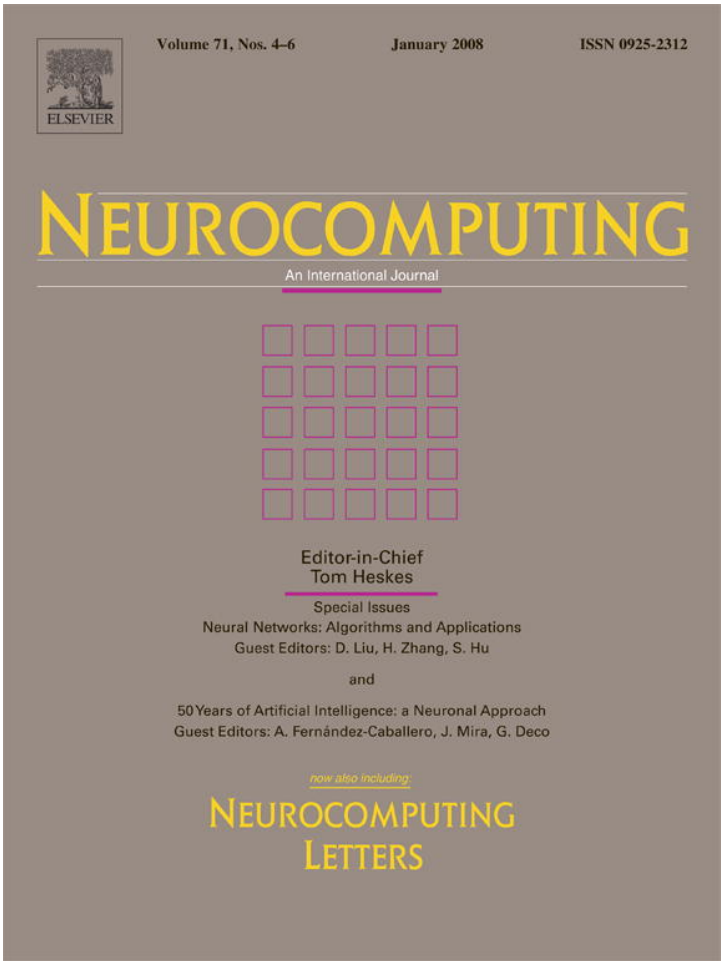


Volume 71, Nos. 4–6 January 2008 ISSN 0925-2312

ELSEVIER

NEUROCOMPUTING
An International Journal

Editor-in-Chief
Tom Heskes

Special Issues
Neural Networks: Algorithms and Applications
Guest Editors: D. Liu, H. Zhang, S. Hu

and

50 Years of Artificial Intelligence: a Neuronal Approach
Guest Editors: A. Fernández-Caballero, J. Mira, G. Deco

now also including:
NEUROCOMPUTING
LETTERS

Available online at www.sciencedirect.com
ScienceDirect

ELSEVIER

Neurocomputing 71 (2008) 931–943

NEUROCOMPUTING

www.elsevier.com/locate/neucom

An orthogonal forward regression technique for sparse kernel density estimation

S. Chen[a,*], X. Hong[b], C.J. Harris[a]

[a]*School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK*
[b]*School of Systems Engineering, University of Reading, Reading RG6 6AY, UK*

Received 8 November 2006; received in revised form 15 February 2007; accepted 23 February 2007
Communicated by J. Zhang
Available online 16 March 2007

Abstract

Using the classical Parzen window (PW) estimate as the desired response, the kernel density estimation is formulated as a regression problem and the orthogonal forward regression technique is adopted to construct sparse kernel density (SKD) estimates. The proposed algorithm incrementally minimises a leave-one-out test score to select a sparse kernel model, and a local regularisation method is incorporated into the density construction process to further enforce sparsity. The kernel weights of the selected sparse model are finally updated using the multiplicative nonnegative quadratic programming algorithm, which ensures the nonnegative and unity constraints for the kernel weights and has the desired ability to reduce the model size further. Except for the kernel width, the proposed method has no other parameters that need tuning, and the user is not required to specify any additional criterion to terminate the density construction procedure. Several examples demonstrate the ability of this simple regression-based approach to effectively construct a SKD estimate with comparable accuracy to that of the full-sample optimised PW density estimate.

Keywords: Probability density function; Parzen window estimate; Sparse kernel modelling; Orthogonal forward regression; Cross validation; Regularisation; Multiplicative nonnegative quadratic programming

1. Introduction

In this contribution, we consider the nonparametric approach for estimating the probability density function (PDF) based on a realisation sample drawn from the underlying density [2,18,21]. The best-known nonparametric density estimation technique is perhaps the classical Parzen window (PW) estimate [18], which is remarkably simple and accurate. As the PW estimate, also known as the kernel density estimate, employs the full data sample set in defining density estimate for subsequent observation, its computational cost for testing scales directly with the sample size. In today's data-rich environment, this may become a practical difficulty in employing the PW estimator. It also motivates the research on the so-called sparse kernel density (SKD) estimation techniques. The support vector machine (SVM) method with its ability to perform function approximations in high-dimensional spaces from finite data using sparse representations has been proposed as a promising tool for SKD estimation [16,24,25]. More recently, an interesting SKD estimation technique referred to as the reduced set density estimator (RSDE) is proposed [9]. Similar to the SVM methods, this technique employs the full data sample set as the kernel set and tries to make as many kernel weights to (near) zero as possible, and thus to obtain a sparse representation. The difference with the SVM approach is that it adopts directly the criterion of the integrated squared error between the unknown underlying density and the kernel density estimate, calculated on the training sample set.

A regression-based SKD estimation method was reported in [8]. By converting the kernels into the associated cumulative distribution functions and using the empirical distribution function calculated on the training data sample

[*] Corresponding author.
E-mail addresses: sqc@ecs.soton.ac.uk (S. Chen), x.hong@reading.ac.uk (X. Hong), cjh@ecs.soton.ac.uk (C.J. Harris).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.02.008

set as the desired response, just like the SVM-based density estimation [16,24,25], this technique transfers the kernel density estimation into a regression problem and it selects SKD estimates based on an orthogonal forward regression (OFR) algorithm that incrementally minimises the usual training mean square error (MSE). This sparse density estimation algorithm is computationally efficient, and the results shown in [8] have demonstrated the potential of this method. In order to terminate the kernel density construction procedure at an appropriate stage, this method requires additional termination criteria, and it was suggested in [8] that the minimum descriptive length [11] or Akaike's information criterion [1] is adopted to help terminating the density construction process. However, the empirical results in [8] showed that models so obtained were still often oversized and at the end, a maximum model size was imposed in order to avoid an over-fitted model. Motivated by our previous work on sparse regression modelling [5,7], recently we have extended the work of [8] and proposed an efficient construction algorithm for SKD estimation using the OFR based on a leave-one-out (LOO) test score and local regularisation [6]. This method is capable of constructing very SKD estimates with comparable accuracy to that of the full-sample optimised PW density estimate. Moreover, the process is fully automatic and the user is not required to specify any additional criterion to terminate the density construction procedure [6].

In this contribution, we propose a simple regression-based alternative for SKD estimation. In the works of Choudhury [8] and Chen et al. [6], the "regressors" are the cumulative distribution functions of the corresponding kernels and the desired response is the empirical distribution function calculated on the training data sample set. Computing the cumulative distribution functions from the kernels can be inconvenient and may be difficult for certain types of kernels. We propose to directly use the kernels as the regressors and to view the PW estimate as the desired response. The same OFR algorithm based on the LOO test score and local regularisation [6,7] can readily be employed to select an SKD estimate. As a probability density estimate, the kernel weights must satisfy nonnegative and unity constraints. In the work of Chen et al. [6], the unity constraint is met by normalising the kernel weight vector of the selected model, and the nonnegative constraint is ensured by adding a test to the OFR selection procedure of Chen et al. [7]. In each selection stage, a candidate that causes the resulting kernel weight vector to have negative elements, if included, will not be considered at all. This nonnegative test imposes considerable computational cost to the OFR procedure. In our proposed alternative, we simply use the efficient OFR selection procedure of Chen et al. [7] to construct a sparse model. The kernel weights of the final sparse model are then computed using the multiplicative nonnegative quadratic programming (MNQP) algorithm [20], which will ensure that the kernel weights meet the required nonnegative and unity constraints. The MNQP algorithm additionally has a desired property that it will force some kernel weights to (near) zero values [9,20] and thus further makes the model sparser. Our empirical results involving several numerical examples show that the proposed method offers a viable simple alternative to the regression-based SKD estimation.

The remainder of the paper is organised as follows. Section 2 formulates the kernel density estimation directly as a regression problem, where we also point out why it is more desirable to use the PW estimate as the target function of the unknown true PDF than to use the empirical distribution function as the target function of the unknown true cumulative distribution function. Our proposed combined OFR-LOO-LR and MNQP algorithm for SKD estimation is detailed in Section 3. Several numerical examples are experimented in Section 4 to illustrate the effectiveness of the proposed simple algorithm in constructing an SKD estimate with comparable accuracy to that of the PW estimate. The paper concludes at Section 5.

2. A regression-based approach for kernel density estimation

Based on a finite data sample set $\mathscr{D} = \{\mathbf{x}_k\}_{k=1}^N$ drawn from a density $p(\mathbf{x})$, where $\mathbf{x}_k \in \mathscr{R}^m$, the task is to estimate the unknown density $p(\mathbf{x})$ using the kernel density estimate of the form

$$\hat{p}(\mathbf{x};\boldsymbol{\beta},\rho) = \sum_{k=1}^{N} \beta_k K_\rho(\mathbf{x},\mathbf{x}_k) \tag{1}$$

with the constraints

$$\beta_k \geqslant 0, \quad 1 \leqslant k \leqslant N, \tag{2}$$

and

$$\boldsymbol{\beta}^{\mathrm{T}}\mathbf{1}_N = 1, \tag{3}$$

where $\boldsymbol{\beta} = [\beta_1 \beta_2 \cdots \beta_N]^{\mathrm{T}}$ is the kernel weight vector, $\mathbf{1}_N$ denotes the vector of ones with dimension N, and $K_\rho(\bullet,\bullet)$ is a chosen kernel function with the kernel width ρ. In this study, we use the Gaussian kernel of the form

$$K_\rho(\mathbf{x},\mathbf{x}_k) = \frac{1}{(2\pi\rho^2)^{m/2}} \mathrm{e}^{-\|\mathbf{x}-\mathbf{x}_k\|^2/2\rho^2}. \tag{4}$$

However, many other types of kernel functions can also be used in the density estimate (1).

The well-known PW estimate $\hat{p}(\mathbf{x};\boldsymbol{\beta}_{\mathrm{Par}},\rho_{\mathrm{Par}})$ is obtained by setting all the elements of $\boldsymbol{\beta}_{\mathrm{Par}}$ to $1/N$. The optimal kernel width ρ_{Par} is typically determined via cross validation [17,22]. The PW estimate in fact can be derived as the maximum likelihood estimator using the divergence-based criterion [14]. The negative cross-entropy or divergence between the true density $p(\mathbf{x})$ and the estimate $\hat{p}(\mathbf{x};\boldsymbol{\beta},\rho)$ is defined as

$$\int_{\mathscr{R}^m} p(\mathbf{u}) \log \hat{p}(\mathbf{u};\boldsymbol{\beta},\rho)\, \mathrm{d}\mathbf{u} \approx \frac{1}{N}\sum_{k=1}^{N} \log \hat{p}(\mathbf{x}_k;\boldsymbol{\beta},\rho)$$

$$= \frac{1}{N}\sum_{k=1}^{N}\log\left(\sum_{n=1}^{N}\beta_n K_\rho(\mathbf{x}_k, \mathbf{x}_n)\right). \tag{5}$$

Minimising this divergence subject to the constraints (2) and (3) leads to $\beta_n = 1/N$ for $1 \leqslant n \leqslant N$, i.e. the PW estimate. Because of this property, we may view the PW estimate as the "observation" of the true density contaminated by some "observation noise", namely

$$\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{\text{Par}}, \rho_{\text{Par}}) = p(\mathbf{x}) + \tilde{\varepsilon}(\mathbf{x}). \tag{6}$$

Thus the generic kernel density estimation problem (1) can be viewed as the following regression problem with the PW estimate as the "desired response"

$$\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{\text{Par}}, \rho_{\text{Par}}) = \sum_{k=1}^{N}\beta_k K_\rho(\mathbf{x}, \mathbf{x}_k) + \varepsilon(\mathbf{x}) \tag{7}$$

subject to the constraints (2) and (3), where $\varepsilon(\mathbf{x})$ is the modelling error at $\mathbf{x}$.

Define $y_k = \hat{p}(\mathbf{x}_k; \boldsymbol{\beta}_{\text{Par}}, \rho_{\text{Par}})$, $\boldsymbol{\phi}(k) = [K_{k,1} K_{k,2} \cdots K_{k,N}]^{\text{T}}$ with $K_{k,i} = K_\rho(\mathbf{x}_k, \mathbf{x}_i)$, and $\varepsilon(k) = \varepsilon(\mathbf{x}_k)$. Then model (7) at the data point $\mathbf{x}_k \in \mathscr{D}$ can be expressed as

$$y_k = \hat{y}_k + \varepsilon(k) = \boldsymbol{\phi}^{\text{T}}(k)\boldsymbol{\beta} + \varepsilon(k). \tag{8}$$

Model (8) is obviously a standard regression model, and over the training data set $\mathscr{D}$ it can be written in the matrix form

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \tag{9}$$

with the following additional notations $\boldsymbol{\Phi} = [K_{i,k}] \in \mathscr{R}^{N\times N}$, $1 \leqslant i, k \leqslant N$, $\boldsymbol{\varepsilon} = [\varepsilon(1)\varepsilon(2)\cdots\varepsilon(N)]^{\text{T}}$, and $\mathbf{y} = [y_1 y_2 \cdots y_N]^{\text{T}}$. For convenience, we will denote the regression matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N]$ with $\boldsymbol{\phi}_k = [K_{1,k} K_{2,k} \cdots K_{N,k}]^{\text{T}}$. Note that $\boldsymbol{\phi}_k$ is the kth column of $\boldsymbol{\Phi}$, while $\boldsymbol{\phi}^{\text{T}}(k)$ is the kth row of $\boldsymbol{\Phi}$.

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

$$\boldsymbol{\Phi} = \mathbf{W}\mathbf{A}, \tag{10}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{11}$$

and

$$\mathbf{W} = [\mathbf{w}_1 \mathbf{w}_2 \cdots \mathbf{w}_N] \tag{12}$$

with orthogonal columns satisfying $\mathbf{w}_i^{\text{T}}\mathbf{w}_j = 0$, if $i \neq j$. The regression model (9) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\mathbf{g} + \boldsymbol{\varepsilon}, \tag{13}$$

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

$$\hat{y}_k = \mathbf{w}^{\text{T}}(k)\mathbf{g}, \tag{14}$$

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

Before turning to the proposed SKD estimation algorithm, a comparison with the previous regression-based approach for SKD estimation is offered. In most of the SKD estimation techniques [6,8,16,24,25], the kernel density estimation problem (1) is reformulated into a regression problem by using the empirical distribution function as the desired response of the true cumulative distribution function, which is defined as

$$F(\mathbf{x}) = \int_{-\infty}^{\mathbf{x}} p(\mathbf{u})\,\mathrm{d}\mathbf{u}, \tag{15}$$

and converting the kernels into corresponding cumulative distribution functions, that is,

$$q_\rho(\mathbf{x}, \mathbf{x}_k) = \int_{-\infty}^{\mathbf{x}} K_\rho(\mathbf{u}, \mathbf{x}_k)\,\mathrm{d}\mathbf{u}. \tag{16}$$

The empirical distribution function $\hat{F}(\mathbf{x}; N)$ is defined by

$$\hat{F}(\mathbf{x}; N) = \frac{1}{N}\sum_{k=1}^{N}\prod_{j=1}^{m}\theta(x_j - x_{j,k}) \tag{17}$$

with

$$\theta(x) = \begin{cases} 1, & x > 0, \\ 0, & x \leqslant 0, \end{cases} \tag{18}$$

where $\mathbf{x}_k = [x_{1,k} x_{2,k} \cdots x_{m,k}]^{\text{T}} \in \mathscr{D}$. Thus, the regression modelling for density estimation is expressed as

$$\hat{F}(\mathbf{x}; N) = \sum_{k=1}^{N}\beta_k q_\rho(\mathbf{x}, \mathbf{x}_k) + \varepsilon(\mathbf{x}). \tag{19}$$

Our regression-based approach can use any type of kernel function and it is computationally simpler, as it does not need to compute the values of "regressors" (16) on the training data set $\mathscr{D}$. Computing the values of the PW estimator on $\mathscr{D}$ is no more complex than calculating the values of $\hat{F}(\mathbf{x}; N)$ on $\mathscr{D}$. The only drawback of using the PW estimate is that the kernel width for the PW estimator must be determined. The most significant observation, however, is that the use of the PW estimate as the target function of the unknown true PDF is theoretically more sound than the use of the empirical distribution function as the target function of the unknown true cumulative distribution function. As mentioned previously, $\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{\text{Par}}, \rho_{\text{Par}})$ is the maximum likelihood estimator of $p(\mathbf{x})$ based on the divergence between the true density and the PW estimate, while there exists no similar optimality property between $\hat{F}(\mathbf{x}; N)$ and $F(\mathbf{x})$.

3. The proposed regression-based technique for sparse density estimation

Our aim is to seek a sparse representation for $\hat{p}(\mathbf{x}; \boldsymbol{\beta}, \rho)$ with most elements of $\boldsymbol{\beta}$ being zero and yet maintaining a comparable test performance or generalisation capability to that of the full-sample PW estimate. Since this density construction problem is formulated as a constrained regression one, we can apply the OFR algorithm based on the LOO test score and local regularisation [7] to select a sparse model and use the MNQP algorithm [9,20] to calculate the kernel weights for the selected model. This combined OFR-LOO-LR and MNQP algorithm for SKD estimation is now summarised.

3.1. Orthogonal forward regression with leave-one-out test score and local regularisation

First, we point out that the local regularisation aided least squares solution for the weight parameter vector **g** is obtained by minimising the following regularised error criterion [3]

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

where $\boldsymbol{\lambda} = [\lambda_1 \lambda_2 \cdots \lambda_N]^{\mathrm{T}}$ is the regularisation parameter vector, which is optimised based on the evidence procedure [13] with the iterative updating formulas [3,5,7]

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

where

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

Usually a few iterations (typically less than 10) are sufficient to find a (near) optimal $\boldsymbol{\lambda}$. The detailed derivation of the updating formulas (21) and (22) can be found in [3]. The use of multiple-regularisers or local regularisation is known to be capable of providing very sparse solutions [3,23].

It is highly desired to select a sparse model by directly optimising the model generalisation capability, rather than minimising the training MSE. The algorithm achieves this objective by incrementally minimising the LOO test score, which is a measure of the model's generalisation performance [10,12,15,17]. At the nth stage of the OFR procedure, an n-term model is selected. It can be shown that the LOO test error, denoted as $\varepsilon_{n,-k}(k)$, for the selected n-term model is [7,12]

$$\varepsilon_{n,-k}(k) = \frac{\varepsilon_n(k)}{\eta_n(k)}, \tag{23}$$

where $\varepsilon_n(k)$ is the n-term modelling error and $\eta_n(k)$ is the associated LOO error weighting. The mean square LOO error for the model with a size n is defined by

$$J_n = \frac{1}{N}\sum_{k=1}^{N} \varepsilon_{n,-k}^2(k) = \frac{1}{N}\sum_{k=1}^{N} \frac{\varepsilon_n^2(k)}{\eta_n^2(k)}. \tag{24}$$

This LOO test score can be computed efficiently due to the fact that the n-term model error $\varepsilon_n(k)$ and the associated LOO error weighting $\eta_n(k)$ can be calculated recursively according to [7,12]

$$\varepsilon_n(k) = y_k - \sum_{i=1}^{n} w_{k,i} g_i = \varepsilon_{n-1}(k) - w_{k,n} g_n \tag{25}$$

and

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

respectively.

The subset model selection procedure is carried as follows: at the nth stage of the selection procedure, a model term is selected among the remaining n to N candidates if the resulting n-term model produces the smallest LOO test score J_n. The selection procedure is terminated when

$$J_{n_s+1} \geqslant J_{n_s}, \tag{27}$$

yielding an n_s-term sparse model. It has been shown in [12] that the LOO statistic J_n is at least locally convex with respect to the model size n. That is, there exists an "optimal" model size n_s such that for $n \leqslant n_s$ J_n decreases as n increases while condition (27) holds. This property is extremely useful, as it enables the selection procedure to be automatically terminated with an n_s-term model, without the need for the user to specify a separate termination criterion. The sparse model selection procedure is summarised as follows.

Initialisation: Set $\lambda_i = 10^{-6}$ for $1 \leqslant i \leqslant N$, and set iteration index $I = 1$.

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

$$\varepsilon_0(k) = y_k \quad \text{and} \quad \eta_0(k) = 1, \quad 1 \leqslant k \leqslant N,$$
$$J_0 = \frac{1}{N}\mathbf{y}^{\mathrm{T}}\mathbf{y} = \frac{1}{N}\sum_{k=1}^{N} y_k^2, \tag{28}$$

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

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

3.2. Multiplicative nonnegative quadratic programming for kernel weights

After the structure determination using the above OFR-LOO-LR algorithm, we obtain an n_s-term subset kernel model, where $n_s \ll N$. Let $\mathbf{A}_{n_s}$ denote the subset matrix of $\mathbf{A}$, corresponding to the selected n_s-term subset model. The kernel weight vector $\boldsymbol{\beta}_{n_s}$, computed from $\mathbf{A}_{n_s}\boldsymbol{\beta}_{n_s} = \mathbf{g}_{n_s}$, may not satisfy the nonnegative constraint (2) and the unity constraint (3). We propose to use the MNQP algorithm [9,20] to calculate $\boldsymbol{\beta}_{n_s}$ instead. Since n_s is very small, the extra computation involved is small. Formally, this task is defined as follows. Find $\boldsymbol{\beta}_{n_s}$ for the regression model

$$\mathbf{y} = \mathbf{\Phi}_{n_s}\boldsymbol{\beta}_{n_s} + \varepsilon \tag{29}$$

subject to the constraints

$$\beta_i \geqslant 0, \quad 1 \leqslant i \leqslant n_s, \tag{30}$$

$$\boldsymbol{\beta}_{n_s}^{\mathrm{T}}\mathbf{1}_{n_s} = 1, \tag{31}$$

where $\mathbf{\Phi}_{n_s}$ is the selected subset regression matrix and $\boldsymbol{\beta}_{n_s}^{\mathrm{T}} = [\beta_1\beta_2\cdots\beta_{n_s}]$. The kernel weight vector can be obtained by solving the following constrained nonnegative quadratic programming

$$\min_{\boldsymbol{\beta}_{n_s}}\{\tfrac{1}{2}\boldsymbol{\beta}_{n_s}^{\mathrm{T}}\mathbf{B}_{n_s}\boldsymbol{\beta}_{n_s} - \mathbf{v}_{n_s}^{\mathrm{T}}\boldsymbol{\beta}_{n_s}\}$$
$$\text{s.t.}\boldsymbol{\beta}_{n_s}^{\mathrm{T}}\mathbf{1}_{n_s} = 1 \quad \text{and} \quad \beta_i \geqslant 0, \quad 1 \leqslant i \leqslant n_s, \tag{32}$$

where $\mathbf{B}_{n_s} = \mathbf{\Phi}_{n_s}^{\mathrm{T}}\mathbf{\Phi}_{n_s} = [b_{i,j}] \in \mathscr{R}^{n_s \times n_s}$ is the related design matrix and $\mathbf{v}_{n_s} = \mathbf{\Phi}_{n_s}^{\mathrm{T}}\mathbf{y} = [v_1\, v_2 \cdots v_{n_s}]^{\mathrm{T}}$. Although there exists no closed-form solution for this optimisation problem, the solution can readily be obtained iteratively using a modified version of the MNQP algorithm [20].

Since the elements of $\mathbf{B}_{n_s}$ and $\mathbf{v}_{n_s}$ are strictly positive, the Lagrangian for the above problem can be formed as [9]

$$\mathscr{L} = \tfrac{1}{2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s} b_{i,j}\frac{\beta_j^{(t)}\left(\beta_i^{(t+1)}\right)^2}{\beta_i^{(t)}} - \sum_{i=1}^{n_s} v_i\beta_i^{(t+1)} - h^{(t)}\left(\sum_{i=1}^{n_s}\beta_i^{(t+1)} - 1\right), \tag{33}$$

where the superindex ${}^{(t)}$ denotes the iteration index and h is the Lagrangian multiplier. Setting

$$\frac{\partial\mathscr{L}}{\partial\beta_i^{(t+1)}} = 0 \quad \text{and} \quad \frac{\partial\mathscr{L}}{\partial h^{(t)}} = 0 \tag{34}$$

leads to the following updating equations

$$c_i^t = \beta_i^t\left(\sum_{j=1}^{n_s} b_{i,j}\beta_j^{(t)}\right)^{-1}, \quad 1 \leqslant i \leqslant n_s, \tag{35}$$

$$h^{(t)} = \left(\sum_{i=1}^{n_s} c_i^{(t)}\right)^{-1}\left(1 - \sum_{i=1}^{n_s} c_i^{(t)} v_i\right), \tag{36}$$

$$\beta_i^{(t+1)} = c_i^{(t)}\left(v_i + h^{(t)}\right). \tag{37}$$

It is easy to check that if $\boldsymbol{\beta}_{n_s}^{(t)}$ meets the constraints (30) and (31), $\boldsymbol{\beta}_{n_s}^{(t+1)}$ updated according to (35)–(37) also satisfies (30) and (31). The initial condition can be set as $\beta_i^{(0)} = 1/n_s$, $1 \leqslant i \leqslant n_s$, or chosen to be the normalised kernel weight vector obtained by the OFR-LOO-LR algorithm with those negative elements replaced by a small positive number. During the iterative procedure, some of the kernel weights may be driven to (near) zero. The corresponding kernels can then be removed from the kernel model, leading to a further reduction in the subset model size.

4. Numerical examples

Six examples were used in the simulation to test the proposed combined OFR-LOO-LR and MNQP algorithm and to compare its performance with the PW estimator. Comparisons with other existing SKD estimation techniques were also given by quoting the results from the existing literatures. The value of the kernel width ρ used was determined by test performance via cross validation. The first five cases were the density estimation problems. In each of these cases, a data set of N randomly drawn samples was used to construct kernel density estimates, and a separate test data set of $N_{\text{test}} = 10,000$ samples was used to calculate either the L_2 test error or the L_1 test error for the resulting estimate according to

$$L_2 = \frac{1}{N_{\text{test}}}\sum_{k=1}^{N_{\text{test}}} |p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k;\boldsymbol{\beta},\rho)|^2, \tag{38}$$

and

$$L_1 = \frac{1}{N_{\text{test}}}\sum_{k=1}^{N_{\text{test}}} |p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k;\boldsymbol{\beta},\rho)|, \tag{39}$$

respectively. The experiment was repeated by N_{run} different random runs for each example. The sixth example was a two-class two-dimensional classification problem taken from [19].

Example 1. This was a one-dimensional example, and the density to be estimated was the mixture of eight Gaussian distributions given by

$$p(x) = \tfrac{1}{8}\sum_{i=0}^{7}\frac{1}{\sqrt{2\pi}\sigma_i}\mathrm{e}^{-(x-\mu_i)^2/2\sigma_i^2} \tag{40}$$

with

$$\sigma_i = \sqrt{\left(\frac{2}{3}\right)^i}, \quad \mu_i = 3\left(\left(\frac{2}{3}\right)^i - 1\right), \quad 0 \leqslant i \leqslant 7. \tag{41}$$

The number of data points for density estimation was $N = 200$. The experiment was repeated $N_{\text{run}} = 200$ times. The optimal kernel widths were found to be $\rho = 0.17$ and 0.3 empirically for the PW estimate and the SKD estimate obtained using the combined OFR-LOO-LR and MNQP algorithm, respectively. Table 1 compares the performance of the two kernel density estimates, in terms of the L_2 test error and the number of kernels required. Fig. 1(a) depicts

Table 1
Performance of the Parzen window estimate and the sparse kernel density estimate in terms of L_2 test error and number of kernels required for the one-dimensional example of eight Gaussian mixture, quoted as mean ± standard deviation over 200 runs

Method	L_2 test error	kernel number
PW estimate	$(2.9311 \pm 2.0601) \times 10^{-3}$	200 ± 0
Proposed SKD estimate	$(3.0181 \pm 2.0991) \times 10^{-3}$	10.2 ± 1.6

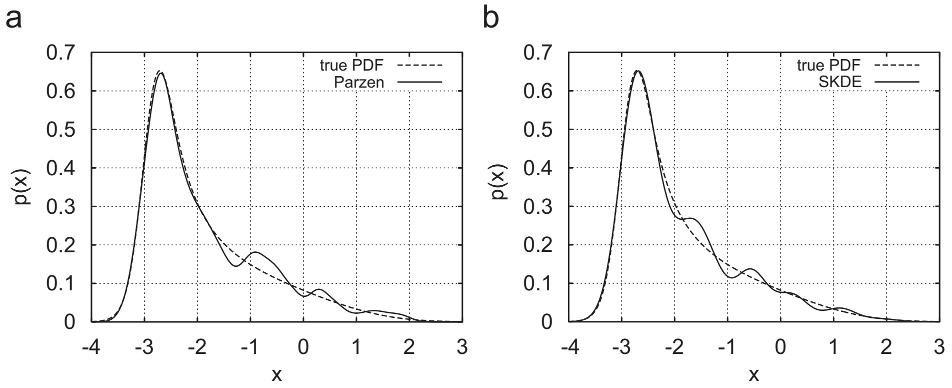


Fig. 1. (a) True density (dashed) and a Parzen window estimate (solid) and (b) true density (dashed) and a sparse kernel density estimate (solid), for the one-dimensional example of eight Gaussian mixture.

the PW estimate obtained in a run while Fig. 1(b) shows the SKD estimate obtained in a run, in comparison with the true distribution. For this one-dimensional example, it can be seen that the accuracy of the SKD estimate was comparable to that of the PW estimate, and the combined OFR-LOO-LR and MNQP algorithm realised sparse estimates with an average kernel number less than 6% of the data samples. The maximum and minimum numbers of kernels over 200 runs were 15 and 5, respectively, for the SKD estimator.

This example was used by Girolami and He [9] to test their RSDE algorithm. Under the same experimental conditions, the median value and the interquartile range for the L_2 test error obtained by the RSDE were 0.0035 and 0.0030, respectively, while the median value of the nonzero kernel weights was 13, with the maximum and minimum values of nonzero kernel weights over the 200 runs being 42 and 5, respectively. It can be seen that our proposed SKD estimator achieved a slightly better test performance with a sparser kernel density estimate, compared with the RSDE.

Example 2. The density to be estimated for this one-dimensional example was the mixture of Gaussian and Laplacian given by

$$p(x) = \frac{1}{2\sqrt{2\pi}} \mathrm{e}^{-(x-2)^2/2} + \frac{0.7}{4} \mathrm{e}^{-0.7|x+2|}. \tag{42}$$

The number of data points for density estimation was $N = 100$. The optimal kernel widths were found to be $\rho = 0.54$ and 1.1 empirically for the PW estimate and the proposed SKD estimate, respectively. The experiment was repeated $N_{\mathrm{run}} = 200$ times. Table 2 compares the performance of these two kernel density estimates, in terms of the L_1 test error and the number of kernels required. Fig. 2(a) plots a PW estimate obtained while Fig. 2(b) illustrates an SKD estimate obtained, in comparison with the true distribution. Again it can be seen that the accuracy of our proposed SKD estimate was comparable to that of the PW estimate for this one-dimensional example, and the combined OFR-LOO-LR and MNQP algorithm achieved sparse estimates with an average kernel number less than 6% of the data samples. The maximum and minimum numbers of kernels over 200 runs were 9 and 2, respectively, for the SKD estimator.

Table 2
Performance of the Parzen window estimate and the two sparse kernel density estimates in terms of L_1 test error and number of kernels required for the one-dimensional example of Gaussian and Laplacian mixture, quoted as mean ± standard deviation over 200 runs

Method	L_1 test error	kernel number
PW estimate	$(1.9503 \pm 0.5881) \times 10^{-2}$	100 ± 0
Proposed SKD estimate	$(1.9436 \pm 0.6208) \times 10^{-2}$	5.1 ± 1.3
SKD estimate of [6]	$(2.1785 \pm 0.7468) \times 10^{-2}$	4.8 ± 0.9

Our previous SKD estimator using the empirical distribution function as the desired response [6] was also applied to this example. Under the identical experimental conditions, the results obtained by this SKD estimator are also given in Table 2, where it can be seen that the both SKD estimators had the similar performance. The current SKD estimator had a slightly better L_1 test error performance than the previous SKD estimator, while the latter achieved a slightly sparser estimator than the former.

Example 3. The density to be estimated for this two-dimensional example was defined by the mixture of Gaussian and Laplacian given as follows

$$p(x,y) = \frac{1}{4\pi} \mathrm{e}^{-(x-2)^2/2} \mathrm{e}^{-(y-2)^2/2} + \frac{0.35}{8} \mathrm{e}^{-0.7|x+2|} \mathrm{e}^{-0.5|y+2|}. \tag{43}$$

Fig. 3 shows this density distribution and its contour plot. The estimation data set contained $N = 500$ samples, and the empirically found optimal kernel widths were $\rho = 0.42$ for the PW estimate and $\rho = 1.1$ for the proposed SKD estimate, respectively. The experiment was repeated $N_{\mathrm{run}} = 100$ times. Table 3 lists the L_1 test errors and the numbers of kernels required for these two density estimates. A typical PW estimate and a typical SKD estimate are

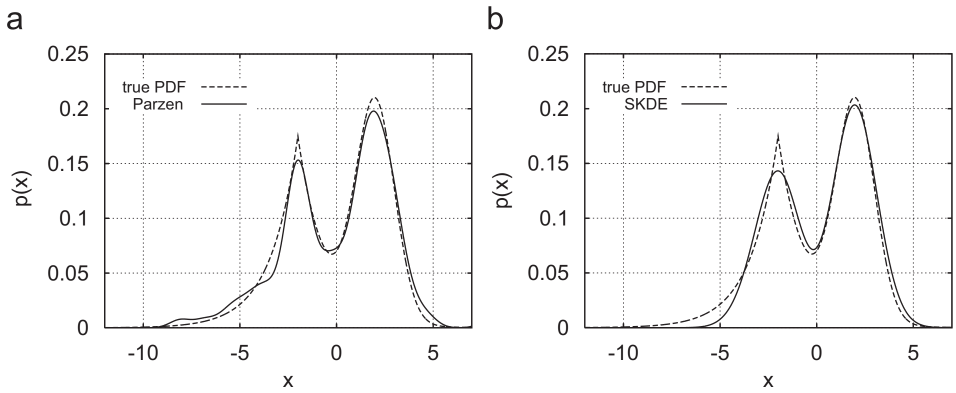


Fig. 2. (a) True density (dashed) and a Parzen window estimate (solid) and (b) true density (dashed) and a sparse kernel density estimate (solid), for the one-dimensional example of Gaussian and Laplacian mixture.

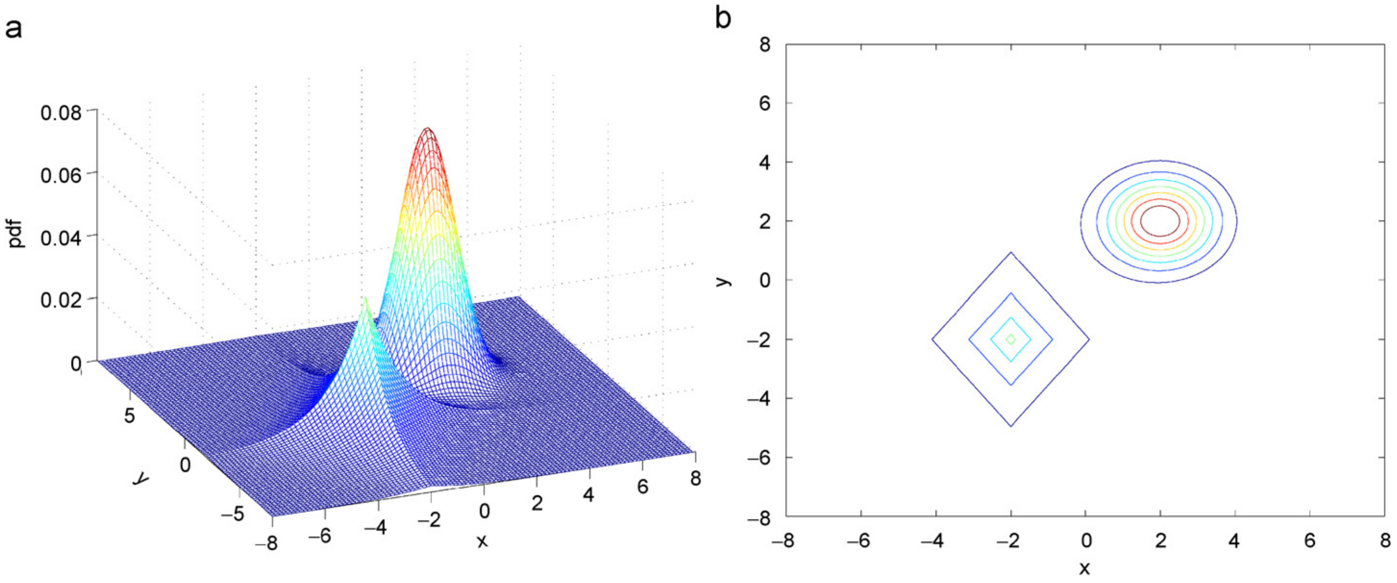


Fig. 3. True density (a) and contour plot (b) for the two-dimensional example of Gaussian and Laplacian mixture.

depicted in Figs. 4 and 5 respectively. Again, for this example, the two density estimates had comparable accuracies, but the proposed SKD estimation method achieved sparse estimates with an average number of required kernels less than 4% of the data samples. The maximum and minimum numbers of kernels over 100 runs were 25 and 8, respectively, for the proposed SKD estimator.

This example was used in [6] to test our previous SKD estimator using the empirical distribution function as the desired response under the same experimental conditions. The results obtained for this example quoted from [6] are also listed in Table 3 as a comparison. It can be seen from Table 3 that for this example both the SKD estimates had a similar test performance, while the SKD estimator of [6] achieved a slightly sparser estimate.

Example 4. For this second two-dimensional example, the true density to be estimated was defined by the mixture of five Gaussian distributions given as follows

$$p(x,y)=\sum_{i=1}^{5}\frac{1}{10\pi}\mathrm{e}^{-(x-\mu_{i,1})^2/2}\mathrm{e}^{-(y-\mu_{i,2})^2/2} \tag{44}$$

and the means of the five Gaussian distributions, $[\mu_{i,1}\ \mu_{i,2}]$, $1\leqslant i\leqslant 5$, were $[0.0\ -4.0]$, $[0.0\ -2.0]$, $[0.0\ 0.0]$, $[-2.0\ 0.0]$, and $[-4.0\ 0.0]$, respectively. The true density and its

Table 3
Performance of the Parzen window estimate and the two sparse kernel density estimates in terms of L_1 test error and number of kernels required for the two-dimensional example of Gaussian and Laplacian mixture, quoted as mean ± standard deviation over 100 runs

Method	L_1 test error	kernel number
PW estimate	$(4.2453\pm0.8242)\times10^{-3}$	500 ± 0
Proposed SKD estimate	$(3.8379\pm0.7797)\times10^{-3}$	15.3 ± 3.9
SKD estimate of [6]	$(3.8281\pm0.8263)\times10^{-3}$	11.9 ± 2.6

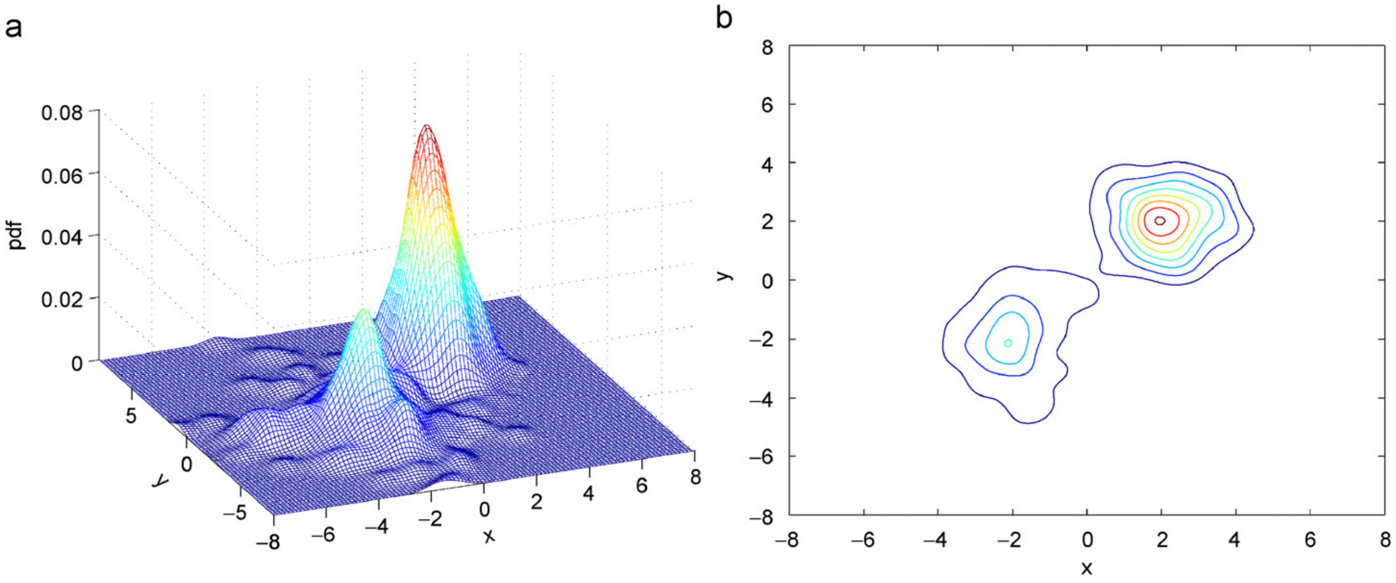


Fig. 4. A Parzen window estimate (a) and contour plot (b) for the two-dimensional example of Gaussian and Laplacian mixture.

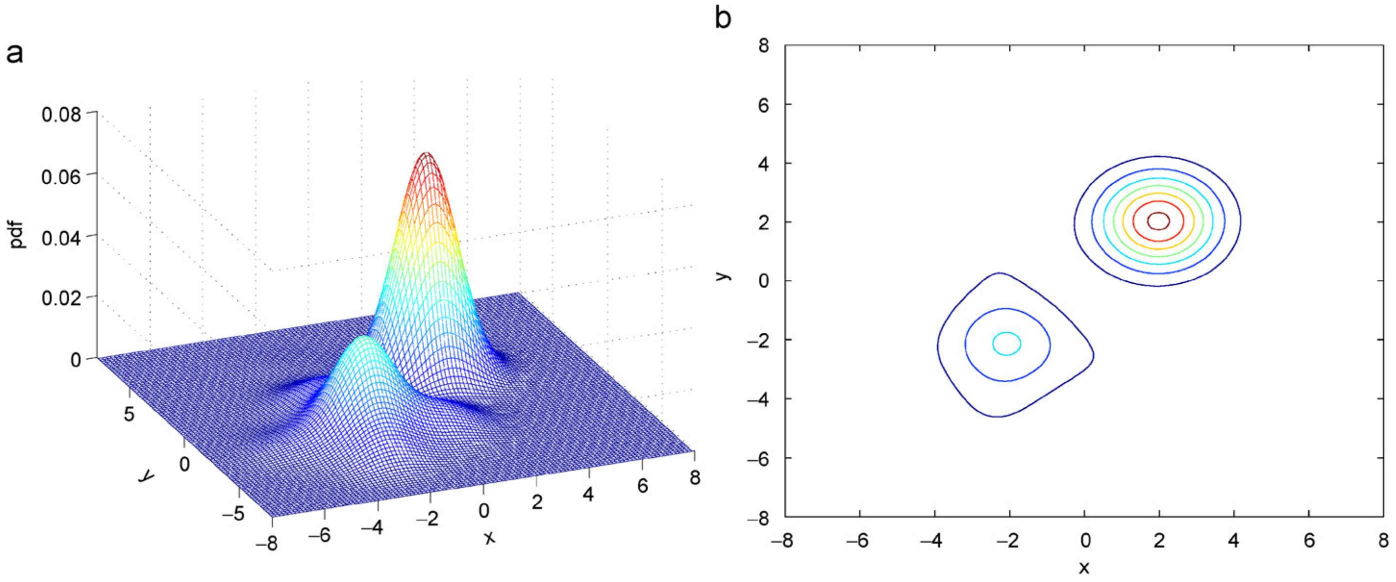


Fig. 5. A sparse kernel density estimate (a) and contour plot (b) for the two-dimensional example of Gaussian and Laplacian mixture.

contour plot are depicted in Fig. 6. The number of data points for density estimation was $N = 500$. The optimal kernel widths were found to be $\rho = 0.5$ and 1.1 for the PW estimate and the proposed SKD estimate, respectively. The experiment was repeated $N_{\text{run}} = 100$ times. Table 4 compares the L_1 test errors and the numbers of kernels required for the two density estimates. A typical PW estimate and a typical SKD estimate are shown in Figs. 7 and 8, respectively. Again, the two density estimates were seen to have comparable accuracies, but the proposed SKD estimation method achieved sparse estimates with an average number of required kernels less than 3% of the data samples. The maximum and minimum numbers of kernels over 100 runs were 22 and 8, respectively, for the SKD estimator.

Example 5. In this six-dimensional example, the underlying density to be estimated was given by

$$p(\mathbf{x}) = \frac{1}{3}\sum_{i=1}^{3}\frac{1}{(2\pi)^{6/2}}\frac{1}{\det^{1/2}|\boldsymbol{\Gamma}_i|}\mathrm{e}^{-1/2(\mathbf{x}-\boldsymbol{\mu}_i)^{\mathrm{T}}\boldsymbol{\Gamma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \tag{45}$$

with

$$\boldsymbol{\mu}_1 = [1.0\,1.0\,1.0\,1.0\,1.0\,1.0]^{\mathrm{T}},$$
$$\boldsymbol{\Gamma}_1 = \operatorname{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\}, \tag{46}$$

$$\boldsymbol{\mu}_2 = [-1.0\ \ -1.0\ \ -1.0 - 1.0\ \ -1.0\ \ -1.0]^{\mathrm{T}},$$
$$\boldsymbol{\Gamma}_2 = \operatorname{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}, \tag{47}$$

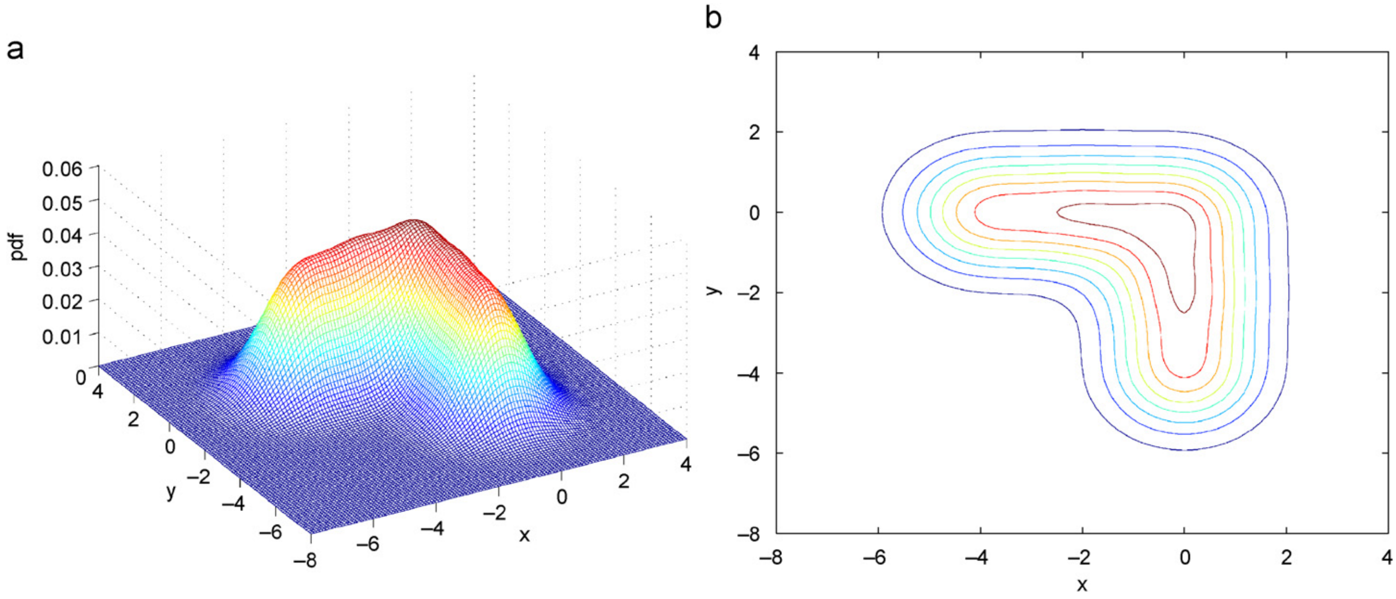


Fig. 6. True density (a) and contour plot (b) for the two-dimensional example of five Gaussian mixture.

Table 4
Performance of the Parzen window estimate and the sparse kernel density estimate in terms of L_1 test error and number of kernels required for the two-dimensional example of five Gaussian mixture, quoted as mean ± standard deviation over 100 runs

Method	L_1 test error	kernel number
PW estimate	$(3.6204 \pm 0.4394) \times 10^{-3}$	500 ± 0
Proposed SKD estimate	$(3.6100 \pm 0.5025) \times 10^{-3}$	13.2 ± 2.9

$$\boldsymbol{\mu}_3 = [0.0\,0.0\,0.0\,0.0\,0.0\,0.0]^{\mathrm{T}},$$
$$\boldsymbol{\Gamma}_3 = \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}. \tag{48}$$

The estimation data set contained $N = 600$ samples. The optimal kernel width was found to be $\rho = 0.65$ for the PW estimate and $\rho = 1.2$ for the SKD estimate, respectively, via cross validation. The experiment was repeated $N_{\mathrm{run}} = 100$ times. The results obtained by the two density estimator are summarised in Table 5. For this example, again, the two density estimates were seen to have comparable accuracies, but the proposed method achieved very sparse estimates with an average number of required kernels less than 2% of the data samples. The maximum and minimum numbers of kernels over 100 runs were 16 and 7, respectively, for the SKD estimator.

This example was used to test the SKD estimation method of [6] under the same experimental conditions. The results obtained by our previous SKD estimator, quoted from [6], are also given in Table 5 for comparison. It is seen from Table 5 that for this high-dimensional example the proposed SKD estimator outperformed our previous SKD estimator in terms of both the test performance and the level of sparsity.

Example 6. This was a two-class classification problem in a two-dimensional feature space [19]. The training set contained 250 samples with 125 points for each class, and the test set had 1000 points with 500 samples for each class. The optimal Bayes test error rate based on the true underlying probability distribution for this example is known to be 8%. We first estimated the two conditional density functions $\hat{p}(\mathbf{x};\boldsymbol{\beta},\rho|\mathrm{C}0)$ and $\hat{p}(\mathbf{x};\boldsymbol{\beta},\rho|\mathrm{C}1)$ from the training data, and then applied the Bayes decision rule

$$\left.\begin{array}{ll} \text{if } \hat{p}(\mathbf{x};\boldsymbol{\beta},\rho|\mathrm{C}0) \geqslant \hat{p}(\mathbf{x};\boldsymbol{\beta},\rho|\mathrm{C}1), & \mathbf{x} \text{ belongs to class } 0 \\ \qquad\qquad \text{else}, & \mathbf{x} \text{ belongs to class } 1 \end{array}\right\} \tag{49}$$

to the test data set and calculated the corresponding error rate. Table 6 lists the results obtained by the two kernel density estimates, the PW and the proposed SKD estimates, where the value of ρ was determined by minimising the test error rate. It can be seen that the proposed SKD estimation method yielded the sparse conditional density estimates and achieved the optimal Bayes classification performance. This clearly demonstrated the accuracy of the density estimates. Fig. 9(a) and (b) depicts the decision boundaries of the classifier (49) for the PW estimate and the proposed SKD estimate, respectively.

The results obtained for this example using our previous SKD estimator based on the empirical distribution function as the desired response, quoted from [6], are also summarised in Table 6. It can be seen from Table 6 that for this example the proposed SKD estimator was more accurate than the SKD estimator of [6] but the latter achieved a sparser density estimate.

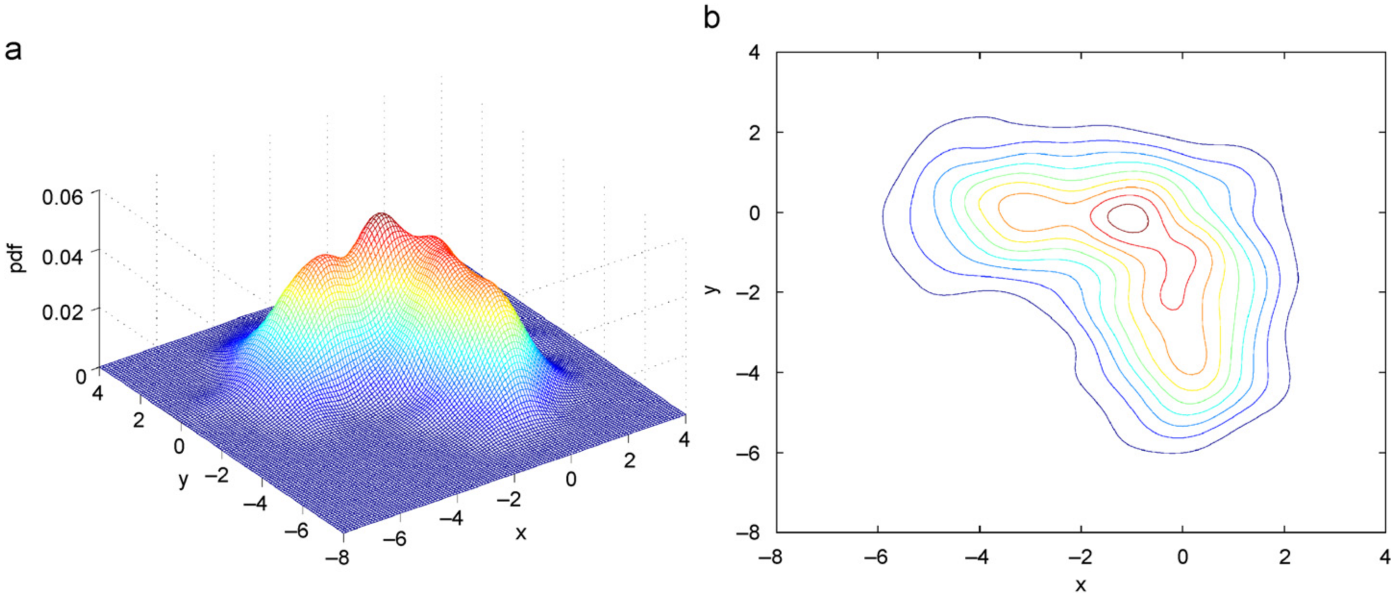


Fig. 7. A Parzen window estimate (a) and contour plot (b) for the two-dimensional example of five Gaussian mixture.

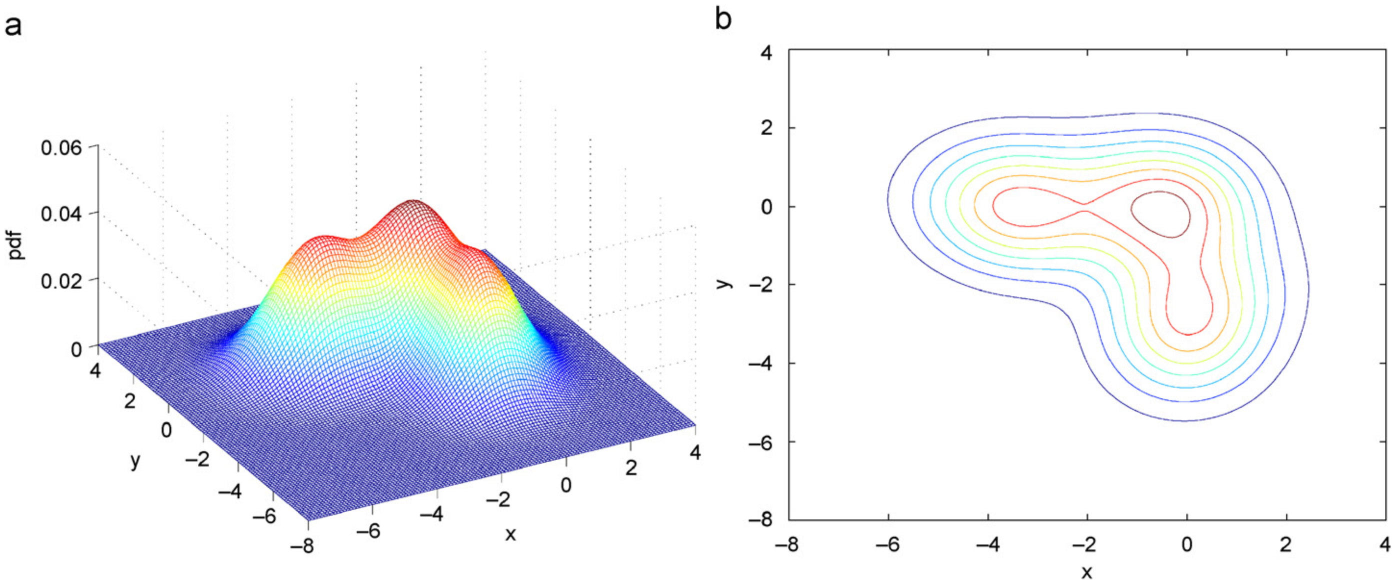


Fig. 8. A sparse kernel density estimate (a) and contour plot (b) for the two-dimensional example of five Gaussian mixture.

Table 5
Performance of the Parzen window estimate and the two sparse kernel density estimates in terms of L_1 test error and number of kernels required for the six-dimensional example of three Gaussian mixture, quoted as mean ± standard deviation over 100 runs

Method	L_1 test error	kernel number
PW estimate	$(3.5195 \pm 0.1616) \times 10^{-5}$	600 ± 0
Proposed SKD estimate	$(3.1134 \pm 0.5335) \times 10^{-5}$	9.4 ± 1.9
SKD estimate of [6]	$(4.4781 \pm 1.2292) \times 10^{-5}$	14.9 ± 2.1

5. Conclusions

A simple kernel density estimation method has been proposed based on a regression approach with the PW estimate as the desired response. The OFR algorithm has been employed to select SKD estimates, by incrementally minimising an LOO test score coupled with local regularisation to further enforce the sparseness of density estimates. The kernel weights of the final selected sparse model are computed using the MNQP algorithm to meet

Table 6
Performance of the Parzen window estimate and the two sparse kernel density estimates for the two-class two-dimensional classification example

Method	$\hat{p}(\bullet\vert C0)$	kernel width	$\hat{p}(\bullet\vert C1)$	kernel width	Test error rate (%)
PW estimate	125 kernels	0.24	125 kernels	0.23	8.0
Proposed SKD estimate	6 kernels	0.28	5 kernels	0.28	8.0
SKD estimate of [6]	5 kernels	0.20	4 kernels	0.20	8.3

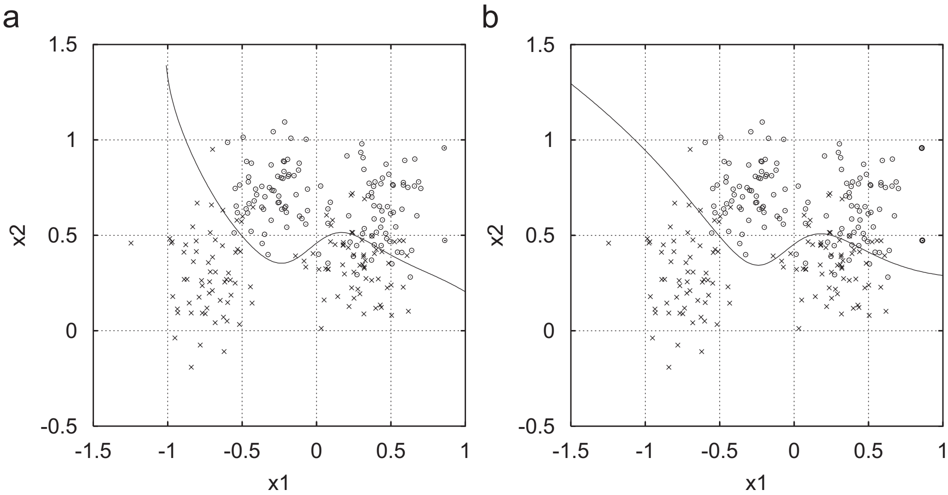


Fig. 9. (a) Decision boundary of the Parzen window estimate and (b) decision boundary of the sparse kernel density estimate for the two-class two-dimensional classification example, where circles represent the class-1 training data and crosses the class-0 training data.

the required nonnegative and unity constraints for probability density estimation. The MNQP algorithm also has a desired property of reducing the model size further. The proposed method is simple to implement, and except for the kernel width the algorithm contains no other free parameters that require tuning. The ability of the proposed method to construct a SKD estimate with a comparable accuracy to that of the full-sample optimised PW estimate has been demonstrated using several examples. The results obtained have shown that the proposed method offers a viable alternative for SKD estimation.

Appendix A. The OFR-LOO-LR algorithm

The modified Gram–Schmidt orthogonalisation procedure [4] calculates the **A** matrix row by row and orthogonalises $\mathbf{\Phi}$ as follows: at the lth stage make the columns $\boldsymbol{\phi}_j$, $l+1 \leqslant j \leqslant N$, orthogonal to the lth column and repeat the operation for $1 \leqslant l \leqslant N-1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)} = \boldsymbol{\phi}_j$, $1 \leqslant j \leqslant N$, then for $l = 1, 2, \ldots, N-1$,

$$\left.\begin{aligned} &\mathbf{w}_l = \boldsymbol{\phi}_l^{(l-1)}, \\ &a_{l,j} = \mathbf{w}_l^{\mathrm{T}} \boldsymbol{\phi}_j^{(l-1)} / (\mathbf{w}_l^{\mathrm{T}} \mathbf{w}_l), \quad l+1 \leqslant j \leqslant N, \\ &\boldsymbol{\phi}_j^{(l)} = \boldsymbol{\phi}_j^{(l-1)} - a_{l,j} \mathbf{w}_l, \quad\quad l+1 \leqslant j \leqslant N. \end{aligned}\right\} \tag{50}$$

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

$$\left.\begin{aligned} &g_l = \mathbf{w}_l^{\mathrm{T}} \mathbf{y}^{(l-1)} / (\mathbf{w}_l^{\mathrm{T}} \mathbf{w}_l + \lambda_l) \\ &\mathbf{y}^{(l)} = \mathbf{y}^{(l-1)} - g_l \mathbf{w}_l \end{aligned}\right\} 1 \leqslant l \leqslant N. \tag{51}$$

At the beginning of the lth stage of the OFR procedure, the $l-1$ regressors have been selected and the regression matrix can be expressed as

$$\mathbf{\Phi}^{(l-1)} = [\mathbf{w}_1 \cdots \mathbf{w}_{l-1} \; \boldsymbol{\phi}_l^{(l-1)} \cdots \boldsymbol{\phi}_N^{(l-1)}]. \tag{52}$$

Let a very small positive number T_z be given, which specifies the zero threshold and is used to automatically avoid any ill-conditioning or singular problem. With the initial conditions as specified in (28), the lth stage of the selection procedure is given as follows.

Step 1: For $l \leqslant j \leqslant N$:

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

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

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

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

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

$\mathcal{J}_l = \{l \leqslant j \leqslant N$ and j passes **Test**$\}$.

Step 2: Find

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

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

Step 3: The selection procedure is terminated with a $(l-1)$-term model, if $J_l \geqslant J_{l-1}$. Otherwise, perform the orthogonalisation as indicated in (50) to derive the lth row of $\mathbf{A}$ and to transform $\mathbf{\Phi}^{(l-1)}$ into $\mathbf{\Phi}^{(l)}$; calculate g_l and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (51); update the LOO error weightings

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

and go to Step 1.

References

[1] H. Akaike, A new look at the statistical model identification, IEEE Trans. Autom. Control AC-19 (1974) 716–723.

[2] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK, 1995.

[3] S. Chen, Local regularization assisted orthogonal least squares regression, Neurocomputing 69 (4–6) (2006) 559–585.

[4] S. Chen, S.A. Billings, W. Luo, Orthogonal least squares methods and their application to non-linear system identification, Int. J. Control 50 (5) (1989) 1873–1896.

[5] S. Chen, X. Hong, C.J. Harris, Sparse kernel regression modeling using combined locally regularized orthogonal least squares and *D*-optimality experimental design, IEEE Trans. Autom. Control 48 (6) (2003) 1029–1036.

[6] 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 Cybern. Part B 34 (4) (2004) 1708–1717.

[7] S. Chen, X. Hong, C.J. Harris, P.M. Sharkey, Sparse modeling using orthogonal forward regression with PRESS statistic and regularization, IEEE Trans. Syst. Man Cybern. Part B 34 (2) (2004) 898–911.

[8] A. Choudhury, Fast Machine Learning Algorithms for Large Data, PhD Thesis, Computational Engineering and Design Center, School of Engineering Sciences, University of Southampton, 2002.

[9] M. Girolami, C. He, Probability density estimation from optimally condensed data samples, IEEE Trans. Pattern Analy. Mach. Intell. 25 (10) (2003) 1253–1264.

[10] L.K. Hansen, J. Larsen, Linear unlearning for cross-validation, Adv. Comput. Math. 5 (1996) 269–280.

[11] M.H. Hansen, B. Yu, Model selection and the principle of minimum description length, J. Am. Statist. Assoc. 96 (454) (2001) 746–774.

[12] 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.

[13] D.J.C. MacKay, Bayesian interpolation, Neural Comput. 4 (3) (1992) 415–447.

[14] G. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.

[15] G. Monari, G. Dreyfus, Local overfitting control via leverages, Neural Comput. 14 (2002) 1481–1506.

[16] S. Mukherjee, V. Vapnik, Support vector method for multivariate density estimation, Technical Report, A.I. Memo No. 1653, MIT AI Lab, 1999.

[17] R.H. Myers, Classical and Modern Regression with Applications, Second Ed., PWS-KENT, Boston, 1990.

[18] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Statist. 33 (1962) 1066–1076.

[19] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, 1996.

[20] F. Sha, L.K. Saul, D.D. Lee, Multiplicative updates for nonnegative quadratic programming in support vector machines, Technical Report, MS-CIS-02-19, University of Pennsylvania, USA, 2002.

[21] B.W. Silverman, Density Estimation, Chapman Hall, London, 1996.

[22] M. Stone, Cross validation choice and assessment of statistical predictions, J. R. Statist. Soc. Ser. B 36 (1974) 111–147.

[23] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (2001) 211–244.

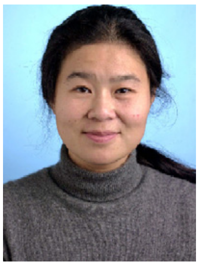
[24] V. Vapnik and S. Mukherjee, Support vector method for multivariate density estimation, in: S. Solla, T. Leen, K.R. Müller (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, MA, 2000, pp. 659–665.

[25] 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.

Sheng Chen received his PhD degree in control engineering from the City University, London, UK, in 1986. He was awarded the DSc by the University of Southampton, UK, in 2005.

He joined the School of Electronics and Computer Science, the University of Southampton, Southampton, in September 1999. He previously held research and academic appointments at the University of Sheffield, Sheffield, the University of Edinburgh, Edinburgh, and the University of Portsmouth, Portsmouth, all in UK. Professor Chen's research works include wireless communications, machine learning and neural networks, finite-precision digital controller design, and evolutionary computation methods. He has published over 270 research papers.

In the database of the world's most highly cited researchers, compiled by Institute for Scientific Information (ISI) of the USA, Dr. Chen is on the list of the highly cited researchers in the engineering category.



Xia Hong received her university education at National University of Defence Technology, P.R. China (BSc, 1984, MSc, 1987), and University of Sheffield, UK (PhD, 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 Department of Electronics and Computer Science at the University of Southampton from 1997 to 2001. She is currently a lecturer at the School of Systems Engineering, the University of Reading. 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 80 research papers, and co-authored a research book.

Dr. Hong was awarded a Donald Julius Groen Prize by IMechE in 1999.

Chris J. Harris received his PhD degree from the University of Southampton, Southampton, UK. He was also awarded the DSc by the University of Southampton.

He previously held appointments at the University of Hull, Hull, the UMIST, Manchester, the University of Oxford, Oxford, and the University of Cranfield, Cranfield, all in UK, as well as being employed by the UK. Ministry of Defence. He returned to the University of Southampton as the Lucas Professor of Aerospace Systems Engineering in 1987 to establish the Advanced Systems Research Group and, more recently, Image, Speech and Intelligent Systems Group. His research interests lie in the general area of intelligent and adaptive systems theory and its application to intelligent autonomous systems such as autonomous vehicles, management infrastructures such as command and control, intelligent control, and estimation of dynamic processes, multi-sensor data fusion, and systems integration. He has authored and co-authored 12 research books and over 400 research papers, and he is the associate editor of numerous international journals.

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