Neurocomputing 73 (2010) 727–739

Contents lists available at ScienceDirect

Neurocomputing

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

Regression based D-optimality experimental design 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

ARTICLE INFO

Article history:
Received 22 May 2009
Received in revised form
26 October 2009
Accepted 1 November 2009
Communicated by K. Li
Available online 18 November 2009

Keywords:
Probability density function
Parzen window estimate
Sparse kernel modelling
Orthogonal forward regression
Optimal experimental design
D-optimality

ABSTRACT

This paper derives an efficient algorithm for constructing sparse kernel density (SKD) estimates. The algorithm first selects a very small subset of significant kernels using an orthogonal forward regression (OFR) procedure based on the D-optimality experimental design criterion. The weights of the resulting sparse kernel model are then calculated using a modified multiplicative nonnegative quadratic programming algorithm. Unlike most of the SKD estimators, the proposed D-optimality regression approach is an unsupervised construction algorithm and it does not require an empirical desired response for the kernel selection task. The strength of the D-optimality OFR is owing to the fact that the algorithm automatically selects a small subset of the most significant kernels related to the largest eigenvalues of the kernel design matrix, which counts for the most energy of the kernel training data, and this also guarantees the most accurate kernel weight estimate. The proposed method is also computationally attractive, in comparison with many existing SKD construction algorithms. Extensive numerical investigation demonstrates the ability of this regression-based approach to efficiently construct a very sparse kernel density estimate with excellent test accuracy, and our results show that the proposed method compares favourably with other existing sparse methods, in terms of test accuracy, model sparsity and complexity, for constructing kernel density estimates.

1. Introduction

The problem of estimating probability density functions (PDFs) is of fundamental importance to all fields of engineering [1–6]. A powerful approach for density estimation is the finite mixture model (FMM) [7]. If the number of mixture components in the FMM is known, the problem is reduced to determine the FMM's parameters, and the maximum likelihood (ML) estimate of these parameters can be obtained using the expectation-maximisation (EM) algorithm [8]. The associated ML optimisation, in general, is a highly nonlinear optimisation process requiring extensive computation but for the Gaussian mixture model (GMM), the EM algorithm can be derived in an explicit and simple iterative form [9]. However, this ML estimation is well-known to be ill-posed and, in order to tackle the associated numerical difficulties, it is often required to apply resampling techniques such as the bootstrap [10,11] or other Bayesian methods [12,13]. In general, the correct number of mixture components is unknown, and simultaneously determining the required number of mixture components as well as estimating the associated parameters of the FMM is a challenging problem.

Alternatively, non-parametric techniques, which do not assume a particular functional form for PDF, are widely used in practical applications for density estimation. The classical Parzen window (PW) estimate [14], a well-known non-parametric density estimation technique, is remarkably simple and accurate. As the PW estimate, also known as the kernel density (KD) 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 sparse KD (SKD) estimation techniques. Various SKD estimation techniques can be divided into the two approaches.

The first class of SKD estimators starts with the full training data sample set as the kernel set and it then attempts to make as many kernel weights to near zero values as possible based on some chosen criteria. The corresponding kernels related to these very small kernel weights can then be removed from the kernel estimate, leading to a sparse representation. This class of SKD estimators include the support vector machine (SVM) based SKD estimation technique [15–17] and the related SKD estimator in reproducing kernel space [18] as well as the SKD estimation

* 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 © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2009.11.002

technique proposed in [19], which is known as the reduced set density estimator (RSDE). The RSDE [19] is a typical representative of this first class of SKD estimation technique, which is said to be based on minimisation of the integrated squared error (ISE) between the unknown underlying density and the KD estimate, calculated on the training set. A close examination of this training based ISE criterion reveals that it is equivalent to the training based ISE between the KD estimator and the PW estimator.

The second class of SKD estimation techniques by contrast selects a small subset of significant kernels based on various selection criteria. Subset kernel selection is typically carried out in an orthogonal forward regression (OFR) to achieve computational efficiency. A first regression-based SKD estimation method is reported in [20]. By converting the kernels into the associated cumulative distribution functions (CDFs) and using the empirical distribution function calculated on the training data set as the desired response, just like the SVM-based density estimation, this technique transfers the KD estimation into a regression problem and it selects SKD estimates based on an OFR algorithm that incrementally minimises the training mean square error (MSE). Motivated by our previous work on sparse regression modelling [21,22], a SKD construction algorithm is developed in [23] using the OFR based on the leave-one-out (LOO) test MSE and local regularisation (LR). This method is capable of constructing very sparse KD estimates with excellent generalisation capability. Moreover, the process is automatic and the user is not required to specify any additional criterion to terminate the density construction procedure.

The OFR-based SKD estimation methods of [20,23] carry out kernel selection on the associated CDF space, and they also adopt some *ad hoc* mechanisms to ensure the nonnegative and unity constraints for the kernel weights at the cost of increased computation in the model construction procedure. Recently, an interesting OFR-based SKD estimation alternative has been proposed [24]. Using the PW estimate as the desired response, this method performs SKD estimation directly in the PDF space and it automatically selects a SKD estimate using the OFR algorithm based on the LOO test MSE and LR. The nonnegative and unity constraints required for the kernel weights are met by updating the kernel weights of the selected SKD estimate using a modified multiplicative nonnegative quadratic programming (MNQP) algorithm of [25]. The MNQP algorithm has an additional desired property of further reducing the model size, yielding an even sparser density estimate. Extensive numerical results reported in [24] demonstrate that this SKD estimation method compares favourably with other existing SKD estimation methods, such as the SVM-based method [15–17] and the RSDE method [19] as well as the SKD construction methods of [20,23], in terms of model generalisation capability and model sparsity as well as model construction complexity. A computationally simpler method is also proposed for SKD estimation based on a forward constraint regression algorithm coupled with jackknife parameter estimator [26].

Optimal experimental designs [27] have been used for data analysis to construct smooth model response surface based on the setting of the experimental variables under well controlled experimental conditions. In optimal experimental design, model adequacy is evaluated by design criteria that are statistical measures of goodness of experimental designs by virtue of design efficiency and experimental effort. For regression models, quantitatively model adequacy is measured as a function of the eigenvalues of the design matrix, as it is known that the eigenvalues of the design matrix are linked to the covariance matrix of the least squares (LS) parameter estimate. There exist a variety of optimal experimental design criteria based on different aspects of experimental design [27], and the *D*-optimality criterion is most effective in optimising the parameter efficiency and model robustness via maximisation of the determinant of the design matrix. In regression application, optimal experimental designs have been adopted to construct sparse regression models based on an OFR procedure [21,28–31]. These previous works have demonstrated the effectiveness of optimal experimental design methods in obtaining a robust and parsimonious model structure with unbiased and accurate model parameter estimate.

Motivated by the success of applying optimal experimental designs in constructing robust and sparse regression models, we propose a simple yet effective regression-based method for SKD estimation using the *D*-optimality criterion. Our proposed method first selects a very small subset of significant kernels from the full kernel set generated from the training data set. Note that the problem of KD estimation is an unsupervised learning problem and typically an ill-conditioned one. Our proposed OFR procedure based on the *D*-optimality is a computationally efficient unsupervised learning method and, unlike many other existing SKD estimation methods, it does not require an empirical desired response for selecting kernels. The most significant advantages of the *D*-optimality based OFR are that the algorithm automatically identifies a small subset of the most significant kernels related to the largest eigenvalues of the kernel design matrix, which counts for the most energy of the kernel training data, and as a consequence this also guarantees the most accurate kernel weight estimation for the selected SKD estimate. No existing SKD estimator possesses these optimality properties. Therefore, this *D*-optimality based OFR is well-suited to the problem of KD estimation and it is capable of yielding robust and accurate as well as very sparse kernel model structure. After obtaining a very sparse kernel model structure, the associated kernel weights can readily be calculated using a modified version of the MNQP algorithm [25]. Because the size of the selected kernel model is extremely small, this MNQP algorithm requires little extra computational effort. Moreover, it can further set some kernel weights to near zero, yielding an even sparser KD estimate. This *D*-optimality based OFR algorithm has a lower computational complexity for density estimation than the existing SKD estimation methods [15–17,19,20,23,24]. Our experimental results also demonstrate that this new algorithm is capable of constructing much sparser KD estimates than the best existing SKD estimation methods, with equally accurate test performance.

2. Kernel density estimation as regression

Let a finite data sample set $\mathcal{D}_N = \{\mathbf{x}_k\}_{k=1}^N$ be drawn from a density $p(\mathbf{x})$, where $\mathbf{x} = [x_1 \ x_2 \ \cdots \ x_m]^T \in \mathcal{R}^m$ and the data sample $\mathbf{x}_k = [x_{1,k} \ x_{2,k} \ \cdots \ x_{m,k}]^T$. The non-parametric approach estimates the unknown density $p(\mathbf{x})$ using the KD estimate of the form

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

with the constraints

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

and

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

where $\boldsymbol{\beta}_N = [\beta_1 \ \beta_2 \ \cdots \ \beta_N]^T$ 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

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

However, any other kernel functions, satisfying

$$K_\rho(\mathbf{x},\mathbf{x}_k)\geq 0,\quad \forall \mathbf{x}\in\mathcal{R}^m, \tag{5}$$

$$\int_{\mathcal{R}^m} K_\rho(\mathbf{x},\mathbf{x}_k)\,d\mathbf{x}=1, \tag{6}$$

can also be used in the density estimate (1).

2.1. Parzen window estimate

The well-known PW estimate $\hat{p}_{\text{Par}}(\mathbf{x};\rho_{\text{Par}})$ is obtained by setting all the elements of $\boldsymbol{\beta}_N$ to $1/N$ in (1)

$$\hat{p}_{\text{Par}}(\mathbf{x};\rho_{\text{Par}})=\frac{1}{N}\sum_{k=1}^{N} K_{\rho_{\text{Par}}}(\mathbf{x},\mathbf{x}_k). \tag{7}$$

The kernel width ρ_{Par} of the PW estimate is typically determined via cross validation [32,33]. The PW estimate in fact can be derived as the ML estimator using the divergence-based criterion [7]. The negative cross-entropy or divergence between the true density $p(\mathbf{x})$ and the estimate $\hat{p}(\mathbf{x};\boldsymbol{\beta}_N,\rho)$, calculated on the training set, is defined as

$$\begin{aligned}\int_{\mathcal{R}^m} p(\mathbf{u})\log\hat{p}(\mathbf{u};\boldsymbol{\beta}_N,\rho)\,d\mathbf{u} &\approx \frac{1}{N}\sum_{k=1}^{N}\log\hat{p}(\mathbf{x}_k;\boldsymbol{\beta}_N,\rho)\\ &=\frac{1}{N}\sum_{k=1}^{N}\log\left(\sum_{n=1}^{N}\beta_n K_\rho(\mathbf{x}_k,\mathbf{x}_n)\right).\end{aligned} \tag{8}$$

Minimising this divergence subject to the constraints (2) and (3) leads to $\beta_n=1/N$ for $1\leq n\leq N$, i.e. the PW estimate. The PW estimate (7) is known to process a mean ISE convergence rate at order of N^{-1} [14] but it is nonsparse.

2.2. Existing sparse kernel density estimates

The density estimation problem (1) is an unsupervised learning problem. In most of the SKD estimation techniques [15–17,20,23], it is reformulated into a supervised regression problem by using the empirical distribution function as the desired response and converting the kernels into the associated CDFs. The true CDF of the PDF $p(\mathbf{x})$ is

$$F(\mathbf{x})=\int_{-\infty}^{\mathbf{x}} p(\mathbf{u})\,d\mathbf{u}, \tag{9}$$

and the CDF associated with the kernel $K_\rho(\mathbf{x},\mathbf{x}_k)$ is given by

$$q_\rho(\mathbf{x},\mathbf{x}_k)=\int_{-\infty}^{\mathbf{x}} K_\rho(\mathbf{u},\mathbf{x}_k)\,d\mathbf{u}. \tag{10}$$

Further define the empirical distribution function $\hat{F}(\mathbf{x};\mathcal{D}_N)$ on the training set $\mathcal{D}_N$ as

$$\hat{F}(\mathbf{x};\mathcal{D}_N)=\frac{1}{N}\sum_{k=1}^{N}\prod_{j=1}^{m}\theta(x_j-x_{j,k}), \tag{11}$$

with

$$\theta(x)=\begin{cases}1, & x>0,\\ 0, & x\leq 0,\end{cases} \tag{12}$$

where $\mathbf{x}_k\in\mathcal{D}_N$. Using $\hat{F}(\mathbf{x};\mathcal{D}_N)$ as the desired response for $F(\mathbf{x})$, the density estimation can be expressed as a regression modelling

$$\hat{F}(\mathbf{x};\mathcal{D}_N)=\sum_{k=1}^{N}\beta_k q_\rho(\mathbf{x},\mathbf{x}_k)+\hat{\varepsilon}(\mathbf{x}) \tag{13}$$

subject to the constraints (2) and (3), where $\hat{\varepsilon}(\mathbf{x})$ denotes the modelling error at $\mathbf{x}$. According to Glivenko-Cantelli theorem [34], the empirical distribution function (11) converges to the true CDF almost surely as the number of observations $N\to\infty$, under the assumption of independently identically distributed observations, which provides some theoretical justification for using (11) as the desired response of (9).

An alternative approach is proposed in [24] which directly performs a regression modelling in the PDF space by using the PW estimate (7) as the desired response of the true PDF $p(\mathbf{x})$. The PW estimate can be viewed as the "observation" of the true density contaminated by some "observation noise" $\hat{p}_{\text{Par}}(\mathbf{x};\rho_{\text{Par}})=p(\mathbf{x})+\tilde{\varepsilon}(\mathbf{x})$. Thus the KD estimation problem (1) can be viewed as the following regression problem with the PW estimate as the desired response

$$\hat{p}_{\text{Par}}(\mathbf{x};\rho_{\text{Par}})=\sum_{k=1}^{N}\beta_k K_\rho(\mathbf{x},\mathbf{x}_k)+\varepsilon(\mathbf{x}) \tag{14}$$

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

Define $\boldsymbol{\phi}(k)=[K_{k,1}\ K_{k,2}\ \cdots\ K_{k,N}]^T$ with $K_{k,i}=K_\rho(\mathbf{x}_k,\mathbf{x}_i)$, $y_k=\hat{p}_{\text{Par}}(\mathbf{x}_k;\rho_{\text{Par}})$, and $\varepsilon_k=\varepsilon(\mathbf{x}_k)$. Then the model (14) at the data point $\mathbf{x}_k\in\mathcal{D}_N$ is expressed as

$$y_k=\hat{y}_k+\varepsilon_k=\boldsymbol{\phi}^T(k)\boldsymbol{\beta}_N+\varepsilon_k. \tag{15}$$

The model (15) over the training data set $\mathcal{D}_N$ can be written in the matrix form

$$\mathbf{y}=\boldsymbol{\Phi}_N\boldsymbol{\beta}_N+\boldsymbol{\varepsilon} \tag{16}$$

with the following additional notations $\boldsymbol{\Phi}_N=[K_{i,k}]\in\mathcal{R}^{N\times N}$, $1\leq i,k\leq N$, $\boldsymbol{\varepsilon}=[\varepsilon_1\ \varepsilon_2\ \cdots\ \varepsilon_N]^T$, and $\mathbf{y}=[y_1\ y_2\ \cdots\ y_N]^T$. For convenience, we will denote the regression matrix $\boldsymbol{\Phi}_N=[\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2\ \cdots\ \boldsymbol{\phi}_N]$ with $\boldsymbol{\phi}_k=[K_{1,k}\ K_{2,k}\ \cdots\ K_{N,k}]^T$. Note that $\boldsymbol{\phi}_k$ is the k-th column of $\boldsymbol{\Phi}_N$, while $\boldsymbol{\phi}^T(k)$ is the k-th row of $\boldsymbol{\Phi}_N$.

The construction algorithm of [24] first selects a small subset of N_s significant kernels from the full kernel model (16) and then calculates the associated kernel weights using the MNQP algorithm. Experimental results presented in [24] demonstrate that this SKD estimator compares favourably with other existing SKD estimation methods [15–17,19,20,23], in terms of test accuracy and sparsity of constructed KD estimates. Therefore, we will use this SKD estimator as a benchmark for comparison with our proposed new method. Obviously, the SKD estimator of [24] is equally applicable when using (13) in the supervised subset kernel selection. A significant advantage of using (14) instead of (13) in the supervised subset kernel selection is a lower computational complexity, as it does not required to evaluate numerically the CDFs associated with the kernels based on (10).

A different SKD estimator that will be used as a benchmark for comparison with our proposed new method is the RSDE [19], which works on the full regression matrix $\boldsymbol{\Phi}_N$ and tries to make as many kernel weights to near zero as possible based on the empirical ISE criterion, thus yielding a sparse representation. Specifically, with the Gaussian kernel (4), the kernel weight vector of the RSDE estimator is obtained by solving the constrained nonnegative quadratic programming

$$\min_{\boldsymbol{\beta}_N}\{\tfrac{1}{2}\boldsymbol{\beta}_N^T\mathbf{G}_N\boldsymbol{\beta}_N-\hat{\mathbf{p}}_N^T\boldsymbol{\beta}_N\}$$

$$\text{s.t.}\quad \boldsymbol{\beta}_N^T\mathbf{1}_N=1\quad\text{and}\quad \beta_i\geq 0,\ 1\leq i\leq N, \tag{17}$$

where $\mathbf{G}_N=[g_{i,j}]\in\mathcal{R}^{N\times N}$ with

$$g_{i,j}=\int_{\mathcal{R}^m} G_\rho(\mathbf{x},\mathbf{x}_i)G_\rho(\mathbf{x},\mathbf{x}_j)\,d\mathbf{x}=G_{\sqrt{2}}\rho(\mathbf{x}_i,\mathbf{x}_j) \tag{18}$$

and

$$\hat{\mathbf{p}}_N=[\hat{p}_{\text{Par}}(\mathbf{x}_1;\rho)\ \hat{p}_{\text{Par}}(\mathbf{x}_2;\rho)\ \cdots\ \hat{p}_{\text{Par}}(\mathbf{x}_N;\rho)]^T, \tag{19}$$

i.e. the i-th element of $\hat{\mathbf{p}}_N$ is $\hat{p}_{\rm Par}(\mathbf{x}_i;\rho)$, the PW estimate at the data point $\mathbf{x}_i$ with the same kernel width ρ as the KD estimate to be determined. Note that the ISE between the unknown underlying density and the KD estimate, calculated on the training set, is equivalent to the ISE between the KD estimator and the PW estimator, as is illustrated below:

$$\min_{\boldsymbol{\beta}_N}\int_{\mathcal{R}^m}|\hat{p}_{\rm Par}(\mathbf{x};\rho_{\rm Par})-\hat{p}(\mathbf{x};\boldsymbol{\beta}_N,\rho)|^2\,d\mathbf{x} = \min_{\boldsymbol{\beta}_N}\int_{\mathcal{R}^m}\hat{p}^2(\mathbf{x};\boldsymbol{\beta}_N,\rho)\,d\mathbf{x}-2\sum_{i=1}^{N}\beta_i\mathcal{E}_{\hat{p}_{\rm Par}}[K_\rho(\mathbf{x},\mathbf{x}_i)], \tag{20}$$

where $\mathcal{E}_{\hat{p}_{\rm Par}}[\bullet]$ denotes the expectation with respect to $\hat{p}_{\rm Par}(\mathbf{x};\rho_{\rm Par})$. Given, $K_\rho(\bullet,\bullet)=G_\rho(\bullet,\bullet)$, the first term in the righthand side of (20) is the first term of the cost function in (17), while the second term in righthand side of (20) can be expressed as

$$\sum_{i=1}^{N}\beta_i\mathcal{E}_{\hat{p}_{\rm Par}}[K_\rho(\mathbf{x},\mathbf{x}_i)]\approx\sum_{i=1}^{N}\beta_i\frac{1}{N}\sum_{k=1}^{N}K_\rho(\mathbf{x}_k,\mathbf{x}_i)=\sum_{i=1}^{N}\beta_i\hat{p}_{\rm Par}(\mathbf{x}_i;\rho), \tag{21}$$

which is identical to the second term of the cost function in (17). In order to solve the constrained nonnegative quadratic programming (17), in particular to obtain a SKD estimate, the MNQP algorithm [25] can be used. However, because the full kernel matrix has a very high dimension of $N\times N$, the MNQP algorithm converges slowly. The RSDE [19] uses the alternative sequential minimal optimisation (SMO) [35] to solve (17). Note that the optimisation process can only drive many kernel weights to small values and, therefore, a zero threshold has to be specified to remove these weights. Appropriate zero threshold can only be determined empirically.

2.3. Gaussian mixture model estimate

As we will also use the GMM as a benchmark to compare with our new SKD estimator, this subsection briefly introduces the GMM. The general FMM is described by

$$\hat{p}_{\rm FMM}(\mathbf{x};\boldsymbol{\Omega})=\sum_{l=1}^{N_s}\beta_l K_{\boldsymbol{\Gamma}_l}(\mathbf{x},\mathbf{c}_l), \tag{22}$$

where N_s is the number of mixture components, and the kernel weights satisfy the constraints $\beta_l\ge 0$ for $1\le l\le N_s$ and $\sum_{l=1}^{N_s}\beta_l=1$. In this FMM, $\mathbf{c}_l=[c_{1,l}\ c_{2,l}\ \cdots\ c_{m,l}]^T$ denotes the l-th kernel centre vector, the l-th kernel's covariance matrix takes a diagonal form $\boldsymbol{\Gamma}_l=\mathrm{diag}\{\rho_{1,l}^2,\rho_{2,l}^2,\ldots,\rho_{m,l}^2\}$, and

$$\boldsymbol{\Omega}=\{\beta_l,\mathbf{c}_l,\boldsymbol{\Gamma}_l\}_{l=1}^{N_s} \tag{23}$$

denotes all the parameters of the FMM. When the Gaussian kernel function $K_{\boldsymbol{\Gamma}}(\mathbf{x},\mathbf{c})=G_{\boldsymbol{\Gamma}}(\mathbf{x},\mathbf{c})$, where

$$G_{\boldsymbol{\Gamma}}(\mathbf{x},\mathbf{c})=\frac{1}{(2\pi)^{m/2}\det^{1/2}[\boldsymbol{\Gamma}]}e^{-(1/2)(\mathbf{x}-\mathbf{c})^T\boldsymbol{\Gamma}^{-1}(\mathbf{x}-\mathbf{c})}, \tag{24}$$

is used, the FMM (22) is the GMM.

The EM algorithm for estimating the parameters of the GMM takes an explicit iterative form [9]. Given a value of $\boldsymbol{\Omega}$, labelled as $\boldsymbol{\Omega}^{\rm old}$, define

$$P(l|\mathbf{x}_k,\boldsymbol{\Omega}^{\rm old})=\frac{\beta_l^{\rm old}K_{\boldsymbol{\Gamma}_l^{\rm old}}(\mathbf{x}_k,\mathbf{c}_l^{\rm old})}{\sum_{i=1}^{N_s}\beta_i^{\rm old}K_{\boldsymbol{\Gamma}_i^{\rm old}}(\mathbf{x}_k,\mathbf{c}_i^{\rm old})} \tag{25}$$

for $1\le l\le N_s$ and $1\le k\le N$. Then a new value of $\boldsymbol{\Omega}$ is obtained according to [9]

$$\beta_l^{\rm new}=\frac{1}{N}\sum_{k=1}^{N}P(l|\mathbf{x}_k,\boldsymbol{\Omega}^{\rm old}), \tag{26}$$

$$\mathbf{c}_l^{\rm new}=\frac{\sum_{k=1}^{N}\mathbf{x}_kP(l|\mathbf{x}_k,\boldsymbol{\Omega}^{\rm old})}{\sum_{k=1}^{N}P(l|\mathbf{x}_k,\boldsymbol{\Omega}^{\rm old})}, \tag{27}$$

$$\boldsymbol{\Gamma}_l^{\rm new}=\sum_{k=1}^{N}P(l|\mathbf{x}_k,\boldsymbol{\Omega}^{\rm old})\mathrm{diag}\left\{(x_{1,k}-c_{1,l}^{\rm new})^2,\ldots,(x_{m,k}-c_{m,l}^{\rm new})^2\right\}/\sum_{k=1}^{N}P(l|\mathbf{x}_k,\boldsymbol{\Omega}^{\rm old}), \tag{28}$$

where $x_{i,k}-c_{i,l}^{\rm new}$ denotes the i-th element of $\mathbf{x}_k-\mathbf{c}_l^{\rm new}$.

This simple EM algorithm for the GMM, however, is generally ill-posed. In particular, the updating Eq. (28) may cause numerical problems, which leads to divergence. Often more complicated robust techniques such as the bootstrap [10,11] may need to be used to overcome numerical difficulties. The choice of the initial $\boldsymbol{\Omega}$ is also critical, as the algorithm can only converge to local minima, and whether or not the algorithm converges may depend on the initial parameter value. We find out in our previous experience [11] that it is necessary to impose a minimum bound, $\rho_{\rm min}^2$, for all the variances $\rho_{i,l}^2$, $1\le i\le m$ and $1\le l\le N_s$. During the iteration process, any $\rho_{i,l}^2$ goes below the value $\rho_{\rm min}^2$ is reset to this minimum value. This helps to alleviate numerical problem and improve the chance of convergence. Appropriate $\rho_{\rm min}^2$ is problem dependant and can only be found by experiment.

3. Proposed sparse density estimator

Our aim is to seek a sparse representation for $\hat{p}(\mathbf{x};\boldsymbol{\beta}_N,\rho)$ with most elements of $\boldsymbol{\beta}_N$ being zero and yet processing accurate test performance or generalisation capability. As mentioned in the Introduction section, two alternative methods can be adopted to achieve this objective. The first approach works on the full regression matrix $\boldsymbol{\Phi}_N$ and tries to make as many kernel weights to near zero as possible based on some appropriate criteria, thus yielding a sparse representation, as in [15–19]. The second approach adopts the efficient OFR procedure to select a small subset of significant kernels based on some relevant criteria, thus constructing a sparse kernel model, as in [20,23,24]. We adopt the second approach here. However, our subset kernel selection method is very different from any of the previous works.

3.1. Subset kernel selection using D-optimality criterion

Consider the model (16) in the generic data modelling context. In experimental design, the matrix $\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N$ is called the design matrix. The LS estimate of $\boldsymbol{\beta}_N$ is given by

$$\hat{\boldsymbol{\beta}}_N=(\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N)^{-1}\boldsymbol{\Phi}_N^T\mathbf{y}. \tag{29}$$

Under the assumption that (16) represents the true data generating process and $\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N$ is nonsingular, the estimate $\hat{\boldsymbol{\beta}}_N$ is unbiased and the covariance matrix of the estimate is determined by the design matrix, namely,

$$\begin{cases}E[\hat{\boldsymbol{\beta}}_N]=\boldsymbol{\beta}_N,\\ \mathrm{Cov}\,[\hat{\boldsymbol{\beta}}_N]\propto(\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N)^{-1}.\end{cases} \tag{30}$$

It is well known that the model based on LS estimate tends to be unsatisfactory for an ill-conditioned regression matrix, i.e. ill-conditioned design matrix. The condition number of the design matrix is given by

$$C=\frac{\max\{\lambda_i,1\le i\le N\}}{\min\{\lambda_i,1\le i\le N\}} \tag{31}$$

with λ_i, $1\le i\le N$, being the eigenvalues of $\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N$. Too large a condition number will result in unstable LS parameter estimate while a small C improves model robustness. The D-optimality design criterion [27] maximises the determinant of the design

matrix for the constructed model. More specifically, let $\boldsymbol{\Phi}_{N_s}$ be a column subset of $\boldsymbol{\Phi}_N$ representing a constructed N_s-term subset model. According to the *D*-optimality criterion, the selected subset model is the one that maximises $\det(\boldsymbol{\Phi}_{N_s}^T\boldsymbol{\Phi}_{N_s})$. This helps to prevent the selection of an oversized ill-posed model and the problem of high parameter estimate variances. Thus, the *D*-optimality design is aimed to optimise model efficiency and robustness of parameter estimate. Moreover, the design matrix does not depend on $\mathbf{y}$ explicitly. Hence, the *D*-optimality design is an unsupervised learning, making it particularly suitable for determining the structure of KD estimate, as the latter is also essentially an unsupervised learning problem.

Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}_N$ be $\boldsymbol{\Phi}_N=\mathbf{W}_N\mathbf{A}_N$, where

$$\mathbf{A}_N=\begin{bmatrix}1 & a_{1,2} & \cdots & a_{1,N}\\ 0 & 1 & \ddots & \vdots\\ \vdots & \ddots & \ddots & a_{N-1,N}\\ 0 & \cdots & 0 & 1\end{bmatrix} \tag{32}$$

and $\mathbf{W}_N=[\mathbf{w}_1\ \mathbf{w}_2\ \cdots\ \mathbf{w}_N]$ with orthogonal columns satisfying $\mathbf{w}_i^T\mathbf{w}_j=0$, if $i\neq j$. Similarly, the orthogonal matrix corresponding to $\boldsymbol{\Phi}_{N_s}$ is denoted as $\mathbf{W}_{N_s}$. It is straightforward to verify that maximising $\det(\boldsymbol{\Phi}_{N_s}^T\boldsymbol{\Phi}_{N_s})$ is identical to maximising $\det(\mathbf{W}_{N_s}^T\mathbf{W}_{N_s})$ or, equivalently, minimising $-\log\det(\mathbf{W}_{N_s}^T\mathbf{W}_{N_s})$. In fact,

$$\det(\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N)=\prod_{i=1}^{N}\lambda_i. \tag{33}$$

But

$$\det(\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N)=\det(\mathbf{A}_N^T)\det(\mathbf{W}_N^T\mathbf{W}_N)\det(\mathbf{A}_N)=\det(\mathbf{W}_N^T\mathbf{W}_N)=\prod_{i=1}^{N}\lambda_i. \tag{34}$$

We also have

$$-\log\det(\mathbf{W}_N^T\mathbf{W}_N)=\sum_{i=1}^{N}-\log(\mathbf{w}_i^T\mathbf{w}_i). \tag{35}$$

Denote the design matrix as $\mathbf{B}_N=\boldsymbol{\Phi}_N^T\boldsymbol{\Phi}_N=[b_{i,j}]\in\mathcal{R}^{N\times N}$. The fast algorithm for the modified Gram–Schmidt orthogonalisation procedure [36] can readily be used to orthogonalise $\mathbf{B}_N$ and to calculate the $\mathbf{A}_N$ matrix. For the notational convenience, we will use the same notation $\mathbf{B}_N=[b_{i,j}]$ to denote the design matrix after its first $n\times n$ block has been orthogonalised. We can now summarise the *D*-optimality based OFR procedure. The *n*-th stage of the selection procedure is given as follows.

***D*-optimality based OFR.** *Begin*: For $n\le j\le N$, calculate $J_n^{(j)}=-\log(b_{j,j})$ and find

$$J_n=J_n^{(j_n)}=\min\{J_n^{(j)},n\le j\le N\}.$$

- If
  $$J_n>\xi, \tag{36}$$
  where ξ is a threshold value that determines the size of the subset model, goto *Stop*.
- Otherwise, the j_n- th column of $\mathbf{B}_N$ is interchanged from the *n*-th row upwards with the *n*-th column of $\mathbf{B}_N$, and then the j_n- th row of $\mathbf{B}_N$ is interchanged from the *n*-th column upwards with the *n*-th row of $\mathbf{B}_N$.
  The j_n-th column of $\mathbf{A}_N$ is interchanged up to the $(n-1)$- th row with the *n*-th column of $\mathbf{A}_N$.
  This effectively selects the j_n- th candidate as the *n*-th regressor in the subset model.
- For $n+1\le j\le N$, compute $\alpha_{n,j}=b_{n,j}/b_{n,n}$, and for $n+1\le j\le N$ and $j\le l\le N$, compute
  $$\begin{cases} b_{j,l}=b_{j,l}-\alpha_{n,j}\alpha_{n,l}b_{n,n},\\ b_{l,j}=b_{j,l}.\end{cases}$$
  Set $n=n+1$ and go to *Begin*.

Stop: This selects $n-1$ most significant kernels according to the *D*-optimality criterion to form the selected subset model.

The desired threshold value ξ is problem dependent, and it can typically be determined by simply observing the values of $-\log(\mathbf{w}_i^T\mathbf{w}_i)=-\log(b_{i,i})$ for $i=1,2,\ldots,$ and terminating the selection when it is appropriate. Alternatively, one can simply set a maximum number N_s for the selected significant kernels, where $N_s\ll N$. It does not really matter if N_s is set to be larger than necessary, as the MNQP algorithm [25] used to compute the kernel weights will automatically make some of the kernel weights to near zero, and thus reduces the model size to an appropriate level. It can be seen that the computational complexity of this *D*-optimality based OFR algorithm is no more than $\mathcal{O}(N^2)$. In fact, it can easily be shown that the complexity of this *D*-optimality based OFR for subset kernel selection is lower than any of the existing SKD estimators [15–20,23,24].

Specifically, the computational complexity of the proposed *D*-optimality based SKD algorithm can be expressed by

$$C_{\text{prop.SKD}}=N_s\cdot\tau_{\text{prop.SKD}}\cdot N^2,$$

where N_s is the number of kernels selected and $\tau_{\text{prop.SKD}}$ is a scaling factor. Similarly, the complexity of the previous SKD algorithm [24] can be expressed by

$$C_{\text{prev.SKD}}=N_s'\cdot\tau_{\text{prev.SKD}}\cdot N^2,$$

with N_s' denoting the number of selected kernels and $\tau_{\text{prev.SKD}}$ the related scaling factor, while the complexity of the RSDE algorithm [19] can be written as

$$C_{\text{RSDE}}=N_s''\cdot\tau_{\text{RSDE}}\cdot N^2,$$

with N_s'' denoting the number of selected kernels and τ_{RSDE} the corresponding scaling factor. It can easily be shown that $\tau_{\text{prop.SKD}}$ is much smaller than $\tau_{\text{prev.SKD}}$ and τ_{RSDE}. Furthermore, the proposed *D*-optimality based SKD algorithm typically yields sparser PDF estimates than the previous SKD algorithm [24] and the RSDE [19], as will be confirmed in the simulation study. Thus, N_s is smaller than N_s' and N_s''. Therefore, the proposed method is computationally simpler than the previous methods of [19,24].

The unsupervised *D*-optimality based OFR possesses two remarkable optimality properties for SKD construction. The "evidence" of the unknown underlying density distribution is given in the data sample set $\mathcal{D}_N$, i.e. in the full kernel matrix $\boldsymbol{\Phi}_N$. The *D*-optimality based OFR algorithm automatically identifies a small subset of the N_s most significant kernels related to the largest eigenvalues of $\boldsymbol{\Phi}_N$, which counts for the most energy of the kernel training data. This is similar to kernel principal component analysis (KPCA) which constructs the N_s eigenvector bases that counts for the most energy of the full kernel matrix. However, in a conventional KPCA, each constructed orthogonal base is a linear combination of all the original regressors and, therefore, it does not provide a sparse representation with respect to the given training data set $\mathcal{D}_N$. This first optimality property is not guaranteed in any of the existing SKD estimators [15–20,23,24]. As a consequence of this "optimal sparse property", we will demonstrate later in the numerical experiment that the *D*-optimality based SKD estimator is capable of producing sparser KD estimates, compared with some existing benchmark SKD estimation techniques. As a direct result of this first optimality

property, the subsequent kernel weight vector estimate has the minimum estimation variance, i.e. the most accurate estimate, among all the N_s−term subset models of the full kernel matrix $\boldsymbol{\Phi}_N$. Note that, unlike regularisation aided techniques which sacrifice the bias in parameter estimate for the reduction in estimation variance, the *D*-optimality criterion does not sacrifice the estimation bias in order to reduce the estimation variance.

3.2. Calculating kernel weights

After the structure determination using the *D*-optimality based OFR, we obtain a N_s- term subset kernel model, where $N_s \ll N$. The resulting regression modelling problem is re-written in the following:

$$\mathbf{y} = \boldsymbol{\Phi}_{N_s}\boldsymbol{\beta}_{N_s} + \boldsymbol{\varepsilon} \tag{37}$$

subject to the constraints

$$\boldsymbol{\beta}_{N_s}^T \mathbf{1}_{N_s} = 1 \quad \text{and} \quad \beta_i \geq 0, \ \ 1 \leq i \leq N_s, \tag{38}$$

where $\boldsymbol{\beta}_{N_s}^T = [\beta_1 \ \ \beta_2 \ \ \cdots \ \ \beta_{N_s}]$. The kernel weight vector can be obtained by solving the following constrained nonnegative quadratic programming

$$\min_{\boldsymbol{\beta}_{N_s}} \{\tfrac{1}{2}\boldsymbol{\beta}_{N_s}^T \mathbf{B}_{N_s}\boldsymbol{\beta}_{N_s} - \mathbf{v}_{N_s}^T\boldsymbol{\beta}_{N_s}\}$$

$$\text{s.t.} \quad \boldsymbol{\beta}_{N_s}^T \mathbf{1}_{N_s} = 1 \quad \text{and} \quad \beta_i \geq 0, \ \ 1 \leq i \leq N_s, \tag{39}$$

where $\mathbf{B}_{N_s} = \boldsymbol{\Phi}_{N_s}^T\boldsymbol{\Phi}_{N_s} = [b_{i,j}] \in \mathcal{R}^{N_s \times N_s}$ and $\mathbf{v}_{N_s} = \boldsymbol{\Phi}_{N_s}^T\mathbf{y} = [v_1 \ \ v_2 \ \ \cdots \ \ v_{N_s}]^T$. This constrained optimisation can of course be solved using the SMO [35]. Because the subset kernel matrix size $N_s \times N_s$ is so small, we find this optimisation problem can be solved efficiently using a modified version of the MNQP algorithm [25].

Since the elements of $\mathbf{B}_{N_s}$ and $\mathbf{v}_{N_s}$ are strictly positive, the auxiliary function [25] for the above problem is given by

$$\frac{1}{2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} b_{i,j}\frac{\beta_j^{(t)}(\beta_i^{(t+1)})^2}{\beta_i^{(t)}} - \sum_{i=1}^{N_s} v_i\beta_i^{(t+1)}, \tag{40}$$

and the Lagrangian associated with this auxiliary problem can be formed as [19]

$$\mathcal{L} = \frac{1}{2}\sum_{i=1}^{N_s}\sum_{j=1}^{N_s} b_{i,j}\frac{\beta_j^{(t)}(\beta_i^{(t+1)})^2}{\beta_i^{(t)}} - \sum_{i=1}^{N_s} v_i\beta_i^{(t+1)} - \eta^{(t)}\left(\sum_{i=1}^{N_s}\beta_i^{(t+1)} - 1\right), \tag{41}$$

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

$$\frac{\partial\mathcal{L}}{\partial\beta_i^{(t+1)}} = 0 \quad \text{and} \quad \frac{\partial\mathcal{L}}{\partial\eta^{(t)}} = 0 \tag{42}$$

leads to the following updating equations:

$$c_i^{(t)} = \beta_i^{(t)}\left(\sum_{j=1}^{N_s} b_{i,j}\beta_j^{(t)}\right)^{-1}, \quad 1 \leq i \leq N_s, \tag{43}$$

$$\eta^{(t)} = \left(\sum_{i=1}^{N_s} c_i^{(t)}\right)^{-1}\left(1 - \sum_{i=1}^{N_s} c_i^{(t)}v_i\right), \tag{44}$$

$$\beta_i^{(t+1)} = c_i^{(t)}(v_i + \eta^{(t)}). \tag{45}$$

It is easy to check that, if $\boldsymbol{\beta}_{N_s}^{(t)}$ meets the constraints (38), $\boldsymbol{\beta}_{N_s}^{(t+1)}$ updated according to (43)–(45) also satisfies (38). The initial condition can thus be set as $\beta_i^{(0)} = 1/N_s$, $1 \leq i \leq N_s$.

During the iterative procedure, some of the kernel weights may be driven to near zero, particularly when the subset model size N_s is chosen to be larger than really necessary. The corresponding kernels can then be removed from the kernel model, leading to a reduction in the subset model size. It is due to this desired property that the setting of the maximum selected subset model size is not too critical in the *D*-optimality based OFR. Because N_s is typically very small, this MNQP algorithm imposes only a very small extra amount of computational. Thus, the overall complexity of the proposed method is still no more than $\mathcal{O}(N^2)$.

4. Numerical experiments

Several examples were used in the simulation to test the proposed SKD estimator using the *D*-optimality based OFR with the MNQP updating and to compare its performance with the PW estimator, the previous SKD estimator [24], the RSDE estimator [19] and the GMM estimator. Majority of the cases were the density estimation problems. In each of these cases, a data set of N randomly drawn samples was used to construct KD estimates, and a separate test data set of $N_{\text{test}} = 10,000$ samples was used to calculate the L_1 test error for the resulting estimate according to

$$L_1 = \frac{1}{N_{\text{test}}}\sum_{k=1}^{N_{\text{test}}} |p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k; \boldsymbol{\beta}_{N_s}, \rho)|, \tag{46}$$

with N_s denoting the number of kernels in the estimate. The experiment was repeated by N_{run} different random runs for each example. Two of the examples were two-class classification problems.

The Kullback–Leibler divergence (KLD) is a measure of the difference between the two probability distributions, $p(\mathbf{x})$ and $\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{N_s}, \rho)$, and is defined by

$$D_{\text{KL}}(p|\hat{p}) = \int_{\mathcal{R}^m} p(\mathbf{x})\log\frac{p(\mathbf{x})}{\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{N_s}, \rho)}\, d\mathbf{x}. \tag{47}$$

For the one-dimensional and two-dimensional problems, the KLD was also used to test the resulting estimates. For a one-dimensional problem, the KLD can be approximated accurately by partitioning the integration range $[x_{\min}, x_{\max}]$ into the N_{p} small equal-length intervals and computing the summation

$$D_{\text{KL}}(p|\hat{p}) \approx \sum_{k=1}^{N_{\text{p}}} p(k)\log\frac{p(k)}{\hat{p}(k)}\Delta x, \tag{48}$$

where $\Delta x = (x_{\max} - x_{\min})/N_{\text{p}}$, $p(k) = p(x_{\min} + k\Delta x)$ and $\hat{p}(k) = \hat{p}(x_{\min} + k\Delta x; \boldsymbol{\beta}_{N_s}, \rho)$. In the experiment, we chose $N_{\text{p}} \geq 10,000$ to ensure the accuracy of the approximation. Similarly, for a two-dimensional problem, the KLD is approximated by partitioning the integration range $[x_{1,\min}, x_{1,\max}] \times [x_{2,\min}, x_{2,\max}]$ into the $N_{\text{p}} \times N_{\text{p}}$ small equal-area intervals and calculated the double summation

$$D_{\text{KL}}(p|\hat{p}) \approx \sum_{k=1}^{N_{\text{p}}}\sum_{l=1}^{N_{\text{p}}} p(k,l)\log\frac{p(k,l)}{\hat{p}(k,l)}(\Delta x)^2, \tag{49}$$

where $\Delta x = (x_{1,\max} - x_{1,\min})/N_{\text{p}} = (x_{2,\max} - x_{2,\min})/N_{\text{p}}$, $p(k,l) = p(x_{1,\min} + k\Delta x, x_{2,\min} + l\Delta x)$ and $\hat{p}(k,l) = \hat{p}(x_{1,\min} + k\Delta x, x_{2,\min} + l\Delta x; \boldsymbol{\beta}_{N_s}, \rho)$. To ensure the accuracy of the approximation, we chose $N_{\text{p}} > 100$. For higher-dimensional problems, calculation of the KLD becomes computationally too expensive.

The Gaussian kernel function was employed. The value of kernel width ρ used for a KD estimator was determined via cross validation. For the GMM, instead of exhaustedly trying different values for the number of mixing components, N_s, based on cross validation, we determined the number of mixing components for the GMM according to the average model size obtained for the proposed SKD estimate. For the EM algorithm, all the initial mixing weights β_l were set to $1.0/N_s$, the initial centre vectors $\mathbf{c}_l$

Table 1
Performance comparison of the PW estimator, previous SKD estimator [24], proposed SKD estimator, RSDE estimator [19] and GMM estimator for the one-dimensional example of eight-Gaussian mixture over 200 runs.

Estimator	PW	Previous SKD [24]	Proposed SKD	RSDE [19]	GMM
Kernel type	Fixed, $\rho_{\text{Par}} = 0.17$	Fixed, $\rho = 0.30$	Fixed, $\rho = 0.31$	Fixed, $\rho = 0.56$	Tunable
L_1 test error $\times 10^2$	4.119 ± 1.351	4.189 ± 1.346	4.091 ± 1.392	5.816 ± 0.836	5.229 ± 1.574
KLD $\times 10^2$	4.478 ± 2.774	8.211 ± 11.28	6.875 ± 5.409	6.956 ± 2.522	7.022 ± 4.590
Kernel no.	200	10.2 ± 1.7	8.7 ± 0.9	14.0 ± 4.3	8
Maximum	200	15	11	32	8
Minimum	200	5	6	6	8

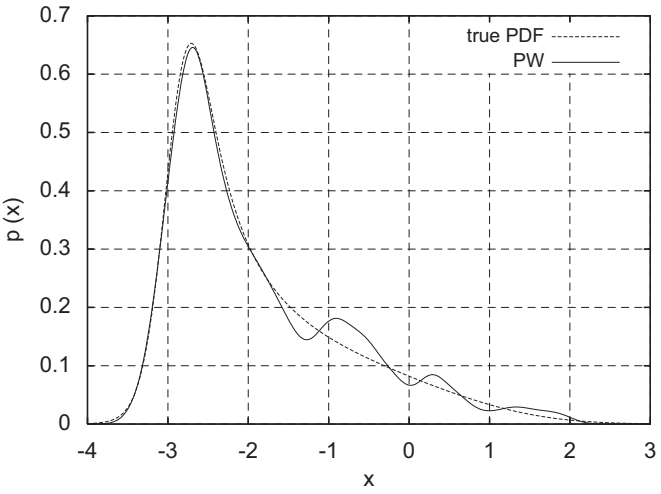


Fig. 1. A PW estimate (solid) in comparison with the true density (dashed) for the one-dimensional example of eight-Gaussian mixture.

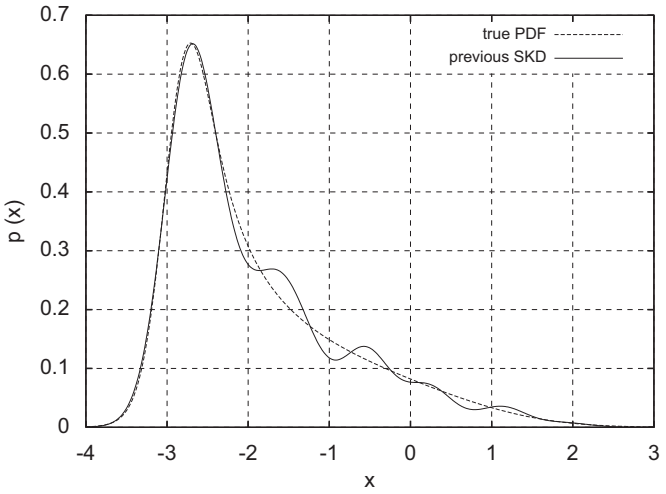


Fig. 2. A previous SKD estimate [24] (solid) in comparison with the true density (dashed) for the one-dimensional example of eight-Gaussian mixture.

were randomly chosen from the region $[a,b]^m \in \mathcal{R}^m$, and all the initial variances $\rho_{i,l}^2$ were set to the same value ρ_{ini}^2. A minimum bound, ρ_{min}^2, for the variances was also assigned. If some runs of the EM algorithm were observed to diverge, the region $[a,b]^m$, the values of ρ_{ini}^2 and/or ρ_{min}^2 were re-chosen until all the N_{run} of the EM algorithm were converged.

4.1. One-dimensional examples

Example 1. The density to be estimated was the mixture of eight Gaussian distributions given by

$$p(x) = \frac{1}{8}\sum_{i=0}^{7} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-(x-\mu_i)^2/2\sigma_i^2} \tag{50}$$

with

$$\sigma_i = \sqrt{\left(\frac{2}{3}\right)^i}, \quad \mu_i = 3\left(\left(\frac{2}{3}\right)^i - 1\right), \quad 0 \le i \le 7. \tag{51}$$

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 $\rho_{\text{Par}} = 0.17$, $\rho = 0.30$, $\rho = 0.31$ and $\rho = 0.56$ empirically for the PW estimator, the previous SKD estimator [24], the proposed SKD estimator and the RSDE estimator [19], respectively.

We observed that the significant kernels according to the D-optimality criterion were in the range of 8–10 and the threshold value could be set to $\xi = -1.0$. However, we simply set the maximum number of selected kernels by the D-optimality based OFR to be $N_s = 16$. The maximum and minimum values of nonzero

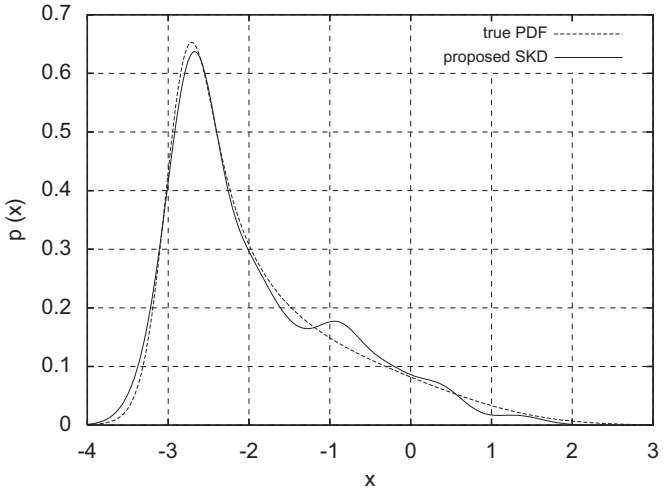


Fig. 3. A proposed SKD estimate (solid) in comparison with the true density (dashed), for the one-dimensional example of eight-Gaussian mixture.

kernel weights obtained by the MNQP algorithm over the 200 runs were 11 and 6, respectively, and the average model size for the proposed SKD estimator was $N_s = 8.7$. We used $N_s = 8$ for the GMM. After considerable experiments, all the $N_{\text{run}} = 200$ runs of the EM algorithm converged with the initialisation $[a,b] = [-4,3]$, $\rho_{\text{ini}}^2 = 0.1$ and $\rho_{\text{min}}^2 = 0.01$. Table 1 compares the performance of the five density estimates, where it can be seen that the proposed SKD estimator yielded sparser estimates with better test accuracy

than our previous SKD estimator [24] as well as the RSDE estimator [19]. Figs. 1–5 depict the five density estimates obtained in a typical experimental run.

Example 2. The density to be estimated was the mixture of Gaussian and Laplacian defined by

$$p(x)=\frac{1}{2\sqrt{2\pi}}e^{-(x-2)^2/2}+\frac{0.7}{4}e^{-0.7|x+2|}. \tag{52}$$

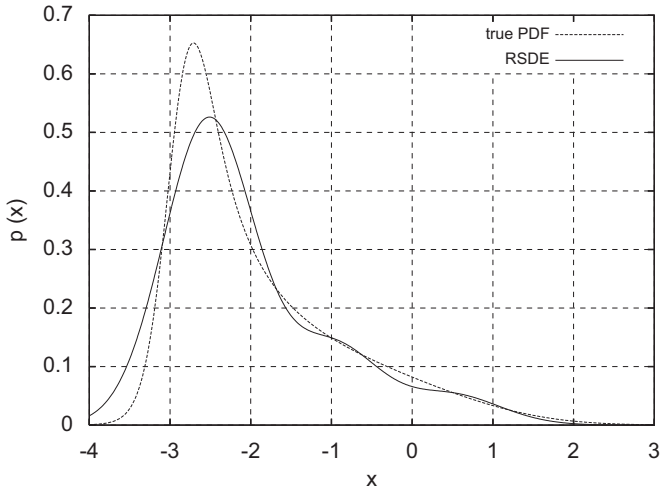


Fig. 4. A RSDE estimate [19] (solid) in comparison with the true density (dashed) for the one-dimensional example of eight-Gaussian mixture.

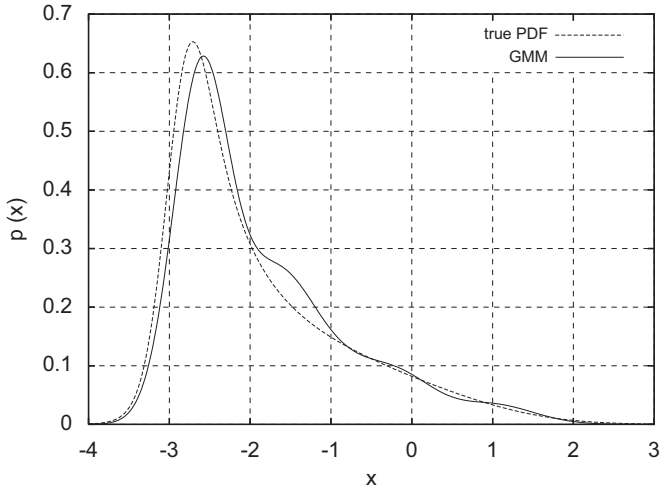


Fig. 5. A GMM estimate (solid) in comparison with the true density (dashed) for the one-dimensional example of eight-Gaussian mixture.

The number of data points for density estimation was $N=100$. The optimal kernel widths were found to be $\rho_{\text{Par}}=0.54$ for the PW estimator, $\rho=1.1$ for our previous SKD estimator [24] as well as the proposed SKD estimator, and $\rho=1.2$ for the RSDE estimator [19], respectively. The experiment was repeated $N_{\text{run}}=1000$ times.

According to the *D*-optimality criterion, only three kernels were significant and the threshold value could be set to $\xi=0.0$. But we simply set the maximum number of selected kernels by the *D*-optimality based OFR to be $N_s=10$ and let the MNQP algorithm to further reduce the model size. The maximum and minimum numbers of nonzero kernel weights determined over the 1000 runs were 5 and 2, respectively, and the average model size was $N_s=4.5$. We chose $N_s=5$ for the GMM, while appropriate initialisation was found to be $[a,b]=[-12,7]$, $\rho^2_{\text{ini}}=0.1$ and $\rho^2_{\text{min}}=0.01$, which ensured the convergence for all the $N_{\text{run}}=1000$ runs. Table 2 compares the performance of the five density estimators, while Figs. 6–10 plot the five density estimates obtained in a typical run, in comparison with the true density. For this example, the RSDE estimator achieved better test performance than the proposed *D*-optimality based SKD estimator but the latter arrived at a much sparser solution.

4.2. Two-dimensional examples

Example 3. The density to be estimated was defined by the mixture of Gaussian and Laplacian given as follows:

$$p(x,y)=\frac{1}{4\pi}e^{-(x-2)^2/2}e^{-(y-2)^2/2}+\frac{0.35}{8}e^{-0.7|x+2|}e^{-0.5|y+2|}. \tag{53}$$

The estimation data set contained $N=500$ samples, and the empirically found optimal kernel widths were $\rho_{\text{Par}}=0.42$ for the PW estimator, $\rho=1.1$ for our previous as well as proposed SKD estimators, and $\rho=1.2$ for the RSDE estimator [19], respectively. The experiment was repeated $N_{\text{run}}=100$ times.

We simply set the maximum selected kernels by the *D*-optimality based OFR to be $N_s=16$, and let the MNQP algorithm to determine the final model size. The maximum and minimum numbers of nonzero kernel weights turned out to be 14 and 5, respectively, over the 100 runs, while the average model size was $N_s=11.0$. For the GMM, the number of mixture components was chosen as $N_s=11$. After serval tries, an appropriate initialisation was found to be $[a,b]^2=[-8,8]^2$, $\rho^2_{\text{ini}}=0.4$ and $\rho^2_{\text{min}}=0.01$ for the EM algorithm to converge in all the $N_{\text{run}}=100$ runs. Table 3 lists the performance of the five density estimators, where it can be seen that the proposed *D*-optimality based SKD estimator obtained the sparsest density estimate with similarly good test performance in comparison with the other three benchmark KD density estimators.

Table 2
Performance comparison of the PW estimator, previous SKD estimator [24], proposed SKD estimator, RSDE estimator [19] and GMM estimator for the one-dimensional example of Gaussian and Laplacian mixture over 1000 runs.

Estimator	PW	Previous SKD [24]	Proposed SKD	RSDE [19]	GMM
Kernel type	Fixed, $\rho_{\text{Par}}=0.54$	Fixed, $\rho=1.1$	Fixed, $\rho=1.1$	Fixed, $\rho=1.2$	Tunable
L_1 test error $\times 10^2$	2.011 ± 0.621	2.011 ± 0.649	1.945 ± 0.644	1.886 ± 0.586	2.511 ± 0.904
KLD $\times 10^2$	8.090 ± 5.198	8.657 ± 5.122	8.309 ± 3.931	5.600 ± 3.771	12.08 ± 7.885
Kernel no.	100	5.2 ± 1.2	4.5 ± 0.8	9.7 ± 4.6	5
Maximum	100	10	5	44	5
Minimum	100	2	2	2	5

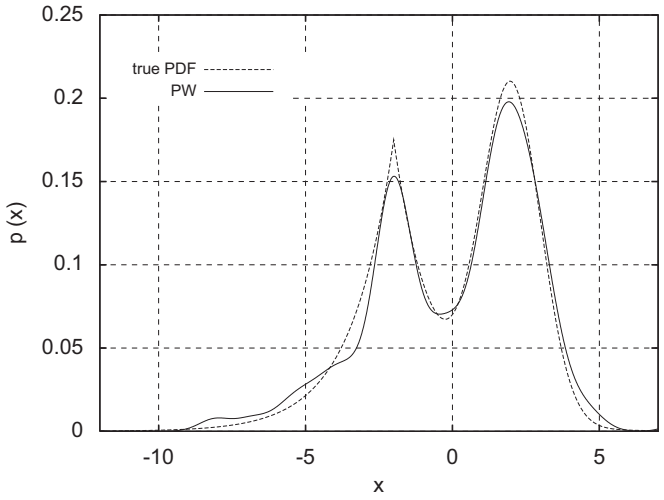


Fig. 6. A PW estimate (solid) in comparison with the true density (dashed) for the one-dimensional example of Gaussian and Laplacian mixture.

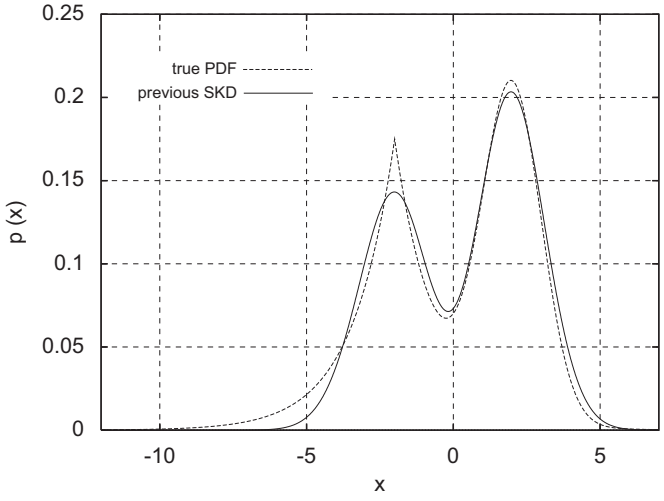


Fig. 7. A previous SKD estimate [24] (solid) in comparison with the true density (dashed) for the one-dimensional example of Gaussian and Laplacian mixture.

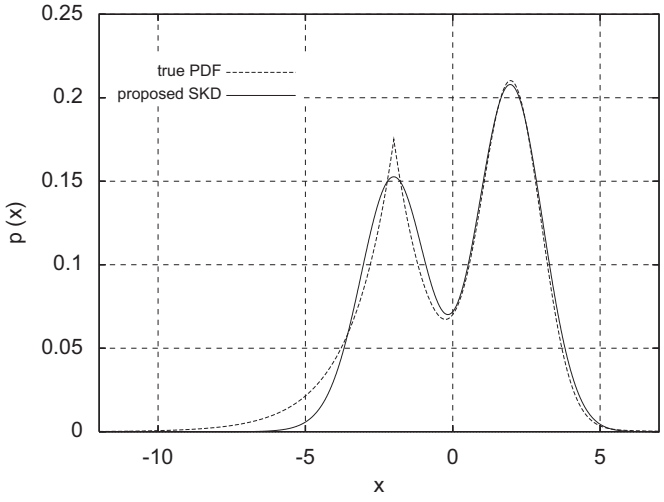


Fig. 8. A proposed SKD estimate (solid) in comparison with the true density (dashed) for the one-dimensional example of Gaussian and Laplacian mixture.

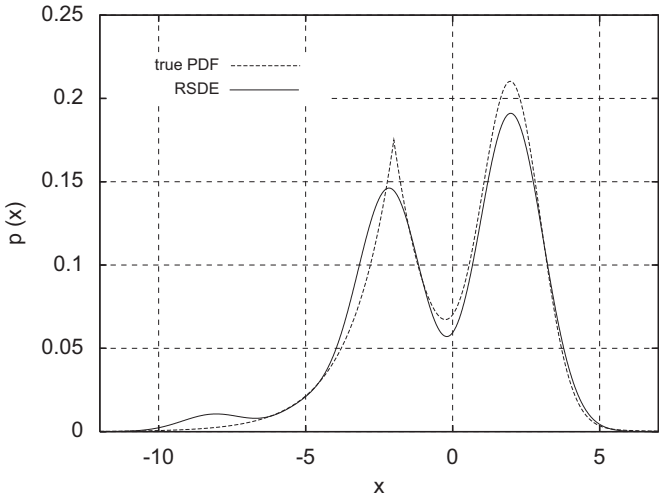


Fig. 9. A RSDE estimate [19] (solid) in comparison with the true density (dashed) for the one-dimensional example of Gaussian and Laplacian mixture.

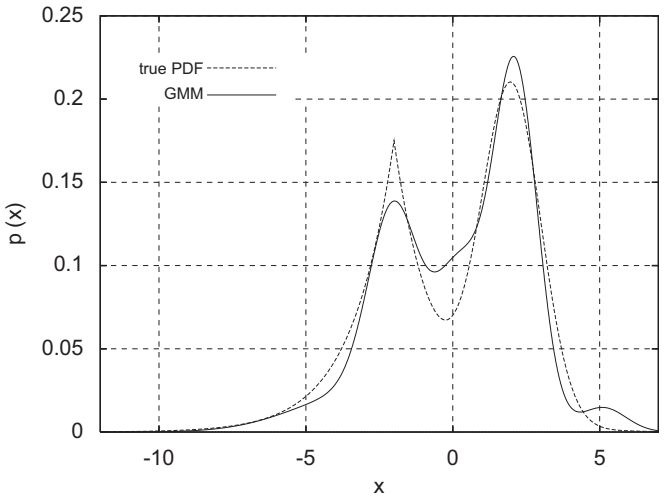


Fig. 10. A GMM estimate (solid) in comparison with the true density (dashed) for the one-dimensional example of Gaussian and Laplacian mixture.

Example 4. The true density to be estimated was defined by the mixture of five Gaussian distributions given as

$$p(x,y)=\sum_{i=1}^{5}\frac{1}{10\pi}e^{-(x-\mu_{i,1})^2/2}e^{-(y-\mu_{i,2})^2/2} \tag{54}$$

and the means of the five Gaussian distributions, $[\mu_{i,1}\ \ \mu_{i,2}]$, $1\le i\le 5$, were [0.0 −4.0], [0.0 −2.0], [0.0 0.0], [−2.0 0.0], and [−4.0 0.0], respectively. The number of data points for density estimation was $N=500$. The optimal kernel widths were found to be $\rho_{\mathrm{Par}}=0.5$, $\rho=1.1$, $\rho=1.0$ and $\rho=1.2$ for the PW, previous SKD, proposed SKD and RSDE estimators, respectively. The experiment was repeated $N_{\mathrm{run}}=100$ times.

The maximum number of selected kernels by the *D*-optimality based OFR was set to $N_s=16$. The maximum and minimum numbers of nonzero kernel weights found by the MNQP algorithm over the 100 runs were 9 and 6, respectively, and the average model size was $N_s=7.9$. We then used $N_s=8$ for the GMM, with the initialisation $[a,b]^2=[-8,4]^2$, $\rho^2_{\mathrm{ini}}=0.1$ and $\rho^2_{\mathrm{min}}=0.01$ for the EM algorithm. Table 4 compares the performance of the five density estimators studied, where it can be seen that the new SKD

Table 3
Performance comparison of the PW estimator, previous SKD estimator [24], proposed SKD estimator, RSDE estimator [19] and GMM estimator for the two-dimensional example of Gaussian and Laplacian mixture over 100 runs.

Estimator	PW	Previous SKD [24]	Proposed SKD	RSDE [19]	GMM
Kernel type	Fixed, $\rho_{\text{Par}} = 0.42$	Fixed, $\rho = 1.1$	Fixed, $\rho = 1.1$	Fixed, $\rho = 1.2$	Tunable
L_1 test error $\times 10^3$	4.036 ± 0.693	3.838 ± 0.780	3.694 ± 0.851	4.053 ± 0.446	3.474 ± 0.990
KLC $\times 10$	1.466 ± 0.228	1.403 ± 0.534	1.463 ± 1.067	0.896 ± 0.411	0.608 ± 0.172
Kernel no.	500	15.3 ± 3.9	11.0 ± 1.6	16.2 ± 3.4	11
Maximum	500	25	14	24	11
Minimum	500	8	5	9	11

Table 4
Performance comparison of the PW estimator, previous SKD estimator [24], proposed SKD estimator, RSDE estimator [19] and GMM estimator for the two-dimensional example of five-Gaussian mixture over 100 runs.

Estimator	PW	Previous SKD [24]	Proposed SKD	RSDE [19]	GMM
Kernel type	Fixed, $\rho_{\text{Par}} = 0.5$	Fixed, $\rho = 1.1$	Fixed, $\rho = 1.0$	Fixed, $\rho = 1.2$	Tunable Gaussian
L_1 test error $\times 10^3$	3.620 ± 0.439	3.610 ± 0.503	3.236 ± 0.558	3.631 ± 0.362	3.675 ± 0.672
KLC $\times 10^2$	3.422 ± 0.548	3.665 ± 0.920	3.474 ± 1.298	3.537 ± 0.485	3.392 ± 0.870
Kernel no.	500	13.2 ± 2.9	7.9 ± 0.8	13.2 ± 3.0	8
Maximum	500	22	9	21	8
Minimum	500	8	6	6	8

Table 5
Performance comparison for the two-class two-dimensional classification example.

Estimator	Kernel type	$\hat{p}(\bullet\|\text{C0})$	Kernel width	$\hat{p}(\bullet\|\text{C1})$	Kernel width	Test error rate (%)
PW	Fixed Gaussian	125 kernels	0.24	125 kernels	0.23	8.0
Previous SKD [24]	Fixed Gaussian	6 kernels	0.28	5 kernels	0.28	8.0
Proposed SKD	Fixed Gaussian	2 kernels	0.38	2 kernels	0.38	7.9
RSDE [19]	Fixed Gaussian	3 kernels	0.30	2 kernels	0.30	7.9
GMM	Tunable Gaussian	2 kernels	–	2 kernels	–	9.1

estimator had a similar test performance as the other two benchmark SKD estimators but it achieved a much sparser density estimate. The proposed SKD estimator had a further advantage of a much simpler computational complexity in the density construction process.

Example 5. This was a two-class classification problem in a two-dimensional feature space [37] and we obtained the data from [38]. 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 error rate based on the true underlying probability distribution for this example was known to be 8%. We first estimated the two conditional density functions $\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{N_s}, \rho|\text{C0})$ and $\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{N_s}, \rho|\text{C1})$ from the training data, and then applied the Bayes decision rule

$$\left.\begin{array}{ll} \text{if } \hat{p}(\mathbf{x}; \boldsymbol{\beta}_{N_s}, \rho|\text{C0}) \geq \hat{p}(\mathbf{x}; \boldsymbol{\beta}_{N_s}, \rho|\text{C1}), & \mathbf{x} \in \text{C0} \\ \text{else}, & \mathbf{x} \in \text{C1} \end{array}\right\} \tag{55}$$

to the test data set and calculated the corresponding error rate. Table 5 compares the results obtained by the five density estimates investigated, where the values of the kernel width ρ were determined by cross validation. Except for the GMM method, the other four methods all achieved the optimal Bayes classification performance. This clearly demonstrated the accuracy of these density estimates. The proposed SKD estimation method was seen to produce sparser density estimates than our previous SKD estimation method of [24] as well as the RSDE estimator of [19]. Figs. 11–15 illustrate the decision boundaries of the classifier (55) for the five density estimation methods investigated, respectively.

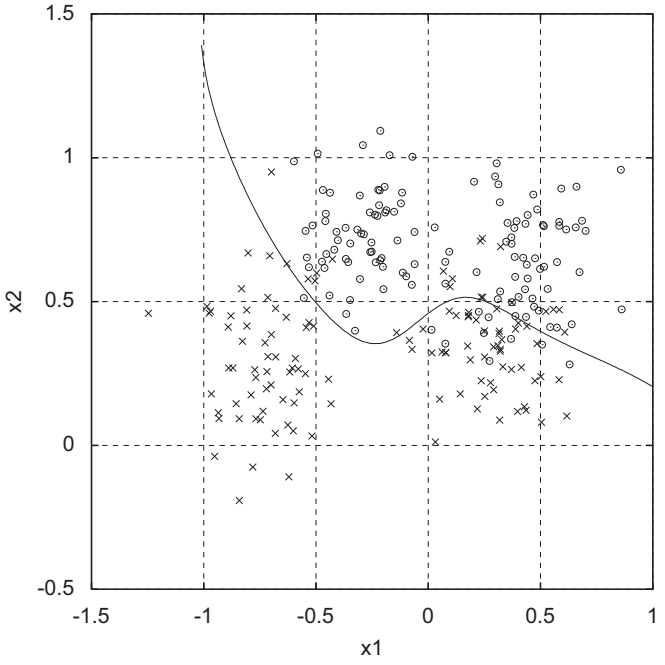


Fig. 11. Decision boundary of the PW estimator for the two-class two-dimensional classification example, where circles represent the class-1 training data and crosses the class-0 training data.

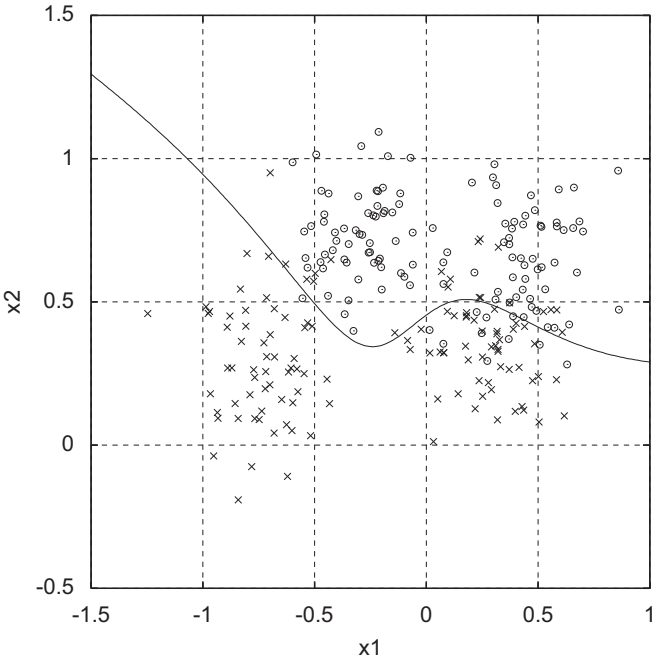


Fig. 12. Decision boundary of the previous SKD estimator [24] for the two-class two-dimensional classification example, where circles represent the class-1 training data and crosses the class-0 training data.

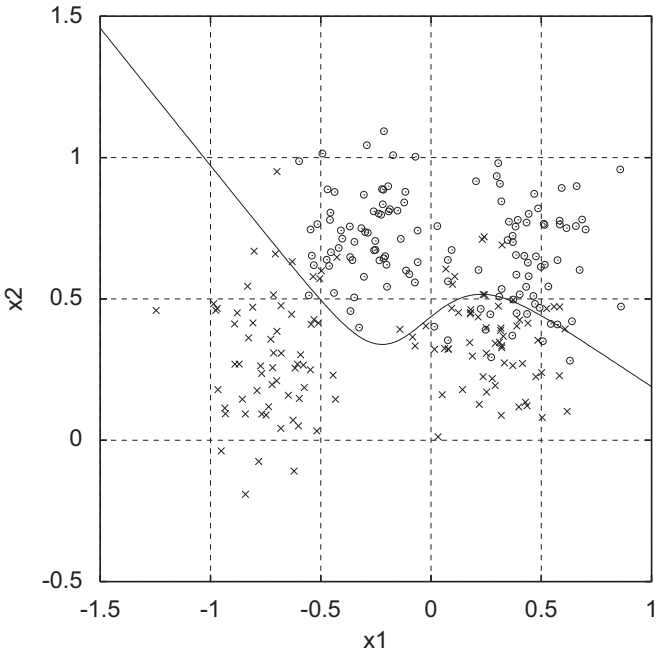


Fig. 14. Decision boundary of the RSDE estimator [19] for the two-class two-dimensional classification example, where circles represent the class-1 training data and crosses the class-0 training data.

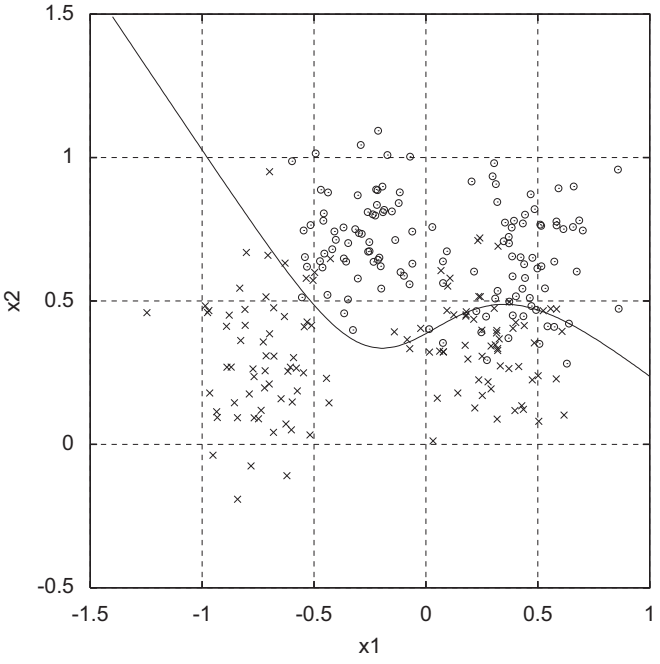


Fig. 13. Decision boundary of the proposed SKD estimator for the two-class two-dimensional classification example, where circles represent the class-1 training data and crosses the class-0 training data.

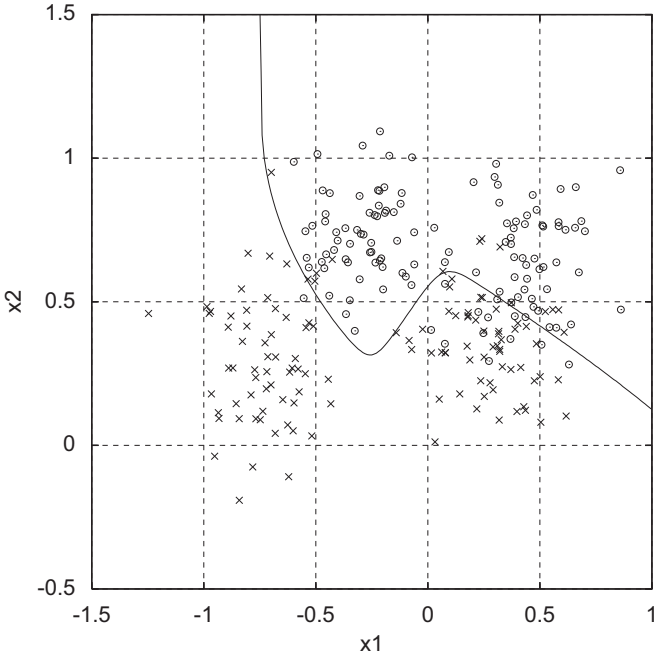


Fig. 15. Decision boundary of the GMM estimator for the two-class two-dimensional classification example, where circles represent the class-1 training data and crosses the class-0 training data.

4.3. Multi-dimensional examples

Example 6. In this six-dimensional example, the underlying density to be estimated was given by

$$p(\mathbf{x}) = \frac{1}{3}\sum_{i=1}^{3} \frac{1}{(2\pi)^{6/2}} \frac{1}{\det^{1/2}|\boldsymbol{\Gamma}_i|} e^{-(1/2)(\mathbf{x}-\boldsymbol{\mu}_i)^T \boldsymbol{\Gamma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)} \tag{56}$$

with

$$\boldsymbol{\mu}_1 = [1.0 \;\; 1.0 \;\; 1.0 \;\; 1.0 \;\; 1.0 \;\; 1.0]^T,$$

$$\boldsymbol{\Gamma}_1 = \mathrm{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\}, \tag{57}$$

$$\boldsymbol{\mu}_2 = [-1.0 \;\; -1.0 \;\; -1.0 \;\; -1.0 \;\; -1.0 \;\; -1.0]^T,$$

$$\boldsymbol{\Gamma}_2 = \mathrm{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}, \tag{58}$$

Table 6
Performance of the PW estimator, previous SKD estimator [24], proposed SKD estimator, RSDE estimator [19] and GMM estimator for the six-dimensional example of three-Gaussian mixture over 100 runs.

Estimator	PW	Previous SKD [24]	Proposed SKD	RSDE [19]	GMM
Kernel type	Fixed, $\rho_{\text{Par}}=0.65$	Fixed, $\rho=1.2$	Fixed, $\rho=1.2$	Fixed, $\rho=1.2$	Tunable
L_1 test error $\times 10^5$	3.520 ± 0.162	3.113 ± 0.534	2.782 ± 0.227	2.739 ± 0.500	1.743 ± 0.285
Kernel no.	600	9.4 ± 1.9	8.4 ± 0.9	14.2 ± 3.6	8
Maximum	600	16	10	25	8
Minimum	600	7	6	8	8

Table 7
Performance comparison for the Titanic classification data set, quoted as mean $\pm$ standard deviation over 100 realisations.

Estimator	Kernel no. $\hat{p}(\bullet\|\text{C0})+\hat{p}(\bullet\|\text{C1})$	Test error rate (%)
PW	150 ± 0	22.48 ± 0.43
Proposed SKD	7.8 ± 4.4	22.34 ± 0.34
RSDE [19]	36.9 ± 5.7	22.57 ± 0.93
GMM	8 ± 0	23.86 ± 3.22

$$\boldsymbol{\mu}_3 = [0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0 \;\; 0.0]^T,$$

$$\boldsymbol{\Gamma}_3 = \text{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\}. \tag{59}$$

The estimation data set contained $N=600$ samples. The optimal kernel widths were found empirically to be $\rho=0.65$ for the PW estimator and $\rho=1.2$ for the three SKD estimators, respectively. The experiment was repeated $N_{\text{run}}=100$ times. The number of kernels selected by the *D*-optimality based OFR was again set to $N_s=16$. The maximum and minimum numbers of nonzero kernels weights determined by the MNQP algorithm were 10 and 6, respectively, over the $N_{\text{run}}=100$ runs and the final average model size was $N_s=8.4$. The number of mixture components used for the GMM was therefore $N_s=8$. An appropriate initialisation for the EM algorithm was found to be $[a,b]^6=[-5,5]^6$, $\rho^2_{\text{ini}}=0.1$ and $\rho^2_{\min}=0.01$. The results obtained by the five density estimators are summarised in Table 6. It can be seen from Table 6 that the proposed SKD estimator achieved a similar test accuracy with much sparser estimates than our previous SKD estimator [24] as well as the RSDE estimator [19]. For this example, the GMM estimator achieved the best test accuracy.

Example 7. This was a two-class classification data set, Titanic, and we obtained the data set from [39]. The feature space dimension was $m=3$, and there were 100 realisations of the data set. Each realisation contained 150 training samples and 2051 test data samples. Note that the two-class data samples were imbalanced, with the class-0 training samples approximately twice of the class-1 training samples. In [40], a range of classifiers were applied to this data set, and the best classification test error rate in %, obtained by the SVM classifier, averaged over the 100 realisations was 22.42 ± 1.02.

We first estimated the two conditional density functions $\hat{p}(\mathbf{x};\boldsymbol{\beta}_{N_s},\rho|\text{C0})$ and $\hat{p}(\mathbf{x};\boldsymbol{\beta}_{N_s},\rho|\text{C1})$ from the training data, and then applied the Bayes decision rule (55) to the test data and calculated the corresponding error rate. Four density estimation methods, the PW, proposed SKD, RSDE and GMM estimators, were tested. The values of kernel width for the first three density estimators were determined via cross validation. For the GMM method, we use four mixture components for each conditional density estimation. The results obtained by the four methods are listed in Table 7, where it can be seen that the proposed density estimation method produced the best result. The optimal sparsity property of the proposed *D*-optimality based SKD estimator was demonstrated by the fact that it produced the sparsest density estimates and furthermore the two conditional density estimates had approximately equal numbers of kernels, despite the highly imbalanced two class training data sets. This desired property for example was not observed for the RSDE estimator, which produced the class-0 density estimate having much larger number of kernels than the class-1 density estimate.

5. Conclusions

An efficient *D*-optimality based construction algorithm has been proposed for obtaining SKD estimates. A very small subset of significant kernels are first selected using the OFR procedure based on the *D*-optimality criterion. The associated kernel weights are then calculated using a modified version of the MNQP algorithm, which can further reduce the kernel model size by making some of the kernel weights to zero. The proposed method possesses a highly desired optimal sparsity property owing to the ability of the *D*-optimality based OFR algorithm to automatically identify a very small subset of the most significant kernels related to the largest eigenvalues of the kernel matrix, which counts for the most energy of the kernel training data. As a consequence of this optimal property, the subset kernel weight vector estimate is guaranteed to be the most accurate estimate. Furthermore, the proposed method is simple to implement and computationally efficient in comparison with other existing SKD estimation methods. The experimental results obtained have demonstrated that the proposed method compares favourably with other existing sparse kernel density estimation methods, in terms of test accuracy and sparsity of the estimate as well as complexity of density estimation process. Thus it provides a viable alternative to these existing state-of-the-art methods for constructing sparse kernel density estimates in practical applications.

Recently, research effort has also been directed to construct the RBF network or kernel model with tunable nodes [41–47]. In particular, the work [48] has investigated the application of the tunable RBF network to the PDF estimation. Further work is warranted to compare the proposed efficient sparse fixed-kernel density estimation approach with the nonlinear optimisation aided tunable-kernel density estimation method of [48].

References

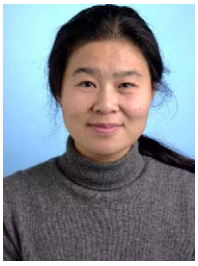
[1] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[2] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, UK, 1995.
[3] B.W. Silverman, Density Estimation, Chapman & Hall, London, 1996.
[4] A.W. Bowman, A. Azzalini, Applied Smoothing Techniques for Data Analysis, Oxford University Press, Oxford, UK, 1997.
[5] H. Wang, Robust control of the output probability density functions for multivariable stochastic systems with guaranteed stability, IEEE Trans. Autom. Control 44 (1998) 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 (2001) 1240–1247.

[7] G. McLachlan, D. Peel, Finite Mixture Models, Wiley, New York, 2000.
[8] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. R. Stat. Soc. B 39 (1977) 1–38.
[9] 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 Berkeley, USA, 1997.
[10] B. Efron, R.J. Tibshirani, An Introduction to Bootstrap, Chapman & Hall, London, 1993.
[11] Z.R. Yang, S. Chen, Robust maximum likelihood training of heteroscedastic probabilistic neural networks, Neural Networks 11 (1998) 739–747.
[12] M. Svensén, C.M. Bishop, Robust Bayesian mixture modelling, Neurocomputing 64 (2005) 235–252.
[13] C. Archambeau, M. Verleysen, Robust Bayesian clustering, Neural Networks 20 (2007) 129–138.
[14] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (1962) 1066–1076.
[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] S. Mukherjee, V. Vapnik, Support vector method for multivariate density estimation, Technical Report A.I. Memo No. 1653, MIT AI Lab, USA, 1999.
[17] 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.
[18] L. Song, X. Zhang, A. Smola, A. Gretton, B. Schöolkopf, Tailoring density estimation via reproducing kernel moment matching, in: Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, July 5–9, 2008, pp. 992–999.
[19] M. Girolami, C. He, Probability density estimation from optimally condensed data samples, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 1253–1264.
[20] A. Choudhury, Fast machine learning algorithms for large data, Ph.D. Thesis, Computational Engineering and Design Center, School of Engineering Sciences, University of Southampton, Southampton, UK, August 2002.
[21] 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 (2003) 1029–1036.
[22] 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 (2004) 898–911.
[23] 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 (2004) 1708–1717.
[24] S. Chen, X. Hong, C.J. Harris, An orthogonal forward regression techniques for sparse kernel density estimation, Neurocomputing 71 (2008) 931–943.
[25] 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.
[26] X. Hong, S. Chen, C.J. Harris, A forward-constrained regression algorithm for sparse kernel density estimation, IEEE Trans. Neural Networks 19 (2008) 193–1981.
[27] A.C. Atkinson, A.N. Donev, Optimum Experimental Designs, Clarendon Press, Oxford, UK, 1992.
[28] X. Hong, C.J. Harris, Neurofuzzy design and model construction of nonlinear dynamical processes from data, IEE Proc. Control Theory Appl. 148 (2001) 530–538.
[29] X. Hong, C.J. Harris, Nonlinear model structure detection using optimum design and orthogonal least squares, IEEE Trans. Neural Networks 12 (2001) 435–439.
[30] X. Hong, C.J. Harris, Nonlinear model structure design and construction using orthogonal least squares and D-optimality design, IEEE Trans. Neural Networks 13 (2002) 1245–1250.
[31] X. Hong, C.J. Harris, S. Chen, P.M. Sharkey, Robust nonlinear model identification methods using forward regression, IEEE Trans. Syst. Man Cybern. Part A 33 (2003) 514–523.
[32] M. Stone, Cross validation choice and assessment of statistical predictions, J. R. Stat. Soc. Ser. B 36 (1974) 111–147.
[33] R.H. Myers, Classical and Modern Regression with Applications, second ed., PWS-KEN, Boston, 1990.
[34] Glivenko-Cantelli theorem. [Online] Available 〈http://en.wikipedia.org/wiki/Glivenko-Cantelli_theorem〉.
[35] B. Schölkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson, Estimating the support of a high-dimensional distribution, Neural Comput. 13 (2001) 1443–1471.
[36] S. Chen, J. Wigger, Fast orthogonal least squares algorithm for efficient subset model selection, IEEE Trans. Signal Process. 43 (1995) 1713–1715.
[37] B.D. Ripley, Pattern Recognition and Neural Networks, Cambridge University Press, Cambridge, UK, 1996.
[38] [Online] Available 〈http://www.stats.ox.ac.uk/PRNN〉.
[39] [Online] Available 〈http://ida.first.fhg.de/projects/bench/benchmarks.htm〉.
[40] G. Rätsch, T. Onoda, K.R. Müller, Soft margins for AdaBoost, Mach. Learn. 42 (2001) 287–320.
[41] S. Chen, X. Hong, C.J. Harris, Orthogonal forward selection for constructing the radial basis function network with tunable nodes, in: Proceedings of the 2005 International Conference Intelligent Computing, Hefei, China, August 23–26, 2005, pp. 777–786.
[42] S. Chen, X. Hong, C.J. Harris, Construction of RBF classifiers with tunable units using orthogonal forward selection based on leave-one-out misclassification rate, in: Proceedings of the 2006 International Joint Conference on Neural Networks, Vancouver, Canada, July 16–21, 2006, pp. 6390–6394.
[43] X.X. Wang, S. Chen, D. Lowe, C.J. Harris, Parsimonious least squares support vector regression using orthogonal forward selection with the generalised kernel mode, Int. J. Modelling, Identification Control 1 (2006) 245–256.
[44] X.X. Wang, S. Chen, D. Lowe, C.J. Harris, Sparse support vector regression based on orthogonal forward selection for the generalised kernel model, Neurocomputing 70 (2006) 462–474.
[45] J.-X. Peng, G. Li, G.W. Irwin, A new Jacobian matrix for optimal learning of single-layer neural networks, IEEE Trans. Neural Networks 19 (2008) 119–129.
[46] S. Chen, X. Hong, B.L. Luk, C.J. Harris, Construction of tunable radial basis function networks using orthogonal forward selection, IEEE Trans. Syst. Man Cybern. Part B 39 (2009) 457–466.
[47] K. Li, J.-X. Peng, E.-W. Bai, Two-stage mixed discretecontinuous identification of radial basis function (RBF) neural models for nonlinear systems, IEEE Trans. Circuits Syst. Part I 56 (2009) 630–643.
[48] S. Chen, X. Hong, C.J. Harris, Probability density estimation with tunable kernels using orthogonal forward regression, IEEE Trans. Syst. Man Cybern. Part B (2010), to appear.

Sheng Chen received his PhD degree in control engineering from the City University, London, UK, in September 1986. He was awarded the Doctor of Sciences (DSc) degree by the University of Southampton, Southampton, UK, in 2004.

From October 1986 to August 1999, he held research and academic appointments at the University of Sheffield, the University of Edinburgh and the University of Portsmouth, all in UK. Since September 1999, he has been with the School of Electronics and Computer Science, University of Southampton, UK. Professor Chen's research interests include wireless communications, adaptive signal processing for communications, machine learning, and evolutionary computation methods. He has published over 400 research papers.

Dr. Chen is a Fellow of IET and a Fellow of IEEE. 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, China (BSc 1984, MSc, 1987), and her PhD degree from University of Sheffield, 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. Dr. Hong 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 180 research papers, and coauthored 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, in 1972. He was awarded the Doctor of Sciences (DSc) degree by the University of Southampton in 2001.

He previously held appointments at the University of Hull, the UMIST, the University of Oxford, and the University of 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 the Royal Academy of Engineering in 1996, was awarded the IEE Senior Achievement medal in 1998 for his work in autonomous systems, and the IEE Faraday medal in 2001 for his work in intelligent control and neurofuzzy systems.