Nonparametric Multivariate Density Estimation: A Comparative Study

Jenq-Neng Hwang, *Member, IEEE,* Shyh-Rong Lay, and Alan Lippman

Abstract— **This paper algorithmically and empirically studies two major types of nonparametric multivariate density estimation techniques, where no assumption is made about the data being drawn from any of known parametric families of distribution. The first type is the popular kernel method (and several of its variants) which uses locally tuned radial basis (e.g., Gaussian) functions to interpolate the multidimensional density; the second type is based on an exploratory projection pursuit technique which interprets the multidimensional density through the construction of several 1-D densities along highly "interesting" projections of multidimensional data. Performance evaluations using training data from mixture Gaussian and mixture Cauchy densities are presented. The results show that the curse of dimensionality and the sensitivity of control parameters have a much more adverse impact on the kernel density estimators than on the projection pursuit density estimators.**

I. Introduction

IN signal-processing applications, most algorithms work properly if the probability densities of the multivariate signals (or noises) are known. Unfortunately, in reality these densities are usually not available, and parametric or nonparametric estimation of the densities becomes critically needed. Unlike the parametric density estimation where assumptions are made about the parametric form of the distribution that generates the data, the nonparametric density estimation makes less rigid assumptions about the distribution of the data [24].

A probability density function (pdf), $f(\mathbf{y})$, of a p-dimensional data $\mathbf{y}$ is a continuous and smooth function which satisfies the following positivity and integrate-to-one constraints

$$f(\mathbf{y}) \geq 0, \quad \int_{R^p} f(\mathbf{y}) d\mathbf{y} = 1. \tag{1}$$

Given a set of p-dimensional observed data $\{\mathbf{y}_n, n = 1, \cdots, N\}$, the task of multivariate density estimation is to find an estimated function $\hat{f}$ which "best" approximates the true probability density function f. On the other hand, a probability mass function (pmf) is a discrete function which also satisfies the positivity and sum-to-one constraints and has been successful in some classification and regression applications [2], [19]. The success of a pmf results from several well developed clustering algorithms (e.g., [16]) which cluster multidimensional data $\{\mathbf{y}_n, n = 1, \cdots, N\}$ into several centroids $\{\mathbf{m}_k, k = 1, \cdots, K\}$ and the pmf can thus be obtained by estimating the proportion c_k of data population in each cluster. In this paper, we are only dealing with the continuous pdf which has been successfully applied in applications like classifier design [28], image restoration and compression [20], [21], etc.

Traditionally and statistically, the pdf is constructed by locating a Gaussian kernel at each observed datum, e.g., the fixed-width kernel density estimator (FKDE) and the adaptive kernel density estimator (AKDE). Although the FKDE, which constructs a density by placing fixed width kernels at all of the observed data, is widely used for nonparametric density estimation, this method normally suffers from several practical drawbacks [25]. For example, the inability to deal satisfactorily with tails of distributions without oversmoothing the main part of the density. The other is the curse of dimensionality, i.e., the exponentially increasing sample size required to effectively estimate a multivariate density when the number of dimensions increases.

The AKDE [1], [25] is thus introduced to improve the performance of an FKDE. Similar to an FKDE, the AKDE constructs a density by placing kernels at all of the observed data. Unlike an FKDE that uses kernels of fixed width, an AKDE allows the widths of kernels to vary from one point to another. Although the AKDE slightly improves the estimation capability of an FKDE, it does not reduce the high cost incurred in computation and memory storage commonly required in an FKDE.

To overcome the problem of high cost in computation and memory storage, a (clustered) radial basis function (RBF) based kernel density estimator, named RBF network, can be used [14], [20], [21]. The RBF network uses a reduced number of (radial basis) kernels, with each kernel being representative of a cluster of training data, to approximate the unknown density function. This method is often referred as mixture (Gaussian) modeling [23]. The RBF networks are also widely used in regression and classification applications [18]. Similar to the construction of a pmf, the construction of an RBF network requires the determination of the cluster centroids $\{\mathbf{m}_k\}$. Furthermore, the estimates of the data correlation and proportion within or between clusters are translated into the bandwidths (as well as orientations) and heights of the (interpolating) Gaussian kernels to be deployed on the cluster centroids so that a smooth and continous pdf can be constructed. The determination of centroids and associated kernel parameters can be accomplished in two-

Manuscript received August 23, 1993; revised April 12, 1994. This work was supported by grants from the National Science Foundation under Grant No. ECS-9014243, from NASA under Contract No. NAGW-1702, and by a postdoctoral fellowship from Office of Naval Research under Grant No. N00014-90-J-1478. The associate editor coordinating the review of this paper and approving it for publication was Dr. R. D. Preuss.

The authors are with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA.

IEEE Log Number 9403752.

stage batch process or can be done simulataneously in an iterative manner. The two-stage batch process starts with acquiring a satisfactory set of cluster centroids, then determine the kernel bandwidths, orientations, and heights through batch statistical analysis in the sense of maximum likelihood [14], [20], [21]. The iterative kernel deploying approaches for construction of RBF density estimators use the iterative expectation-and-maximization (EM) algorithm [17], [23], [27], a maximum likelihood optimization procedure, by treating the cluster label that indicates which kernel a datum belong to as missing data and maximizes the likelihood with respect to the kernel parameters (centroids, bandwidths, orientations, and heights). There are some drawbacks of this approach, namely, slow convergence and the sensitivity of the initial label parameter guesses. In some cases where the likelihood is unbounded in certain parameter space, the procedure will diverge if the initial guess is too close to this space. Like most optimization approaches, the EM algorithm also suffers the local optimum issues. In this paper, we only focus on the discussion of two-stage batch process for RBF network construction.

In two-stage batch construction of an RBF network, sequential and batch clustering algorithms are commonly used in determining the cluster centroids [4], [16], [18]. These clustering algorithms perform poorly in the presence of probabilistic outlying data or data of large variations of dynamic range among dimensions, the latter imposing high sensitivity to the selection of distance measures in the clustering. To overcome these difficulties, statistical data sphering technique combined with a centroid splitting generalized Lloyd clustering technique (also known as the LBG algorithm [16]) is used in the robust RBF density estimator construction. This robust construction method has been successfully applied to classification tasks [15].

Although the robust RBF construction technique can overcome some of the difficulties encountered in using conventional RBF networks for density estimation, it still can not overcome the drawback of the estimators' performance being too sensitive to the settings of some control parameters, e.g., the number of kernels used, the locations of kernels, the orientation of kernels, the kernel smoothing parameters, the excluding threshold radius for data sphering, the size of training data, etc. We are thus motivated to study the statistical projection pursuit density estimation technique [5], [7]. In contrast to the locally tuned kernel methods, where data are analyzed directly in high dimensional space around the vicinity of the kernel centers, a projection pursuit method globally projects the data onto 1-D or 2-D subspaces, and analyzes the projected data in these low dimensional subspaces to construct the multivariate density. More specifically, the projection pursuit first defines some index of interest of a projected configuration (instead of using the variance adopted by the principal component analysis) and then uses a numerical optimization technique to find the projections of most interest [12], [17]. The projection index adopted for density estimation is the degree of departure of the projection data from normality. This technique has been applied to exploratory multivariate data analysis in some statistical tools [13].

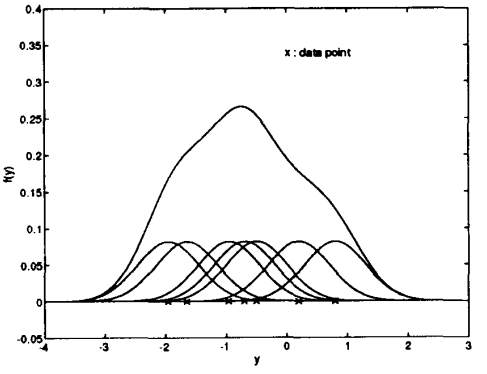

Fig. 1. An example of fixed-width kernel density estimation.

This paper is organized as follows: Section II presents various versions of kernel based density estimators: the fixed-width kernel method, the adaptive kernel method, and the robust RBF method. Section III discusses the algorithms used for implementing the projection pursuit density estimator. Extensive comparative simulations and discussions of results are performed in Section IV, which is followed by the concluding remarks in Section V.

II. Kernel—Based Density Estimation

Given a set of N p-dimensional training data $\{\mathbf{y}_n, n = 1, \cdots, N\}$, a multivariate fixed-width kernel density estimator (FKDE), with the kernel function ϕ and a fixed (global) kernel width parameter h, gives the estimated density $\hat{f}(y)$ for a multivariate data $\mathbf{y} \in R^p$ based on

$$\hat{f}(\mathbf{y}) = \frac{1}{Nh^p} \sum_{n=1}^{N} \phi\left(\frac{1}{h}(\mathbf{y} - \mathbf{y}_n)\right). \tag{2}$$

The kernel function ϕ should be chosen to satisfy

$$\phi(\mathbf{y}) \geq 0, \quad \text{and} \int_{R^p} \phi(\mathbf{y}) d\mathbf{y} = 1. \tag{3}$$

A popular choice of ϕ is the Gaussian kernel

$$\phi(\mathbf{y}) = (2\pi)^{-p/2} \exp\left(-\frac{1}{2}\mathbf{y}^T\mathbf{y}\right),$$

which is a symmetric kernel with its value smoothly decaying away from the kernel center. An illustration of FKDE using a small training data set of size 7 is given in Fig. 1.

Normally, the observed data is not equally spread in all directions. It is thus highly desired to pre-scale the data to avoid extreme differences of spread in the various coordinate directions. One attractive approach [8] is to first sphere (whiten) the data by a linear transformation yielding data with zero mean and unit covariance matrix, then apply (2) to the sphered data. More specifically, given a set of p-dimensional observed data, $\{\mathbf{y}\}$, we can define the sphered data $\mathbf{z}$ of $\mathbf{y}$ to be

$$\mathbf{z} = \mathbf{S}^{-1/2}(\mathbf{y} - E\mathbf{y}). \tag{4}$$

where the expectation E is evaluated through the sample mean, and $\mathbf{S} \in R^{p \times p}$ is the data covariance matrix

$$\mathbf{S} = E[(\mathbf{y} - E\mathbf{y})(\mathbf{y} - E\mathbf{y})^T] = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{5}$$

or

$$\mathbf{S}^{-1/2} = \mathbf{U}\mathbf{D}^{-1/2}\mathbf{U}^T. \tag{6}$$

Note that $\mathbf{U}$ is an orthonormal matrix and $\mathbf{D}$ is a diagonal matrix. Robust statistics methods [11] can be used for the derivation of the data covariance matrix $\mathbf{S}$.

It can be easily shown that after sphering $[E\mathbf{z}] = \mathbf{0}$ and $E[\mathbf{z}\mathbf{z}^T] = \mathbf{I}$ (the identity matrix). The resulting FKDE for the sphered data performs a more sophisticated density estimation

$$\hat{f}(\mathbf{z}) = \frac{1}{Nh^p} \sum_{n=1}^{N} \phi\left(\frac{1}{h}(\mathbf{z} - \mathbf{z}_n)\right) \tag{7}$$

$$\hat{f}(\mathbf{y}) = \frac{(\det S)^{-1/2}}{Nh^p} \sum_{n=1}^{N} \phi\left(\frac{1}{h}\mathbf{S}^{-1/2}(\mathbf{y} - \mathbf{y}_n)\right). \tag{8}$$

An optimal kernel width h^* for an FKDE can be determined through the minimization of mean integrated squared error (MISE) [25]. For example, the h^* for Gaussian kernels was proposed [25] for estimating normally distributed data with unit covariance

$$h^* = AN^{-\frac{1}{(p+4)}}, \quad \text{where } A = [4/(2p+1)]^{\frac{1}{(p+4)}}. \tag{9}$$

More complicated methods for determining the kernel width, such as the least-square cross-validation method [25], are also available with increasing complication and computation.

The probabilistic neural network (PNN), introduced by Specht [26], is a multivariate kernel density estimator with fixed kernel width. The kernel width of a PNN is commonly obtained by a trial-and-error procedure. A small value of h causes the estimated density function to have distinct modes corresponding to the locations of the observed data. A larger value of h produces a greater degree of interpolation between data points.

Although the FKDE's are widely used for nonparametric density estimation, they normally suffer from several practical drawbacks [25]: e.g., the inability to deal satisfactorily with tails of distributions without oversmoothing the main part of the density, and the curse of dimensionality that calls for the requirement of an exponentially increasing sample size to estimate the multivariate density when the number of dimensions increases. The latter drawback also reflects a potential computational burden in using the density estimator after its construction due to the fact that for every observed training datum a kernel is deployed on and an extra term is added in (2).

A. Adaptive Kernel Density Estimator

An improved alternative to an FKDE is the adaptive kernel density estimator (AKDE) [25]. Similar to an FKDE, an AKDE constructs the density by placing a kernel on every observed datum, but it allows the kernel width to vary from one point to another. The intent is to use different widths of kernels in regions of different smoothness. This method adopts a two-step algorithm for computing a data-adaptive kernel width. The algorithm can be summarized as follows:

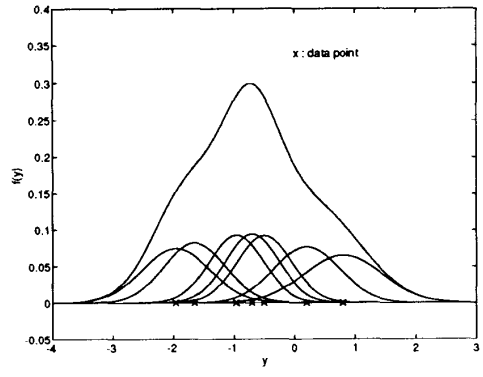

Fig. 2. An example of adaptive kernel density estimation.

Step 0: Sphere the observed data $\{\mathbf{y}_n\}$ to be $\{\mathbf{z}_n\}$, so that $E[\mathbf{z}] = \mathbf{0}$ and $E[\mathbf{z}\mathbf{z}^T] = \mathbf{I}$.

Step 1: Find a pilot estimate $\tilde{f}(\mathbf{z})$ that satisfies $\tilde{f}(\mathbf{z}_n) > 0, \forall n$.

Step 2: Set the local width factor λ_n to be $(\tilde{f}(\mathbf{z}_n)/g)^{-\gamma}$, where g is the geometric mean of $\tilde{f}(\mathbf{z})$, i.e., $\log g = \frac{1}{N}\sum_{i=1}^{N} \log \tilde{f}(\mathbf{z}_i)$, and γ is a user defined sensitivity parameter satisfying $0 \le \gamma \le 1$.

Step 3: Construct the adaptive kernel estimate $\hat{f}(\mathbf{z})$ by

$$\hat{f}(\mathbf{z}) = \frac{1}{N} \sum_{n=1}^{N} h^{-p}\lambda_n^{-p}\phi\left\{\frac{1}{h\lambda_n}(\mathbf{z} - \mathbf{z}_n)\right\} \tag{10}$$

where h is still the global width parameter used in (2). A natural pilot estimate would be a kernel estimate with fixed optimal kernel width (see (9)). The larger the γ, the more sensitive the performance will be to the selection of pilot density. It is quite common to set $\gamma = \frac{1}{2}$ [1], [25]. The estimate $\hat{f}$ of an AKDE using the small data set of size 7 is illustrated in Fig. 2.

B. Radial Basis Function Density Estimator

Due to the requirement that a kernel is placed at every observed datum, the implementations of FKDE's and AKDE's require too many kernels when the number of training data is huge. A density estimator, such as the radial basis function (RBF) network, which uses a reduced number of (radial basis) kernels with each kernel being representative of a cluster of training data, is highly desired. As shown in Fig. 3, the training data are grouped into three clusters, and the density is estimated through constructing three kernels of different heights and widths on each cluster center.

Several *supervised* RBF networks were recently introduced [18] for classification and data regression applications. For example, Moody and Darken proposed a hybrid learning method [18] which used a self-organizing adaptive K-mean clustering algorithm to locate the positions of kernel functions, and then a "nearest-neighbor" heuristic to determine the kernel widths. This heuristic varies the widths in order to achieve a

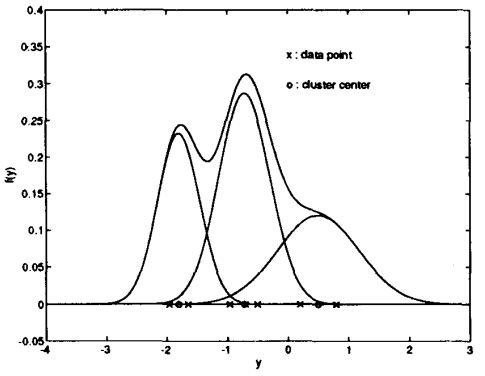

Fig. 3. An example of radial basis function-based density estimation.

certain amount of response overlap between each unit and its neighbors. Finally, a least mean squares (LMS) supervised training rule is used for the updating of the heights of the deployed kernels.

1) Data Sphering and Outlier Removing: Since a density estimation task is an unsupervised learning task, a few modifications of the learning procedures for RBF classification/regression networks are needed. Since an RBF network possesses a local tuning property, the positions of the kernels searched by the clustering algorithm should cover the areas which are most representative of the data in the region around the cluster centers. Unfortunately, most clustering methods are vulnerable to the data outliers which are generated by long-tailed portion of the density. In the classification applications, some outlying training data are useful and can be carefully regularized to increase the generalization capability of the classifiers [22]. However, the outlying training data in a density estimation application usually carry very little information about the density and do not represent any meaningful isolated class as in a classification application. If the RBF network construction is based on the FKDE or AKDE, where a symmetric kernel is placed on every observed training data, the outlying data will not play a significant role in approximating the true density since the amount of outlying data are usually quite small. On the other hand, when clustering techniques are adopted to reduce the number of kernels deployed in an RBF construction, the outlying data play a more significant role. More specifically, most clustering algorithms are types of least squares estimators which are sensitive to outliers. Therefore, we are motivated to remove the outliers after the data sphering and before the data clustering processes [7], [15]. An additional benefit of applying data sphering before data clustering is to simplify the correlation structures of the data so that the distance measures used in the clustering algorithm can also be simplified since different dimensions of the non-sphered data have different scales.

Our RBF density estimation starts with data sphering on the observed training data to get rid of probabilistic outliers and at the same time, if desired, to normalize the spread of data in all directions to facilitate the data clustering. All sphered data with larger norm (e.g., $\|\mathbf{z}\| \geq \beta$, where β is a prespecified threshold) are excluded for clustering. This data sphering and outlier removing process continues for several iterations until no outlying data can be removed.

To verify our assumption of the adverse impact of outlying data on density estimation, a simple 2-D density estimation experiment is conducted here. Fig. 4(e) shows the true density of a long tailed single-mode Cauchy density function:

$$C(\mathbf{y}, \mathbf{m}, \mathbf{u}) = \frac{u_1}{\pi[u_1^2 + (y_1 - m_1)^2]} \cdot \frac{u_2}{\pi[u_2^2 + (y_2 - m_2)^2]} \tag{11}$$

where $\mathbf{y} = [y_1,\ y_2]^T$, $\mathbf{m} = [m_1,\ m_2]^T = [0.0, 0.0]^T$ and $\mathbf{u} = [u_1, u_2]^T = [0.84, 1.02]^T$.

Based on 1600 observed data randomly sampled from this distribution, the corresponding 32 cluster centers (centroids) found by some clustering algorithm (to be discussed later) without outlier removal are wide spread as shown in Fig. 4(a). The RBF approximated kernel density built upon these 32 centroids is shown in Fig. 4(b). Note that this estimated density is nothing near the true density. On the other hand, when outlier removal is applied before data clustering, the 32 centroids shown in Fig. 4(c) found by the LBG algorithm are much more representative to the true data distribution. Therefore, the estimated density is a better approximation to the true density (see Fig. 4(d)).

2) Data Clustering and Centroid Splitting: After the data sphering and outlier removing, a clustering method can be applied to the search of representative centroids so that the reduced number of kernels in the RBF network can be deployed. The generalized Lloyd algorithm with centroid splitting (also known as LBG algorithm [9], [16]), originally developed for codebook generation in vector quantization applications, is used. Compared with the sequential (or batch) K-mean algorithm, the performance of LBG algorithm with centroid splitting is more consistent since it is not affected by the initial guess as the K-means algorithm is. More specifically, the LBG algorithm performs a distortion descent search to find a set of cluster centers which comprise a local minimum in the sense of the least mean squared errors. The basic LBG algorithm can be summarized as follows:

Step 0: Given: a set of training data and an initial codebook.

Step 1: Cluster the training data using the old codebook based on prespecified distance measures (e.g., the Euclidean distance). If the average distortion is small enough, quit.

Step 2: Replace the old codebook with the centroids of clusters obtained in Step 1. Go to Step 1.

The centroid splitting approach [9], [16] is applied to reduce the sensitive dependence of locations and size of the initial codebook to the performance of the clustering. One first finds the optimum codebook of size one, i.e., the centroid of the entire training data set. This single codeword is then split to form the initial codebook of size two and the LBG algorithm is run to reach the local minimum. The procedure is then repetitively applied to enlarge the codebook size.

3) Construction of an RBF Density: Due to the employment of the data sphering, the covariance matrix for each data cluster of the sphered data $\mathbf{z}$ is expected to be close to a diagonal matrix, i.e., the data variance in each dimension can be independently computed. Therefore the RBF density

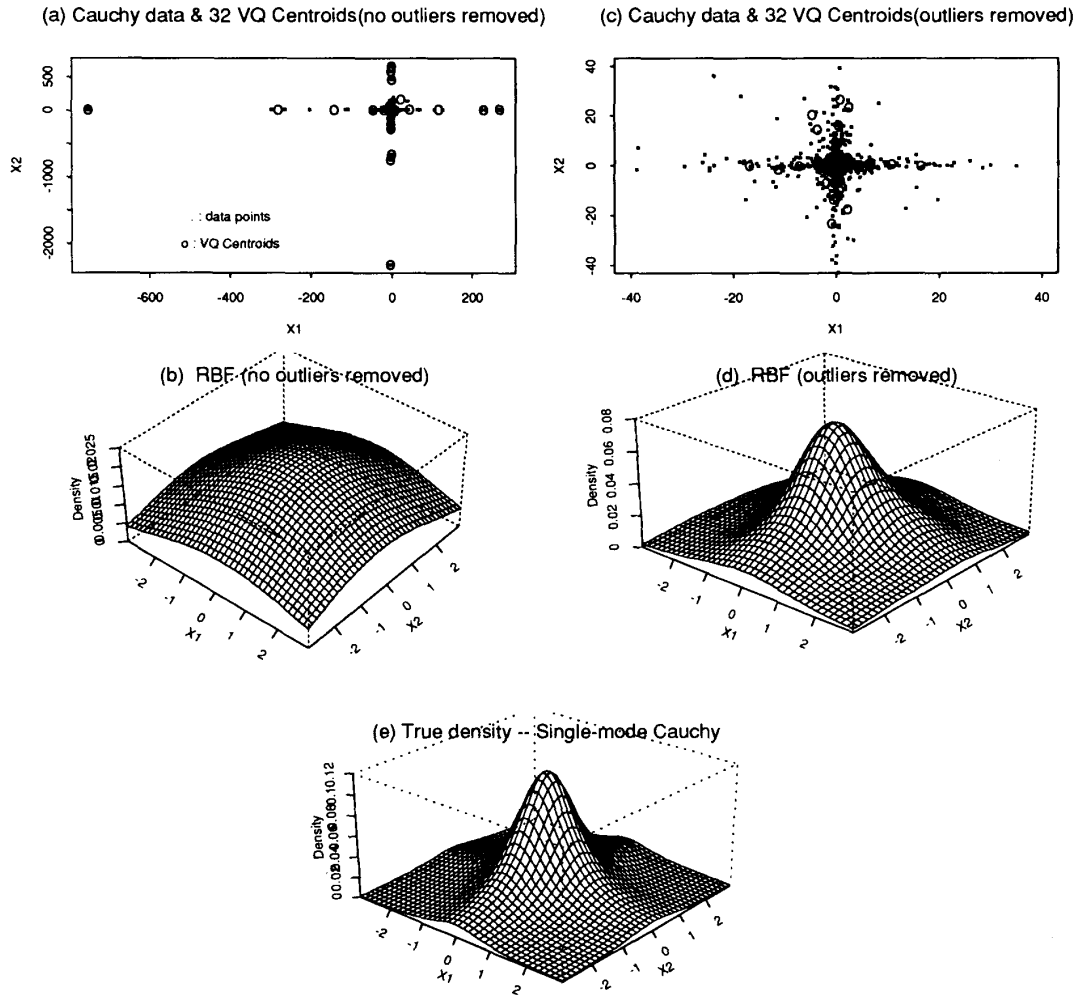

Fig. 4. (a) The 32 centroids found by the LBG algorithm without outlier removal. (b) The estimated density based on the 32 centroids in (a). (c) The 32 centroids found by the LBG algorithm afteroutlier removal. (d) The estimated density based on the 32 centroids in (c). (e) The true density of a long-tailed Cauchy density function. Note that the data coordinates $[u_1, u_2]$ are labeled as $[x_1, x_2]$ in the plots.

estimator of q kernels can have a simplified overall response function

$$\hat{f}(\mathbf{z}) = \sum_{i=1}^{q} c_i \phi_i(\mathbf{z}, \mathbf{m}_i, \mathbf{v}_i) \tag{12}$$

$$\phi_i(\mathbf{z}, \mathbf{m}_i, \mathbf{v}_i) = \frac{1}{(\sqrt{2\pi})^p \prod_{j=1}^{p} v_{ij}} e^{-\frac{1}{2}\sum_{j=1}^{p} \frac{(z_j - m_{ij})^2}{v_{ij}^2}} \tag{13}$$

where $\mathbf{m}_i = (m_{i1}, m_{i2}, \cdots, m_{ip})^T$ denotes the centroid vector of the ith Gaussian kernel obtained from the LBG clustering. $\mathbf{c} = (c_1, c_2, \cdots, c_q)^T$ is the kernel height vector, and $\mathbf{v}_i = (v_{i1}, v_{i2}, \cdots, v_{ip})^T$ is a width vector for ith kernel.

In our implementation of the RBF density estimator, the heights $\{c_i\}$ of kernels are determined by the percentages of training data clustered to various centroids; the kernel widths $\{v_{ij}\}$ are designed to be proportional to (with a factor η, empirically in our simulations $1 \leq \eta^2 \leq 2.0$) the standard (sample) deviation in each dimension for each cluster. In the case of very few data points clustered to a centroid, the average standard deviation among all dimensions is used to regularize the estimation so that a very steep kernel can be avoided. One can also deploy asymmetric kernels in each clustered region if the number of data points are large enough to compute the full covariance matrix.

III. Projection Pursuit Density Estimation

The spirit of projection pursuit density estimation (PPDE) is based on looking for "interesting" low dimensional data projections which reveal distribution structures. Although the notion of "interestingness" may be difficult to quantify, Huber [12] gave a heuristic suggestion that the Gaussian (normal) distribution ought to be considered to be the least interesting. Building upon this suggestion, Friedman [7] proposed an algorithmic procedure, called exploratory projection pursuit, for

nonparametric multivariate density estimation. In this PPDE procedure, five steps are involved:

1) *Data Sphering:* Simplify the location, scale, and correlation structures and remove outliers (as discussed in RBF density estimators, see Section II.B).
2) *Projection Index:* Indicate the degree of interestingness of different projection directions.
3) *Optimization Strategy:* Search efficiently the direction of maximal projection index.
4) *Structure Removing:* Perform 1-D density estimation on the projection data and transform the data to remove this structure.
5) *Density Formation:* Combine the 1-D densities from all searched interesting directions to form the multivariate density function.

A. Projection Index: Which Projection Direction is Interesting?

It is known that all projections of a multivariate Gaussian density are Gaussian, and therefore evidence for the data being non-Gaussian in any projection is evidence against the data being multivariate joint Gaussian. One intuitive definition of projection index $\tilde{I}(\alpha)$, which indicates how close the probability $f_\alpha(x)$ of the 1-D projection data, $x = \alpha^T \mathbf{z}$ along a direction α, being Gaussian (where $\mathbf{z}$ is the sphered version of $\mathbf{y}$), is [10]

$$\tilde{I}(\alpha) = \int_{-\infty}^{\infty} (f_\alpha(x) - g(x))^2 dx, \text{ with } g(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}. \tag{14}$$

A projection direction α that maximizes $\tilde{I}(\alpha)$ yields a projected distribution that exhibit clustering (multimodality) or other kinds of nonlinear structure. If we transform the data x by the following equation

$$r = 2G(x) - 1 = 2G(\alpha^T \mathbf{z}) - 1, \qquad r \in [-1, 1] \tag{15}$$

where $G(x)$ is the standard normal cumulative distribution function (CDF)

$$G(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-t^2}{2}} dt. \tag{16}$$

According to the fundamental theorem of random variable transform

$$f_r(r) = \frac{f_\alpha(x)}{|\frac{\partial r}{\partial x}|} = \frac{f_\alpha(x)}{2g(x)} \tag{17}$$

therefore we can rewrite (14) in terms of r as

$$\begin{aligned} \tilde{I}(\alpha) &= \int_{-1}^{1} 2g(x)(f_r(r) - 1/2)^2 dr \\ &= \int_{-1}^{1} 2g\left(G^{-1}\left(\frac{r+1}{2}\right)\right)(f_r(r) - 1/2)^2 dr. \end{aligned} \tag{18}$$

Friedman [7] adopted a slightly different form for the projection index $I(\alpha)$

$$\begin{aligned} I(\alpha) &= \int_{-1}^{1} (f_r(r) - 1/2)^2 dr \\ &= \int_{-1}^{1} f_r^2(r) dr - 1/2. \end{aligned} \tag{19}$$

Note that if x is Gaussian distributed, then $f_r(r) = \frac{1}{2}$, $\forall r$ and projection index $I(\alpha)$ is zero. The more departure of the distribution of x from normality, the larger the value of index $I(\alpha)$. Since $r \in [-1, 1]$, $f_r(r)$ can be expanded in terms of orthogonal Legendre polynomials $\{\psi_j(r),\ j = 0, \cdots, J\}$, i.e., $f_r(r) = \sum_{j=0}^{J} b_j \psi_j(r)$

$$\begin{aligned} I(\alpha) &= \int_{-1}^{1} f_r^2(r) dr - 1/2 \\ &= \int_{-1}^{1} \left[\sum_{j=0}^{J} b_j \psi_j(r)\right] f_r(r) dr - 1/2. \end{aligned} \tag{20}$$

The orthogonal Legendre polynomials have recursive relation as follows

$$\begin{aligned} &\psi_0(r) = 1, \qquad \psi_1(r) = r \\ &\psi_j(r) = [(2j-1) r \psi_{j-1}(r) - (j-1)\psi_{j-2}(r)]/j, \\ &\qquad \text{for } j \geq 2. \end{aligned} \tag{21}$$

Through the orthogonal property, the weighting coefficients $\{b_j\}$ can be computed via *sample average*

$$\begin{aligned} b_j &= \frac{2j+1}{2} \int_{-1}^{1} \psi_j(r) f_r(r) dr \\ &= \frac{2j+1}{2} E_r[\psi_j(r)] \\ &= \frac{(2j+1)}{2} \frac{1}{N} \sum_{i=1}^{N} \psi_j(2G(x_i) - 1) \end{aligned} \tag{22}$$

where $\int_{-1}^{1} \psi_j(r) f_r(r) dr = E_r[\psi_j(r)]$ is approximated by sample average. Therefore, (19) can be rewritten as

$$\begin{aligned} I(\alpha) &= \int_{-1}^{1} f_r^2(r) dr - 1/2 \\ &= \sum_{j=1}^{J} \frac{2j+1}{2} E_r^2[\psi_j(r)]. \end{aligned} \tag{23}$$

B. Optimization Strategy: The Search for a Best Projection

Once the analytical form of the projection index is defined, its gradient with respect to a projection direction α can be derived as (under the constraint $\alpha^T \alpha = 1$ [7]

$$\begin{aligned} \frac{\partial I}{\partial \alpha} &= \frac{2}{\sqrt{2\pi}} \sum_{j=1}^{J} (2j+1) E[\psi_j(r)] \\ &\quad \times E[\psi_j'(r) e^{-x^2/2} (\mathbf{z} - \alpha x)] \end{aligned} \tag{24}$$

where the derivative of each Legendre polynomial can be easily calculated by the recursive formula

$$\psi_1'(r) = 1, \text{ and } \psi_j'(r) = r\psi_{j-1}'(r) + j\psi_{j-1}(r), \text{ for } j > 1. \tag{25}$$

A *hybrid optimization* strategy [7] was used to search for the most interesting projection direction. A "coarse stepping optimizer" is first applied to perform a search on main axes (principal component directions) and their combination directions so that an initial estimate for a maximum can be quickly reached. A "gradient directed optimizer" (steepest ascent) is then adopted to fine-tune the projection direction to ascend to a (local) maximum of projection index.

C. Structure Removal: Gaussianize Data Along the Projection

To construct a PPDE, several interesting projections are usually required. After an interesting projection α is found, we have to remove the least Gaussian structure along α to avoid future search of this direction again, in other words, we have to "Gaussianize" the data along α without affecting the density along other directions. Let's denote the 1-D projection data before and after Gaussianization as x and $\tilde{x}$, respectively. Gaussianization of the 1-D projection data x is accomplished by

$$\tilde{x} = G^{-1}(F_\alpha(x)) \tag{26}$$

where G^{-1} is the inverse of the standard normal CDF given in (16) and $F_\alpha(x)$ is an estimate of the CDF of x. Friedman [7] suggested to use the empirical CDF $\hat{F}_\alpha(x) = \text{rank}(x)/N - \frac{1}{2N}$, where $\text{rank}(x)$ is the rank of x among all the N observed data points. However, this empirical distribution formulation is quite inaccurate and usually results in very unsmooth estimated densities. We estimate $F_\alpha(x)$ through intergration of a linear interpolation of $f_\alpha(x)$. Based on this modification, we then have to compute the high dimensional structure removed data $\tilde{\mathbf{z}}$ from $\mathbf{z}$. Let $\mathbf{U}$ be an orthonormal matrix, $\mathbf{U} = [\alpha, \beta_1, \beta_2, \cdots, \beta_{p-1}]^T$, where $\{\beta_i\}$ are found through Gram-Schmidt algorithm

$$\begin{aligned} \mathbf{Uz} &= [\alpha^T\mathbf{z}, \beta_1^T\mathbf{z}, \beta_2^T\mathbf{z}, \cdots, \beta_{p-1}^T\mathbf{z}]^T \\ \Theta(\mathbf{Uz}) &= [\tilde{x}, \beta_1^T\mathbf{z}, \beta_2^T\mathbf{z}, \cdots, \beta_{p-1}^T\mathbf{z}]^T \\ \tilde{\mathbf{z}} &= \mathbf{U}^T\Theta(\mathbf{Uz}). \end{aligned} \tag{27}$$

The same projection index maximization procedure is reapplied to the data $\tilde{\mathbf{z}}$ for the searching of other interesting projection structures until the multivariate data is close to Gaussian distribution in any direction. It was noted [7] that "Gaussianizing" along one solution projection perturbs the normality along previously found solution projections so that they no longer have exactly zero interest. However, empirical experience indicates that the induced perturbation is very small. If desired, the *backfitting* procedure [5] can be reapplied to the previous projections.

D. Density Formation: From Projections to Density

The density of the original sphered data is estimated by combining those projected 1-D density estimations. The density relation between the high-dimensional data $\mathbf{z}^{(m)}$ and $\mathbf{z}^{(m-1)}$ is (where $\mathbf{z}^{(m)}$ is the structure removed data of $\mathbf{z}^{(m-1)}$ along the mth projection α_m)

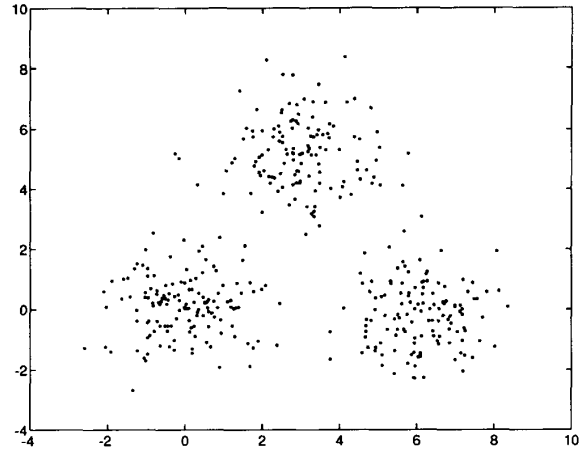

Fig. 5. Four hundred randomly sampled Gaussian mixture data.

$$\begin{aligned} f_{\alpha_m}\left(\mathbf{z}^{(m)}\right) &= \frac{f_{\alpha_{m-1}}\left(\mathbf{z}^{(m-1)}\right)}{\left|J_m\left(\mathbf{z}^{(m-1)}\right)\right|} \\ f_{\alpha_{m-1}}\left(\mathbf{z}^{(m-1)}\right) &= f_{\alpha_m}\left(\mathbf{z}^{(m)}\right)\left|J_m\left(\mathbf{z}^{(m-1)}\right)\right| \end{aligned} \tag{28}$$

where the Jacobian

$$\begin{aligned} J_m\left(\mathbf{z}^{(m-1)}\right) &= \frac{\partial \mathbf{z}^{(m)}}{\partial \mathbf{z}^{(m-1)}} = \frac{\partial\left(\mathbf{U}\mathbf{z}^{(m)}\right)}{\partial\left(\mathbf{U}\mathbf{z}^{(m-1)}\right)} \\ &= \frac{\partial x^{(m)}}{\partial x^{(m-1)}} = \frac{f_{\alpha_m}\left(x^{(m-1)}\right)}{g\left(x^{(m)}\right)} \\ &= \frac{f_{\alpha_m}\left(\alpha_m^T\mathbf{z}^{(m-1)}\right)}{g\left(\alpha_m^T\mathbf{z}^{(m)}\right)} \geq 0. \end{aligned} \tag{29}$$

Starting from the original multivariate data $\mathbf{z}^{(0)}$, the Gaussianization procedure is applied to every interesting projection found by the optimization procedure. At some point, say after M projections, the multivariate data $\mathbf{z}^{(M)}$ no more exhibits much deviation from normality, i.e., $f_{\alpha_M}(\mathbf{z}^{(M)}) \approx g(\mathbf{z}^{(M)})$, where $g(\mathbf{z}) = \frac{1}{(2\pi)^{p/2}}\exp(-\mathbf{z}^T\mathbf{z}/2)$ is a standard multivariate Gaussian distribution. The density of $\mathbf{z}^{(0)}$ can now be estimated to be

$$\begin{aligned} f(\mathbf{z}^{(0)}) &= f_{\alpha_1}(\mathbf{z}^{(1)})\, J_1(\mathbf{z}^{(0)}) \\ &= f_{\alpha_2}(\mathbf{z}^{(2)})\, J_2(\mathbf{z}^{(1)})\, J_1(\mathbf{z}^{(0)}) \\ &= f_{\alpha_M}(\mathbf{z}^{(M)}) \prod_{m=1}^{M} J_m(\mathbf{z}^{(m-1)}) \\ &\approx g(\mathbf{z}^{(M)}) \prod_{m=1}^{M} J_m(\mathbf{z}^{(m-1)}) \\ &\approx g(\mathbf{z}^{(M)}) \prod_{m=1}^{M} \frac{f_{\alpha_m}(\alpha_m^T\mathbf{z}^{(m-1)})}{g(\alpha_m^T\mathbf{z}^{(m)})}. \end{aligned} \tag{30}$$

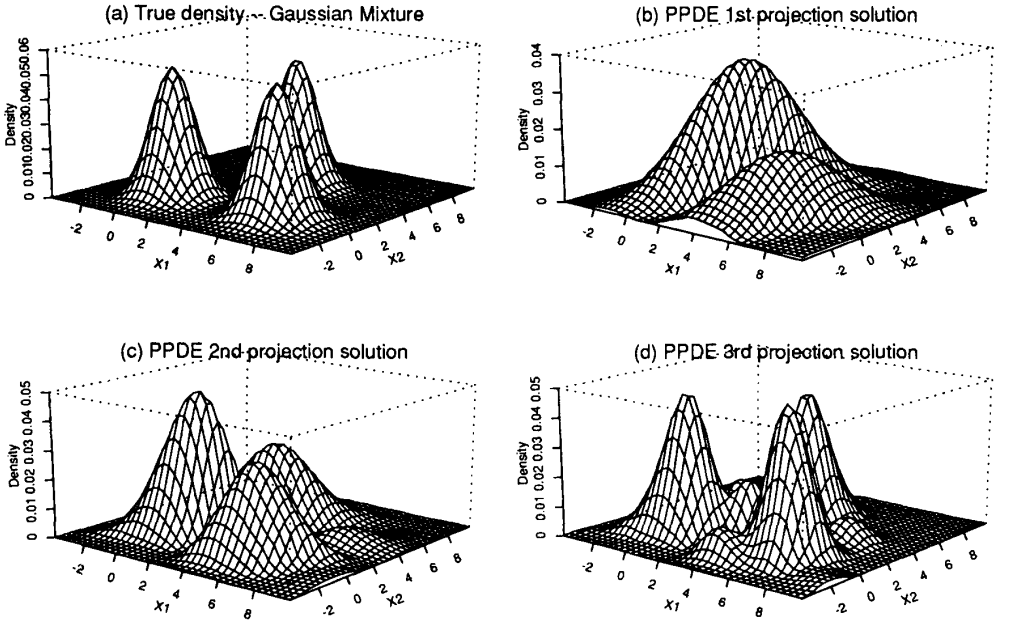

Fig. 6. (a) The true density of a Gaussian mixture. (b) The PPD eestimate, $f_{\alpha_1}(\mathbf{z}^{(1)})J_1(\mathbf{z}^{(0)})$, after the first projection. (c) The PPDE estimate, $f_{\alpha_2}(\mathbf{z}^{(2)})J_2(\mathbf{z}^{(1)})\ J_1(\mathbf{z}^{(0)})$, after the second projection. (d) The PPDE estimate, $f_{\alpha_3}(\mathbf{z}^{(3)})J_3(\mathbf{z}^{(2)})\ J_2(\mathbf{z}^{(1)})J_1(\mathbf{z}^{(0)})$, after the third projection. Note that the data coordinates $[z_1, z_2]$ are labeled as $[x_1, x_2]$ in the plots.

The 1-D probability $f_{\alpha_m}(\alpha_m^T\mathbf{z}^{(m-1)})$ is estimated according to (17), i.e., $f_\alpha = 2g(x)f_r(r)$, or more specifically

$$f_{\alpha_m}\left(\alpha_m^T\mathbf{z}^{(m-1)}\right) = g\left(\alpha_m^T\mathbf{z}^{(m-1)}\right)\sum_{j=0}^{J}(2j+1) \times E_{m-1}\left[\psi_j(r^{(m-1)})\right]\psi_j\left(r^{(m-1)}\right). \tag{31}$$

Due to the polynomial form of the projection index and the recursive relations in the polynomials and its first derivatives, PPDE can be rapidly computed. Figs. 5 and 6 gives a step-by-step illustration of the PPDE construction from the first three projections using 400 training data sampled from a Gaussian mixture.

IV. Comparative Simulations

We have discussed the nonparametric "kernel-based" and "projection-pursuit" density estimators from structural and computational viewpoints. We carry out in this section a detailed comparison of performance among these methods via a simulation study.

A. Simulated Data

Three types of multidimensional (2-D–5-D) data of Gaussian and Cauchy mixture distributions are generated. The Cauchy distribution has a long tail while the Gaussian distribution does not. The data are generated such that all elements in the same data vector are independent of each other. These data have the following distribution forms

$$\text{Gaussian Mixture: } \sum_{k=1}^{K} c_k N(\mathbf{y}, \mathbf{m}_k, \mathbf{v}_k)$$
$$\text{Cauchy Mixture: } \sum_{k=1}^{K} c_k C(\mathbf{y}, \mathbf{m}_k, \mathbf{u}_k) \tag{32}$$

with the constraint $\sum_{k=1}^{M} c_k = 1$ and

$$N(\mathbf{y}, \mathbf{m}_k, \mathbf{v}_k) = \frac{1}{\prod_{j=1}^{p}\sqrt{2\pi v_{kj}}}\, e^{-\frac{1}{2}\sum_{j=1}^{p}\frac{1}{v_{kj}}(y_j - m_{kj})^2}$$
$$C(\mathbf{y}, \mathbf{m}_k, \mathbf{u}_k) = \prod_{j=1}^{p}\frac{u_{kj}}{\pi\left[u_{kj}^2 + (y_j - m_{kj})^2\right]}. \tag{33}$$

1) Single Mode Distribution: The first type of data is a single-mode distribution with parameters chosen as follows (note that, for 2-D–4-D cases we take the first 2-D–4-D elements from the 5-D parameter set as shown below):

Gaussian Distribution:

$$c = 1.0, \quad \mathbf{m} = [0.0, 0.0, 0.0, 0.0, 0.0]^T,$$
$$\mathbf{v} = [0.84, 1.02, 0.70, 1.20, 0.96]^T.$$

Cauchy Distribution:

$$c = 1.0, \quad \mathbf{m} = [0.0,\ 0.0,\ 0.0,\ 0.0,\ 0.0]^T,$$
$$\mathbf{u} = [0.84, 1.02, 0.70, 1.20, 0.96]^T.$$

2) Lightly Overlapped Two-Mode Distribution: The second type of data is a lightly overlapped two-mode distribution with parameters chosen as follows:

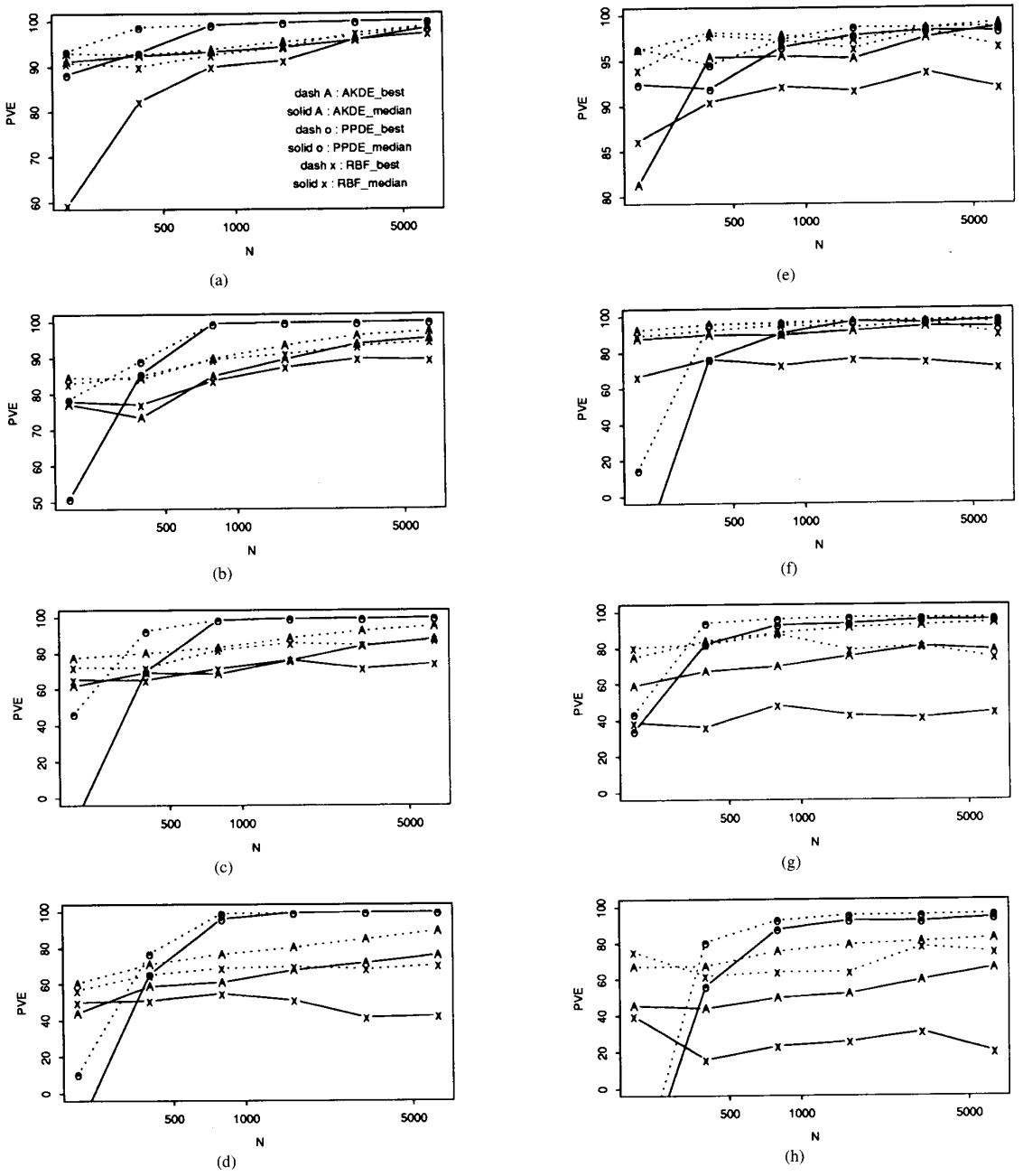

Fig. 7. Estimation accuracy based on PVE measures for single-mode data of Gaussian (a) 2-D, (b) 3-D, (c) 4-D, and (d) 5-D; Cauchy (e) 2-D, (f) 3-D, (g) 4-D, and (h) 5-D.

Gaussian Distribution:

$$c_1 = 0.65, \quad \mathbf{m}_1 = [0.0, 0.0, 0.0, 0.0, 0.0]^T,$$
$$\mathbf{v}_1 = [0.42, 0.51, 0.35, 0.60, 0.48]^T.$$
$$c_2 = 0.35, \quad \mathbf{m}_2 = [2.0, 2.0, 2.0, 2.0, 2.0]^T,$$
$$\mathbf{v}_2 = [0.33, 0.46, 0.53, 0.43, 0.45]^T.$$

Cauchy Mixture:

$$c_1 = 0.65, \quad \mathbf{m}_1 = [0.0, 0.0, 0.0, 0.0, 0.0]^T,$$
$$\mathbf{u}_1 = [0.42, 0.51, 0.35, 0.60, 0.48]^T.$$
$$c_2 = 0.35, \quad \mathbf{m}_2 = [2.0, 2.0, 2.0, 2.0, 2.0]^T,$$
$$\mathbf{u}_2 = [0.33, 0.46, 0.53, 0.43, 0.45]^T.$$

3) Heavily Overlapped Two-Mode Distribution: The third type of data is a heavily overlapped two-mode distribution with parameters chosen as follows:

Gaussian Mixture:

$$c_1 = 0.65, \quad \mathbf{m}_1 = [0.0, 0.0, 0.0, 0.0, 0.0]^T,$$
$$\mathbf{v}_1 = [0.84, 1.02, 0.70, 1.20, 0.96]^T.$$

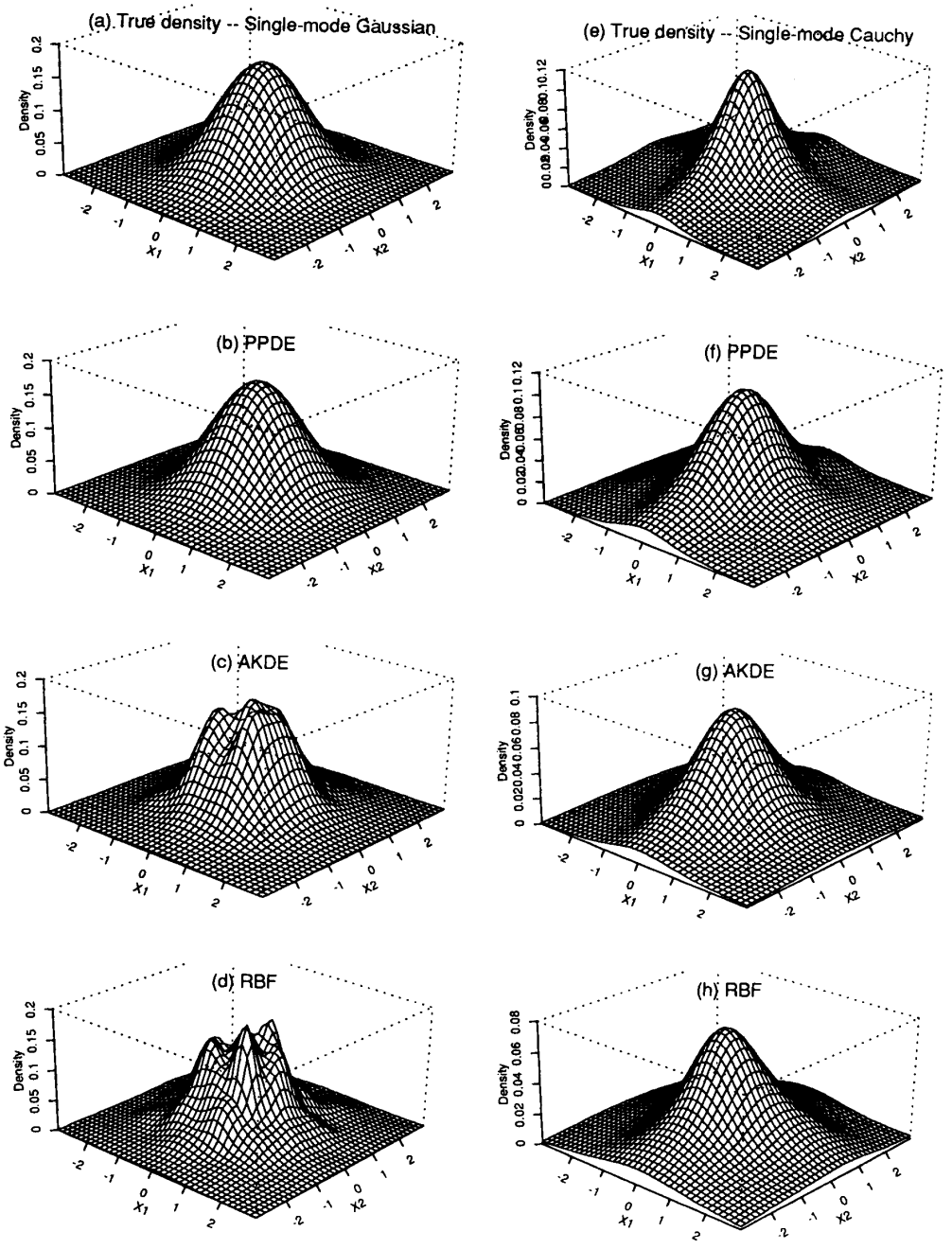

Fig. 8. Perspective plots of single-mode distributions of 2-D Gaussian: (a) True density. (b) PPDE estimation. (c) AKDE estimation. (d) RBF estimation; 2-D Cauchy (e) True density. (f) PPDE estimation. (g) AKDE estimation. (h) RBF estimation.

$$c_2 = 0.35, \quad \mathbf{m}_2 = [2.0, 2.0, 2.0, 2.0, 2.0]^T,$$
$$\mathbf{v}_2 = [0.66, 0.92, 1.06, 0.86, 0.90]^T.$$

Cauchy Mixture:

$$c_1 = 0.65, \quad \mathbf{m}_1 = [0.0, 0.0, 0.0, 0.0, 0.0]^T,$$
$$\mathbf{u}_1 = [0.84, 1.02, 0.70, 1.20, 0.96]^T.$$
$$c_2 = 0.35, \quad \mathbf{m}_2 = [2.0, 2.0, 2.0, 2.0, 2.0]^T,$$
$$\mathbf{u}_2 = [0.66, 0.92, 1.06, 0.86, 0.90]^T.$$

For each type of data (either mixture Gaussian or mixture Cauchy) of any dimension (2-D to 5-D), six randomly sampled data sets of different sizes (200, 400, 800, 1600, 3200, 6400) are created for training and an additional randomly sampled data set of size 20 000 is created for testing.

B. Performance Evaluation

To objectively compare the performance, *Monte Carlo* approximation of *percentage of variance explained* (PVE) [5]

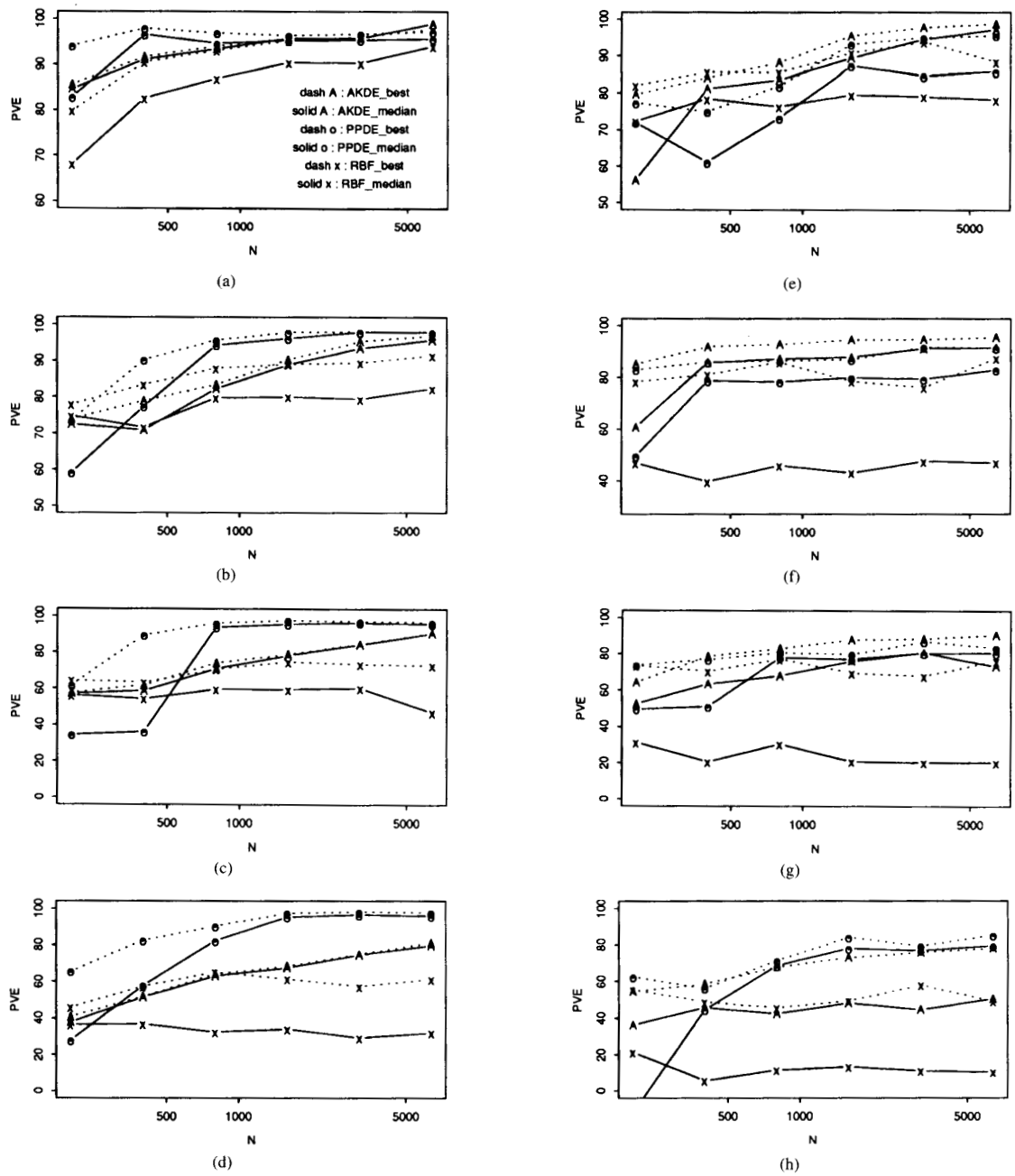

Fig. 9. Estimation accuracy based on PVE measures for two-mode lightly overlapped data of Gaussian (a) 2-D, (b) 3-D, (c) 4-D, and (d) 5-D; Cauchy (e) 2-D, (f) 3-D, (g) 4-D, and (h) 5-D.

measures is used

$$\text{PVE} = 100(1 - \text{Err}/\text{Var})\%$$

where $\text{Err} = \frac{1}{20\,000}\sum_{n=1}^{20\,000}(\hat{f}_n - f_n)^2$ denotes the mean squared error between the estimated density $\hat{f}_n$ and the true density f_n over 20 000 testing data, and $\text{Var} = \frac{1}{20{,}000}\sum_{n=1}^{20\,000}(f_n - \bar{f})^2$ denotes the sample variance over 20 000 testing data, where $\bar{f}$ denotes the sample average of the true density.

C. Experimental Setup

The experiments for the comparative simulations are done for the three estimators (AKDE, RBF, and PPDE) discussed in this paper. Since the Gaussian distribution is not a long tail distribution, an outlier removing procedure is not necessary

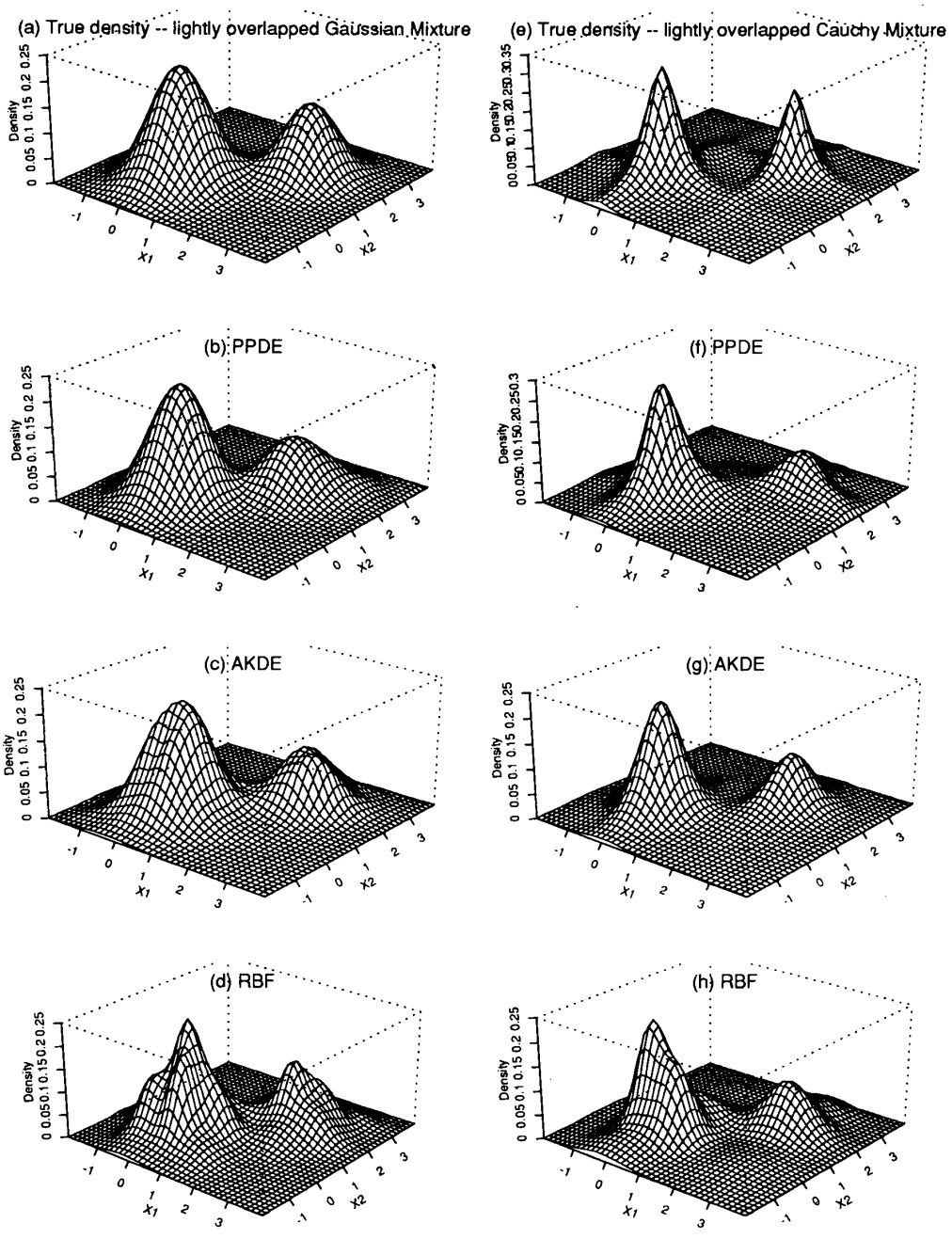

Fig. 10. Perspective plots of two-mode lightly overlapped distributions of 2-D Gaussian. (a) True density. (b) PPDE estimation. (c) AKDE estimation. (d) RBF estimation; 2-D Cauchy (e) True density. (f) PPDE estimation. (g) AKDE estimation. (h) RBF estimation.

and therefore is not applied to the training data. However, in the Cauchy distribution there exists probabilistic outliers which bias the covariance estimation and mislead the search of kernel locations, therefore two sphering radii $\beta = 5$ and $\beta = 6$ (based on our observation of data that the probability of a Gaussian distribution is almost zero with a radius 5 or 6), were tried for outlier removing.

For AKDE, in addition to the choices of sphering radii, several values were tried for γ (=0.2, 0.4, 0.6, and 0.8). The reported AKDE performance is chosen from the best and the median of (in terms of PVE measure) all parameter combinations. In the simulations of RBF density estimators, several combinations of control parameters were tried. For example, two different numbers of clustered kernels, q =

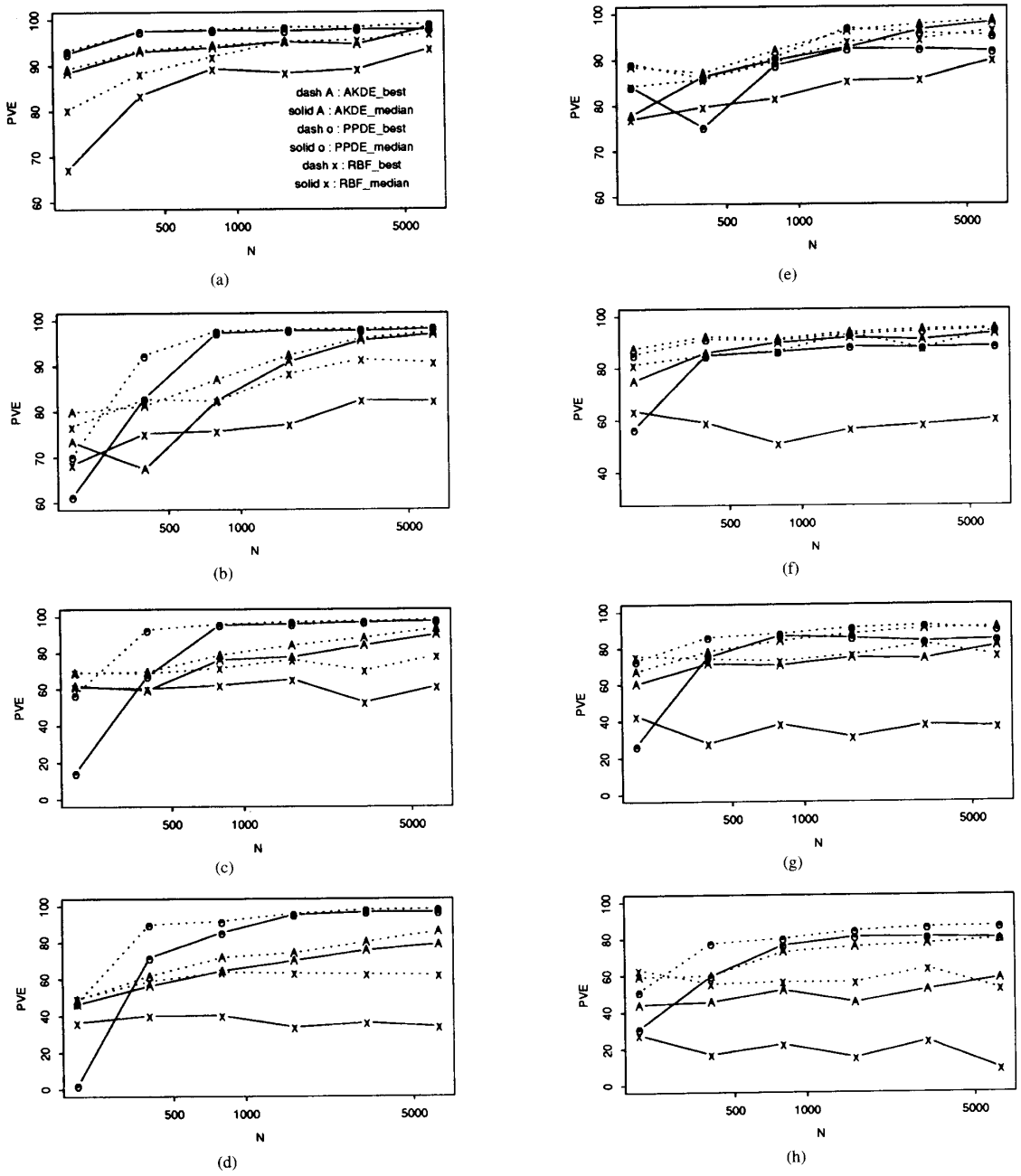

Fig. 11. Estimation accuracy based on PVE measures for two-mode highly overlapped data of Gaussian. (a) 2-D, (b) 3-D, (c) 4-D, and (d) 5-D; Cauchy (e) 2-D, (f) 3-D, (g) 4-D, and (h) 5-D.

16 and $q = 32$, were used. After clustering, the data in each cluster region is assumed to be independent enough in each dimension, therefore the variance of clustered data in each dimension was independently calculated. The kernel smoothing parameter η^2 was chosen to be 1.2, 1.4, 1.6, and 1.8. Among the PVE values corresponding to all different parameter combinations, the median PVE values and the best PVE values of RBF estimation were reported. As for PPDE simulations, three Legendre polynomial orders, 4, 5, and 6, were tried. The number of interesting projections required in constructing the density was not fixed in advance, it was determined dynamically when the new projection index was smaller than either 0.01 or 0.005. Among the PVE values corresponding to all different parameter combinations, the median PVE values and the best PVE values of PPDE estimation were reported.

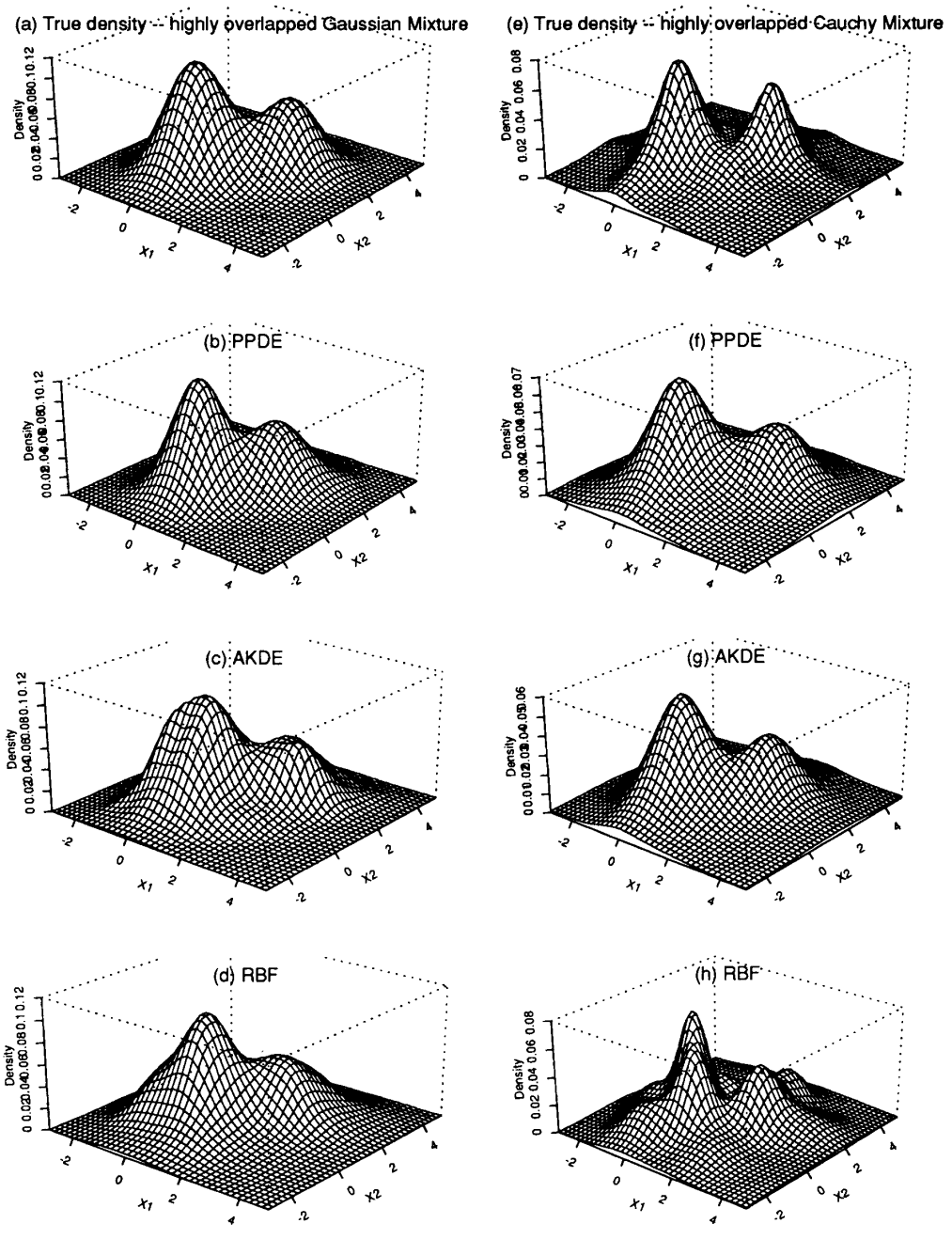

Fig. 12. Perspective plots of two-mode highly overlapped distributions of 2-D Gaussian. (a) True density. (b) PPDE estimation. (c) AKDE estimation. (d) RBF estimation; 2-D Cauchy (e) True density. (f) PPDE estimation. (g) AKDE estimation. (h) RBF estimation.

D. Simulation Results

Fig. 7 shows the (median and best) PVE performance plots for single-mode Gaussian and Cauchy data of various dimensions versus various training data sizes. The perspective plots of the true and estimated densities (based on 1600 data) corresponding to the median PVE for single-mode distribution of 2-D data are shown in Fig. 8. Fig. 9 shows the (median and best) PVE performance plots for two-mode Gaussian and Cauchy lightly overlapped data of various dimensions versus various training data sizes. The perspective plots of the true and estimated densities (based on 1600 data) corresponding to the median PVE for two-mode Gaussian and Cauchy lightly overlapped distribution of 2-D data are shown in Fig. 10. Fig. 11 shows the (median and best) PVE performance plots

for two-mode Gaussian and Cauchy heavily overlapped data of various dimensions versus various training data sizes. The perspective plots of the true and estimated densities (based on 1600 data) corresponding to the median PVE for two-mode Gaussian and Cauchy heavily overlapped data of 2-D data are shown in Fig. 12.

It is observed that the PPDE outperforms the AKDE and RBF, in approximation accuracy based on PVE measures in almost all the simulations. From the PVE plots, one can clearly see that the performances of PPDE median curves do not degrade much from their corresponding PPDE best curves. On the other hand, the performances of RBF median curves degrade a lot from their corresponding RBF best curves. This fact indicates that the PPDE is more robust in that it is less sensitive to the setting of the control parameters values, e.g., the number of (projections) kernels used, the locations of kernels, the orientation of kernels, the kernel smoothing parameters, the excluding threshold radius for data sphering, the size of training data, etc. We can also observe the impact of dimensionality on each method, the PPDE, as expected, suffers much less on the curse of dimensionality when compared to AKDE and RBF methods. More specifically, RBF suffers the curse of dimensionality most in estimating the Cauchy mixtures. Note that PPDE does require at least some minimum number of training data (e.g., 400) to reasonably perform the gaussianization procedure, while the AKDE and RBF can survive at small number of training data (say from 200 to 400) due to their prespecified implicit kernel structures. All three methods exhibit somewhat degraded performance estimation of long-tailed (Cauchy) distribution. However, the performance of AKDE and RBF degrades much more than that of PPDE.

It is also worthwhile to mention the comparative computational complexities of these density estimation methods. Since the construction of projection pursuit density estimator (based on recursive Legendre polynomials) is based on the iterative optimization procedure, a conclusive quantitative comparison of computational complexity of these density estimator methods is very difficult. In general, from our intensive simulations we found that these two methods took quite comparable amount of CPU time (projection pursuit is slightly faster) during the construction of the estimators. While in the testing stage after the estimators are constructed, the robust RBF methods are fastest in responding the density values, the AKDE's are the slowest.

V. Conclusion

We have extensively examined the algorithmic aspects of several nonparametric multivariate density estimators, and have carried out a thorough comparative study via simulations. In our simulation study, the PPDE outperformed the kernel methods in approximation accuracy based on PVE measures in most data sets. In particular, one would expect the RBF kernel method to be a natural fit for estimating the density of Gaussian mixtures, however the PPDE performs better for this set of data. This emphasizes the success of PPDE's.

In spite of its superior performance, the PPDE still suffers from several potential drawbacks which require further research. More specifically, the PPDE can not satisfactorily deal with structures *hidden* behind others, e.g., 2-D data density of doughnut shape. Although this problem can be solved by transforming the original data to other coordinates, such as the polar coordinate, before the application of the PPDE, appropriate use of coordinate transforms and identification of hidden structures in densities remains challenging. Another severe problem is the numerical instability caused by the denominator (Jacobian) term in a long density tail, which should be solved by sophisticated data analysis techniques.

Acknowledgment

The authors wish to thank the anonymous reviewers of this paper for their valuable and constructive suggestions, from which the revision of this paper has benefited significantly.

References

[1] I. S. Abramson, "On bandwidth variation in kernel estimates—A square root law," *Annals Statist.*, vol. 10, pp. 1217–1223, 1982.
[2] P. C. Cosman, K. L. Oehler, E. A. Riskin, and R. M Gray, "Using vector quantization for image processing," *Proc. IEEE*, vol. 81, no. 9, pp. 1326–1341, Sept. 1993.
[3] D. L. Donoho and I. M. Johnstone, "Regression approximation using projection and isotropic kernels," *Contemporary Mathematics*, vol. 59, pp. 153–167, 1986.
[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
[5] J. H. Friedman, W. Stuetzle, and A. Schroeder, "Projection pursuit density estimation," *J. Am. Statistical Assoc.*, vol. 79, pp. 599–608, 1984.
[6] J. H. Friedman and J. W. Tukey, "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. C-23, pp. 881–890, 1974.
[7] J. H. Friedman, "Exploratory projection pursuit," *J. Am. Statist. Assoc.*, vol. 82, pp. 249–266, 1987.
[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1972.
[9] R. M. Gray, "Vector quantization," *IEEE Acoust., Speech, Signal Processing Mag.*, fol. 1, pp. 4–29, Apr. 1984.
[10] P. Hall, "On polynomial-based projection indices for exploratory projection pursuit," *Annals Statist.*, vol. 17, no. 2., pp. 589–605, 1989.
[11] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
[12] P. J. Huber, "Projection pursuit," *Annals Statist.*, vol. 13, no. 2, pp. 435–475, 1985.
[13] C. Hurley and A. Buja, "Analyzing high-dimensional data with motion graphics," *SIAM J. Scientific, Statist. Computing*, vol. 11, no. 6, pp. 1193–1211, 1990.
[14] J. N. Hwang, S. R. Lay, and A. Lippman, "Unsupervised learning for multivariate probability density estimation: Radial basis and projection pursuit," *IEEE Int. Conf. Neural Networks* (San Francisco, CA), Mar. 1993, pp. 1486-1491.
[15] S. R. Lay and J. N. Hwang, "Robust construction of radial basis function neural networks for classification," *IEEE Int. Conf. Neural Networks* (San Francisco, CA), Mar.1993, pp. 1859–1864.
[16] Y. Linde, A. Buzo, and R. M. Gray "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, Jan. 1980.
[17] G. J. Mclachlan and K. E. Basford, *Mixture Models—Inference and Applications to Clustering*. New York: Marcel Dekker, 1988.
[18] J. Moody and C. J. Darken, "Fast learning in networks of locally tuned processing units," *Neural Computation*, vol. 1, no. 3, pp. 281–294, 1989.
[19] K. L. Oehler and R. M. Gray, "Combining image classification and image compression using vector quantization," in *IEEE Data Compression Conf. Proc.*, 1993, pp. 2–11.
[20] K. Popat and R. W. Picard, "Novel cluster-based probability model for texture synthesis, classification, and compression," in *Proc. SPIE Visual Commun. Image Processing'93* (Boston, MA), Nov. 8–11, 1993.
[21] K. Popat and R. W. Picard, "Cluster-based probability model applied to image restoration and compression," to appear in *Proc. ICASSP* (Adelaide, Australia), Apr. 1994.

[22] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc.*, vol. 78, no. 9, Sept. 1990, pp. 1481–1497.
[23] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, Feb. 1989, pp. 257–286.
[24] D. W. Scott, "Multivariate density estimation: Theory, practice, and visualization," *Wiley Series in Probability and Mathematical Statistics*. New York: Wiley, 1992.
[25] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. New York: Chapman and Hall, 1986.
[26] D. F. Specht, "Probabilistic neural networks," *Neural Networks*, vol. 3, pp. 109–118, 1990.
[27] D. M. Titerington, A. F. M. Smith, and U. E. Makov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.
[28] Q. Xie, C. A. Laszlo, and R. K. Ward, "Vector quantization technique for nonparametric classifier design," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 15, no. 12, pp. 1326–1330, Dec. 1993.

Jenq-Neng Hwang (S'82–M'88) received the B.S. and M.S. degrees from the National Taiwan University, Taipei, Taiwan, in 1981 and 1983, respectively, both in electrical engineering. He received the Ph.D. degree from the University of Southern California, Los Angeles, in December 1988.

After two years of obligatory military service, he enrolled as a Research Assistant in 1985 at the Signal and Image Processing Institute, Department of Electrical Engineering, University of Southern California. He was a visiting student at Princeton University, Princeton, NJ, from 1987 to 1989. Since summer 1989, he has been with the Department of Electrical Engineering, University of Washington, Seattle, as an Assistant Professor. His research interests include signal/image processing, statistical data analysis, computational neural networks, parallel algorithm design, and VLSI array architecture.

Dr. Hwang served as the Secretary of the Neural Systems and Applications Committee of the IEEE Circuits and Systems Society from 1989 to 1991 and is a member of Technical Committees in the IEEE Signal Processing Society: VLSI Signal Processing and Neural Networks Signal Processing. Currently, he is also serving as an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING and IEEE TRANSACTIONS ON NEURAL NETWORKS. He is the Conference Program Chair of the 1994 IEEE Workshop on Neural Networks for Signal Processing, to be held in Ermioni, Greece He is also the Program Co-Chair of the International Symposium on Artificial Neural Networks to be held in Tainan, Taiwan, R.O.C. in December 1994.

Shyh-Rong Lay was born in I-Lan, Taiwan, on May 21, 1961. He received the B.S. degree in electronics engineering from National Chiao Tung University, Taiwan, in 1983, and the M.S. degree from Pennsylvania State University, State College, in 1990 and the Ph.D. degree from the University of Washington, Seattle, in 1994, both in electrical engineering.

He served at the Chung-Shan Institute of Science and Technology, Taiwan, from 1983 to 1988. His research interests include digital signal processing, digital communications, computational neural networks, pattern recognition, data compression, and statistical data analysis.

Alan Lippman received the B.S. degree in mathematics from the University of Washington, Seattle, and the M.S. and Ph.D. degrees from Brown University, Providence, RI, both in applied mathematics.

From 1990 to 1993, he was a Research Associate in the School of Oceanography at the University of Washington. He presently is a Senior Software Engineer and the Ultrasound group of Siemens Medical Systems, Inc. His current interest is the design and implementation of algorithms for real-time systems.

He was awarded a National Science Foundation mathematical sciences postdoctoral fellowship in 1987.