A Sparse Kernel Density Estimation Algorithm using Forward Constrained Regression

Xia Hong[†], Sheng Chen[‡], and Chris Harris[‡]

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

Abstract. Using the classical Parzen window (PW) estimate as the target function, the sparse kernel density estimator is constructed in a forward constrained regression manner. The leave-one-out (LOO) test score is used for kernel selection. The jackknife parameter estimator subject to positivity constraint check is used for the parameter estimation of a single parameter at each forward step. As such the proposed approach is simple to implement and the associated computational cost is very low. An illustrative example is employed to demonstrate that the proposed approach is effective in constructing sparse kernel density estimators with comparable accuracy to that of the classical Parzen window estimate.

Key words: cross validation, jackknife parameter estimator, Parzen window, probability density function, sparse modelling.

1 Introduction

A basic problem that is pertinent to many machine learning and pattern recognition applications is to estimate the probability density function (pdf) from observed data samples [1–4]. A general and powerful approach to the problem of probability density function estimation is the finite mixture model [5]. The finite mixture model includes the well known PW estimate [4] as a special case.

It is useful to develop methods of fitting a finite mixture model with the capability to infer a minimal number of mixtures from the data efficiently. Researches into sparse density estimators include the support vector machines[6, 7], the reduced set density estimator (RSDE) [8], and sparse pdf estimator using forward orthogonal regression (OFR) [9–11].

In this paper a new algorithm for sparse kernel density estimator is introduced, using the classical Parzen window estimate as the target function, and the kernels as regressors. The proposed sparse kernel density estimator construction using forward constrained regression algorithm (FCR-SDC) is based on the forward constrained regression [12] in which mixing weights are estimated through a set of parameters, each of which relates to the model at the current regression stage and a new candidate term. In each forward stage, the model term selection is based on the criterion of a minimal leave-one-out (LOO) test score, subject

to a simple positivity constraint. A one parameter jackknife parameter estimator is utilized in each regression step, subject to the same positivity constraint check. The proposed algorithm has the advantage of maximal computationally efficiency due to that (i) the parameter estimation is reduced to the solution of the minimal possible number of one parameter; and (ii) the positivity constraint on the mixing weights can be easily accommodated.

2 The Kernel Density Estimator

Given a finite data set consisting of N data samples, $D = \{\mathbf{x}_1, ..., \mathbf{x}_j, ...\mathbf{x}_N\}$, where the feature vector variable $\mathbf{x}_j \in \Re^m$ follows an unknown probability density function $p(\mathbf{x})$, the problem under study is to find a sparse approximation of $p(\mathbf{x})$ based on D.

A general kernel based density estimate of $p(\mathbf{x})$ is given by

$$\begin{aligned} \hat{p}(\mathbf{x}; \mathbf{g}, \sigma) &= \textstyle\sum_{j=1}^{N} g_j K(\mathbf{x}, \mathbf{x}_j) \\ \text{subject to} \quad & g_j \geq 0, \quad j = 1, ..., N, \quad \mathbf{g}^T \mathbf{1} = 1. \end{aligned} \tag{1}$$

where $\mathbf{g} = [g_1, g_2, ..., g_N]^T$. g_j's are the kernels weights. $\mathbf{1}$ is a vector with an appropriate dimension and all elements as ones. $K(\mathbf{x}, \mathbf{x}_j)$ is a chosen kernel function with kernel width σ. In this study,

$$K(\mathbf{x}, \mathbf{x}_j) = \frac{1}{(2\pi\sigma^2)^{m/2}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_j\|^2}{2\sigma^2}\right) \tag{2}$$

is used. Let the well known Parzen window estimator be denoted by $\hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par})$, where $\mathbf{g}^{Par} = [g_1^{Par}, ..., g_N^{Par}]^T$, $g_j^{Par} = \frac{1}{N}$, $\forall j$. Clearly the Parzen window estimator is a special case of (1).

The log-likelihood for $\mathbf{g}$ can be formed using observed data D as

$$\log L = \frac{1}{N} \sum_{i=1}^{N} \log \hat{p}(\mathbf{x}_i; \mathbf{g}, \sigma) = \frac{1}{N} \sum_{i=1}^{N} \log \left(\sum_{j=1}^{N} g_j K(\mathbf{x}_i, \mathbf{x}_j) \right) \tag{3}$$

Note that by the law of large numbers the log-likelihood of (3) tends to

$$\int_{\Re^m} p(\mathbf{x}) \log \hat{p}(\mathbf{x}; \mathbf{g}, \sigma) d\mathbf{x} \tag{4}$$

as $N \to \infty$ with probability one. (4) is simply the negative cross-entropy or divergence between the true density $p(\mathbf{x})$ and the estimate $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$. It can be shown that for a given kernel width $\sigma = \sigma^{Par}$, the Parzen window estimator $g_j^{Par} = \frac{1}{N}$, $\forall j$ can be obtained as an optimal estimator via the maximization of (3) with respective to $\mathbf{g}$ subject to the constraints $g_j \geq 0$, $j = 1, ..., N$, $\mathbf{g}^T \mathbf{1} = 1$. Clearly for the PW estimator, the associated computational cost for evaluating a point probability density estimate scales directly with the sample size N. Hence

it is desirable to devise a spare representation of $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$, in which the terms are composed of a small subset of data samples.

In the proposed proposed sparse kernel density estimator algorithm the PW estimator is initially constructed and used as the target function [11]. Specifically we can write a regression equation [11] linking $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ and $\hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par})$ as

$$\begin{aligned} \hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par}) &= \hat{p}(\mathbf{x}; \mathbf{g}, \sigma) + \varepsilon(\mathbf{x}) \\ &= \sum_{j=1}^{N} g_j K(\mathbf{x}, \mathbf{x}_j) + \varepsilon(\mathbf{x}) \end{aligned} \tag{5}$$

where $\varepsilon(\mathbf{x})$ is the modelling error at $\mathbf{x}$ between the sparse kernel density estimator $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ and the PW density estimator $\hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par})$ constructed based on D. The aims are to obtain g_j that minimize some modelling error criterion, e.g. $E[\varepsilon^2(\mathbf{x})]$, and simultaneously to achieve a sparse representation of $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ (with most elements in $\mathbf{g}$ being zeros in (5)) subject to the constraints $g_j \geq 0$, $j = 1, ..., N$, $\mathbf{g}^T \mathbf{1} = 1$.

3 The Sparse Kernel Density Estimator Construction Algorithm using Forward Constrained Regression

The proposed sparse kernel density estimator is based on the general idea of the mixtures of experts network (MEN) [13] and forward constraint regression[12] described below.

3.1 The Mixtures of Experts Network and the Forward Constraint Regression Algorithm

The mixture of experts network [13], as depicted in Figure 1, can be viewed as a set of linear-in-the-parameter models with convex constraints on the combination parameters through

$$\hat{y}(\mathbf{x}) = \sum_{j=1}^{M} g_j \hat{y}_j(\mathbf{x}) \tag{6}$$

where $g_j \geq 0$, $\sum_{j=1}^{M} g_j = 1$, $\hat{y}_j(\mathbf{x})$, $j = 1, \cdots, M$ are the output of each expert, and $\hat{y}(\mathbf{x})$ is the composite output of the MEN.

Suppose that M experts $\hat{y}_j(\mathbf{x})$ are ordered in a sequence labelled by j, $j = 1, 2, ..., M$, and the MEN is constructed sequentially. Let a superscript (k) denote the kth forward step. At the kth forward step, the system is constructed using the first k experts, such that the MEN system at the kth step is

$$\hat{y}^{(k)}(\mathbf{x}) = \sum_{j=1}^{k} g_j^{(k)} \hat{y}_j(\mathbf{x}) \tag{7}$$

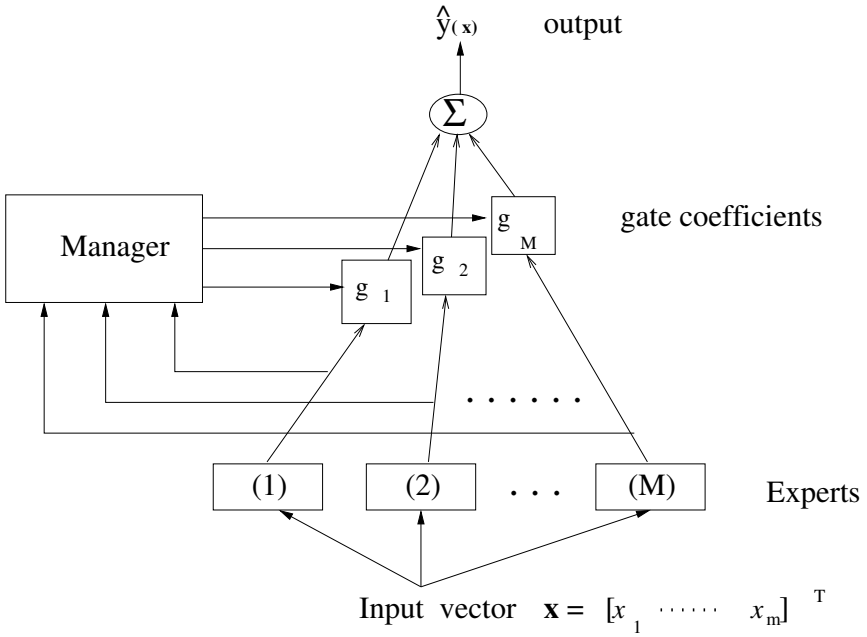


Fig. 1. The mixture of experts network

where $g_j^{(k)}$'s are the combination coefficients at the kth step, with $g_j^{(k)} \geq 0$, $\sum_{j=1}^{k} g_j^{(k)} = 1$, for $k = 2, ..., M$. The MEN system can be constructed using a FCR procedure described below [12]:

(i) At the first step, the MEN system is the first expert.

$$\hat{y}^{(1)}(\mathbf{x}) = \hat{y}_1(\mathbf{x}) \tag{8}$$

This means that $g_1^{(1)} = 1$.

(ii) At the kth step, $k = 2, \cdots, M$, the MEN system is constructed by including the kth expert into the MEN as

$$\hat{y}^{(k)}(\mathbf{x}) = \lambda_{k-1}\hat{y}^{(k-1)}(\mathbf{x}) + (1 - \lambda_{k-1})\hat{y}_k(\mathbf{x}) \tag{9}$$

where $0 \leq \lambda_{k-1} \leq 1$, $\forall k$.

It can be shown [12] that the system constructed using the FCR procedure satisfies the convex constraints condition for weights; $g_j^{(k)} \geq 0$, $\sum_{j=1}^{k} g_j^{(k)} = 1$, for $k = 2, ..., M$.

3.2 The Forward Constrained Regression Algorithm for Sparse Kernel Density Estimation

Suppose that a sparse kernel density estimator is based on the kernels formed from $D_s = [\mathbf{x}'_1, ..., \mathbf{x}'_s]$, a subset of s data samples selected from D. That is, if

$\mathbf{x}_6$ is selected to form the first kernel, this is denoted as $\mathbf{x'}_1$. The sparse kernel density estimator $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ in (5) can be regarded as a MEN system with the kernel functions $K(\mathbf{x}, \mathbf{x}'_j)$ as the experts $\hat{y}_j(\mathbf{x})$. The kernel functions $K(\mathbf{x}, \mathbf{x}_j)$ with nonzero g_j's are included into the model in a forward manner. At the kth forward step, the intermediate kernel density estimator $\hat{p}^{(k)}(\mathbf{x}; \mathbf{g}^{(k)}, \sigma)$ can then be denoted by $\hat{y}^{(k)}(\mathbf{x})$ as

$$\hat{y}^{(k)}(\mathbf{x}) = \sum_{j=1}^{k} g_j^{(k)} K(\mathbf{x}, \mathbf{x'}_j) \tag{10}$$

where $g_j^{(k)}$, $j = 1, ..., k$ are the kernels weights at the k^{th} forward step.

Initialization The initialization of the MEN system is to determine the first expert by selecting the first kernel $K(\mathbf{x}, \mathbf{x}'_1)$, so that

$$\hat{y}^{(1)}(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}'_1) \tag{11}$$

and $g_1^{(1)} = 1$. From (5) and (11)

$$\hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par}) = K(\mathbf{x}, \mathbf{x'}_1) + \varepsilon(\mathbf{x}) \tag{12}$$

From N kernels $K(\mathbf{x}, \mathbf{x}_j)$, $j = 1, ...N$, one is to be determined as $K(\mathbf{x}, \mathbf{x}'_1)$. This is simply done by searching for the term that produces the smallest value of mean squares modelling errors over D, i.e.

$$j_1 = \arg\min\{\sum_{i=1}^{N} [\hat{p}(\mathbf{x}_i; \mathbf{g}^{Par}, \sigma^{Par}) - K(\mathbf{x}_i, \mathbf{x}_j)]^2, \forall j\} \tag{13}$$

and $\mathbf{x}_{j_1}$ is then set as $\mathbf{x'}_1$.

Kernel selection using leave-one-out (LOO) test score and the jackknife parameter estimator Now consider the model term selection for forward step $k \geq 2$. (9) can be rewritten as

$$\hat{y}^{(k)}(\mathbf{x}) = \lambda_{k-1} \hat{y}^{(k-1)}(\mathbf{x}) + (1 - \lambda_{k-1}) K(\mathbf{x}, \mathbf{x'}_k) \tag{14}$$

The right hand side of (14) is a convex combination of two terms, the current MEN system $\hat{y}^{(k-1)}(\mathbf{x})$ and the kth kernel $K(\mathbf{x}, \mathbf{x'}_k)$ to be included into the model at the kth forward step. The following proposed algorithm aims to resolve two problems simultaneously; (i) which kernel is to be selected as $K(\mathbf{x}, \mathbf{x'}_k)$ from $(N-k+1)$ candidate kernels and (ii) what type of parameter estimator is adopted for λ_{k-1}.

From (5) and (14) we have

$$\begin{aligned} \hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par}) &= \sum_{j=1}^{k} g_j^{(k)} K(\mathbf{x}, \mathbf{x'}_j) + \varepsilon(\mathbf{x}) \\ &= \lambda_{k-1} \hat{y}^{(k-1)}(\mathbf{x}) + (1 - \lambda_{k-1}) K(\mathbf{x}, \mathbf{x'}_k) + \varepsilon(\mathbf{x}) \end{aligned} \tag{15}$$

With N data samples, define $\hat{\mathbf{p}}^{Par} = [\hat{p}(\mathbf{x}_1; \mathbf{g}^{Par}, \sigma^{Par}), ..., \hat{p}(\mathbf{x}_N; \mathbf{g}^{Par}, \sigma^{Par})]^T$, $\hat{\mathbf{y}}^{(k-1)} = [\hat{y}^{(k-1)}(\mathbf{x}_1), ..., \hat{y}^{(k-1)}(\mathbf{x}_N)]^T$, $\boldsymbol{\psi} = [K(\mathbf{x}_1, \mathbf{x}'_k), ..., K(\mathbf{x}_N, \mathbf{x}'_k)]^T$ and $\boldsymbol{\varepsilon} = [\varepsilon(\mathbf{x}_1), ..., \varepsilon(\mathbf{x}_N)]^T$. Then (15) can be rewritten in the vector form as

$$\hat{\mathbf{p}}^{Par} = \lambda_{k-1}\hat{\mathbf{y}}^{(k-1)} + (1 - \lambda_{k-1})\boldsymbol{\psi} + \boldsymbol{\varepsilon} \tag{16}$$

or

$$\mathbf{t} = \lambda_{k-1}\mathbf{w} + \boldsymbol{\varepsilon} \tag{17}$$

with $\mathbf{t} = [t_1, ..., t_N]^T = \hat{\mathbf{p}}^{Par} - \boldsymbol{\psi}$, $\mathbf{w} = [w_1, ..., w_N]^T = \hat{\mathbf{y}}^{(k-1)} - \boldsymbol{\psi}$.

Minimizing the loss function $J = \boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$ with respect to λ_{k-1} to yield the least squares solution

$$\begin{aligned} \lambda_{k-1}^{LS} &= \frac{\mathbf{w}^T\mathbf{t}}{\mathbf{w}^T\mathbf{w}} \\ &= \frac{b_{k-1}}{a_{k-1}} \end{aligned} \tag{18}$$

where $b_{k-1} = \mathbf{w}^T\mathbf{t}$ and $a_{k-1} = \mathbf{w}^T\mathbf{w}$.

The kth step of the MEN system involves the selection of $K(\mathbf{x}, \mathbf{x}'_k)$. Note that by using each of the $(N - k + 1)$ candidate kernels to form $\boldsymbol{\psi}$ in turn, (18) is repeated calculated. For some candidate kernels, the solution may not satisfy the constraints $0 \leq \lambda_{k-1}^{LS} \leq 1$. These kernels will then not be considered to be appropriate.

For all model terms which satisfy the constraints $0 \leq \lambda_{k-1}^{LS} \leq 1$, the following proposed model term selection algorithm is applied, which combines the leave-one-out cross validation with the jackknife parameter estimator for λ_{k-1} (given by (21) below), subject to $0 \leq \lambda_{k-1} \leq 1$.

The leave-one-out cross validation involves the removal of each $\mathbf{x}_j$ in turn from the estimation data set D, $j = 1, ..., N$. The removed data point is used as a test point for the model constructed using the modified data set. It is easy to verify that the least squares solution using $(D \setminus \mathbf{x}_j)$, is given by

$$\lambda_{k-1}^{(-j)} = \frac{b_{k-1} - w_j t_j}{a_{k-1} - w_j^2}, \quad j = 1, ..., N \tag{19}$$

and the mean squares of LOO errors $\varepsilon^{(-j)}(\mathbf{x}_j)$ is given by

$$J_k = E\{[\varepsilon^{(-j)}(\mathbf{x}_j)]^2\} = \frac{1}{N}\sum_{j=1}^{N}\left(t_j - \lambda_{k-1}^{(-j)}w_j\right)^2 \tag{20}$$

It is known that the jackknife parameter estimator is able to improve the accuracy of parameter estimation [14, 15]. The jackknife parameter estimator for λ_{k-1} given by

$$\lambda_{k-1} = \lambda_{k-1}^{LS} - \frac{N-1}{N}\sum_{j=1}^{N}\lambda_{k-1}^{(-j)} \tag{21}$$

is employed for parameter estimation. Although in general the jackknife parameter estimator is regarded as computationally intensive, the additional computation is minimal in the proposed algorithm. This is because in the FCR procedure, only a minimal number of one parameter $\lambda_{k-1}^{(-j)}$, $(j = 1, ..., N)$ is involved for each candidate term. In addition, most of the calculation in parameter estimation can be regarded as the byproducts of the above leave-one-out cross validation procedure.

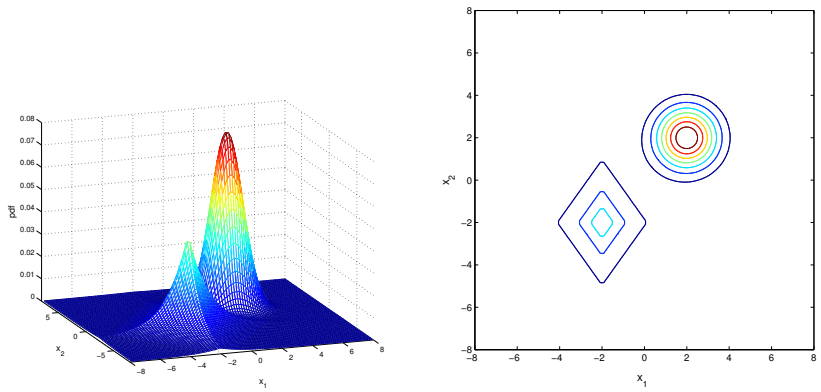
For all model terms which satisfy the constraints $0 \leq \lambda_{k-1}^{LS} \leq 1$, (19)-(21) are repeatedly calculated. Amongst all solutions satisfying the constraints $0 \leq \lambda_{k-1} \leq 1$, the data point that produces the smallest J_k is selected as $\mathbf{x}'_k$ and then used to form kernel $K(\mathbf{x}, \mathbf{x}'_k)$.

The parameters $g_j^{(k)}$ is readily computed by applying the recursion [12], given by

$$\begin{aligned} g_j^{(k)} &= \lambda_{k-1} g_j^{(k-1)}, \qquad j = 1, ..., k-1 \\ g_k^{(k)} &= 1 - \lambda_{k-1} \end{aligned} \tag{22}$$

with $g_1^{(1)} = 1$.

The above procedure iterates for a finite number of forward steps, with k increases by one each step until the final model achieves a satisfactory modelling performance. In this work we terminate the procedure when the accuracy of the sparse kernel density estimator $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ is sufficiently close to that of the PW density estimator $\hat{p}(\mathbf{x}; \mathbf{g}^{Par}, \sigma^{Par})$.



(a) (b)

Fig. 2. (a) The true density and (b) its contours for Example 1.

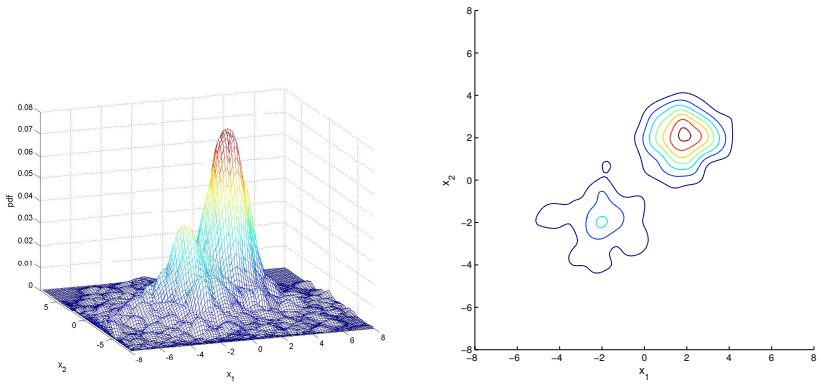


Fig. 3. (a) The Parzen window probability density estimate and (b) its contour for Example 1.

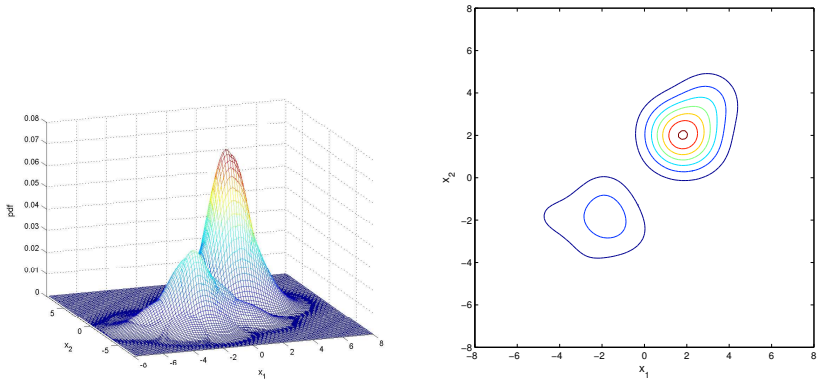


Fig. 4. (a)The proposed FCR-SDC density estimate and (b) its contour for Example 1.

4 An Illustrative Example

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

$$p(\mathbf{x}) = \frac{1}{4\pi}\exp\left(-\frac{(x_1-2)^2}{2}\right)\exp\left(-\frac{(x_2-2)^2}{2}\right) + \frac{0.35}{8}\exp(-0.7|x_1+2|)\exp(-0.5|x_2+2|) \tag{23}$$

The true density and its contour are shown in Figure 2. A data set of $N = 500$ points was randomly drawn from this distribution and used to construct the probability density function $\hat{p}(\mathbf{x}; \mathbf{g}, \sigma)$ using the proposed FCR-SDC approach. The kernel width of $\sigma^{Par} = 0.4$ was empirically found and used in the Parzen window estimate initially, and then the kernel width of $\sigma = 1$ was used in the FCR-SDC algorithm. A separate test data set of $N_{test} = 10000$ points was used for evaluation according to

$$L_1 = \frac{1}{N_{test}}\sum_{k=1}^{N_{test}} |p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k; \mathbf{g}, \sigma)| \tag{24}$$

The experiment was repeated for 100 different random runs. The results of the proposed method in comparison with the PW estimate and the SDC [10] are shown in Table 1. It is shown that the proposed FCR-SDC has comparable accuracy to that of PW, with an average number of required kernels less that 6% of the data samples. In term of model sparsity and accuracy the best performance is that of SDC [10]. The typical Parzen window estimate and the FCR-SDC estimate were depicted in Figures 3–4.

Table 1. Performance of the three kernel density estimates for Example 1.

Method	L_1 test error (mean ± STD)	Kernel numbers (mean ± STD)
PW	$(4.20 \pm 0.8) \times 10^{-3}$	500 ± 0
SDC [10]	$(3.63 \pm 0.8) \times 10^{-3}$	11.9 ± 2.6
proposed FCR-SDC	$(4.26 \pm 0.7) \times 10^{-3}$	33.6 ± 4.7

5 Conclusions

A simple and efficient algorithm has been introduced to construct a kernel model representation using a much smaller number of kernels than the training data set. An illustrative example is used to demonstrate that the models from the

proposed algorithm are able to model the probability density function with comparable accuracy, but with a much sparser representation than Parzen window estimate. Hence the proposed algorithm offers as a viable alternative for sparse probability density function estimation.

References

1. Silverman, B. W.: Density Estimation for Statistics and Data Analysis, Chapman and Hall (1986)
2. Duda, R. O., Hart, P. E.: Pattern Classification and Scene Analysis, J. Wiley (1973)
3. Bishop, C. M.: Neural Networks for Pattern Recognition, Oxford University Press (1995)
4. Parzen, E.: On estimation of a probability density function and mode, The Annals of Mathematical Statistics **33**(1962) 1065–1076
5. McLachlan, G., Peel, D.: Finite Mixture Models, J. Wiley (2000)
6. Weston,J., Gammerman, A., Stitson,M.O., Vapnik,V., Vovk,v. , Watkins,C.: Suppot vector density estimation, in Burges C., Schölkopf B., Smola, A. J.(eds) Advances in Kernel Methods, MIT Press (1999) 293–306.
7. Vapnik, V., Mukherjee,S.: Support vector machine for multivariate density estimation, in Leen, T., Solla, S.,Müller, K. R.(eds) Advances in Neural Infromation Processing Systems. MIT Press (2000) 659–665
8. Girolami, M., He, C.: Probability density estimation from optimally condensed data samples, IEEE Trans. on Pattern Analysis and Machine Intelligence **25** (2003) 1253–1264
9. Choudhury, A.: Fast Machine Learning Algorithms for Large Data, Ph.D. thesis, School of Engineering Sciences, University of Southampton, UK (2002)
10. Chen, S., Hong, X., Harris, C. J.: Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization, IEEE Trans. on Systems, Man and Cybernetics, Part B **34**(2004) 1708–1717
11. Chen, S., Hong, X., Harris, C. J.: An orthogonal forward regression technique for sparse kernel density estimation, Neurocomputing (2007) To appear
12. Hong, X., Harris, C. J: A mixture of experts network structure construction algorithm for modelling and control, Applied Intelligence **16** (2002) 59–69
13. Jordan, M., Jacobs, R. A.: Hierarchical mixtures of experts and the EM algorithm, Neural Computation **6** (1994) 181–214
14. Quenoulle, M.: Notes on bias in estimation, Biometrica **43** (1956) 353–360
15. Miller, R. G.: An unbalanced jacknife, The Annals of Statistics **2**(1974) 880–891