Probability Density Estimation Using Elliptical Basis Functions

Lloyd P. M. Johnston and Mark A. Kramer
Dept. of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139

Elliptical basis function (EBF) networks are introduced as a new nonparametric method of estimating probability density functions for process data. Unlike Parzen window density estimators that use identical hyperspherical basis functions, the EBF method uses elliptical basis functions adapted to the local character of the data. This technique overcomes the spikiness problem associated with Parzen windows, where in high dimension, they can fail to produce smooth probability density estimates. The EBF estimator produces valid density functions that converge to the underlying distribution of the data in the limit of an infinite number of training examples. A technique based on statistical cross validation is introduced for evaluating different density estimators. The criterion is a measure of how well the density estimator estimates the density of data not used in the training. The EBF density estimation method and the evaluation technique are demonstrated using several examples of fault diagnosis.

Introduction

Statistical characterization of a process is required for many process monitoring and control tasks. Although many statistics can be calculated, one of the most useful statistics is the joint probability density function (PDF) of the process variables. Once the PDF is obtained, it is possible to perform a variety of monitoring and control tasks. The PDF can be used to describe the normal operating regime of a process or the parametric region capable of producing product within acceptable quality specifications, and to evaluate the probability that current process conditions will produce product within specifications. If this probability is judged to be too low, then adjustments can be made to the process until the conditions are such that "good" product is highly likely. In fault diagnosis, PDFs can be used to estimate the relative probability of classes of events, given the current information about the process. Using separate PDF estimators for each class, Bayes theorem can determine the class with the highest probability given the current conditions (Duda and Hart, 1973). In sensor validation and gross error detection, a low joint PDF of the current measurements indicates that there is a gross error in one or more of the sensors (assuming that the plant is in a normal operating state). There is also the potential of applying the PDF to data rectification (Kramer and Mah, 1993).

Because of the wide applicability of PDFs in process monitoring and control tasks, accurate estimation of PDFs from process data is an important task. There are currently several techniques, both parametric and nonparametric, for estimating PDFs from data. The parametric techniques assume a model of the density function and then regress any required parameters. The major shortcoming of parametric approaches is that often the model form is unknown. A common assumption is that the data approximates normal behavior and a multivariate Gaussian (MVG) model is used to approximate the density function. This method is not adequate for approximating density functions that are multimodal, or density functions in which the process variables are related by nonlinear correlations. When there is prior knowledge that the underlying distribution is of a certain type (for example, Gaussian) then the appropriate parametric technique can be used, but for most processes the underlying distribution of the data is not known and most likely does not follow a particular class of density function. In these cases nonparametric techniques are the methods of choice. Nonparametric techniques avoid prior assumptions of the form of the PDF, and thus maintain a broader applicability than parametric methods.

Correspondence concerning this article should be addressed to M. A. Kramer at Gensym Corp., 125 Cambridgepark Dr., Cambridge, MA 02140.

One of the most common nonparametric techniques for density estimation is Parzen windows (PW) (Parzen, 1962). In this estimator, an identical basis (window) function is centered at every point in the data set. The density estimate is found by summing up the density estimates from the individual basis density functions. There are three problems with this approach. First, there are as many basis functions as data points, leading to a high computational load. Second, individual basis functions of the Parzen window estimator are unable to capture the local distribution characteristics of the data because each basis function is identical, leading to the requirement for many basis functions to converge to the true local density. It may be much more efficient to use fewer basis functions that are adapted to the local character of the data. Third, the use of a large number of identical basis functions in Parzen windows can lead to spiky (as opposed to smooth) density estimates (Duda and Hart, 1973). The primary reasons for this are because the identical window size is inappropriate when the data is more dense in some areas than others, and because the hyperspherical shape of the units are inappropriate when the data is correlated. With a finite data set, the distribution of the data may require large window sizes to completely cover the data in one area of the space while in another area of the same space, the large window size may have the effect of smoothing over correlations in the underlying density function. On the other hand, using small window sizes to capture the detailed shape of the underlying distribution in one area of the space may have the effect of producing a spiky estimate in another area of the space where the training data is more sparse (see Figure 1). The proposed elliptical basis function (EBF) estimator is also nonparametric, however, because it is capable of capturing the local characteristics of the data, it is less prone to produce spiky or poorly fit density functions. The EBF estimator overcomes these problems by allowing for differently sized and shaped basis functions. In areas where the density function is rapidly changing, many small, differently shaped basis functions are placed to capture these variations, and in areas of relative smoothness fewer larger units are placed to reflect the characteristics of that region of the data space (see Figure 1). The proposed EBF estimator is shown later in this article to be a valid data density function and to converge to the underlying data density distribution in the limit of infinite data.

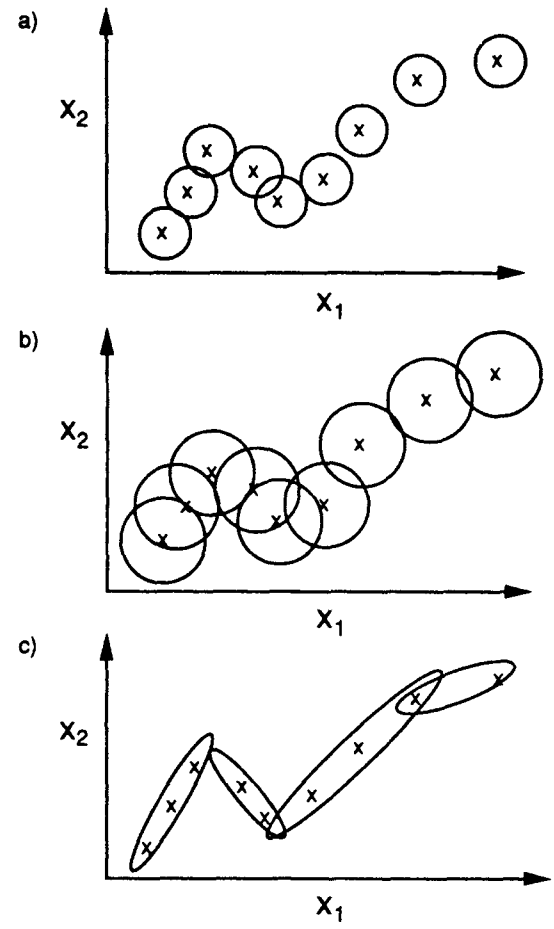

Figure 1. Conceptual view of Parzen windows vs. EBF fit to data.

(a) Small Parzen window size captures variation in lower left corner, but leaves gaps in top right corner; the density estimate in top right portion of graph would be "spiky." (b) Larger Parzen window size eliminates spikiness problem in top right portion but would give an overly smooth estimate in lower left corner missing the correlations among the data. (c) EBF covering the data set in a smooth manner, all correlation among variables is captured and there are no discontinuities.

Kavuri and Venkatasubramanian (1993), Renals and Rohwer (1989), Holcomb and Morari (1991), and Poggio and Girosi (1990) have reported on the use of networks of elliptical basis functions for functional approximation and classification, but not for estimation of probability density functions. Kavuri and Venkatasubramanian solved classification problems by using elliptical units that are oriented with their axes parallel to the axes of the input space (that is, units had diagonal covariance matrices). Renals and Rohwer used elliptical units with diagonal covariance matrices to solve static speech classification problems. The simplification of a diagonal covariance matrix reduces the computational load but is inefficient, in the sense that it takes many units to capture class regions with large correlations between variables. Poggio and Girosi used regularization techniques to arrive at generalized radial basis function (GRBF) networks that use elliptical units with nondiagonal covariance matrices, but have as many units as data points. Based on the high computational load that would result from the large number of units used, Poggio and Girosi advocated a heuristic approach involving fewer basis function units than data points. The GRBF method was directed at solving functional approximation and classification problems. Holcomb and Morari also allowed for arbitrary orientation of their elliptical units but used a penalty function method that requires some tuning. They also failed to define a technique for selecting the number of units to use and the penalty function to ensure elliptical shapes that conform well to the data.

This article not only introduces a new technique for density estimation but it also proposes a methodology for selecting the best density estimator for a given application. The well-known technique of cross-validation (Weiss and Kulikowski, 1991) is used in conjunction with the "log probability" utility function, to rank the predictive power of each estimator. This methodology can be used to compare estimators of the same basic type (for example, two Parzen window estimators with different window widths), or estimators of different types (for

example, EBF vs. MVG). The estimator with the highest log probability is chosen as the optimal estimator for the particular application.

This article first describes current parametric and nonparametric density estimation techniques and then introduces the proposed EBF technique. The selection of the best technique using cross validation and the log probability is then shown. This process is demonstrated with three examples of density estimation.

Density Estimators

Multivariate Gaussian

The multivariate Gaussian (MVG) is a standard functional form for parametric density estimation. The assumption underlying the use of the MVG function is that the data are samples from a multivariate Gaussian distribution. The mean $\boldsymbol{m}$ and the covariance $\boldsymbol{Q}$ of the data are calculated in the standard way, and the resulting density function is (Ross, 1987):

$$\rho(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^n \det(\boldsymbol{Q})}} \exp\left(-\frac{(\boldsymbol{x}-\boldsymbol{m})\boldsymbol{Q}^{-1}(\boldsymbol{x}-\boldsymbol{m})^T}{2}\right) \quad (1)$$

Parzen windows

The Parzen window technique is described by Parzen (1962). Each point in the training set is designated as a unit center and an identical basis function is constructed at each center. The functional form of the estimator is:

$$\rho(\boldsymbol{y}) = \frac{1}{K}\sum_{k=1}^{K} \Psi(\boldsymbol{y}-\boldsymbol{x}_k) \quad (2)$$

where Ψ is a basis function that satisfies the following criteria to ensure $\rho(\boldsymbol{y})$ is a valid density function:

$$\Psi(\boldsymbol{u}) \geq 0 \quad (3)$$

$$\int_{-\infty}^{\infty} \Psi(\boldsymbol{u})d\boldsymbol{u} = 1 \quad (4)$$

By satisfying the following conditions:

$$\sup_{u} \Psi(\boldsymbol{u}) < \infty \quad (5)$$

$$\lim_{\|u\|\to\infty} \Psi(\boldsymbol{u}) \prod_{i=1}^{n} u_i = 0 \quad (6)$$

$$\lim_{K\to\infty} \sigma^n(K) = 0 \quad (7)$$

$$\lim_{K\to\infty} K\sigma^n(K) = \infty \quad (8)$$

the Parzen window estimate will converge to the true distribution of the data in the limit of infinite data (Duda and Hart, 1973). Equations 5 and 6 ensure Ψ is bounded in the vertical and horizontal directions respectively, and are satisfied by most density functions that would be considered for use as basis functions. Equations 7 and 8 impose conditions on the choice of the window width as a function of the number of data points, $\sigma(K)$, to ensure that the window size shrinks to zero but at a slower rate than the growth of the number of data points. Parzen investigated many different forms of basis functions (Ψ) that satisfy Eqs. 3 to 8, but the one most often used is spherically symmetrical MVG, Eq. 1 with diagonal covariance $\boldsymbol{Q} = \sigma^2\boldsymbol{I}$. Because the training set fixes the unit centers, the method is fully described by the single parameter σ. The optimal choice of σ is described later in this article.

Elliptical basis functions

The proposed EBF density estimator has fewer basis functions than Parzen window estimators and the basis functions are nonspherical and not identical for each unit. This allows for capture of the local character of the data. The technique has two parameters, the number of basis functions H, and the amount of unit overlap P. The form of each basis function is similar to that of a multivariate Gaussian, excluding a constant multiplicative factor:

$$a_h(\boldsymbol{x}) = \exp\left[-(\boldsymbol{x}-\boldsymbol{m}_h)\boldsymbol{Q}_h^{-1}(\boldsymbol{x}-\boldsymbol{m}_h)^T\right] \quad (9)$$

a_h is to be interpreted as the "activation" of each basis function. There are two values that must be found for each unit, the unit center $\boldsymbol{m}_h$, and the unit shape matrix $\boldsymbol{Q}_h^{-1}$. The first two steps in finding these values are essentially identical to the training procedure for radial basis function networks proposed by Moody and Darken (1989), and involve setting unit centers and hyperspherical unit widths. In the first step, H unit centers ($\boldsymbol{m}_h$) are assigned using the k-means clustering algorithm (MacQueen, 1967). Then basis functions are placed at each center. Initially these basis functions are given hyperspherical shapes. Thus, each of the H centers has associated with it a single width parameter σ_h. The unit widths are found using the following P-nearest neighbor heuristic. The width of any unit (σ_h) is taken as the root mean square distance to the P nearest centers:

$$\sigma_h = \sqrt{\sum_{p=1}^{P} \frac{\|\boldsymbol{m}_h - \boldsymbol{m}_p\|^2}{P}} \quad (10)$$

The next step is to adapt the units to the local character of the data. The covariance of the training points "captured" by each unit is taken to get the $\boldsymbol{Q}_h$ matrix for each basis function. The inverse of this matrix defines the unit's shape. To determine which training points are "captured" by a particular unit the following weight function is used:

$$w_h(\boldsymbol{x}_k) = \exp\left(-\|\boldsymbol{x}_k - \boldsymbol{m}_h\|^2/\sigma_h^2\right) \quad (11)$$

The weight for point $\boldsymbol{x}_k$ in unit h is $w_h(\boldsymbol{x}_k)$. The weighting function is used instead of a hard or crisp boundary because such hard boundaries imply a threshold value that would have to be set arbitrarily. Using the continuous membership function, the points that are near the unit center have a significant influence on the shape of the unit, while points far away from

the center have a small effect on the shape. Using all the data in the shape calculation, even though some of it is weighted very little, helps orient all of the basis functions in a smooth manner. Smoothness would not necessarily be achieved if hard boundaries around each unit were used to define the data that belongs to and does not belong to a particular unit.

The shape matrix Q_h is calculated as a weighted covariance of the data points. The membership function used in calculating the weighted covariance matrix for unit h is the activation of each training point, $w_h(x_k)$. To calculate the weighted covariance matrix for a particular unit h, the weighted mean is calculated for each unit and each dimension of the input space:

$$wm_{h,j} = \frac{\sum_{k=1}^{K} x_{k,j}\, a_h(x_k)}{\sum_{k=1}^{K} a_h(x_k)} \tag{12}$$

where j goes from 1 to n (thus forming a vector wm_h), x_k is the kth training example, and $x_{k,j}$ is the (k, j)th element in the X matrix, which is made up of the K row vectors x_k. The weighted mean vector is then subtracted from each of the training examples to form the centered data matrix C_h,

$$C_h = \begin{bmatrix} x_1 - wm_h \\ x_2 - wm_h \\ \vdots \\ x_K - wm_h \end{bmatrix} \tag{13}$$

The weighted covariance matrix for a particular unit h is $C_h^T \operatorname{diag}(w_h) C_h/(K-1)$. This gives the desired local shape of the unit, but in order to get the overlap specified from the P heuristic, the covariance matrix for each unit h is rescaled. The Mahalanobis distances from unit h to all other unit centers are calculated using $C_h^T \operatorname{diag}(w_h) C_h/(K-1)$ as the covariance matrix for unit h. Then the root mean square Mahalanobis distance (RMSM) of the P nearest centers is used as the scaling factor:

$$\mathrm{RMSM}_h = \sqrt{\sum_{p=1}^{P} \frac{(m_h - m_p)[C_h^T \operatorname{diag}(w_h) C_h/(K-1)]^{-1} (m_h - m_p)}{P}} \tag{14}$$

Introducing these scale factors to assure the desired overlap, the final form for the unit shape factors is:

$$Q_h = \left[\frac{C_h^T \operatorname{diag}(w_h) C_h}{K-1}\right] * \mathrm{RMSM}_h \tag{15}$$

After the data in the input space has been covered by elliptical basis functions, the next task is to combine the basis functions to calculate the overall PDF. To do this, we first calculate a local density in the region covered by unit h. The local density in any defined space is estimated by (Duda and Hart, 1973):

$$\rho_h = \frac{n_h}{n_t V_h} \tag{16}$$

where n_h is the number of data points in the defined space, n_t is the total number of points, and V_h is the volume of the defined space. If the defined space has a hard boundary then it is easy to count every point inside the boundary once to get the number of points in the defined space. EBF employ overlapping continuous basis functions that do not have hard boundaries. Thus each term in Eq. 16 requires redefinition consistent with the idea of a region whose boundaries are not crisply defined. We will state these definitions without justification, and then show that they lead to a valid density estimator. To obtain a count of the number of points associated with a particular unit h we use the activation of a point with respect to that unit as a weighting function. The total number of effective points associated with unit h is given by:

$$n_h = \sum_{k=1}^{K} a_h(x_k) \tag{17}$$

This defines an effective number of points that are contained within unit h. For the mutually-exclusive hard boundary case, the total number of points in the total space is equal to the sum of the number of points in each of the defined spaces. The number of points in a particular unit for the continuous case is defined as an effective number of points. Thus, the total number of points in the space must also be defined in terms of an effective number of points. This is accomplished by summing the effective number of points from each of the individual units:

$$n_t = \sum_{h=1}^{H} n_h \tag{18}$$

The volume of a unit is taken as the integral of the activation function:

$$V_h = \int_{-\infty}^{\infty} a_h(x)dx = \sqrt{\pi^n \det(Q_h)} \tag{19}$$

Equations 16 to 19 give a local density estimate for basis function h:

$$\rho_h = \frac{\sum_{k=1}^{K} a_h(x_k)}{\left(\sum_{h=1}^{H}\sum_{k=1}^{K} a_h(x_k)\right)\sqrt{\pi^n \det(Q_h)}} \tag{20}$$

In Parzen windows, the density estimate is made by an equally weighted average of the contributions of each of the basis functions. The Parzen window density estimate is:

$$\rho(x) = \frac{1}{H}\sum_{n=1}^{H} \Psi_h(x) \tag{21}$$

The term $1/H$ can be considered the portion of the total probability mass given to each unit. In Parzen windows, the probability mass is equally divided among units. In the EBF estimator, units may represent different portions of the probability mass, since units are not identical. Intuitively, the prob-

ability mass associated with unit h is given by the ratio of the effective number of points captured by the unit to the total effective number of points, n_h/n_t. This leads to the analogous form for nonidentical units:

$$\rho(x)=\sum_{h=1}^{H}\left(\frac{n_h}{n_t}\right)\Psi_h(x) \tag{22}$$

By definition Ψ must integrate to 1, but it is convenient in EBF estimators to divide it into two parts:

$$\Psi_h(x)=\frac{a_h(x)}{\int_{-\infty}^{\infty}a_h(x)dx}=\frac{a_h(x)}{V_h} \tag{23}$$

Using this definition of Ψ in Eq. 22 leads to the density estimate for the EBF estimator:

$$\rho(x)=\sum_{h=1}^{H}\left(\frac{n_h}{n_t}\right)\left(\frac{a_h(x)}{V_h}\right)=\sum_{h=1}^{H}\left(\frac{n_h}{n_tV_h}\right)a_h(x)$$

$$=\sum_{h=1}^{H}\rho_h a_h(x) \tag{24}$$

To summarize, to construct an EBF estimator, first the number of units H, and the overlap parameter P are chosen. Optimal choice of these values is shown later in this article. The unit centers m_h, and the unit shapes Q_h are then found using k-means clustering and Eq. 15, respectively. The density estimate for each unit, ρ_h, is then calculated from Eqs. 16 to 19. The density value for new points is obtained by using Eq. 24.

Properties of EBF Estimators

The definition of a probability density function requires that it be nonnegative at all points and that it integrates to unity over the entire space. It is obvious from Eq. 24, and the definitions of a_h (Eq. 9) and ρ_h (Eq. 16), that the EBF density estimate is always nonnegative. The integral of the EBF estimator comes from integrating Eq. 24:

$$\int_{-\infty}^{\infty}\rho(x)dx$$

$$=\int_{-\infty}^{\infty}\sum_{h=1}^{H}a_h(x)\rho_h dx \tag{25a}$$

$$=\int_{-\infty}^{\infty}\sum_{h=1}^{H}\exp\left[-(x-m_h)Q_h^{-1}(x-m_h)^T\right]\frac{n_h/n_t}{V_h}\,dx \tag{25b}$$

$$=\sum_{h=1}^{H}\left\{\frac{n_h/n_t}{V_h}\int_{-\infty}^{\infty}\exp\left[-(x-m_h)Q_h^{-1}(x-m_h)^T\right]dx\right\} \tag{25c}$$

$$=\sum_{h=1}^{H}\frac{n_h/n_t}{V_h}V_h \tag{25d}$$

$$=1 \tag{25e}$$

and satisfies this criterion. Equality between Eqs. 25c and 25d results from the definition of the volume of unit being the integral of its activation function over the entire space.

The most essential feature of a density estimator is that it converges to the true underlying distribution of the data. The proof we give here for the convergence of the EBF estimator to the underlying distribution function follows directly from the proof of Parzen window estimators converging to the underlying density function in the limit of infinite data, first given by Parzen (1962) and then by Duda and Hart (1973). In Parzen windows, letting the number of data points go to infinity also implies that the number of basis functions goes to infinity. In EBFs there are fewer basis functions than the number of data points. For convergence, we must allow the number of basis functions, H, to go to infinity but at a slower rate than the number of data points K. We also let P tend to infinity but at a slower rate than H:

$$K\to\infty,\ H\to\infty,\ P\to\infty,\ H<<K,\text{ and }P<<H \tag{26}$$

To prove convergence, consider a differential area of the input space in the case where we have infinite data. By proving that the EBF estimator converges to the underlying distribution in an arbitrary differential area, it can be directly concluded that the estimator will converge to the underlying distribution over the entire space. In this differential element there are K' data points and H' units that satisfy the K and H relationship in Eq. 26. If the differential element is small enough then the data will be uniformly distributed over the element. The k-means clustering algorithm that sets the unit centers will distribute the centers uniformly over the element because the data is uniform (H' centers are chosen randomly from the K' data points in the first step of the k-means algorithm, as the k-means algorithm starts moving the unit centers; the center is pulled equally in all directions because of the uniform distribution of the data in the differential element resulting in no net movement of the center).

The shape matrix of the EBF becomes diagonal in the infinite data limit because there is no correlation among the input variables in the differential element. Furthermore, each diagonal element of the covariance matrix is the same, because the data is distributed uniformly in all directions. This means that the EBF units are hyperspherical.

Each unit has the same activation function and therefore the sum of the activations in Eq. 17, n_h, will be the same for each unit in the differential element. Thus in the limit of infinite data the ratio of n_h/n_t becomes:

$$\frac{n_h}{n_t}=\frac{n_h}{\sum_{h=1}^{H'}n_h}=\frac{n_h}{H'n_h}=\frac{1}{H'} \tag{27}$$

Using Eqs. 9 and 16 to write Eq. 24 the result is:

$$\rho(y)=\frac{1}{n_t}\sum_{h=1}^{H'}\frac{n_h}{V_h}\exp\left[-(y-m_h)Q_h^{-1}(y-m_h)^T\right] \tag{28}$$

applying the simplifications that have been shown, in the limit of infinite data for a differential element in the input space we have:

$$\rho(y)=\frac{1}{H'}\sum_{h=1}^{H'}\frac{1}{V_h}\exp\left[-(y-m_h)(\sigma^2 I)^{-1}(y-m_h)^T\right] \quad (29)$$

which is exactly a Parzen window estimator for the unit centers m_h. In the differential area the data is uniformly distributed and the centers are also uniformly distributed, thus converging to the distribution of the unit centers is the same as converging to the distribution of the data points. This means that in the differential area, the EBF density estimator will converge to the underlying distribution. The same result holds for any differential element in the space. Since the EBF estimator converges to the underlying distribution function in every differential element, then it can be concluded that it converges to the underlying distribution in the entire space in the limit of infinite data.

Model Selection and Optimum Model Structure

When one (as always) has a finite data set, an optimal estimator with particular finite H and P will exist. The choice of H and P or choice of the Parzen window width should be based on a measure of future performance. The purpose of these density estimators is to predict the density of a new, previously unseen data point. The measure of how well a particular estimator approximates a density function is the product of the probabilities of sample points independently and randomly drawn from the true underlying distribution (Traven, 1991). For a test set drawn from the true underlying distribution of size n_{test}, the overall fit measure is given by:

$$J'=\prod_{i=1}^{n\,\text{test}}\rho(x_i) \quad (30)$$

For convenience of calculation, usually the log of Eq. 30 is used:

$$J=\log(J')=\sum_{i=1}^{n\,\text{test}}\log\left[\rho(x_i)\right] \quad (31)$$

The parameters of a particular method that maximizes the log probability are the ones most suited to the particular data density estimation task.

The difficulty in using the log probability is that an independent test set of data is required. Data is limited in most applications and it is desirable to calculate the estimator using all available data. The well-known technique of cross-validation (Weiss and Kulikowski, 1991) can be used to fully utilize the data set while still providing test sets which were not used to calculate the estimator. The s-fold cross validation procedure, where s is the number of test sets, is as follows:

(1) Divide the training data into s equally-sized (or almost equally-sized if s does not divide evenly into the number of training points) randomly chosen sets.

(2) Fix parameters for the density estimation technique (for example, H and P)

(3) For $i=1$ to s, repeat: withhold set i; train estimator on all training data except set i; and use Eq. 31 to calculate log probability of set i.

(4) Sum up the values of J calculated for each subset in step 3. This is the total log probability for the parameters chosen in step 2.

This procedure can be repeated for various estimator parameters. The parameters that maximize the log probability are the best parameters for the density estimation task at hand. Once the parameters have been decided upon then the entire training data set is used to train an estimator with those optimal parameters. Determination of the log probability allows for optimization of the parameters of Parzen window and EBF techniques; it also is a valid criterion for selecting one method of density estimation over another. After parameter optimization, the method that has the highest log probability is the most applicable method for approximating the underlying density function of the data. Cross validation can also be used for estimators whose parameters are found deterministically (such as MVG) to find a comparable log probability. This can then be compared to other log probabilities to select the best method. For the MVG method, step 2 of the cross validation algorithm can be omitted.

For the purposes of validating the cross validation method, if the underlying distribution is known, we can normalize the log probability calculated by cross validation by the number of training points, and compare it to the normalized log probability of a new test set drawn from the parent distribution. If the cross validation estimate is valid, these two numbers should be close to one another. This is demonstrated in the next section.

Examples

Three abstract examples of using the EBF technique for estimating probability density functions were investigated. Case 1 is a two-dimensional Gaussian distribution, centered at the origin with an identity covariance matrix, $N(\mathbf{0}, I)$. Case 2 is a two-dimensional bimodal distribution. Case 2 data has a 0.55 probability of being drawn from a $N(\mathbf{0}, I)$ distribution and a 0.45 chance that the data is drawn from a Gaussian distribution centered at (4, 2.5) with a covariance matrix of diag (1.5, 0.2). Case 3 data is seven-dimensional. It has the same distribution as case 2 in the first two dimensions, and has five other dimensions, each of which has an independent Gaussian distribution of data centered at the origin with a variance of 0.07.

Although all of the examples have diagonal covariance, they are equivalent to more general problems. In the bimodal case, translations, rotations, and linear scaling can be used to move one of the distributions to the origin and make it spherical. The distributions can then be rotated about the origin until the other distribution has its axes aligned with the principal axes, having a diagonal covariance matrix (Van Ness, 1980). Therefore, this problem represents two Gaussian distributions with arbitrary centers and arbitrary covariance matrices.

For each case, a training set of 100 points was generated, as well as a separate test set of 1,000 points. The training set was used in the cross validation methodology to choose optimal parameters and estimate log probabilities for multivariate Gaussian, Parzen windows, and EBF density estimators. Density estimates for the 1,000 point test sets were then made for each of the estimators and a log probability calculated. Table 1 shows that the log probabilities estimated by the cross validation procedure are good estimates of the actual performance

Table 1. Normalized Specific Log Probabilities from Six-Fold Cross Validation Procedure (CV) and from the 1,000 Point Test Sets*

Case		log(p[x]) MVG	PW	EBF
1	CV	−1.214	−1.249	−1.257
	Test	−1.253	−1.261	−1.278
2	CV	−1.639	−1.326	−1.299
	Test	−1.604	−1.464	−1.397
3	CV	1.116	−0.1468	1.111
	Test	0.995	−0.0162	1.051

*Multivariate Gaussian (MVG), Parzen windows (PW), and elliptical basis functions (EBF).

of the estimator. The test set log probability is within 12% of the cross validation log probability for all estimators except the Parzen window estimator of case 3 data. This estimate is an order of magnitude different than the test set log probability. The order of magnitude difference in the log probability however is only a 25% absolute difference. The absolute (nonlogged) cross validation and test set probabilities differ by 3 to 37% for the cases studied.

Figures 2 through 4 show mesh plots of the actual density function and the estimates generated by the three methods under consideration. In each plot, the two horizontal axes are the two independent variables and the vertical axis is the density value. For case 3 the two dimensions of interest are shown and the other five dimensions are held at their nominal value of 0. The J' value appearing is the normalized absolute probability (not logged) from the 1,000 point test set.

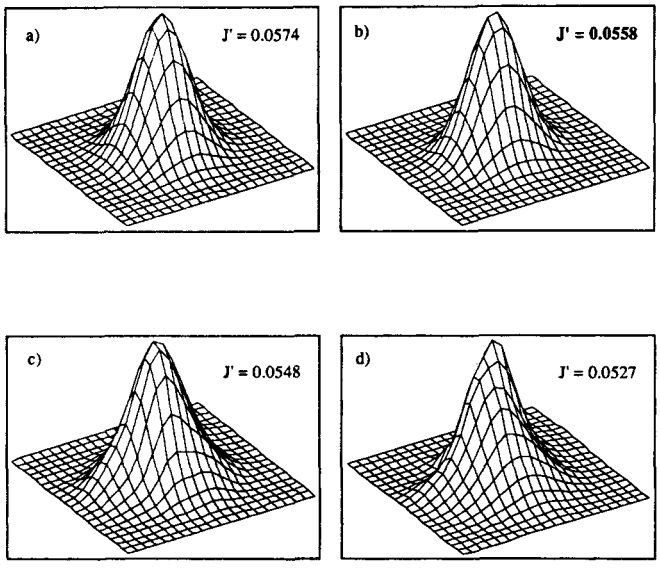

Figure 2. Mesh plots for (a) actual density function, (b) multivariate Gaussian estimator, (c) Parzen window estimator, and (d) elliptical basis function estimator for case 1.

Parzen window width was 0.5419, and EBF had $H = 15$ units with an overlap of $P = 1$. Parzen window width and EBF parameters were optimal parameters for data set for this case. J' values shown are product of the densities for 1,000 point test set, normalized by number of test cases. Largest value, indicating best estimator, is shown in bold.

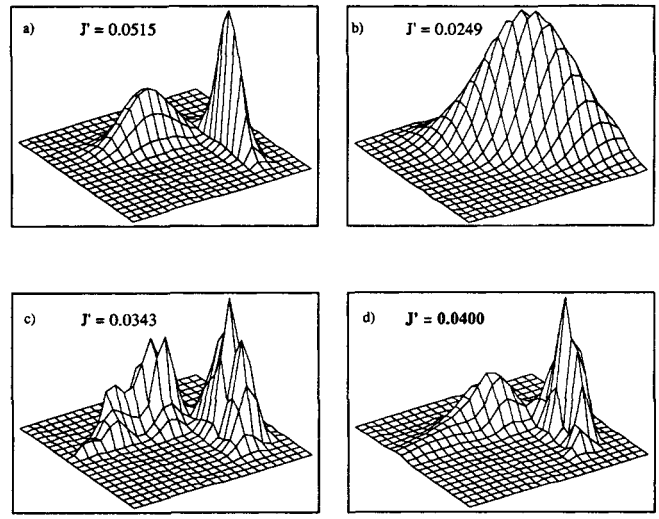

Figure 3. Mesh plots for (a) actual density function, (b) multivariate Gaussian estimator, (c) Parzen window estimator, and (d) elliptical basis function estimator for case 2.

Parzen window width was 0.3420, and EBF had $H = 12$ units with an overlap of $P = 1$. Parzen window width and EBF parameters were the optimal parameters for data set for this case. J' values shown are product of densities for 1,000 point test set, normalized by number of test cases. Largest value, indicating best estimator, is shown in bold.

The best predictor in case 1 was the multivariate Gaussian estimator, as expected. The Parzen window and EBF estimators give adequate estimates, but are poorer than the multivariate Gaussian. For the bimodal (case 2) data the

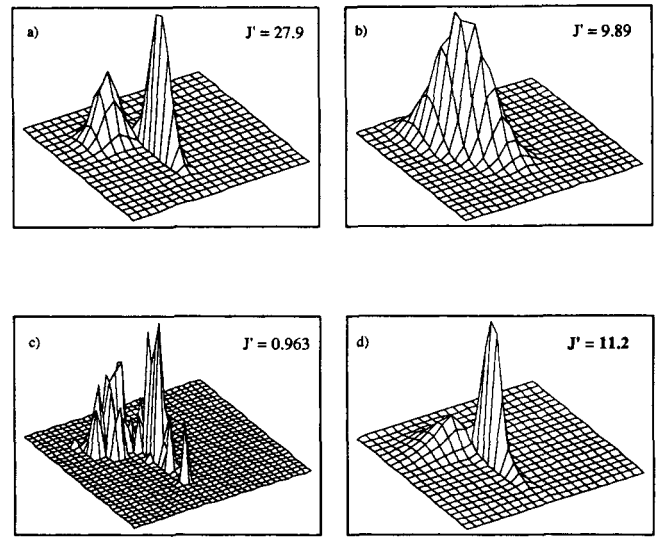

Figure 4. Mesh plots (a) actual density function, (b) multivariate Gaussian estimator, (c) Parzen window estimator, and (d) elliptical basis function estimator for case 3.

Parzen window width was 0.1503, and EBF had $H = 8$ units with an overlap of $P = 1$. Parzen window width and EBF parameters were optimal parameters for data set for this case. J' values shown are product of densities for 1,000 point test set, normalized by number of test cases. Largest value, indicating best estimator, is shown in bold.

multivariate Gaussian is unable to capture the bimodal feature and thus is a poor estimate of the true density function. The Parzen window estimator captures the bimodal nature of the data but produces a density estimate that is not smooth enough. The optimal value of $\sigma = 0.342$ was obtained by optimizing the predictive capability of the estimator. Increasing sigma, which would have the effect of smoothing the density estimate, would not improve the estimator's predictive ability. The best estimate of the case 2 data comes from the EBF estimator. The bimodal nature of the data is captured and the density estimate is relatively smooth. The case 3 density estimates are similar to the case 2 estimates. The multivariate Gaussian is unable to capture the bimodal nature of the data in the two dimensions shown. In the other five dimensions, which are Gaussian, the multivariate Gaussian fits these very well. The resulting probability score is $J' = 9.89$. The added dimensionality shows the shortcoming of Parzen window estimators. There is only one unit width and the close packing of the points in the five dimensions not shown cause small unit sizes ($\sigma = 0.1503$) resulting in a very spiky density estimate and a correspondingly low J' value of 0.963. The EBF density estimate captures the bimodal nature of the data in a smooth fashion in the two dimensions shown. The variable width of the elliptical units in each dimension allows for small widths in the five remaining dimensions. This eliminates the spikiness problem exhibited by Parzen windows.

Using cross validation to decide on the best method of density estimation for a given set of data has been shown to work on three sets of test data. The elliptical basis function estimation has also been shown to overcome some of the problems of more conventional density estimation techniques such as multivariate Gaussian and Parzen window estimators.

Classification Examples

Elliptical basis function estimators can be used to make maximum likelihood decisions about the classification of a data point. This ability is useful in fault detection and fault diagnosis. To illustrate the application of EBF estimators, the problem first presented by Kramer and Leonard (1990) is used. This problem is a simplified version of many static fault diagnosis problems. The measured states are represented by vector Y, and the process is at a nominal steady state Y_0. Malfunctions are represented by changes in fault parameters q. These parameters are the extent of the fault, and are zero at the normal operating condition. The measurement Y is given by:

$$Y = Y_0 + f(q) + v \tag{32}$$

where v represents measurement noise. In this example the extent of the fault has a linear effect on Y:

$$Y = Y_0 + \alpha q + v \tag{33}$$

Each column of the matrix α represents the directional effect of one of the fault parameters on the measurement vector. Fault categories are then defined by inequalities on the fault parameters:

$$C_k: g_k(q) > 0, \quad k = 1, \ldots nc \tag{34}$$

This formulation of the failure model is quite general and conceptually fits the Hoskins and Himmelblau (1988) model of three CSTRs in series, the reactor example of Watanabe et al. (1989), and other classification problems.

In the work of Kramer and Leonard (1990) and Holcomb and Morari (1991), a two-dimensional form of the problem posed in Eq. 33, was studied. In both of these works, cases are limited to only one element of q being nonzero at a time. The noise v is taken to be zero mean Gaussian, and the fault direction matrix α is:

$$\alpha = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

Normal (C_0): $|q_1| < 0.05, \ |q_2| < 0.05$

Fault 1 (C_1): $|q_1| \geq 0.05$

Fault 2 (C_2): $|q_2| \geq 0.05$

$$v_1, v_2 = N(0, 0.015)$$

Fault 1 is when Y_1 and Y_2 both fail in the same direction, and fault 2 is when Y_1 and Y_2 fail in opposite directions. This results in the "X" shape, with the normal class at the intersection of the two fault classes, shown in Figure 5. Training and test data were generated by sampling the fault extent q_i from a normal distribution, $N(0, 0.25)$. If the absolute value of the fault extent was less than 0.05 then the point was assigned to the normal class.

Training sets consisting of 30 examples per class were generated, as well as test sets of 1,000 examples per class. Elliptical basis function density estimators and Parzen window density estimators were fit to the data from each class. The EBF H and P parameters and the optimal window width for the Parzen window estimator were decided upon using the cross validation technique previously described in this article. For each type of estimator, the class that the point had the highest density in was assigned as the class for that point. This is a maximum likelihood decision, assuming equal class priors. To make a statistically meaningful comparison between methods, the comparison was repeated 500 times and average results were calculated. Table 2 shows the EBF and Parzen window results for 500 independently chosen pairs of training data and test data. The mean correct identification rate is given as well as the standard deviation of correct identification for each class.

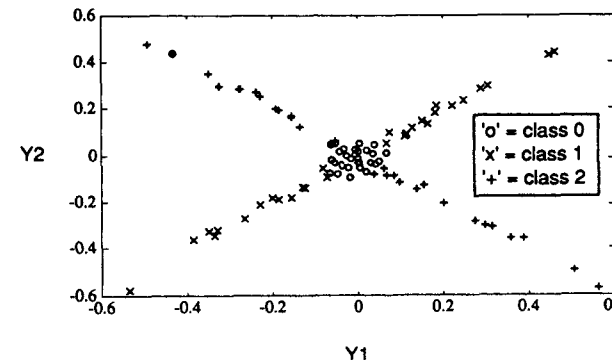

Figure 5. Typical set of training data for two-dimensional classification problem.

Table 2. Performance of EBF and Parzen Window Density Estimators on the Two-Dimensional Classification Problem*

	Normal	Fault 1	Fault 2	Overall
EBF				
Mean	94.1	93.1	92.4	93.2
Std. Dev.	2.9	4.7	4.3	2.2
PW				
Mean	95.3	90.2	89.2	91.6
Std. Dev.	2.4	5.2	4.6	2.8

*Results are for 500 independently chosen sets of 30 training points per class, and 1,000 test points per class; results given in percent.

The overall score is the total percentage of correctly identified cases. The table shows that the EBF slightly outperforms the Parzen window estimator, 93.2% vs. 91.6%. It is important to note that the EBF had about one-third the number of basis functions of the Parzen windows. The optimal EBF had from 9 to 11 basis functions and the Parzen window estimator had in each case 30 basis functions. It is also not surprising that the Parzen window estimator does well as the data in this problem have about the same variance in each direction and the data is quite evenly spread over the entire range of the sensors.

The classification problem formulated in Eq. 33 is quite general and applies to any number of dimensions. Kramer and Leonard (1990) present a ten-dimension classification problem that has an α matrix of:

$$\alpha^T = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}$$

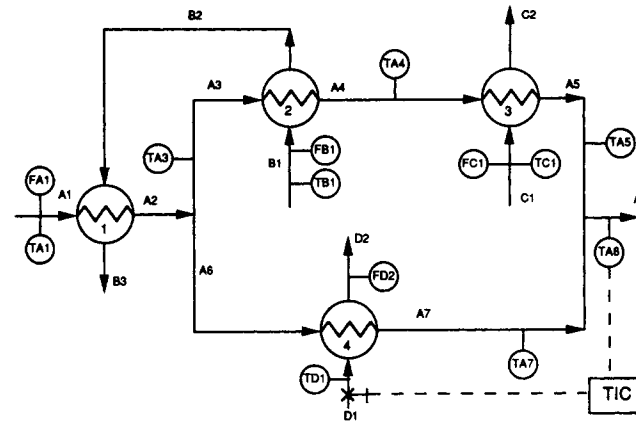

Figure 6. Heat exchanger network example of sensor fault identification.

In all exchangers stream A is cold stream.

The same testing procedure for the two-dimension problem was used for the ten-dimension problem, although only 100 train-and-test set pairs were used to find the average performance, as opposed to the 500 train-and-test set pairs used for the two-dimension problem. For the ten-dimension problem, 90 examples per class were used for training. The overall score for the EBF was 89.4% correctly identified classes, with a standard deviation of 2.9%. The Parzen window estimator performed almost as well, with an overall score of 85.3%, and a standard deviation of 3.7%. In this problem, the Parzen window estimator had about six times as many basis functions as the EBF. The Parzen window estimator had 90 basis functions and the EBF estimator had from 13 to 16 basis functions.

The heat exchanger network (HEN) used by Tjoa and Biegler (1991) with modifications was used as a process example of fault classification using the density estimation technique. The HEN consists of four input streams and four heat exchangers. Fluid A is heated by fluid B in two of the exchangers and by fluids C and D in the other exchangers (see Figure 6). A controller manipulates the flow of stream D to maintain the temperature of the outlet stream at 615°C. Although a controller is used, only steady-state measurements were used. Streams A and B were assumed to be process streams that operate at low, medium, and high flow rates, with Gaussian process noise about each operating point. The input flows and temperatures were drawn from distributions described in Table 3. The flow split between FA3 and FA6 was specified as FA3 = 0.42*FA2.

The heat transfer for each heat exchanger was calculated using:

$$q = UA \frac{[(T_{hi} - T_{co}) - (T_{ho} - T_{ci})]}{\ln\left(\frac{T_{hi} - T_{co}}{T_{ho} - T_{ci}}\right)} \tag{35}$$

where q is the heat transferred, U is the overall heat-transfer coefficient, A is the heat exchanger area, T is temperature, and subscript h refers to the hot stream, c to the cold stream, i to the inlet, and o to the outlet. A heat capacity of unity for all fluids was used. The values of UA for each exchanger were calculated from Tjoa and Beigler's results and were 400, 265, 492, and 1,158, for exchangers 1 to 4, respectively. Equation

Table 3. Input Distributions of Flows and Temperatures for the Heat Exchanger Network

FC1 = $N(240,2)$
TA1 = $N(450,1)$
TB1 = $N(620,1)$
TC1 = $N(670,1)$
TD1 = $N(690,1)$
TA8 = $N(615,1)$
FA3/FA2 = 0.42
Probability = 0.3
FA1 = $N(800,2)$
FB1 = $N(200,2)$
Probability = 0.4
FA1 = $N(900,2)$
FB1 = $N(250,2)$
Probability = 0.3
FA1 = $N(990,2)$
FB1 = $N(320,2)$

35 for each exchanger was used in conjunction with the steady-state mass and energy balances to calculate all flows and temperatures given the input conditions. The flow rate of stream D was set by the controller such that temperature TA8 would be 615°C. Sensor noise was added to each of the measurements using the same method of Tjoa and Biegler. All temperature measurements had zero-centered Gaussian noise with a standard deviation of 0.75°C. All flow measurements had zero-centered Gaussian noise with a standard deviation equal to 2% of the flow.

Before doing density estimation on the data set, the dimensionality was reduced from 13 to 2 using principal component analysis. It was found that 95% of the variation in the data set could be captured by two principal components. Density estimation on the two-dimensional data was done using a 200 point data set with multivariate Gaussian, Parzen window, and elliptical basis function estimators. Six-fold cross-validation was used to find the predicted log probability, and a 1,000 point separate test set was used to find an application log probability. The EBF estimator with 22 units and overlap of 1 was the best at capturing the underlying distribution of the data, having a predicted J' value of 0.0471. This was larger than both the Parzen window (optimal $\sigma = 0.987$) value of 0.0241 and the multivariate Gaussian value of 0.0173. The test set J' values were 0.0467, 0.0276, and 0.0172 for EBF, Parzen window, and multivariate Gaussian estimators respectively.

Data sets were calculated for positive and negative biases of two and four degrees in the outlet temperature sensor TA8, using the same method and inputs previously described. EBF, Parzen window, and multivariate Gaussian density estimators were fit to these data sets after principal component analysis was used to reduce the dimensionality to two. Separate test sets of 1,000 points each were also calculated for the positive and negative biases of two and four degrees in TA8. The bias in TA8 effects the flow of stream D selected by the controller.

The following procedure was used for the EBF, Parzen window, and multivariate Gaussian estimators. The density of each point in the test set was calculated using the estimators for positive and negative biases of two and four degrees in TA8, and using the normal state. The point was assigned membership in the class of the estimator with the highest density. The results are shown in confusion matrices shown in Table 4. The training and test points both contained zero-centered Gaussian noise

Table 4. Fault Classification Tables for Sensor Bias Classification in a Heat Exchanger Network

Bias Identified	Bias Present −4	−2	0	+2	+4
EBF Estimator					
−4	96.2	1.5	0	0	0
−2	3.7	94.4	2.7	0	0
0	0.1	4.1	93.8	2.6	0
+2	0	0	3.5	95.9	15.0
+4	0	0	0	1.5	85.0
Multivariate Gaussian Estimator					
−4	65.5	17.6	4.2	0.7	0
−2	32.2	47.4	10.0	0.1	0
0	2.2	32.4	56.1	11.8	1.0
+2	0.1	2.5	28.2	60.6	19.5
+4	0	0.1	1.5	26.8	79.5
Parzen Window Density Estimator					
−4	91.0	6.7	0	0	0
−2	9.0	79.2	3.1	0	0
0	0	14.1	92.2	5.7	0
+2	0	0	4.7	86.2	6.9
+4	0	0	0	8.1	93.1

with a standard deviation of 0.75°C. The class boundaries used in Table 4 are 2°C (only 2.67 standard deviations), which leads to some misclassification of the faults. Despite this, the EBF estimator was able to identify the correct class over 90% of the time, in all but one case. The diagonal sum of the confusion matrices represents the total number of correctly identified cases. The EBF estimator had the highest diagonal sum of 465.3 out of a possible total of 500. The Parzen window estimator was second best, having a diagonal sum of 441.7, and the multivariate Gaussian was the poorest having a diagonal sum of 309.1. The relative performance of the estimators in the class identification problem is the same as the estimators' relative performance in the density estimation task, as measured by the predicted log probability. This confirms the probability as a figure of merit for predicting performance of density estimators. The EBF outperformed the Parzen window estimator even though the Parzen window estimator used 200 basis functions compared to the EBF estimator's 22. Since the EBF estimator has fewer basis functions, calculations with the EBF are an order of magnitude faster than the Parzen window estimator.

Conclusions

A new technique of estimating data density functions has been introduced. The elliptical basis function density estimator has been shown to overcome the restrictions of parametric density estimators, while reducing the number of units and spikiness problem of Parzen window estimators. A methodology for selecting the best density estimator for a particular application, using cross validation and the log probability, has also been presented. The cross validation methodology and the EBF estimator was demonstrated on three density estimation test problems. The results lead the authors to believe that both the EBF estimator and the cross validation methodology are good techniques for data density estimation and model discrimination for multivariate correlated data. The EBF estimator was also shown to be the best estimator at fault classification for two abstract fault classification problems previously studied in the literature, as well as being the best

classifier for detecting sensor biases in a heat exchanger network. The EBFs outperformed the Parzen window estimator in all classification examples, even though the EBF method used up to an order of magnitude fewer basis functions. The lower computational load of the EBF estimator makes it more practical to use on-line where speed of calculation may be critical.

Notation

a_h = activation function for EBF (scalar)
C_h = centered data matrix in weighted covariance calculation $(K \times n)$
H = number of basis functions (scalar)
I = identity matrix
J = logarithm of J' (scalar)
J' = product of probability densities (scalar)
K = number of training set points (scalar)
m = mean vector of data set, used in multivariate Gaussian estimator $(1 \times n)$
m_h = center of unit h $(1 \times n)$
n = dimension of input space (scalar)
n_h = number of effective points contained in unit h (scalar)
n_t = total effective number of points (scalar)
n_{test} = number of points in test set (scalar)
nc = number of fault classes (scalar)
P = overlap heuristic parameter (scalar)
$P(x)$ = probability density of x (scalar)
Q = covariance matrix of data $(n \times n)$
Q_h = weighted covariance matrix of unit h $(n \times n)$
RMSM_h = root mean square Mahalanobis distance of unit h (scalar)
V = volume of Parzen window basis function (scalar)
V_h = volume of unit h (scalar)
w_h = weight function vector for unit $h(K \times 1)$
wm_h = weighted mean vector for unit h $(1 \times n)$
x = input vector $(1 \times n)$
X = matrix of input vectors $(K \times n)$
y = input vector $(1 \times n)$

Greek letters

ρ_h = density of unit h (scalar)
$\rho(x)$ = density of x (scalar)
σ = Parzen window width (scalar)
σ_h = width of weighting function for unit h (scalar)
Ψ = basis function (scalar)

Literature Cited

Duda, R. O., and P. E. Hart, *Pattern Classification and Scene Classification*, Wiley Interscience, New York (1973).

Holcomb, T., and M. Morari, "Local Training for Radial Basis Function Networks: Towards Solving the Hidden Unit Problem," Amer. Control Conf., Boston, p. 2331 (1991).

Hoskins, J., and D. M. Himmelblau, "Artificial Neural Network Models of Knowledge Representation in Chemical Engineering," *Comput. Chem. Eng.*, **12**, 881 (1988).

Kavuri, S. N., and V. Venkatasubramanian, "Representing Bounded Fault Classes Using Neural Networks With Ellipsoidal Activation Functions," *Comput. Chem. Eng.*, **17**, 139 (1993).

Kramer, M. A., and J. A. Leonard, "Diagnosis Using Backpropagation Neural Networks—Analysis and Criticism," *Comput. Chem. Eng.*, **14**, 1323 (1990).

Kramer, M. A., and R. S. H. Mah, "Model-Based Monitoring," Foundations of Computer Aided Process Operations, Crested Butte, CO (1993).

MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Stat. and Prob.*, Berkeley, CA, p. 281 (1967).

Moody, J., and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Comput.*, **1**, 281 (1989).

Parzen, E., "On Estimation of Probability Density Function and Mode," *Ann. Math. Stat.*, **33**, 1065 (1962).

Poggio, T., and F. Girosi, "Networks for Approximation and Learning," *Proc. IEEE*, **78**, 1481 (1990).

Renals, S., and R. Rohwer, "Phoneme Classification Experiments Using Radial Basis Functions," Int. Joint Conf. Neural Networks, Washington, DC, I, 461 (1989).

Ross, S. M., *Introduction to Probability and Statistics for Engineers and Scientists*, Wiley, Toronto (1987).

Tjoa, I. B., and L. T. Biegler, "Simultaneous Strategies for Data Reconciliation and Gross Error Detection of Nonlinear Systems," *Comput. Chem. Eng.*, **15**, 679 (1991).

Traven, H. G. C., "A Neural Network Approach to Statistical Pattern Classification by 'Semiparametric' Estimation of Probability Density Functions," *IEEE Trans. Neural Networks*, **2**, 366 (1991).

Van Ness, J., "On the Dominance of Non-Parametric Bayes Rule Discriminant Algorithms in High Dimensions," *Pattern Recognition*, **12**, 355 (1980).

Watanabe, K., I. Matsuura, M. Abe, M. Kubota, and D. M. Himmelblau, "Incipient Fault Diagnosis of Chemical Processes via Artificial Neural Networks," *AIChE J.*, **35**, 1803 (1989).

Weiss, S. M., and C. A. Kulikowski, *Computer Systems That Learn*, Morgan-Kaufman, San Mateo, CA (1991).

Manuscript received Mar. 22, 1993, and revision received Nov. 22, 1993.