Pattern Anal Applic (2015) 18:367–375
DOI 10.1007/s10044-013-0330-1

THEORETICAL ADVANCES

# Robust sparse kernel density estimation by inducing randomness

**Fei Chen · Huimin Yu · Jincao Yao · Roland Hu**

Received: 3 June 2012 / Accepted: 30 March 2013 / Published online: 17 April 2013

**Abstract** In this paper, a robust sparse kernel density estimation based on the reduced set density estimator is proposed. The key idea is to induce randomness to the plug-in estimation of weighting coefficients. The random fluctuations can inhibit these small nonzero weighting coefficients to cluster in regions of space with greater probability mass. By sequential minimal optimization, these coefficients are merged into a few larger weighting coefficients. Experimental studies show that the proposed model is superior to several related methods both in sparsity and accuracy of the estimation. Moreover, the proposed density estimation is extensively validated on novelty detection and binary classification.

**Electronic supplementary material** The online version of this article (doi:10.1007/s10044-013-0330-1) contains supplementary material, which is available to authorized users.

F. Chen · H. Yu (✉) · J. Yao · R. Hu
Department of Information Science and Electronic Engineering, Zhejiang University, No. 38 Zheda Road, Hangzhou 310027, China
e-mail: yhm2005@zju.edu.cn

F. Chen
e-mail: chenfei314@zju.edu.cn

J. Yao
e-mail: yaojincao@zju.edu.cn

R. Hu
e-mail: haoji_hu@zju.edu.cn

F. Chen
School of Sciences, Jimei University, Xiamen 361021, China

H. Yu
State Key Laboratory of CAD&CG, Hangzhou 310027, China

**Keywords** Kernel density estimator · Reduced set density estimator · Integrated squared error · Sequential minimal optimization · Randomness

## 1 Introduction

Density estimation is widely used in statistical feature models in computer vision and pattern recognition. Given a set of training data of the features (e.g., intensity, shape, texture), the underlying probability density can be described by a simple or more complicated distribution function. Uniform distribution [1], Gaussian distribution [2] or nonparametric functional [3] were considered in the past. By contrast, without the assumption that the forms of the underlying densities are known, a kernel density estimator (KDE) (also called the Parzen window estimate) is an efficient nonparametric approach to model nonlinear distributions of training data. In this technique the density function is estimated by a sum of kernel functions. The kernel number is equal to the size of the training data. When the training data set is very large, the KDE suffers from high computational cost and becomes intractable for subsequent use. In addition, the feature space is complex, noisy and most often not all the training data obey the same parametric model. This leads to a need for robust estimators to handle data in the presence of severe contaminations, i.e., outliers. In this work, we focus on the problem of how to employ a small percentage of the available data sample to provide a robust and highly accurate density estimator.

KDE is frequently used for various computer vision problems, such as mean shift [4], background subtraction [5], object tracking [6], image segmentation [3, 7] and classification [8]. Even though there have been several attempts to improve the computational efficiency [9, 10], its very high memory requirements and computational complexity inhibit the use of kernel density estimation in real applications. In [11], the support vector approach was used to obtain an estimate from the training data in the form of a mixture of densities. This approach has no additional free parameters. However, for large sample sizes, it requires $O(n^3)$ optimization routines. The reduced set density estimator (RSDE) was proposed by Girolami and He [12] to solve the above problem by providing a KDE which employs a small subset of the available training data. It is optimal in the integrated squared error (ISE) between the unknown true density and the RSDE. In contrast to the support vector approach, the RSDE only requires $O(n^2)$ optimization routines to provide similar levels of performance. In order to increase the sparsity further in the weight coefficients, Chen et al. [13] constructed a sparse kernel density estimate using an orthogonal forward regression technique using the classical Parzen window estimate as the desired response. In addition, sparse kernel density estimate has gained attention toward the integration of explicit sparse constraint to the weight coefficients as regularization term [14, 15]. These methods create a trade-off between the sparsity and the quality of the density estimation. They can produce sparsity in the samples at the cost of a slight reduction in the quality of the estimates.

Instead of creating a new probability density estimator, we try to generalize the RSDE to provide more satisfying performance. In this paper, our work focuses on the RSDE based on KDE with Gaussian kernel. In RSDE, there exist many nonzero coherent weighting coefficients which are clustered in regions of space with greater probability mass, specifically for low dimensional data. In order to break the relationship between coherent coefficients, our idea is to induce randomness to the plug-in estimation of weighting coefficients. By means of sequential minimal optimization (SMO), these coherent weighting coefficients can be replaced approximately by one or several larger incoherent weighting coefficients. In contrast to the RSDE, the proposed model can improve the sparsity and accuracy of the density estimation. Moreover, this technique is robust to outliers by analysis in feature space.

This paper is organized as follows. In Sect. 2 the RSDE is reviewed briefly, and in Sect. 3 the proposed robust sparse kernel density estimation by random fluctuations for coherent coefficients is presented. Experimental results are provided in Sect. 4, and the conclusions in Sect. 5.

## 2 Reduced set density estimator

Given $n$ data samples $x_1, x_2, \ldots, x_n \in R^\ell$, each have a weight $\omega_i \geq 0, \sum_{i=1}^n \omega_i = 1$, and the distribution density can be estimated by a KDE with weight coefficients,

$$\hat{f}(x;\omega) = \sum_{i=1}^n \omega_i k_\sigma(x, x_i), \tag{1}$$

where $k_\sigma(x, x_i)$ is a kernel function (satisfying non-negativity and normalization conditions), and $\sigma$ is a parameter which controls the kernel width. The most commonly used kernel function is a Gaussian kernel

$$k_\sigma(x, x_i) = (2\pi\sigma^2)^{-\frac{\ell}{2}} \exp\left\{ -\frac{\|x - x_i\|^2}{2\sigma^2} \right\} \tag{2}$$

Girolami and He [12] estimated KDE by minimizing ISE between the true density $f(x)$ and the estimated density $\hat{f}(x;\omega)$, which was defined as

$$\begin{aligned} \mathrm{ISE}(\omega) &= \int \left| f(x) - \hat{f}(x;\omega) \right|^2 \mathrm{d}x \\ &= \int \hat{f}^2(x;\omega)\mathrm{d}x - 2\int \hat{f}(x;\omega) f(x)\mathrm{d}x \\ &\quad + \int f^2(x)\mathrm{d}x \end{aligned} \tag{3}$$

Notice that the first term is

$$\begin{aligned} \int \hat{f}^2(x;\omega)\mathrm{d}x &= \int \left( \sum_{i=1}^n \omega_i k_\sigma(x, x_i) \right)^2 \mathrm{d}x \\ &= \sum_{i=1}^n \sum_{j=1}^n \omega_i \omega_j \int k_\sigma(x, x_i) k_\sigma(x, x_j) \mathrm{d}x \\ &= \omega^{\mathrm{T}} Q \omega \end{aligned}$$

Here, $\omega = [\omega_1, \omega_2, \ldots, \omega_n]^{\mathrm{T}}$ and $Q = \int k_\sigma(x, x_i) k_\sigma(x, x_j) \mathrm{d}x$ is a $n \times n$ matrix whose elements are defined as $Q_{ij} = k_{\sqrt{2}\sigma}(x_i, x_j)$. Since we do not know the true $f(x)$, we need to estimate the second term, which is denoted as $M(\omega)$. An unbiased estimate of it for a KDE can be written as

$$\begin{aligned} M(\omega) &= \int \hat{f}(x;\omega) f(x) \mathrm{d}x \\ &\approx \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \omega_i k_\sigma(x_i, x_j) = \sum_{i=1}^n \omega_i \frac{1}{n} \sum_{j=1}^n k_\sigma(x_i, x_j) \\ &= \sum_{i=1}^n \omega_i d_i = \omega^{\mathrm{T}} D \end{aligned}$$

Here, the KDE for the point $x_i$ is denoted as $d_i = \frac{1}{n} \sum_{j=1}^n k_\sigma(x_i, x_j)$, and $D = [d_1, d_2, \ldots, d_n]^{\mathrm{T}}$. Notice that $D = K 1_n$, where $K$ is the Gram matrix whose elements are defined as $K_{ij} = k_\sigma(x_i, x_j)$, and $1_n$ is the $n \times 1$ vector whose elements are all $\frac{1}{n}$. The last term in (3) can been dropped due to its independence of $\omega$. Then the minimum ISE can be written as follows

$$\begin{aligned} \hat{\omega} = \arg\min_{\omega} &\left\{ \mathrm{ISE}(\omega) = \frac{1}{2} \omega^{\mathrm{T}} Q \omega - \omega^{\mathrm{T}} D \right\} \\ \text{s.t.} \quad &\omega^{\mathrm{T}} 1 = 1 \quad \text{and} \quad \omega_i \geq 0 \ \forall i \end{aligned} \tag{4}$$

Observe that the $Q$ is positive semi-definite. Thus, the object function is convex with respect to $\omega$, and can be solved using SMO [12, 16].

## 3 Robust sparse kernel density estimation

The RSDE was shown in [12] to be able to provide a sparse representation in the weighting coefficients. The authors observed that the weights obtained from minimizing the estimated ISE were sparse. This is because the right hand term $\omega^{\mathrm{T}}D$ in (4) is a convex combination of positive numbers. Such a convex combination is maximized by assigning a unit weight to the largest, and setting the rest to zero. SMO is a simple algorithm that can quickly solve the quadratic programming (QP) optimization problem (4), by breaking the large QP problem into a series of smallest possible QP problems. The optimal solution should satisfy the following rules

(R1) If $\omega_i = 0, \omega_j > 0$ then $I_i \geq I_j$.

(R2) If $\omega_i, \omega_j > 0$ then $I_i = I_j$.

Here, $I_i = \sum_{j=1}^{n} Q_{ij}\omega_j - d_i$. Clearly, if more points satisfy rule (R1), then we can obtain a more sparse solution. In rule (R2), $\omega_i$ and $\omega_j$ are positive and updated only when $I_i \neq I_j$. Therefore, we hope to reduce the number of points which satisfy rule (R2) in order to increase the sparsity further in the weight coefficients. Due to the convex constraint, the coefficients obtained from SMO are naturally sparse. If we suppose that the precision of numbers is sufficient ($I_i \neq I_j, i \neq j$), then we will finally obtain only one nonzero weighting coefficient in $\omega$ by using SMO. It implies that the true density $f(x)$ is estimated by only one kernel function with a nonzero weight coefficient $\omega_k$,$\hat{f}(x;\omega) = \omega_k k_\sigma(x, x_k)$. Obviously, this is a time-consuming method to enforce sparsity and the resulting density estimator is inaccurate. Even when the precision of numbers is increased, the RSDE allows for many relatively close points and their corresponding weighting coefficients are small and nonzero in the optimal solution. As can be seen easily in Fig. 1a, these nonzero weighting coefficients are clustered in regions of space with greater probability mass. If these points in each cluster could be replaced approximately by one or several points with larger weighting coefficients, then this can improve the sparsity further in the weight coefficients.

For the Gaussian kernel, note that there exists a feature mapping functional $\phi_\sigma : \mathbb{R}^\ell \rightarrow \mathbb{R}^L (\ell < L)$. It maps the feature to a high dimensional feature space: $x \rightarrow \phi_\sigma(x)$, such that $K_{ij} = k_\sigma(x_i, x_j) = \langle \phi_\sigma(x_i), \phi_\sigma(x_j) \rangle$ [17]. Then the KDE with Gaussian kernel can be represented as the inner product between a mapped test point and the centroid of mapped training points in kernel feature space [18]. We have $d_i = \frac{1}{n}\sum_{k=1}^{n} k_\sigma(x_i, x_k) = \langle \phi_\sigma(x_i), \frac{1}{n}\sum_{k=1}^{n} \phi_\sigma(x_k) \rangle$. Here, $\frac{1}{n}\sum_{k=1}^{n} \phi_\sigma(x_k)$ can be treated as nonzero constants which clearly do not depend upon the value $x_i$. Similarly, there exists $\phi_{\sqrt{2}\sigma}$ such that $Q_{ij} = k_{\sqrt{2}\sigma}(x_i, x_j) = \left\langle \phi_{\sqrt{2}\sigma}(x_i), \phi_{\sqrt{2}\sigma}(x_j) \right\rangle$ and $H_i = \sum_{k=1}^{n} Q_{ik}\omega_k = \left\langle \phi_{\sqrt{2}\sigma}(x_i), \sum_{k=1}^{n} \omega_k \phi_{\sqrt{2}\sigma}(x_k) \right\rangle$. By analysis in feature space, we have

$$I_i - I_j = \left\langle \phi_{\sqrt{2}\sigma}(x_i) - \phi_{\sqrt{2}\sigma}(x_j), \sum_{k=1}^{n} \omega_k \phi_{\sqrt{2}\sigma}(x_k) \right\rangle - \left\langle \phi_\sigma(x_i) - \phi_\sigma(x_j), \frac{1}{n}\sum_{k=1}^{n} \phi_\sigma(x_k) \right\rangle$$

If $d_i = d_j$, then we have $\langle \phi_\sigma(x_i) - \phi_\sigma(x_j), \frac{1}{n}\sum_{k=1}^{n} \phi_\sigma(x_k) \rangle = 0$, and $\phi_\sigma(x_i) = \phi_\sigma(x_j)$. Thus, $\phi_{\sqrt{2}\sigma}(x_i) = \phi_{\sqrt{2}\sigma}(x_j)$, and $I_i = I_j$.

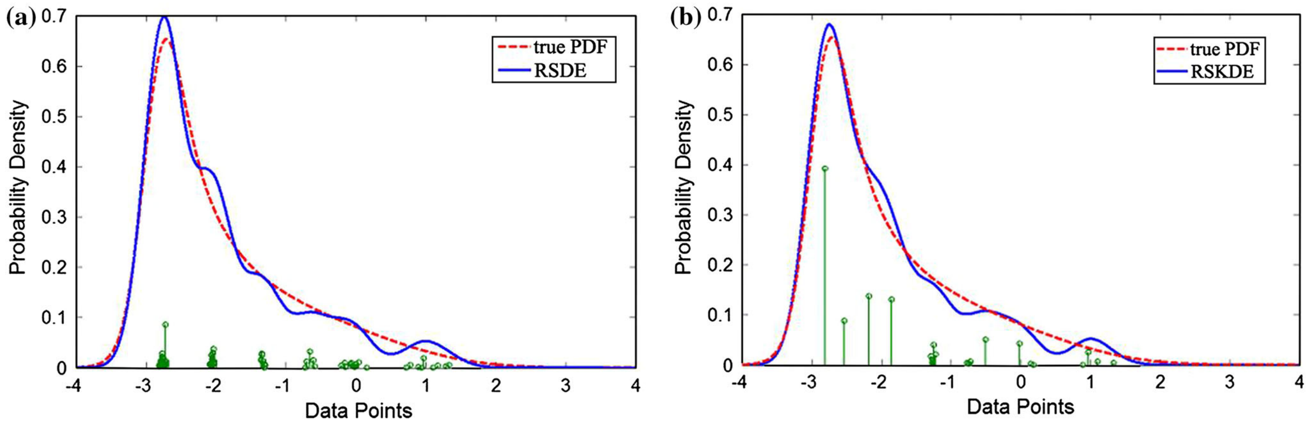

**Fig. 1** **a** The true density and the RSDE; each of the 87 nonzero weighting coefficients is placed at the appropriate sample data point and the value is denoted by the length of the vertical line. The corresponding $L_1$ error and $L_2$ error between them are 0.0147 and $4.54 \times 10^{-4}$. **b** The true density and the RSKDE; each of the 20 nonzero weighting coefficients is placed at the appropriate sample data point. The corresponding $L_1$ error and $L_2$ error between them are 0.014 and $3.71 \times 10^{-4}$

In fact, $d_i, d_j$ are unequal but may be very close. Assume that $d_i$ is very close to $d_j$, then their feature points $\phi_\sigma(x_i)$ and $\phi_\sigma(x_j)$ are also very close. It implies that it is easy to meet the condition $I_i = I_j$ in the rule (R2). In other words, if $d_i, d_j$ are close enough, and $\omega_i$ is nonzero, then $\omega_j$ is more likely to be nonzero. Hence, there are two cases for $d_i$ and $d_j$ in the rule (R2): (a) $|d_i - d_j| < \delta$ ($\delta$ is small enough), (b) $|d_i - d_j| \geq \delta$. If $\omega_i$ and $\omega_j$ satisfy case (a) in rule (R2), then $\omega_i, \omega_j > 0$ are called coherent coefficients.

### 3.1 Random perturbation of coherent coefficients

In this section, we hope to break the relationship between coherent coefficients, which are clustered in regions of space with greater probability mass. A natural approach is to induce randomness to $D$, in order to produce incoherence for most of $d_i \in D$. Based on the existing structure of $D$, a part of values that stay very close are added a small random values, while keeping the rest unchanged. The randomness can make all these elements in each cluster stay apart from each other. Assume that there are $n_0$ coherent coefficients $\omega_1, \ldots, \omega_{n_0} > 0$, and the corresponding $d_1, \ldots, d_{n_0}$ are close enough. After inducing random values to $d_i$, the relationship $I_1 = \cdots = I_{n_0}$ in rule (R2) does not exist. Clearly, case (a) in rule (R2) would be reduced and a part of them would be reclassified to rule (R1). The weight coefficients in the optimal solution could be made more sparse. Assume that all the elements of $D$ are collected into a set $\Omega$. Then we have the following definitions to partition the set $\Omega$.

**Definition 1** Coherent relation $\approx$ is defined as: $d_i \approx d_j \Leftrightarrow \lfloor d_i \rfloor_m = \lfloor d_j \rfloor_m$, $d_i, d_j \in \Omega$.

There are many methods to describe the coherent relation. Here, considering that $d_i$ is a positive decimal number, the truncated $m$-digit approximation to it is the number $\lfloor d_i \rfloor_m$ obtained by simply discarding all digits beyond the $m$th. Hence, we have $\left| d_i - \lfloor d_i \rfloor_m \right| < 10^{-m}$. Clearly, the coherent relation $d_i \approx d_j$ is an equivalence relation that identifies those numbers of $\Omega$ that stay very close. Moreover, this relationship gives rise to a partition of $\Omega$ into equivalence classes.

**Definition 2** Coherent set is an equivalence class defined as: $\Omega(\tilde{d}_i) = \left\{ d_j | (d_j \in \Omega) \wedge (\tilde{d}_i \approx d_j) \right\}$. Here, $\tilde{d}_i$ is called a generator. The partition induced by the coherent relation is given by: $\Pi(\Omega, \approx) = \{\Omega_1, \ldots, \Omega_c\}, c \leq |\Omega|$, where $|\Omega|$ is the cardinality of $\Omega$, and $|\Omega_1| \geq |\Omega_2| \geq \cdots \geq |\Omega_c|$. Let $\tilde{d}_i$ be the corresponding generators of $\Omega_i$, $i = 1, 2, \ldots, c$, and they form a set $\Omega_B = \left\{ \tilde{d}_1, \tilde{d}_2, \ldots, \tilde{d}_c \right\}$. Subsequently, we define $\Omega_N = \Omega - \Omega_B$, and obtain the corresponding partition $\Pi(\Omega_N, \approx) = \left\{ \Omega_1^N, \Omega_2^N, \ldots, \Omega_t^N \right\}$, where $\left| \Omega_1^N \right| \geq \left| \Omega_2^N \right| \geq \cdots \geq \left| \Omega_t^N \right|$. Obviously, $t < c$ and $\left| \Omega_i^N \right| = |\Omega_i| - 1, \forall i = 1, 2, \ldots, t$. Here, we select an appropriate $m$ or magnify the values of $\Omega$ for partition such that $|\Omega_N| < |\Omega_B| < n$.

**Definition 3** $\phi_\sigma(x_i)$ and $\phi_\sigma(x_j)$ are coherent feature points, such that $\phi_\sigma(x_i) \approx \phi_\sigma(x_j) \Leftrightarrow d_i \approx d_j$. Since one to one correspondence between $d_i$ and $\phi_\sigma(x_i)$, we define the corresponding feature sets $\Theta = \{\phi_\sigma(x_1), \phi_\sigma(x_2), \ldots, \phi_\sigma(x_n)\}$, $\Theta_B, \Theta_N = \{\Theta_1, \Theta_2, \ldots, \Theta_t\}$ of $\Omega, \Omega_B, \Omega_N$.

Given a coherent relation $\approx$, the set $\Omega$ is divided into $\Omega_B$ and $\Omega_N$ (Fig. 2). Our method is to induce randomness to $\Omega_N$, and keep $\Omega_B$ unchanged. For any $d \in \Omega_i^N, i = 1, 2, \ldots, t$, we obtain $\bar{\Omega}_i^N$ by setting $d^* = d + \lambda_i r$, where $r$ is a random value from a uniform distribution on the interval [0, 1], and $\lambda_i$ is a scaling parameter for $\Omega_i^N$. Here, $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_t$. In our experiments, a simple method is used to define these scaling parameters, $\lambda_i = \left( d_i - \lfloor d_i \rfloor_m \right) \left| \Omega_i^N \right|$. Then, $\Omega_N^* = \left\{ \bar{\Omega}_1^N, \bar{\Omega}_2^N, \ldots, \bar{\Omega}_t^N \right\}$ is obtained and $\Omega_B, \Omega_N, \Omega_N^*$ can be written in matrix form as $D_B, D_N, D_N^*$. After rearrangement, the proposed ISE approximation model can be minimized as

$$\begin{aligned} \left[ \hat{\omega}_B \quad \hat{\omega}_N \right] &= \underset{\omega_B, \omega_N}{\operatorname{argmin}} \{ \mathrm{ISE}^*(\omega_B, \omega_N) \\ &= \frac{1}{2} \left[ \omega_B \quad \omega_N \right] \begin{bmatrix} Q_{BB} & Q_{BN} \\ Q_{NB} & Q_{NN} \end{bmatrix} \begin{bmatrix} \omega_B \\ \omega_N \end{bmatrix} \\ &\quad - \left[ \omega_B \quad \omega_N \right] \begin{bmatrix} D_B \\ D_N^* \end{bmatrix} \Bigg\}, \end{aligned} \tag{5}$$

which is called the robust sparse kernel density estimation (RSKDE). In contrast to the RSDE, we use $D_N^*$ instead of $D_N$, $D_N^* = D_N + R$. Here, $R = [R_1, R_2, \ldots, R_{b_t}]^{\mathrm{T}} = [\lambda_1 r_1, \ldots, \lambda_1 r_{b_1}, \lambda_2 r_{b_1+1}, \ldots, \lambda_2 r_{b_2}, \ldots, \lambda_t r_{b_t}]^{\mathrm{T}}$, where $b_j = \sum_{i=1}^{j} \left| \bar{\Omega}_i^N \right|, j = 1, 2, \ldots, t$ and $r_i$ is a random value from a uniform distribution on the interval $[0, 1]$, $i = 1, 2, \ldots, b_t$.

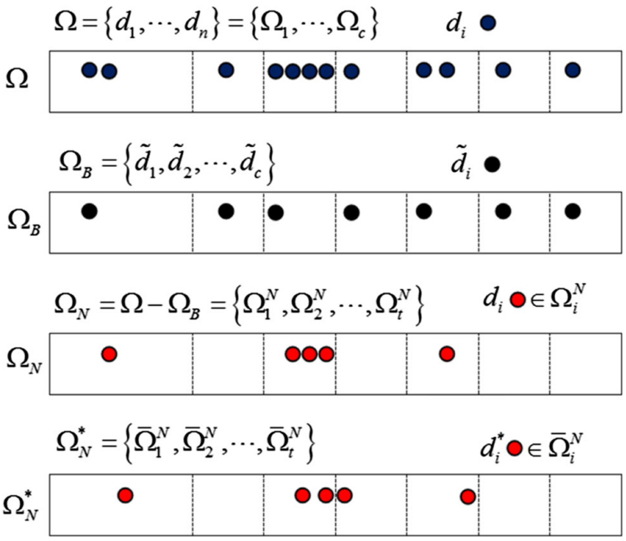
$\Omega = \{d_1, \cdots, d_n\} = \{\Omega_1, \cdots, \Omega_c\}$
$d_i$ ●
$\Omega$
$\Omega_B = \{\tilde{d}_1, \tilde{d}_2, \cdots, \tilde{d}_c\}$
$\tilde{d}_i$ ●
$\Omega_B$
$\Omega_N = \Omega - \Omega_B = \{\Omega_1^N, \Omega_2^N, \cdots, \Omega_t^N\}$
$d_i$ ● $\in \Omega_i^N$
$\Omega_N$
$\Omega_N^* = \{\bar{\Omega}_1^N, \bar{\Omega}_2^N, \cdots, \bar{\Omega}_t^N\}$
$d_i^*$ ● $\in \bar{\Omega}_i^N$
$\Omega_N^*$

**Fig. 2** A diagram of set partition

Observe that the object function (5) stays convex with respect to $\omega$, and SMO is also used to solve this problem. Set partitioning is easy to implement in computer program. Our implementation approach is summarized in Algorithm 1.

**Algorithm 1** Robust sparse kernel density estimation

```
Step 1: Choose appropriate $m$, and generate a random vector
    $[r_1, \ldots, r_n],\ r_i \in [0, 1]$
Step 2: Compute $Q$ and $D = [d_1, d_2, \ldots, d_n]^{\mathrm{T}}$, where $Q_{ij} =$
    $k_{\sqrt{2}\sigma}(x_i, x_j)$ and $d_i = \frac{1}{n}\sum_{j=1}^{n} k_\sigma(x_i, x_j)$
Step 3: Compute $\lfloor d_i \rfloor_m$, $i = 1, \ldots, n$, and indicate $D_B$ and $D_N$
Step 4: Compute $\lambda_i$, where $\lambda_i = \begin{cases} 0, & d_i \in D_B \\ (d_i - \lfloor d_i \rfloor_m)|\Omega_i^N|, & d_i \in D_N \end{cases}$
Step 5: Compute $D^* = D + R$, where $R_i = \lambda_i r_i$
Step 6: Solve $\hat{\omega} = \mathrm{SMO}(Q, D^*)$
```

Specially, the proposed model can be viewed as a sparse KDE based on a random weighted $L_1$ penalty. Then it can be also written as

$$\begin{aligned} \hat{\omega} &= \arg\min_{\omega} \mathrm{ISE}(\omega) - \beta\|P\omega\|_1 \\ &\text{s.t. } \omega^{\mathrm{T}}1 = 1 \text{ and } \omega_i \geq 0\ \forall i. \end{aligned} \tag{6}$$

Here $P = [0 \quad R]^{\mathrm{T}}$, where $0$ is a $(n - b_t) \times 1$ column vector of zeros, and $\beta > 0$ is the regularization parameter.

## 3.2 Analysis of RSKDE

In the RSDE, an unbiased estimate of $M(\omega)$ in Sect. 2 can be obtained as a $\omega_i$ weighted sum of KDE $d_i$ of each point $x_i$. However, the $d_i$ can be expressed as the inner product between a mapped test point and the mean of mapped training points in kernel feature space, $d_i = \frac{1}{n}\sum_{j=1}^{n} k_\sigma(x_i, x_j) = \left\langle \phi_\sigma(x_i), \frac{1}{n}\sum_{j=1}^{n} \phi_\sigma(x_j) \right\rangle$. As we know, the mean estimator $\hat{\theta} = \frac{1}{n}\sum_{j=1}^{n} \phi_\sigma(x_j)$ can be drastically influenced by outliers. The following proposition shows that the proposed algorithm improves performance of the density estimate.

**Proposition 1** *In contrast to the RSDE, a small increment of $d_i \in \Omega_N$ can make the proposed model more robust against outliers, and improve the quality of the density estimates.*

*Proof* The RSDE uses mean estimation for KDE, which is not robust against outliers in the data. In our case, the larger the value of $|\Omega_k|$, $k = 1, \ldots, t$, the more coherent feature points in $\Theta_k$. It implies that $\phi_\sigma(x_j) \in \Theta_k$ is more unlikely to be an outlier. To reduce the influence of possible outliers among the training data, we would like to set small weight values for outliers. Instead of giving the concrete implement algorithm, a feasible robust estimate for the sample mean is described below just for the proof.

$$\begin{aligned} \hat{\theta} = &\sum_{\phi_\sigma(x_j) \in \Theta_k} \alpha_0 \phi_\sigma(x_j) + \sum_{\phi_\sigma(x_j) \in \Theta_N - \Theta_k} \frac{1}{n} \phi_\sigma(x_j) \\ &+ \sum_{\phi_\sigma(x_j) \in \Theta_B} \alpha_1 \phi_\sigma(x_j) \end{aligned}$$

where $\alpha_0 \geq \frac{1}{n} \geq \alpha_1 > 0$, and $\alpha_0|\Theta_k| + \frac{1}{n}|\Theta_N - \Theta_k| + \alpha_1|\Theta_B| = 1$. It can be seen that the influence from $\phi_\sigma(x_i) \in \Theta_B$ is decreased. If $\phi_\sigma(x_i) \in \Theta_k$, then we have $d_i^* = \left\langle \phi_\sigma(x_i), \hat{\theta} \right\rangle = \left\langle \phi_\sigma(\mathbf{x}_i), \sum_{\phi_\sigma(\mathbf{x}_j) \in \Theta_k} \alpha_0 \phi_\sigma(\mathbf{x}_j) + \sum_{\phi_\sigma(\mathbf{x}_j) \in \Theta_N - \Theta_k} \frac{1}{n} \phi_\sigma(\mathbf{x}_j) + \sum_{\phi_\sigma(\mathbf{x}_j) \in \Theta_B} \alpha_1 \phi_\sigma(\mathbf{x}_j) \right\rangle > \left\langle \phi_\sigma(\mathbf{x}_i), \frac{1}{n}\sum_{j=1}^{n} \phi_\sigma(\mathbf{x}_j) \right\rangle$.

Hence, $d_i^* > d_i$. There exists a small enough $\lambda_i$, such that $d_i^* = d_i + \lambda_i r_i > d_i$. On the contrary, we suppose that $d_i^*$ is a little larger than $d_i = \left\langle \phi_\sigma(x_i), \frac{1}{n}\sum_{j=1}^{n} \phi_\sigma(x_j) \right\rangle$, then $d_i^* = \left\langle \phi_\sigma(x_i), \sum_{j=1}^{n} \alpha_j \phi_\sigma(x_j) \right\rangle$ satisfies $\alpha_i > \frac{1}{n}$. Therefore, a small increment of $d_i \in \Omega_N$ makes the proposed model more robust against outliers.

In addition, suppose that $\omega_1 = \mathrm{argmin}_\omega \mathrm{ISE}(\omega)$ and $\omega_2 = \mathrm{argmin}_\omega \mathrm{ISE}^*(\omega)$, the density estimation based on $\omega_2$ is more accurate than $\omega_1$, since $\mathrm{ISE}(\omega_1) \geq \mathrm{ISE}^*(\omega_1) \geq \mathrm{ISE}^*(\omega_2)$. □

The matrix $[D_B \quad D_N^*]^{\mathrm{T}}$ defined in (5) can be interpreted as the more robust estimation of $M(\omega)$. It can be written in the form $M^*(\omega) = \sum_{d_i \in \Omega_B} \omega_i d_i + \sum_{d_i \in \Omega_N} \omega_i (d_i + R_i)$. In this case, the error of estimation is bounded as follows: $|M(\omega) - M^*(\omega)| = \left|\sum_{d_i \in \Omega_N} \omega_i R_i\right| < \lambda_1$.

As mentioned previously, the RSDE involves Gaussian kernels of bandwidth $\sqrt{2}\sigma$ and $\sigma$, which occurs in $Q$ and $D$. The normalizing constants for these kernels are $(4\pi\sigma^2)^{-\ell/2}$ and $(2\pi\sigma^2)^{-\ell/2}$, respectively. As can be seen, the ratio between them is $2^{-\ell/2}$. If the dimension $\ell$ is large enough, the linear term $D$ dominates the quadratic term $Q$. It implies that, in high dimensional data, it is hard to find the coherent coefficients. In other words, the RSDE has already yielded a more sparse solution on most high dimensional data. There are no significant difference between the RSKDE and RSDE. This agrees with our intuition that the representation of signals is easier in lower dimensions. For high dimensional data, many of the dimensions are often irrelevant. These irrelevant dimensions can hide clusters in noisy data. It is common for all of the training data to be nearly equidistant from each other in very high dimensions [19]. Therefore, the sparsity for lower dimensional data is much more than the sparsity achieved in the case of higher dimensional data for similar quality of estimates.

## 4 Experimental results

We implement the proposed RSKDE in MATLAB based on the KDE Toolbox (written by Ihler and Mandel [20]) and evaluate the performance in density estimation. Because of the good performance of RSKDE, it is extensively validated on novelty detection and binary classification.

### 4.1 Density estimation

We experiment with one-dimensional data which is drawn from a heavily skewed distribution defined as $p_1(x) = \frac{1}{8}\sum_{i=0}^{7} g(x, \mu_i, \sigma_i)$, where $\sigma_i = (2/3)^i$ and $\mu_i = 3(\sigma_i - 1)$ [21]. Here, $g(x, \mu, \sigma)$ is a univariate Gaussian distribution with mean $\mu$ and variance $\sigma$. Data samples of $n$ are randomly drawn from the distribution to construct KDE. The width of the kernel is found by Rule of Thumb [22], and a separate test data set of 10,000 samples is used to calculate the $L_1$ error and $L_2$ error for the resulting estimate which are defined in [13]. The parameter $m$ is set to 4. For $n = 500$, a typical result is shown in Fig. 1b. As we can see, the nonzero weighting coefficients are not concentrated in regions of space with greater probability mass in contrast to Fig. 1a. In addition, there exist one or several points with larger weighting coefficients to represent high probability mass. Therefore, the RSKDE achieves a much sparser estimator than the RSDE estimator. Moreover, the resulting estimate is much closer to the true density. To demonstrate the effectiveness and robustness, we test our model with several recent methods: RSDE, KD-tree based density reduction method of Ihler et al. [20], sparse kernel density estimates (SKDE) with $L_0$ penalty [14]. The experiment is repeated 200 times for different sample sizes. The remaining data (percentage of sample size) are shown in Fig. 3a. The average $L_2$ error (mean ± SD) between the true density and respective density estimators against sample size are shown in Fig. 3b. From the results, it is clear that the proposed method provides a significant improvement both in sparsity and accuracy under the same experimental conditions.

To test the robust performance, we add uncorrelated outliers from a random distribution over [−4, 4]. For $n = 650$ (500 data samples are generated from the previous probability density function, and 150 outliers), a typical result is shown in Fig. 4. The $L_2$ error for the RSKDE is only slightly superior to the RSDE, but the RSKDE has the remarkable advantage for the sparsity. Only 20 nonzero weighting coefficients are need for the RSKDE, while 114 nonzero weighting coefficients are required for the RSDE.

To further compare the results of the proposed algorithm, the experiment is repeated 200 times with 500 fixed data samples and different numbers of outliers. The average $L_1$ error (mean ± SD), $L_2$ error (mean ± SD) and the number of nonzero weighting coefficients against sample size (data sample size + outlier size) are shown in Table 1. After adding outliers to the original data set, it is clear from the results that the RSKDE we have developed is always better than the RSDE both in sparsity and accuracy of the estimates. Moreover, the number of nonzero weighting coefficients provided by the proposed model remains fairly consistent, when the number of outliers is increased.

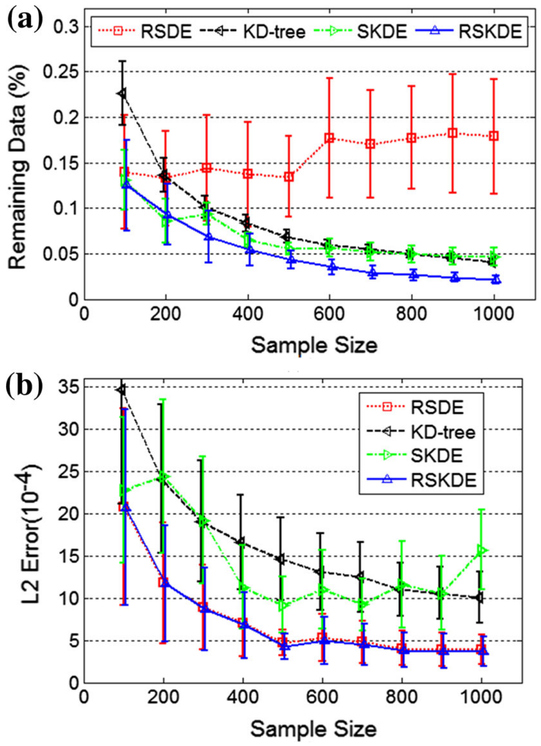

**Fig. 3** **a** Plot of the remaining data (percentage of sample size) for four related methods. **b** $L_2$ error between the true density and respective density estimators against sample size over 200 runs

### 4.2 Novelty detection

Novelty detection is the identification of new or unknown data that a machine learning system is not aware of during training. Novelty detection is one-class classification. The known data form one class, and a novelty-detection method tries to identify outliers that differ from the distribution of ordinary data. The RSKDE for novelty detection is tested on real-world data sets: Banana and Phoneme. Both data sets are available at http://sci2s.ugr.es/keel. The Banana dataset contains a total of 5,300 samples over two classes. The novelty detectors are trained on the first 400 samples in the first class. The remaining samples are used for testing. The Phoneme dataset has two classes and 5,404 samples. The aim of the dataset is to distinguish between nasal (class

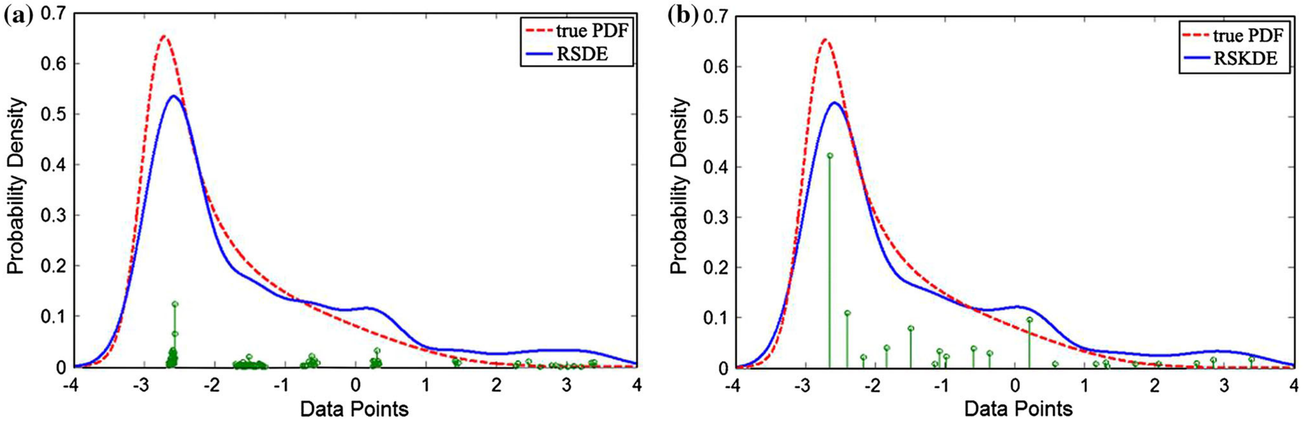

**Fig. 4** **a** The true density and the RSDE; each of the 114 nonzero weighting coefficients is placed at the appropriate sample data point. The corresponding $L_1$ error and $L_2$ error between them are 0.0296 and 0.00197. **b** The true density and the RSKDE; each of the 20 nonzero weighting coefficients is placed at the appropriate sample data point. The corresponding $L_1$ error and $L_2$ error between them are 0.0291 and 0.00192

**Table 1** $L_1$ error and $L_2$ error between the true density and respective density estimators against sample size over 200 runs

| Sample size | 500 + 50 | 500 + 100 | 500 + 150 | 500 + 200 | 500 + 250 |
|---|---|---|---|---|---|
| KDE kernel no. | 550 ± 0.0 | 600 ± 0.0 | 650 ± 0.0 | 700 ± 0.0 | 750 ± 0.0 |
| KDE $L_1$ error × $10^{-2}$ | 2.85 ± 0.409 | 3.66 ± 0.382 | 4.41 ± 0.436 | 5.07 ± 0.364 | 5.66 ± 0.418 |
| KDE $L_2$ error × $10^{-3}$ | 2.63 ± 0.718 | 4.04 ± 0.705 | 5.62 ± 0.879 | 7.05 ± 0.783 | 8.42 ± 0.894 |
| RSDE kernel no. | 80.5 ± 33.0 | 79.5 ± 32.3 | 76.3 ± 27.0 | 77.6 ± 29.5 | 82.2 ± 29.2 |
| RSDE $L_1$ error × $10^{-2}$ | 1.89 ± 0.394 | 2.59 ± 0.367 | 3.26 ± 0.422 | 3.92 ± 0.361 | 4.52 ± 0.425 |
| RSDE $L_2$ error × $10^{-3}$ | 0.756 ± 0.375 | 1.34 ± 0.421 | 2.23 ± 0.591 | 3.2 ± 0.625 | 4.2 ± 0.734 |
| KD-tree kernel no. | 39.0 ± 3.60 | 37.1 ± 3.15 | 34.7 ± 2.89 | 33.6 ± 2.54 | 32.0 ± 2.22 |
| KD-tree $L_1$ error × $10^{-2}$ | 2.85 ± 0.409 | 3.66 ± 0.381 | 4.41 ± 0.436 | 5.08 ± 0.363 | 5.67 ± 0.418 |
| KD-tree $L_2$ error × $10^{-3}$ | 2.63 ± 0.718 | 4.05 ± 0.705 | 5.62 ± 0.880 | 7.05 ± 0.782 | 8.43 ± 0.893 |
| GMM kernel no. | 30 ± 0.0 | 30 ± 0.0 | 30 ± 0.0 | 30 ± 0.0 | 30 ± 0.0 |
| GMM $L_1$ error × $10^{-2}$ | 4.81 ± 0.378 | 4.61 ± 0.314 | 5.44 ± 0.363 | 5.77 ± 0.305 | 5.85 ± 0.327 |
| GMM $L_2$ error × $10^{-3}$ | 7.92 ± 1.72 | 5.96 ± 1.20 | 8.45 ± 1.50 | 7.81 ± 1.22 | 7.75 ± 0.968 |
| SKDE kernel no. | 27.5 ± 3.71 | 23.5 ± 4.24 | 24.7 ± 3.21 | 18.5 ± 6.07 | 21.0 ± 4.62 |
| SKDE $L_1$ error × $10^{-2}$ | 2.07 ± 0.406 | 3.24 ± 0.350 | 3.52 ± 0.374 | 4.91 ± 0.462 | 5.79 ± 0.514 |
| SKDE $L_2$ error × $10^{-3}$ | 1.13 ± 0.658 | 2.89 ± 0.613 | 3.17 ± 0.770 | 5.76 ± 0.839 | 7.55 ± 0.910 |
| RSKDE kernel no. | 21.7 ± 5.39 | 19.9 ± 4.83 | 17.8 ± 3.98 | 16.6 ± 3.7 | 15.3 ± 3.27 |
| RSKDE $L_1$ error × $10^{-2}$ | 1.82 ± 0.397 | 2.53 ± 0.382 | 3.22 ± 0.436 | 3.88 ± 0.375 | 4.5 ± 0.444 |
| RSKDE $L_2$ error × $10^{-3}$ | 0.705 ± 0.355 | 1.29 ± 0.418 | 2.19 ± 0.587 | 3.17 ± 0.617 | 4.19 ± 0.739 |

0) and oral sounds (class 1). There are five features. The novelty detectors are trained on the first 730 samples in class 0. The remaining samples are used for testing.

The density estimator $\hat{f}(x;\omega,\sigma)$ obtained from the training set give us a quantitative measure of the degree of novelty for each test sample. This is used to reject samples where the estimate $\hat{f}(x;\omega,\sigma) < \rho$ for some threshold $\rho$ [23]. Thus, any sample where the likelihood $\hat{f}(x;\omega,\sigma)$ is below some threshold is considered to be novel. It implies that all test samples are classified into one of two classes: those which are similar to the training data, and those which are novel. Therefore, we adopt the standard definitions [24] used in binary classification to compare the results of RSKDE with existing algorithms. TP and TN stand for the number of true positives and true negatives, respectively. FP and FN represent, respectively, the number of misclassified positive and negative cases. In two-class problems, the accuracy rate on the positives, called sensitivity, is defined as TP/(TP + FN), whereas the accuracy rate on the negative class, also known as specificity, is TN/(TN + FP). Classification accuracy is (TP + TN)/$N$, where $N =$ TP + TN + FP + FP is the total number of

**Table 2** Performance of the kernel density estimation (KDE), reduced set density estimation (RSDE) [25], Gaussian mixture model (GMM) [26], *k*-nearest neighbor algorithm (*k*-NN), one-class support vector machines SVM [27] and the proposed RSKDE

| Dataset ($N_1$, $N_2$, $l$) | Method | Accuracy | Sensitivity | Specificity | No. points |
|---|---|---|---|---|---|
| Banana (400, 4,900, 2) | KDE | 0.800 | 0.902 | 0.649 | 400 |
| Banana (400, 4,900, 2) | RSDE | 0.807 | 0.909 | 0.656 | 100 |
| Banana (400, 4,900, 2) | GMM | 0.821 | 0.911 | 0.695 | 5 |
| Banana (400, 4,900, 2) | *k*-NN | 0.805 | 0.788 | 0.830 | 400 |
| Banana (400, 4,900, 2) | SVM | 0.818 | 0.796 | 0.849 | 226 |
| Banana (400, 4,900, 2) | RSKDE | 0.819 | 0.922 | 0.667 | 70 |
| Phoneme (730, 4,674, 5) | KDE | 0.725 | 0.910 | 0.364 | 730 |
| Phoneme (730, 4,674, 5) | RSDE | 0.719 | 0.727 | 0.702 | 191 |
| Phoneme (730, 4,674, 5) | GMM | 0.712 | 0.880 | 0.385 | 5 |
| Phoneme (730, 4,674, 5) | *k*-NN | 0.741 | 0.879 | 0.472 | 730 |
| Phoneme (730, 4,674, 5) | SVM | 0.676 | 0.550 | 0.922 | 527 |
| Phoneme (730, 4,674, 5) | RSKDE | 0.728 | 0.704 | 0.774 | 96 |

**Table 3** Performance of the six related methods

| Dataset ($N_1$, $N_2$, $l$) | Method | Mean accuracy | Mean sensitivity | Mean specificity | Mean no. points |
|---|---|---|---|---|---|
| Banana (1,000, 4,300, 2) | KDE | 0.894 | 0.938 | 0.840 | 1,000 |
| Banana (1,000, 4,300, 2) | RSDE | 0.896 | 0.940 | 0.842 | 274.1 |
| Banana (1,000, 4,300, 2) | GMM | 0.896 | 0.886 | 0.908 | 5 |
| Banana (1,000, 4,300, 2) | k-NN | 0.899 | 0.937 | 0.853 | 1,000 |
| Banana (1,000, 4,300, 2) | SVM | 0.864 | 0.873 | 0.855 | 840.5 |
| Banana (1,000, 4,300, 2) | RSKDE | 0.898 | 0.949 | 0.836 | 187.8 |
| Phoneme (1,000, 4,404, 5) | KDE | 0.821 | 0.811 | 0.846 | 1,000 |
| Phoneme (1,000, 4,404, 5) | RSDE | 0.798 | 0.780 | 0.841 | 266.3 |
| Phoneme (1,000, 4,404, 5) | GMM | 0.772 | 0.837 | 0.621 | 5 |
| Phoneme (1,000, 4,404, 5) | *k*-NN | 0.838 | 0.925 | 0.635 | 1,000 |
| Phoneme (1,000, 4,404, 5) | SVM | 0.788 | 0.880 | 0.573 | 524.5 |
| Phoneme (1,000, 4,404, 5) | RSKDE | 0.799 | 0.783 | 0.836 | 252.6 |

cases. Table 2 compares qualitatively RSKDE for novelty detection with other algorithms. Here, $N_1$ is the number of training data, and $N_2$ is the number of test data. The likelihood cross-validation is employed in selecting the kernel width for fair comparison. In the *k*-nearest neighbor algorithm, *k* is set to 3. The weighting coefficient $\omega$ of the RSKDE is obtained by optimizing (5) over training samples. We can see that in both datasets the RSKDE outperforms the KDE and RSDE.

### 4.3 Binary classification

This section further evaluates the RSKDE's performance for two-class classification problem. The experiments are carried out on the datasets that were used in novelty detection. The number of training samples is 1,000. The remaining samples are used for testing. There are ten randomly permuted partitions of each dataset into training and test sets. We first estimate the two conditional density functions $\hat{f}(x;\omega,\sigma|C_0)$ and $\hat{f}(x;\omega,\sigma|C_1)$ for class $C_0$ and $C_1$ from the training data, and then apply the Bayes' rule to the test data set and calculate the corresponding accuracy (ACC).

$$\left.\begin{array}{ll}\text{if}\ \hat{f}(x;\omega,\sigma|C_0) \geq \hat{f}(x;\omega,\sigma|C_1), & x \in C_0 \\ \text{else}, & x \in C_1\end{array}\right\} \tag{7}$$

During training, the kernel width $\sigma$ is tuned by likelihood cross-validation, and the weighting coefficient $\omega$ is obtained by optimizing (5) over training samples. Table 3 compares the performance of the six related methods. As can be seen, the test mean accuracy for the RSKDE is 0.898 which is only slightly superior to 0.894 of the KDE, but the RSKDE has the remarkable advantage for the test complexity. Only 187 mean samples in the reduced set are needed for the RSKDE classifier while all 1,000 training samples are required for KDE classifier. Meanwhile, on average the RSKDE classifier reduces test computational costs by ~80 %. For high dimensional data, results show no significant difference between the RSKDE and RSDE.

## 5 Conclusion

In this paper, a novel robust sparse kernel density estimation based on the RSDE is presented. Instead of sparse representation by regularization term, the proposed model induces randomness to the plug-in estimation of the RSDE and yield a more sparse representation in the weighting coefficients. By means of SMO, the randomness can make those nonzero and small weighting coefficients get together into one or several points with larger weighting coefficients. The proposed model shows good performance both in sparsity and accuracy of the estimates for the low dimensional data. Numerical experiments show promising results.

**Acknowledgment** This work was supported by a National Key Basic Research Project of China (973 Program No. 2012CB316400) and NSFC (No. 60872069).

## References

1. Tsai A, Yezzi A, Wells W, Tempany C, Tucker D, Fan A, Grimson E, Willsky A (2003) A shape-based approach to the segmentation of medical imagery using level sets. IEEE Trans Med Imaging 22(2):137–154
2. Leventon M, Grimson W, Faugeras O (2000) Statistical shape influence in geodesic active contours. IEEE Int Conf Comput Vis Pattern Recogn 1:316–323
3. Rousson M, Cremers D (2005) Efficient kernel density estimation of shape and intensity priors for level set segmentation. Int Conf Med Image Comput Comput Assist Interv 3750:757–764
4. Comaniciu D, Meer P (2002) Mean shift: a robust approach toward feature space analysis. IEEE Trans Pattern Anal Machine Intell 24:603–619
5. Elgammal A, Duraiswami R, Harwood D, Davis L (2002) Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc IEEE 90:1151–1163
6. Han B, Comaniciu D, Zhu Y, Davis LS (2008) Sequential kernel density approximation and its application to real-time visual tracking. IEEE Trans Pattern Anal Machine Intell 30(7):1186–1197
7. Cremers D, Osher S, Soatto S (2006) Kernel density estimation and intrinsic alignment for shape priors in level set segmentation. Int J Comput Vis 69(3):335–351
8. Kim J, Scott CD (2010) $L_2$ kernel classification. IEEE Trans Pattern Anal Machine Intell 32(10):1822–1831
9. Silverman BW (1982) Kernel density estimation using the fast Fourier transform. Appl Stat 31:93–99
10. Yang C, Duraiswami R, Gumerov N, Davis L (2003) Improved fast gauss transform and efficient kernel density estimation. IEEE Int Conf Comput vis 1:664–671
11. Vapnik V, Mukherjee S (1999) Support vector method for multivariate density estimation. In: Proceedings of NIPS, pp 659–665
12. Girolami M, He C (2003) Probability density estimation from optimally condensed data samples. IEEE Trans Pattern Anal Machine Intell 25(10):1253–1264
13. Chen S, Hong X, Harris CJ (2008) An orthogonal forward regression technique for sparse kernel density estimation. Neurocomputing 71(4):931–943
14. Gopalakrishnan B, Bellala G, Devadas G, Sricharan K (2008) EECS 545 machine learning-sparse kernel density estimates. http://www-personal.umich.edu/~gowtham/. Accessed 12 Apr 2013
15. Hong X, Chen S, Harris CJ (2010) Sparse kernel density estimation technique based on zero-norm constraint. In: Proceeding of the IJCNN, pp 3782–3787
16. Schölkopf B, Platt J, Shawe-Taylor J, Smola A, Williamson R (2001) Estimating the support of a high-dimensional distribution. Neural Comput 13:1443–1471
17. Schölkopf B, Smola AJ (2002) Learning with kernels. MIT Press, Cambridge
18. Kim J, Scott CD (2008) Robust kernel density estimation. ICASSP, pp 3381–3384
19. Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. SIGKDD Explor 6(1):90–105
20. Ihler A, Mandel M (2003) Kernel density estimation toolbox for MATLAB. http://www.ics.uci.edu/~ihler/code. Accessed 12 Apr 2013
21. Sain S (1994) Adaptive kernel density estimation. PhD thesis. Rice University, Houston
22. Silverman BW (1986) Density estimation for statistics and data analysis. Chapman and Hall, London
23. Bishop CM (1994) Novelty detection and neural network validation. IEEE Proc Vis Image Signal Process 141(4):217–222
24. Metz C (1978) Basic principles of ROC analysis. Semin Nucl Med 8(4):283–298
25. Chao H, Girolami M (2004) Novelty detection employing an $L_2$ optimal nonparametric density estimator. Pattern Recogn Lett 25(12):1389–1397
26. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley-interscience, New York
27. Chang CC, Lin CJ (2006) LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/~cjlin/libsvm/. Accessed 12 Apr 2013