Kernel Density Construction Using Orthogonal Forward Regression

S. Chen[†], X. Hong[‡] and C.J. Harris[†]

[†] School of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, U.K.

[‡]Department of Cybernetics
University of Reading, Reading, RG6 6AY, U.K.

Abstract

An automatic algorithm is derived for constructing kernel density estimates based on a regression approach that directly optimizes generalization capability. Computational efficiency of the density construction is ensured using an orthogonal forward regression, and the algorithm incrementally minimizes the leave-one-out test score. Local regularization is incorporated into the density construction process to further enforce sparsity. Examples are included to demonstrate the ability of the proposed algorithm to effectively construct a very sparse kernel density estimate with comparable accuracy to that of the full sample Parzen window density estimate.

I. Introduction

Estimation of probability density functions is a recurrent theme in machine learning and many fields of engineering. A well-known non-parametric density estimation technique is the classical Parzen window estimate [1], which is remarkably simple and accurate. The particular problem associated with the Parzen window estimate however is the computational cost for testing which scales directly with the sample size, as the Parzen window estimate employs the full data sample set in defining density estimate for subsequent observation. Recently, the support vector machine (SVM) has been proposed as a promising tool for sparse kernel density estimation [2],[3].

Motivated by our previous work on sparse data modeling [4],[5], we propose an efficient algorithm for sparse kernel density estimation using an orthogonal forward regression (OFR) based on leave-one-out (LOO) test score and local regularization. This construction algorithm is fully automatic and the user does not require to specify any criterion to terminate the density construction procedure. We will refer to this algorithm as the sparse density construction (SDC) algorithm. Some examples are used to illustrate the ability of this SDC algorithm to construct efficiently a sparse density estimate with comparable accuracy to that of the Parzen window estimate.

II. Kernel density estimation as regression

Given $\mathcal{D} = \{\mathbf{x}_k\}_{k=1}^N$ drawn from an unknown density $p(\mathbf{x})$, where the data samples $\mathbf{x}_k = [x_{1,k}\ x_{2,k} \cdots x_{m,k}]^T \in \mathcal{R}^m$ are assumed to be independently identically distributed, the task is to estimate $p(\mathbf{x})$ using the kernel density estimate of the form

$$\hat{p}(\mathbf{x}) = \sum_{k=1}^{N} \beta_k K(\mathbf{x}, \mathbf{x}_k) \tag{1}$$

with the constraints

$$\beta_k \geq 0, \quad k = 1, 2, \cdots, N, \quad \text{and} \quad \sum_{k=1}^{N} \beta_k = 1 \tag{2}$$

In this study, the kernel function is assumed to be the Gaussian function of the form

$$K(\mathbf{x}, \mathbf{x}_k) = \frac{1}{(2\pi\rho^2)^{m/2}} \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_k\|^2}{2\rho^2}\right) \tag{3}$$

where ρ is a common kernel width. The well-known Parzen window estimate [1] is obtained by setting $\beta_k = \frac{1}{N}$ for all k. Our aim is to seek a spare representation for $\hat{p}(\mathbf{x})$, i.e. with most of β_k being zero and yet maintaining a comparable test performance or generalization capability to that of the full sample optimized Parzen window estimate.

Following the approach [2],[3], the kernel density estimation problem is posed as the following regression modeling problem

$$f(\mathbf{x}; N) = \sum_{k=1}^{N} \beta_k q(\mathbf{x}, \mathbf{x}_k) + \epsilon(k) \tag{4}$$

subject to (2), where the empirical distribution function $f(\mathbf{x}; N)$ is defined by

$$f(\mathbf{x}; N) = \frac{1}{N} \sum_{k=1}^{N} \prod_{j=1}^{m} \theta\left(x_j - x_{j,k}\right) \tag{5}$$

with

$$\theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{6}$$

the "regressor" $q(\mathbf{x}, \mathbf{x}_k)$ is given by

$$q(\mathbf{x}, \mathbf{x}_k) = \int_{-\infty}^{\mathbf{x}} K(\mathbf{u}, \mathbf{x}_k)\, d\mathbf{u} = \prod_{j=1}^{m} \left(1 - Q\left(\frac{x_j - x_{j,k}}{\rho}\right)\right) \tag{7}$$

with the usual Gaussian Q-function, and $\epsilon(k)$ denotes the modeling error. Let $\boldsymbol{\beta} = [\beta_1\ \beta_2 \cdots \beta_N]^T$, $f_k = f(\mathbf{x}_k; N)$ and $\boldsymbol{\phi}(k) = [q_{k,1}\ q_{k,2} \cdots q_{k,N}]^T$ with $q_{k,i} = q(\mathbf{x}_k, \mathbf{x}_i)$. Then the regression model (4) for the data point $\mathbf{x}_k \in \mathcal{D}$ can be expressed as

$$f_k = \hat{f}_k + \epsilon(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\beta} + \epsilon(k) \tag{8}$$

Furthermore, the regression model (4) over the training data set $\mathcal{D}$ can be written together in the matrix form

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

with the following additional notations $\boldsymbol{\Phi} = [q_{i,k}] \in \mathcal{R}^{N \times N}$, with $1 \leq i, k \leq N$, $\boldsymbol{\epsilon} = [\epsilon(1)\ \epsilon(2) \cdots \epsilon(N)]^T$, and $\mathbf{f} = [f_1\ f_2 \cdots f_N]^T$. For convenience, we will denote the regression matrix $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2 \cdots \boldsymbol{\phi}_N]$ with $\boldsymbol{\phi}_k = [q_{1,k}\ q_{2,k} \cdots q_{N,k}]^T$. $\boldsymbol{\phi}_k$ should not be confused with $\boldsymbol{\phi}(k)$ (the former is the kth column of $\boldsymbol{\Phi}$, and the latter the kth row of $\boldsymbol{\Phi}$).

Let an orthogonal decomposition of the regression matrix $\mathbf{\Phi}$ be

$$\mathbf{\Phi} = \mathbf{WA} \tag{10}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & a_{1,2} & \cdots & a_{1,N} \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{N-1,N} \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{11}$$

and $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_N]$ with columns satisfying $\mathbf{w}_i^T \mathbf{w}_j = 0$, if $i \neq j$. The regression model (9) can alternatively be expressed as

$$\mathbf{f} = \mathbf{Wg} + \boldsymbol{\epsilon} \tag{12}$$

where the orthogonal weight vector $\mathbf{g} = [g_1\ g_2 \cdots g_N]^T$ satisfies the triangular system $\mathbf{A}\boldsymbol{\beta} = \mathbf{g}$. The model $\hat{f}_k$ is equivalently expressed by

$$\hat{f}_k = \mathbf{w}^T(k)\mathbf{g} \tag{13}$$

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

III. The sparse density construction

Let $\boldsymbol{\lambda} = [\lambda_1\ \lambda_2 \cdots \lambda_N]^T$ be the regularization parameter vector associated with $\mathbf{g}$. If an n-term model is selected from the full model (12), the LOO test error [6]–[9], denoted as $\epsilon_{n,-k}(k)$, for the selected n-term model can be shown to be [9],[5]

$$\epsilon_{n,-k}(k) = \frac{\epsilon_n(k)}{\eta_n(k)} \tag{14}$$

where $\epsilon_n(k)$ is the n-term modeling error and $\eta_n(k)$ is the associated LOO error weighting given by

$$\eta_n(k) = 1 - \sum_{i=1}^{n} \frac{w_{k,i}^2}{\mathbf{w}_i^T \mathbf{w}_i + \lambda_i} \tag{15}$$

The mean square LOO error for the model with a size n is defined by

$$J_n = E\left[\epsilon_{n,-k}^2(k)\right] = \frac{1}{N} \sum_{k=1}^{N} \frac{\epsilon_n^2(k)}{\eta_n^2(k)} \tag{16}$$

This LOO test score can be computed efficiently due to the fact that the n-term model error $\epsilon_n(k)$ and the associated LOO error weighting can be calculated recursively according to

$$\epsilon_n(k) = f_k - \sum_{i=1}^{n} w_{k,i} g_i = \epsilon_{n-1}(k) - w_{k,n} g_n \tag{17}$$

$$\eta_n(k) = 1 - \sum_{i=1}^{n} \frac{w_{k,i}^2}{\mathbf{w}_i^T \mathbf{w}_i + \lambda_i} = \eta_{n-1}(k) - \frac{w_{k,n}^2}{\mathbf{w}_n^T \mathbf{w}_n + \lambda_n} \tag{18}$$

The model selection procedure is carried as follows: at the nth stage of selection, a model term is selected among the remaining n to N candidates if the resulting n-term model produces the smallest LOO test score J_n. It has been shown in [9] that there exists an "optimal" model size n_s such that for $n \leq n_s$ J_n decreases as n increases while for $n \geq n_s + 1$ J_n increases as n increases. This property enables the selection procedure to be automatically terminated with an n_s-term model when $J_{n_s+1} > J_{n_s}$, without the need for the user to specify a separate termination criterion. The iterative SDC procedure based on this OFR with LOO test score and local regularization is summarized:

Initialization. Set λ_i, $1 \leq i \leq N$, to the same small positive value (e.g. 0.001). Set iteration $I = 1$.

Step 1. Given the current $\boldsymbol{\lambda}$ and with the following initial conditions

$$\epsilon_0(k) = f_k \text{ and } \eta_0(k) = 1, k = 1, 2, \cdots, N, \quad J_0 = \mathbf{f}^T \mathbf{f}/N$$

use the procedure as described in [4],[5] to select a subset model with n_I terms.

Step 2. Update $\boldsymbol{\lambda}$ using the evidence formula as described in [4],[5]. If $\boldsymbol{\lambda}$ remains sufficiently unchanged in two successive iterations or a pre-set maximum iteration number (e.g. 10) is reached, stop; otherwise set $I+ = 1$ and go to *Step 1*.

The computational complexity of the above algorithm is dominated by the 1st iteration. After the 1st iteration, the model set contains only $n_1 (\ll N)$ terms, and the complexity of the subsequent iteration decreases dramatically. As a probability density, the constraint (2) must be met. The non-negative condition is ensured during the selection with the following simple measure. Let $\boldsymbol{\beta}_n$ denote the weight vector at the nth stage. A candidate that causes $\boldsymbol{\beta}_n$ to have negative elements, if included, will not be considered at all. The unit length condition is easily met by normalizing the final n_s-term model weights.

IV. Numerical examples

In order to remove the influence of different ρ values to the quality of the resulting density estimate, the optimal value for ρ, found empirically by cross validation, was used. In each case, 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_2 test error for the resulting estimate according to

$$L_2 = \frac{1}{N_{test}} \sum_{k=1}^{N_{test}} \left(p(\mathbf{x}_k) - \hat{p}(\mathbf{x}_k)\right)^2 \tag{19}$$

The experiment was repeated by 100 different random runs for each example.

Example 1. This was a 1-D example with the density to be estimated given by

$$p(x) = \frac{1}{2\sqrt{0.5\pi}} \exp\left(-\frac{(x+4)^2}{0.5}\right) + \frac{1}{2\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) \tag{20}$$

TABLE I

PERFORMANCE OF THE PARZEN WINDOW ESTIMATE AND THE PROPOSED SPARSE DENSITY CONSTRUCTION ALGORITHM FOR THE ONE-DIMENSIONAL EXAMPLE. STD: STANDARD DEVIATION.

method	L_2 test error (mean $\pm$ STD)	kernel number (mean $\pm$ STD)
Parzen	$(1.2663 \pm 0.8243) \times 10^{-3}$	200 ± 0
SDC	$(1.4301 \pm 1.2456) \times 10^{-3}$	6.1 ± 1.3

The number of data points for density estimation was $N = 200$. The optimal kernel widths were found to be $\rho = 0.3$ and $\rho = 0.5$ empirically for the Parzen window estimate and the SDC estimate, respectively. Table I compares the performance of the two kernel density construction methods, in terms of the L_2 test error and the number of kernels required. Fig. 1 (a) depicts the Parzen window estimated obtained in a run while Fig. 1 (b) shows the density obtained by the SDC algorithm in a run, in comparison with the true distribution. It is seen that the accuracy of the SDC algorithm was comparable to that of the Parzen window estimate, and the algorithm realized very sparse estimates with an average kernel number less than 4% of the data samples.

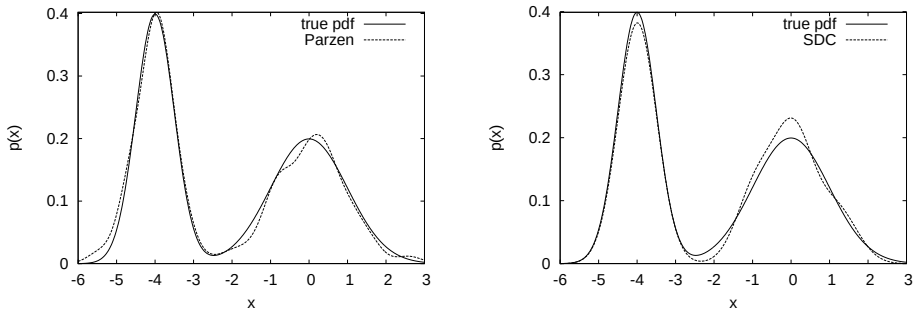
Example 2. In this 6-D example, the underlying density to be estimated was given by

$$p(\mathbf{x}) = \frac{1}{3\,(2\pi)^{6/2}} \left\{ \frac{1}{\det|\mathbf{\Gamma}_1|} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T \mathbf{\Gamma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right) + \right.$$
$$\frac{1}{\det|\mathbf{\Gamma}_2|} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T \mathbf{\Gamma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right) +$$
$$\left. \frac{1}{\det|\mathbf{\Gamma}_3|} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_3)^T \mathbf{\Gamma}_3^{-1}(\mathbf{x}-\boldsymbol{\mu}_3)\right) \right\} \tag{21}$$

with

$$\begin{aligned} \boldsymbol{\mu}_1 &= [1.0\ 1.0\ 1.0\ 1.0\ 1.0\ 1.0]^T \\ \mathbf{\Gamma}_1 &= \text{diag}\{1.0, 2.0, 1.0, 2.0, 1.0, 2.0\} \end{aligned} \tag{22}$$

$$\begin{aligned} \boldsymbol{\mu}_2 &= [-1.0\ -1.0\ -1.0\ -1.0\ -1.0\ -1.0]^T \\ \mathbf{\Gamma}_2 &= \text{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\} \end{aligned} \tag{23}$$



(a) (b)

Fig. 1. (a) true density (solid) and a Parzen window estimate (dashed), and (b) true density (solid) and a sparse density construction estimate (dashed), for the one-dimensional example.

TABLE II

PERFORMANCE OF THE PARZEN WINDOW ESTIMATE AND THE PROPOSED SPARSE DENSITY CONSTRUCTION ALGORITHM FOR THE SIX-DIMENSIONAL EXAMPLE. STD: STANDARD DEVIATION.

method	L_2 test error (mean $\pm$ STD)	kernel number (mean $\pm$ STD)
Parzen	$(3.38 \pm 0.42) \times 10^{-9}$	600 ± 0
SDC	$(5.47 \pm 2.76) \times 10^{-9}$	14.9 ± 2.1

$$\begin{aligned} \boldsymbol{\mu}_3 &= [0.0\ 0.0\ 0.0\ 0.0\ 0.0\ 0.0]^T \\ \boldsymbol{\Gamma}_3 &= \text{diag}\{2.0, 1.0, 2.0, 1.0, 2.0, 1.0\} \end{aligned} \tag{24}$$

The estimation data set contained $N = 600$ samples. The optimal kernel width was found to be $\rho = 0.6$ for the Parzen window estimate and $\rho = 1.1$ for the SDC estimate, respectively. The results obtained by the two density construction algorithms are summarized in Table II. It can be seen that the SDC algorithm achieved a similar accuracy to that of the Parzen window estimate with a much sparser representation. The average number of required kernels for the SDC method was less than 3% of the data samples.

V. CONCLUSIONS

An efficient algorithm has been proposed for obtaining sparse kernel density estimates based on an OFR procedure that incrementally minimizes the LOO test score, coupled with local regularization. The proposed method is simple to implement and computationally efficient, and except for the kernel width the algorithm contains no other free parameters that require tuning. The ability of the proposed algorithm to construct a very sparse kernel density estimate with a comparable accuracy to that of the full sample Parzen window estimate has been demonstrated using two examples. The results obtained have shown that the proposed method provides a viable alternative for sparse kernel density estimation in practical applications.

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] 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.

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

[4] 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.

[5] 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*, to appear, 2004.

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

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

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

[9] 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.