Information Theoretic Clustering

Erhan Gokcay and Jose C. Principe, *Fellow*, *IEEE*

Abstract—Clustering is one of the important topics in pattern recognition. Since only the structure of the data dictates the grouping (unsupervised learning), information theory is an obvious criteria to establish the clustering rule. This paper describes a novel valley seeking clustering algorithm using an information theoretic measure to estimate the cost of partitioning the data set. The information theoretic criteria developed here evolved from a Renyi's entropy estimator that was proposed recently and has been successfully applied to other machine learning applications. An improved version of the k-change algorithm is used in optimization because of the stepwise nature of the cost function and existence of local minima. Even when applied to nonlinearly separable data, the new algorithm performs well, and was able to find nonlinear boundaries between clusters. The algorithm is also applied to the segmentation of magnetic resonance imaging data (MRI) with very promising results.

Index Terms—Information theory, clustering, MRI segmentation, entropy, optimization.

1 Introduction

Clustering [20], [19], [36] is an unsupervised way of data grouping using a given measure of similarity. Clustering algorithms attempt to organize unlabeled feature vectors into clusters or "*natural groups*" such that samples within a cluster are "*more similar*" to each other than to samples belonging to different clusters. In clustering, there is neither information given about the underlying data structure nor is there a single similarity measure to differentiate all clusters. Hence, it should not come as a surprise that there is no unifying theory to uniquely describe clustering.

Clustering the input data is advantageous when input data labeling for classification requires human expertise which is normally very expensive. There are many applications of clustering with great practical significance such as speech recognition [28], [34], [42], handwritten character classification [35], [26], fault detection [30], and medical diagnosis [1], [5], [23], [24]. There are also many variations of clustering in the literature, so, here, we briefly summarize the basic algorithms.

There are two basic approaches to clustering, which we call *parametric* and *nonparametric*. In parametric clustering we assume a predefined distribution for the data set, and calculate the sufficient statistics [37], [27] which will describe the data set in a compact way. For example, for a normal distribution $N(M, \Sigma)$ the sufficient statistics are the sample mean $M = E\{X\}$ and the sample covariance matrix $\Sigma = E\{XX^T\}$, which will describe the distribution perfectly. Unfortunately, if the data set is not distributed according to our choice, then the statistics can be very misleading. Another approach uses a mixture of distributions to describe the data [32], [33], [11], [45]. We can approximate virtually any density function in this way, but estimating the parameters of a mixture is not a trivial operation. And the question of how to separate the data set into different clusters is still unanswered, since estimating the distribution does not tell us how to divide the data set into clusters.

The nonparametric approach to clustering divides the data set into groups of points which possess strong internal similarities [12], [13]. To measure the similarities we use a criterion function and seek the grouping that maximizes (or minimizes) the criterion. This kind of algorithms requires a cost function to evaluate how well the clustering fits to the data, and an algorithm to minimize the cost function. For a given clustering problem, the input data X is fixed. The clustering algorithm varies only by the sample assignment C, which means that the minimization algorithm will change only C. Because of the discrete and unordered nature of C, classical steepest descent search algorithms can not be applied easily.

So far, the majority of clustering metrics are based on a minimum variance criterion. For instance, merging and splitting, neighborhood dependent, hierarchical methods, and ART networks [12], [18], [6], [7]. Competitive networks [18], [31], [46], [39], [21] can be used in clustering procedures [12], [20] where they form Voronoi cells [16], [14] with a minimum variance flavor. Valley seeking clustering [13] is a different concept that exploits not the regions of high sample density but the regions of less data. In a sense, valley seeking clustering attempts to divide the data in a way similar to supervised classifiers, that is, by positioning discriminant functions in data space. Valley seeking algorithms provide nonlinear discriminants among the data clusters (unlike the previous methods). The problem in valley seeking clustering is the pulverization of the number of clusters, if there is a slight nonuniformity in one of them. Valley seeking clustering still uses a variance measure to split the data [13].

- *E. Gokcay is with the Computational NeuroBiology Lab, Salk Institute, 10010 N. Torrey Pines Rd., La Jolla, CA 92037-1099. E-mail: erhan@salk.edu.*
- *J.C. Principe is with the Computational NeuroEngineering Laboratory, NEB 451, Electrical and Computer Engineering Department, University of Florida, Gainesville, FL 32611. E-mail: principe@cnel.ufl.edu.*

Manuscript received 8 Aug. 2000; revised 19 Apr. 2001; accepted 11 May 2001.
Recommended for acceptance by I. Sethi.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 112671.

Information theory [47], [48] was used in clustering by Watanabe [50] and by many other researchers [8], [22], [49]. Watanabe used a coalescence model and a cohesion method to aggregate and shrink the data into desired clusters. The appeal of an information theoretic measure of similarity is to capture data structure beyond second order (variance) statistics. The major problem of clustering based on information theoretic measures has been the difficulty to evaluate the metric without imposing unrealistic assumptions about the data distributions. Here, we will develop a novel clustering algorithm based on a sample-by-sample estimator of Renyi's entropy that will avoid this shortcoming.

In Section 2, divergence measures based on information theory are summarized to provide a framework for our work. Derivation and analysis of the new clustering evaluation function (CEF) is done in Section 3. Section 3 also explains how to extend the CEF to multiple clusters. Section 4 describes the optimization algorithm developed for this particular problem with a new grouping method. The results on synthetic and real data are described in Section 5 followed by conclusions in Section 6.

2 Divergence Measures

The clustering problem has been formulated as a distance between two distributions [20], but most of the proposed measures are limited to second order statistics (i.e., covariance) [13]. This does not need to be the case since cross-entropy measures the separation between two distributions [29] and utilizes the full information contained in the data as specified by the probability density function (pdf). Cross-entropy is also called directed divergence [10], [17] since it is not symmetrical. Assume $D(p,q)$ is a measure for the distance between two distributions p and q. If $D(p,q)$ is not symmetric, then it can be made symmetric by introducing $D'(p,q) = D(p,q) + D(q,p)$. Under certain conditions, the minimization of directed divergence is equivalent to the maximization of the entropy [25]. The Kullback-Leibler's (K-L) cross-entropy measure is defined as [29]

$$D_{KL}(f,g) = f(x)\int \ln\left(\frac{f(x)}{g(x)}\right)dx, \tag{1}$$

where f(x) and g(x) are two probability density functions of the random variable x.

Another important measure of divergence was given by Bhattacharya [2]. The distance $D_B(f,g)$ is defined by

$$D_B(f,g) = -\ln\left(\int \sqrt{f(x)g(x)dx}\right). \tag{2}$$

$D_B(f,g)$ vanishes iff $f = g$ almost everywhere. The so-called Chernoff distance [9] or generalized Bhattacharya distance is a nonsymmetric measure defined by

$$D_C(f,g) = -\ln\left(\int [f(x)]^{1-s}[g(x)]^s dx\right) \quad 0 < s < 1. \tag{3}$$

Another important divergence measure is Renyi's measure [44] which is given as

$$D_R(f,g) = \frac{1}{\alpha-1}\ln\left(\int [f(x)]^{\alpha}[g(x)]^{1-\alpha}dx\right) \quad \alpha \neq 1 \quad \alpha > 0 \tag{4}$$

Note that the Bhattacharya distance corresponds to $s = \frac{1}{2}$ in (4) and the generalized Bhattacharya distance corresponds to $\alpha = 1 - s$. All these measures involve the product or the ratio of the data pdfs and the log function. They will be compared performance wise in Section 3, but the true bottleneck for their application is the difficulty of estimating the functional forms (1) to (4) nonparametrically or without making too restrictive assumptions about the data distributions (such as the Gaussian assumption).

3 The Clustering Evaluation Function

Entropy measures uncertainty about a stochastic event, which is a good starting point to create a cost function for clustering. In fact, when we assign a sample to one of the different data clusters we incur an entropy cost. Minimizing this incremental entropy cost could be used as an evaluation function for clustering. Unfortunately, we were unable to obtain reasonable results with this approach. Another alternative, is to exploit the boundaries among the data clusters as in valley seeking algorithms. Samples should be clustered when there is natural boundaries between them, which can be measured as divergence or cross-entropy between the clustered subsets of the data. A sample should be assigned to a cluster when the distance to the other cluster is maximized. The most utilized divergence measure is the Kullback-Leibler divergence [12], [20] which is based on Shannon's entropy. However, the problem is how to estimate the K-L divergence directly from data in a nonparametric fashion as required by pattern recognition and machine learning applications [4], [13].

The Hungarian mathematician Alfred Renyi [44] proposed in the 60s a new information measure

$$H_R(x) = \frac{1}{1-\alpha}\ln\int f^{\alpha}(x)dx \quad \alpha > 0 \quad \alpha \neq 1, \tag{5}$$

which became known as Renyi's entropy and provided the starting point for an easier nonparametric estimator for entropy [39]. In order to use (5) in the calculations, we need a way to estimate the probability density function. One of the most productive nonparametric methodologies is the Parzen Window Method [28]. For simplicity and to take full advantage of the properties of the multidimensional Gaussian function, the kernel used here is

$$G(x) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)|\Sigma|^{-1}(x-\mu)^T\right), \tag{6}$$

where Σ is the covariance matrix. For simplicity, we will assume that $\Sigma = \sigma^2 I$. For a data set $X(i) = \{x_1(1), \ldots x_i(n)\}^T | i = 1 \ldots N$ the probability density function can be estimated as

$$f_X(x) = \frac{1}{N}\sum_{i=1}^{N} G(x - x_i, \sigma^2). \tag{7}$$

Substituting (7) into (5) with $\alpha = 2$, we obtain

$$H_R(x) = -\ln\int f_X(x)^2 dx \qquad \alpha = 2. \tag{8}$$

We call (8) Renyi's quadratic entropy. The reason for using $\alpha = 2$ is the exact calculation of the integral in (8) directly from the samples (i.e., nonparametrically) as

$$\begin{aligned} H_R(x) &= -\ln \int \left(\frac{1}{N}\sum_{i=1}^{N} G(x - x_i, \sigma^2\right)\left(\frac{1}{N}\sum_{j=1}^{N} G(x - x_j, \sigma^2\right) dx \\ &= -\ln \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(x_i - x_j, 2\sigma^2). \end{aligned} \tag{9}$$

Because of the Gaussian kernels and the quadratic form of Renyi's formulation, the result does not need any numerical evaluation of integrals [51], unlike Shannon's entropy measure [47]. We call the quantity

$$V(x) = \sum_{i=1}^{N}\sum_{j=1}^{N} G(x_i - x_j, 2\sigma^2) \tag{10}$$

the *Information Potential* [41] since it is a positive decreasing function of the distance between samples x_i and x_j, similarly to the potential energy between physical particles. Since this potential energy is related to *information*, we call it the Information Potential. There are many applications of the information potential as described in [41]. Here, we extend this idea to clustering by calculating the information potential between samples of different subgroupings. This will form the basis for our clustering algorithm.

3.1 Derivation

Let's assume that we have two data subgroupings *p(x)* and *q(x)*. We would like to evaluate the information potential between these two subgroups and how it is affected when the subgroupings change. Let's consider the following clustering evaluation function (CEF)

$$CEF(p,q) = \frac{1}{N_1 N_2}\sum_{i=1}^{N_1}\sum_{j=1}^{N_2} G(x_i - x_j, 2\sigma^2) \tag{11}$$

with $x_i \in p(x)$ $x_j \in q(x)$, that is, each index is associated with one subgrouping. In order to be more explicit, we include a membership function in (11) which shows the distribution for each sample. The membership function $M_j(x_i)$ will be defined for the first distribution as $M_1(x_i) = 1$ iff $x_i \in p(x)$, and $M_1(x_i) = 0$ otherwise. Likewise for the second distribution, a similar function can be defined as $M_2(x_i) = 1$ iff $x_i \in q(x)$. Now, we can redefine the *CEF* function as

$$CEF(p,q) = \frac{1}{2N_1 N_2}\sum_{i=1}^{N}\sum_{j=1}^{N} M(x_i, x_j) G(x_i - x_j, 2\sigma^2), \tag{12}$$

where $M(x_i, x_j) = M_1(x_i)^* M_2(x_j)$. The value of the function $M(.)$ is equal to 1, when both samples are from different distributions and 0 when both samples come from the same distribution. For two distributions, the function $M(.)$ can also be rewritten as $M(x_i, x_j) = |M_1(x_i) - M_2(x_j)|$. This means that the summation is calculated only when the samples belong to different distributions. By changing $M(.)$, we are changing the subgroupings among the two clusters.

3.2 CEF as a Nonlinear Weighted Distance

One conventional way of measuring the distance between two clusters is the average distance between pairs of samples given by

$$D_{avg}(X_1, X_2) = \frac{1}{N_1 N_2}\sum_{x_i \in X_1}\sum_{x_j \in X_2} \| x_i - x_j \|. \tag{13}$$

This measure works well when the clusters are well separated and compact, but it will fail if the clusters are close to each other producing nonlinear boundaries. A better way of measuring the distance between clusters is to weight the distance between samples nonlinearly. The issue is to find a reasonable nonlinear weighting function. The information potential concept suggests that the weighting should be a localized symmetric window such as the Gaussian kernel. When we use the kernel function the average distance function becomes

$$DN_{avg}(X_1, X_2) = \frac{1}{N_1 N_2}\sum_{x_i \in X_1}\sum_{x_j \in X_2} G(x_i - x_j, 2\sigma^2). \tag{14}$$

which is exactly the CEF function that we derived before. The CEF has an additional parameter α, which controls the variance of the kernel. Creating a principled approach for selecting the kernel variance is a difficult problem, for which we do not have yet a solution, but our experience shows that appropriate values can be found for real data.

3.3 CEF as a Distance

Intuitively, we expect that CEF is measuring some type of distance between the subgroupings. Let's define the following distance measure

$$\begin{aligned} D_{CEFnorm}(p,q) &= -\ln\left(\frac{\int p(x)q(x)dx}{\sqrt{\int p^2(x)dx \int q^2(x)dx}}\right) \\ &= -\ln\left(\frac{CEF(p,q)}{\sqrt{\int p^2(x)dx \int q^2(x)dx}}\right). \end{aligned} \tag{15}$$

The logarithm is included to be consistent with the other divergence measures. Certain properties must be verified for $D_{CEFnorm}(p,q)$ to be a distance [25]. It is trivial to show positiveness of $D_{CEFnorm}(p,q)$. The second property which should be satisfied is $D_{CEFnorm}(p,q) = 0$ whenever $p(x) = q(x)$, which is also easily proven. The third property is the symmetry condition which is obvious.

The numerator of (15) is exactly the CEF since $CEF(p.q) = \int p(x)q(x)dx$. It is interesting to observe that the square of the argument of the log in (15) can be interpreted as the inverse of the Cauchy-Schwartz distance between p(x) and q(x). This shows that in fact the numerator of (15) controls the behavior of $D_{CEFnorm}(p,q)$. In optimization, the fact that the minimum distance is not zero is irrelevant, so we can write a pseudo distance just with the *CEF* as

$$D_{CEF}(p,q) = -\ln CEF(p,q). \tag{16}$$

Writing the *CEF* as the integral of the product of *p(x)* and *q(x)* also shows the analogy between (16) and the Bhattacharya distance of (2). Note that for optimization the

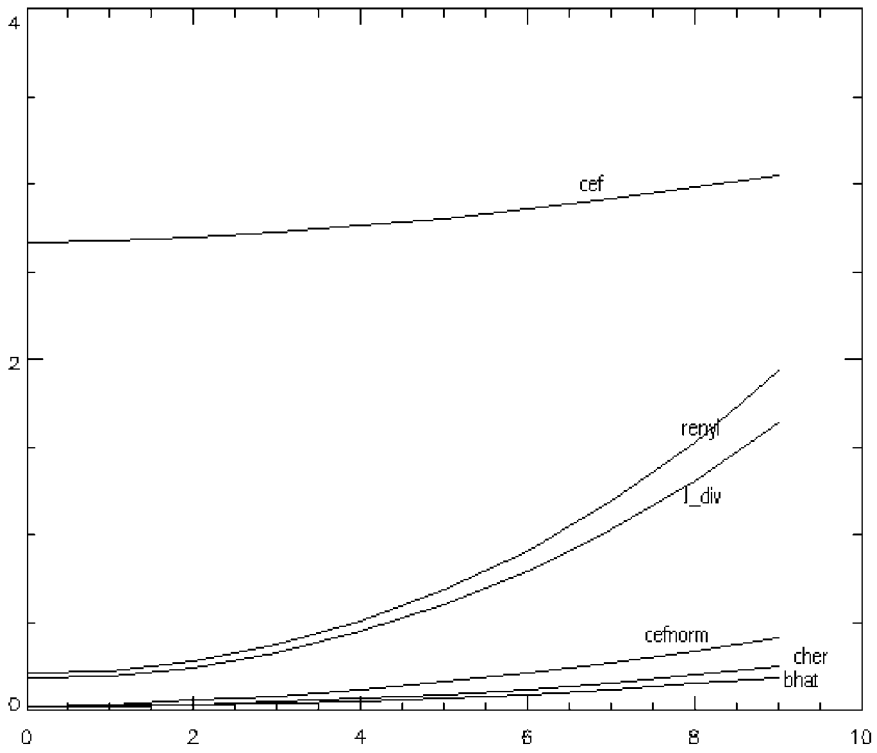


Fig. 1. Distance with regard to mean.

logarithm is not required since it is a monotonic function nor it's the scale given by the denominator of (15), so we can say that we are optimizing $D_{\mathrm{CEFnorm}}(p,q)$ when finding the extremes of $CEF(p,q)$. This independence from scale is one of the true advantages of optimization of cost functions (but hardly ever discussed) when compared to evaluation procedures.

We compared experimentally $D_{CEF}(p,q)$ with the other distance measures reviewed in Section 2. The comparison is done between two Gaussian distributions, when the mean of one of the Gaussian distributions is changed. The $\mathrm{D_{CEF}}$, $\mathrm{D_{CEFnorm}}$, and Bhattacharya are calculated with a kernel variance of 0.3, while Chernoff is calculated with a variance of 0.2, and Renyi's divergence is calculated with a variance of 1.1. In Fig. 1, we observe that although the minimum of $D_{CEF}(p,q)$ is not zero when the two pdf functions are equal, its behavior is consistent with the other measures, hence it can be used in optimization. Many more tests were conducted in comparing these distance measures and we concluded that the product measures (Bhattacharya and $\mathrm{D_{CEF}}$) are superior to the ratio measures (K-L and Renyi) in experiments with few data samples. The ratio measures are unstable due to the sampling errors in the estimation of $q(x)$ in (1), (3), and (4). We also would like to stress the fact that $D_{CEF}(p,q)$ is by far the measure that displays the smallest algorithmic complexity due to the efficient computation of the Information Potential. This makes it practical for machine learning algorithms. The calculation $D_{CEF}(p,q)$ requires $O(NxN)$ operations. Since the labels are changed for a small group of pixels, this calculation can be done with a complexity of $O(N)$ after the first exhaustive calculation. When we use numerical procedures to evaluate the integral, we need more iterations to achieve a predefined accuracy. Practically, the run time will be at least 5-10 times longer than the information potential calculation.

3.4 Multiple Clusters

The previous definition was given for two clusters. In this section, we will generalize the CEF function to more than two clusters. Since we want to measure the divergence between different clusters, the measure should include the divergence from one cluster to all the others. One way of achieving this is to estimate pairs of divergence measures between each possible pair of clusters. Let's assume that we have C clusters, and if the divergence measure is symmetric we need $\frac{N^*(N-1)}{2}$ pairs. The following extension of *CEF* will be used for more than two clusters:

$$CEF(x,s) = \frac{1}{2N_1N_2\ldots N_C}\sum_{i=1}\sum_{j=1} M(x_{ij},s)G(x_i - x_j, 2\sigma^2), \tag{17}$$

where $N_1 + N_2 + \ldots + N_C = N$.

Let's introduce a scoring function $s(.)$ for C clusters, where $s(.)$ is a C-bit long function defined as

$$s^k(x_i) = 1 \quad x_i \in C_k \tag{18}$$

and $s^k(.)$ is the kth bit of $s(.)$.

Using the scoring function, we can write $M(x_{ij},s) = s(x_i) \cup s(x_j)$. If both samples are in the same cluster, then $M(.)$ is 0, while if both samples are in different clusters, $M(.)$ is 1. Basically, the function is calculated between points of different clusters, not between points inside the same cluster. Since the label information is expressed in bits strings, this equation is valid for any number of clusters. We

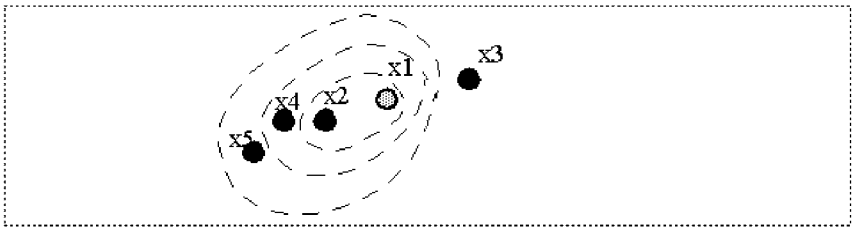


Fig. 2. Grouping method.

can use (17) to cluster the input data by minimizing the function with respect to $M(.)$. The stepwise nature of the function *CEF* with regard to $M(.)$ makes it impossible to use gradient based methods.

4 Optimization

4.1 Grouping Algorithm

We know the clustering algorithm should give the same label to samples that are close to each other in some distance metric. Instead of taking a fixed region around a sample for clustering as done in K nearest neighbors, we used the following grouping algorithm. Let's assume that the group size is KM. We will group samples from the same cluster starting from a large KM, which will be decreased during the iterations according to a predefined rule (exponential or linear). The grouping algorithm is explained in Fig. 2. Assume that the starting sample is x_1 and the subgroup size is 4. The first sample that is closest to x_1 is x_2, so x_2 is included in the subgroup. The next sample that is closest to the initial sample x_1 is x_3, but we are looking for a sample that is closest to any sample in the subgroup, which in this case is x_4. The next sample closest to the subgroup is still not x_3, but x_5, which is close to x_4. The resulting subgroup $\{x_1, x_2, x_4, x_5\}$ is a more connected group than just by selecting the four samples $\{x_1, x_2, x_3, x_4\}$ that are closest to the initial sample x_1.

The initial cluster labels come from the random initialization at the beginning of the algorithm. The grouping is done independently for each cluster label among its samples. The grouping starts with every pixel, and the groups that are equal are eliminated. The group size is initialized as $KM = \frac{N}{C}$, where N is the total number of data samples and C is the number of clusters.

In Fig. 3, the group is selected starting from the sample x_1 and the closest N samples is selected. The group shown in Fig. 3a is selected using N = 10. This group is actually the kNN of point x_1. The proposed grouping algorithm, on the other hand, creates the group shown in Fig. 3b. As can be seen from both figures, the proposed grouping is more sensitive to the structure of the data, when compared to the kNN method. The grouping algorithm is more calculation intensive than taking the samples around the initial sample, but experience showed that it helps the algorithm escape from local minima.

It should be mentioned that although the grouping algorithm catches the structures in the data better, this will not help us to cluster the data properly without the cost function. The grouping algorithm helps the algorithm to escape from the local minima, but we still need a cost function to decide how good the clustering is.

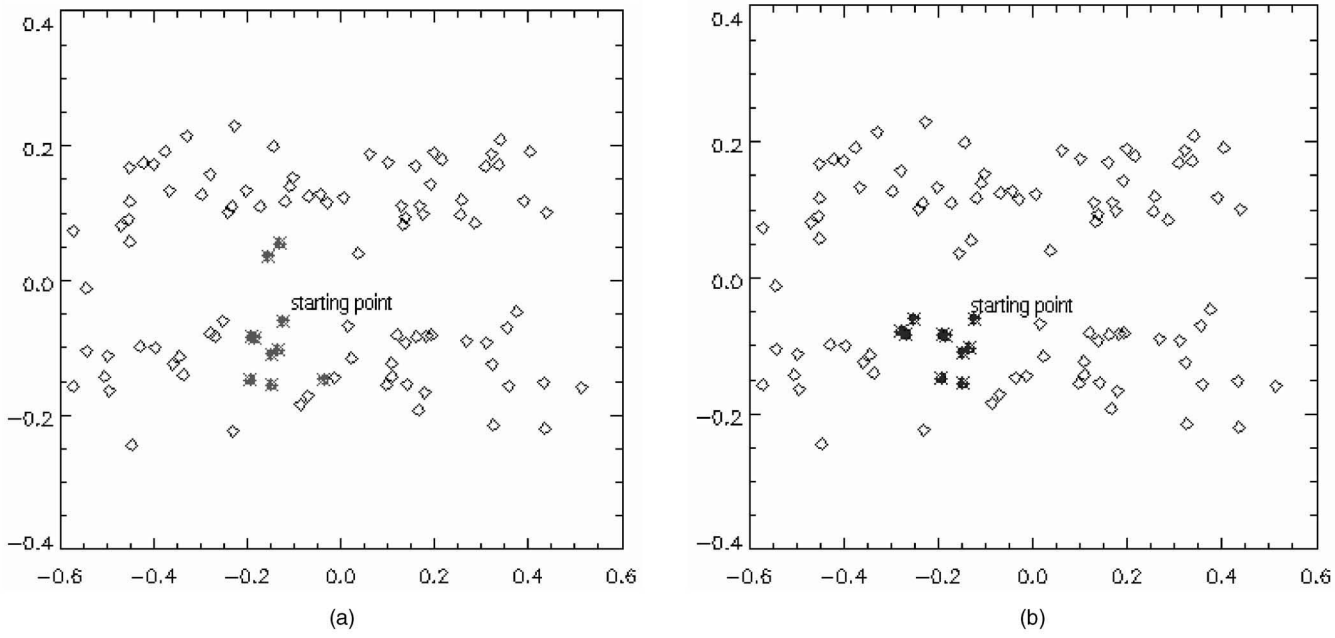


Fig. 3. Grouping using 10 nearest samples using kNN and new method.

```
INITIALIZE ( KM, var, ClusterLabels )
ClusterLabels = random()
WHILE ( KM >= 1 ) {
        GRP = CREATE_GROUPS ( InputData, KM, ClusterLabels )
        REPEAT UNTIL NOCHANGE {
        FOR i=0 to SIZE(GRP) {
                Change the clustering labels of GRP(i) and record the improvement if
                any
                FOR j=i+1 to SIZE(GRP) {
                        Change the clustering labels of GRP(i) and GRP(j)
                        and record the improvement if any
                }
        }
    }
    ; Decrease the group size. This can be linear or exponential
    KM = GENERATE_NEWGROUPSIZE(KM)
}
```

Fig. 4. Optimization algorithm.

4.2 Optimization Algorithm

We adapted and improved the k-change optimization algorithm [12], so that the algorithm can escape from local minima. The optimization starts by choosing an initial group size KM and randomly initializing the cluster labels. Next, the groups are formed as explained above and a new data set is created which is organized by groups instead of individual data points (similar to multiresolution) to save computation. We then select one group and check whether changing the label of this group reduces the cost function *CEF*. We should mention that the cost function is always computed between points of different clusters and it is not computed between points inside a cluster. This is controlled by the function $M(.)$. If the change reduces *CEF*, we will record this change as our new minimum cluster assignment. If the change does not reduce the value of *CEF*, we choose a second group from the list and change the cluster labels of that group in addition to the change of labels of the first group. We record this assignment if it reduces the minimum value of the cost function. In this algorithm, we allow a cluster assignment to be used even if it increases the cost function very similar to an annealing process, although the decision to increase the cost function is not random unlike in simulating annealing optimization, but it is done at every step. We repeat the calculations of a new cluster label until there is no improvement possible with the given group size KM. The next step is to reduce the group size KM and repeat the whole process, until the group size is 1, as shown in Fig. 4. When selecting the groups there are two choices. It is possible to choose the groups not to overlap, but our experience shows by letting them to overlap and letting one of the groups dominate the result, we can add more diversity to the tested cases which may let the algorithm escape from the local minimum. It is also possible to test the groups with nonoverlapping boundaries, overlapping with first group dominates and overlapping the second group dominates. This will increase the number of iterations, but it will increase the chance to escape from local minima.

Since there is a finite number of combinations tested and since the algorithm continues to work only when there is an improvement in the cost function, the algorithm is guaranteed to stop in a finite number of iterations. However, the global optimum may not be reached. The

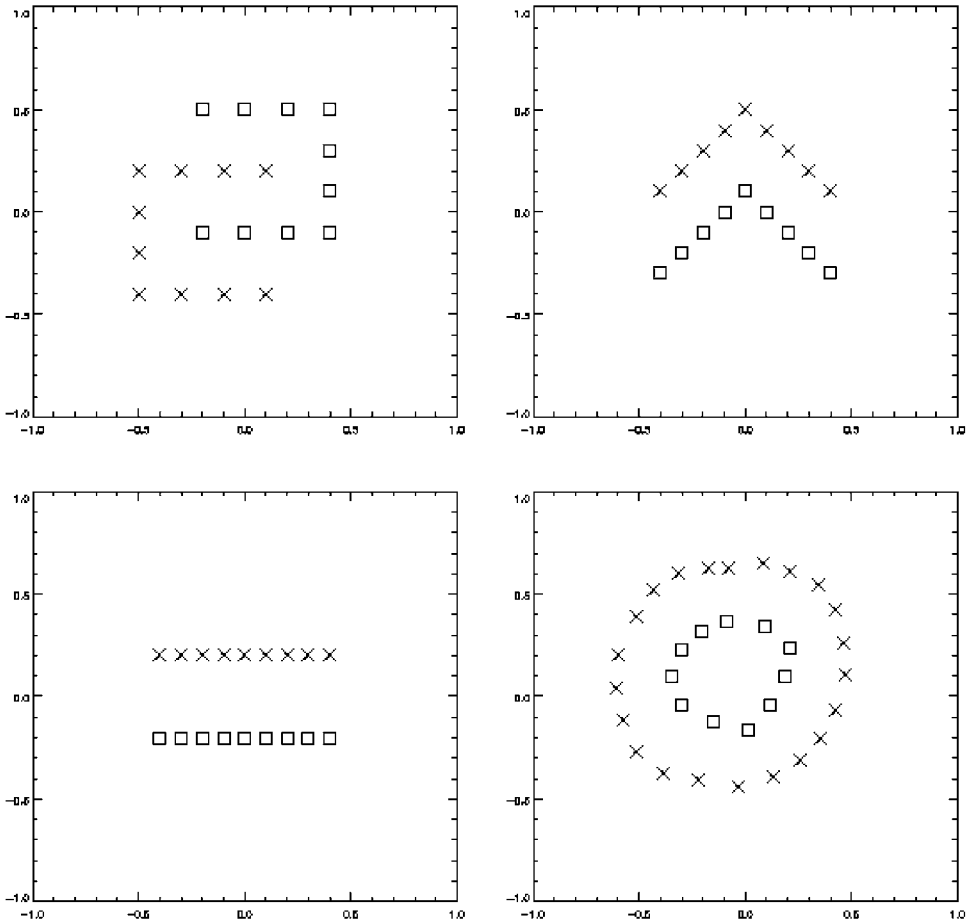


Fig. 5. Output of CEF using two clusters.

group size can at most be N, which is the number of samples, although practically it will be less than N since there will be groups containing the same samples. Therefore, the two inner loops have a complexity of $O(NxN)$. Practically, the number of iterations until no change recorded is not more than 5. The total complexity is therefore $O(KMxNxN)$. To calculate the divergence measure, we need an operation of complexity $O(NxN)$, but by omitting the previously calculated parts of the summation, the complexity of this calculation reduces to $O(N)$. The total complexity becomes $O(KMxNxNxN)$. If we calculate the divergence measure using other methods that require numerical integration, the calculation time is practically 5-10 times more than the current method and it is highly variable depending on the precision required and on the values of the parameters.

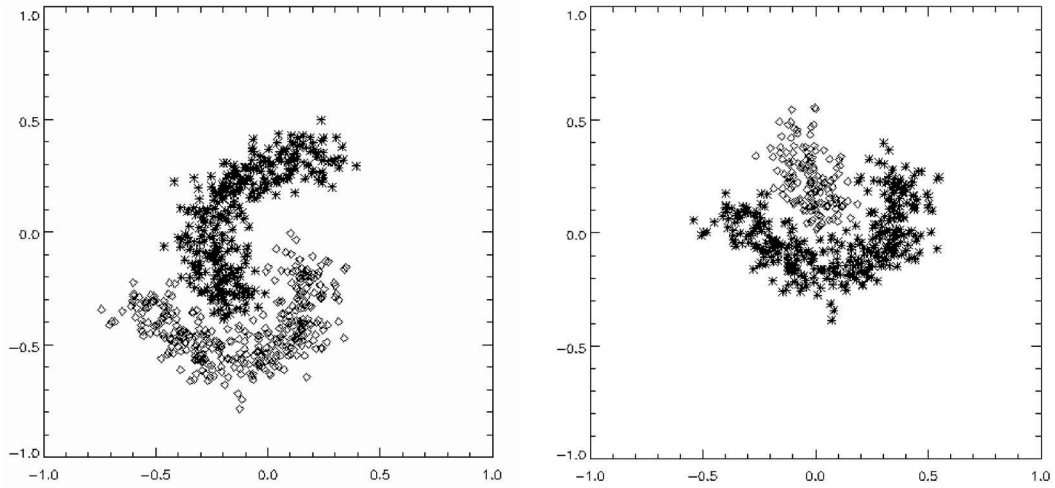
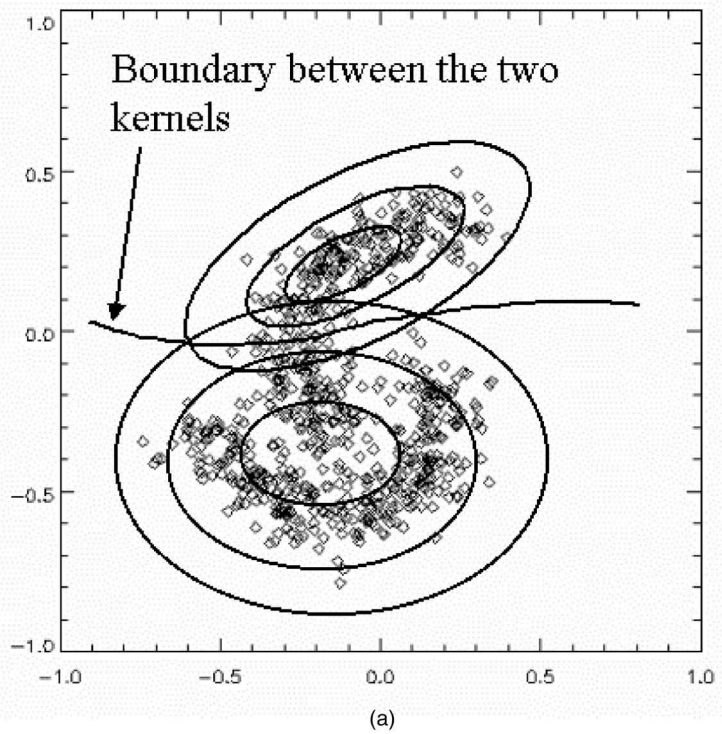
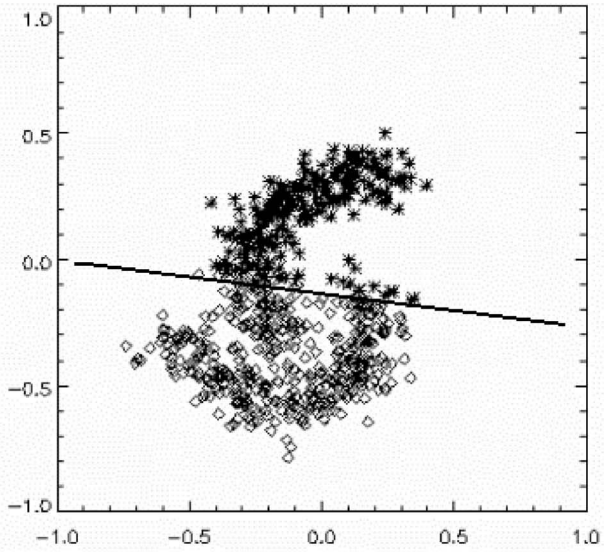


Fig. 6. Overlapping boundaries with even and uneven points in each cluster.



(a)



(b)

Fig. 7. Comparison with EM and K-means.

5 Experimental Results

5.1 Testing on Synthetic Data

Let us see how the clustering evaluation function (*CEF*) behaves and performs in the clustering of several different synthetic data sets. Fig. 5 shows four of the data sets used in [15]. The data sets are chosen to represent several different data distributions with a small number of samples, so that evaluation and verification of the results will be easy. The different clusters are shown using the symbols "square" and "star," although our clustering algorithm, of course, did not have access to the data labels. We designed these data sets to require nonlinear discriminant functions (MLPs or RBFs) in a supervised paradigm and that would display a natural valley between the clusters. For the data sets shown in Fig. 5, the minimum of the CEF initialized with two clustering centers provided perfect clustering in all cases (according to the labels we used to create the data). We used a variance ranging from 0.01 to 0.05, when the data is normalized between -1 and +1. We acknowledge that the variance is important to produce the assignments of Fig. 5, but we found that the clustering is not too sensitive to the actual value provided it is in the correct range, i.e., in the range where the pdf estimation shows the structure of the data. We should realize that none of the conventional

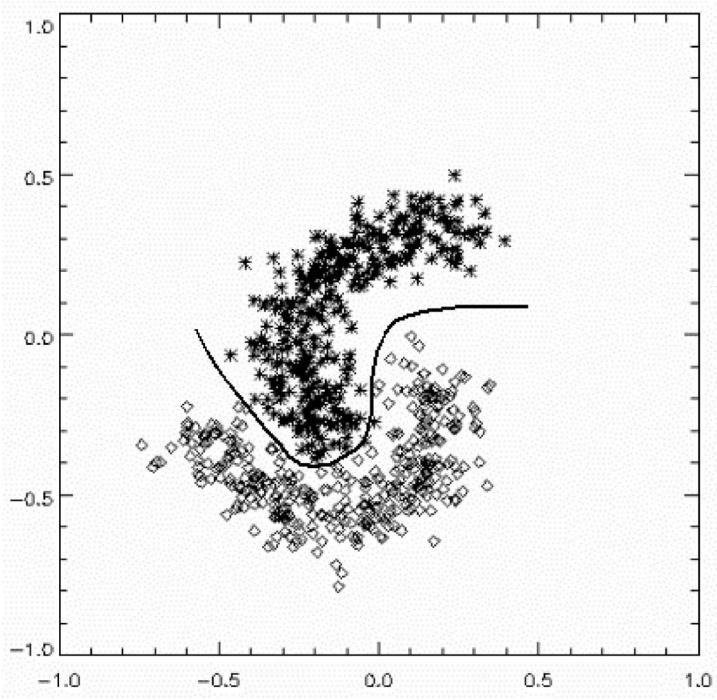


Fig. 8. Supervised classification.

algorithms for clustering would divide the data sets in the way the *CEF* did. The main difference is that conventional clustering uses a minimum distance metric which provides from a discrimination point of view a Voronoi tessellation (i.e., a division of the space in convex regions). Clustering using the *CEF* exploits valleys among clusters and, therefore, seems more appropriate to realistic data structures.

The *CEF* also works very well in more realistic data, as shown in Fig. 6, where the valleys are not as obvious. Fig. 6 shows the results of clustering (crosses one class, circles the other class) using the *CEF* algorithm initialized with two clustering centers. Against our expectation (see discussion in Section 3.3), *CEF* was not sensitive to the unequal size of the data clusters (see Fig. 6b). The variance of the Parzen window kernel is set in the range from 0.005 and 0.02 without basically changing the clustering. The convergence of the algorithm is normally smooth. In the early portion of the training, the number of samples that changed cluster was N/8-N/4 and it decreased to about 1-2 per iteration towards the end. This can lead to a natural stopping criterion. Samples that stand alone in data space (outliers) pose a problem to the algorithm, because they tend to define its own cluster.

5.2 Comparison

We compared our method with two well-known and widely used methods, i.e., the EM and the K-Means algorithms using the data in Fig. 6a. The results can be seen in Fig. 7. K-means could not separate the data as expected since the cost function has a minimum variance flavor. The EM algorithm can actually capture the data structure much better if we use more than two kernels, but how two separate these kernels into clusters remains a difficult question. We also trained a one hidden layer MLP and show its output in Fig. 8. The performance of our clustering algorithm is almost the same as the supervised classifier.

The second test of the CEF is done using the well-known iris data [12]. The Iris data consist of three different classes with a 4-dimensional feature vector, with 50 samples for each class. We should mention that a trained perceptron makes 19 mistakes, while a one hidden layer MLP classifier makes only three mistakes [39]. This clearly shows the requirement for a nonlinear discriminant function to solve this problem. Table 1 shows that the CEF makes 14 mistakes, better than the perceptron but still far from the optimum value achieved with the MLP. This results fuels shows that in many problems the CEF provides performance rivaling simple supervised classifiers.

5.3 Testing on MRI Data

The *CEF* algorithm was tested on a three class (white matter, gray matter, and cerebro spinal fluid (CSF)) identification of brain tissue from magnetic resonance imaging (MRI). This is known to be a challenging problem [3]. Although the topic of this paper is not feature extraction, the following feature set was found to

TABLE 1
Confusion Matrix for the Iris Data

	Class1	Class2	Class3
Class1	50	0	0
Class2	0	42	8
Class3	0	6	44

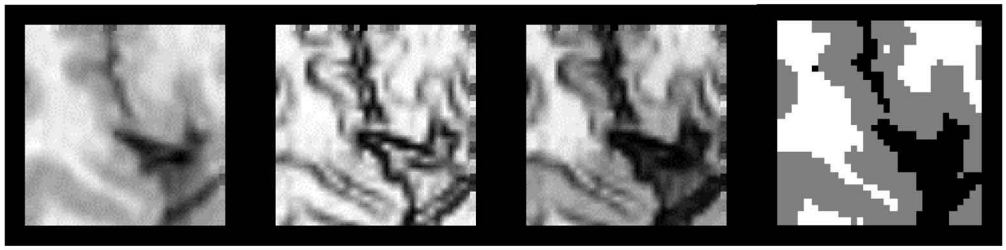


Fig. 9. Test image, information potential image, feature image, and segmented image.

be very useful during our research. The first feature is the brain image itself. The second feature is the information potential of 2x2 square masks around each pixel of the brain image calculated using equation (10). The information potential processed image enhanced edges among the different tissues. We propose the information potential processed image as a contrast enhancer (useful in edge detection), although properties of the detector should be investigated further.

A small test image is used here to illustrate the power of the information potential enhancement algorithm. We selected a small region of size 30x30 from a brain MRI image and calculated the local information potential explained above with a Gaussian kernels of variance 20.0.

When all the pixels of the 2x2 mask are in the same brain area, the entropy is low (information potential is high), and when the mask is over two different brain tissues, the entropy is larger (information potential is low). This can be seen in Fig. 9b, which is the processed version of Fig. 9a. Notice also that the actual gray values of the original image are lost in the information potential processed image, so we decided to multiply it pixel by pixel with the original image (Fig. 9c). The three brain areas (white matter, gray matter, and CSF) are much more distinguishable in Fig. 9c than in Fig. 9a. We can see the difference in the histograms (Fig. 10) of the original (Fig. 9a) and the enhanced image (Fig. 9c). The peaks of the histograms are further apart and we can clearly distinguish three peaks corresponding to the three types of tissue. This preprocessing facilitates the job of the clustering algorithm.

The 2D feature set (the original image and the information potential image) was then used as inputs to the *CEF* algorithm initialized with three clusters. To improve the class assignment, we used a 3-dim block of size 2x2x2 during the calculation of the information potential. The variance of the kernel used in the clustering is 5.0 and the algorithm reached a minimum point of $2.08x10^{-7}$ after approximately 50,000 iterations. Fig. 9d shows the results of the pixel assignments for this small image block. Through visual inspection performed by a specialist, we conclude that the classification is rather accurate.

The rest of the brain image was classified using the results obtained from this selected image block using the *CEF*. Each pixel is assigned to the closest class in terms of the *CEF* distance. This type of operation, where clustering is done on a small set of training data and the rest of the data is just classified without running the clustering algorithm (test data), is preferred because of the high computational cost of the minimization function. The issue is accuracy since we are generalizing the clustering obtained in the small training image. Visual inspection of classification of MRI images is a time consuming task and it is not very qualitative, so we sought a more quantitative assessment for our algorithm.

5.4 Validation

For validation, we adopted the fact that the percentage of the gray matter decreases with age and the percentage of white matter increases with age in known ways. The total cerebral volume is not changed significantly after age 5, but the percentage of white matter and gray matter change by about 1 percent per year [43]. Detecting this change with

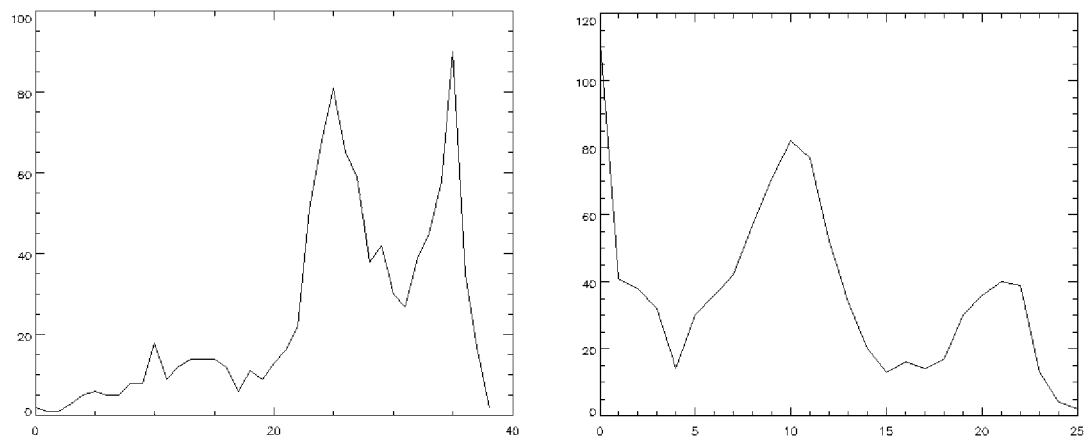


Fig. 10. Histogram of original and feature image.

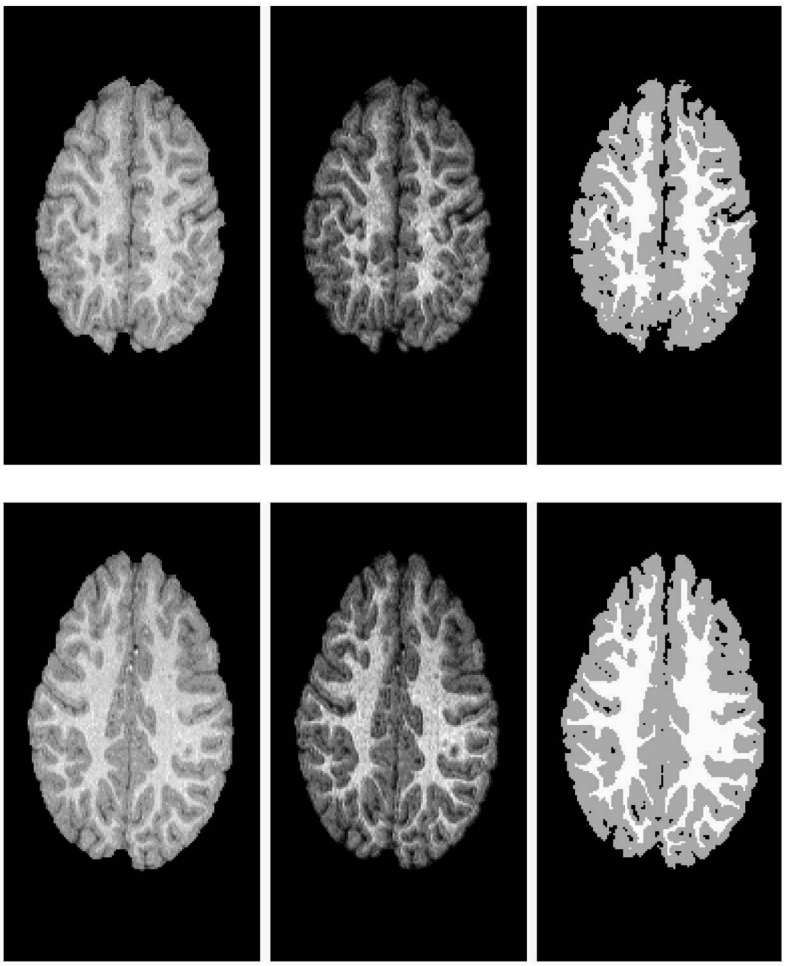


Fig. 11. Two Examples of MRI, enhanced MRI, and segmented MR images.

our segmentation algorithm should be a good measure of its accuracy, although the validation of the accuracy at the individual structures can not be done using this method.

Scan sequences were acquired in a 1.5T Siemens Magnetom using a quadrature head coil: a gradient echo volumetric acquisition "Turboflash" MP Rage sequence (TR=10 ms, TE=4 ms, FA=10o, 1 acquisition, 25 cm field of view, matrix=130x256) that was reconstructed into a gapless series of 128 1.25-mm thick images in the sagittal plane.

5.5 Segmentation

The *CEF* algorithm with three clusters is applied to MR images of two different children of ages 5 and 8 years at first scan and 7.5 years and 10.2 years at second scan. The preprocessing described in Section 5.2 was utilized here. Because of the large data set, the algorithm is applied to a small section of the brain (30x30x30) where there is an reasonable amount white and gray matter including CSF and afterwards the rest of the pixels are classified by measuring the *CEF* value between the unclassified pixels and the labeled pixels. The range of the features are between 0 and 400.0. The selected value for the kernel variance is set to 15.0. The group size in the optimization algorithm is selected as N/3, where N is the total number of points in the training data set. The group size is decreased by 2. The bigger the initial group size, the better the chance of escaping from local minima. Of course the price is increased computation time.

5.6 Segmentation Results

Fig. 11 shows a typical result. The left image is the MRI, the middle image is the information potential enhanced image, and the right image is the clustered image using the *CEF*. After all the images are segmented, the percentages of white matter, gray matter, and CSF with regard to the cerebral volume are calculated and are given in Table 2. The

TABLE 2
Percentages of Brain Matter

	white (%)	gray(%)	CSF(%)
5.5 years	31.45	57.97	10.56
7.5 years	33.37	55.73	10.89
8 years	36.39	54.21	9.39
10.2 years	37.60	50.87	11.51

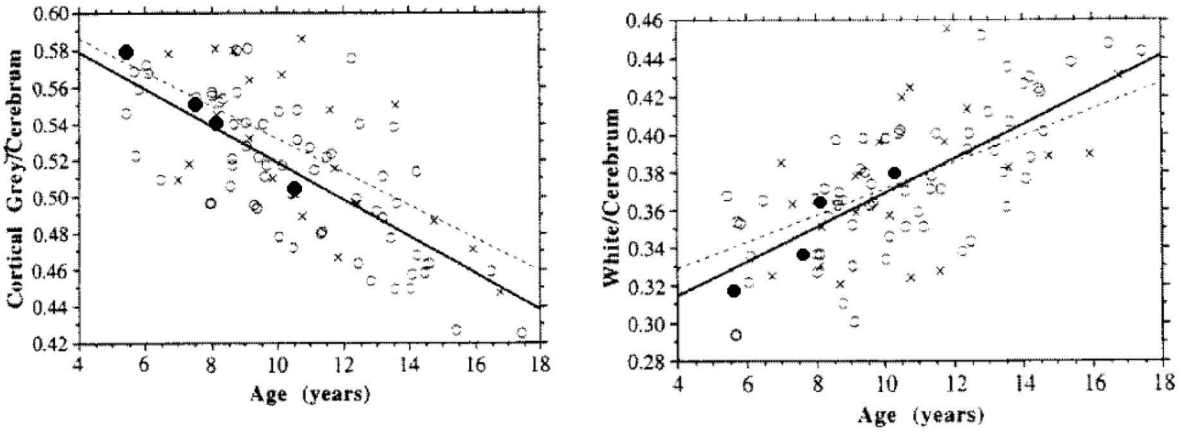


Fig. 12. Change of gray/white matter.

percentage volume of white matter to the cerebral volume increased with age, whereas the percentage volume of gray matter to the cerebral volume decreased with age as we expected. The results are in good agreement with the published results in [43], which are given in Fig. 12.[1]

Fig. 12 shows the measurements done on a group of males and females during several years (crosses and circles for boys and girls respectively), with the regression line in bold (straight line for boys, dashed line for girls) for both the gray and white matter tissues. Following the regression line is a good test for any segmentation algorithm since the change is small. We embedded our results in the same figures as big black dots. We see that our results fall right over the regression line. The comparison shows that the clustering algorithm provides a segmentation of the brain image that is compatible with the expected tissue change, which is indicative of a robust and possible accurate anatomical segmentation.

6 Conclusion

The goal of this research was to develop a better clustering algorithm because second order statistics are not sufficient to distinguish nonlinearly separable clusters. In order to achieve this goal, we developed an information theoretic cost function to measure cluster separability and which can be used as the basis of an automated algorithm for clustering. The cost function is developed using Renyi's quadratic entropy to evaluate a distance between probability density functions. The appeal of our proposed clustering evaluation function (*CEF*) is that it is computationally efficient to estimate directly from samples by using the Information Potential. The cost function is basically a valley seeking algorithm, where the distance is calculated from the data without requiring numerical integration. An optimization procedure which is efficient and simple is also developed to be used with the proposed cost function, although it is not limited to the method proposed in this paper. We also find out that certain distance measures are not suitable for use in our clustering algorithm due to inaccuracies in the estimation. In the second part of the paper, we applied the algorithm for the segmentation of MR images. The segmentation was found to be very successful and replicated results obtained by human experts.

The *CEF* algorithm can be improved further. It is possible to adapt the kernel shape and size dynamically according to the data. Instead of fixing the kernel shape, we can adapt it using the samples around the pixel as has been done in RBF networks. We can use the samples already found during our grouping algorithm, and fit a gaussian kernel to each group of pixels whenever the grouping is changed. This may improve the result of our clustering algorithm. The success of the algorithm depends on the estimation of the probability density function of each cluster generated during iterations, which is controlled to some degree by the variance of the kernel used. A more systematic way should be developed to adjust the variance of the kernel. This is a problem related to probability density estimation and it is more general than the clustering problem.

Running the algorithm with large data sets is still very time consuming. This is an area which needs improvement. A possibility is to create a network that is trained with the *CEF*. But, the real issue is how to find a training algorithm that can search efficiently the *CEF* cost function. Gradient descent is out of the question due to the discrete nature of the *CEF* cost function. It may be that some form of fuzzification may yield a smooth *CEF* functional.

Acknowledgments

The authors would like to thank Dr. Tiana Leonard for providing the MRI data and her technical expertise to evaluate the results and for her support. This work was partially funded by US National Science Foundation grants IBN-9634419 and ECS-9900394.

References

[1] M. Ashtari, J.L. Zito, B.I. Gold, J.A. Lieberman, M.T. Borenstein, and P.G. Herman, "Computerized Volume Measurement of Brain Structure," *Invest Radiology*, vol. 25, pp. 798-805, 1990.

[2] A. Bhattacharya, "On a Measures of Divergence between Two Statistical Populations Defined by their Probability Distributions," *Bull. Calculus Math. Soc.*, vol. 35, pp. 99-109, 1943.

1. Fig. 12 is reprinted with the permission of Oxford University Press.

[3] J.C. Bezdek, "Review of MRI Image Segmentation Techniques Using Pattern Recognition," *Medical Physics,* vol. 20, no. 4, pp. 1033-1048, 1993.
[4] C. Bishop, *Neural Networks for Pattern Recognition.* New York: Oxford University Press, 1995.
[5] B.R. Buchbinder, J.W. Belliveau, R.C. McKinstry, H.J. Aronen, and M.S. Kennedy, "Functional MR Imaging of Primary Brain Tumors with PET Correlation," *Soc. Magnetic Resonance in Medicine,* vol. 1, 1991.
[6] G.A. Carpenter and S. Grossberg, "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing,* vol. 37, pp. 54-115, 1987.
[7] G.A. Carpenter and S. Grossberg, "ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns," *Applied Optics,* vol. 26, pp. 4919-4930, 1987.
[8] M. Charikar, C. Chekuri, T. Feder, and R. Motwani, "Incremental Clustering and Dynamic Information Retrieval," *Proc. Ann. ACM Symp. Theory of Computing,* pp. 626-635, 1997.
[9] H. Chernoff, "A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on a Sum of Observations," *Ann. Math. Statistics,* vol. 23, pp. 493-507, 1952.
[10] T.M. Cover and J.A. Thomas, *Elements of Information Theory.* New York: Wiley, 1991.
[11] A.P. Demspter, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistic Soc.,* Series B, no. 39, pp. 1-38, 1977.
[12] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.
[13] K. Fugunaga, *Introduction to Statistical, Pattern Recognition.* New York: Academic Press, 1990.
[14] A. Gersho, "On the Structure of Vector Quantizers," *IEEE Trans. Information Theory,* vol. 28, pp. 157-166, 1982.
[15] E. Gokcay and J. Principe, "A New Clustering Evaluation Function Using Renyi's Information Potential," *IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* 2000.
[16] R.M. Gray, "Vector Quantization," *IEEE ASSP Magazine,* vol. 1, pp. 4-29, 1984.
[17] R.M. Gray, *Entropy and Information Theory.* New York: Springer-Verlag, 1990.
[18] S. Grossberg, "Adaptive Pattern Classification and Universal Recording: I. Parallel Development and Coding of Neural Feature Detectors," *Biological Cybernetics,* vol. 23, pp. 121-134, 1976.
[19] R.M. Haralick and L.G. Shapiro, "Image Segmentation Techniques," *Computer Vision, Graphics, and Image Processing,* vol. 29, pp. 100-132, 1985.
[20] J. Hartigan, *Clustering Algorithms.* New York: Wiley 1975.
[21] S. Haykin, *Neural Networks.* Piscataway, N.J.: IEEE Press, 1994.
[22] T. Hofmann and J.M. Buhmann, "Pairwise Data Clustering by Deterministic Annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 1-14, 1997.
[23] C.R. Jack, M.D. Bentley, C.K. Twomey, and A.R. Zinsmeister, "MR Imaging-Based Volume Measurement of the Hippocampal Formation and Anterior Temporal Lobe," *Radiology*, vol. 176, pp. 205-209, 1990.
[24] E.F. Jackson, P.A. Narayana, J.S. Wolinksy, and T.J. Doyle, "Accuracy and Reproducibility in Volumetric Analysis of Multiple Sclerosis Lesions," *J. Computer Assisted Tomogrophy,* vol. 17, pp. 200-205, 1993.
[25] J.N. Kapur and H.K. Kesavan, *Entropy Optimization Principles with Applications.* Boston, Mass.: Academic Press, 1992.
[26] N. Kato and Y. Nemoto, "Large Scale Hand-Written Character Recognition System Using Subspace Method," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics,* vol. 1, pp. 432-437, 1996.
[27] D. Kazakos and P. Kazakos Papantoni, *Detection and Estimation.* New York: Computer Science Press, 1990.
[28] K. Kido, J. Miwa, S. Makino, and Y. Niitsu, "Spoken Word Recognition System for Unlimited Speakers," *IEEE Int'l Conf. Acoust Speech Signal Process,* pp. 735- 738, 1978.
[29] S. Kullback, *Information Theory and Statistics,* New York: John Wiley, 1959.
[30] K. Le, Z. Huang, C.W. Moon, and A. Tzes, "Adaptive Thresholding—A Robust Fault Detection Approach," *Proc. IEEE Conf. Decision and Control,* vol. 5, pp. 4490-4495, 1997.
[31] R.P. Lippman, "Review of Neural Networks for Speech Recognition," *Neural Computation,* vol. 1, pp. 1-38, 1989.
[32] G.J. McLachlan and K.E. Basford, *Mixture Models: Inference and Applications to Clustering.* New York: Marcel Dekker, 1988.
[33] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions.* New York: John Wiley & Sons, 1996.
[34] N. Morgan and H. Franco, " Applications of Neural Networks to Speech Recognition," *IEEE Signal Processing Magazine,* vol. 14, no. 6, pp. 46-47, 1997.
[35] A. Navarro and C.R. Allen, "Adaptive Classifier Based on K-Means Clustering and Dynamic Programing," *Proc. Int'l Soc. for Optical Eng.,* vol. 3027, pp. 31-38, 1997.
[36] N.R. Pal and S.K. Pal, "A Review on Image Segmentation Techniques," *Pattern Recognition,* vol. 26, no. 9, pp. 1277-1294, 1993.
[37] A. Papoulis, *Probability, Random Variables and Stochastic Processes.* New York: McGraw-Hill, 1965.
[38] E. Parzen, "On Estimation of a Probability Density Function and Mode," *Annals of Math. Statistics,* vol. 33, pp. 1065-1076, 1962.
[39] J.C. Principe, N.R. Euliano, and W.C. Lefebvre, *Neural and Adaptive Systems: Fundamentals through Simulations.* New York: John Wiley & Sons, 2000.
[40] J.C. Principe, C. Wang, and D. Xu, "Speaker Verification and Identification Using Gamma Neural Networks," *IEEE Int'l Conf. Neural Networks,* vol. 4, pp. 2085-2088, 1997.
[41] J. Principe, D. Xu, and J. Fisher, "Information Theoretic Learning," *Unsupervised Adaptive Filtering,* S. Haykin, ed., New York: John Wiley & Sons, 2000.
[42] V. Radova and V. Psutka, "Approach to Speaker Identification Using Multiple Classifiers," *Speech Processing ICASSP, IEEE Int'l Conf. Acoustics, Speech and Signal Processing,* vol. 2, pp. 1135-1138, 1997.
[43] A. Reiss, M. Abrams, H. Singer, J. Ross, and M. Denckla, "Brain Development, Gender and IQ in Children: A Volumetric Imaging Study," *Brain,* vol. 119, pp. 1763-1774, 1996.
[44] A. Renyi, "On Measures of Entropy and Information," *Proc. Fourth Berkeley Symp. Math., Statistics, and Probability,* pp. 547-561, 1960.
[45] S.J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian Approaches to Gaussian Mixture Modelling," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 11, pp. 1133-1142, Nov. 1998.
[46] D.E. Rumelhart and D. Zipser, "Feature Discovery by Competitive Learning," *Cognitive Science,* vol. 9, pp. 75-112, 1985.
[47] C.E. Shannon, "A Mathematical Theory of Communications," *Bell System Technical J.,* vol. 27, pp. 370-423, 1948.
[48] C.E. Shannon and W. Weaver, *The Mathematical Theory of Communication.* Urbana, Ill.: The Univ. of Illinois Press, 1962.
[49] S. Shimoji and S. Lee, "Data Clustering with Entropical Scheduling," *IEEE Int'l Conf. Neural Networks,* pp. 2423-2428, 1994.
[50] S. Watanabe, *Pattern Recognition: Human and Mechanical.* New York: John Wiley & Sons, 1985.
[51] D. Xu and J. Principe, "Learning from Examples with Quadratic Mutual Information," *Neural Networks for Signal Processing—Proc. IEEE Workshop,* pp. 155-164, 1998.

Erhan Gokcay received the BS degree in electrical and electronic engineering in 1986. In 1986, he began graduate studies at the Middle East Technical University in Ankara, Turkey, graduating with the MS degree in electrical and electronic engineering in 1991, and continued the graduate studies for a PhD degree in computer engineering at the same university. In 1993, he was accepted to the University of Florida and continued his PhD studies in the Computer and Information Sciences and Engineering Department at the University of Florida, in Gainesville, Florida. He worked as an application programmer in ASELSAN Military Electronics and Telecommunications Company, in Ankara, Turkey, and as a system programmer in STFA Enercom Computer Center, in Ankara, Turkey, from 1986 to 1990. He worked as a technical manager at Tulip Computers in Ankara, Turkey, until 1991, and he was responsible from the installation and maintenance of the computer systems and training the technical support team. He worked as a network administrator and hardware supervisor in Bilkent University, in Ankara, Turkey, from 1991 to 1993, where he was responsible from the installation, maintenance, and support for the computer systems and campus-wide network. He worked as a system administrator in the CNEL and BME labs from 1993 until graduation at University of Florida, in Gainesville, Florida. Currently, he is working as a system manager and scientific programmer at Salk Institute, in San Diego, California.

Jose C. Principe is a professor of electrical and computer engineering and Biomedical Engineering at the University of Florida where he teaches advanced signal processing, machine learning, and artificial neural networks (ANNs) modeling. He is a Bell South Professor and the Founder and Director of the University of Florida Computational Neuro Engineering Laboratory (CNEL). His primary area of interest is processing of time varying signals with adaptive neural models. The CNEL Lab has been studying pattern recognition principles based on information theoretic criteria (entropy and mutual information), the use of ANNs for dynamical modeling, and speech and object recognition applications. He is collaborating with neuroscientists in a computational model of the olfactory cortex towards its analog VLSI implementation. He is a fellow of the IEEE. He is the chair of the technical committee on neural networks of the IEEE Signal Processing Society, a member of the Board of Governors of the International Neural Network Society, and editor-in-chief of the *IEEE Transactions on Biomedical Engineering*. He is a member of the advisory board of the University of Florida Brain Institute. Dr. Principe has more than 70 publications in refereed journals and book chapters, and 160 conference papers. He has directed 35 PhD dissertations and 45 master theses. He recently wrote an interactive electronic book entitled *Neural and Adaptive Systems: Fundamentals Through Simulation* published by John Wiley and Sons.