Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Probability Density Function Estimation Using Orthogonal Forward Regression

S. Chen, X. Hong and C.J. Harris

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

I. Introduction

An effective method of estimating the probability density function (PDF) based on a realisation sample drawn from the underlying density is based on a non-parametric approach [1]-[3]. The Parzen window (PW) estimate [1] is a remarkably simple and accurate non-parametric density estimation technique. Because the PW estimate, also known as the kernel density estimate, employs the full data sample set in defining density estimate for subsequent observation, its computational cost for testing scales directly with the sample size, and this imposes a practical difficulty in employing the PW estimator. It also motivates the research on the so-called sparse kernel density estimation techniques. The support vector machine (SVM) method has been proposed as a promising tool for sparse kernel density estimation [4]-[6]. An interesting sparse kernel density estimation technique is proposed in [7]. Similar to the SVM methods, this technique employs the full data sample set as the kernel set and tries to make as many kernel weights to (near) zero as possible, and thus to obtain a sparse representation. The difference with the SVM approach is that it adopts the criterion of the integrated squared error between the unknown underlying density and the kernel density estimate, calculated on the training sample set.

A regression-based sparse kernel density estimation method was reported in [8]. By converting the kernels into the associated cumulative distribution functions and using the empirical distribution function as the desired respone, just like the SVM-based density estimation [4]-[6], this technique transfers the kernel density estimation into a regression problem and it selects sparse kernel density estimates based on an orthogonal forward regression (OFR) algorithm that incrementally minimises the training mean square error (MSE). An additional termination criterion based on the minimum descriptive length [9] or Akaike's information criterion [10] is adopted to stop the kernel density construction procedure. Motivated by our previous work on sparse regression modelling [11],[12], recently we have proposed an efficient construction algorithm for sparse kernel density estimation using the OFR based on the leave-one-out (LOO) MSE and local regularisation [13]. This method is capable of constructing very sparse kernel density estimates with comparable accuracy to that of the full-sample optimised PW density estimate. Moreover, the process is fully automatic and the user is not required to specify when to terminate the density construction procedure [13].

In the works [8],[13], the "regressors" are the cumulative distribution functions of the corresponding kernels and the target function is the empirical distribution function calculated on the training data set. Computing the cumulative distribution functions can be inconvenient and may be difficult for certain types of kernels. We propose a simple regression-based alternative, which directly uses the PW estimate as the desired response. The same OFR algorithm based on the LOO MSE and local regularisation [12] can readily be employed to select a sparse model. Unlike the work [13], we use the multiplicative nonnegative quadratic programming (MNQP) algorithm [14] to compute the final weights of the kernel density estimate, which has a desired property of driving many kernel weights to (near) zero and thus is capable of further reducing the model size. Our empirical results show that this method offers a viable simple alternative to the regression-based sparse kernel density estimation.

II. Regression-Based Approach for Kernel Density Estimation

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

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

with the constraints

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

S. Chen and C.J. Harris are with School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK, E-mails: {sqc,cjh}@ecs.soton.ac.uk

X. Hong is with School of Systems Engineering, University of Reading, Reading RG6 6AY, UK, E-mail: x.hong@reading.ac.uk

and

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

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

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

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

The well-known PW estimate $\hat{p}(\mathbf{x}; \boldsymbol{\beta}_{\rm Par}, \rho_{\rm Par})$ is obtained by setting all the elements of $\boldsymbol{\beta}_{\rm Par}$ to $\frac{1}{N}$. The optimal kernel width $\rho_{\rm Par}$ is typically determined via cross validation [15],[16]. The PW estimate in fact can be derived as the maximum likelihood estimator using the divergence-based criterion [17]. The negative cross-entropy or divergence between the true density $p(\mathbf{x})$ and the estimate $\hat{p}(\mathbf{x}; \boldsymbol{\beta}, \rho)$ is defined as

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

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

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

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

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

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

subject to the constraints (2) and (3), where $\epsilon(\mathbf{x})$ is the modelling error at $\mathbf{x}$. Define $y_k = \hat{p}(\mathbf{x}_k; \boldsymbol{\beta}_{\rm Par}, \rho_{\rm Par})$, $\boldsymbol{\phi}(k) = [K_{k,1} \ K_{k,2} \cdots K_{k,N}]^T$ with $K_{k,i} = K_\rho(\mathbf{x}_k, \mathbf{x}_i)$, and $\epsilon(k) = \epsilon(\mathbf{x}_k)$. Then the model (7) at the data point $\mathbf{x}_k \in \mathcal{D}$ can be expressed as

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

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

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

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

Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}$ be $\boldsymbol{\Phi} = \mathbf{W}\mathbf{A}$, where $\mathbf{W} = [\mathbf{w}_1 \ \mathbf{w}_2 \cdots \mathbf{w}_N]$ with orthogonal columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \ne j$, and

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix}. \tag{10}$$

The regression model (9) can alternatively be expressed as

$$\mathbf{y} = \mathbf{W}\mathbf{g} + \boldsymbol{\epsilon} \tag{11}$$

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

$$\hat{y}_k = \mathbf{w}^T(k)\mathbf{g} \tag{12}$$

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

III. Orthogonal Forward Regression for Sparse Density Estimation

Our aim is to seek a sparse representation for $\hat{p}(\mathbf{x}; \boldsymbol{\beta}, \rho)$ and yet maintaining a comparable test performance to that of the PW estimate. Since this density construction problem is formulated as a standard regression problem, the OFR algorithm based on the LOO MSE and local regularisation [12] can readily be applied to select a sparse model representation. For the completeness, this OFR-LOO-LR algorithm is summarised.

The local regularisation aided least squares solution for the weight parameter vector $\mathbf{g}$ is obtained by minimising the regularised error criterion [18]

$$J_R(\mathbf{g}, \boldsymbol{\lambda}) = \boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \sum_{i=1}^{N} \lambda_i g_i^2 \tag{13}$$

where $\boldsymbol{\lambda} = [\lambda_1 \ \lambda_2 \cdots \lambda_N]^T$ is the regularisation parameter vector, which is optimised based on the evidence procedure [19] with the iterative updating formulas [11],[12],[18]

$$\lambda_i^{\rm new} = \frac{\gamma_i^{\rm old}}{N - \gamma^{\rm old}} \frac{\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}}{g_i^2}, \ 1 \le i \le N, \tag{14}$$

where

$$\gamma_i = \frac{\mathbf{w}_i^T \mathbf{w}_i}{\lambda_i + \mathbf{w}_i^T \mathbf{w}_i} \quad \text{and} \quad \gamma = \sum_{i=1}^{N} \gamma_i. \tag{15}$$

Typically a few iterations are sufficient to find a (near) optimal $\boldsymbol{\lambda}$. The use of multiple-regularisers or local regularisation is capable of providing very sparse solutions [18],[20].

It is highly desired to select a sparse model by directly optimising the model generalisation capability, rather than minimising the training MSE. The algorithm achieves this objective by incrementally minimising the LOO MSE, which

is a measure of the model's generalisation performance [16],[21]-[23]. At the nth stage of the OFR procedure, an n-term model is selected. It can be shown that the LOO test error, denoted as $\epsilon_{n,-k}(k)$, for the selected n-term model is [12],[23]

$$\epsilon_{n,-k}(k) = \frac{\epsilon_n(k)}{\eta_n(k)} \tag{16}$$

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

$$J_n = \frac{1}{N}\sum_{k=1}^{N}\epsilon_{n,-k}^2(k) = \frac{1}{N}\sum_{k=1}^{N}\frac{\epsilon_n^2(k)}{\eta_n^2(k)}. \tag{17}$$

J_n can be computed efficiently due to the fact that the n-term model error $\epsilon_n(k)$ and the associated LOO error weighting $\eta_n(k)$ can be calculated recursively according to [12],[23]

$$\epsilon_n(k) = \epsilon_{n-1}(k) - w_{k,n}g_n \tag{18}$$

and

$$\eta_n(k) = \eta_{n-1}(k) - \frac{w_{k,n}^2}{\mathbf{w}_n^T\mathbf{w}_n + \lambda_n}. \tag{19}$$

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

$$J_{n_s+1} \geq J_{n_s}, \tag{20}$$

yielding a n_s-term sparse model. It is known that J_n is at least locally convex with respect to the model size n [23]. That is, there exists an "optimal" model size n_s such that for $n \leq n_s$ J_n decreases as n increases while the condition (20) holds. This property enables the selection procedure to be automatically terminated with an n_s-term model, without the need for the user to specify a separate termination criterion. The sparse model selection procedure is summarised as follows.

Initialisation: Set all λ_i to 10^{-6} and iteration index to $I = 1$.

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

$$\begin{aligned} &\epsilon_0(k) = y_k \quad \text{and} \quad \eta_0(k) = 1, \ 1 \leq k \leq N, \\ &J_0 = \tfrac{1}{N}\mathbf{y}^T\mathbf{y} = \tfrac{1}{N}\textstyle\sum_{k=1}^{N} y_k^2, \end{aligned} \tag{21}$$

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

Step 2: Update $\boldsymbol{\lambda}$ using (14) and (15) with $N = n_I$. If a pre-set maximum iteration number (e.g. 10) is reached, stop; otherwise set $I+ = 1$ and go to *Step 1*.

In the work [13], the nonnegative constraint (2) is guaranteed by modifying the selection procedure as follows. In the nth stage, a candidate that causes the weight vector $\boldsymbol{\beta}_n$ to have negative elements, if included, will not be considered at all. The unit length condition (3) is met by normalising the final n_s-term model weights. We adopt an alternative means of meeting constraints (2) and (3) by updating the weights of the sparse model using MNQP algorithm [14], which is known to be capable of driving many kernel weights to (near) zero and thus further reducing the size of the kernel density estimate. Denote the design matrix of the selected sparse model as $\mathbf{B} = \boldsymbol{\Phi}_{n_s}^T\boldsymbol{\Phi}_{n_s} = [b_{i,j}]$ and the vector $\mathbf{v} = \boldsymbol{\Phi}_{n_s}^T\mathbf{y} = [v_1 \cdots v_{n_s}]^T$. The MNQP algorithm updates the kernel weights according to

$$c_i^{(t)} = \beta_i^{(t)}\left(\sum_{j=1}^{n_s} b_{i,j}\beta_j^{(t)}\right)^{-1}, \ 1 \leq i \leq n_s, \tag{22}$$

$$h^{(t)} = \left(\sum_{i=1}^{n_s} c_i^{(t)}\right)^{-1}\left(1 - \sum_{i=1}^{n_s} c_i^{(t)}v_i\right), \tag{23}$$

$$\beta_i^{(t+1)} = c_i^{(t)}\left(v_i + h^{(t)}\right), \tag{24}$$

where the superindex $^{(t)}$ denotes the iteration index. The initial condition can be set as $\beta_i^{(0)} = \frac{1}{n_s}$, $1 \leq i \leq n_s$.

IV. Two Numerical Examples

Two examples were used in the simulation to test the proposed algorithm for constructing sparse kernel density (SKD) estimate and to compare its performance with the PW estimator. The value of the kernel width ρ used was determined by test performance via cross validation. For each example, a data set of N randomly drawn samples was used to construct kernel density estimates, and a separate test data set of $N_{test} = 10,000$ samples was used to calculate the L_1 test error for the resulting estimate according to

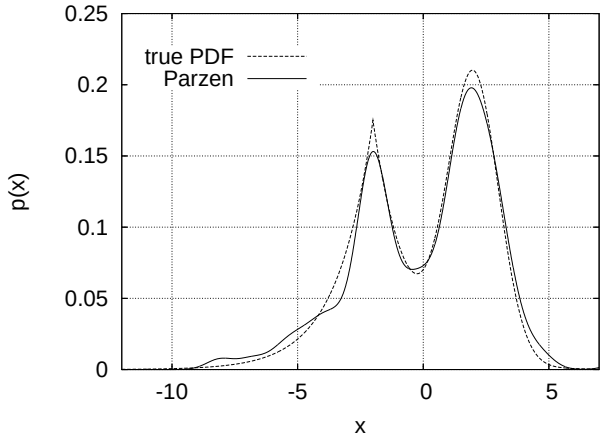
$$L_1 = \frac{1}{N_{test}}\sum_{k=1}^{N_{test}}\left|p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k;\boldsymbol{\beta},\rho)\right|. \tag{25}$$

The experiment was repeated by N_{run} random runs.

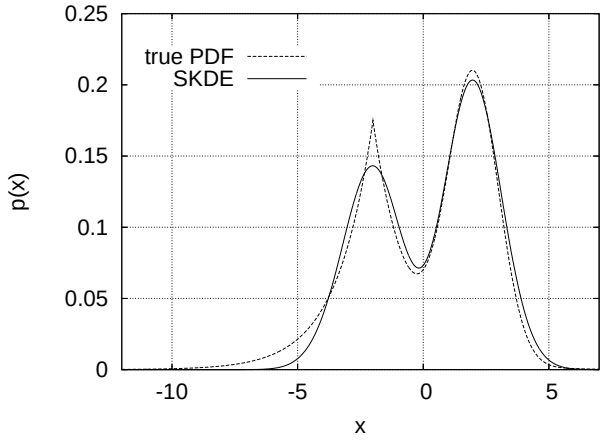
Example 1. This was a one-dimensional example, and the density to be estimated was the mixture of Gaussian and Laplacian given by

$$p(x) = \frac{1}{2\sqrt{2\pi}}e^{-\frac{(x-2)^2}{2}} + \frac{0.7}{4}e^{-0.7|x+2|}. \tag{26}$$

The number of data points for density estimation was $N = 100$. The optimal kernel widths were found to be $\rho = 0.54$ and $\rho = 1.1$ empirically for the PW estimate and the SKD estimate, respectively. The experiment was repeated $N_{run} = 200$ times. Table I compares the performance of the two kernel density estimates, in terms of the L_1 test error and the number of kernels required. Fig. 1 (a) plots a PW estimate obtained while Fig. 1 (b) illustrate a SKD estimate obtained, in comparison with the true distribution. It can be seen that the accuracy of the SKD estimate was comparable to that of the PW estimate for this example, and the combined OFR-LOO-LR and MNQP algorithm achieved sparse estimates with an average kernel number less than 6% of the data samples. The maximum and minimum numbers of kernels over 200 runs were 9 and 2, respectively, for the SKD estimate.



(a)



(b)

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

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

$$p(x,y) = \frac{1}{4\pi}e^{-\frac{(x-2)^2}{2}}e^{-\frac{(y-2)^2}{2}} + \frac{0.35}{8}e^{-0.7|x+2|}e^{-0.5|y+2|}. \quad (27)$$

Fig. 2 shows this density distribution and its contour plot. The estimation data set contained $N = 500$ samples, and the empirically found optimal kernel widths were $\rho = 0.42$ for the PW estimate and $\rho = 1.1$ for the SKD estimate, respectively. The experiment was repeated $N_{run} = 100$ times. Table II lists the L_1 test errors and the numbers of kernels required for the two density estimates. A typical PW estimate and a typical SKD estimate are depicted in Figs. 3 and 4, respectively. Again, for this example, the two density estimates had comparable accuracies, but the SKD estimate method achieved sparse estimates with an average number of required kernels less than 4% of the data samples. The maximum and minimum numbers of kernels over 100 runs were 25 and 8, respectively, for the SKD estimate.

TABLE I

PERFORMANCE OF THE PARZEN WINDOW ESTIMATE AND THE SPARSE KERNEL DENSITY ESTIMATE IN TERMS OF L_1 TEST ERROR AND NUMBER OF KERNELS REQUIRED FOR THE ONE-DIMENSIONAL EXAMPLE OF GAUSSIAN AND LAPLACIAN MIXTURE, QUOTED AS MEAN $\pm$ STANDARD DEVIATION OVER 200 RUNS.

method	L_1 test error	kernel number
PW estimate	$(1.9503 \pm 0.5881) \times 10^{-2}$	100 ± 0
SKD estimate	$(1.9436 \pm 0.6208) \times 10^{-2}$	5.1 ± 1.3

V. CONCLUSIONS

An simple kernel density estimation method has been proposed based on a regression approach with the Parzen window estimate as the target function. The orthogonal forward regression algorithm has been employed to select sparse kernel density estimates, by incrementally minimising a leave-one-out mean square error coupled with local regularisation to further enforce the sparseness of density estimates. The kernel weights are then updated using the MNQP algorithm. The proposed method is simple to implement, and except for the kernel width the algorithm contains no other free parameters that require tuning. The ability of the proposed method to construct a sparse kernel density estimate with a comparable accuracy to that of the full-sample optimised Parzen window estimate has been demonstrated using two examples. The results obtained have shown that the proposed method offers a viable alternative for sparse kernel density estimation.

APPENDIX THE OFR-LOO-LR ALGORITHM

The modified Gram-Schmidt orthogonalisation procedure [24] calculates the $\mathbf{A}$ matrix row by row and orthogonalises $\mathbf{\Phi}$ as follows: at the lth stage make the columns $\boldsymbol{\phi}_j$, $l+1 \leq j \leq N$, orthogonal to the lth column and repeat the operation for $1 \leq l \leq N-1$. Specifically, denoting $\boldsymbol{\phi}_j^{(0)} = \boldsymbol{\phi}_j$, $1 \leq j \leq N$, then for $l = 1, 2, \cdots, N-1$,

$$\left.\begin{aligned} &\mathbf{w}_l = \boldsymbol{\phi}_l^{(l-1)}, \\ &a_{l,j} = \mathbf{w}_l^T \boldsymbol{\phi}_j^{(l-1)} / \left(\mathbf{w}_l^T \mathbf{w}_l\right), \ l+1 \leq j \leq N, \\ &\boldsymbol{\phi}_j^{(l)} = \boldsymbol{\phi}_j^{(l-1)} - a_{l,j}\mathbf{w}_l, \ l+1 \leq j \leq N. \end{aligned}\right\} \quad (28)$$

TABLE II

PERFORMANCE OF THE PARZEN WINDOW ESTIMATE AND THE SPARSE KERNEL DENSITY ESTIMATE IN TERMS OF L_1 TEST ERROR AND NUMBER OF KERNELS REQUIRED FOR THE TWO-DIMENSIONAL EXAMPLE OF GAUSSIAN AND LAPLACIAN MIXTURE, QUOTED AS MEAN $\pm$ STANDARD DEVIATION OVER 100 RUNS.

method	L_1 test error	kernel number
PW estimate	$(4.2453 \pm 0.8242) \times 10^{-3}$	500 ± 0
SKD estimate	$(3.8379 \pm 0.7797) \times 10^{-3}$	15.3 ± 3.9

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

$$\left.\begin{aligned} g_l &= \mathbf{w}_l^T \mathbf{y}^{(l-1)} / \left(\mathbf{w}_l^T \mathbf{w}_l + \lambda_l\right), \\ \mathbf{y}^{(l)} &= \mathbf{y}^{(l-1)} - g_l \mathbf{w}_l, \end{aligned}\right\} 1 \le l \le N. \quad (29)$$

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

$$\boldsymbol{\Phi}^{(l-1)} = \left[\mathbf{w}_1 \cdots \mathbf{w}_{l-1} \ \boldsymbol{\phi}_l^{(l-1)} \cdots \boldsymbol{\phi}_N^{(l-1)}\right]. \quad (30)$$

Let a very small positive number T_z be given, which specifies the zero threshold and is used to automatically avoiding any ill-conditioning or singular problem. With the initial conditions as specified in (21), the lth stage of the selection procedure is given as follows.

Step 1. For $l \le j \le N$:

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

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

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

for $k = 1, \cdots, N$, and

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

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

$$\mathcal{J}_l = \{l \le j \le N \text{ and } j \text{ passes } \textbf{Test}\}$$

Step 2. Find

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

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

Step 3. The selection procedure is terminated with a $(l-1)$-term model, if $J_l \ge J_{l-1}$. Otherwise, perform the orthogonalisation as indicated in (28) to derive the l-th row of $\mathbf{A}$ and to transform $\boldsymbol{\Phi}^{(l-1)}$ into $\boldsymbol{\Phi}^{(l)}$; calculate g_l and update $\mathbf{y}^{(l-1)}$ into $\mathbf{y}^{(l)}$ in the way shown in (29); update the LOO error weightings

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

and go to *Step 1*.

REFERENCES

[1] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol.33, pp.1066–1076, 1962.

[2] C.M. Bishop, *Neural Networks for Pattern Recognition*. Oxford, U.K.: Oxford University Press, 1995.

[3] B.W. Silverman, *Density Estimation*. London: Chapman Hall, 1996.

[4] J. Weston, A. Gammerman. M.O. Stitson, V. Vapnik, V. Vovk and C. Watkins, "Support vector density estimation," in: B. Schölkopf, C. Burges and A.J. Smola, eds., *Advances in Kernel Methods — Support Vector Learning*, MIT Press, Cambridge MA, 1999, pp.293–306.

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

[6] V. Vapnik and S. Mukherjee, "Support vector method for multivariate density estimation," in: S. Solla, T. Leen and K.R. Müller, eds., *Advances in Neural Information Processing Systems*, MIT Press, 2000, pp.659–665.

[7] M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.25, no.10, pp.1253–1264, 2003.

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

[9] M.H. Hansen and B. Yu, "Model selection and the principle of minimum description length," *J. American Statistical Association*, vol.96, no.454, pp.746–774, 2001.

[10] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automatic Control*, vol.AC-19, pp.716–723, 1974.

[11] S. Chen, X. Hong and C.J. Harris, "Sparse kernel regression modeling using combined locally regularized orthogonal least squares and D-optimality experimental design," *IEEE Trans. Automatic Control*, vol.48, no.6, pp.1029–1036, 2003.

[12] S. Chen, X. Hong, C.J. Harris and P.M. Sharkey, "Sparse modeling using orthogonal forward regression with PRESS statistic and regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol.34, no.2, pp.898–911, 2004.

[13] S. Chen, X. Hong and C.J. Harris, "Sparse kernel density construction using orthogonal forward regression with leave-one-out test score and local regularization," *IEEE Trans. Systems, Man and Cybernetics, Part B*, vol.34, no.4, pp.1708–1717, 2004.

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

[15] M. Stone, "Cross validation choice and assessment of statistical predictions," *J. Royal Statistics Society Series B*, vol.36, pp.111–147, 1974.

[16] R.H. Myers, *Classical and Modern Regression with Applications*. 2nd Edition, Boston: PWS-KENT, 1990.

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

[18] S. Chen, "Local regularization assisted orthogonal least squares regression," *Neurocomputing*, vol.69, no.4-6, pp.559–585, 2006.

[19] D.J.C. MacKay, "Bayesian interpolation," *Neural Computation*, vol.4, no.3, pp.415–447, 1992.

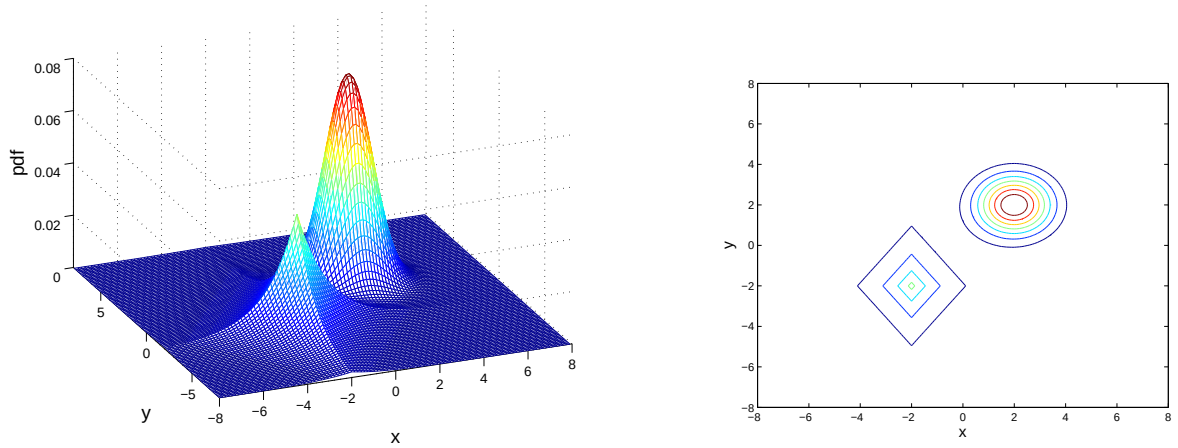
[20] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Research*, vol.1, pp.211–244, 2001.

[21] L.K. Hansen and J. Larsen, "Linear unlearning for cross-validation," *Advances in Computational Mathematics*, vol.5, pp.269–280, 1996.

[22] G. Monari and G. Dreyfus, "Local overfitting control via leverages," *Neural Computation*, vol.14, pp.1481–1506, 2002.

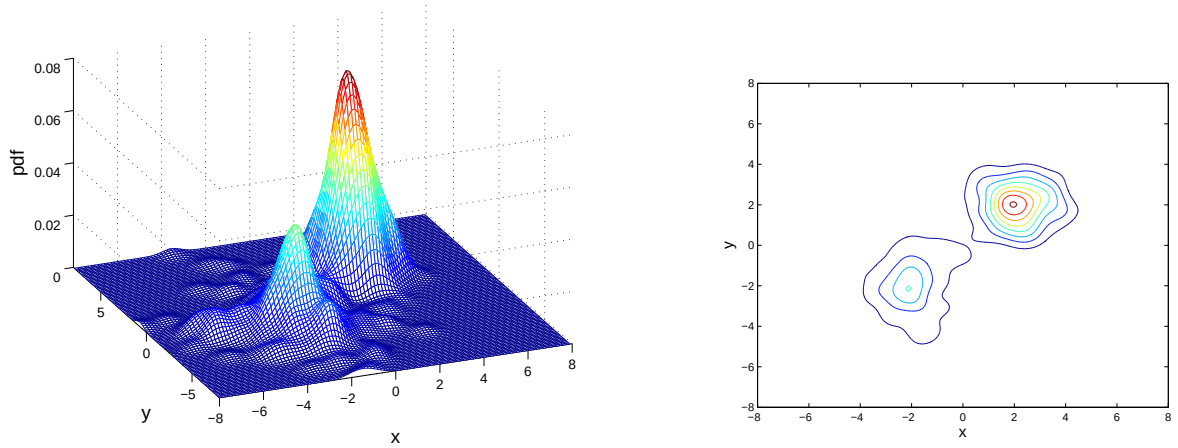
[23] X. Hong, P.M. Sharkey and K. Warwick, "Automatic nonlinear predictive model construction algorithm using forward regression and the PRESS statistic," *IEE Proc. Control Theory and Applications*, vol.150, no.3, pp.245–254, 2003.

[24] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *Int. J. Control*, vol.50, no.5, pp.1873–1896, 1989.



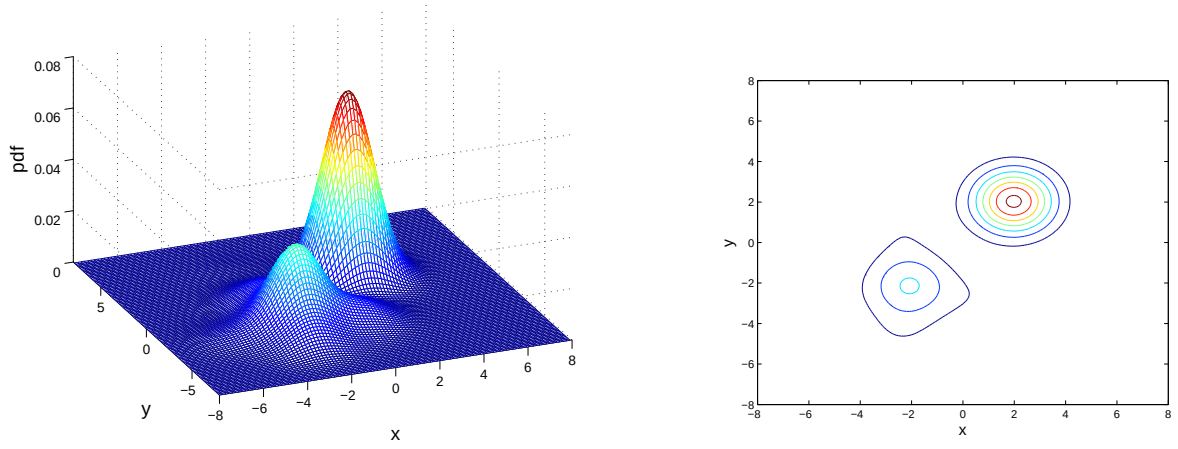
(a) (b)

Fig. 2. True density (a) and contour plot (b) for the two-dimensional example of Gaussian and Laplacian mixture.



(a) (b)

Fig. 3. A Parzen window estimate (a) and contour plot (b) for the two-dimensional example of Gaussian and Laplacian mixture.



(a) (b)

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