Neurocomputing 115 (2013) 122–129

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Sparse probability density function estimation using the minimum integrated square error

Xia Hong [a], Sheng Chen [b,c,*], Abdulrohman Qatawneh [c], Khaled Daqrouq [c], Muntasir Sheikh [c], Ali Morfeq [c]

[a] School of Systems Engineering, University of Reading, Reading RG6 6AY, UK
[b] Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK
[c] Electrical & Computer Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 5 December 2012
Received in revised form
25 January 2013
Accepted 1 February 2013
Communicated by K. Li
Available online 19 February 2013

Keywords:
Probability density function
Sparse modelling
Minimum integrated square error
Forward constrained regression

ABSTRACT

We develop a new sparse kernel density estimator using a forward constrained regression framework, within which the nonnegative and summing-to-unity constraints of the mixing weights can easily be satisfied. Our main contribution is to derive a recursive algorithm to select significant kernels one at time based on the minimum integrated square error (MISE) criterion for both the selection of kernels and the estimation of mixing weights. The proposed approach is simple to implement and the associated computational cost is very low. Specifically, the complexity of our algorithm is in the order of the number of training data N, which is much lower than the order of N^2 offered by the best existing sparse kernel density estimators. Numerical examples are employed to demonstrate that the proposed approach is effective in constructing sparse kernel density estimators with comparable accuracy to those of the classical Parzen window estimate and other existing sparse kernel density estimators.

1. Introduction

The finite mixture model [1] is a general approach to the probability density function (PDF) estimation problem that is fundamental to many pattern recognition, data analysis and other engineering applications [2–7]. The celebrated Parzen window (PW) estimate [8] can be regarded as a special case of the finite mixture model, in which the number of mixtures is equal to that of the training data samples and all the mixing weights are equal. However, the point density estimate using the PW estimator for a future data sample can be computationally expensive if the number of training data samples is very large. Much of the existing works in the fitting of a finite mixture model are based on fixing the number of mixtures and applying the expectation-maximisation (EM) algorithm [9] to provide the maximum likelihood (ML) estimate of the mixture model's parameters. This associated ML optimisation, in general, is a highly nonlinear optimisation process requiring extensive computation, but for the Gaussian mixture model, the EM algorithm can be derived in an explicit iterative form [10]. However, this EM algorithm based ML estimation is well known to be ill posed and has a slow convergence speed, and to tackle the associated numerical difficulties, it is often required to apply resampling techniques [11–14]. In general, the correct number of mixture components is unknown, and simultaneously determining the required number of mixture components and estimating the associated parameters of the finite mixture model is a challenging problem. Hence it is highly desirable to develop new methods of fitting a finite mixture model with the capability to infer a minimum number of mixtures from the data automatically and efficiently.

There is a considerable interest into research on the sparse PDF estimation. The support vector machine (SVM) density estimation technique has been proposed [15,16], in which the density estimation problem is formulated as a supervised learning mode whilst the mean absolute deviation between the empirical cumulative distribution function (CDF) calculated from the training data and the CDF based on the PDF estimator also calculated from the training data are minimised. The optimisation in the SVM method is to solve a constrained quadratic optimisation problem. This yields the sparsity inducing property, i.e. at the optimality, many kernels' weights are driven to zeros. Alternatively a novel regression-based PDF estimation method has been introduced [17], in which the empirical CDF is constructed, in the same manner as in the SVM density estimation approach, to be used as the desired response. The orthogonal forward regression (OFR) approach is an efficient supervised regression model construction method [18]. In order to automatically determine the model

* Corresponding author at: Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK.
E-mail addresses: x.hong@reading.ac.uk (X. Hong), sqc@ecs.soton.ac.uk (S. Chen), qatawneh@kau.edu.sa (A. Qatawneh), haleddaq@yahoo.com (K. Daqrouq), mshaikh@kau.edu.sa (M. Sheikh), morfeq@kau.edu.sa (A. Morfeq).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.02.003

structure with the improved model generalisation, the OFR method has been combined with a leave-one-out test score and local regularisation [19,20]. The regression-based idea of [17] and the approach in [19,20] have been extended to yield a new OFR based sparse density estimation algorithm [21] which is capable of automatically constructing very sparse kernel density estimate, with comparable performance to that of the PW estimate. In [17,21], the regressors are the CDFs of the kernels and the target response is the empirical CDF. The calculation of CDFs becomes inconvenient and difficult for many types of kernels whose corresponding CDFs are difficult to compute. A simple and viable alternative approach has been proposed to use kernels directly as regressors by adopting the PW estimate as the target response [22].

The desirable property of sparsity inducing also happens in the interesting approach of reduced set density estimator (RSDE) [23]. The RSDE is different from the SVM in that it is based on the minimisation of the integrated square error (ISE) between the estimator and the true density. The minimum integrated square error (MISE) is a classical goodness of fit criterion for probability density estimation [2,23,24]. The optimisation problem in RSDE is a constrained quadratic optimisation one, and two efficient optimisation algorithms of both multiplicative updating of the weighting coefficients and sequential minimisation optimisation were introduced for the RSDE that has a complexity of $\mathcal{O}(N^2)$ per iteration, where N is the number of data samples and $\mathcal{O}(M)$ denotes the order of M, compared to a standard quadratic optimisation solver at $\mathcal{O}(N^3)$. Note that the RSDE is mainly restricted to using the Gaussian kernel, but the sparse density estimators of [21,22] do not have this restriction. The complexity of the sparse density estimators [21,22] is also $\mathcal{O}(N^2)$ scaled by the number of regressors selected, which is generally very small. Our extensive experience has shown that all the sparse density estimators [15,16,21–23] discussed here are capable of automatically producing sparse PDF estimates with comparable performance to that of the PW estimate, but the density estimators of [21–23] produce much sparser estimates than the SVM based density estimator.

Against this background, this paper introduces a new algorithm for sparse kernel density estimation based on the MISE and the forward constrained regression (FCR) [25]. In our proposed new sparse kernel density estimator, referred to as the FCR-MISE algorithm, a kernel term is selected one at a time which has the minimum ISE value among all the candidate kernels formed from the data points. Within the FCR framework, the mixing weights are computed using a recursion linking the weight for the newly selected kernel and the set of the mixing weights of the previous stages [25]. Thus the parameter estimation problem is reduced to a one-dimensional one, which is shown to have a closed-form solution using the MISE criterion. The proposed density estimation algorithm is very efficient due to the recursive computation and the closed-form solution of only one parameter per step. Specifically, the complexity of our proposed new algorithm is $\mathcal{O}(N)$ scaled by the squared number of kernels selected. Numerical examples are employed to demonstrate that our new sparse kernel density estimator is capable of producing very sparse PDF estimates with comparable accuracy to those of the PW estimator and other existing sparse kernel density estimators.

The paper is organised as follows. Section 2 introduces the idea of sparse kernel density estimator construction via the FCR framework. Section 3 proposes the new algorithm of joint kernel selection and mixing weight estimation based on the MISE and FCR. Numerical experiments are utilised to illustrate the effectiveness of the proposed algorithm in Section 4 and our conclusions are given in Section 5.

2. Sparse kernel density estimator construction via forward constrained regression

Given the finite data set $D_N = \{\boldsymbol{x}_j\}_{j=1}^N$ consisting of N data samples, where the data vector $\boldsymbol{x}_j \in \mathbb{R}^m$ follows an unknown PDF $p(\boldsymbol{x})$, the problem under study is to find a sparse approximation of $p(\boldsymbol{x})$ based on D_N. A general kernel based density estimate of $p(\boldsymbol{x})$ is given by

$$\hat{p}^{(N)}(\boldsymbol{x};\boldsymbol{\beta}_N,\rho) = \sum_{j=1}^{N} \beta_j K_\rho(\boldsymbol{x},\boldsymbol{x}_j) \tag{1}$$

subject to

$$\beta_j \ge 0,\ 1 \le j \le N, \quad \text{and} \quad \boldsymbol{\beta}_N^{\mathrm{T}}\mathbf{1}_N = 1, \tag{2}$$

where β_js are the kernel weights, $\boldsymbol{\beta}_N = [\beta_1\beta_2 \ldots \beta_N]^{\mathrm{T}}$, and $\mathbf{1}_N$ is the N-dimensional vector whose elements are all equal to one, while $K_\rho(\boldsymbol{x},\boldsymbol{x}_j)$ is a chosen kernel function with the kernel centre vector $\boldsymbol{x}_j$ and a suitable kernel width ρ. In this study, we use the Gaussian kernel of

$$K_\rho(\boldsymbol{x},\boldsymbol{x}_j) = \frac{1}{(2\pi\rho^2)^{m/2}} \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{x}_j\|^2}{2\rho^2}\right) \tag{3}$$

but many other kernels can also be used. The sparse kernel density estimation involves the determination of the model structure of (1) where most elements in $\boldsymbol{\beta}_N$ become zeros. This can be achieved either by solving the constrained quadratic optimisation problem which initially works on the full model set of all the N kernels [15,16,23], or alternatively by selecting significant model terms one at a time forwardly which initially works on an empty model set [17,21,22].

The proposed sparse kernel density estimation algorithm also initially works on an empty model set, as in the cases of [17,21,22]. Specifically, in our proposed algorithm, the kernel functions $K_\rho(\boldsymbol{x},\boldsymbol{x}_j)$ with nonzero weights β_j are included into the model set selected in a forward regression manner. The final sparse kernel density estimator are based on the kernels formed from the subset $D_s = \{\boldsymbol{x}'_1,\boldsymbol{x}'_2,\ldots,\boldsymbol{x}'_s\}$ of s data samples selected from D_N in this way. For example, if $\boldsymbol{x}_6$ is selected to form the first kernel, it is denoted as $\boldsymbol{x}'_1$ in the selected data subset. Let the superscript $^{(l)}$ denote the lth forward selection step. At the lth forward selection step, further denote the intermediate kernel density estimator $\hat{p}^{(l)}(\boldsymbol{x};\boldsymbol{\beta}_l^{(l)},\rho)$ as $\hat{y}^{(l)}(\boldsymbol{x})$, that is,

$$\hat{y}^{(l)}(\boldsymbol{x}) = \sum_{j=1}^{l} \beta_j^{(l)} K_\rho(\boldsymbol{x},\boldsymbol{x}'_j), \tag{4}$$

where $\beta_j^{(l)}$, $1 \le j \le l$, are the kernels weights at the lth forward selection step, and $\boldsymbol{\beta}_l^{(l)} = [\beta_1^{(l)}\beta_2^{(l)} \ldots \beta_l^{(l)}]^{\mathrm{T}}$.

The proposed algorithm uses the FCR procedure [25] described below:

(i) At the first step, the PDF estimator is simply the first selected kernel

$$\hat{y}^{(1)}(\boldsymbol{x}) = K_\rho(\boldsymbol{x},\boldsymbol{x}'_1). \tag{5}$$

This means that $\beta_1^{(1)} = 1$.

(ii) At the lth step, where $l \ge 2$, the PDF estimator is constructed by adding the lth selected kernel $K_\rho(\boldsymbol{x},\boldsymbol{x}'_l)$ to $\hat{y}^{(l-1)}(\boldsymbol{x})$ via

$$\hat{y}^{(l)}(\boldsymbol{x}) = \lambda_l \hat{y}^{(l-1)}(\boldsymbol{x}) + (1-\lambda_l) K_\rho(\boldsymbol{x},\boldsymbol{x}'_l), \tag{6}$$

where $0 \le \lambda_l \le 1$, $\forall l$, and $\lambda_1 = 0$.

It is a straightforward matter to verify that the model constructed using the FCR procedure satisfies the convex constraint conditions of (2), namely, $\beta_j^{(l)} \geq 0$, $1 \leq j \leq l$, and $\sum_{j=1}^{l} \beta_j^{(l)} = 1$, $\forall l \geq 1$, see [25]. If λ_l and $\boldsymbol{\beta}_{l-1}^{(l-1)}$ are given, $\boldsymbol{\beta}_l^{(l)}$ can be recursively computed via

$$\boldsymbol{\beta}_l^{(l)} = \begin{bmatrix} \lambda_l \boldsymbol{\beta}_{l-1}^{(l-1)} \\ 1-\lambda_l \end{bmatrix}, \tag{7}$$

where $l > 1$ and $\boldsymbol{\beta}_1^{(1)} = \beta_1^{(1)} = 1$.

It can be seen that the key issues are how to select the kernel $K_\rho(\boldsymbol{x},\boldsymbol{x}_l')$ as well as how to compute λ_l and hence the kernel weights $\boldsymbol{\beta}_l^{(l)}$, which are addressed in the next section.

3. Joint kernel selection and weight estimation based on the MISE

The MISE between a PDF estimator and the true density is a classical goodness of fit criterion for both nonparametric density estimation [2,23] and parametric density estimation [24]. In the following, we introduce a new algorithm integrating the kernel term selection and the kernel weight estimation based on the MISE measure, within the general FCR framework described in the previous section. More specifically, the joint kernel selection and weight estimation at the lth forward selection stage is detailed in this section. We initially formulate the kernel weight estimation problem using the MISE criterion for a given kernel per forward selection step, and following this, we present the full algorithm including the kernel selection also based on the MISE.

3.1. Kernel weight estimation

Assuming that at the lth forward selection stage $K_\rho(\boldsymbol{x},\boldsymbol{x}_l')$ has been selected, we consider the problem of determining λ_l based on the global accuracy measure for density estimate, the integrated square error (ISE) which is given as (see for example [23])

$$\begin{aligned} I(\boldsymbol{\beta}_l^{(l)}) &= \int \left(p(\boldsymbol{x}) - \sum_{j=1}^{l} \beta_j^{(l)} K_\rho(\boldsymbol{x},\boldsymbol{x}_j') \right)^2 d\boldsymbol{x} \\ &= \int p^2(\boldsymbol{x})\, d\boldsymbol{x} + \int \left(\sum_{j=1}^{l} \beta_j^{(l)} K_\rho(\boldsymbol{x},\boldsymbol{x}_j') \right)^2 d\boldsymbol{x} \\ &\quad -2E\left[\sum_{j=1}^{l} \beta_j^{(l)} K_\rho(\boldsymbol{x},\boldsymbol{x}_j') \right] = \int p^2(\boldsymbol{x})\, d\boldsymbol{x} \\ &\quad + \sum_{i=1}^{l} \sum_{j=1}^{l} \beta_i^{(l)} \beta_j^{(l)} \int K_\rho(\boldsymbol{x},\boldsymbol{x}_i') K_\rho(\boldsymbol{x},\boldsymbol{x}_j')\, d\boldsymbol{x} \\ &\quad -2 \sum_{j=1}^{l} \beta_j^{(l)} E[K_\rho(\boldsymbol{x},\boldsymbol{x}_j')] \\ &= \int p^2(\boldsymbol{x})\, d\boldsymbol{x} + Q^{(l)}(\lambda_l), \end{aligned} \tag{8}$$

in which $E[\bullet]$ denotes the expectation with respect to the true density $p(\boldsymbol{x})$. Since the unknown term $\int p^2(\boldsymbol{x})\, d\boldsymbol{x}$ is independent of $\boldsymbol{\beta}_l^{(l)}$, it can be dropped from the objective function. We write the argument directly as λ_l for the last term $Q^{(l)}(\lambda_l)$, which becomes our objective function. We point out that since our algorithm is based on the FCR framework, this is the only parameter that needs to be estimated at the lth selection stage. $\boldsymbol{\beta}_l^{(l)}$ depends on λ_l and $\boldsymbol{\beta}_{l-1}^{(l-1)}$, i.e. the sequence $\{\lambda_1, \lambda_2, \ldots, \lambda_{l-1}\}$, that have already been obtained from the previous forward selection steps (see (7)).

Using the following unbiased estimator of $E[K_\rho(\boldsymbol{x},\boldsymbol{x}_j')]$:

$$E[K_\rho(\boldsymbol{x},\boldsymbol{x}_j')] \approx \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_j'), \tag{9}$$

as well as noting the Gaussian kernel yield

$$\begin{aligned} Q^{(l)}(\lambda_l) &= \sum_{i=1}^{l} \sum_{j=1}^{l} \beta_i^{(l)} \beta_j^{(l)} K_{\sqrt{2}\rho}(\boldsymbol{x}_i',\boldsymbol{x}_j') \\ &\quad - \frac{2}{N} \sum_{j=1}^{l} \beta_j^{(l)} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_j'). \end{aligned} \tag{10}$$

For the first forward selection step, since we only have one kernel with $\lambda_1 = 0$, the only problem is to do with kernel selection but not with parameter estimation. For the convenience of derivation, we specifically write $Q^{(1)}(\lambda_1)$ as

$$Q^{(1)}(\lambda_1) = \boldsymbol{C}_1^{(1)} - 2\boldsymbol{p}_1^{(1)}, \tag{11}$$

with

$$\boldsymbol{p}_1^{(1)} = \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_1'), \tag{12}$$

$$\boldsymbol{C}_1^{(1)} = K_{\sqrt{2}\rho}(\boldsymbol{x}_1',\boldsymbol{x}_1') = \gamma, \tag{13}$$

where $\gamma = 1/(4\pi\rho^2)^{m/2}$. Using matrix expression, we can easily obtain the general recursive form of $Q^{(l)}(\lambda_l)$ for $l \geq 2$ given by

$$Q^{(l)}(\lambda_l) = (\boldsymbol{\beta}_l^{(l)})^{\mathrm{T}} \boldsymbol{C}_l^{(l)} \boldsymbol{\beta}_l^{(l)} - 2(\boldsymbol{\beta}_l^{(l)})^{\mathrm{T}} \boldsymbol{p}_l^{(l)}, \tag{14}$$

with

$$\boldsymbol{p}_l^{(l)} = \left[(\boldsymbol{p}_{l-1}^{(l-1)})^{\mathrm{T}} \ \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_l') \right]^{\mathrm{T}}, \tag{15}$$

$$\boldsymbol{C}_l^{(l)} = \begin{bmatrix} \boldsymbol{C}_{l-1}^{(l-1)} & \boldsymbol{b}_{l-1}^{(l)} \\ (\boldsymbol{b}_{l-1}^{(l)})^{\mathrm{T}} & \gamma \end{bmatrix}, \tag{16}$$

where $\boldsymbol{b}_{l-1}^{(l)} = [K_{\sqrt{2}\rho}(\boldsymbol{x}_1',\boldsymbol{x}_l') \ldots K_{\sqrt{2}\rho}(\boldsymbol{x}_{l-1}',\boldsymbol{x}_l')]^T$.

By substituting (7), (15) and (16) into (14), we have

$$\begin{aligned} Q^{(l)}(\lambda_l) &= \begin{bmatrix} \lambda_l \boldsymbol{\beta}_{l-1}^{(l-1)} \\ 1-\lambda_l \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{C}_{l-1}^{(l-1)} & \boldsymbol{b}_{l-1}^{(l)} \\ (\boldsymbol{b}_{l-1}^{(l)})^{\mathrm{T}} & \gamma \end{bmatrix} \begin{bmatrix} \lambda_l \boldsymbol{\beta}_{l-1}^{(l-1)} \\ 1-\lambda_l \end{bmatrix} \\ &\quad -2[\lambda_l (\boldsymbol{\beta}_{l-1}^{(l-1)})^{\mathrm{T}} 1-\lambda_l] \begin{bmatrix} \boldsymbol{p}_{l-1}^{(l-1)} \\ \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_l') \end{bmatrix} \\ &= \lambda_l^2 \mu^{(l)} + (1-\lambda_l)^2 \gamma + 2\lambda_l (1-\lambda_l) (\boldsymbol{b}_{l-1}^{(l)})^{\mathrm{T}} \boldsymbol{\beta}_{l-1}^{(l-1)} \\ &\quad -2\lambda_l v^{(l)} - \frac{2(1-\lambda_l)}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_l'), \end{aligned} \tag{17}$$

where

$$\begin{cases} \mu^{(l)} = (\boldsymbol{\beta}_{l-1}^{(l-1)})^{\mathrm{T}} \boldsymbol{C}_{l-1}^{(l-1)} \boldsymbol{\beta}_{l-1}^{(l-1)}, \\ v^{(l)} = (\boldsymbol{\beta}_{l-1}^{(l-1)})^{\mathrm{T}} \boldsymbol{p}_{l-1}^{(l-1)}. \end{cases} \tag{18}$$

It happens that $Q^{(l)}(\lambda_l)$ is a quadratic function with respect to λ_l. Hence there exists a unique minimum of $Q^{(l)}(\lambda_l)$, which can be found by setting $(\partial/\partial\lambda_l) Q^{(l)}(\lambda_l) = 0$, followed by the constraint satisfaction operation. This yields the closed-form solution for λ_l given as

$$\lambda_l = \min\{\max\{u_l, 0\}, 1\}, \tag{19}$$

with

$$u_l = \frac{\gamma - (\boldsymbol{b}_{l-1}^{(l)})^{\mathrm{T}} \boldsymbol{\beta}_{l-1}^{(l-1)} + v^{(l)} - \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_l')}{\mu^{(l)} + \gamma - 2(\boldsymbol{b}_{l-1}^{(l)})^{\mathrm{T}} \boldsymbol{\beta}_{l-1}^{(l-1)}}. \tag{20}$$

It is easy to verify that the constraint satisfaction operator

$$\min\{\max\{u,0\},1\} = \begin{cases} 1, & u > 1, \\ 0, & u < 0, \\ u, & 0 < u < 1. \end{cases} \tag{21}$$

Therefore, $0 \le \lambda_l \le 1$ is guaranteed. By plugging λ_l back to (17), we obtain the MISE value $Q^{(l)}(\lambda_l)$ for this given kernel. The computational cost of parameter estimation for a given kernel per forward selection step is in the order of $\mathcal{O}(l)$, which is extremely low owing to the recursive computation and the closed-form solution for the only parameter λ_l.

3.2. Joint kernel selection and weight estimation algorithm

The basic idea for kernel selection is to select the subset D_s of s data samples one at a time from the full data set D_N and to form the kernels $K_\rho(\boldsymbol{x},\boldsymbol{x}'_j)$ so that the ISE is minimised sequentially. Specifically, at the lth forward selection stage a data sample is selected from the remaining $(N-l+1)$ candidate data samples. We review the contribution of each candidate data sample according to its associated MISE value to decide if this sample is to be added to the model. The data point producing the smallest MISE value amongst all the candidate data samples is assigned as $\boldsymbol{x}'_l$ and is used to form $K_\rho(\boldsymbol{x},\boldsymbol{x}'_l)$.

First define $\boldsymbol{X}_N^{(l-1)} \in \mathbb{R}^{m \times N}$ as

$$\boldsymbol{X}_N^{(l-1)} = [\boldsymbol{x}'_1 \ldots \boldsymbol{x}'_{l-1} \boldsymbol{x}_l^{(l-1)} \ldots \boldsymbol{x}_N^{(l-1)}], \tag{22}$$

and $\boldsymbol{q}_N^{(l-1)} \in \mathbb{R}^{1 \times N}$ as

$$\boldsymbol{q}_N^{(l-1)} = \left[\frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}'_1) \ldots \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}'_{l-1}) \right.$$
$$\left. \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_l^{(l-1)}) \ldots \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_N^{(l-1)}) \right], \tag{23}$$

with

$$\boldsymbol{X}_N^{(0)} = [\boldsymbol{x}_1^{(0)} \boldsymbol{x}_2^{(0)} \ldots \boldsymbol{x}_N^{(0)}] = [\boldsymbol{x}_1 \boldsymbol{x}_2 \ldots \boldsymbol{x}_N], \tag{24}$$

$$\boldsymbol{q}_N^{(0)} = \left[\frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_1) \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_2) \right.$$
$$\left. \ldots \frac{1}{N} \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_N) \right]. \tag{25}$$

If the j_lth column, where $l \le j_l \le N$, and the lth column of $\boldsymbol{X}_N^{(l-1)}$ are interchanged, $\boldsymbol{X}_N^{(l-1)}$ becomes $\boldsymbol{X}_N^{(l)}$. Similarly, if the j_lth column and the lth column of $\boldsymbol{q}_N^{(l-1)}$ are interchanged, $\boldsymbol{q}_N^{(l-1)}$ becomes $\boldsymbol{q}_N^{(l)}$. Further define the jth element of $\boldsymbol{q}_N^{(l-1)}$ as $q^{(l-1)}(j) = (1/N) \sum_{i=1}^{N} K_\rho(\boldsymbol{x}_i,\boldsymbol{x}_j^{(l-1)})$ for $l \le j \le N$. We are now ready to present our proposed algorithm.

Initialisation: At the 1st stage of the selection procedure, set $\boldsymbol{\beta}_1^{(1)} = \beta_1^{(1)} = 1$ and $\lambda_1 = 0$. For $1 \le j \le N$, compute

$$Q^{(1,j)}(\lambda_1) = \gamma - 2\boldsymbol{p}_1^{(1,j)}, \tag{26}$$

where $\boldsymbol{p}_1^{(1,j)} = q^{(0)}(j)$. Next find

$$Q^{(1,j_1)}(\lambda_1) = \min\{Q^{(1,j)}(\lambda_1), 1 \le j \le N\}. \tag{27}$$

Then the j_1th column and the first column of $\boldsymbol{X}_N^{(0)}$ are interchanged to yield $\boldsymbol{X}_N^{(1)}$, and the j_1th column and the first column of $\boldsymbol{q}_N^{(0)}$ are interchanged to yield $\boldsymbol{q}_N^{(1)}$. This effectively selects the first kernel. Update $Q^{(1)}(\lambda_1) = Q^{(1,j_1)}(\lambda_1)$ with $\boldsymbol{C}_1^{(1)} = \gamma$ and $\boldsymbol{p}_1^{(1)} = \boldsymbol{p}_1^{(1,j_1)}$.

The lth *stage of the selection procedure, where* $l \ge 2$:

Step (1). Calculate $\mu^{(l)}$ and $v^{(l)}$ according to (18). Then, for $l \le j \le N$, compute

$$\boldsymbol{b}_{l-1}^{(l,j)} = [K_{\sqrt{2}\rho}(\boldsymbol{x}'_1,\boldsymbol{x}_j^{(l-1)}) \ldots K_{\sqrt{2}\rho}(\boldsymbol{x}'_{l-1},\boldsymbol{x}_j^{(l-1)})]^{\mathrm{T}},$$
$$d^{(l,j)} = (\boldsymbol{b}_{l-1}^{(l,j)})^{\mathrm{T}} \boldsymbol{\beta}_{l-1}^{(l-1)},$$
$$\lambda_l^{(j)} = \min\left\{ \max\left\{ \frac{\gamma - d^{(l,j)} + v^{(l)} - q^{(l-1)}(j)}{\mu^{(l)} + \gamma - 2d^{(l,j)}}, 0 \right\}, 1 \right\},$$

$$Q^{(l,j)}(\lambda_l^{(j)}) = (\lambda_l^{(j)})^2 \mu^{(l)} + (1-\lambda_l^{(j)})^2 \gamma$$
$$+ 2\lambda_l^{(j)}(1-\lambda_l^{(j)}) d^{(l,j)} - 2\lambda_l^{(j)} v^{(l)}$$
$$-2(1-\lambda_l^{(j)}) q^{(l-1)}(j).$$

Step (2): Find

$$Q^{(l,j_l)}(\lambda_l^{(j_l)}) = \min\{Q^{(l,j)}(\lambda_l^{(j)}), l \le j \le N\}. \tag{28}$$

Then the j_lth column and the lth column of $\boldsymbol{X}_N^{(l-1)}$ are interchanged to yield $\boldsymbol{X}_N^{(l)}$. Also the j_lth column and the lth column of $\boldsymbol{q}_N^{(l-1)}$ are interchanged to yield $\boldsymbol{q}_N^{(l)}$. This effectively selects the lth kernel. Update $\lambda_l = \lambda_l^{(j_l)}$ and $Q^{(l)}(\lambda_l) = Q^{(l,j_l)}(\lambda_l^{(j_l)})$ as well as

$$\boldsymbol{\beta}_l^{(l)} = \begin{bmatrix} \lambda_l^{(j_l)} \boldsymbol{\beta}_{l-1}^{(l-1)} \\ 1-\lambda_l^{(j_l)} \end{bmatrix},$$

$$\boldsymbol{p}_l^{(l)} = [(\boldsymbol{p}_{l-1}^{(l-1)})^{\mathrm{T}} q^{(l)}(l)]^{\mathrm{T}},$$

and

$$\boldsymbol{C}_l^{(l)} = \begin{bmatrix} \boldsymbol{C}_{l-1}^{(l-1)} & \boldsymbol{b}_{l-1}^{(l,j_l)} \\ (\boldsymbol{b}_{l-1}^{(l,j_l)})^{\mathrm{T}} & \gamma \end{bmatrix}.$$

Termination: The selection procedure is terminated at the $(s+1)$th stage when the following condition is detected:

$$|Q^{(s+1)}(\lambda_{s+1}) - Q^{(s)}(\lambda_s)| \le \delta Q,$$

where δQ is a predetermined very small positive number, and this produces a subset model with the s significant kernels.

An appropriate value of δQ is problem dependent and can be found empirically. Basically, δQ provides a trade off between the fitting accuracy and the sparsity of the kernel estimator obtained. Alternatively, cross-validation may be employed to determine δQ.

The computational cost of our proposed algorithm is extremely low. In fact, the lth stage of the selection procedure has the complexity of $2l(N-l+1)$. Therefore, the overall computational complexity of our proposed algorithm is approximately $s^2 N$, that is, $\mathcal{O}(N)$ scaled by s^2, where s is the number of kernels selected, which in general will not necessarily increase with the data set size. Note that for large data sets $s \ll N$. This computation complexity compares very favourably with the existing efficient sparse kernel density estimators, e.g. the RSDE which has a complexity of $\mathcal{O}(N^2)$ scaled by the number of iterations as well as the sparse kernel density (SKD) estimators of [21,22] which also have the complexity of $\mathcal{O}(N^2)$.

Remark. For all the kernel density estimation algorithms considered in this papers, including the PW estimator of [8], the sparse kernel density estimators of [15–17,21–23] as well as our proposed FCR-MISE estimator, the kernel width ρ is fixed. Appropriate value for ρ can be determined empirically through trial and error based on cross-validation. More specifically, a suitable value for ρ can be found using a line search based on the cross-validation performance. Let the number of line search points carried out to find an appropriate kernel width ρ be L_{lsp}. The total computational complexity of a sparse density estimation procedure is approximately equal to the complexity of constructing a sparse kernel density estimate given the kernel width scaled by the number of line search points L_{lsp}, since the complexity imposed in evaluating the cross-validation performance of a sparse density estimate is negligible compared with the complexity imposed in constructing a sparse kernel density estimate.

4. Simulation study

The first two examples are pure PDF estimation examples. In each of these two examples, a data set of N samples was randomly drawn from a distribution $p(\boldsymbol{x})$ and used to construct the PDF estimator $\widehat{p}^{(s)}(\boldsymbol{x};\boldsymbol{\beta}_s,\rho)$ using the proposed FCR-MSIE approach. A separate test data set of $N_{\text{test}} = 10\,000$ samples was used for evaluating the density estimate according to the L_1 test error

$$L_1 = \frac{1}{N_{\text{test}}} \sum_{k=1}^{N_{\text{test}}} \left| p(\boldsymbol{x}_k) - \widehat{p}^{(s)}(\boldsymbol{x}_k;\boldsymbol{\beta}_s,\rho) \right|. \tag{29}$$

The experiment was repeated for 100 different random runs. The benchmark PDF estimators used for comparison include the non-sparse PW estimator as well as the three efficient existing sparse PDF estimators, the SKD estimator of [21], the SKD estimator of [22], and the RSDE of [23]. The Gaussian kernel was used for all the algorithms.

It is worth emphasising that all the three sparse PDF estimator benchmarks are known to be very efficient with the complexity of $\mathcal{O}(N^2)$ in constructing a PDF estimate. However, our proposed FCR-MSIE estimator is even more efficient with the complexity of $\mathcal{O}(N)$. The RSDE is also particularly relevant to our FCR-MSIE algorithm, as the both methods are based on the MISE criterion and, therefore, mainly restricted to use the Gaussian kernel.

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

$$p(\boldsymbol{x}) = \frac{1}{4\pi} \exp\left(-\frac{(x_1-2)^2}{2} \right) \exp\left(-\frac{(x_2-2)^2}{2} \right) + \frac{0.35}{8} \exp(-0.7|x_1+2|)\exp(-0.5|x_2+2|). \tag{30}$$

The estimation data set contained $N=500$ points.

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

$$p(\boldsymbol{x}) = \frac{1}{3} \sum_{i=1}^{3} \frac{1}{(2\pi)^3 \sqrt{|\boldsymbol{\Gamma}_i|}} \exp\left(-\frac{1}{2} (\boldsymbol{x}-\boldsymbol{\mu}_i)^{\mathrm{T}} \boldsymbol{\Gamma}_i^{-1} (\boldsymbol{x}-\boldsymbol{\mu}_i) \right), \tag{31}$$

with

$$\boldsymbol{\mu}_1 = [1.0\ \ 1.0\ \ 1.0\ \ 1.0\ \ 1.0\ \ 1.0]^{\mathrm{T}},$$
$$\boldsymbol{\Gamma}_1 = \mathrm{diag}\{1.0,2.0,1.0,2.0,1.0,2.0\},$$
$$\boldsymbol{\mu}_2 = [-1.0\ \ -1.0\ \ -1.0\ \ -1.0\ \ -1.0\ \ -1.0]^{\mathrm{T}},$$
$$\boldsymbol{\Gamma}_2 = \mathrm{diag}\{2.0,1.0,2.0,1.0,2.0,1.0\},$$
$$\boldsymbol{\mu}_2 = [0.0\ \ 0.0\ \ 0.0\ \ 0.0\ \ 0.0\ \ 0.0]^{\mathrm{T}},$$
$$\boldsymbol{\Gamma}_2 = \mathrm{diag}\{2.0,1.0,2.0,1.0,2.0,1.0\},$$

where $|\boldsymbol{\Gamma}|$ denotes the determinant of $\boldsymbol{\Gamma}$. The estimation data set contained $N=600$ points.

The results of the five density estimators for Examples 1 and 2 are listed in Table 1(a) and (b), respectively. For the PW PDF estimator, the kernel width was determined by the MSIE criterion (see for example [2]). For the RSDE and the proposed FCR-MSIE estimator, the kernel widths were empirically set through trial and error. The results for the other two SKD estimators are quoted from [21,22], respectively. It is seen that the proposed algorithm can construct sparse kernel density estimates with the competitive accuracy to the PW estimator and the other three existing SKD estimators. Our proposed FCR-MSIE estimator has a significant advantage in that it offers a much lower complexity in constructing PDF estimate than the three existing SKD estimators of [21–23].

Table 1
Performance comparison of five kernel density estimators for Examples 1 and 2.

Method	L_1 test error (mean ± STD)	Kernel number (mean ± STD)
(a) Example 1		
PW	$(4.18 \pm 0.8) \times 10^{-3}$	500 ± 0
SKD estimator [21]	$(3.83 \pm 0.8) \times 10^{-3}$	11.9 ± 2.6
SKD estimator [22]	$(3.84 \pm 0.8) \times 10^{-3}$	15.3 ± 3.9
RSDE [23]	$(4.24 \pm 0.8) \times 10^{-3}$	129.4 ± 35.7
Proposed FCR-MISE	$(3.33 \pm 0.8) \times 10^{-3}$	25.1 ± 2.7
(b) Example 2		
PW	$(3.18 \pm 0.13) \times 10^{-5}$	600 ± 0
SKD estimator [21]	$(4.48 \pm 1.2) \times 10^{-5}$	14.9 ± 2.1
SKD estimator [22]	$(3.11 \pm 0.5) \times 10^{-5}$	9.4 ± 1.9
RSDE [23]	$(3.67 \pm 0.7) \times 10^{-5}$	29.4 ± 10.1
Proposed FCR-MISE	$(2.82 \pm 0.1) \times 10^{-5}$	19.4 ± 0.9

To illustrate the application of the proposed method, the three two-class classification examples are also presented. For each of these three examples, the training data set is provided, which is divided into the two-class training data sets, C_0 and C_1, respectively. The proposed method can readily be applied to estimate the two conditional PDFs, $\widehat{p}(\boldsymbol{x};\boldsymbol{\beta}_{C_0},\rho_{C_0}|C_0)$ and $\widehat{p}(\boldsymbol{x};\boldsymbol{\beta}_{C_1},\rho_{C_1}|C_1)$, based on the data sets C_0 and C_1, respectively. The Bayes decision rule given by

$$\begin{cases} \boldsymbol{x} \in C_0 & \text{if } \widehat{p}(\boldsymbol{x};\boldsymbol{\beta}_{C_0},\rho_{C_0}|C_0) \ge \widehat{p}(\boldsymbol{x};\boldsymbol{\beta}_{C_1},\rho_{C_1}|C_1), \\ \boldsymbol{x} \in C_1 & \text{otherwise}, \end{cases} \tag{32}$$

can be applied to the test data set to obtain the corresponding classification error rate. Gaussian kernel was adopted in all the following three examples.

Example 3. This was the synthetic two-class classification problem in a 2-D feature space [26]. The data set was taken from [27]. The training set contained 250 samples with 125 points for each class. The test set had 1000 points with 500 samples for each class. The optimal Bayes error rate based on the true probability distribution is known to be 8%. For the same data set, the test error rates of 10.6% and 9.3% were reported in [28], using a SVM classifier with 38 Gaussian kernels and a relevance vector machine classifier with four Gaussian kernels, respectively.

Table 2 lists the test classification results obtained by the PW estimator and the proposed FCR-MISE estimator. The widths of the two class conditional PDF estimates for the both algorithms were empirically set to minimise the test error rate. Fig. 1(a) and (b) depicts the classification boundaries obtained using the PW estimate and that of the proposed FCR-MISE, together with the locations of the four selected kernels for the FCR-MISE. We point out that the proposed FCR-MISE algorithm selected only two kernels for each class conditional PDF estimate, while the each PW based conditional PDF estimate contained the 125 kernels of the full training data set. Clearly, the proposed FCR-MISE algorithm achieved a comparable classification performance to the PW estimator, both are very close to the known optimal Bayes test error rate. The SKD estimators of [21,22] were also applied to this example in [22], and we quote the results of these two SKD estimators in Table 2 for comparison. It can be seen that the FCR-MISE algorithm achieved a comparable classification test performance to these two existing SKD estimators.

Example 4. The breast cancer data, taken from [29], has the input dimension of $m=9$. The data set contained 100 realisations, each having 200 training patterns and 77 test patterns. In [30], six state-of-the-arts classifiers were applied to the data set, and we quote the results of [30] in Table 3. For the first five classifiers studied in [30], the nonlinear Gaussian radial basis function (RBF) network with five optimised RBF units was used. For the SVM

Table 2
Performance comparison of four kernel density estimators for Example 3.

Method	$\widehat{p}(\bullet\|C0)$		$\widehat{p}(\bullet\|C1)$		Test error rate (%)
	Kernel number	Kernel width	Kernel number	Kernel width	
PW	125	0.24	125	0.24	8.1
Proposed FCR-MISE	2	0.13	2	0.13	8.3
SKD estimator [21]	5	0.20	4	0.20	8.3
SKD estimator [22]	6	0.28	5	0.28	8.0

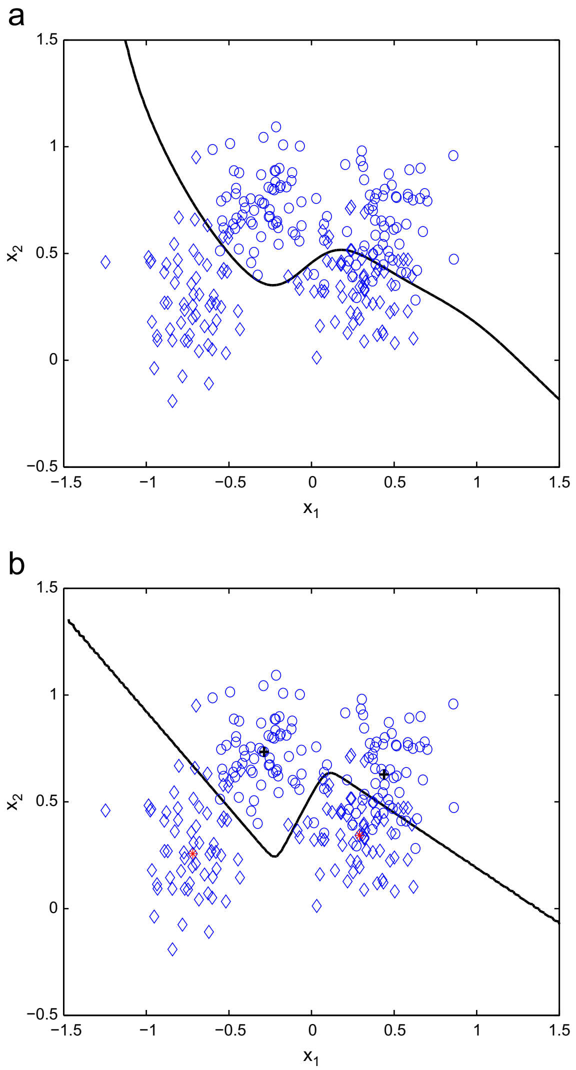

Fig. 1. Decision boundaries for the synthetic 2-class 2-D data set with circles and diamonds representing respectively the two class data points: (a) PW estimator using the full training data set for two class conditional PDF estimates, and (b) the proposed FCR-MISE using 4 selected kernels, 2 for each class conditional PDF estimate, represented by + and ⁎, respectively.

Table 3
Average misclassification rate in % over the 100 realisations of the Breast Cancer test data set and model size.

Method	Misclassification rate	Model size
RBF	27.6 ± 4.7	5
Adaboost with RBF	30.4 ± 4.7	5
AdaBoost-Reg	26.5 ± 4.5	5
LP-Reg-AdaBoost	26.8 ± 6.1	5
QP-Reg-AdaBoost	25.9 ± 4.6	5
SVM with RBF kernel	26.0 ± 4.7	Not available
Proposed FCR-MISE	26.1 ± 4.7	92 ± 0

Table 4
Average misclassification rate in % over the 100 realisations of the Titanic test data set and model size.

Method	Misclassification rate	Model size
RBF	23.3 ± 1.3	4
Adaboost with RBF	22.6 ± 1.2	4
AdaBoost-Reg	22.6 ± 1.2	4
LP-Reg-AdaBoost	24.0 ± 4.4	4
QP-Reg-AdaBoost	22.7 ± 1.1	4
SVM with RBF kernel	22.4 ± 1.0	Not available
Proposed FCR-MISE	22.2 ± 0.4	83.8 ± 6.8

classifier with Gaussian kernel, no average model size was reported in [30], but our experience with the SVM classifier suggests that it could likely contains around 100 or more kernels.

The classification results obtained by the proposed FCR-MISE algorithm are also listed in Table 3 for comparison. For the FCR-MISE algorithm, the two widths in the two conditional PDF estimates were set empirically as $\rho_{C_0} = 1.8$ and $\rho_{C_1} = 1.9$, respectively, for all the 100 realisations of the data set, and the model size for our method is the sum of the kernels in building the two conditional PDFs, selected from a total of 400 training patterns. Clearly the classification accuracy of our FCR-MISE algorithm is competitive, compared with the six state-of-the-arts classifiers studied in [30]. It is worth emphasising that the modelling paradigms of [30] are discriminative models constructed based on both the input and output (class label) information. By contrast, the proposed FCR-MISE algorithm only relies on the input information to construct each conditional PDF, and the total number of the kernels for constructing the Bayes classifier (32) is unavoidably larger than the discriminative classifiers of [30]. However, we believe that the classifier model size of our FCR-MISE algorithm is likely to be smaller than the SVM classifier. This further demonstrates the efficiency of our proposed FCR-MISE algorithm for estimating PDF.

Example 5. The Titanic data, also taken from [29], has the input dimension of $m = 3$. The data set contained 100 realisations, each having 150 training patterns and 2051 test patterns. Table 4 lists the classification results obtained by the proposed FCR-MISE algorithm in comparison with the results of the six classifiers quoted from [30]. The two widths used in the proposed FCR-MISE algorithm were set empirically as $\rho_{C_0} = 1.8$ and $\rho_{C_1} = 1.7$, respectively, for all 100 realisations. The model size for the FCR-MISE algorithm denotes the sum of the kernels used for the two conditional PDF estimates, selected from the total of 300 training patterns. From Table 4, it can be seen that the classification accuracy of the FCR-MISE method is competitive, compared with the six state-of-the-arts classifiers studied in [30]. We point out that the size of the Bayes classifier obtained by the FCR-MISE density estimator is most likely to be smaller than the SVM classifier, even though the latter is a discriminative model.

5. Conclusions

In this paper, a new sparse kernel density estimator has been derived using the forward constrained regression procedure. Our novel contribution is to derive a recursive algorithm which selects significant kernels one at time based on the minimum integrated square error criterion within the FCR procedure. The most significant advantage of our proposed FCR-MISE approach is that it has an extremely low computational complexity, since at each FCR step, only a single parameter is estimated using a closed-form solution developed in this contribution. Specifically, our proposed method has a computational complexity in the order of N for selecting a sparse kernel density estimate from the data set of size N. This compares very favourably with the most efficient existing sparse kernel density estimators, which have the computational complexity in the order of N^2. Numerical examples have been employed to demonstrate that the proposed approach can construct sparse kernel density estimates with competitive accuracy to the existing kernel density estimators.

Acknowledgements

X. Hong acknowledges the support of the UK EPSRC.

The authors acknowledge the financial support of King Abdulaziz University (Grant No. 1-4-1432/HiCi).

References

[1] G.J. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.
[2] B.W. Silverman, Density Estimation for Statistics and Data Analysis, Chapman and Hall, London, 1986.
[3] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[4] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.
[5] H. Wang, Robust control of the output probability density functions for multivariable stochastic systems with guaranteed stability, IEEE Trans. Autom. Control 44 (November (11)) (1999) 2103–2107.
[6] S. Chen, A.K. Samingan, B. Mulgrew, L. Hanzo, Adaptive minimum-BER linear multiuser detection for DS-CDMA signals in multipath channels, IEEE Trans. Signal Process. 49 (June (6)) (2001) 1240–1247.
[7] S. Chen, X. Hong, C.J. Harris, Particle swarm optimization aided orthogonal forward regression for unified data modelling, IEEE Trans. Evol. Comput. 14 (August (4)) (2010) 477–499.
[8] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (September (3)) (1962) 1066–1076.
[9] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B 39 (1) (1977) 1–38.
[10] J.A. Bilmes, A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, Technical Report ICSI-TR-97-021, University of California, Berkeley, 1998.
[11] B. Efron, R.J. Tibshirani, An Introduction to Bootstrap, Chapman & Hall, London, 1993.
[12] Z.R. Yang, S. Chen, Robust maximum likelihood training of heteroscedastic probabilistic neural networks, Neural Networks 11 (June (4)) (1998) 739–747.
[13] M. Svensén, C.M. Bishop, Robust Bayesian mixture modelling, Neurocomputing 64 (March) (2005) 235–252.
[14] C. Archambeau, M. Verleysen, Robust Bayesian clustering, Neural Networks 20 (January (1)) (2007) 129–138.
[15] J. Weston, A. Gammerman, M.O. Stitson, V. Vapnik, V. Vovk, C. Watkins, Support vector density estimation, in: B. Schölkopf, C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1999, pp. 293–306.
[16] V. Vapnik, S. Mukherjee, Support vector method for multivariate density estimation, in: S. Solla, T. Leen, K.R. Müller (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2000, pp. 659–665.
[17] A. Choudhury, Fast Machine Learning Algorithms for Large Data, Ph.D. Dissertation, School of Engineering Sciences, University of Southampton, 2002.
[18] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their applications to non-linear system identification, Int. J. Control 50 (5) (1989) 1873–1896.
[19] X. Hong, P.M. Sharkey, K. Warwick, Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic, IEE Proc. Control Theory Appl. 150 (3) (2003) 245–254.
[20] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modelling using forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. Part B 34 (2) (2004) 898–911.
[21] S. Chen, X. Hong, C.J. Harris, Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization, IEEE Trans. Syst. Man Cybernet. Part B 34 (August (4)) (2004) 1708–1717.
[22] S. Chen, X. Hong, C.J. Harris, An orthogonal forward regression techniques for sparse kernel density estimation, Neurocomputing 71 (January (46)) (2008) 931–943.
[23] M. Girolami, C. He, Probability density estimation from optimally condensed data samples, IEEE Trans. Pattern Anal. Mach. Intell. 25 (October (10)) (2003) 1253–1264.
[24] S.W. Scott, Parametric statistical modeling by minimum integrated square error, Technometrics 43 (August (3)) (2001) 274–285.
[25] X. Hong, C.J. Harris, A mixture of experts network structure construction algorithm for modelling and control, Appl. Intell. 16 (1) (2002) 59–69.
[26] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, UK, 1996.
[27] 〈http://www.stats.ox.ac.uk/PRNN/〉.
[28] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.
[29] 〈http://www.fml.tuebingen.mpg.de/members/raetsch/benchmark〉.
[30] G. Rätsch, T. Onoda, K.R. Müller, Soft margins for AdaBoost, Mach. Learn. 42 (3) (2001) 287–320.

Xia Hong received her B.Sc. and M.Sc. degrees from the National University of Defence Technology, China, in 1984 and 1987, respectively, and her Ph.D. degree from the University of Sheffield, UK, in 1998, all in automatic control.

She worked as a research assistant in Beijing Institute of Systems Engineering, Beijing, China from 1987 to 1993. She worked as a research fellow in the School of Electronics and Computer Science at the University of Southampton, UK, from 1997 to 2001. Since 2001, She has been with the School of Systems Engineering, the University of Reading, UK, where she currently holds a Readership post. She is actively engaged in research into nonlinear systems identification, data modelling, estimation and intelligent control, neural networks, pattern recognition, learning theory and their applications. She has published over 100 research papers, and coauthored a research book. She was awarded a Donald Julius Groen Prize by the Institution of Mechanical Engineers in 1999.

Sheng Chen received his B.Eng. degree from the East China Petroleum Institute, China, in January 1982, and his Ph.D. degree from the City University, London, in September 1986, both in control engineering. In 2005, he was awarded the higher doctorate degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK.

From 1986 to 1999, He held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with Electronics and Computer Science, the University of Southampton, UK, where he currently holds the post of Professor in Intelligent Systems and Signal Processing. His research interests include adaptive signal processing, wireless communications, modelling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods and optimisation. He has published over 460 research papers.

He is a Fellow of IEEE and a Fellow of IET. He is a Distinguished Adjunct Professor at the King Abdulaziz University, Jeddah, Saudi Arabia. He is an ISI highly cited researcher in the engineering category (March 2004).

Abdulrohman Qatawneh received his B.S. degree from the University of Jordan in 1994 and his M.S. degree from the Jordan University of Science and Technology in 1997, both in electrical engineering. He received his Ph.D. degree in telecommunication engineering from the Polytechnic University of Madrid in 2006.

He is currently an assistant professor at the King Abdulaziz University, Saudi Arabia. His research interests include mobile communications systems, differential MIMO coding and computational intelligence algorithms.

Khaled Daqrouq received his B.S. and M.S. degrees in biomedical engineering from the Wroclaw University of Technology in Poland, in 1995, as one certificate, and his Ph.D. degree in electronics engineering from the Wroclaw University of Technology, Poland, in 2001.

He is currently an associate professor at the King Abdulaziz University, Saudi Arabia. His research interests are in ECG signal processing, wavelet transform applications for speech recognition, as well as in the general area of speech and audio signal processing and improving auditory prostheses in noisy environments.

Muntasir Sheikh received the B.S. degree in electronics and communication engineering from the King Abdulaziz University, Saudi Arabia, in 1987, the M.Sc. degree in RF communications engineering from the University of Bradford, UK, in 1991, and the Ph.D. degree in electrical engineering from the University of Arizona, USA, in 1999.

Since 1999, he has been with the King Abdulaziz University, Saudi Arabia, where currently he is an assistant professor. His research interests are in remotely monitoring for security applications, and robotics.

Ali Morfeq received the B.S. degree in computer engineering from the King Abdulaziz University, Saudi Arabia, in 1982, the M.S. degree in computer engineering from the Oregon state University, Corvallis, USA in 1985, and the Ph.D. degree in computer science from the University of Colorado, Boulder, USA in 1990.

Since 1990, he has been with the King Abdulaziz University, Saudi Arabia, where currently he is an assistant professor and the chair of the Electrical & Computer Engineering Department, Faculty of Engineering. His research interests are in software engineering, and software systems for hospital applications.