Probability Density Estimation from Optimally Condensed Data Samples

Mark Girolami and Chao He

Abstract—The requirement to reduce the computational cost of evaluating a point probability density estimate when employing a Parzen window estimator is a well-known problem. This paper presents the Reduced Set Density Estimator that provides a kernel-based density estimator which employs a small percentage of the available data sample and is optimal in the L_2 sense. While only requiring $\mathcal{O}(N^2)$ optimization routines to estimate the required kernel weighting coefficients, the proposed method provides similar levels of performance accuracy and sparseness of representation as Support Vector Machine density estimation, which requires $\mathcal{O}(N^3)$ optimization routines, and which has previously been shown to consistently outperform Gaussian Mixture Models. It is also demonstrated that the proposed density estimator consistently provides superior density estimates for similar levels of data reduction to that provided by the recently proposed Density-Based Multiscale Data Condensation algorithm and, in addition, has comparable computational scaling. The additional advantage of the proposed method is that no extra free parameters are introduced such as regularization, bin width, or condensation ratios, making this method a very simple and straightforward approach to providing a reduced set density estimator with comparable accuracy to that of the full sample Parzen density estimator.

Index Terms—Kernel density estimation, Parzen window, data condensation, sparse representation.

1 Introduction

The estimation of the probability density function (PDF) of a continuous distribution from a representative sample drawn from the underlying density is a problem of fundamental importance to all aspects of machine learning and pattern recognition; see, for example, [3], [29], [33]. When it is reasonable to assume, a priori, a particular functional form for the PDF, then the problem reduces to the estimation of the required functional parameters. Finite mixture models [17] are a very powerful approach to estimating arbitrary density functions and are routinely employed in many practical applications. One can consider a finite mixture model as providing a condensed representation of the data sample in terms of the sufficient statistics of each of the mixture components and their respective mixing weights.

The kernel density estimator, also commonly referred to as the Parzen window estimator [20], can be viewed as the limiting form of a mixture model where the number of mixture components will equal the number of points in the data sample. Unlike parametric or finite-mixture approaches to density estimation where only sufficient statistics and mixing weights are required in estimation, Parzen[1] density estimates employ the full data sample in defining density estimates for subsequent observations. So, while large sample sizes ensure reliable density estimates, they bring with them a computational cost for testing which scales directly with the sample size. Herein lies the main practical difficulty with employing kernel-based Parzen window density estimators.

This paper considers the case where data scarcity is not an application constraint and that the continuous distributional characteristics of the data suggest the existence of a well-formed density function which requires to be estimated. Such situations are quite the norm in the majority of practical applications such as continuous monitoring of the condition of a machine or biomedical process and computer vision, e.g., [4], [22]—indeed, the reverse "problem" is often experienced in many situations where there is an overwhelming amount of data logged [18]. In situations where the volume of data to be processed is large, a semiparametric mixture model can provide a condensed representation of the reference data sample, in the form of the estimated model parameters. On the other hand, the Parzen window density estimator requires the full reference set for estimation [11], which in such practical circumstances can be prohibitively expensive for online testing purposes.

This paper addresses the above problem by providing a Parzen window density estimator which employs a reduced set of the available data sample. The proposed *Reduced Set Density Estimator* (RSDE) is optimal in the L_2 sense in that the integrated squared error between the unknown true density and the RSDE is minimized in devising the estimator. The required optimization turns out to be a straightforward quadratic optimization with simple positivity and equality constraints and, thus, suitable forms of Multiplicative Updating [27] or Sequential Minimal Optimisation, as introduced in [30], can be employed, which ensures at most quadratic scaling in the original sample size. This is a significant improvement over the cubic

[1] Both the terms "kernel" and "Parzen" will be used interchangeably in the text and will refer to the same form of nonparametric density estimator.

• *The authors are with the Applied Computational Intelligence Research Unit, School of Information and Communication Technologies, University of Paisley, High Street, Paisley, PA1 2BE, Scotland, UK.*
E-mail: {mark.girolami, chao.he}@paisley.ac.uk.

Manuscript received 30 Aug. 2002; revised 21 Mar. 2003; accepted 25 Mar. 2003.
Recommended for acceptance by Y. Amit.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117229.

scaling optimization required of the Support Vector Method of density estimation proposed in [19], [34]. The additional advantage of the proposed method is that, apart from the weighting coefficients, no additional free parameters are introduced into the representation such as regularization terms [35], bin widths [9], [25], or number of nearest neighbors [18]. The RSDE is shown to have similar convergence rates as the Parzen window estimator and performs, in terms of accuracy, similarly to the SVM density estimator [19], while requiring a much less costly optimization, and consistently outperforms the multiscale data condensation method [18] at specified data reduction rates when used for density estimation.

The following section now provides a brief review of methods which have been proposed in reducing the computational cost of density estimation using a kernel (Parzen window) density estimator.

2 Computation Reduction Methods for Kernel Density Estimation

The Parzen window form of nonparametric probability density estimation [20] is particularly attractive when no a priori information is available to guide the choice of the precise form of density with which to fit the data, for example, the number of components in a mixture model. Indeed, iterative methods have been proposed that employ a Parzen density estimator as a reference density for the purpose of fitting a finite mixture model when the number of components is unknown [21], [26]. A probability density estimate $\hat{p}(\mathbf{x};\boldsymbol{\theta})$ can be obtained from the finite data sample $\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \in \mathcal{R}^d$, drawn from the density $p(\mathbf{x})$ by employing the isotropic product form of the univariate Parzen window density estimator [11], [28]

$$\hat{p}(\mathbf{x};h) = \frac{1}{Nh^d}\sum_{n=1}^{N}\mathcal{K}\Big(\frac{\mathbf{x}-\mathbf{x}_n}{h}\Big), \tag{1}$$

where the well-known constraints on the window (also referred to as the weighting or kernel) function hold, i.e., it should also be a density function, see [11] for a comprehensive review. However, as already stated, the main disadvantage of such an approach is the high-computational requirements when large data samples are available as the estimation of the density at one point is an $\text{order-}N$ type problem.

Two distinct approaches to resolving this practical problem of computational load have been adopted. The first concentrates on providing an approximation to the kernel function which decouples the point under consideration from the points of the sample in such a way that the summation over the sample can be performed separately in a manner akin to orthogonal series density estimators [11]. The second approach focuses on reducing the required number of computations by reducing the effective size of the sample.

2.1 Approximate Kernel Decompositions

The notion of multipole expansions of potential functions is exploited in [15] to provide a reduced cost kernel density estimator. In [15], it is noted that, if it is possible to identify two sets of functions $\Phi_l(\mathbf{x})$ and $\Psi_l(\mathbf{x})$ such that the following expansion holds:

$$\mathcal{K}\Big(\frac{\mathbf{x}-\mathbf{x}_n}{h}\Big) = \sum_{l=1}^{\infty}\lambda_l\Phi_l(\mathbf{x})\Psi_l(\mathbf{x}_n). \tag{2}$$

The summation in (1) can be approximated by truncating the inner-product summation defining the kernel at M terms such that

$$\sum_{n=1}^{N}\mathcal{K}\Big(\frac{\mathbf{x}-\mathbf{x}_n}{h}\Big) = \sum_{n=1}^{N}\sum_{l=1}^{\infty}\lambda_l\Phi_l(\mathbf{x})\Psi_l(\mathbf{x}_n) \approx \sum_{l=1}^{M}\Phi_l(\mathbf{x})a_l,$$

where the M terms $a_l = \sum_{n=1}^{N}\lambda_l\Psi_l(\mathbf{x}_n)$ can be precomputed and stored so that a point density estimate will scale as $\mathcal{O}(M)$ rather than $\mathcal{O}(N)$, which clearly denotes a computational saving when $M << N$. However, there is no longer any guarantee that point estimates will necessarily be positive using this approach; Izenman [11] discusses such truncated orthogonal series estimators in detail, and Girolami [7] points out the relationship between such estimators and kernel principal component analysis [31].

2.2 Data Reduction Methods

A number of approaches have been taken in reducing the effective number of computations required in giving a point estimate of the density. In [28], the Fourier transform is used to reduce the effective number of computations required, while in [25], the data sample is prebinned and the kernel density estimator employs the bin centers as the "sample" points which are each weighted by the normalized bin-counts. Somewhat recently, the multivariate form of the binned kernel density estimator has been analyzed in [9]. However, now the bin width and also possible binning strategies (equal width bins or variable spacing) have to be selected for each dimension in the multivariate case.

Rather than binning the sample data, an alternative strategy is to cluster the sample and employ the cluster centres as the reduced data set. In [12], a clustering-based branch and bound approach is adopted, while in [2], clustering is employed in identifying a set of reference vectors to be employed in a Parzen-window classifier. In [10], the Self-Organizing Map [14] is used to provide the reference vectors for the density estimators. The main detractor of employing clustering-based data reduction methods is that a nonlinear optimization is required for the data partitioning and, as such, the solution is dependent on initial conditions, so the relative simplicity of the nonparametric density estimator is lost.

In [18], a data reduction method is proposed which employs hyperdiscs of varying radii which are dependent on the density of the data in the region being considered. This provides a very elegant density dependent data reduction method, in other words, a multiscale approach to data reduction is employed so that larger numbers of points will be removed from regions of high density. This has the additional benefit that the algorithm is deterministic based on the value of the free parameter k the number of "nearest neighbors" which determines the rate of data reduction. The value of k can, of course, be selected to minimize an error criterion between the estimate based on the reduced sample and the full sample, the algorithm has at most $\mathcal{O}(kN^2)$ scaling, where N is the number of points in the full sample.

2.3 Data Reduction via Sparse Functional Approximations

In [5], [6], a computationally costly search-based approach is adopted in approximating an entropic distance between the density estimate based on a subset of the available data sample and that based on the full sample. Support vector regression [33] was originally proposed in [35] as a means of providing a sparse Parzen density estimator, i.e., many of the points in the sample are not used in the density estimate. The trade off between sparsity and accuracy is controlled by the regularization term which requires to be selected in addition to the width of the kernel.

In [19], [34], [35], the support vector approach to density estimation has been proposed as a means of solving the ill-posed linear operator problem $\int_{-\infty}^{x} p(t)dt = F(x)$, where $p(t)$ denotes the PDF and the distribution function at the point x is given as $F(x)$. The support vector density estimator $\hat{p}(\mathbf{x}) = \sum_{i=1}^{N} \beta_i \mathcal{K}_h(\mathbf{x}, \mathbf{x}_i)$, where $\mathcal{K}_h(\mathbf{x}, \mathbf{x}_i) \equiv \frac{1}{h^d}\mathcal{K}\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)$, can be considered as a generalization of the Parzen density estimator, where now, each β_i act as the nonuniform weighting coefficients. The following constrained quadratic optimization is required to define the weighting coefficients [19].

$$\begin{aligned} &\arg\min_{\boldsymbol{\beta}} \boldsymbol{\beta}^{\mathrm{T}}\mathbf{K}\boldsymbol{\beta} \\ &\texttt{s.t} \quad |\mathbf{f} - \mathbf{E}\boldsymbol{\beta}| \leq \boldsymbol{\epsilon}, \text{and} \;\; \boldsymbol{\beta}^{\mathrm{T}}\mathbf{1} = 1 \;\; \beta_i \geq 0 \;\; \forall \;\; i, \end{aligned} \tag{3}$$

where $\mathbf{K}$ is the $N \times N$ matrix whose elements are all $\mathcal{K}_h(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{1}$ is the $N \times 1$ vector of ones, and $\mathbf{f}$ is the $N \times 1$ vector whose $i\mathrm{th}$ element $\hat{F}_N(\mathbf{x}_i)$ is the empirical distribution function of the random vector $\mathbf{x}_i$ computed as the product of the empirical distribution of each vector element. The $N \times N$ matrix $\mathbf{E}$ whose $i, j\mathrm{th}$ element corresponds to $\prod_{k=1}^{d} \int_{-\infty}^{x_j^k} \mathcal{K}_h(x_i^k, t)dt$ and $\boldsymbol{\epsilon}$, the $N \times 1$ vector whose elements are all ϵ_N completes the definitions required for the above optimization. The ϵ_N denotes the accuracy value of the Kolmogorv-Smirnov statistic (the absolute deviation between the empirical distribution function and the distribution function derived from the model) [19], which the solution is desired to achieve and this is used in selecting the bandwidth of the kernel [19]. The constraints required for this optimization are dense and there is no dual form [33] which reduces the complexity of the constraints, as such the solution of (3) requires generic quadratic optimization packages which typically scale as $\mathcal{O}(N^3)$.

The support vector approach to density estimation provides a sparse representation in the weighting coefficients and, therefore, reduced computational cost when testing, it has also been shown to provide excellent results in testing [19], [34]. However, for large sample sizes, it is essential to obtain an optimization which will have scaling better than $\mathcal{O}(N^3)$ as in [19], and does not require the setting of any additional free parameters which control the regularization of the solution as in [25], [35]. The following section presents the RSDE which enjoys at most $\mathcal{O}(N^2)$ scaling to estimate the weighting coefficients and only has one free parameter to set, the width of the kernel as in a standard Parzen estimator.

3 Reduced Set Density Estimator

3.1 Divergence and Distance-Based Density Estimation

Based on a data sample $\mathcal{S} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\} \in \mathcal{R}^d$, the general form of a kernel density estimator is given as $\hat{p}(\mathbf{x}; h, \gamma) = \sum_{n=1}^{N} \gamma_n \mathcal{K}_h(\mathbf{x}, \mathbf{x}_n)$. For a given kernel with width h, the maximum likelihood estimator (MLE) criterion [17] can be employed to estimate the weighting coefficients such that

$$\hat{\boldsymbol{\gamma}}_{MLE} = \arg\max_{\boldsymbol{\gamma}} \frac{1}{N} \sum_{m=1}^{N} \log \sum_{n=1}^{N} \gamma_n \mathcal{K}_h(\mathbf{x}_m, \mathbf{x}_n)$$

subject to the constraints $\sum_n \gamma_n = 1$ and $\gamma_n \geq 0 \;\; \forall \;\; n$. It is a straightforward matter to show that the above MLE criterion yields values for the coefficients such that $\gamma_n = \frac{1}{N} \; \forall \, \mathbf{x}_n \in \mathcal{S}$ and, as such, the Parzen window density estimator can be seen to be a maximum likelihood kernel density estimator. The MLE criterion can be considered as a *Divergence*-based criterion in that it is a plug-in estimate of the negative cross-entropy or divergence between the true density and the estimate, .i.e.,

$$\int_{\mathcal{R}^d} p(\mathbf{x}) \log \hat{p}(\mathbf{x}; h, \boldsymbol{\gamma}) d(\mathbf{x}) \approx \frac{1}{N} \sum_{n=1}^{N} \log \hat{p}(\mathbf{x}_n; h, \boldsymbol{\gamma}).$$

Alternative *Distance*-based criteria have been considered for the purposes of density estimation when employing mixture models [24]. In particular, the L_2 criterion based on the Integrated Squared Error (ISE) has been investigated as a robust error criterion which will be less influenced by the presence of outliers in the sample and model mismatch than the MLE criterion [24]. The fitting of finite mixture models employing the L_2 criterion has been investigated in [26], where the sufficient statistics of each mixture component (Gaussians) are estimated by the nonlinear optimization of the ISE.

The ISE is a measure of the global accuracy of a density estimate [11], [29], which converges to the mean squared error asymptotically. For a density estimate with parameters $\boldsymbol{\theta}$ denoted as $\hat{p}(\mathbf{x}; \boldsymbol{\theta})$, the argument which provides the minimum ISE is as follows:

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} I(\boldsymbol{\theta}) \\ &= \arg\min_{\boldsymbol{\theta}} \int_{\mathcal{R}^d} |p(\mathbf{x}) - \hat{p}(\mathbf{x}; \boldsymbol{\theta})|^2 d\mathbf{x} \\ &= \arg\min_{\boldsymbol{\theta}} \int_{\mathcal{R}^d} \hat{p}^{\,2}(\mathbf{x}; \boldsymbol{\theta}) d\mathbf{x} - 2E_{p(\mathbf{x})}\{\hat{p}(\mathbf{x}; \boldsymbol{\theta})\}, \end{aligned} \tag{4}$$

where the term $\int_{\mathcal{R}^d} p^{\,2}(\mathbf{x})d\mathbf{x}$ has been dropped from the above due to its independence of the $\boldsymbol{\theta}$ parameters and $E_{p(\mathbf{x})}\{\cdot\}$ denotes expectation with respect to $p(\mathbf{x})$. We now show that direct minimisation of a plug-in estimate of the ISE for a general kernel density estimator yields a sparse representation in the weighting coefficients.

3.2 Plug-In Estimation of Weighting Coefficients

An unbiased estimate of the right-hand expectation in the above expression for a kernel density estimator can be written as

$$\begin{aligned} E_{p(\mathbf{x})}\{\hat{p}(\mathbf{x};\boldsymbol{\theta})\} &= \sum_{i=1}^{N} \gamma_i E_{p(\mathbf{x})}\{\mathcal{K}_h(\mathbf{x}_i,\mathbf{x})\} \\ &\simeq \sum_{i=1}^{N} \gamma_i \frac{1}{N} \sum_{j=1}^{N} \mathcal{K}_h(\mathbf{x}_i,\mathbf{x}_j) \\ &= \sum_{i=1}^{N} \gamma_i \hat{p}_h(\mathbf{x}_i), \end{aligned}$$

where the full Parzen density estimator for the point $\mathbf{x}_i$ is denoted as $\hat{p}_h(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{N} \mathcal{K}_h(\mathbf{x}_i,\mathbf{x}_j)$. A little investigation of the right-hand term shows that it is *sparsity* inducing, in other words, its presence in the required optimization of ISE will cause many of the γ_i terms to be driven to zero. This is due to the simple observation that maximizing a convex combination of positive numbers is obtained by assigning a unit weight to the largest. We observe that if the density has a dominant mode, then the optimization of the right-hand term of the plug-in estimate of ISE will set the weighting coefficient value of the sample point closest to the mode to unity and all others to zero. In the general case, if there is a unique maximum in the sample of the estimate $\hat{p}_h(\mathbf{x}_i)$, then it alone will be assigned unit weighting. So, it can be seen that minimization of the estimated ISE, due to the right-hand term, will provide a sparse representation placing finite weighting on a reduced set of points from regions of high density in the sample. We now consider the remaining quadratic term.

The left-hand term $\int_{\mathcal{R}^d} \hat{p}^{\,2}(\mathbf{x};\boldsymbol{\theta})d\mathbf{x}$ can be computed exactly as

$$\sum_{i,j=1}^{N} \gamma_i\gamma_j \int_{\mathcal{R}^d} \mathcal{K}_h(\mathbf{x},\mathbf{x}_i)\mathcal{K}_h(\mathbf{x},\mathbf{x}_j)d\mathbf{x},$$

denoting $\int_{\mathcal{R}^d} \mathcal{K}_h(\mathbf{x},\mathbf{x}_i)\mathcal{K}_h(\mathbf{x},\mathbf{x}_j)d\mathbf{x}$ by $\mathcal{C}(\mathbf{x}_i,\mathbf{x}_j)$, then the quadratic left-hand term can be written as $\sum_{i,j=1}^{N}\gamma_i \gamma_j \mathcal{C}(\mathbf{x}_i,\mathbf{x}_j)$. A similar constrained quadratic form has been utilized previously to obtain a minimum volume description of a data sample [32] or to obtain a sample estimate of the distribution support [30], where it has been observed empirically that the extremal points in the sample are given a finite weighting coefficient. This can be viewed as placing finite weight to points in regions of low density, which is in contrast to the effect which the linear term in the ISE has, that is placing finite weight to points in regions of high density.

Combining both terms then for a fixed bandwidth window the optimization of a plug-in estimate of ISE (4) over γ satisfying the requirements of a density function viz. $\sum_{n=1}^{N}\gamma_n = 1$ and $\gamma_n \geq 0 \;\forall\; n$ is

$$\arg\min_{\boldsymbol{\gamma}} \sum_{i=1}^{N}\sum_{j=1}^{N} \gamma_i\gamma_j\mathcal{C}(\mathbf{x}_i,\mathbf{x}_j) - 2\sum_{i=1}^{N}\gamma_i\hat{p}_h(\mathbf{x}_i).$$

As discussed, a by product of the summation and positivity constraints on the weighting coefficients is that many of the γ terms associated with points having low density estimate $\hat{p}_h(\mathbf{x})$ will be set to zero in the above optimization, thus effectively selecting a reduced set from high density regions in the data sample.

So, the minimization of a plug-in estimate of the ISE of the reduced set density estimator can be written as a constrained quadratic optimization which, in familiar matrix form,[2] is

$$\arg\min_{\boldsymbol{\gamma}} \frac{1}{2}\boldsymbol{\gamma}^{\mathrm{T}}\mathbf{C}\boldsymbol{\gamma} - \boldsymbol{\gamma}^{\mathrm{T}}\mathbf{p} \qquad \text{subject to } \boldsymbol{\gamma}^{\mathrm{T}}\mathbf{1} = 1 \;\texttt{and}\; \gamma_i \geq 0 \;\forall\; i, \tag{5}$$

where the $N \times N$ matrices with elements $\mathcal{C}(\mathbf{x}_i,\mathbf{x}_j) = \int_{\mathcal{R}^d} \mathcal{K}_h(\mathbf{x},\mathbf{x}_i)\mathcal{K}_h(\mathbf{x},\mathbf{x}_j)d\mathbf{x}$ and $\mathcal{K}_h(\mathbf{x}_i,\mathbf{x}_j)$ are defined as $\mathbf{C}$ and $\mathbf{K}$, respectively. The $N \times 1$ vector of Parzen density estimates of each point in the sample $\hat{p}_h(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{N}\mathcal{K}_h(\mathbf{x}_i,\mathbf{x}_j)$ is defined as $\mathbf{p} = \mathbf{K}\mathbf{1}_N$, where $\mathbf{1}_N$ is the $N \times 1$ vector whose elements are all $\frac{1}{N}$.

As one specific example,[3] we can employ an isotropic Gaussian window at a point $\mathbf{x}$ with common width (variance) h and center $\mathbf{x}_i$ denoted as $\mathcal{G}_h(\mathbf{x},\mathbf{x}_i)$, then the individual terms of the matrices $\mathbf{K}$ and $\mathbf{C}$ have the specific form of $\mathcal{K}_h(\mathbf{x}_i,\mathbf{x}_j) = \mathcal{G}_h(\mathbf{x}_i,\mathbf{x}_j)$ and $\mathcal{C}(\mathbf{x}_i,\mathbf{x}_j) = \int_{\mathcal{R}^d}\mathcal{G}_h(\mathbf{x},\mathbf{x}_i)\mathcal{G}_h(\mathbf{x},\mathbf{x}_j)d\mathbf{x} = \mathcal{G}_{2h}(\mathbf{x}_i,\mathbf{x}_j)$, and so (5) can be written simply as

$$\arg\min_{\boldsymbol{\gamma}} \frac{1}{2}\sum_{i,j=1}^{N}\gamma_i\gamma_j\mathcal{G}_{2h}(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{N}\gamma_i\hat{p}_h(\mathbf{x}_i), \tag{6}$$

where $\hat{p}_h(\mathbf{x}_i) = \frac{1}{N}\sum_{j=1}^{N}\mathcal{G}_h(\mathbf{x}_i,\mathbf{x}_j)$. Note that the only free parameter (apart from the weighting coefficients) which requires to be set is the window width; there are no regularization or additional parameters which require to be determined. In addition, the constraints on the optimization are simpler than those required for the SVM density estimator (3), thus enabling a possibly faster means of optimization. Unlike the binned Parzen density estimator [25] or the data condensation approach [18], the problematic choice of bin width (binning strategy), or effective disc width selection is not required. Examining the form of (6), an intuitive insight into how the data reduction mechanism operates can be obtained. The minimum value of ISE will be penalized by contributions of large interpoint distances in the window function $\mathcal{G}_h(\cdot,\cdot)$ so the empirical expected value of the right-hand term will be maximized by selecting a small number of points (due to the summation constraint) in regions of high-density (low average interpoint distance). The left-hand term alone will cause the selection of points with high interpoint distances, as defined by the metric associated with the left-hand convolution operator, therefore, the overall effect will be that points in regions of high-density (as defined by the specific width of the window function) will be selected to provide a smoothed density estimate.

3.3 Optimization

As the quadratic program specified by (5) only has simple positivity and equality constraints, then a number of alternative optimization strategies are now available. A standard trick of introducing a dummy variable and applying the soft-max [3] function such that

[2] During the review of this paper, it was pointed out that the above formulation was proposed in the unpublished thesis of Kim [13].

[3] Other kernels, such as the finite-support Bartlett-Epanenchnikov kernel, can be easily numerically integrated over the range of the sample to obtain the $\mathcal{C}(\cdot,\cdot)$ terms.

$$\gamma_i = \frac{exp(\alpha_i)}{\sum_{n=1}^{N} exp(\alpha_n)},$$

converts the required constrained quadratic optimization (5) to an unconstrained nonlinear optimization over the dummy variables and conjugate gradients [3] provide a linear $\mathcal{O}(N)$ scaling optimization. However, moving from a linear to nonlinear optimization is not particularly appealing due to the inherent initialization dependent variability of the solutions. Somewhat recently a multiplicative updating method for the nonnegative quadratic programming of support vector machines [33] has been proposed in [27]. It is a straightforward matter to adopt multiplicative updating as developed in [27], specifically for the required optimization of (5).

3.3.1 Multiplicative Updating of the Weighting Coefficients

Denote the estimate of $\boldsymbol{\gamma}$ at iteration t of an iterative optimization procedure as $\boldsymbol{\gamma}^t$, then, as detailed in [27], an auxiliary function $G(\boldsymbol{\gamma}^{t+1}, \boldsymbol{\gamma}^t)$ can be formed such that $I(\boldsymbol{\gamma}^{t+1}) \leq G(\boldsymbol{\gamma}^{t+1}, \boldsymbol{\gamma}^t) \leq G(\boldsymbol{\gamma}^t, \boldsymbol{\gamma}^t) = I(\boldsymbol{\gamma}^t)$. The iterative minimization of the auxiliary function $G(\cdot,\cdot)$ then guarantees a series of estimates for $\boldsymbol{\gamma}^t$ which monotonically minimize the original function $I(\cdot)$, this approach was originally taken in the development of the Expecation Maximization algorithm [3]. As the matrix and vector components of (5) are strictly positive, i.e., denoting C_{ij} as the ijth element of $\mathbf{C}$ and p_i as the ith element of $\mathbf{p}$, then a simplified version of the auxiliary function in [27] can be employed for our purposes and so $G(\boldsymbol{\gamma}^{t+1}, \boldsymbol{\gamma}^t)$ is given as the following expression:

$$\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} C_{ij}\gamma_j^t \frac{(\gamma_i^{t+1})^2}{\gamma_i^t} - \sum_{i=1}^{N} p_i \gamma_i^{t+1}. \tag{7}$$

If the equality constraint requires to be satisfied, then the Lagrangian $G(\boldsymbol{\gamma}^{t+1}, \boldsymbol{\gamma}^t) - \varrho(\sum_{i=1}^{N} \gamma_i^{t+1} - 1)$ is formed, where ϱ is the required multiplier. It is a straightforward matter to show that the following monotonically convergent iterative routine for the estimation of $\boldsymbol{\gamma}$ follows:

$$\gamma_i^{t+1} = a_i^t(p_i + \varrho^t), \tag{8}$$

where $a_i^t = \gamma_i^t(\sum_{j=1}^{N} C_{ij}\gamma_j^t)^{-1}$ and $\varrho^t = (\sum_{n=1}^{N} a_n^t)^{-1}(1 - \sum_{m=1}^{N} p_m a_m^t)$. Note that each iteration requires a matrix-vector multiplication and element-wise division so the complexity per iteration is $\mathcal{O}(N^2)$. This is a useful routine for the estimation of the required weighting coefficient, however, in terms of overall speed of convergence, it has been found in our experiments that a form of the Sequential Minimal Optimization (SMO) as presented in [30] suitable for solving (5) is superior to multiplicative updating.

3.3.2 Sequential Minimal Optimization for RSDE

As detailed in [30], SMO can achieve overall $\mathcal{O}(N^2)$ scaling as opposed to $\mathcal{O}(N^3)$ scaling achievable for the standard quadratic optimization packages. In the following experiments, an appropriate variant of SMO to solve (5) is employed and this is detailed below. The updates for (5) are almost identical to those of [30] apart from the one additional term in (5), which requires to be incorporated. For completeness, the derivation is included here and follows [30].

To fulfil the summation constraint, we resort to optimizing over pairs of variables as in [30]. The SMO elementary optimization step for optimizing γ_1 and γ_2 with all other variables fixed follows. The general quadratic optimization problem

$$\min\ \frac{1}{2}\sum_{ij}\gamma_i\gamma_j C_{ij} - \frac{1}{N}\sum_{ij}\gamma_i K_{ij}$$

subject to $\sum_{i=1}^{N}\gamma_i = 1$ and $\gamma_i \geq 0\ \forall\ i$, can be written as

$$\min_{\gamma_1,\gamma_2}\frac{1}{2}\sum_{i,j=1}^{2}\gamma_i\gamma_j C_{ij} + \sum_{i=1}^{2}\gamma_i S_i - \sum_{i=1}^{2}\gamma_i T_i + \sigma, \tag{9}$$

where $\sigma = S - T$ and

$$S_i = \sum_{j=3}^{N}\gamma_j C_{ij};\ S = \frac{1}{2}\sum_{i,j=3}^{N}\gamma_i\gamma_j C_{ij};$$

$$T_i = \frac{1}{N}\sum_{j} K_{ij};\ T = \frac{1}{N}\sum_{i=3}^{N}\sum_{j}\gamma_i K_{ij},$$

subject to $\sum_{i=1}^{2}\gamma_i = \Delta,\quad \gamma_1,\gamma_2 \geq 0$, where $\Delta = 1 - \sum_{i=3}^{N}\gamma_i$. Following [30], we discard $\sigma = S - T$ in (9), which is independent of γ_1 and γ_2, and eliminate γ_1 to obtain

$$\min_{\gamma_1,\gamma_2}\frac{1}{2}\{[(\Delta-\gamma_2)^2 C_{11} + 2(\Delta-\gamma_2)\gamma_2 C_{12} + {\gamma_2}^2 C_{22}] + (\Delta-\gamma_2)S_1 + \gamma_2 S_2 - (\Delta-\gamma_2)T_1 - \gamma_2 T_2\}.$$

Setting the derivative of the above to zero and solving γ_2 then equals

$$\frac{\Delta(C_{11}-C_{12}) + (S_1-S_2) - (T_1-T_2)}{C_{11} - 2C_{12} + C_{22}}. \tag{10}$$

γ_1 can then be recovered from $\gamma_1 = \Delta - \gamma_2$. Let γ_1^*, γ_2^* denote the parameter values before the step, and $I_i = C_{1i}\gamma_1^* + C_{2i}\gamma_2^* + S_i - T_i$, we can give the update equation for γ_2 as

$$\gamma_2 = \gamma_2^* + \frac{I_1 - I_2}{C_{11} - 2C_{12} + C_{22}}, \tag{11}$$

which does not explicitly depend on γ_1^*. The complete optimization procedure is now given.

Initialization: The variables are initialized as $\gamma_i = p_i/\sum_i p_i$, where $p_i = \frac{1}{N}\sum_j K_{ij}$ is the Parzen window density estimate. This results in the points with higher density initially having larger γ values.

Optimization algorithm:

1. Searching γ_2 and γ_1: After initialization, the points with higher density will have larger γ values; we select the largest γ value in turn as the first variable γ_2 for the elementary optimization step, and search for the second variable γ_1 which can generate the largest value of I_i. When γ_2 is less than a preset tolerance, stop the current search loop, and go to check the terminating criterion.
2. Updating γ_2 and γ_1: If γ_1 is greater than the preset tolerance, update γ_2. If the updated $\gamma_2 < 0$, set

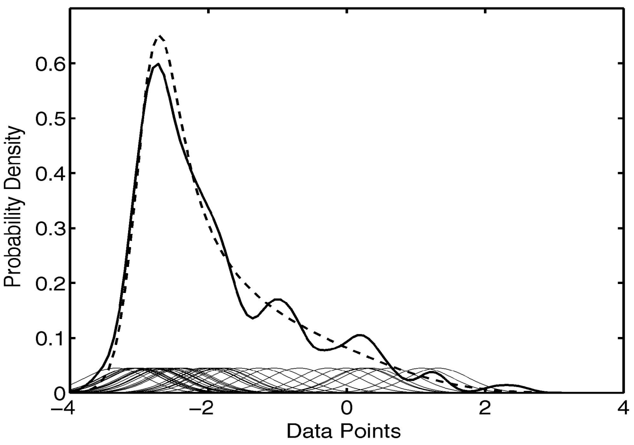

Fig. 1. The true density (dashed line) and the Parzen window estimate (solid line); each of the kernel functions is placed at the appropriate sample data point.

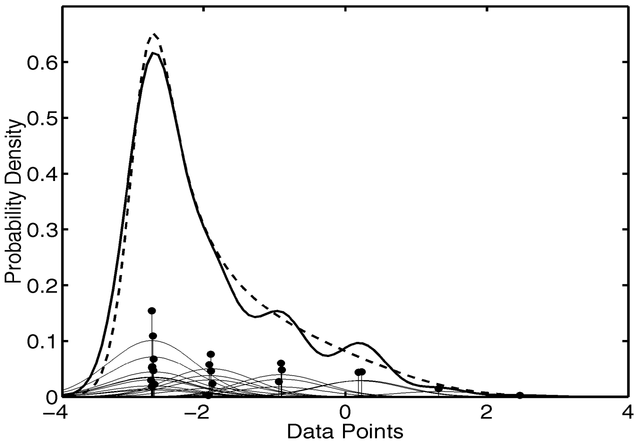

Fig. 2. The true density (dashed line) and the RSDE (solid line); each of the 21 nonzero kernel functions ($\sim$ 10 percent of the original sample size) is placed at the appropriate sample data point and the length of the vertical line denotes the value of the corresponding weighting coefficient.

$\gamma_2 = 0$. Then, update γ_1 by $\Delta - \gamma_2$, if the updated $\gamma_1 < 0$, set $\gamma_1 = 0$ and $\gamma_2 = \Delta$.

3. Terminating criterion: There are two criteria to terminate the algorithm.

   a. Comparing the value of the objective function with the same value obtained in the previous search loop: If it decreases and the difference is greater than the preset error tolerance, then restart another search loop, otherwise, recover the previous γ value and terminate the algorithm.
   b. If no variables are updated during a loop, terminate the algorithm.

So, the above optimization (5), in the case of a Gaussian window, will provide a nonparametric estimate of the data density based on a subset of the original data sample defined as $\hat{p}(\mathbf{x}) = \sum_{\gamma_n \neq 0} \gamma_n \mathcal{G}_h(\mathbf{x}, \mathbf{x}_n)$. A number of experiments[4] are now provided to demonstrate the proposed RSDE method.

4 Experiments

4.1 One-Dimensional Example

The first demonstration of the RSDE employs a 1D data sample which is drawn from a heavily skewed distribution defined as $p(x) = \frac{1}{8}\sum_{i=0}^{7} \mathcal{G}_{h_i}(\mu_i, x)$, where $h_i = \left(\frac{2}{3}\right)^i$ and $\mu_i = 3(h_i - 1)$ [23]. A sample of 200 points is drawn from the distribution and a Parzen window density estimator employing a Gaussian kernel is devised using the data sample. The width of the kernel is found by leave-one-out cross validation. A further sample of 10,000 data points are then drawn from the density and the L_2 error between the Parzen estimate and true density is computed; this procedure is repeated 200 times. The error was found to be (median value & interquartile range) 0.0033 & 0.0033. Fig. 1 shows the true density and the estimated density for a particular sample realization along with the individual kernel functions placed at the sample points.[5]

The RSDE is applied to this data using, as above, a Gaussian kernel, and the width of the kernel is also set by cross-validation. However, it was noted in the reported experiments that measuring the cross-entropy [3] between the RSDE and the existing Parzen estimator, and then selecting the width value which returns the minimal cross-entropy, is found to give similar results to cross-validation, while reducing the effective number of optimization runs (time taken) required for width selection. From the 200 samples, the median value for the number of nonzero weighting coefficients was 13—amounting to less than 8 percent of the original sample—the minimum and maximum values of nonzero weighting coefficient was 5 and 42, respectively. The corresponding L_2 error based on 10,000 data points for 200 sample realizations was measured to be 0.0035 and 0.0030. Due to the highly asymmetric nature of the distribution of errors, a Rank sum Wilcoxon test [16] is applied and shows that both error distributions for the full Parzen and RSDE estimators, at the 5 percent significance level, are identical. This is a somewhat satisfying result in that the accuracy of the RSDE is shown to be the same as the Parzen for this particular density function. The resulting estimate for one sample realization is shown in Fig. 2. Notice that both methods estimate the mode well and the ripples in the tail, which are characteristic of finite sample Parzen estimates of long tailed behavior, can be seen to be somewhat smoothed by the RSDE.

As an illustration of how the weighting coefficients evolve during the optimization of ISE, Fig. 3 shows the weighting coefficients as a number of stem-plots, each corresponding to the estimated weighting coefficients after a given number of SMO steps. It is clear that the number of nonzero coefficients drops as the number of steps increases. Fig. 4 also shows that as the level of sparsity increases the plug-in estimate of ISE (minus the unknown density dependent constant term) decreases.

[4] A MATLAB implementation of RSDE, as well as the data sets employed in the reported experiments, is available at the following website http://cis.paisley.ac.uk/giro-ci0/reddens.

[5] Every fifth data point is used in the figure for the purposes of clarity.

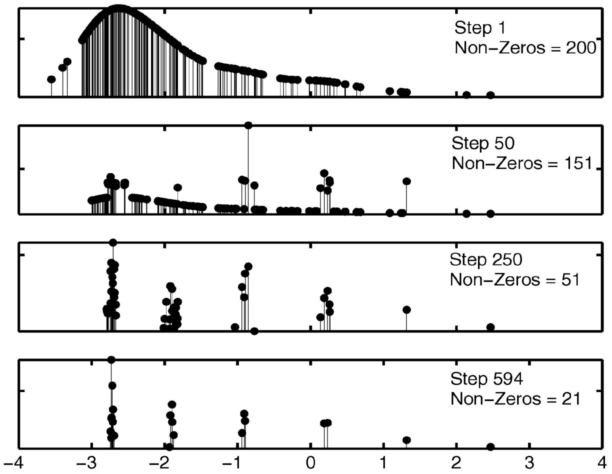

Fig. 3. A visual representation of the evolution of the weighting coefficients. The top chart shows the γ coefficients placed at the position of the corresponding point after initialization, where they take on the normalized values of the point Parzen density estimates. The following charts show the weighting coefficients after 50, 250, and 594 steps, along with the value of the number of nonzero coefficients remaining.

4.2 Two-Dimensional Examples

The second demonstration is primarily illustrative and employs a sample (200 points) of 2D data which is generated with equal probability from an isotropic Gaussian, and two Gaussians with both positive and negative correlation structure. The probability density is estimated using a Parzen window employing a Gaussian kernel and leave-one-out cross-validation was employed in selecting the kernel bandwidth. The probability density isocontours, along with the data sample, is shown in Fig. 5a. By way of a comparison, the multiscale density-based data condensation method of [18] is applied to this toy example and the results are shown in Fig. 5c. A similar level of data reduction to that of RSDE is achieved, where large circles denote identified regions of low density with smaller ones defining regions of high density. The selected data points are encircled. As a means of data condensation with the specific aim of nonparametric density estimation, the multiscale approach [18] has been shown to consistently outperform the data reduction methods proposed by Fukunaga and Mantock [6] and Astrahan [1].

The RSDE is obtained by optimizing (5) and employing a Gaussian kernel in this case. As before, the kernel bandwidth is selected by minimizing the cross-entropy between the Parzen window estimate and the RSDE. Fig. 5b shows the corresponding isocontours along with the reduced data set, denoted by the encircled points, which amounts to a 91 percent reduction in the number of points required to estimate the density of further data points. It is interesting to note that the selected points (nonzero weighting) occur in the regions of highest density of the sample and, indeed, lie approximately on the principal axis of the two elongated Gaussians.

To illustrate this further, 3,000 data points from the 2D S-shaped distribution[6] are used to estimate the associated PDF. Fig. 6a shows the data sample and the isocontours of the Parzen density estimate. Fig. 6b shows the density isocontours obtained using RSDE and the selected points (12 percent of the original sample) are encircled as in the previous example. The selected points lie in the center of the distribution and the shape they form is somewhat reminiscent of that obtained by Principal Curves [8]. This similarity may form an interesting area of future investigation. This observation is in contrast to the support vector data description methods [30], [32] where the boundary points of the sample tend to be selected.

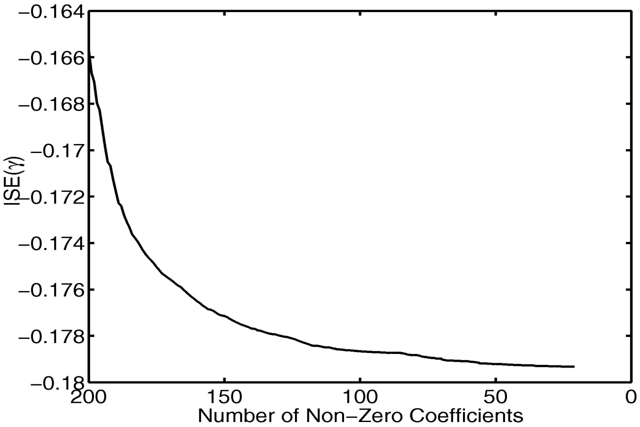

Fig. 4. The relationship between the number of nonzero weighting coefficients and the estimated ISE during the optimization process.

4.3 Comparative Experiments

The first experiment in this section compares the RSDE with the SVM approach to density estimation [19]. The 1D density function employed in [19] is used in this experiment, i.e., $p(x) = \frac{1}{2\sqrt{2\pi}} exp(-0.5|x-2|^2) + \frac{0.7}{4} exp(-|x+2|)$. This density is a particularly useful test as it possesses both bimodality and long tailed behavior in one of the modes. As in [19], samples of 100 points are drawn from the density and then the SVM, RSDE, and Parzen density estimators are devised, a further 10,000 samples are then drawn from the PDF and used to compute, in this case, as in [19], the L_1 error, the integrated absolute deviation of the estimate from the true density value. This procedure was then repeated 1,000 times to assess the bias and variance associated with each of the estimators. The free parameter (kernel width and ϵ) values reported in [19] for the SVM estimator were employed throughout, while leave-one-out cross-validation was used to set the Gaussian width for the Parzen window, and minimum cross-entropy between the Parzen and RSDE was used to set the kernel width for the RSDE. The results are shown in Fig. 7.

The box shows the quartiles of the distribution of error values, while the whiskers show the range of the error values, and the points beyond the whiskers fall outwith 1.5 times the interquantile range. It is interesting to note that both the RSDE and SVM estimators introduce an equally small amount of difference from the Parzen estimator, though the variability of the SVM estimator is slightly larger in this case. However, the RSDE can take advantage of the less computationally costly SMO routine in estimating the weighting coefficients. Fig. 8 shows the number of nonzero

[6] This data set is used to demonstrate the use of Principal Curves [8] and is available at http://www.iro.umontreal.ca/~kegl/research/pcurves/.

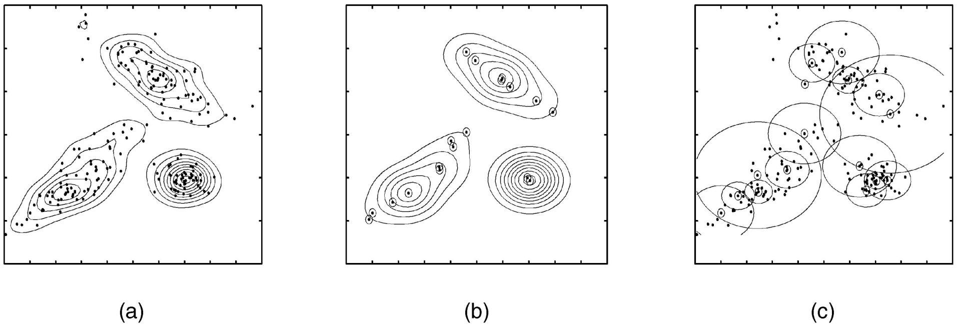

Fig. 5. (a) The Parzen window density estimate. (b) The RSDE with the retained points circled. (c) The results of the multiscale data condensation method where the selected points are encircled and the corresponding discs are shown.

values for both the SVM and RSDE estimators. Both have the same median value of 4, while the RSDE shows greater variability in the number of nonzero coefficients, primarily due to the kernel width value varying based on each sample in RSDE, while the value of ϵ in the SVM approach stayed fixed for each sample.

To further test RSDE, varying sizes of sample are drawn from both unimodal and bimodal distributions at different dimensionalities, and the accuracy[7] of the RSDE is compared with the Parzen density estimator. By way of further comparison with an alternative data reduction method, the density-based multiscale data condensation method [18] is employed to obtain a reduced size data sample from which to obtain a Parzen estimator with reduced computational complexity. This was chosen primarily due to the excellent results obtained in [18] with this method.

4.4 Multidimensional Unimodal Distribution

A multivariate (2D and 5D) Gaussian which is centered at the origin and has a covariance matrix such that $C_{ij} = 1$ where $i = j$ and $C_{ij} = 0.5$ where $i \neq j$, is used in this experiment. Samples of size 30 to 700 data points are drawn from the distribution and both a Parzen estimator and RSDE are fit to the data. A test sample of 10,000 points is then drawn and both the L_2 and L_1 error is computed; this is then repeated 200 times for each sample size. For each sample size, the average value of the level of sample size reduction achieved by the RSDE is then used to set the value of the number of nearest neighbors k the free parameter value in the multiscale condensation method of [18] which defines the associated condensation level. This reduced set is then used to devise a Parzen density estimate which is then tested alongside the full Parzen and the proposed RSDE.

The results are summarized in Fig. 9, where the L_2 error for the Parzen, RSDE, and multiscale method is plotted against sample size. The corresponding L_1 errors are detailed in Table 1. The following abbreviations are used in the tables: Sample Size (SS), Remaining Data (percentage of original sample size) (RD), Parzen Window (PW), and Multiscale Data Condensation (DC). The points to note from Fig. 9 are that the rate of convergence of both the full Parzen estimators and RSDE are similar and that the variance of the estimators both decrease at the same rate with sample size. The levels of data reduction for the various sample sizes are given in Table 1.

The levels of data reduction remain relatively constant at sample sizes of 400 and beyond with only on average 7 percent of the sample being used. These data reduction rates are then used to select the appropriate parameter value for the data condensation method of [18] in order to yield a similar level of data reduction. The accuracy of the Parzen estimator obtained by the multiscale data condensation method [18] is measured as above and is shown in Fig. 9 and Table 1. It is clear that, for similar levels of data reduction, the RSDE provides a significant improvement in accuracy in terms of L_2 and L_1 metric for this type of data.

The same experiment is conducted for data samples drawn from a similar 5D Gaussian. The results are given in Fig. 10 and Table 2.

From the results, a similar trend in the accuracy of the estimate is observed as for the 2D case. However, it can be seen that the level of data reduction is not so aggressive at the small sample sizes with 58 percent of the sample being retained for the small 30 point sample. This is a nice example showing that the data reduction obtained is driven by the reduction of ISE. Clearly, excessive reduction of the small sample size would result in large residual error due to the higher dimensionality of the data in this case. This is in contrast to the data reduction method of [18], where the data reduction is governed by the chosen value of k, as such, there is no automatic or implicit means of controlling the ensuing error in density estimate by the adoption of the method of [18].

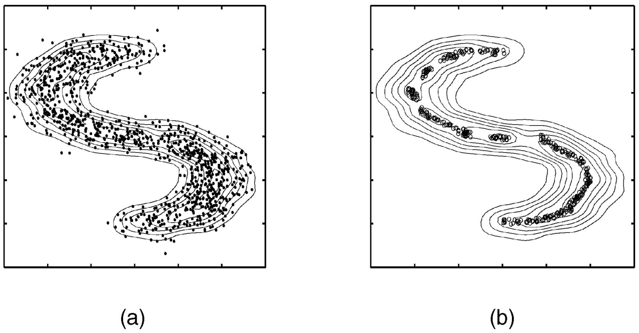

Fig. 6. (a) The data sample and the Parzen window density estimate contours. (b) The RSDE with the retained points circled.

[7] We report both L_2 and L_1 errors in the subsequent set of experiments.

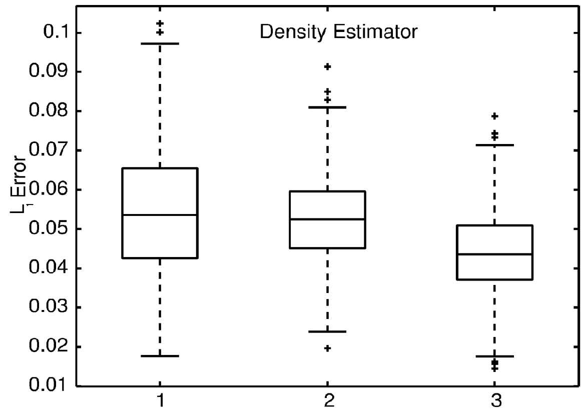

Fig. 7. Boxplot of the L_1 error for the SVM (1), RSDE (2), and Parzen (3) density estimators.

4.5 Multidimensional Bimodal Distribution

The experiments in the previous section are now repeated for a bimodal distribution composed of two Gaussians centered at $(1,1)$ and $(-1,-1)$, with common covariance $(1\ 0.5; 0.5\ 1)$ in the 2D case. In the 5D case, each Gaussian is centered at $(1,1,1,1,1)$ and $(-1,-1,-1,-1,-1)$, with common covariance as defined for the Gaussian of the previous section. The accuracy results for the 2D and 5D cases are given in Figs. 11 and 12, and Tables 3 and 4. As with the unimodal density, the RSDE has similar bias and variance to the Parzen density estimator, while the data condensation approach has a higher bias level for the same amount of data reduction.

As in the case of unimodal data, the bias of the RSDE follows that of the full sample Parzen estimator with the Parzen estimator based on the data condensation method showing a consistently larger bias.

5 Conclusions and Discussion

The experiments reported have demonstrated that the RSDE provides very similar estimation accuracy as the Parzen window estimator, while employing greatly reduced numbers of points from the available sample. The SVM approach to density estimation [19], [34], [35] sets out to solve the inverse linear operator problem and so estimates the empirical distribution function from the sample. The ϵ-insensitive loss employed [19], [34] provides the sparse representation of the density, however, from the perspective of practical implementation, the dense nature of the constraints requires generic quadratic optimization routines. One of the alternate SVM approaches proposed in [35] was to minimize the L_2 error between the SVM density estimate and a Parzen estimate, while enforcing sparsity of representation by a suitable regularizing term, which then introduces the added complexity of selecting the appropriate trade off between sparsity and accuracy. The approach taken herein is fundamentally different in that the ISE between the true (unknown) density and the reduced set estimator is minimized. The sparsity of representation (data condensation) emerges naturally from direct minimisation of ISE due to the required constraints on the functional form of $\hat{p}(\mathbf{x})$, without the requirement to resort to additional sparsity inducing regularization terms or employing L_1 or ϵ-insensitive losses [33], [34], [35].

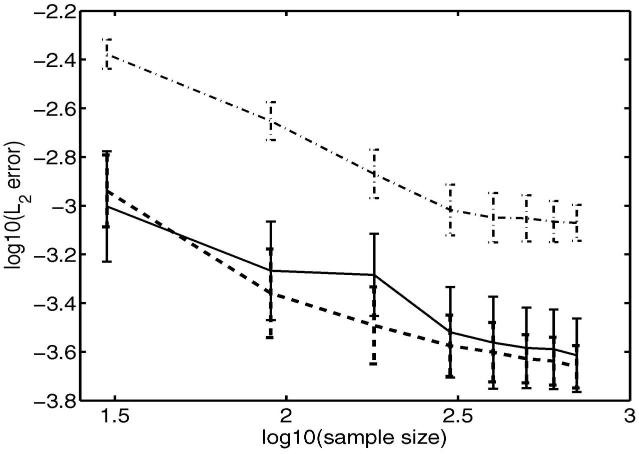

Fig. 9. L_2 error between the true density of a 2D Gaussian and density estimators against sample size. The bars denote one standard deviation. The Parzen window is denoted by a solid line, RSDE is denoted by the dashed line, and Multiscale Data Condensation is denoted by a dash-dot line.

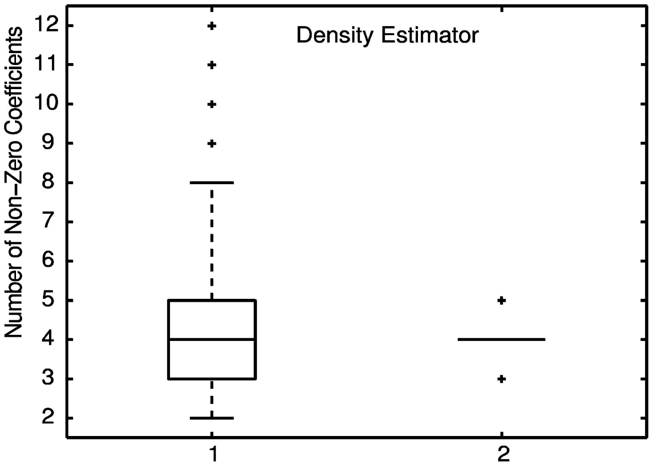

Fig. 8. Boxplot of the number of nonzero weighting coefficients for (1) RSDE and the (2) SVM density estimators.

TABLE 1
L_1 Error (Computed over 200 Trials) between True Density of 2D Gaussian and Respective Density Estimators against Sample Size

		L_1 Error (Mean ± STD)×10^{-2}		
SS	RD (%)	PW	RSDE	DC
30	11.60	$\mathbf{2.57 \pm 0.61}$	2.77±0.43	5.18±0.38
90	10.00	1.87±0.42	$\mathbf{1.69 \pm 0.35}$	3.73±0.32
180	8.67	1.81±0.32	$\mathbf{1.45 \pm 0.22}$	2.89±0.32
300	8.29	1.37±0.27	$\mathbf{1.32 \pm 0.15}$	2.43±0.27
400	7.88	1.30±0.25	$\mathbf{1.29 \pm 0.14}$	2.34±0.27
500	7.30	1.26±0.22	$\mathbf{1.25 \pm 0.10}$	2.33±0.23
600	7.30	1.25±0.21	$\mathbf{1.24 \pm 0.10}$	2.29±0.21
700	6.99	1.21±0.19	$\mathbf{1.21 \pm 0.09}$	2.27±0.18

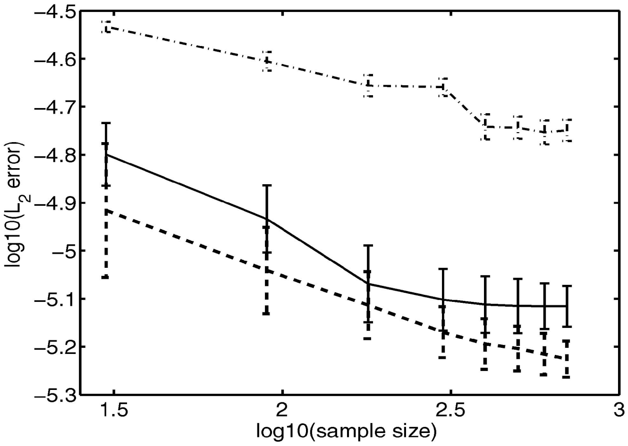

Fig. 10. L_2 error between the true density of a 5D Gaussian and density estimators charted against sample size. The bars denote one standard deviation. The Parzen window is denoted by a solid line, RSDE is denoted by the dashed line, and Multiscale Data Condensation is denoted by the dash-dot line.

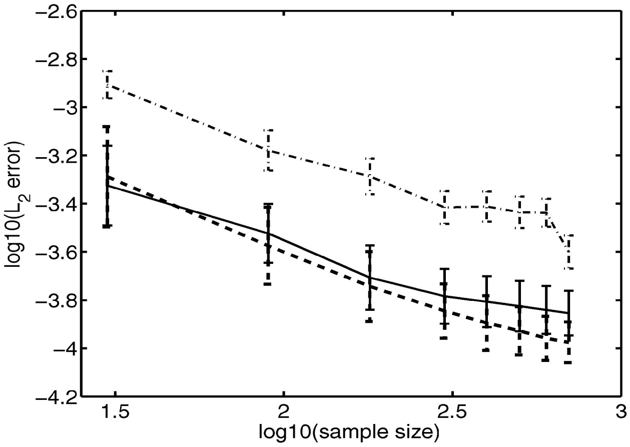

Fig. 11. L_2 error between the true density of a 2D mixture of two Gaussians and density estimators charted against sample size. The bars denote one standard deviation. The Parzen window is denoted by a solid line, RSDE is denoted by the dashed line, and Multiscale Data Condensation is denoted by the dash-dot line.

The Density-Based Multiscale Data Condensation method [18] offers a straightforward means of providing a sparse representation of a kernel density estimator once the parameter (k—number of nearest neighbors) which controls the rate of data condensation is set. It should be noted that this method returns a subset of the original data sample, the representation is multiscale as regions of estimated high density have more points removed than regions of low density. This sample can then be employed, among other uses, in devising a density estimator. When a predefined data reduction ratio (that obtained by RSDE in the reported experiments) is employed to define the free parameter k, the accuracy of the resulting Parzen density estimators have more bias than that obtained by RSDE.

One final point to note is that the reduced sample set returned by RSDE has a prototypical nature and this has been demonstrated on multivariate Gaussians where the selected points tend to lie on the principal axis of the distribution, and with isotropic Gaussians the points selected lie close to the distribution mean. Further, for an arbitrary non-Gaussian distribution, the selected points tend to lie on what could be considered to be the principal curve of the distribution.

In summary, this paper has presented a method that provides a kernel (Parzen) density estimator which employs a small subset of the available data sample based on the minimization of the integrated square error between the estimator and the true density. Other than the weighting coefficients which can be obtained through straightforward quadratic optimization, no additional free parameters, e.g., regularization term, bin width, or condensation ratio, are introduced into the proposed estimator. Due to the simple constraints on the error criterion optimization methods which have scaling of the order of $\mathcal{O}(N) \sim \mathcal{O}(N^2)$ can be employed. In testing, it has been shown that the proposed density estimation method has similar convergence rates to the Parzen window estimator which employs the full data

TABLE 2
L_1 Error (Computed over 200 Trials) between True Density of 5D Gaussian and Respective Density Estimators against Sample Size

		L_1 Error (Mean ± STD)$\times 10^{-3}$		
SS	RD (%)	PW	RSDE	DC
30	58.07	2.55±0.19	**2.31 ± 0.34**	3.60±0.06
90	8.62	2.14±0.17	**1.92 ± 0.19**	3.25±0.09
180	5.70	1.81±0.15	**1.74 ± 0.12**	3.03±0.09
300	3.85	1.73±0.12	**1.62 ± 0.08**	3.01±0.07
400	2.89	1.71±0.11	**1.58 ± 0.08**	2.70±0.09
500	2.33	1.69±0.10	**1.56 ± 0.07**	2.69±0.08
600	1.88	1.68±0.09	**1.54 ± 0.06**	2.63±0.09
700	1.71	1.68±0.08	**1.52 ± 0.06**	2.67±0.08

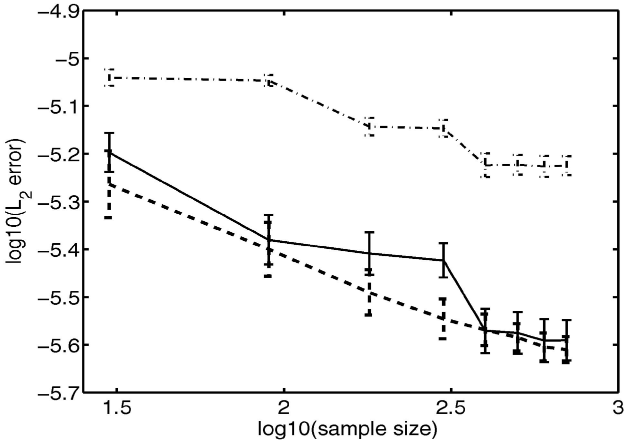

Fig. 12. L_2 error between true density of 5D mixture of two Gaussians charted against sample size. The Parzen window is denoted by a solid line, RSDE is denoted by the dashed line, and Multiscale Data Condensation is denoted by the dash-dot line.

TABLE 3
L_1 Error (Computed over 200 Trials) between True Density of 2D Mixture of Two Gaussians and Respective Density Estimators against Sample Size

SS	RD (%)	L_1 Error (Mean ± STD)×10^{-2}		
		PW	RSDE	DC
30	24.53	**1.77 ± 0.33**	1.87±0.46	2.94±0.18
90	16.38	1.39±0.19	**1.31 ± 0.24**	2.11±0.20
180	12.46	1.12±0.17	**1.08 ± 0.17**	1.86±0.16
300	11.36	1.03±0.13	**0.96 ± 0.11**	1.60±0.12
400	11.16	1.00±0.11	**0.91 ± 0.10**	1.60±0.11
500	10.40	0.98±0.11	**0.88 ± 0.08**	1.56±0.11
600	10.43	0.97±0.11	**0.85 ± 0.07**	1.56±0.10
700	10.84	0.95±0.10	**0.84 ± 0.06**	1.29±0.10

TABLE 4
L_1 Error (Computed over 200 Trials) between True Density of 5D Mixture of Two Gaussians and Respective Density Estimators against Sample Size

SS	RD (%)	L_1 Error (Mean ± STD)×10^{-3}		
		PW	RSDE	DC
30	35.80	1.66±0.09	**1.55 ± 0.14**	2.06±0.05
90	15.56	1.32±0.08	**1.29 ± 0.09**	2.04±0.03
180	8.76	1.26±0.06	**1.15 ± 0.06**	1.78±0.04
300	6.37	1.23±0.05	**1.07 ± 0.05**	1.77±0.04
400	5.27	1.04±0.05	**1.04 ± 0.04**	1.60±0.05
500	4.51	1.02±0.05	**1.02 ± 0.03**	1.60±0.04
600	3.98	1.00±0.05	**0.99 ± 0.03**	1.59±0.05
700	3.53	1.00±0.05	**0.98 ± 0.03**	1.59±0.04

sample and has been shown to have comparable performance to the SVM density estimation method [19], [34].

It has also been shown to have improved performance over the density-based multiscale data condensation method at predefined condensation rates. The proposed RSDE will find application in the many instances where a high-accuracy estimate of a PDF with low computational cost is required.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments which assisted in improving the quality of this paper. They would also like to acknowledge the helpful discussions with Dr. Ewan MacArthur and Dr. Allan MacLeod regarding this work. This work was supported by Scottish Higher Educational Funding Council Research Development grant "INCITE" http://www.incite.org.uk. A Matlab implementation of RSDE and example data sets are available at http://cis.paisley.ac.uk/giro-ci0/reddens.

References

[1] M.M. Astrahan, "Speech Analysis by Clustering or the Hyperplane Method," Stanford A.I. Project Memo, Stanford Univ., Calif., 1970.

[2] G.A. Babich and O. Camps, "Weighted Parzen Windows for Pattern Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 5, pp. 567-570, May 1996.

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

[4] E. Elgammal, D. Harwood, and L. Davis, "Nonparametric Model for Background Subtraction," *Proc. Sixth European Conf. Computer Vision,* pp. 751-761, 2000.

[5] K. Fukunaga and R.R. Hayes, "The Reduced Parzen Classifier," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 4, pp. 423-425, Apr. 1989.

[6] K. Fukunaga and J.M. Mantock, "Nonparametric Data Reduction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 6, pp. 115-118, 1984.

[7] M. Girolami, "Orthogonal Series Density Estimation and the Kernel Eigenvalue Problem," *Neural Computation,* vol. 14, no. 3, pp. 669-688, MIT Press, 2002.

[8] T. Hastie and W. Stuetzle, "Principal Curves," *J. Am. Statistical Assoc.,* vol. 84, no. 406, pp. 502-516, 1989.

[9] L. Holmström, "The Error and the Computational Complexity of a Multivariate Binned Kernel Density Estimator," *J. Multivariate Analysis,* vol. 72, no. 2, pp. 264-309, 2000.

[10] L. Holmström and A. Hämäläinen, "The Self-Organising Reduced Kernel Density Estimator," *Proc. IEEE Int'l Conf. Neural Networks,* vol. 1, pp 417-421, 1993.

[11] A.J. Izenman, "Recent Developments in Nonparametric Density Estimation," *J. Am. Statistical Assoc.,* vol. 86, pp. 205-224, 1991.

[12] B. Jeon and D.A. Landgrebe, "Fast Parzen Density Estimation Using Clustering-Based Branch and Bound," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 9, pp 950-954, Sept. 1994.

[13] D. Kim, "Least Squares Mixture Decomposition Estimation," unpublished doctoral dissertation, Dept. of Statistics, Virginia Polytechnic Inst. and State Univ., 1995.

[14] T. Kohonen, *Self-Organizing Maps.* Springer-Verlag, 1995.

[15] C. Lambert, S. Harrington, C. Harvey, and A. Glodjo, "Efficient Online Nonparametric Kernel Density Estimation," *Algorithmica,* vol. 25, pp 37-57, 1999.

[16] E.L. Lehmann, *Nonparametric Statistical Methods Based on Ranks.* New York: McGraw-Hill, 1975.

[17] G. McLachlan and D. Peel, *Finite Mixture Models.* Wiley, 2000.

[18] P. Mitra, C.A. Murthy, and S.K. Pal, "Density Based Multiscale Data Condensation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 6, June 2002.

[19] S. Mukherjee and V. Vapnik, "Support Vector Method for Multivariate Density Estimation," CBCL Paper #170, AI Memo #1653, 1999.

[20] E. Parzen, "On Estimation of a Probability Density Function and Mode," *Annals of Math. Statistics,* vol. 33, pp. 1065-1076, 1962.

[21] C.E. Priebe and D.J. Marchette, "Alternating Kernel and Mixture Density Estimates," *Computational Statistics and Data Analysis,* vol. 35, pp. 43-65, 2000.

[22] S. Roberts, "Extreme Value Statistics for Novelty Detection in Biomedical Signal Processing," *IEE Proc. Science, Technology, and Measurement,* vol. 47, no. 6, pp. 363-367, 2000.

[23] S. Sain, "Adaptive Kernel Density Estimation," PhD thesis, Rice Univ., 1994.

[24] D.W. Scott, "Remarks on Fitting and Interpreting Mixture Models," *Computing Science and Statistics,* K. Berk and M. Pourahmadi, eds., vol. 31, pp. 104-109, 1999.

[25] D.W. Scott and S.J. Sheather, "Kernel Density Estimation with Binned Data," *Comm. Statistics—Theory and Methods,* vol. 14, pp. 1353-1359, 1985.

[26] D.W. Scott and W.F. Szewczyk, "From Kernels to Mixtures," *Technometrics,* vol. 43, pp. 323-335,

[27] F. Sha, L. Saul, and D.D. Lee, "Multiplicative Updates for Non-Negative Quadratic Programming in Support Vector Machines." Technical Report MS-CIS-02-19, Univ. of Pennsylvania, 2002.

[28] B.W. Silverman, "Kernel Density Estimation Using the Fast Fourier Transform," *Applied Statistics,* vol. 31, pp. 93-99, 1982.

[29] B.W. Silverman, *Density Estimation for Statistics and Data Analysis.* Chapman and Hall, 1986.

[30] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, and R. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Computation,* vol. 13, pp. 1443-1471, 2001.

[31] B. Schölkopf, A. Smola, and K.R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation,* vol. 10, no. 5, pp. 1299-1219, 1998.

[32] D.M.J. Tax and R.P.W. Duin, "Support Vector Data Description," *Pattern Recognition Letters,* vol. 20, nos. 11-13, pp. 1191-1199, 1999.

[33] V.N. Vapnik, *Statistical Learning Theory.* New York: John Wiley and Sons, 1998.

[34] V. Vapnik and S. Mukherjee, "Support Vector Method for Multivariate Density Estimation," *Advances in Neural Information Processing Systems,* S. Solla, T. Leen, and K.-R. Müller, eds., MIT Press pp 659-665, 2000.

[35] J. Weston, A. Gammerman, M.O. Stitson, V. Vapnick, V. Vovk, and C. Watkins, "Support Vector Density Estimation," *Advances in Kernel Methods,* MIT Press, 1999.

Mark Girolami received a degree in mechanical engineering from the University of Glasgow (1985) and the PhD degree in computer science from the University of Paisley (1998). He was a development engineer with IBM from 1985 until 1995, when he left to pursue an academic career. From May to December 2000, Dr. Girolami was the TEKES visiting professor at the Laboratory of Computing and Information Science at the Helsinki University of Technology. From 1998 to 1999, he was a research fellow at the Laboratory for Advanced Brain Signal Processing at the Brain Science Institute, RIKEN, Wako-Shi, Japan. He is currently a staff member at the University of Paisley.

Chao He received the BSc degree with the highest honor in automatic control in 1996, and the PhD degree with the outstanding dissertation award in control theory and engineering in 2001, both from the Beijing Institute of Technology, China. After undertaking postdoctoral research in the Department of Electrical and Computer Engineering at the University of Alberta, Canada, he joined the School of Information and Communication Technologies at the University of Paisley, United Kingdom, in March 2002 as a postdoctoral research assistant.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.