TABLE II
AVERAGE TEST ERRORS ERROR RATE (IN PERCENT) WITH $\varepsilon = 1e - 6$

Data Sets	SVM	RSVM(k)	Decreasing
Ringnorm	1.66 ± 0.12	1.48 ± 0.11(k=5)	0.18
Twonorm	2.96±0.23	2.40±0.12(k=4)	0.56
Waveform	9.88±0.43	9.48±0.53(k=4)	0.40

and Euclidean distances is inconclusive from our experiments, and we then decided to only report the results by Euclidean distance. It should also be pointed out that better classification accuracy can be achieved if we search all the optimal parameters in RSVM using the cross-validation strategy.

V. CONCLUSION

In this paper, a multidimensional maximum margin feature extraction approach for constructing a completely orthogonal basis and thus conducting efficient dimensionality reduction, called RSVM, is presented. Theoretical analysis shows that the SVM objective function is decreasing along the recursive components. In contrast to PCA, we use supervised information (labels) to conduct dimensionality reduction. Compared with LDA and regular SVM, the proposed method has no singularity problems and can further improve the accuracy. The general multilevel margin direction idea in this letter can be easily extended to SVM regression and several weighted SVM cases [17], [18] helping us to achieve more accurate results. Our future work will focus on using the recursive and multidimensional maximum margin idea to solve multiclassifications, especially face recognition problems. it may be that a new representing and recognizing approach for face patterns can be expected.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the referees for their valuable comments.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[2] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York: Springer-Verlag, 2001.

[3] I. T. Jolliffe, *Principal Component Analysis*. New York: Springer-Verlag, 1986.

[4] K. Fukunaga, *Introduction to Statistical Pattern Classification*. San Diego, CA: Academic, 1990.

[5] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 711–720, Jul 1997.

[6] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 831–836, Aug. 1996.

[7] X. Wang and X. Tang, "A unified framework for subspace face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1222–1228, Sep. 2004.

[8] W. J. Krzanowski, P. Jonathan, W. V. McCarthy, and M. R. Thomas, "Discriminant analysis with singular covariance matrices: Methods and applications to spectroscopic data," *Appl. Statist.*, vol. 44, pp. 101–115, 1995.

[9] J. Ye and T. Xiong, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis," *J. Mach. Learn. Res.*, vol. 7, pp. 1183–1204, 2006.

[10] C. Xiang, X. A. Fan, and T. H. Lee, "Face recognition using recursive fisher linear discriminant," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2097–2105, Aug. 2006.

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

[12] V. Vapnik, *Statistical Learning Theory*. Reading, MA: Addison-Wiley, 1998.

[13] N. Cristianini and J. Schawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2000.

[14] Q. Tao, G. Wu, and J. Wang, "The theoretical analysis of FDA and applications," *Pattern Recognit.*, vol. 39, no. 6, pp. 1199–1204, 2006.

[15] C. Cortes and V. Vapnik, "Support vector networks," *Mach. Learn.*, vol. 20, pp. 273–297, 1995.

[16] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.

[17] Q. Tao, G. Wu, F. Y. Wang, and J. Wang, "Posterior probability support vector machines for unbalanced data," *IEEE Trans. Neural Netw.*, vol. 16, no. 6, pp. 1561–1573, Nov. 2005.

[18] Q. Tao and J. Wang, "A new fuzzy support vector machine based on the weighted margin," *Neural Process. Lett.*, vol. 20, pp. 139–150, 2004.

A Forward-Constrained Regression Algorithm for Sparse Kernel Density Estimation

Xia Hong, Sheng Chen, and Chris J. Harris

***Abstract*—Using the classical Parzen window (PW) estimate as the target function, the sparse kernel density estimator is constructed in a forward-constrained regression (FCR) manner. The proposed algorithm selects significant kernels one at a time, while the leave-one-out (LOO) test score is minimized subject to a simple positivity constraint in each forward stage. The model parameter estimation in each forward stage is simply the solution of jackknife parameter estimator for a single parameter, subject to the same positivity constraint check. For each selected kernels, the associated kernel width is updated via the Gauss–Newton method with the model parameter estimate fixed. The proposed approach is simple to implement and the associated computational cost is very low. Numerical examples are employed to demonstrate the efficacy of the proposed approach.**

***Index Terms*—Cross validation, jackknife parameter estimator, Parzen window (PW), probability density function (pdf), sparse modeling.**

I. INTRODUCTION

The estimation of the probability density function (pdf) from observed data samples is a fundamental problem in many machine learning and pattern recognition applications [1]–[3]. The Parzen window (PW) estimate is a simple yet remarkably accurate nonparametric density estimation technique [2]–[4]. A general and powerful approach to the problem of pdf estimation is the finite mixture model [5]. The finite mixture model includes the PW estimate as a special case in that equal weights are adopted in the PW, with the number of mixtures equal to the number of training data samples. A disadvantage associated with the PW estimate is its high computational cost of the point density estimate for a future data sample in the cases whereby the training data set is very large. Clearly, by taking a much smaller

Manuscript received January 12, 2007; revised April 20, 2007 and May 25, 2007; accepted July 2, 2007.

X. Hong is with the School of Systems Engineering, University of Reading, Hampshire RG6 6AY, U.K. (e-mail: x.hong@reading.ac.uk).

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

Digital Object Identifier 10.1109/TNN.2007.908645

number of mixture components, the finite mixture model can be regarded as a condensed representation of data [5]. Note that the mixing weights in the finite mixture model need to be determined through parametric optimization, unlike just adopting equal weights in the PW. Much of the work in the fitting of a finite mixture model is based on a fixed number of mixtures and the expectation–maximization (EM) algorithms [5]. The disadvantages are as follows: 1) the predetermined model size may not be suitable to the data and 2) the convergence speed of EM is generally slow. Hence, it is desirable to develop new methods of fitting a finite mixture model with the capability to infer a minimal number of mixtures from the data efficiently.

Motivated by this, there is a considerable interest in the research into the sparse pdf estimate. The support vector machine (SVM) density estimation technique has been proposed in [6] and [7], in which the density estimation problem is formulated as a supervised learning mode while the mean absolute deviation between the empirical cumulative distribution function and that from the model is minimized. The optimization method in SVM is to solve a constrained quadratic optimization problem. This yields to the *sparsity inducing* property, i.e., at optimality, many kernels weights are driven to zeros. The desirable property of *sparsity inducing* also happens in the interesting approach of reduced set density estimator (RSDE) [8]. The RSDE is different from the SVM in that it is based on the minimization of integrated squared error (ISE) between the estimator and the true density. Two efficient optimization algorithms were introduced for RSDE that has a complexity of $O(N^2)$ per iteration, where N is the number of data samples, compared to a standard quadratic optimization solver at $O(N^3)$.

Alternatively, a novel regression-based probability density estimation method has been introduced [9], in which the empirical cumulative distribution function was constructed in the same manner as in SVM density estimation approach [6], as the desired response. By extending an efficient supervised model construction method, the forward regression approach [10], the orthogonal forward regression (OFR) combined with a leave-one-out (LOO) test score and local regularization has been introduced [11], [12]. The regression-based idea of [9] and the approach in [11] and [12] have been extended to yield a new OFR-based sparse density estimation algorithm [13], which is capable of automatically constructing very sparse kernel density estimate with comparable performance to that Parzen window estimate. Alternatively, a simple and viable alternative approach has been proposed to use the kernels directly as regressors and the target response as Parzen window estimate [14]. In practice, for the implementation of any of the aforementioned approaches and the proposed approach to the massive data sets (e.g., $N > 10^6$) with personal computers, the hybrid methods are recommended which combine the probability density estimation methods with the use of other data reduction approaches [15].

This letter introduces a new algorithm for sparse kernel density estimator using the classical PW as the target function and the kernels as regressors. The proposed sparse kernel density estimator construction using forward-constrained regression algorithm (FCR-SDC) is based on the FCR [16] in which mixing weights are estimated through a set of parameters, each of which relates to the model at the current regression stage and a new candidate term. In each forward stage, the model term selection is based on the criterion of a minimal LOO test score, subject to a simple positivity constraint. A one parameter jackknife parameter estimator is utilized in each regression step, subject to the same positivity constraint check. For each selected kernels, the associated kernel width is updated via the Gauss–Newton method [17] with the model parameter estimate fixed. The proposed algorithm has the advantage of maximal computationally efficiency due to the following: 1) the parameter estimation is reduced to the solution of the minimal possible number of one parameter, 2) the kernel width updating using the Gauss–Newton method involves also the minimal possible number of one parameter, that is the width of the selected kernel, and 3) the positivity constraint on the mixing weights can be easily accommodated.

II. Kernel Density Estimator

Given a finite data set consisting of N data samples, $D = \{\mathbf{x}_1,\ldots,\mathbf{x}_j,\ldots\mathbf{x}_N\}$, where the feature vector variable $\mathbf{x}_j \in \Re^m$ follows an unknown pdf $p(\mathbf{x})$, the problem under study is to find a sparse approximation of $p(\mathbf{x})$ based on D.

A general kernel-based density estimate of $p(\mathbf{x})$ is given by

$$\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma}) = \sum_{j=1}^{N} g_j K(\mathbf{x},\mathbf{x}_j,\sigma_j)$$

$$\text{subject to} \quad g_j \geq 0, \qquad j=1,\ldots,N, \qquad \mathbf{g}^T\mathbf{1}=1. \qquad (1)$$

where $\mathbf{g} = [g_1, g_2, \ldots, g_N]^T$. g_j's are the kernels weights. $\boldsymbol{\sigma} = [\sigma_1,\ldots,\sigma_N]^T$ is kernel width vector. $\mathbf{1}$ is a vector with an appropriate dimension and all elements as ones. $K(\mathbf{x},\mathbf{x}_j,\sigma_j)$ is a chosen kernel function with kernel width σ_j. In this letter

$$K(\mathbf{x},\mathbf{x}_j,\sigma_j) = \frac{1}{(2\pi\sigma_j^2)^{m/2}}\exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_j\|^2}{2\sigma_j^2}\right) \qquad (2)$$

is used. Let the well-known PW estimator be denoted by $\hat{p}(\mathbf{x};\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}})$, where $\mathbf{g}^{\mathrm{Par}} = [g_1^{\mathrm{Par}},\ldots,g_N^{\mathrm{Par}}]^T$ and $g_j^{\mathrm{Par}} = 1/N$, $\forall j$. The log-likelihood for $\mathbf{g}$ can be formed using observed data D as $\log L$ as

$$\frac{1}{N}\sum_{i=1}^{N}\log\hat{p}(\mathbf{x}_i;\mathbf{g},\boldsymbol{\sigma}) = \frac{1}{N}\sum_{i=1}^{N}\log\left(\sum_{j=1}^{N} g_j K(\mathbf{x}_i,\mathbf{x}_j,\sigma_j)\right). \qquad (3)$$

Note that by the law of large numbers, the log-likelihood of (3) tends to

$$\int_{\Re^m} p(\mathbf{x})\log\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})d\mathbf{x} \qquad (4)$$

as $N \rightarrow \infty$ with probability one. Equation (4) is simply the negative cross entropy or divergence between the true density $p(\mathbf{x})$ and the estimate $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$. It can be shown that for a given kernel width $\sigma_j = \sigma^{\mathrm{Par}}$, $\forall j$, the PW estimator $g_j^{\mathrm{Par}} = 1/N$, $\forall j$, can be obtained as an optimal estimator via the maximization of (3), respective to $\mathbf{g}$ subject to the constraints $g_j \geq 0$, $j=1,\ldots,N$, and $\mathbf{g}^T\mathbf{1}=1$. Note that the choice of σ^{Par} is crucial in density estimation using PW [1]. Based on the principle of minimizing the mean integrated square error (MISE) [1], σ^{Par} can be found so as to minimize the least squares cross-validation criterion $M(\sigma)$ given by [1]

$$\frac{1}{N^2}\sum_{i,j=1}^{N} K(\mathbf{x}_i,\mathbf{x}_j,\sqrt{2}\sigma) - \frac{2}{N(N-1)}\sum_{i,j=1,j\neq i}^{N} K(\mathbf{x}_i,\mathbf{x}_j,\sigma)$$

$$\approx \frac{1}{N^2}\sum_{i,j=1}^{N} K^*(\mathbf{x}_i,\mathbf{x}_j,\sigma) + \frac{2}{N(2\pi\sigma^2)^{m/2}} \qquad (5)$$

where $K^*(\mathbf{x}_i,\mathbf{x}_j,\sigma) = K(\mathbf{x}_i,\mathbf{x}_j,\sqrt{2}\sigma) - 2K(\mathbf{x}_i,\mathbf{x}_j,\sigma)$. The computational cost of finding σ^{Par} is $O(N^2)$; this is scaled by the number of grid searches set by the user.

With the PW estimator, the associated computational cost for evaluating the probability density estimate for a future sample scales directly with the sample size N. Therefore, it is desirable to devise a sparse representation of $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$, in which the terms are composed of a small subset of data samples.

Clearly, any good sparse kernel density estimator $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$ should be devised as close as possible to the unknown true density $p(\mathbf{x})$. Because the PW estimators have the property of optimality, it was suggested [14] that it is possible to use the PW estimator as the target of the proposed sparse kernel density estimator. Specifically, we can write a regression equation linking $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$ and $\hat{p}(\mathbf{x};\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}})$ as

$$\hat{p}(\mathbf{x};\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}}) = \sum_{j=1}^{N} g_j K(\mathbf{x},\mathbf{x}_j,\sigma_j) + \varepsilon(\mathbf{x}) \tag{6}$$

where $\varepsilon(\mathbf{x})$ is the modeling error at $\mathbf{x}$ between the sparse kernel density estimator $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$ and the PW density estimator $\hat{p}(\mathbf{x};\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}})$, that is initially constructed based on D. The aims are to obtain g_j via minimizing some modeling error criterion, e.g., $E[\varepsilon^2(\mathbf{x})]$, and simultaneously, to achieve a sparse representation of $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$ [with most elements in $\mathbf{g}$ being zeros in (6)] subject to the constraints $g_j \geq 0$, $j = 1,\ldots,N$, and $\mathbf{g}^T\mathbf{1} = 1$.

III. Sparse Kernel Density Estimator Construction Using FCR Algorithm

Starting from an empty model, the proposed algorithm constructs the model forwardly as in [9], [13], and [14]. However, the construction of the proposed sparse kernel density estimator is based on the idea of the mixtures of experts network (MEN) [18] and forward-constraint regression [16], hence it is very different from [9], [13], and [14], because no orthogonalization is incurred. In the proposed algorithm, the kernel functions $K(\mathbf{x},\mathbf{x}_j,\sigma_j)$ with nonzero g_j's are included into the model in a forward manner. The final sparse kernel density estimators are based on the kernels formed from $D_s = [\mathbf{x}'_1,\ldots,\mathbf{x}'_s]$, a subset of s data samples selected from D. That is, if $\mathbf{x}_6$ is selected to form the first kernel, this is denoted as $\mathbf{x}'_1$. Let a superscript k denote the kth forward step. At the kth forward step, the intermediate kernel density estimator $\hat{p}^{(k)}(\mathbf{x};\mathbf{g}^{(k)},\boldsymbol{\sigma})$ is denoted by $\hat{y}^{(k)}(\mathbf{x})$ as

$$\hat{y}^{(k)}(\mathbf{x}) = \sum_{j=1}^{k} g_j^{(k)} K(\mathbf{x},\mathbf{x}'_j,\sigma_j) \tag{7}$$

where $g_j^{(k)}$, $j = 1,\ldots,k$, are the kernels weights at the kth forward step. $g_j^{(k)} \geq 0$ and $\sum_{j=1}^{k} g_j^{(k)} = 1$. For notational simplicity, kernel width of the kernel being selected at jth step is still denoted by σ_j.

A. FCR Algorithm for Sparse Kernel Density Estimation

1) Initialization: The algorithm initially constructs a PW estimator, in which σ^{Par} is found via minimizing $M(\sigma)$ given by (5) from a grid search of σ values. The kernels in the PW estimator are used as the candidate kernels in (1), i.e., the kernel widths σ_j are initialized as $\sigma_j^{(0)} = \sigma^{(0)} = \gamma\sigma^{\mathrm{Par}}, \forall j$. $\gamma > 1$ is set by the user empirically.

2) Determination of the First Kernel: The sparse kernel density estimator $\hat{p}(\mathbf{x};\mathbf{g},\boldsymbol{\sigma})$ in (7) can be regarded as an MEN system with the kernel functions $K(\mathbf{x},\mathbf{x}'_j,\sigma^{(0)})$ as the experts [16]. The MEN system is initialized by determining the first expert as the first kernel $K(\mathbf{x},\mathbf{x}'_1,\sigma^{(0)})$, so that

$$\hat{y}^{(1)}(\mathbf{x}) = K(\mathbf{x},\mathbf{x}'_1,\sigma^{(0)}) \tag{8}$$

and $g_1^{(1)} = 1$. From (6) and (7)

$$\hat{p}(\mathbf{x};\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}}) = K(\mathbf{x},\mathbf{x}'_1,\sigma^{(0)}) + \varepsilon(\mathbf{x}). \tag{9}$$

From N kernels $K(\mathbf{x},\mathbf{x}_j,\sigma^{(0)})$, $j = 1,\ldots N$, one is to be determined as $K(\mathbf{x},\mathbf{x}'_1,\sigma^{(0)})$. This is simply done by searching for the term that produces the smallest value of mean squares modeling errors over D, i.e.,

$$j_1 = \arg\min\left\{\sum_{i=1}^{N}[\hat{p}(\mathbf{x}_i;\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}}) - K(\mathbf{x}_i,\mathbf{x}_j,\sigma^{(0)})]^2, \forall j\right\} \tag{10}$$

and $\mathbf{x}_{j_1}$ is then set as $\mathbf{x}'_1$. The mean squares modeling errors can be further reduced by adjusting σ_1 using the Gauss–Newton's method which is modified to enable the constraint that σ_1 is greater than a small threshold value set by the user, e.g., 0.01 is used here. We initialize $l = 0$, applying the following Gauss–Newton's method for a predetermined number of L iterations (e.g., $L = 5 \sim 10$)

$$\sigma_1^{(l)} = \max\left\{\left|\sigma_1^{(l-1)} + \eta \times \frac{\mathrm{num}_1}{\mathrm{den}_1}\right|, 0.01\right\} \tag{11}$$

in which

$$\begin{aligned}\mathrm{num}_1 &= \sum_{i=1}^{N}\varepsilon(\mathbf{x}_i) \times \frac{\partial}{\partial\sigma}K(\mathbf{x}_i,\mathbf{x}'_1,\sigma)|_{\sigma=\sigma_1^{(l-1)}}\\ \mathrm{den}_1 &= \sum_{i=1}^{N}\left[\frac{\partial}{\partial\sigma}K(\mathbf{x}_i,\mathbf{x}'_1,\sigma)|_{\sigma=\sigma_1^{(l-1)}}\right]^2\end{aligned} \tag{12}$$

with

$$\frac{\partial}{\partial\sigma}K(\mathbf{x}_i,\mathbf{x}'_1,\sigma) = K(\mathbf{x}_i,\mathbf{x}'_1,\sigma)\left(\frac{\|\mathbf{x}_i - \mathbf{x}'_1\|^2}{\sigma^3} - \frac{m}{\sigma}\right). \tag{13}$$

$\eta > 0$ is a small step size. Note that $\varepsilon(\mathbf{x}_i)$ is computed from (9), in which σ_0 is repeatedly replaced by $\sigma_1^{(l-1)}$ in iteration step l. Set $\sigma_1 = \sigma_1^{(L)}$.

3) Determination of Subsequent Kernels: For the subsequent kernels, these are initially based on using LOO test score and the jackknife parameter estimator, followed by applying the Newton's algorithm to tune the width, also for L iterations. Consider the model term selection for forward step $k \geq 2$. It can be shown that [16]

$$\hat{y}^{(k)}(\mathbf{x}) = \lambda_{k-1}\hat{y}^{(k-1)}(\mathbf{x}) + (1-\lambda_{k-1})K(\mathbf{x},\mathbf{x}'_k,\sigma^{(0)}) \tag{14}$$

with $0 \leq \lambda_{k-1} \leq 1, \forall k$. The right-hand side of (14) is a convex combination of two terms, the current MEN system $\hat{y}^{(k-1)}(\mathbf{x})$ and the kth kernel $K(\mathbf{x},\mathbf{x}'_k,\sigma^{(0)})$ to be included into the model at the kth forward step. Initially, the proposed algorithm resolves two problems simultaneously: 1) which kernel is to be selected as $K(\mathbf{x},\mathbf{x}'_k,\sigma^{(0)})$ from $(N-k+1)$ candidate kernels and 2) what type of parameter estimator is adopted for λ_{k-1}. The proposed algorithm incorporates the two aspects based on the LOO test score for model term selection and the jackknife parameter estimator, subject to a simple convex constraint of $0 \leq \lambda_{k-1} \leq 1$. It is shown that the LOO test score for kernel selection is very easy to compute due to the fact that only one unknown parameter λ_{k-1} is involved in the FCR procedure.

From (6), (7), and (14), we have

$$\begin{aligned}&\hat{p}(\mathbf{x};\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}})\\ &\quad= \lambda_{k-1}\hat{y}^{(k-1)}(\mathbf{x}) + (1-\lambda_{k-1})K(\mathbf{x},\mathbf{x}'_k,\sigma^{(0)}) + \varepsilon(\mathbf{x}).\end{aligned} \tag{15}$$

With N data samples, we define $\hat{\mathbf{p}}^{\mathrm{Par}} = [\hat{p}(\mathbf{x}_1;\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}}),\ldots,\hat{p}(\mathbf{x}_N;\mathbf{g}^{\mathrm{Par}},\sigma^{\mathrm{Par}})]^T$, $\hat{\mathbf{y}}^{(k-1)} = [\hat{y}^{(k-1)}(\mathbf{x}_1),\ldots,\hat{y}^{(k-1)}(\mathbf{x}_N)]^T$, $\boldsymbol{\psi} = [K(\mathbf{x}_1,\mathbf{x}'_k,\sigma^{(0)}),\ldots,K(\mathbf{x}_N,\mathbf{x}'_k,\sigma^{(0)})]^T$, and $\boldsymbol{\varepsilon} = [\varepsilon(\mathbf{x}_1),\ldots,\varepsilon(\mathbf{x}_N)]^T$. Then, (15) can be rewritten in the vector form as

$$\hat{\mathbf{p}}^{\mathrm{Par}} = \lambda_{k-1}\hat{\mathbf{y}}^{(k-1)} + (1-\lambda_{k-1})\boldsymbol{\psi} + \boldsymbol{\varepsilon} \tag{16}$$

or

$$\mathbf{t} = \lambda_{k-1}\mathbf{w} + \boldsymbol{\varepsilon} \tag{17}$$

with $\mathbf{t} = [t_1, \ldots, t_N]^T = \hat{\mathbf{p}}^{\mathrm{Par}} - \boldsymbol{\psi}$ and $\mathbf{w} = [w_1, \ldots, w_N]^T = \hat{\mathbf{y}}^{(k-1)} - \boldsymbol{\psi}$.

We minimize the loss function $J = \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$ with respect to λ_{k-1} to yield the least squares solution

$$\lambda_{k-1}^{\mathrm{LS}} = \frac{\mathbf{w}^T\mathbf{t}}{\mathbf{w}^T\mathbf{w}} = \frac{b_{k-1}}{a_{k-1}} \tag{18}$$

where $b_{k-1} = \mathbf{w}^T\mathbf{t}$ and $a_{k-1} = \mathbf{w}^T\mathbf{w}$.

The kth step of the MEN system involves the selection of $K(\mathbf{x}, \mathbf{x}'_k, \sigma^{(0)})$. Note that by using each of the $(N - k + 1)$ candidate kernels to form $\boldsymbol{\psi}$, in turn, (18) is repeatedly calculated. For some candidate kernels, the solution may not satisfy the constraints $0 \leq \lambda_{k-1}^{\mathrm{LS}} \leq 1$. These kernels will then not be considered to be appropriate.

For all model terms which satisfy the constraints $0 \leq \lambda_{k-1}^{\mathrm{LS}} \leq 1$, the following proposed model term selection algorithm is applied, which combines the LOO cross validation with the jackknife parameter estimator for λ_{k-1} [given by (21)], subject to $0 \leq \lambda_{k-1} \leq 1$.

The LOO cross validation involves the removal of each $\mathbf{x}_j$, in turn, from the estimation data set D, $j = 1, \ldots, N$. The removed data point is used as a test point for the model constructed using the modified data set. It is easy to verify that the least squares solution using $(D \setminus \mathbf{x}_j)$, is given by

$$\lambda_{k-1}^{(-j)} = \frac{b_{k-1} - w_j t_j}{a_{k-1} - w_j^2}, \qquad j = 1, \ldots, N \tag{19}$$

and the mean squares of LOO errors $\varepsilon^{(-j)}(\mathbf{x}_j)$ are given by

$$J_k = E\{[\varepsilon^{(-j)}(\mathbf{x}_j)]^2\} = \frac{1}{N}\sum_{j=1}^{N}\left(t_j - \lambda_{k-1}^{(-j)}w_j\right)^2. \tag{20}$$

It is known that the jackknife parameter estimator is able to improve the accuracy of parameter estimation [19], [20]. The jackknife parameter estimator for λ_{k-1} given by

$$\lambda_{k-1} = \lambda_{k-1}^{\mathrm{LS}} - \frac{N-1}{N}\sum_{j=1}^{N}\lambda_{k-1}^{(-j)} \tag{21}$$

is employed for parameter estimation. Although, in general, the jackknife parameter estimator is regarded as computationally intensive, the additional computation is minimal in the proposed algorithm. This is because, in the FCR procedure, only a minimal number of one parameter $\lambda_{k-1}^{(-j)}$, $j = 1, \ldots, N$, is involved for each candidate term. In addition, most of the calculation in parameter estimation can be regarded as the byproducts of the previous LOO cross-validation procedure.

For all model terms which satisfy the constraints $0 \leq \lambda_{k-1}^{\mathrm{LS}} \leq 1$, (19)–(21) are repeatedly calculated. Among all solutions satisfying the constraints $0 \leq \lambda_{k-1} \leq 1$, the data point that produces the smallest J_k is selected as $\mathbf{x}'_k$ and then used to form kernel $K(\mathbf{x}, \mathbf{x}'_k, \sigma^{(0)})$.

Next, we adjust the width for $K(\mathbf{x}, \mathbf{x}'_k, \sigma^{(0)})$ using the Gauss–Newton's method [17]. We initialize the iteration step $l = 0$, applying the following iteration for L steps:

$$\sigma_k^{(l)} = \max\left\{\left|\sigma_k^{(l-1)} + \frac{\eta}{1 - \lambda_{k-1}} \times \frac{\mathrm{num}_k}{\mathrm{den}_k}\right|, 0.01\right\}. \tag{22}$$

num_k and den_k are computed using (12) and (13), but with num_1 and den_1 replaced by num_k and den_k. $\mathbf{x}'_1$ is also replaced by $\mathbf{x}'_k$ in both (12) and (13). Also note that $\varepsilon(\mathbf{x}_i)$ is repeatedly computed from (14), in which σ_0 needs to be replaced by $\sigma_k^{(l-1)}$ in each iteration. We set $\sigma_k = \sigma_k^{(L)}$.

The previous procedure iterates for a finite number of forward steps, with k increasing by one each step until the final model of size s ($s \ll N$) achieves a satisfactory modeling performance. In this letter, we terminate the procedure when the accuracy of the sparse kernel density estimator $\hat{p}(\mathbf{x}; \mathbf{g}, \boldsymbol{\sigma})$ is sufficiently close to that of the PW density estimator $\hat{p}(\mathbf{x}; \mathbf{g}^{\mathrm{Par}}, \sigma^{\mathrm{Par}})$.

4) Calculating Mixing Weights: The parameter $g_j^{(s)}$ is readily computed by applying the recursion given by [16]

$$\begin{aligned} g_j^{(s)} &= \lambda_{s-1}g_j^{(s-1)}, \qquad j = 1, \ldots, s-1 \\ g_s^{(s)} &= 1 - \lambda_{s-1} \end{aligned} \tag{23}$$

with $g_1^{(1)} = 1$.

IV. Comparative Study and Illustrative Examples

A. Comparison With Other Approaches

The other four methods used for comparison are as follows: 1) the PW estimate, 2) the sparse density construction (SDC) algorithm [13], 3) the sparse kernel density construction (SKD) algorithm [14], and 4) the reduced set density estimator with multiplicative nonnegative quadratic programming (RSDE-MNQP) [8], [21]. Before proceeding to the numerical examples, we discuss the similarities and differences among these approaches and highlight the computational advantages of the proposed approach.

1) The sparse kernel density estimator involves the determination of the model structure of (1) where most elements in $\mathbf{g}$ are zeros. Either this can be achieved by solving constrained quadratic optimization problem which initially work on *the full model* [6]–[8], or alternatively, significant model terms are selected one at a time forwardly [9], [13], [14] and the proposed approach and these methods initially work on *an empty model*.
2) For all algorithms including PW, there are preparation stages for setting up regression matrices, which involve cross validation for optimal width determination. The computation costs are at a similar level. For illustration, the real recorded running times of evaluating (5) using Matlab 6.5 with a Pentium-4 CPU 1.70 GHz, 384 MB are 19 and 32 s for Examples 1 and 2, respectively, in the proposed approach, where the number of grid searches was set as 10. The remaining part of the computational cost of different approaches except the PW is outlined in Table I. In Table I, the total computational cost is estimated via $\mathrm{Cost}_A \times \mathrm{Cost}_B + \mathrm{Cost}_C$. The real-time estimate is given by the mean of the running times, recorded using the same machine for performing the number of operations indicated in the previous column, based on vector operations of length N. For the quadratic-optimization-based approaches [6]–[8], the main computational cost is from quadratic programming. The RSDE-MNQP, one of the simplest approaches based on quadratic programming, is used for illustration. The computation cost per iteration in RSDE-MNQP is small at $O(N^2)$. This cost needs to be multiplied by the number of iterations ρ that is required in order to achieve convergence, which is set by the user.

   For the proposed approach and the OFR-based approaches, e.g., [13] and [14], the main computation costs are from the associated forward regression algorithms. For the OFR-based approaches, e.g., [13] and [14], the cost is approximated by a single term as the upper bound (in the case of excluding the cost of constraint check as used in [13]) to simplify the formula. The total number

TABLE I
COMPARISON OF THE MAIN COMPUTATIONAL COST BETWEEN DIFFERENT APPROACHES

Method	Number of algebraic operations $(Cost_A)$	Number of iteration for parameter estimation $(Cost_B)$	Additional Step $(Cost_C)$	Illustrative value ($N = 500, L = 5,$ $\rho = 3000, s = 50)$ $N_I = 20,)$ $\overline{n_p} = 70\% \max(n_p))$	Estimate of the mean running time (seconds)
SDC [13]	$\approx$ or $>$ $s(17N+2)(N-s/2)$	1	$10N_I(17s+2)$ $(s-N_I/2)$	$\approx$ or $>$ 2.0874×10^8	≈ 3
SKD [14]	$< s(17N+2)(N-s/2)$	1	$10N_I(17s+2)(s-N_I/2)$ $+\rho(N_I^2+5N_I+2)$	$< 2.1024 \times 10^8$	≈ 3
SKDE-MNQP	N^2+5N+2	$\rho = 500 \sim 5000$	0	7.5751×10^8	≈ 11
Proposed FCR-SDC	$3N^2 + \overline{n_p}(s-1)(N-(s-1)/2)$ $+sL(2N+3),$ $n_p = \max(n_p) = 8N+5$ or $\min(n_p) = 2N+1$	1	0	$\mathbf{6.6321 \times 10^7}$	≈ 1

of candidate terms used for evaluation is at $s(N - s/2)$, where s is the number of kernels in the determined subset, and this is then multiplied by the maximum number of multiplications required per candidate term. Cost_C is derived similarly from the multiples (ten times) of Cost_A (with N replaced by s) and that of the RSDE-MNQP (with N replaced by the final model size n_I).

For the proposed approach, there are three components in Cost_A that of the first regression step, the kth regression step ($k > 1$), and the Gauss–Newton's method. For the kth regression step ($k > 1$), count the total number of candidate terms used for evaluation at $(s-1)(N-(s-1)/2)$, and this is then multiplied by the mean of the number of multiplications (n_p) required per candidate term [n_p is either $\max(n_p)$ for candidate term with $\lambda_{k-1}^{\text{LS}}$ satisfying the constraint, or $\min(n_p)$, otherwise, both at $O(N)$].

3) A key difference between SDC [13] with both the proposed algorithm and SKD [14] is the difference in forming the regression models. In SDC, the regressors are $\int_{-\infty}^{\mathbf{x}} K(\mathbf{u}, \mathbf{x}_j, \sigma) d\mathbf{u}$, (which could be generated using erf.m in Matlab). However, in both SDC and the proposed algorithm, the regressors are simply the kernels $K(\mathbf{x}, \mathbf{x}_j, \sigma)$, of which the evaluation is much faster. The mean running time for exp.m is around 15% of that for erf.m in Matlab. The target functions in the associated regression models are also different. In the SDC, the empirical distribution function (given by [13, eq. (8)]) is constructed following the idea of [6], whereas in SKD and the proposed algorithm, the PW estimate is used. The running time for evaluating these two target functions are similar, if the width in PW estimate is given. If the computation cost required for the optimization of the width in PW estimate is taken into account, e.g., using a grid search via (5), it is reasonable to consider that the computational cost of forming the regression models in the three algorithms is comparable.

4) Despite the fact that both the SDC and the SKD use LOO test score and local regularization for model term selection [11], [12], there are differences between the SDC and the SKD. In the SDC, the nonnegative constraint condition is checked for each candidate terms to ensure that the constraint is satisfied during the OFR procedure. Clearly, the nonnegative constraint condition check incurs additional computation cost. In SKD, the OFR procedure is applied without checking the nonnegative constraints, so that only a subset model is found, with its coefficients unconstrained. Following this, a final MNQP step is applied, which is modified from [8]. This extra step recalculates the weighting coefficients so as to ensure that the nonnegative constraint is satisfied. This extra MNQP is fast due to the fact that it is based on a very small subset of the kernels.

5) A key difference between the proposed algorithm and that of [9], [13], and [14] is that no orthogonalization is involved. Note that in the proposed FCR-SDC approach, the regression model is in the same form as the SKD [14], but is different from that of the SDC [13]. For both [13] and [14], there are additional stages, i.e., of determining the regularization parameters in the SDC and SKD and the MNQP step for SKD. These extra stages bring the benefits of the superb sparsity and the excellent model generalization of the final models at some additional small computational cost Cost_C.

TABLE II
PERFORMANCE OF KERNEL DENSITY ESTIMATES FOR EXAMPLES 1 AND 2.

(a) Example 1.

Method	L_1 test error (mean $\pm$ STD) (mean $\pm$ STD)	Kernel numbers (mean $\pm$ STD)
PW	$(4.18 \pm 0.8) \times 10^{-3}$	500 ± 0
SDC [13]	$(\mathbf{3.83 \pm 0.8}) \times \mathbf{10^{-3}}$	$\mathbf{11.9 \pm 2.6}$
SKD [14]	$(3.84 \pm 0.8) \times 10^{-3}$	15.3 ± 3.9
SKDE-MNQP	$(4.24 \pm 0.8) \times 10^{-3}$	129.4 ± 35.7
Proposed FCR-SDC	$(4.21 \pm 0.9) \times 10^{-3}$	36.23 ± 13.7

(b) Example 2.

Method	L_1 test error (mean $\pm$ STD) (mean $\pm$ STD)	Kernel numbers (mean $\pm$ STD)
PW	$(3.18 \pm 0.13) \times 10^{-5}$	600 ± 0
SDC [13]	$(4.48 \pm 1.2) \times 10^{-5}$	14.9 ± 2.1
SKD [14]	$(3.11 \pm 0.5) \times 10^{-5}$	9.4 ± 1.9
SKDE-MNQP	$(3.67 \pm 0.7) \times 10^{-5}$	29.4 ± 10.1
Proposed FCR-SDC	$(\mathbf{3.03 \pm 0.3}) \times \mathbf{10^{-5}}$	$\mathbf{5.6 \pm 3.6}$

6) A unique feature of the proposed algorithm is that each kernel has its individually tuned width. Note that in Table I, the costs for kernel width determination are not taken into account in all other approaches. This means that the advantage of the computational efficiency of the proposed approach is more significant.

B. Illustrative Examples

In the following examples, a data set of N points was randomly drawn from a given distribution described in the following ($N = 500$ in Example 1 and $N = 600$ in Example 2). This was used to construct the pdf $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ using the proposed FCR-SDC approach. For each example, the experiment was repeated for 100 different random runs. For each random run, a separate test data set of $N_{\text{test}} = 10\,000$ points was used for evaluation according to

$$L_1 = \frac{1}{N_{\text{test}}} \sum_{k=1}^{N_{\text{test}}} |p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k; \mathbf{g}, \sigma)|. \tag{24}$$

The results of the proposed method in comparison with other approaches are shown in Table II(a) and (b), where the results of the SDC and SKD are quoted from [13] and [14]. The number of iterations for SKDE-MNQP was set as 3000. The number of iterations for Gauss–Newton algorithm was set as $L = 5$.

Example 1: The density to be estimated for this 2-D example was given by the mixture of two densities of a Gaussian and a Laplacian, as defined by

$$p(\mathbf{x}) = \frac{1}{4\pi}\exp\left(-\frac{(x_1-2)^2}{2}\right)\exp\left(-\frac{(x_2-2)^2}{2}\right) + \frac{0.35}{8}\exp(-0.7|x_1+2|)\exp(-0.5|x_2+2|). \quad (25)$$

Example 2: The density to be estimated for this 6-D example was defined by

$$p(\mathbf{x}) = \frac{1}{3}\sum_{i=1}^{3}\frac{1}{(2\pi)^3\sqrt{\det(\boldsymbol{\Gamma}_i)}}\exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Gamma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)\right) \quad (26)$$

with

$$\begin{aligned}
\boldsymbol{\mu}_1 &= [1.0, 1.0, 1.0, 1.0, 1.0, 1.0]^T \\
\boldsymbol{\mu}_2 &= [-1.0, -1.0, -1.0, -1.0, -1.0, -1.0]^T \\
\boldsymbol{\mu}_3 &= [0, 0, 0, 0, 0, 0]^T \\
\boldsymbol{\Gamma}_1 &= \mathrm{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\} \\
\boldsymbol{\Gamma}_2 &= \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\} \\
\boldsymbol{\Gamma}_3 &= \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}.
\end{aligned}$$

From the results in Table II(a) and (b), it is shown that the proposed FCR-SDC has comparable accuracy to that of PW, with an average number of required kernels lower that 6% of the data samples, for both examples. This means that the computational cost of the point density estimate for a future data sample is around 6% of that of PW.

V. Conclusion

A simple and efficient algorithm has been introduced for the construction of a sparse kernel model representation, based on a new FCR algorithm and using the well-known PW estimate as the desired function. The algorithm integrates several important concepts including LOO test score model term selection, the jackknife parameter estimation, and the Gauss–Newton algorithm to tune the kernel width. Numerical examples in comparison with different approaches are utilized to demonstrate that the models from the proposed algorithm are able to model the pdf with comparable accuracy, but with a much sparser representation than PW. It can be concluded that the proposed algorithm offers a viable alternative for sparse pdf estimation.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments that improved the quality of this letter. X. Hong would like to thank Prof. M. Girolami who promptly replied to queries in [8].

References

[1] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.

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

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

[4] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.

[5] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.

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

[7] V. Vapnik and S. Mukherjee, "Support vector machine for multivariate density estimation," in *Advances in Neural Information Processing Systems*, T. Leen, S. Solla, and K. R. Müller, Eds. Cambridge, MA: MIT Press, 2000, pp. 659–665.

[8] M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1253–1264, Oct. 2003.

[9] A. Choudhury, "Fast machine learning algorithms for large data," Ph.D. dissertation, School Eng. Sci., Univ. Southampton, Southampton, U.K., 2002.

[10] S. Chen, S. A. Billings, and W. Luo, "Orthogonal least squares methods and their applications to non-linear system identification," *Int. J. Control*, vol. 50, pp. 1873–1896, 1989.

[11] X. Hong, P. M. Sharkey, and K. Warwick, "Automatic nonlinear predictive model construction using forward regression and the PRESS statistic," *Inst. Electr. Eng. Proc.—Control Theory Appl.*, vol. 150, no. 3, pp. 245–254, 2003.

[12] 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, Cybern.*, vol. 34, no. 2, pp. 898–911, Apr. 2004.

[13] S. Chen, X. Hong, and C. J. Harris, "Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1708–1717, Aug. 2004.

[14] S. Chen, X. Hong, and C. J. Harris, "An orthogonal forward regression technique for sparse kernel density estimation," *Neurocomputing*, 2007, to be published.

[15] K. F. Cheng, C. K. Chu, and D. K. J. Lin, "Quick multivariate kernel density estimation for massive data sets," *Appl. Stochastic Models Business Industry*, vol. 22, pp. 533–546, 2006.

[16] X. Hong and C. J. Harris, "A mixture of experts network structure construction algorithm for modelling and control," *Appl. Intell.*, vol. 16, no. 1, pp. 59–69, 2002.

[17] M. J. D. Powell, "Problems related to unconstrained optimization," in *Numerical Methods for Unconstrained Optimization*, W. Murray, Ed. London, U.K.: Academic, 1972, pp. 29–55.

[18] M. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," *Neural Comput.*, vol. 6, pp. 181–214, 1994.

[19] M. Quenoulle, "Notes on bias in estimation," *Biometrica*, vol. 43, pp. 353–360, 1956.

[20] R. G. Miller, "An unbalanced jacknife," *Ann. Statist.*, vol. 2, no. 5, pp. 880–891, 1974.

[21] F. Sha, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," Univ. Pensylvania, Philadelphia, PA, Tech. Rep. MS-CIS-02-09, 2002.