points with 500 samples for each class. Tipping [8] reported that the optimal Bayes error rate for this example is around 8%, who also constructed a SVM Gaussian kernel classifier of 38 kernel functions with a test error rate of 10.6% and a relevance vector machine Gaussian kernel classifier of four kernel functions with a test error rate of 9.3%. We first estimated the two conditional density functions $\hat{p}(\mathbf{x}\,|\,\mathrm{C0})$ and $\hat{p}(\mathbf{x}\,|\,\mathrm{C1})$ from the training data, and then applied the Bayes decision rule

$$\left.\begin{array}{rl} \text{if } \hat{p}(\mathbf{x}\,|\,\mathrm{C0}) \geq \hat{p}(\mathbf{x}\,|\,\mathrm{C1}), & \mathbf{x} \text{ belongs to class 0} \\ \text{else}, & \mathbf{x} \text{ belongs to class 1} \end{array}\right\} \tag{37}$$

to the test data set and calculated the corresponding error rate.

Table V lists the results obtained by the two kernel density construction methods, the Parzen window estimate and the SDC algorithm, where the value of ρ was determined by minimizing the test error rate. It can be seen that the SDC method yielded very sparse conditional density estimates and the resulting test error was very close to the optimal Bayes classification perfor-

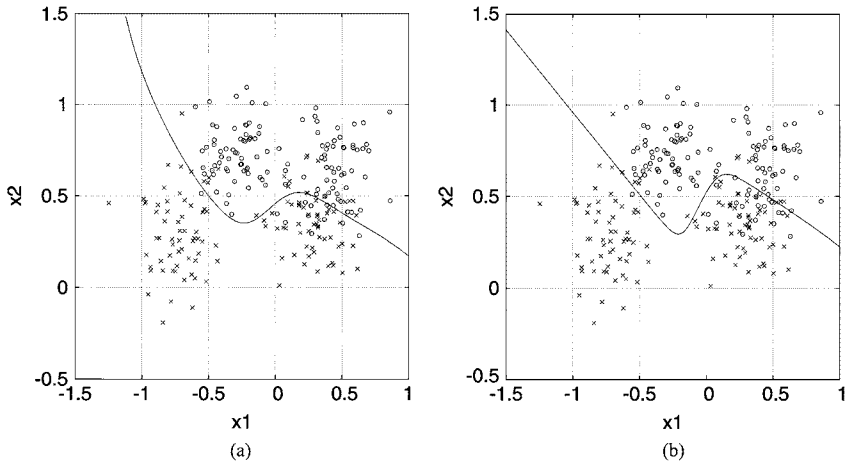

Fig. 5. (a) Decision boundary of the Parzen window estimate, and (b) decision boundary of the sparse density construction estimate, where circles represent the class-1 training data and crosses the class-0 training data.

mance. This clearly demonstrated the accuracy of the density estimates. This result compares favorably with the results of the state-of-art kernel classifiers reported in [8]. Fig. 5(a) and (b) depict the decision boundaries of the classifier (37) for the Parzen window and SDC methods, respectively.

V. Conclusion

An efficient construction algorithm has been presented for obtaining kernel density estimates based on an orthogonal forward regression procedure that incrementally minimizes the leave-one-out test score, coupled with local regularization to further enforce the sparseness of density estimate representations. The proposed method is simple to implement and computationally efficient, and except for the kernel width the algorithm contains no other free parameters that require tuning. The ability of the proposed algorithm to construct a very sparse kernel density estimate with a comparable accuracy to that of the full sample Parzen window estimate has been demonstrated using several examples. The results obtained have shown that the proposed method provides a viable alternative to the state-of-art support vector machine method for sparse kernel density estimation in practical applications.

Appendix A

According to the Bayesian learning theory (e.g., [8] and [23]), the optimal $\mathbf{g}$ is obtained by maximizing the posterior probability of $\mathbf{g}$, which is given by

$$p(\mathbf{g}\,|\,\mathbf{f},\mathbf{h},\eta) = \frac{p(\mathbf{f}\,|\,\mathbf{g},\mathbf{h},\eta)p(\mathbf{g}\,|\,\mathbf{h},\eta)}{p(\mathbf{f}\,|\,\mathbf{h},\eta)} \tag{38}$$

where $p(\mathbf{g}\,|\,\mathbf{h},\eta)$ is the prior with $\mathbf{h} = [h_1\ h_2\ \ldots\ h_N]^T$ denoting the vector of hyperparameters and η a noise parameter (the inverse of the variance of $\epsilon(k)$), $p(\mathbf{f}\,|\,\mathbf{g},\mathbf{h},\eta)$ is the likelihood, and $p(\mathbf{f}\,|\,\mathbf{h},\eta)$ is the evidence that does not depend on $\mathbf{g}$ explicitly. Under the assumption that $\epsilon(k)$ is white and has a Gaussian distribution, the likelihood is expressed as

$$p(\mathbf{f}\,|\,\mathbf{g},\mathbf{h},\eta) = \left(\frac{\eta}{2\pi}\right)^{N/2} \exp\left(-\frac{\eta}{2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}\right). \tag{39}$$

If the Gaussian prior is chosen, namely

$$p(\mathbf{g}\,|\,\mathbf{h},\eta) = \prod_{i=1}^{N} \frac{\sqrt{h_i}}{\sqrt{2\pi}} \exp\left(-\frac{h_i g_i^2}{2}\right) \tag{40}$$

maximizing $\log(p(\mathbf{g}\,|\,\mathbf{f},\mathbf{h},\eta))$ with respect to $\mathbf{g}$ is equivalent to minimizing the following Bayesian cost function:

$$J_B(\mathbf{g},\mathbf{h},\eta) = \eta\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \mathbf{g}^T\mathbf{H}\mathbf{g} \tag{41}$$

where $\mathbf{H} = \mathrm{diag}\{h_1, h_2, \ldots, h_N\}$. It is easily seen that the criterion (20) is equivalent to the criterion (41) with the relationship

$$\lambda_i = \frac{h_i}{\eta}, \quad 1 \le i \le N. \tag{42}$$

The hyperparameters specify the prior distributions of $\mathbf{g}$. Since initially one does not know the optimal value of $\mathbf{g}$, λ_i should be initialized to the same small value, and this corresponds to choose a same flat distribution for each prior of g_i in (40). The beauty of Bayesian learning is "let data speak"—it learns not only the model parameters $\mathbf{g}$ but also the related hyperparameters $\mathbf{h}$. This can be done for example by iteratively optimizing $\mathbf{g}$ and $\mathbf{h}$ using an evidence procedure [23], [8]. Following MacKay [23], it can be shown that the log model evidence for $\mathbf{h}$ and η is approximated as

$$\begin{aligned}\log(p(\mathbf{f}\,|\,\mathbf{h},\eta)) \approx & \sum_{i=1}^{N}\frac{1}{2}\log(h_i) - \frac{N}{2}\log(\pi) \\ & + \frac{N}{2}\log(\eta) - \sum_{i=1}^{N}\frac{1}{2}h_i g_i^2 \\ & - \frac{1}{2}\eta\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} - \frac{1}{2}\log(\det(\mathbf{B}))\end{aligned} \tag{43}$$

where $\mathbf{g}$ is set to the maximum a posterior probability solution, and the "Hessian" matrix $\mathbf{B}$ is diagonal and is given by

$$\begin{aligned}\mathbf{B} = \mathbf{H} + \eta\mathbf{W}^T\mathbf{W} = \mathrm{diag}\,\{&h_1 + \eta\mathbf{w}_1^T\mathbf{w}_1, \\ &h_2 + \eta\mathbf{w}_2^T\mathbf{w}_2, \ldots, h_N + \eta\mathbf{w}_N^T\mathbf{w}_N\}.\end{aligned} \tag{44}$$

Setting $(\partial \log(p(\mathbf{f} \mid \mathbf{h}, \eta)))/(\partial \eta) = 0$ yields the recalculation formula for η

$$\eta \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} = N - \sum_{i=1}^{N} \frac{\eta \mathbf{w}_i^T \mathbf{w}_i}{h_i + \eta \mathbf{w}_i^T \mathbf{w}_i}. \tag{45}$$

Setting $(\partial \log(p(\mathbf{f} \mid \mathbf{h}, \eta)))/(\partial h_i) = 0$ yields the recalculation formula for h_i

$$h_i = \frac{\eta \mathbf{w}_i^T \mathbf{w}_i}{g_i^2 \left(h_i + \eta \mathbf{w}_i^T \mathbf{w}_i\right)}. \tag{46}$$

Note $\lambda_i = h_i/\eta$ and define

$$\gamma = \sum_{i=1}^{N} \gamma_i \tag{47}$$

with

$$\gamma_i = \frac{\eta \mathbf{w}_i^T \mathbf{w}_i}{h_i + \eta \mathbf{w}_i^T \mathbf{w}_i} = \frac{\mathbf{w}_i^T \mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T \mathbf{w}_i}. \tag{48}$$

Then the recalculation formula for λ_i is

$$\lambda_i = \frac{\gamma_i}{N - \gamma} \frac{\boldsymbol{\epsilon}^T \boldsymbol{\epsilon}}{g_i^2}, \quad 1 \le i \le N. \tag{49}$$

APPENDIX B

Consider the model selection problem where a set of n_s models have been identified using the training data set $\mathcal{D}_N = \{f_k, \mathbf{x}_k\}_{k=1}^N$. Denote these models, identified using all the N data points of $\mathcal{D}_N$, as $\hat{f}_j(k)$ and the corresponding modeling errors as

$$\epsilon_j(k) = f_k - \hat{f}_j(k) \tag{50}$$

with index $j = 1, 2, \ldots, n_s$. A commonly used cross validation for model selection is the delete-1 cross validation. The idea is as follows. For every model, each data point in the training set $\mathcal{D}_N$ 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 $\mathcal{D}_{N,-l}$ be the resulting data set by removing the lth data point from $\mathcal{D}_N$, and denote the jth model estimated using $\mathcal{D}_{N,-l}$ as $\hat{f}_{j,-l}(k)$ and the related predicted model residual at l as

$$\epsilon_{j,-l}(l) = f_l - \hat{f}_{j,-l}(l). \tag{51}$$

The mean square LOO test error [17], [18] for the jth model $\hat{f}_j(k)$ is obtained by averaging all these prediction errors

$$E\left[\epsilon_{j,-k}^2(k)\right] = \frac{1}{N} \sum_{k=1}^{N} \epsilon_{j,-k}^2(k). \tag{52}$$

The mean-square LOO test error is a measure of the model generalization capability. To select the best model from the n_s candidate models $\hat{f}_j(k), 1 \le j \le n_s$, the same modeling procedure is applied to each of the n_s predictors, and the model with the minimum LOO test error is selected.

For linear-in-the-weights models, the LOO test errors can be generated, without actually sequentially splitting the training data set and repeatedly estimating the associated models, by using the Sherman–Morrison–Woodbury theorem [17]. Moreover, within the OFR model selection procedure, the LOO test errors for the n-term model can be computed very efficiently. It can readily be shown in [16] and [20] that the computation of the LOO error $\epsilon_{n,-k}(k)$ for the n-term model is based on the previously selected $(n-1)$-term model and the currently selected nth model term via the efficient recursion formulas (26) and (27).

APPENDIX C

The modified Gram-Schmidt orthogonalization procedure [21] calculates the $\mathbf{A}$ matrix row by row and orthogonalizes $\boldsymbol{\Phi}$ as follows: at the lth stage make the columns $\boldsymbol{\phi}_j$, $l+1 \le j \le N$, orthogonal to the lth column and repeat the operation for $1 \le l \le N-1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)} = \boldsymbol{\phi}_j$, $1 \le j \le N$, then for $l = 1, 2, \ldots, N-1$

$$\left.\begin{aligned} &\mathbf{w}_l = \boldsymbol{\phi}_l^{(l-1)} \\ &a_{l,j} = \mathbf{w}_l^T \boldsymbol{\phi}_j^{(l-1)} \Big/ \left(\mathbf{w}_l^T \mathbf{w}_l\right), \quad l+1 \le j \le N \\ &\boldsymbol{\phi}_j^{(l)} = \boldsymbol{\phi}_j^{(l-1)} - a_{l,j} \mathbf{w}_l, \quad l+1 \le j \le N \end{aligned}\right\}. \tag{53}$$

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

$$\left.\begin{aligned} &g_l = \mathbf{w}_l^T \mathbf{f}^{(l-1)} \Big/ \left(\mathbf{w}_l^T \mathbf{w}_l + \lambda_l\right) \\ &\mathbf{f}^{(l)} = \mathbf{f}^{(l-1)} - g_l \mathbf{w}_l \end{aligned}\right\} \quad 1 \le l \le N. \tag{54}$$

This orthogonalization scheme can be used to derive a simple and efficient algorithm for selecting subset models in a forward-regression manner [21]. First define

$$\boldsymbol{\Phi}^{(l-1)} = \left[\mathbf{w}_1 \quad \cdots \quad \mathbf{w}_{l-1} \quad \boldsymbol{\phi}_l^{(l-1)} \quad \cdots \quad \boldsymbol{\phi}_N^{(l-1)}\right]. \tag{55}$$

If some of the columns $\boldsymbol{\phi}_l^{(l-1)}, \cdots, \boldsymbol{\phi}_N^{(l-1)}$ in $\boldsymbol{\Phi}^{(l-1)}$ have been interchanged, this will still be referred to as $\boldsymbol{\Phi}^{(l-1)}$ for notational convenience. Let $\mathbf{A}_n$ denote the subset matrix of $\mathbf{A}$, corresponding to the n-term model, and $\mathbf{g}_n$ and $\boldsymbol{\beta}_n$ the associated orthogonal and original weight vectors, respectively, satisfying $\mathbf{A}_n \boldsymbol{\beta}_n = \mathbf{g}_n$. Let a very small positive number T_z be given, which specifies the zero threshold and is used to automatically avoiding any ill-conditioning or singular problem. With the initial conditions as specified in (28), the lth stage of the selection procedure is given as follows.

Step 1) For $l \le j \le N$:

Test 1—**Conditioning number check**. If $(\boldsymbol{\phi}_j^{(l-1)})^T \boldsymbol{\phi}_j^{(l-1)} < T_z$, the jth candidate is not considered.

Test 2—**Non-negativeness check**. Compute

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

Set $g_l = g_l^{(j)}$ and solve $\mathbf{A}_l\boldsymbol{\beta}_l = \mathbf{g}_l$ for $\boldsymbol{\beta}_l$. If $\boldsymbol{\beta}_l$ contains negative elements, the jth candidate is not considered.

Compute, for $1 \le k \le N$

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

and

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

where $f_k^{(l-1)}$ and $\phi_j^{(l-1)}(k)$ are the kth elements of $\mathbf{f}^{(l-1)}$ and $\boldsymbol{\phi}_j^{(l-1)}$, respectively. Let the index set $\mathcal{J}_l$ be

$$\mathcal{J} = \{l \le j \le N \text{ and } j \text{ passes both Tests 1 and 2}\}.$$

Step 2) Find

$$J_l = J_l^{(j_l)} = \min\left\{J_l^{(j)}, j \in \mathcal{J}_l\right\}$$

Then the j_lth column of $\boldsymbol{\Phi}^{(l-1)}$ is interchanged with the lth column of $\boldsymbol{\Phi}^{(l-1)}$, 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.

Step 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 (53) to derive the lth row of $\mathbf{A}$ and to transform $\boldsymbol{\Phi}^{(l-1)}$ into $\boldsymbol{\Phi}^{(l)}$; calculate g_l and update $\mathbf{f}^{(l-1)}$ into $\mathbf{f}^{(l)}$ in the way shown in (54); update the LOO error weightings

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

and go to *Step 1*.

REFERENCES

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

[2] B. W. Silverman, *Density Estimation*, London, U.K.: Chapman & Hall, 1996.

[3] H. Wang, "Robust control of the output probability density functions for multivariable stochastic systems with guaranteed stability," *IEEE Trans. Automat. Contr.*, vol. 44, pp. 2103–2107, Nov. 1999.

[4] S. Chen, A. K. Samingan, B. Mulgrew, and L. Hanzo, "Adaptive minimum-BER linear multiuser detection for DS-CDMA signals in multipath channels," *IEEE Trans. Signal Processing*, vol. 49, pp. 1240–1247, June 2001.

[5] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1066–1076, 1962.

[6] J. Weston, A. Gammerman, M. O. Stitson, V. Vapnik, V. Vovk, and C. Watkins, "Support vector density estimation," in *Advances in Kernel Methods—Support Vector Learning*, B. Schölkopf, C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 293–306.

[7] S. Mukherjee and V. Vapnik, "Support Vector Method for Multivariate Density Estimation," MIT AI Lab., Tech. Rep., A.I. Memo no. 1653, 1999.

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

[9] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[10] A. Choudhury, *Fast Machine Learning Algorithms for Large Data*. Southhampton, U.K.: Comput. Eng. Design Center, School Eng. Sciences, Univ. Southampton, 2002.

[11] P. B. Nair, A. Choudhury, and A. J. Keane, "Some greedy learning algorithms for sparse regression and classification with Mercer kernels," *J. Mach. Learn. Res.*, vol. 3, pp. 781–801, 2002.

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

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

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

[15] 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. Automat. Contr.*, vol. 48, pp. 1029–1036, June 2003.

[16] 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, pp. 898–911, Apr. 2004.

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

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

[19] G. Monari and G. Dreyfus, "Local overfitting control via leverages," *Neural Comput.*, vol. 14, pp. 1481–1506, 2002.

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

[21] 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.

[22] 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, Mar. 1991.

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

[24] B. D. Ripley, *Pattern Recognition and Neural Networks*, Cambridge, U.K.: Cambridge Univ. Press, 1996.

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 University of Sheffield, Sheffield, U.K., the University of Edinburgh, Edinburgh, U.K., and University of 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 in the Beijing Institute of Systems Engineering, Beijing, China, from 1987 to 1993. She worked as a Research Fellow in 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, 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 received 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 University of Hull, Hull, U.K., the University of Manchester Institute of Science and Technology (UMIST), Manchester, U.K., the University of Oxford, and the University of Cranfield, Cranfield, U.K., as well as being employed by the U.K. Ministry of Defense. 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, Image, Speech, and Intelligent Systems Research Group (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, multisensor 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 medal in 1998 for his work in autonomous systems, and the highest international award in IEE, the IEE Faraday medal, in 2001 for his work in intelligent control and neurofuzzy systems.