11-1-1993

A Fuzzy Locally Sensitive Method for Cluster Analysis

Antonio G. Thome
Purdue University School of Electrical Engineering

Manoel F. Tenorio
Purdue University School of Electrical Engineering

A Fuzzy Locally Sensitive Method for Cluster Analysis

Antonio G. Thome
Manoel F. Tenorio

TR-EE 93-38
November 1993

School of Electrical Engineering
Purdue University
West Lafayette, Indiana 47907-1285

A Fuzzy Locally Sensitive Method for Cluster Analysis

Antonio G. Thome
and
Manoel F. Tenorio

School of Electrical Engineering
Purdue University
West Lafayette, IN 47907-1285

A FUZZY LOCALLY SENSITIVE METHOD FOR CLUSTER ANALYSIS

Abstract: Cluster analysis has been playing an important role in pattern recognition, image processing, and time series analysis. The majority of the existing clustering algorithms depend on initial parameters and assumptions about the underlying data structure. In this paper a fuzzy method of mode separation is proposed. The method addresses the task of multi-modal partition through a sequence of locally sensitive searches guided by a stochastic gradient ascent procedure, and addresses the cluster validity problem through a global partition performance criterion. the algorithm is computational efficient and provided good results when tested with a number of simulated and real data sets.

key words: clustering analysis; fuzzy clustering; mode separation; cluster validity

I - INTRODUCTION

Cluster analysis plays a very important role in pattern recognition. In fact it represents an essential tool for learning and extracting information on those problems where very little previous knowledge is available about the data's structure. Cluster analysis is also known in the literature as unsupervised pattern recognition, and some basic reasons for the interest in such unsupervised procedures are usually cited as follows:

. the collection and labeling of a large set of sample patterns can be very expensive and time consuming;

. in many applications the characteristics of the patterns can change slowly with time, and a classifier running in unsupervised mode may achieve better performance;

. in the early stages of an investigation it may be valuable to gain some insight into the nature of the data set:

. in many applications like filtering and prediction, an unsupervised partition of the input space may lead to better accuracy through a divide-and-conquer approach.

Cluster analysis techniques are based on partitioning a collection of data points into a number of subgroups or clusters, where objects inside a subgroup show a higher degree of similarity as opposed as objects in different subgroups. In other words :it can be said that cluster analysis is used for partitioning multimodal distributions into unimodal subclasses in hope to facilitate the implementation of subsequent discriminant functions.

Three distinct cases for unsupervised learning can arise depending upon which parameters are known and which are unknown (table 1) [Duda73].

case	v_i	Σ_i	$P(w_i)$	C
1	unknown	known	known	known
2	unknown	unknown	unknown	known
3	unknown	unknown	unknown	unknown

Table 1 - Possible unsupervised cases

where

v_i	represents the centroid of the ith class
Σ_i	represents the scatter matrix of the ith class
$P(w_i)$	represents the ith class prior probability
C	represents the total number of classes

Case number 3 is very common in real data environments and imposes at least four major difficulties on any clustering procedure:

1. the lack of knowledge about the number of clusters requires a reliable validity criterion to highlight the optimal partition when achieved;

2. the a priori unknown location of the cluster centers usually requires initial guesses, which makes the algorithms very sensitive to starting points;

3. variations in shape, size, and orientation of each class may lead to meaningless results; and

4. outliers may induce misclassification and impose non-existant structures.

Roughly all existing clustering procedures can be classified into two general categories as globally sensitive methods and locally sensitive methods [Kitt76]. Methods in the first category represent the clusters by centroids or kernels, and globally assign the data to them so that a measure of similarity between the points and the clusters is optimized. Methods in the second category make use of the local structure of the data as reflected, for example, in the probability density.

Unfortunately methods in both categories suffer from inherent drawbacks. Global methods often generate clusters whether they really exist or not, i.e., regardless of the data's probabilistic structure. It is taking it to the extreme to say that the clustering procedure is able to cluster even true random data. Dubes and Jain [Dubes79] throughly discuss the problem and suggest measures for clustering tendency before the application of any clustering procedure. Local methods, on the other hand, often give too much emphasis to the clata's structural details and as result tend to generate an excessive number of clusters.

Global methods are much more popular than local ones due to their simplicity, efficacy, and computational efficiency, which seem to outweight the well known drawbacks for many users. Dynamic clustering [Kitt88,Dida74,Dida78] is one of the various attempts that have: been made to overcome these problems, while retaining the computational attractiveness of the algorithm. The idea is to have multiple, instead of single point, reassignment at each iteration, which suggests that the clustering criterion function may show some plateaus in the search for the local minimum that can be traversed only by simnltaneous reassignment of groups of different points. However, the combinatorial complexity of the reassignments may make the procedure impractical.

The diversity of clustering algorithms is very large. Many are based on iterative relocation, which starts with an initial classification and attempts to improve it iteratively by moving samples from one group to another. Others are based on hierarchical agglomeration, which starts, for example, with each sample forming a separate group and successively merges those groups close to one another. Some are model-free, others are model-based where the classes are assumed to have fixed shapes, say spherical or ellipsoidal, and fixed or varying

sizes. and orientations. And finally, the clustering procedures may work in a crisp (hard) or a fuzzy partition scheme.

The distinction between hard and fuzzy partition scheme is related to the way each sample is attached to the set of clusters. In hard clustering, a sample can only belong to a unique cluster, as opposed as to fuzzy clustering where it may belong to the entire set of clusters through different degrees of membership. The use of fuzzy theory in clustering goes back to the work of Bellman et al. [Bell66], Ruspini [Rusp69, Rusp70], and Gitman and Levine [Gitm70]. In 1973 Dunn [Dunn73] defined the first generalization of the conventional minimum-variance hard clustering, and still in 1973 Bezdek [Bezd73] introduced the well known Fuzzy C-mean Algorithm (FC-mean). Both hard and fuzzy methods are now equally used over all those distinct approaches mentioned in the previous paragraph.

Since the optimal number of clusters and the data structure are usually unknown, it is of fundamental importance to have a kind of performance criterion able to provide a feel for the goodness of the resulting partition. Avoiding imposed structures (data overfit) as well as lack of accuracy (data underfit) are the main goals to be achieved. It is claimed [Dube79,Xie91] that the engineering literature has paid very little attention to cluster validity issues, limiting the effort to showing that the new clustering algorithm performs reasonably well on a few data sets, often in two dimensions.

In this report we propose a new clustering algorithm. This algorithm performs the pattern space partition through a sequence of locally sensitive searches combined with a global validity criterion. The algorithm is computationally efficient and provides good results on both artificially generated and real data sets. In section 2 an overview of the basic clustering concepts and a description of some well known procedures is shown. In sections 3 and 4 the proposed procedure is discussed, and the results of some experiments are reported in section 5. Extensions to the basic procedure are proposed in section 6, and some more results are then reported. Conclusions and ongoing research directions are presented in section 7.

II - CLUSTERING ALGORITHMS

Clusters are defined as groups of points in the feature space that are similar according to a predefined criterion or *measure* of *similarity*. Usually, similarity is defined as proximity of the points according to a distance function. With the similarity criterion on hand it is necessary to partition the space into subgroups or clusters of similar points. The methods for finding the partition may or may not assume parametric forms, may have an heuristic basis, or may be more rigorously dependent on the minimization of a mathematical cost function often called *criterion function*. In all cases, iterative procedures are generally used.

II.1 - Similarity Measures

Once the clustering problem is described as one of finding natural groups among the data set, it is necessary to define what natural group is and how to identify them. Although this issue may be application dependent, the most obvious and widely used measure of similarity is the distance between pair of points. Euclidean distance is by far the most used, which provides characteristics of invariance to translation and to rotation to the clustering procedure. But it does not provide invariance to general linear transformations or any other transformation that distorts the distance relationships.

$$d_{ij} = [(X_j - V_i)^t (X_j - V_i)]^{1/2} \quad (1)$$

where

$X_j \in R^d$, $j=1 \ldots n$ is the sample observation
$V_i \in R^d$, $i=1 \ldots c$ is the cluster centroid

The above observation calls attention to the fact that if clusters are to be meaningful, they should be invariant to those transformations most natural to the problem. Ideally, clustering algorithms should be insensitive to changes in the similarity criterion.

11.2 - Criterion Functions

The definition of a criterion function to measure the quality of the partition at each iteration is the usual way to transform the clustering problem into a well defined optimization problem. Through this transformation the clustering problem becomes one of finding the partition that extremizes the criterion function. Some of the most used criteria are based on the Sum-of-Squared-Errors and Scatter Mamces.

a) Sum-of-Squared-Error Criterion

It is the simplest and most widely used criterion, defined as follows:

$$J(V) = \sum_{i=1}^{c} \sum_{X \in X_i} \| X - V_i \|^2 \quad (2)$$

where

$V = [V_1 \ldots\ldots V_c]$, is a $(d \times c)$ mamx of cluster centers
$X_i = (X \mid X \in \text{cluster } i)$
c is the number of clusters
$\| . \|^2$ is the Euclidean norm
and

$$V_i = \frac{1}{n_i} \sum_{X \in X_i} X \quad (3)$$

This criterion has a simple interpretation which states that for a given cluster X_i, the mean vector V_i is its best representative in the mean squared error sense. Algorithms of this type are often called *minimum variance*. It is well known that minimum variance is an appropriate criterion when classes form well separated compact clouds. Problems arise when there are great differences in terms of class populations and shapes.

b) Scatter Matrix Criteria

This is a family of criterion functions derived from the scatter matrices used in multiple discriminant analysis, which is a generalization of Ficher's linear discriminant [Duda, Fisher]. The criteria are based on the following definitions:

. mean vector of the ith cluster

$$V_i = \frac{1}{n_i} \sum_{X \in X_i} X \tag{4}$$

. total sample mean vector

$$V = \frac{1}{n} \sum_{i=1}^{n} X_i = \frac{1}{n} \sum_{i=1}^{n} n_i V_i \tag{5}$$

. scatter matrix for the ith cluster

$$S_i = \sum_{X \in X_i} (X - V_i)\,(X - V_i)^t \tag{6}$$

. within-cluster scatter mamx

$$S_w = \sum_{i=1}^{c} S_i \tag{7}$$

. between-cluster scatter matrix

$$S_B = \sum_{i=1}^{c} n_i (V_i - V)\,(V_i - V)^t \tag{8}$$

. total scatter mamx

$$S_T = \sum_{X \in X_i} (X - V)\,(X - V)^t \tag{9}$$

It turns out that the total scatter matrix is the sum of the intra-cluster and the inter-cluster matrices; see [Duda73] for more details. The total scatter mamx does not depend on how the samples are partitioned, whereas the intra-cluster and the inter-cluster do, which suggests the existence of a tradeoff between these two matrices, i.e. when one goes up the other must go down. Therefore, by trying to minimize the intra-cluster mamx one is also tending to maximize the inter-cluster matrix.

Scalar measures of the size of these scatter mamces are necessary in order to use them as criterion functions. The three most popular ones are:

.The Trace Criterion

$$tr(S_w) = \sum_{i=1}^{c} tr(S_i) = \sum_{i=1}^{c} \sum_{X \in X_i} \| X - V_i \|^2 \qquad (10)$$

Which is exactly equal to the aforementioned minimum variance criterion. It was shown above that by minimizing **tr** (S,) we are also maximizing **tr** (S_B).

. The Determinant Criterion

$$| S_w | = | \sum_{i=1}^{c} S_i | \qquad (11)$$

This approach measures the square of the scattering volume, since it is proportional to the product of the variances in the directions of the principal axis.

. Invariant Criterion

$$tr(S_w^{-1} S_B) = \sum_{i=1}^{d} \lambda_i \qquad (12)$$

These eigenvalues are invariant under non-singular linear transformations of the data, and their values measure the ratio of intra-cluster to inter-cluster scatter matrices in the direction of the eigenvectors. Partitions leading to large values of the criterion function are desirable.

11.3 - Clustering Algorithms

Once a criterion function has been selected, clustering becomes a well-defined problem in discrete optimization. Since the sample set is finite, there is only a finite number of possible partitions. Thus, in theory, if ones assumes that the number of classes is known then any clustering problem can always be solved by exhaustive enumeration. However, in practice such an approach is completely infeasible for most applications. Iterative optimization is the most frequently used approach in the searching for optimal partitions. The basic idea is to start from a reasonable or even arbitrary partition and then to reassign samples if such a move improves the criterion function. Like hill-climbimg procedures, this approach guarantees local but not global optimization, is dependent on the starting configuration, and as result one never knows whether or not the best solution has been found. *ISODATA* and ***K-means*** are the best known representatives of this class of algorithms.

a) Hard K-means

This is a very simple and intuitive algorithm, which served as basis for *ISODATA*, later developed by Ball and Hall [Ball67]. It assumes previous knowledge of the number of classes, and uses the Euclidean distance as the similarity measure. The algorithm's major steps can be summarized as follows:

step 1 - begin with an arbitrary assignment of samples to the clusters;

step 2 - compute the sample mean of each cluster;

step 3 - reassign each sample according to the nearest mean;

step 4 - if the classification of all samples has not changed, stop; else go to step 2.

b) Fuzzy C-means

This is the fuzzy version of the hard k-means, developed by Bezdek in 1973 [Bezd73]. The algorithm makes use of a weighted version of the sum-of-squared-error criterion function.

$$J_m(U,V)= \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ij}^m \, \| X_i - V_j \|^2 \tag{13}$$

subject to

$$\sum_{j=1}^{c} u_{ij}^m = 1, \qquad 1 \le i \le n \tag{14}$$

$$u_{ij} \ge 0, \qquad 1 \le i \le n, \qquad 1 \le j \le c \tag{15}$$

where

m is an scalar greater than 1;
$V_j \in R^d$ is the cluster center
$\|.\|$ is the Euclidean norm
u_{ij} is the degree of membership of sample i wrt cluster j
$U = \{ u_{ij} \}$, (n x c) membership matrix
$V = \{ V_i \}$, (d x c) matrix of clusters centers

The algorithm computes the cluster centroids and the degrees of membership according to the following rules:

$$V_j = \frac{\sum_{i=1}^{n} u_{ij}^m X_i}{\sum_{i=1}^{n} u_{ij}^m} \tag{16}$$

and

$$u_{ij} = \frac{\left(\frac{1}{d_{ij}^2}\right)^{m-1}}{\sum_{k=1}^{c} \left(\frac{1}{d_{ik}^2}\right)^{m-1}} \tag{17}$$

where

$$d_{ij}^2 = \| X_i - V_j \|^2 \tag{18}$$

and

$$\text{if } d_{ij} = 0, \text{ then } u_{ij} = 1$$

The FCM algorithm major steps are:

step 1 - initialization: arbitrarily select membership values between [0,1] satisfying $\sum_j u_{ij} = 1$, and set k =O;

step 2 - compute the centroids $V_j(k)$ using $u_{ij}(k)$;

step 3 - compute $u_{ij}(k+1)$ using $V_j(k)$;

step 4 - if $f(U(k),U(k+1)) < \varepsilon$ stop, else set k= k+1 and go to step 2.

Bezdek [Bezdek87] shows that the algorithm converges to a local minimum satisfying:

$$J_m(U^*,V^*) \leq J_m(U,V^*), \text{ and} \tag{19}$$

$$J_m(U^*,V^*) \leq J_m(U^*,V) \tag{20}$$

c) ISODATA

Introduced by Ball and Hall [Ball67], this algorithm provides a way for determining the number of clusters through the use of heuristic tools for splitting and merging the existing clusters. The algorithm always mes to split if the total number of clusters is less than half of the user desired number, and to merge if the current number is more than twice this number. Several parameters need to be previously specified, which requires from the user a good intuition or a reasonable knowledge about the structure of the data.

T	threshold on the number of samples in a cluster,
N_D	approximate (desired) number of clusters,
σ_s^2	maximum spread parameter for splitting,
D_m	maximum distance separation for merging, and
N_{max}	maximum number of clusters that can be merged.

The algorithm major steps are:

step 1 - cluster the data set into c classes, eliminating any data and classes with fewer than T members. Exit when classification has not changed;

step 2 - if $c \leq \frac{N_D}{2}$ and iteration is odd, then

a. split any cluster whose spread is larger than σ_s^2
b. if any cluster has been split, go to step 1;

step 3 - if $c > 2N_D$, then merge any pair of clusters whose samples are sufficiently close: and / or overlapping;

step 4 - go to step 1.

d) Hierarchical Clustering

These are based on either hierarchical agglomeration or division of the pattern space. Agglomerative, or bottom-up procedures, start with **n** singleton clusters and successively merge those close to one another. On the other hand, Divisive, or top-clown procedures, start with all samples in one cluster and successively split those far to one another. For every hierarchical clustering there is a corresponding tree, called dendrogram, that shows how the samples are grouped.

The basic steps in agglomerative clustering are:

step 1 - let c = n and $X_i = \{X_i\}$, i=l ... n

step 2 - if c = 1, stop.

step 3 - find the nearest pair within two distinct clusters, say X_i and X_j

step 4 - merge both clusters and decrement c by 1

step 5 - go back to step 2

Beciiuse of their conceptual simplicity, hierarchical procedures are among the best known methods [Dimi88, Murt92]. However, they suffer from drawbacks like sensitivity to outliers, tendency to impose structure, and computational effort required for the pairwise distance computation at every tree level.

e) Locally Sensitive Methods

Techniques in this class try to exploit the local structure of the data as reflected in some statistical parameters like, for example, the probability density function. It is implicitly assumed that the samples form well-defined clouds in a d-dimensional space, and it is often assumed that they come from a mixture of **c** normal distributions which means that the optimal partition falls into hyperellipsoids of various sizes and orientations. Of course,if the samples are definitively far from a normal distribution then the second-order statistics will be incapable of capturing the underlying structure, and a misleading partition may result.

The problem of estimating the parameters of a mixture density is not trivial, particularly in situations where relatively little a priori knowledge about the nature of the data is available. The assumption of any particular paramemc form may lead to poor results where structure may be imposed rather than found. Alternatively, nonparametric methods such as Parzen Windows or K-nearest neighbors may be used.

Parzen window was originally proposed by Parzen in 1962[Parz62] for one-dimensional distributions and later extended to the n-dimensional case by Cacaullos [Caco66]. Its general formulation for the n-dimensional case is

$$\hat{p}(x)=\frac{1}{N}\sum_{i=1}^{N}\rho^{-d}\,\gamma\left(\frac{\hat{x}-x_i}{\rho}\right) \qquad (21)$$

where

N	represents the number of samples
ρ	is a constant parameter denoting the adopted scale (bin)
$\gamma(.)$	is the kernel function or Parzen window
$\hat{x}$	is the estimated mean

In general, both the parameter ρ and the kernel function $\gamma(.)$ can be chosen by the user. The most widely used kernel functions are:

a) hypercubic kernel

$$\gamma(.)= \begin{cases} \frac{1}{(2\rho)^d} & \forall\ \hat{x}\ \text{s.t.}\ |\hat{x}_j - x_{ij}| \leq \rho \\ 0 & \text{otherwise} \end{cases} \tag{22}$$

b) hyperspherical kernel

$$\gamma(.)= \begin{cases} \frac{1}{V} & \forall\ \hat{x}\ \text{s.t.}\ |\hat{x}_j - x_{ij}| \leq \rho \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

c) exponential kernel

$$\gamma(.)= \frac{1}{(2\rho)^d} \exp\left(\frac{-|\hat{x} - x_i|}{\rho}\right) \tag{24}$$

d) Gaussian kernel

$$\gamma(.)= \frac{\rho^{-d}}{(2\pi)^{d/2}\ |\Sigma|^{1/2}} \exp\left(\frac{-1}{2\rho^2} (\hat{x} - x_i)^t\ \Sigma^{-1}\ (\hat{x} - x_i)\right) \tag{25}$$

The major problem inherent to Parzen estimation is that it may easily overestimate the distribution when the kernel is too broad, or underestimate it when the kernel is to narrow.

III - THE FUZZY LOCALLY SENSITIVE PROCEDURE

This paper presents a new clustering algorithm. The proposed algorithm approaches the clustering problem from the perspective of partitioning a multimodal pattern space into a set of unimodal subspaces. This is done by assuming that the data set comes from a mixture density, and that each subspace defines a homogeneous subpopulation or cluster. Here the term homogeneous is used in the sense that all points in the same group are more similar to each other than to points in any other group according to a pre-specified similarity criterion.

The proposed algorithm falls into the class of locally sensitive methods, where local structures of the data are captured and evaluated as possible representatives of significant classes. The algorithm, in contrast to some previous work reported in the literature, as in [Gitm70] and [Kitt76], adopts a simplified version of the exponential Parzen estimate as an energy function. This energy function is continuous and differentiable, which allows the use of simple hill-climbing procedures to detect the modes, or peaks, of the underlying mixture density .

The notion of fuzzy sets adds global measure to the algorithm, which strengthen's its capability of detecting highly concentrated subpopulations and of not being trapped by spurious points or outliers. Low computational effort, guaranteed convergence, and low sensitivity to starting points, are major features of this algorithm.

III.1 - The Mode Detecting Concept

The proposed algorithm is model-based, performs an iterative optimization of a criterion function, and realizes a sequential partition of the pattern space. The following assumptions are made: a) the observations are d_dimensional; b) the observations form a discrete set I? of size n; c) the nature of Γ is a mixture of normal densities; and d) no other information is available. The assumption that the samples come from a mixture of normal observations may be seen as a restriction to the algorithm, but it is useful to recall that the central limit theorem supports the idea for a large variety of situations of interest. Therefore, the larger the training set, the more reliable is the resulting partition.

The: problem of estimating the parameters of a mixture density is not trivial, particularly when almost no a priori knowledge is available. Mixture densities are resultants of random processes where the samples are assumed to be obtained by selecting a state of nature, say class w_j, with a certain probability $P(w_j)$ and then selecting from it one sample X according to the distribution $p(X/w_j,\theta_j)$. The total probability density for the sample is then given by:

$$p(X/\theta)=\sum_{j=1}^{c} p(X/w_j,\theta_j)\ P(w_j), \qquad \forall\ X \in \Gamma \tag{26}$$

where

$\theta=(\theta_1 \ldots \theta_c)$, are the local density parameters
$p(X/w_j,\theta_j)$, is the component (local) density wrt class **j**
$P(w_j)$, is the prior probability of class **j**, also called the mixing parameter

In cases like the one assumed here, where no knowledge is available about the number **c** of modes, the local densities, the prior probabilities, or the local density parameters, the

paramemc approach cannot be used. The alternative is nonparametric tools like the Parzen estirnate or k-nearest neighbors.

Kitler [Kitt76] proposes a mode separation technique based on a cubic Parzen window and a function that maps the d-dimensional observations into a sequence of scalar points. The mapping operation is done such that observations belonging to same mode tend to become successive elements of the sequence. Intervals on the sequence are then assumed to separate distinct modes. The major drawbacks of the method are the necessity of extra discriminant functions to classify those ambiguous points not included into any identified mocle, the possibility of the cubic estimate being trapped by smoothed valleys where the mapping function has no hint for choosing the correct sequence, and also the excessive computational effort required for large data sets.

The procedure proposed here mes to mitigate such problems through the adoption of a continuously differentiable kernel function, here named energy kernel, that navigates freely around the pattern space. The most dense concentrations, which define local maxima in the space, are then detected by an attractive force felt by the itinerant kernel. The energy kernel is a simplified version of the exponential Parzen estimate defined as:

$$g(v_j)=\sum_{i=1}^{n} \exp\left(\frac{-d_{ij}}{b}\right) \tag{27}$$

wheae

d_{ij}	is any similarity measure
b	is a constant specifying the size of the energy kernel
v_j	is the center of gravity of the jth energy kernel prototype

The modes of the underlying mixture density are sequentially detected by throwing a new energy kernel prototype into the pattern space at every new stage and letting it converge, or be attracted, by one of the subpopulations not identified yet. The kernel prototypes navigate through an iterative hill-climbing procedure which searches for local maxima of the surface defined by the energy function. The main advantage of this approach compared to the hypercubic used by Kitler is the infinite and very flexible window provided by the exponential kernel prototype, as seen in figure 1. By controlling the parameter b one controls the degree of resolution of the search process.

The:major steps of the algorithm can be summarized as follows:

step 0 - initialization: set $j = 0$, and goodness= $-\infty$;

step 1 - select starting energy kernel center: set $j=j+1$; and $V_j=X_i \ \forall i$;

step 2 - maximize the energy with respect to v_j: max $g(V_j)$;

step 3 - evaluate the partition, if rejected stop.

step 4 - adjust the pattern space according to the current partition, i.e.

$$X=\{ \mathbf{X} \mid \mathbf{X} \notin Xj\}$$

step 5 - go back to step 1.

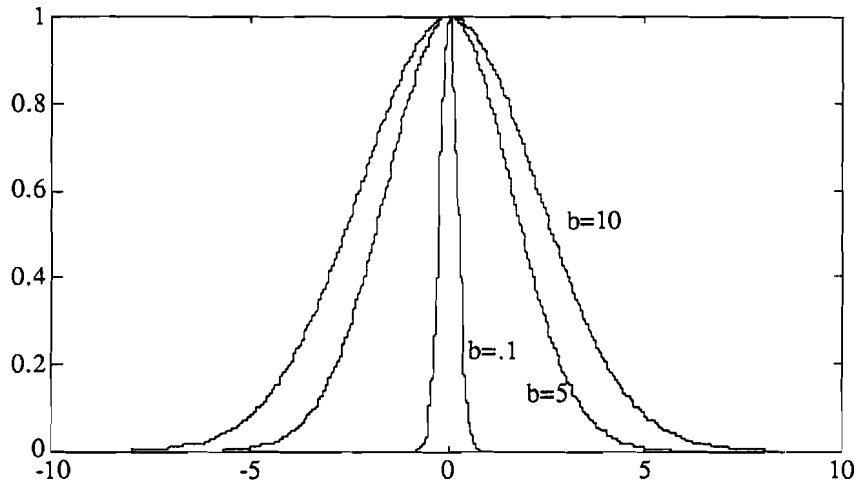


Fig 1 - Kernel prototype window for different values of **b**

Every loop starts with a new energy prototype being thrown into the pattern space and ends with the detection of a significant mode or subclass. The starting energy kernel center is always randomly selected from the sample set, which implicity takes into account the unknown prior probability of the underlying classes, since those with higher probability will have a greater chance of being chosen. This also reduces the convergence time.

The maximum of the kernel energy in the step two is found through the following unconstrained optimization scheme:

$$\max g(V_j) = \sum_{i=1}^{n} \exp\left(\frac{-\|X_i - V_j\|^2}{b}\right) \tag{28}$$

where the stepwise update of Vj is given by

$$V_j(k+1) = V_j(k) + \alpha(k)\frac{\partial g}{\partial V_j(k)} \tag{29}$$

and

$$\alpha(k) = \frac{1}{g(k)} \quad \text{is the normalized gain coefficient}$$

The normalized gain provides stability and speed to the algorithm, so that when the kernel function is far from any subpopulation the gain increases and allows the kernel to be more sensitive to the surrounding forces of attraction. On the other hand, when a particular mode is detected the gain decreases in proportion to its energy intensity, thus trapping the kernel.

The algorithm performs a sequential partition of the space and the goodness at every stage is evaluated through a global fuzzy validity scheme. The fuzzy measurement takes into account all modes already detected up to that point and performs the validation over the entire set of observations. In fact, it adds a global measure to the local searches. If the partition is positively evaluated, i.e., if its degree of goodness improved with respect to the

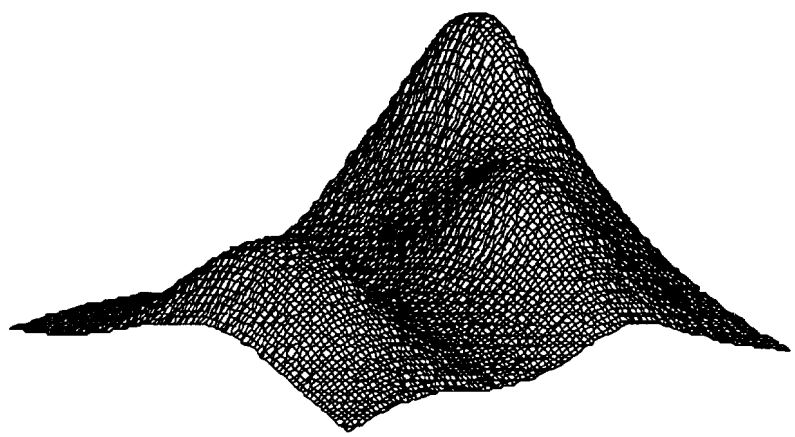
previous one, then all observations seleted as belonging to the detected subpopulation are masked, as in the sense of hard classification. This is done so that they do not influence the navigation of subsequent kernel prototypes.

Three points may have been noted as being critical to the algorithm: the size b of the kernel prototype, the sensitivity to starting configurations, and the partition goodness evaluation criterion. The validy problem is discussed in more detail in the next section, and as it is shown below, the other two points can be addressed with a reasonable extra computational effort.

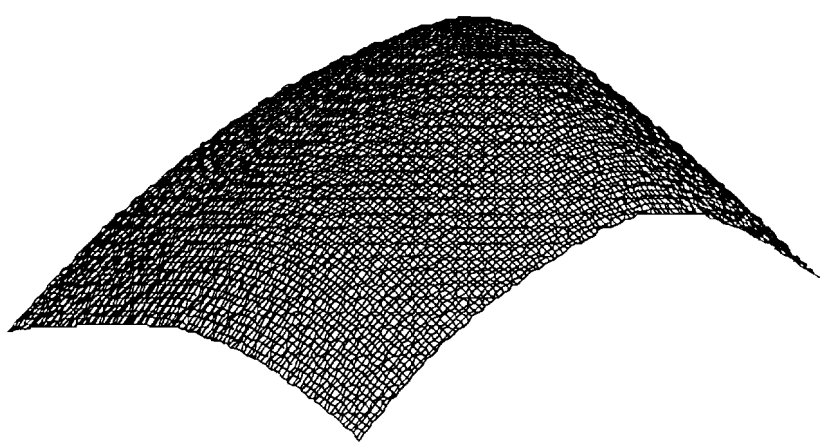
III.2 - Kernel Size

It is well known that the size of the kernel function is directly related to the quality of the resulting estimate of the probability density [Kitt76,Duda73,Ther89]. Figure 2 shows the surface detected by the energy kernel function when applied to a three-modal normal mixture density using different sizes. It is clear that either under or overestimates may easily result depending on the adopted scale, i.e. depending on the degree of resolution used.

(a)



(b)



(c)

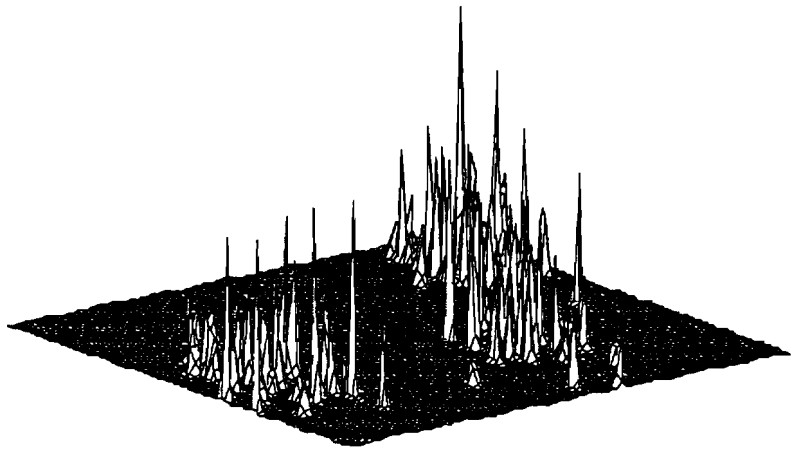


Fig. 2 - Surfaces detected by the energy kernel prototype, (a) adequate size; (b) too broad; and (c) too narrow

Intuitively, the original covariance matrix of the data set can be used as a rough indicaton of the size of the kernel prototype. Using a percentage of the largest eigenvalue, say 50%, we can establish a range of possible sizes for the kernel, repeat the entire algorithm for a sampling of sizes from this interval, and select the best partition according to the validity criterion. The practice of repeating the entire process to identify the optimatl partition is very common in clustering procedures. The advantage here is that this scheme still keeps the algorithm fast.

The classification of ambiguous points, i.e. points located on the boundary of two or more subpopulations, as well as outliers, are automatically resolved by the degree of membership induced by the fuzzy validity measure. The performance of the algorithm was evaluated on artificially generated data sets as well as on real data. The obtained results are discussed in section 5. Extensions to the basic algorithm are presented and discussed in section 6.

111.3 - Starting Points

Occasionally the kernel prototype may be trapped by clouds of spurious data. In general it moves toward the most dense subpopulation surrounding its inital position, but spurious densities on its way may have enough energy to attract and hold the prototype when it is close.

Since the starting center for each prototype is randomly selected from the pool of available samples, which decreases as the partitioning process flows, a reasonable strategy for avoiding spurious clusters is to repeat the search process a number of times, say 3, at each stage and select the best one. In practice, this made the algorithm very robust with respect to starting points.

111.4 - Convergence

The convergence of the algorithm is assured since the objective function is monotonic increased every step and the number of possible allocations is finite. Since the kernel energy function is not quadratic, the algorithm converges to local minima, which hopefully

represent the modes, or most important peaks, of the underlying distribution. Spurious peaks that can be occasionally detected are avoided through the global fuzzy validation.

The gradient optimization ends each step with the kernel prototype center converging to the weighted mean of those active data samples

$$v_i^* = \frac{\sum_{j \in \chi} a_j X_j}{\sum_{j \in \chi} a_j} \tag{29a}$$

where

$$\chi = \{ X / X \text{ is not masked })$$

$$a_j = \exp\left(\frac{-\| X_j - v_i \|^2}{b}\right)$$

IV - CHOICE OF VALIDITY SCHEMES

Despite some comments to the effect that very little attention has been dedicated to cluster validity issues, the research on this topic seems to be very active with several papers directly addressing the problem. Different schemes are available, for hard as well as for fuzzy environments. Through the validity measure, problems like the optimal number of clusters, the separability of clusters, and the overall goodness of the partition are addressed. Current approaches usually take into account parameters the compactness of each cluster, the isolation of the clusters within the environment, and the global fit which relates to the accuracy with which the partition describes the actual structure.

Cluster validity is critical to the performance of the algorithm we propose. A fuzzy criterion was preferred in order to add global information to the locally sensitive search performed by the energy kernel prototype. Several distinct schemes were evaluated and finally we decided on the ones proposed by Xie [Xie91] and Gath [Gath89]. Minor modifications were introduced to better reflect the desired compromise between local compactness and global fitness, and both schemes were tested in combination with the energy kernel prototype.

IV.1 - Xie's Validity Criterion

The suggested compactness and separation validity function S is defined as

$$S = \frac{\sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^2 \, \| X_j - V_i \|^2}{n \min_{ij}\left(\| V_i - V_j \|^2\right)} \tag{30}$$

where

u_{ij}	is the degree of membership of sample j to cluster i
V_i	is the centroid of cluster i
n	is the number of samples
$\|.\|$	is the Euclidean norm

S is the ratio of compactness (n) of the fuzzy c-partition to the separation (τ) of the clusters, defined as follows:

$$\pi = \frac{\sigma}{n} = \frac{1}{n}\sum_{i=1}^{c}\sigma_i = \frac{1}{n}\left(\sum_{j=1}^{n} u_{ij}^2 \, ||V_i - V_j||^2\right) \tag{31}$$

where

σ is the total variation (fuzzy squared error) of the c-partition

and

$$\tau = (d_{min})^2 = \min_{ij}\left(||V_i - V_j||^2\right) \tag{32}$$

where

d_{min} is the shortest distance between cluster centroids

Good partitions are associated with small values of S. Besides the tendency of S eventually decrease with the increase of c, it was also observed that the criterion some times favoured spurious densities, or nearly empty clusters, and rejected true subclasses close to one another. Such problems were mitigated by adopting a slightly different function:

$$S' = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n} u_{ij}^2 \, ||X_j - V_i||^2}{n_i} \tag{33}$$

The cardinality of each fuzzy cluster n_i is given by the summation of all corresponding degrees of membership and satisfies the property of adding to the total number of samples.

$$n_i = \sum_{j=1}^{n} u_{ij} \tag{34}$$

$$\sum_{i=1}^{c} n_i = n \tag{35}$$

If the data set really comes from a mixture of well defined densities then it is expected that the energy kernel prototype is able to produce relatively hard clusters with very small local variation, and that the validity criterion is able to reject those spurious ones and to allow the existence of clusters close to one another.

IV.2 - Gath's Validity Criterion

This criterion is based on the fuzzy covariance matrix, its hypervolume.,and the partition density. The fuzzy covariance matrix is an weighted scatter matrix computed from the perspective of each cluster as follows:

$$F_i = \frac{1}{n_i}\sum_{j=1}^{n} u_{ij} \, (X_j - V_i) \, (X_j - V_i)^t \tag{36}$$

[Bere91], the same entropy concept as it is used in information theory can be extended to the fuzzy clustering. There, in the information theory, the entropy for discrete events is defined as

$$H(x)= -\sum_{i} p_i \log_2 p_i \tag{42}$$

Here, assuming that the degree of membership can be viewed as an estimate of the probability of a particular sample to belong to a particular class, it becomes straightforward to define the fuzzy entropy as

$$H_i(X)= -\frac{1}{n}\sum_{j=1}^{n} (u_{ij}\log_2 u_{ij}+(1-u_{ij})\log_2(1-u_{ij})) \tag{43}$$

This :function generates the graph shown in figure 3, where maximum fuzziness occurs for values of membership equal to 1/2. For the multi-modal mixture case, assuming the independence of the modes, a global partition fuzziness measure can be computed by just adding the individual mode measures together.

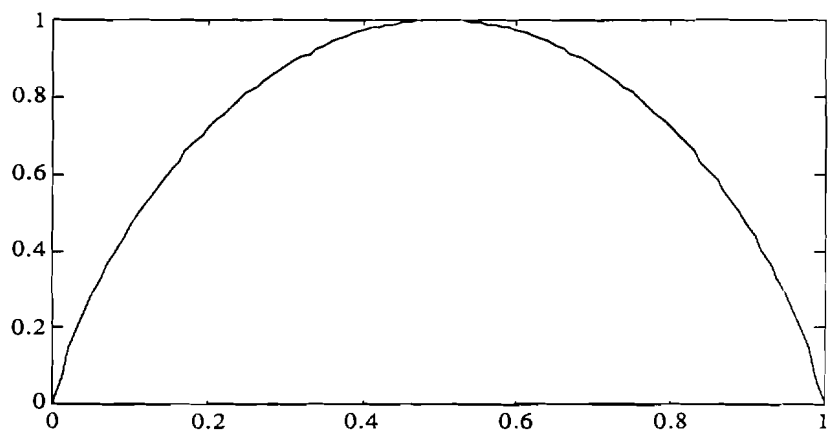


Fig. 3 - Measure of Fuzziness Function

V - ISXPERIMENTAL RESULTS

A simulation environment was created using Matlab and a Sun Sparc workstation. The basic energy maximization procedure was tested in combination with both validity measurement schemes, as described in the last section, and several sets of artificially generated as well as real data points were used. In this section some of the experiments and the results obtained are described and compared to some of the well known algorithms.

V.l - Data Sets

As described in Appendix A, four artificially generated and two real data sets were used in the experiments. The artificial sets are all, but one, from normal mixture densities. Data sets D1 arid D2 are two-dimensional and data set D3 is three-dimensional. Data set D4 is from an uniformly random two-dimensional distribution. It may be clear that not having natural clusters does not necessarily imply that the data is random, but the reverse is necessarily true, i.e. if the data is random no cluster shall be detected.

For the real data environment we chose the four-dimensional Iris data set which has been widely used since the work on linear discriminants reported by Fisher in 1956. The other set is a five-dimensional data used in an attempt to define the nature of chemical diabetes using a multidimensional analysis. According to Andrew [Andr85], this data set was visually inspected at the Stanford Linear Accelerator Computation Center and it was observed that the three primary variables show a configuration resembling a boomerang with a fat middle and two wings. From the clinical point of view the middle points represent the normal subjects and the two wings, the chemical and overt diabetic subjects.

V.2 - Artificial Data Tests

Example 1: Two-Dimensional Data Set (Dl) - This simple case was used to show the performance of the algorithm over a well defined and well separated mixture of Gaussian dismbutions. As shown in figure 5.1, 300 samples clustered in eight independent Gaussian dismbutions of different sizes and densities were considered for this experiment.

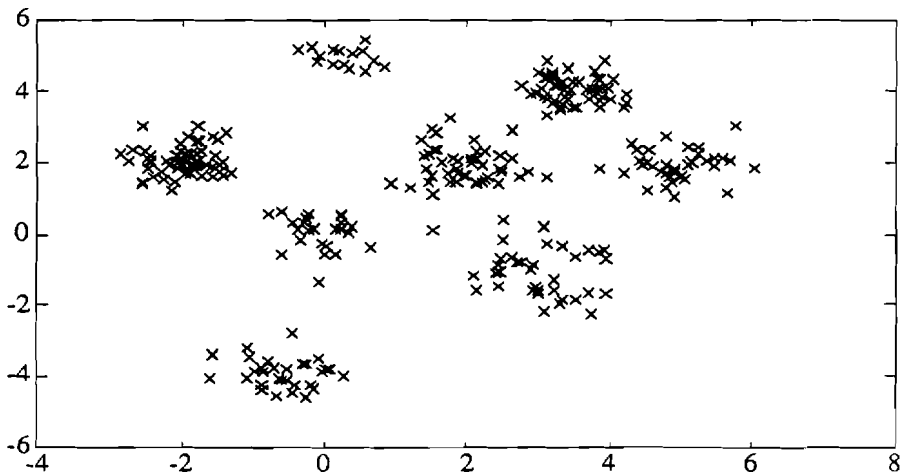


Fig. 5.1 - Two-Dimensional Data Set used for example 1: 300 points clustered in 8 independent Gaussian dismbutions of different sizes and densities

A pool of 15 different sizes for the kernel prototype, ranging from .5 to 6.5, was used and both validity criterion schemes were checked against their ability to identify the ideal kernel size and the best partition. Figure 5.2 shows the results: the number of detected clusters, the partition goodness measure, and the kernel size for both criteria.

(a) (b)

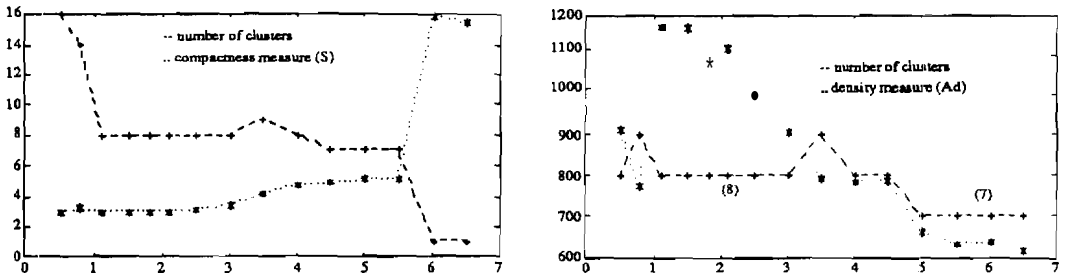
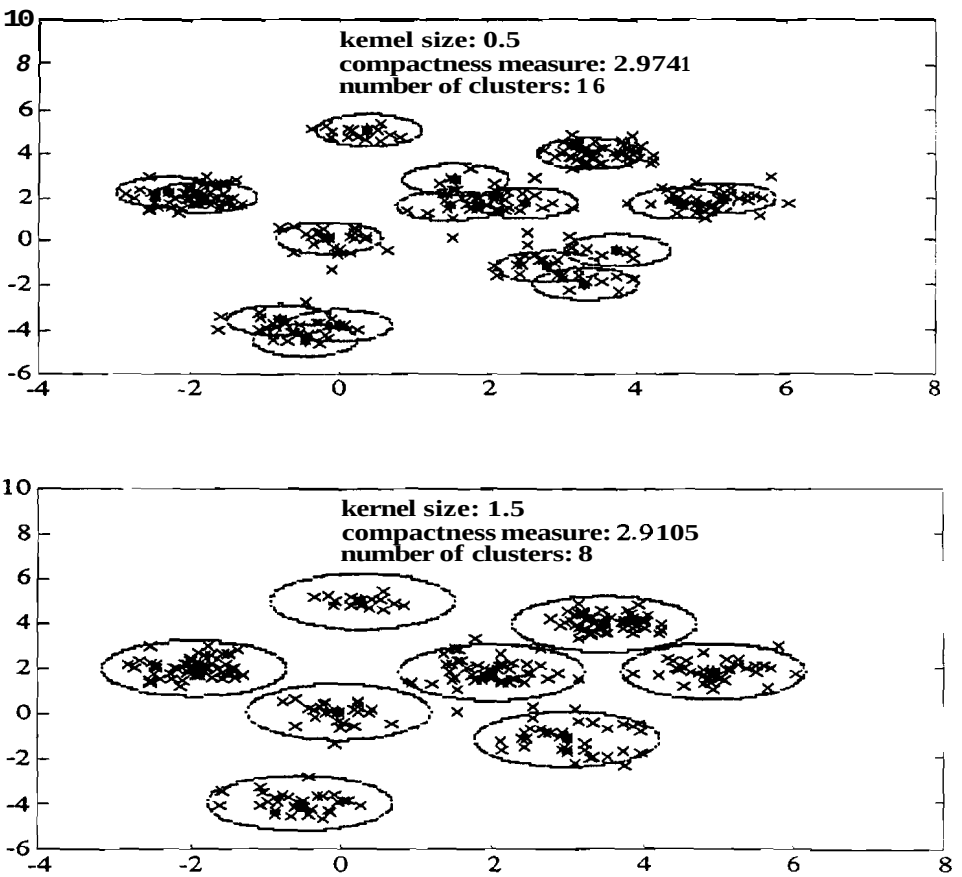


Fig. 5.2 - Number of clusters and correspondent partition goodness measure, (a) with the fuzzy compactness criterion, and (b) with the fuzzy density criterion

Both criteria leaded to the identification of the correct number of clusters and the kernel prototype converged to a very close neighborhood of the true center of mass of each sample. subclass. According to the compactness criterion (S), the best partition was obtained with kernel size of 1.5, and with size of 1.1 according to the average density. criterion **(AD).** Figure 5.3 shows some partitions and correspondent compactness criterion for different kernel sizes.



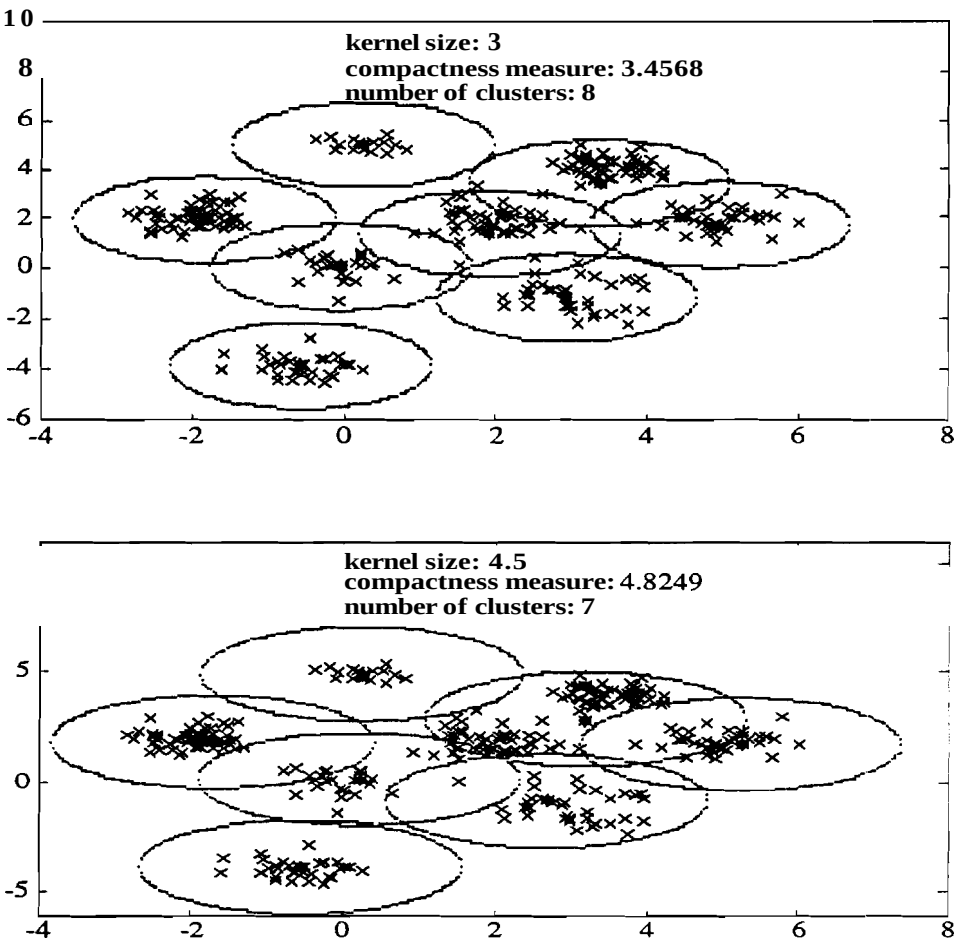


Fig. 5.3 - Partitions obtained with different sizes of the kernel prototype

Because of the global fuzzy validation, small kernel sizes do not necessarily imply larger number of clusters being detected, but the resultant partition goodness is usually low due to the bad placement of the kernel centroids. The fuzzy entropy, normalized to the range 0 - 1, gives an idea about the ambiguity of the cluster boundaries. For the 1.5 kernel size partition showed in figure 5.3, the fuzzy entropy coefficients are .0074; .0304; .0412; .0012; .0211; .0105; .0209; and .0021, which indicates that the clusters are well separated.

Example 2 : Two-Dimensional Data Set (D2) - This case was considered to evaluate the algorithm performance over well separated distributions of different sizes, shapes, and orientations. Figure 5.4 shows the 470 samples, clustered in seven independent Gaussian distributions, used for this experiment.

For this experiment we used the same pool of kernel sizes used in example 1. The results matched the expectations, i.e. because of the fixed model (size, shape, and orientation) of the kernel prototype, those more alonged dismbutions were subdivided in two or more distinct clusters. Figures 5.5 and 5.6 show respectively, the partition goodness vs. kernel size and the best partition for both validity criteria.

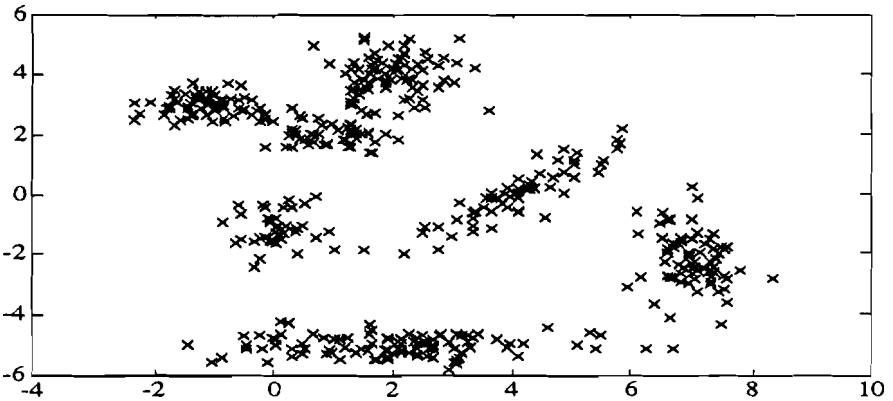


Fig. 5.4 - Two-Dimensional Data Set used for example 2: 470 points clustered in 7 independent Gaussian dismbutions of different sizes, shapes, orientations, and densities

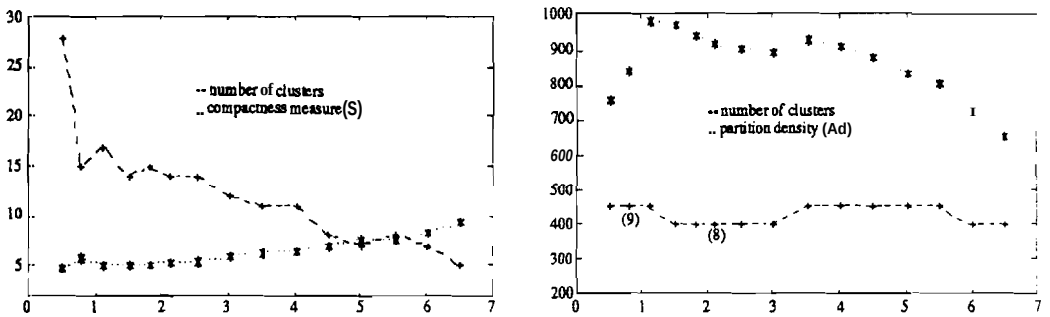
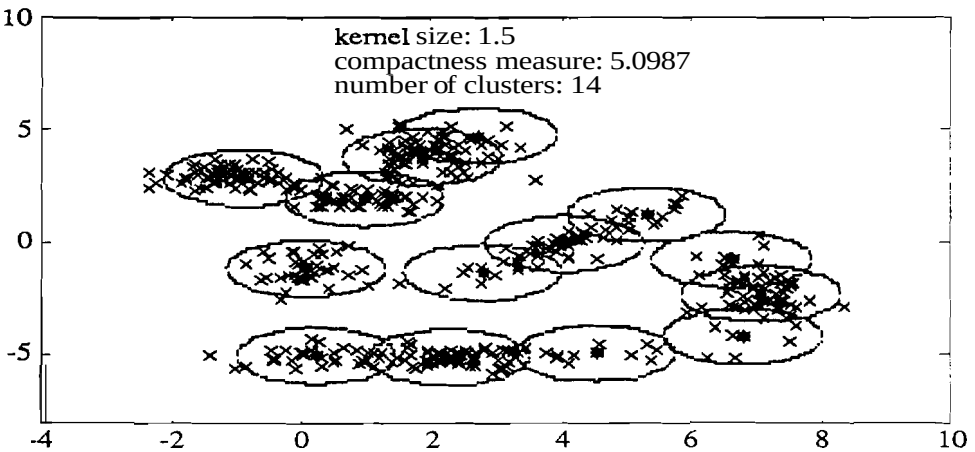


Fig 5.5 - Number of clusters and partition goodness vs kernel size, (a) compactness criterion, and (b) density criterion



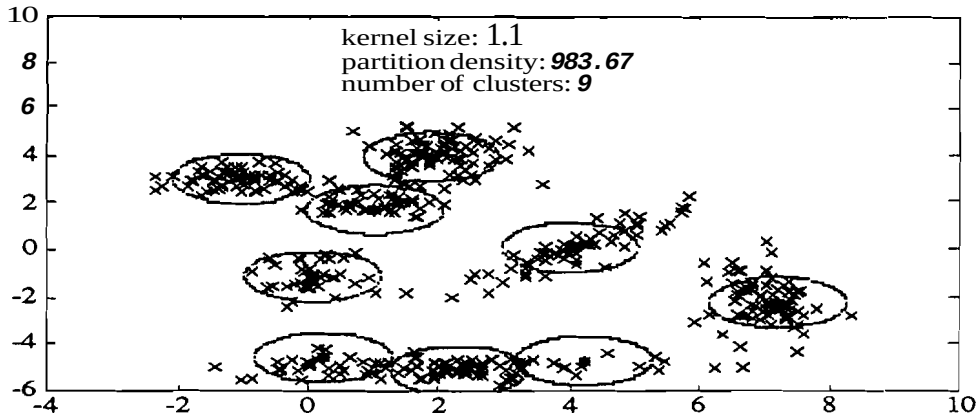


Fig. 5.6 - Partitions obtained with (a) the compactness criterion, (b) the density criterion

example 3: Three-Dimensional Data Set (D3) - This case was considered to evaluate the overhead created by a higher dimension. The same number of samples, 300, and the same number of distributions, 8, as in example 1 were used for this experiment. The elapsed time taken to compute the entire pool of kernel sizes and select the best partition was 299.84 seconds for the two-dimensional case (example 1) and 370.04 seconds for the three-dimensional case (both using compactness validation criterion), ancl253.39 seconds and 399.65 seconds respectively (using density criterion). The extra computational effort introduced by the additional dimension was about 50%. Both criteria identified the correct number of clusters and provided centers of mass very close to those of the sample subclasses. Figure 5.7 shows the number of clusters and partition goodness relation to the kernel sizes.

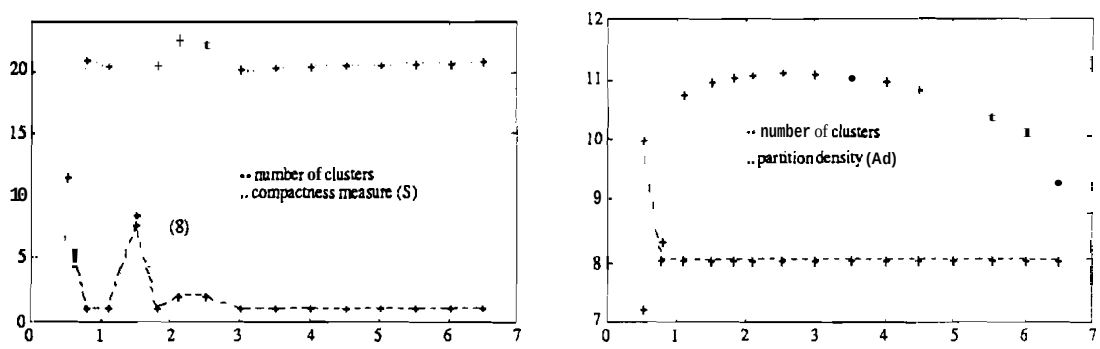


Fig 5.7 - Number of clusters and partition goodness vs kernel size, (a) compactness criterion, and (b) density criterion

example 4: Two-Dimensional Uniformly Random Data Set @4) - This case was considered to evaluate the tendency of the algorithm to impose rather than to find structure. It is true that 'lack of natural structure does not necessarily imply randomness, but the reverse does not follow the same rule, i.e. randomness necessarily implies lack of structure, and no partition should be expected from the algorithm. A pool of 15 kernel sizes was used, and as can be seen in figure 5.8 only the compactness validation criterion provided the expected answer. In all previous experiments both criteria performed well and very close to one another, but here the density criterion provided completely misleading result.

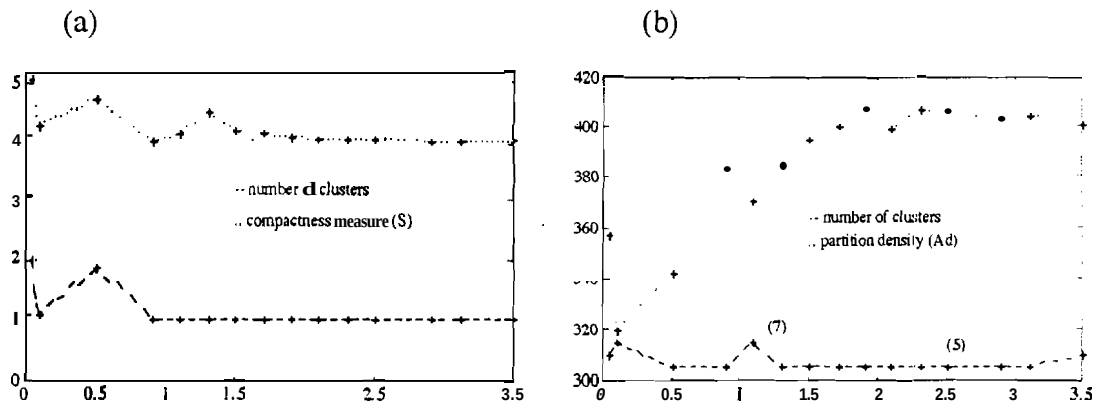


Fig. 5.8 - Partition for the uniformly random data set: (a) number of clusters and partition goodness vs kernel sizes with compactness criterion, and (b) with density criterion

V.3 - Real Data Tests

Example 1: Four-Dimensional Iris Data Set (D5) - Since Fisher (1956) this data set has been frequently used for clustering and pattern discrimination benchmarks. It consists of three apparently non-Gaussian classes represented by equal number of samples (50 each class) in a four-dimensional feature space. A pool of kernel sizes ranging from .04 to 3.38 was used, respectively 2% and 80% of the highest eigenvalue of the sample covariance matrix (fig 5.9). Four clusters, instead of three, were identified as the best partition.

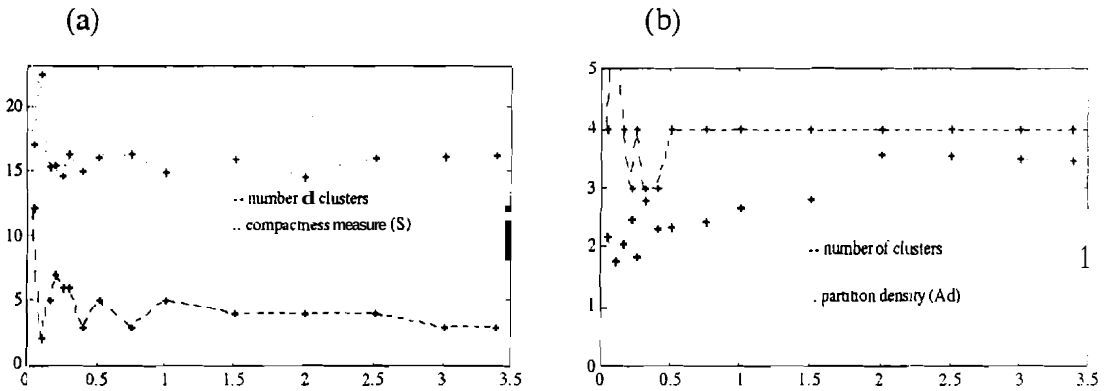


Fig. 5.9 - Number of clusters and partition goodness vs kernel size (a) compactness criterion, and (b) density criterion

The results are presented in terms of confusion matrices (fig 5.10) for the best three cluster cases, and in terms of an extended confusion matrix (fig. 5.11) for those cases where more than three resulting clusters are considered. Figure 5.12 presents the results for the K-means, the ISODATA, and the Fuzzy C-means algorithms.

-	-	50
12	38	-
47	3	-

50	-	-
-	37	13
-	1	49

Fig 5.10 - Best three clusters partition, (a) compactness criterion (kernel size of .4), (b) density criterion (kernel size of .3)

(a)

-	50	-	-
31	-	-	19
27	-	22	1

(b)

-	50	-	-
31	-	-	19
27	-	22	1

Fig. 5.11 - Best partition, (a) compactness criterion (kernel size of 2), and (b) density criterion (kernel size 2)

(a)

-	-	50
47	3	-
14	36	-

(b)

-	-	50
47	3	-
14	36	-

(c)

-	-	50
43	6	-
6	44	-

Fig 5.12 - Confusion matrices for the K-means, the ISODATA, and the FC-means

The fuzziness entropy for the above four cluster cases (fig. 5.13) reveals ambiguity for some: cluster boundaries. The same ambiguity can be also seen on the bottom-up hierarchical clustering (fig. 5.14).

	H1	H2	H3	H4	Average
3-cl compact	.3037	.3101	.0069	-	.2069
3-cl density	.3182	.3239	.0063	-	.2162
best compact	.3896	.1935	.0171	.2266	.2067
best density	.3906	.1949	.0171	.2265	.2073

Fig. 5.13 - Partition fuzziness entropy

example 2: Five-Dimensional Diabetes Data (D6) - Only three dimensions of the data were used: the glucose intolerance (d3); the insulin response to oral glucose (d4); and the insulin resistance (d5), and an scale transformation was applied to reduce the absolute values. The classes seem to be very diffused and very far from the Gaussian shape. As can be seen in figure 5.14, the proposed algorithm was not able to perform a good partition. Assuming the knowledge of the number of existing classes, the K-means, the FC-means, and the ISODATA algorithms provided partitions like is shown in figure 5.15.

(a)

76	-	-
35	-	-
4	17	12

(b)

73	3	-	-	-	-	-
9	18	-	-	5	7	-
-	1	12	9	5	1	5

Fig. 5.14 - Fuzzy locally partition, (a) with compactness cri tenon, and (b) with density criterion

(a)

-	73	3
-	10	26
21	-	12

(b)

-	73	3
-	17	19
26	3	4

(c)

-	73	3
-	9	27
26	-	7

Fig. 5.15 - Confusion matrices for the K-means, the ISODATA, and the FC-means

VI - RELAXING THE KERNEL'S SHAPE

The proposed algorithm, as described in section 3, is model-based and the energy kernel prototype is of the form of a hypersphere of fixed size. A natural relaxation to this assurnptiom is to allow the kernel prototype to be transformed into hyperellipsoid of variable size and orientation. Relaxation of shape, size, and orientation may be very useful and a powerful tool to deal with those strongly heterogeneous structures, where differences among classes' population and distribution are remarkable. However, the additional degrees of freedom considerably increase the computational complexity of the algorithm.

In this section two possible approaches are discussed. The first one is an extension to the basic procedure presented in section 3. The relaxation does not cover the size of the kernel prototype, being restricted to the shape and to the orientation only. The second approach can be seen as a postprocess procedure. It covers all degrees of freedom and is performed over the results provided by a previous partition scheme.

VI.1 - Relaxing Shape and Orientation

This procedure works in combination with the basic fuzzy locally sensitive procedure described in section 3. The shape and the orientation relaxations on the basic hypersperical shaped kernel prototype is accomplished by the introduction of an adaptation routine between steps 2 and 3. The major steps for the new algorithm becomes:

step 1 - initialization;

step 2 - select starting kernel prototype centroid;

step2a - adjust shape and orientation;

step 3 - evaluate the partition, if rejected stop.

step 4 - adjust pattern space;

step 5 - go back to step 1;

The adjustment procedure in step 2a preserves the hypervolume of the kernel prototype and alters both shape and orientation to better fit the population surrounding the centroid. The major steps of the routine are executed as follows:

step 1 - compute the fuzzy scatter matrix for the current (unmasked) set of samples with respect to the kernel centroid;

$$A_i = \frac{1}{\sum_{j=1}^{m} u_{ij}} \sum_{j=1}^{m} u_{ij} (X_j - V_i) (X_j - V_i)^t \qquad (44)$$

step 2 - perform the single value decomposition of the scatter matrix;

step 3 - adjust the eigenvalues to preserve the prototype hypervolume

$$\lambda_i^* = \frac{\lambda_i . b}{\left(\prod_{i=1}^{d} \lambda_i \right)^{1/d}} \tag{45}$$

where

λ_i is the scatter matrix eigenvalue
b is the kernel prototype size parameter
d is the dimension of the samples

step 4 - reconstruct the kernel prototype coordinates from the eigenvectors and adjusted eigenvalues

$$B_i = V.\Lambda^*.V^t \tag{46}$$

where

V is the eigenvectors matrix
A* is the adjusted eigenvalues matrix

The validity coefficient computed in step 3 is also adapted to incorporate the new shape.

$$S' = \sum_{i=1}^{c} \frac{\sum_{j=1}^{n} u_{ij}^2 (X_j - V_i)^t B_i^{-1} (X_j - V_i)}{n_i} \tag{47}$$

and

$$A_D = \lambda^{(c-1)} \sum_{i=1}^{c} S_i . e^{-\det(F_i)^{1/2}}; \qquad 0 << \lambda < 1 \tag{48}$$

The degree of membership (41) is now computed as follows:

$$u_{ij} = \frac{\exp(-[(X_j - V_i)^t B_i^{-1} (X_j - V_i)]^{1/2})}{\sum_{i=1}^{c} \exp(-[(X_j - V_i)^t B_i^{-1} (X_j - V_i)]^{1/2})} \tag{49}$$

This new algorithm was evaluated on the same data sets as in section V and, as expected, it improved the previously obtained partitions for those cases where the subclasses presented a distribution far from hipersphere. The results for the data sets D2, Iris, and Diabetes are illustrated in the figures 6.1 to 6.3. The best partition was found to be 8 clusters for the D2 data set, 2 clusters for the Iris data set, and 4 clusters for the Diabetes data set. The fixed size of the kernel prototype, at least for the D2 case, may be the reason for the procedure had not found the correct number of clusters. Confusion matrices and extended confusion matrices for the cases of 3, 4, and 2 cluster partition are presented for the Iris and for the Diabetes data sets.

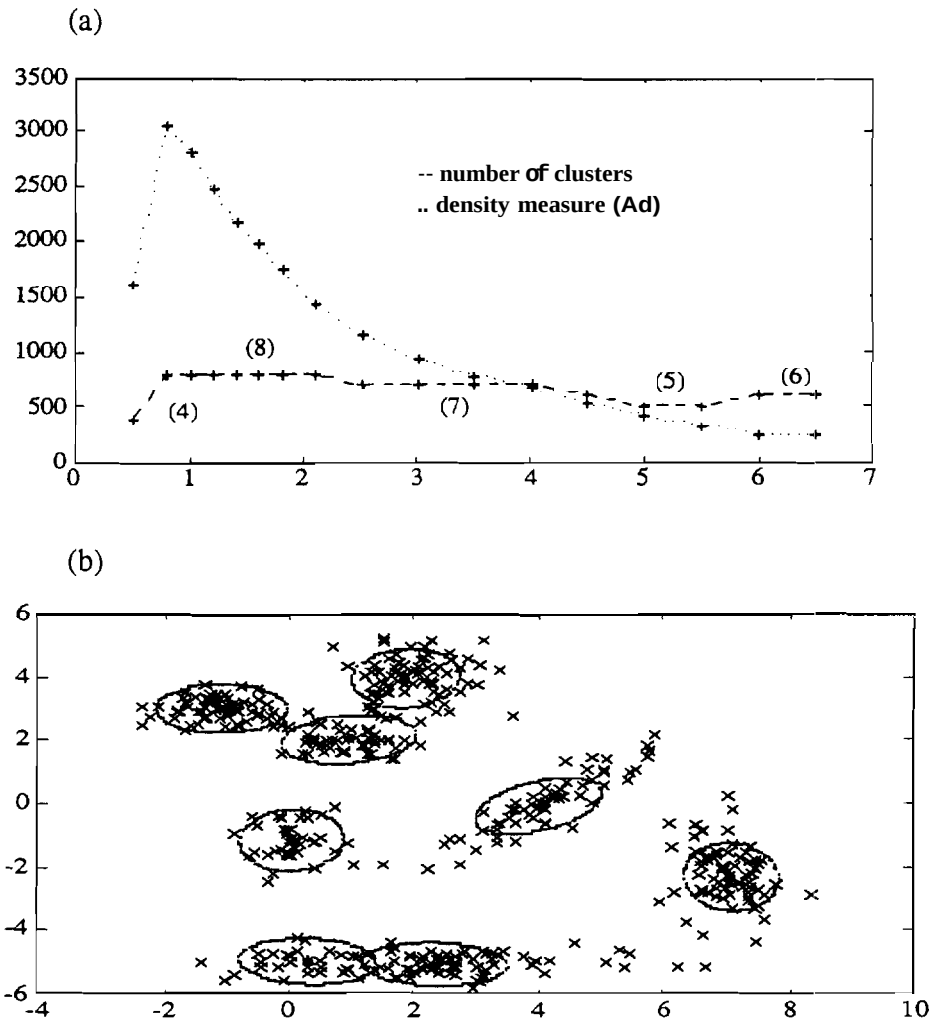
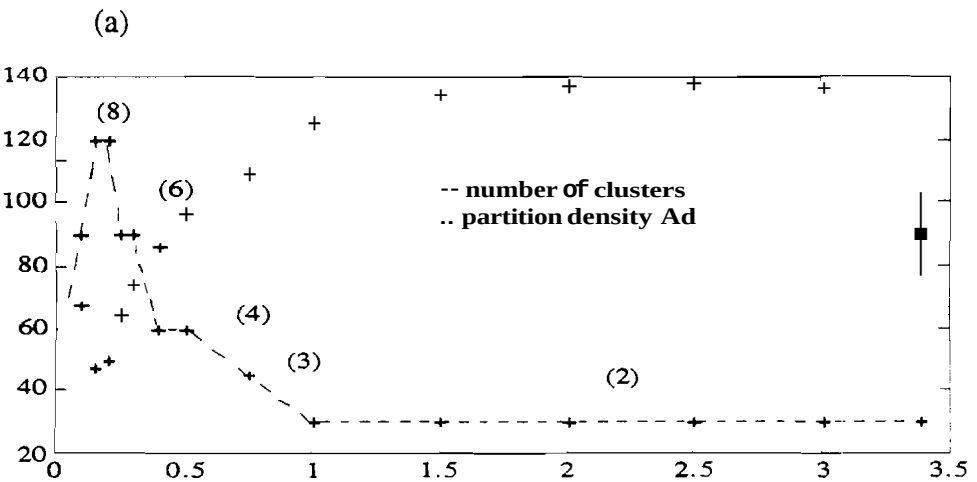


Fig. 6.1 - Two-Dimensional Data Set (example 2 section V), (a) number of clusters and partition performance vs kemel size, and (b) best partition obtained with kernel size of .8



(b)

50	-	-
-	50	-
-	20	30

(c)

50	-	-	-
-	26	-	24
-	16	33	1

(d)

50	-
-	50
-	50

Fig. 6.2 - Iris Data Set: (a) number of clusters and partition performance vs kernel size, (b) 3-cluster partition (kernel size of 0.8), (c) 4-cluster partition (kernel of 0.5), and (d) 2-clusters partition (kernel of 2.5)

(a)

76	-	-
35	1	-
3	18	12

(b)

74	-	2	-
14	-	22	-
1	13	6	13

Fig. 6.3 - Diabetes Data Set: (a) 3-cluster partition (kernel of ..), and (b) 4-cluster partition (kernel of ..)

VI.2 - Relaxing Shape, Orientation and Size

This procedure uses the kernel centroids and sizes provided by any of the previous partition schemes as starting configuration for a more complex optimization process. The optimal partition is searched by maximizing a two parameter compound cost function, where one of the parameters, here called attraction term, takes account of the within-cluster distances, and the other, called repulsion term, takes account of the between-cluster distances. The cost function is continuously differentiable which allows the use of the gradient descent or any other Newton's like technique for the optimization process.

$$J(V,B)= \sum_{i=1}^{c} A_i R_i \tag{50}$$

where

$V = [V_1 \ldots V_c]$	is an (d,c) mamx of kernel centroids
$B = [B_1 \ldots B_c]$	is an (d,(c.d)) mamx of kernel coordinates
c	is the number of clusters on the partition
A_i	is the attraction term
R_i	is the repulsion term

The attraction term is defined as:

$$A_i= \frac{1}{n} \sum_{j=1}^{n} g_i(j) \tag{50}$$

where

$$g_i(j)= \exp\left(\frac{-1}{2} (x_j-v_i)^t B_i^{-1} (x_j-v_i) \right) \tag{51}$$

The repulsion term is defined as:

$$R_i= \frac{c-1}{\sum_{j\neq i}^{c} \{1 + \exp[-a(w_i(j) - b)]\}} \tag{52}$$

where

a is a sigmoid sharpness coefficient; $(a \geq 1)$

b is an overlapping control coefficient; $(0 \leq b < 1)$

$$w_i(j) = (v_j - v_i)^t B_{ij}^{-1} (v_j - v_i) \tag{53}$$

and

$$B_{ij} = B_i + B_j \tag{54}$$

It can be noticed that both terms, A; and R;, can be related as the continuously differentiable version of the Fisher's multi-dimensional scatter matrices (tr S_w and tr S_B) respectively. As described in Appendix B, the first order necessary conditions for the critical points of the cost function leads to the stepwise evolution of the variables V and B as follows:

$$v_j(k+1) = v_j(k) + \alpha_j(k) \frac{\partial J(V,B)}{\partial v_j(k)} \tag{55}$$

and

$$b^j_{pq}(k+1) = b^j_{pq}(k) + \beta \frac{\partial J(V,B)}{\partial b^j_{pq}(k)}, \quad p = 1 \ldots d,\ q = p \ldots d \tag{56}$$

where

p relates to the pth row of B

q relates to the qth column of B

β is a constant gain coefficient

and

$\alpha_j(k) = \frac{\lambda}{g_j(k)}$, is the normalized gain coefficient for the jth subclass at time step k

The algorithm is not allowed to change the properties of symmetry and positive semi-definiteness of the scatter matrices

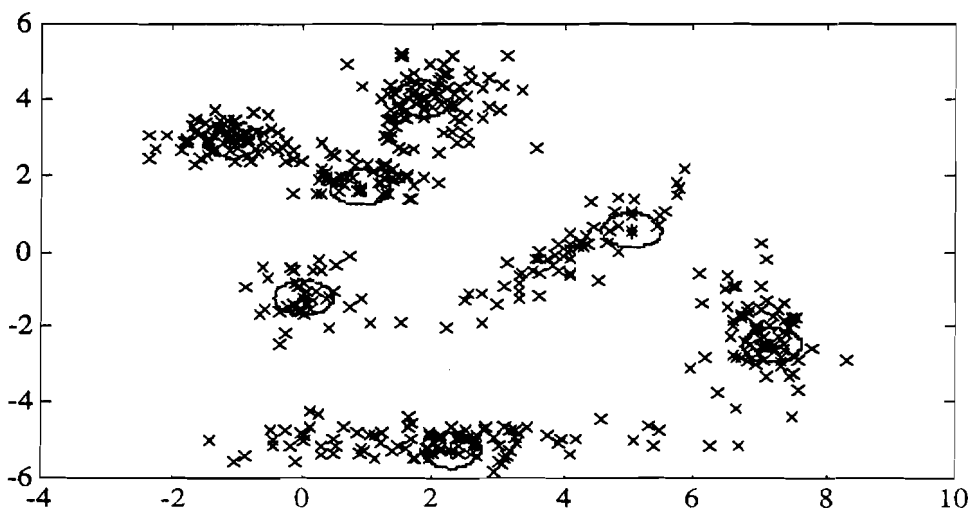


Fig. 6.4 - Two-Dimensional Data Set @2) - Simulated initial partition for the full relaxation prunning

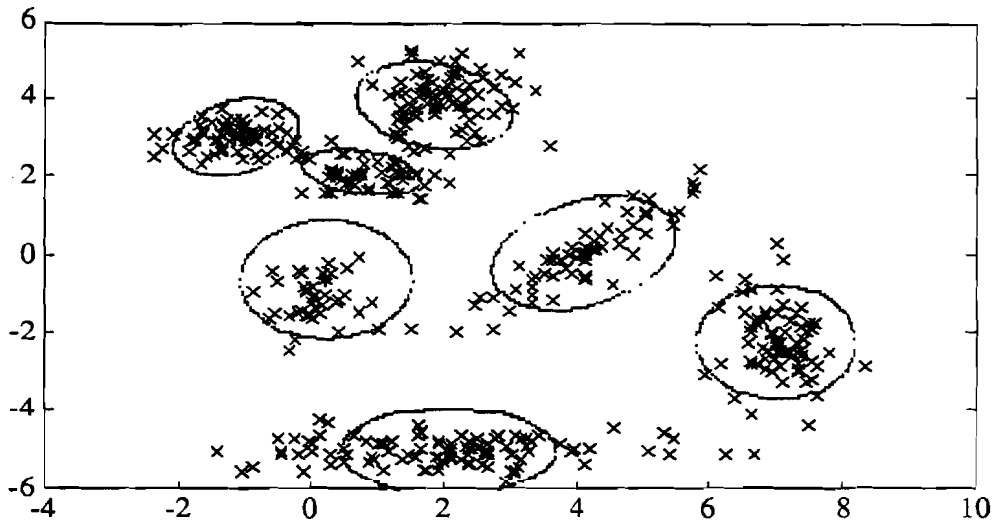


Fig. 6.5 - Result obtained after the relaxation of the kernel size, shape, and orientation

VII - CONCLUSIONS

In this paper a new mode separation procedure has been proposed for the unsupervised clustering problem. The procedure falls into the category of locally sensitive methods, it is model-based, it performs an iterative optimization of a cost criterion function, and it realizes a sequential partition of the pattern space. Extensions to the basic procedure, i.e. shape, size, and orientation of the subclasses model relaxation, have also been developed.

In contrast to some previous work reported in the literature, the proposed algorithm uses a gradient ascent evolution scheme to detect the relevant peaks of the underlying mixture density. The cost function is expressed by a simplified version of the exponential Parzen estimator. A global performance criterion is used as an attempt to automatically overcome the natural tendency of such approaches to over or underestimate the true number of distinct modes present in the mixture density. The combination of these two strategies improves the efficiency and the ability of the algorithm to identify those highly concentrated clusters, and to solve the validity problem.

The algorithm performance has been tested with a number of simulated as well as real data sets. The obtained results are encouraging, being comparable to those obtained through some: well known procedures like K-means, FC-means, and ISODATA with the advantage of not requiring initial parameters. In comparison to other locally sensitive methods, it appears to be superior in computational efficiency and comparable in performance. However, the computational demand may also grow fast for higher dimensions.

APPENDIX A

The data sets used in the simulations are described as follows:

a) Artificially Generated Sets

a.1) Set D1 - A total of 300 points from a 2-dimensional mixture of 8 distinct Gaussian distributions according to the rules: $N(\mu_i,\sigma_i^2)$ and $P(w_i)$

class #	Parameters values	$P(w_i)$	class #	Parameters values	$P(w_i)$
w1	([-2,2]',.15)	.2	w5	([2,2]',.2)	.15
w2	([-.5,4]',.25)	.1	w6	([3,-1]',.3)	.12
w3	([0,0]',.15)	.08	w7	([3.5,4]',.15)	.18
w4	([0,5]',.1)	.05	w8	([5,2]',.2)	.12

a.2) Set D2 - A total of 470 points from a 2-dimensional mixture of 7 distinct Gaussian distributions according to the rules: $N(\mu_i,\Sigma_i)$ and $P(w_i)$

class #	Parameters values	$P(w_i)$	class #	Parameters values	$P(w_i)$
w1	$\begin{pmatrix}1\\2\end{pmatrix},\begin{pmatrix}.2 & 0\\0 & .1\end{pmatrix}$	.1064	w5	$\begin{pmatrix}7\\-2\end{pmatrix},\begin{pmatrix}.2 & 0\\0 & .8\end{pmatrix}$	.1489
w2	$\begin{pmatrix}-1\\3\end{pmatrix},\begin{pmatrix}.31 & 0\\0 & .15\end{pmatrix}$	.1489	w6	$\begin{pmatrix}2\\-5\end{pmatrix},\begin{pmatrix}.2 & 0\\0 & .1\end{pmatrix}$	.2128
w3	$\begin{pmatrix}0\\-1\end{pmatrix},\begin{pmatrix}.2 & 0\\0 & .3\end{pmatrix}$	.0851	w7	$\begin{pmatrix}4\\0\end{pmatrix},\begin{pmatrix}1.0 & 0.9\\0.9 & 1.1\end{pmatrix}$	.1277
w4	$\begin{pmatrix}2\\4\end{pmatrix},\begin{pmatrix}.3 & 0\\0 & .35\end{pmatrix}$	.1702	-	-	-

a3) Set D3 - A total of 300 points from a 3-dimensional mixture of 8 distinct Gaussian distributions according to the rules: $N(\mu_i,\sigma_i^2)$ and $P(w_i)$

class #	Parameters values	P(wi)	class #	Parameters values	P(wi)
w1	([1,1,1]',.3)	.2	w5	([-3,-2,-2]',.3)	.15
w2	([4,2,0]',.5)	.1	w6	([-2,2,-3]',.4)	.12
w3	([3,-2,-1]',.2)	.08	w7	([2,0,5]',.5)	.18
w4	([-1,4,3]',.4)	.05	w8	([0,-5,0]',.4)	.12

a4) Set D4 - A total of 300 points of a 2-dimensional uniformly random distribution

b) Real Data Sets

b.1) Set D5 - Iris Data

It is the 4-dimensional Iris Data Set obtained from [Andr85]. This data set is also referred as the Fisher Iris Data, with measurements of the sepal length and width and petal length and width in centimeters of fitfty plants of each of three types of Iris: Iris Setosa, Iris Versicolor, and Iris Virginica.

b.2) Set D6 - Chemical and Overt Diabetes Data

It is a 5-dimensional data set also obtained from [Andrew]. This data set were used by Reaven and Miller (1979) to examine the relationship between chemical subclinical and oven: nonketotic diabetes in 145 non-obese adult subjects. The primary variables are glucose intolerance (d_3), insulin response to oral glucose (d_4), and insulin resistance (d_5). In addition, the relative weight (d_1) and the fasting plasma glucose (d_2) were measured for each person.

APPENDIX B

The f i t order necessary conditions for the unconstrained maximization problem

$$\max J(V,B)= \sum_{i=1}^{c} A_i R_i$$

$$\frac{\partial J(V,B)}{\partial V(k)} = \sum_{i=1}^{c}\left(\frac{\partial A_i}{\partial V}.B_i + A_i.\frac{\partial B_i}{\partial V}\right) = 0$$

and

$$\frac{\partial J(V,B)}{\partial B(k)} = \sum_{i=1}^{c}\left(\frac{\partial A_i}{\partial B}.B_i + A_i.\frac{\partial B_i}{\partial B}\right) = 0$$

where

$$\frac{\partial A_i}{\partial v_k} = \begin{cases} 0 & i \neq k \\ \sum_{j=1}^{n} B_k^{-1}(x_j - v_k) g_k(j) & i = k \end{cases}$$

and

$$\frac{\partial A_i}{\partial b_{pq}^k} = \begin{cases} 0 & i \neq k \\ \frac{1}{2}\sum_{j=1}^{n} (x_j - v_k)^t B_k^{-1}.B_k'.B_k^{-1} (x_j - v_k) g_k(j) & i = k \end{cases}$$

where

$$B_k' = \frac{\partial B_k}{\partial b_{pq}^k} \qquad \text{(see Appendix C)}$$

$$\frac{\partial B_i}{\partial v_k} = \begin{cases} \dfrac{-\sum\limits_{j\neq k}^{c} 2aB_{kj}^{1}\,(v_j - v_k)\,\exp(-a(w_k(j)-b))}{\left(\sum\limits_{j\neq k}^{c}(1+\exp(-a(w_k(j)-b)))\right)^2} & i = k \\[2ex] \dfrac{\left(-2aB_{ki}^{1}\,(v_i - v_k)\,\exp(-a(w_k(j)-b))\right)}{\left(\sum\limits_{j\neq i}^{c}(1+\exp(-a(w_i(j)-b)))\right)^2} & i \neq k \end{cases}$$

$$\frac{\partial B_i}{\partial b_{pq}^{k}} = \begin{cases} \dfrac{-\sum\limits_{j\neq k}^{c} a(v_j-v_k)^t(B_{kj}^{1}\,B_k^{'}\,B_{kj}^{1})(v_j-v_k)\,\exp(-a(w_k(j)-b))}{\left(\sum\limits_{j\neq k}^{c}(1+\exp(-a(w_k(j)-b)))\right)^2} & i = k \\[2ex] \dfrac{\left(-a(v_i-v_k)^t(B_{kj}^{1}\,B_k^{'}\,B_{ki}^{1})(v_i-v_k)\,\exp(-a(w_k(j)-b))\right)}{\left(\sum\limits_{j\neq i}^{c}(1+\exp(-a(w_i(j)-b)))\right)^2} & i \neq k \end{cases}$$

APPENDIX C

Matrix Operations.

Let A(x) denote a matrix whose elements are functions of the variable x, so the derivative of A(x) with respect to the free variable x is given by a matrix which elements are the derivative of each original element with respect to x.

$$\frac{\partial A(x)}{\partial x} = \begin{bmatrix} \frac{\partial a_{11}(x)}{\partial x} & \frac{\partial a_{12}(x)}{\partial x} \\ \\ \frac{\partial a_{21}(x)}{\partial x} & \frac{\partial a_{22}(x)}{\partial x} \end{bmatrix} = A'$$

The conventional calculus is also valid for matrices which provides the following derivative computations:

a) $\frac{\partial A^2(x)}{\partial x} = A'.A + A.A';$

b) $\frac{\partial A(x)B(x)}{\partial x} = A'.B + A.B';$

c) $\frac{\partial A^{-1}(x)}{\partial x} = -(A^{-1}.A'.A^{-1});$

d) $\frac{\partial A^{-2}(x)}{\partial x} = -(A^{-1}.A'.A^{-1}.A^{-1} + A^{-1}.A^{-1}.A'.A^{-1})$

VIII - REFERENCES

[Andr85] Andrew, D. and Herzberg, A., 1985, Data: *A* collection ofproblemsfor *many fields*for the student and research worker, Spring Verlag.

[Ball67] Ball, G. and Hall, D., 1967, "A clustering technique for summarizing multivariate data", Behav. Sci., vol. 12, pp. 153-155.

[Bere91] Béreau, M. and Dubuisson, B., 1991, "A fuzzy extended k-nearest neighbors rule", Fuzzy Sets and Systems, vol. 44, pp. 17-32.

[Bezd73] Bezdek, J., 1973, "Fuzzy mathematics in pattern classification", Ph.D. dissertation, Cornell University, Ithaca, NY.

[Bezd74] Bezdek, J., 1974, "Cluster Validity with fuzzy sets", J. Cybernet, vol. 3, pp. 58-73.

[Caco66] Cacoullos, T., 1966, "Estimation of a multivariate density", Ann. Inst. Math, vol. 18, pp. 179-190.

[Dida74] Diday, E., 1974, "Optimization in non-hierarchical clustering", Pattern Recognition, vol. 6, pp. 17-33.

[Dida76] Diday, E. and Simon, J., 1976, "Cluster analysis", Digital Pattern Recognition, ed. springer, Berlin.

[Dimi88] Dimitrescu, D., 1988, "Hierarchical pattern classification", Fuzzy Sets and Systems, vol. 28, pp. 145-162.

[Dube78] Dubes, R. and Jain, A., 1978, "Models and methods in cluster validity", Proc. IEEE Conf. on Pattern Recognition and Image Proc., Chicago, pp. 148-155.

[Dube79] Dubes, R. and Jain, A., 1979, "Validity studies in clustering methdologies", Pattern Recognition, vol. 11, pp. 235-254.

[Duda73] Duda, Richard O. and Hart, Peter E., 1973, *Pattern Classification* and Scene Analysis, John Willey & Sons Inc.

[Dunn74] Dunn, J., 1974, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", J. Cybern., vol. 3, No. 3, pp. 32-57.

[Gath89] Gath, I. and Geva, A., 1989, "Unsupervised optimal fuzzy clustering", IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 11, No. 7, pp. 773-781.

[Gitm70] Gitman, I. and Levine, M., 1970, "An algorithm for detecting unimodal fuzzy sets and its application as a clustering technique", IEEE Trans. Comput., vol. c-19, pp. 583-593.

[Kitt76] Kittler, J., 1976, "A locally sensitive method for cluster analysis", Pattern Recognition, vol. 8, pp. 23-33.

[Kitt88] Kittler, J., 1988, "Optimality of reassignement rules; in dynamic clustering", Pattern Recognition, voi. 21, No, 2, pp. 169-174.

[Murt92] Murtag, F., 1992, "Comments on parallel algorithms for hierarchical clustering and cluster validity", IEEE Trans on Pattern Analysis and Machine Inteliigence, vol. 14, No, 10, pp. 1056-1057.

[Rusp69] Ruspini, E., 1969, "A new approach to clustering", Inf. Control, vol. 15, pp. 22-32.

[Rusp70] Ruspini, E., 1970, "Numerical methods for fuzzy clustering", Information Sciences, vol. 2, pp 319-350.

[Parz62] Parzen, E., 1962, "On estimation of probability density fiunction and a mode", Ann. Math. Stat., vol. 33, pp. 1065-1076.

[Ther89] Therrien, C., 1989, *Decision Estimate and Classification*, John Willey & Sons, NY.

[Xie91] Xie, X. and Beni, G., 1991, "A validity measure for fuzzy clustering", IEEE Trans on Pattern Analysis and Machine Intelligence, vol. 13, No. 8, pp. 841-847.