Pattern Recognition: Clustering

For BCSE Final Year
Jadavpur University

Pattern Classification by Distance Functions

The method of pattern classification by distance functions can be expected to yield practical and satisfactory results only when the pattern classes tend to have clustering properties

Contd..

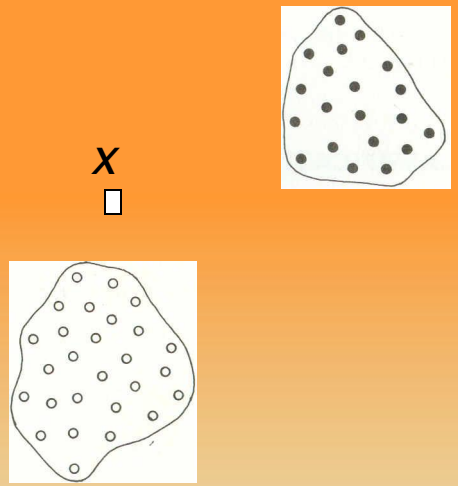


Figure 1. Patterns classifiable by proximity concept

Contd..

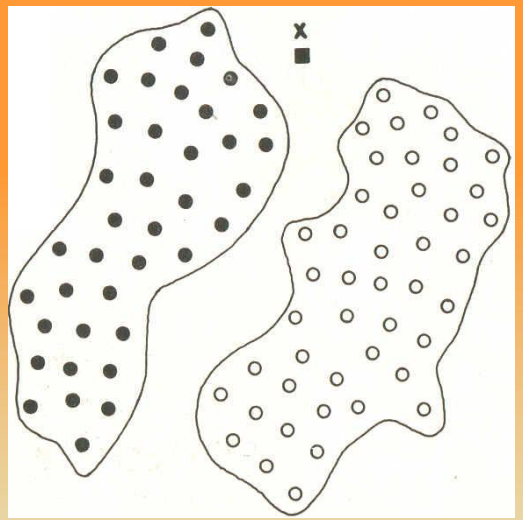


Figure 2. Patterns not easily classifiable by proximity concept

Contd..

Unlike Fig. 2, in Fig. 1, the patterns of each class tend to cluster tightly about a typical or representative pattern for the class. This occurs in case where pattern variability and other corruptive influences are well behaved. A typical example is the problem of reading bank checks by machine.

Contd..

Here the proximity of an unknown pattern to the patterns of a class will serve as a measure for its classification, so the method is called *minimum-distance pattern classification*. And the system which does this is called *minimum-distance classifiers*.

Single Prototypes

In this case, the patterns of each class tend to cluster tightly about a typical or representative pattern for that class. Under these conditions, minimum-distance classifiers can constitute a very effective approach to the classification problem.

Contd..

Consider M pattern classes and assume that these classes are representable by prototype patterns $Z_1, Z_2, \ldots, Z_M$. The Euclidean distance between an arbitrary pattern vector **x** and the *i*th prototype is given by,

$$D_i = \|X' - Z_i\| = \sqrt{(X - Z_i)'(X - Z_i)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

Contd..

A minimum-distance classifier computes the distance from a pattern **x** of unknown classification to the prototype of each class, and assigns the pattern to the class to which it is closest.

Contd..

In other words, **x** is assigned to class ω_i if $D_i < D_j$ for all $i \neq j$. Ties are resolved arbitrarily.

Equation (1) may be expressed as

$$D_i^2 = \|X - Z_i\|^2 = (X - Z_i)'(X - Z_i)$$

Contd..

$$= X'X - 2X'Z_i + Z_i' Z_i$$

$$= X'X - 2(X'Z_i - \frac{1}{2} Z_i' Z_i)$$

Contd..

Choosing the minimum D_i^2 is equivalent to choosing the minimum D_i since all distances are positive, since the term X^tX is independent of *i,* choosing the minimum D_t^2 is equivalent to choosing the maximum $(X^tZ_i - 1/2xZ_i^tZ_i)$
Consequently, we may define the decision functions..

Contd..

$$d_i(X) = X'Z_i - \frac{1}{2}Z_i' Z_i, \qquad i = 1,2,...,M \qquad .....(2)$$

where **x** is assigned to class ω_i , if $d_i(x) > d_j(x)$ for all $j \neq i$.

Observe that $d_i(X)$ is a linear decision function; that is, if $Z_{ij}, j = 1,2,...,n,$ are the components of Z_i, and we let..

Contd..

$$w_{ij} = z_{ij}, \quad j = 1,2,...,n \qquad w_{i.n+1} = -\frac{1}{2} Z_i' Z_i$$

and

$$X = \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ .x_n \\ 1 \end{pmatrix}$$

then we may express Eq. (2) in the familiar linear form

$$d_i(X) = W_i' X, \quad i = 1,2,...,M$$

where $W_i = (w_{i1}, w_{i2}, ..., w_{i.n+1})'$.

Contd..

The decision boundary of a two-class example in which each class is characterized by a single prototype is shown below.

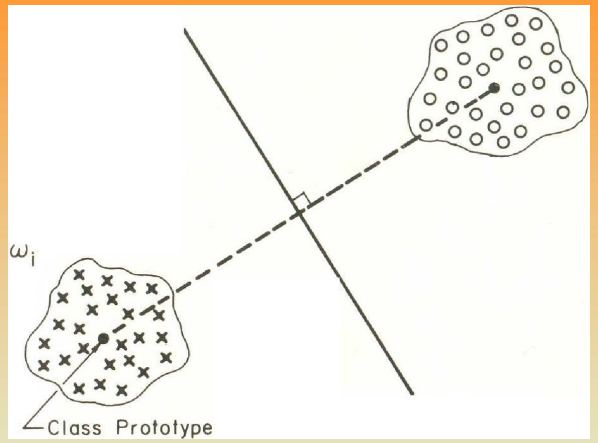


Figure 3. Decision boundary of two classes characterized by single prototype

Contd..

It can be shown that the linear decision surface separating every pair of prototype points z_i and z_j is the hyperplane which is the perpendicular bisector of the line segment joining the two points. Thus the minimum-distance-classifiers are a special case of linear classifiers.

Multiprototypes

Here each class is characterized by several prototypes, that is, each pattern of class ω_i tends to cluster about one of the prototypes $Z_i^1, Z_i^2, \ldots, Z_i^{N_i}$, where N_i is the number of prototypes in the i_{th} pattern class.

Let the distance function between an arbitrary pattern **x** and class ω_i be denoted by

$$D_i = \min \left\| X - Z_i^l \right\|, \quad l = 1,2,\ldots,N_i$$

Contd..

Following the development for single prototype, the decision functions

$$d_i(X) = \max\{(X'Z_i^l) - \frac{1}{2}(Z_i^l)'Z_i^l\}, \quad l = 1,2,...,N_i$$

where, as before, **x** is placed in class ω_i ,if $d_i(X) > d_j(X)$, for all $j \neq i$.

The decision boundaries for a two-class case in which each class contains two prototypes are shown in the next slide.

Contd..

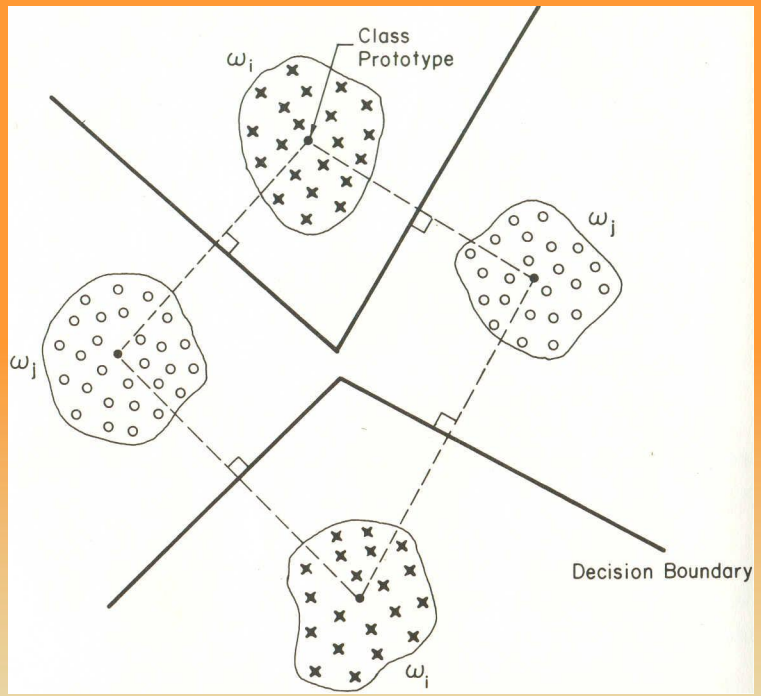


Figure 2. Piecewise-linear decision boundaries *for* two classes, each of which is characterized by two prototypes

Contd..

Note that the boundaries between classes ω_i ω_j and are piecewise linear. Since we could have defined this as a single-prototype, four-class problem, the sections of the boundaries are the perpendicular bisectors of the lines joining the prototypes of different classes.

Contd..

Although general iterative algorithms exist which can be used in the calculation of linear decision function parameters, unfortunately, no truly general algorithm is yet known for the piecewise-linear classifier decision function.

Cluster Seeking

It is evident that the ability to determine characteristic prototypes or cluster centers in a given set of data plays a central role in the design of pattern classifiers based on the minimum-distance concept. So we need to study various cluster-seeking methods.

Contd..

Note that the performance of a given algorithm is not only dependent on the type of data being analyzed, but is also strongly influenced by the chosen measure of pattern similarity

Measures of Similarity

To define a data cluster, it is necessary to first define a measure of similarity which will establish a rule for assigning patterns to the domain of a particular cluster center. Earlier we considered the Euclidean distance between two patterns x and z:

$$D = \|X - Z\|$$

as a measure of their similarity----the smaller the distance, the greater the similarity.

Contd..

There are, however, other meaningful distance measures which are sometimes useful. For example, the *Mahalanobis distance* from ***x*** to ***m.***

$$D = (X - m)' C^{-1} (X - m) \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

Contd..

It is a useful measure of similarity when statistical properties are being explicitly considered. Here **C** is the covariance matrix of a pattern population, **m** is the mean vector, and **x** represents a variable pattern.

Contd..

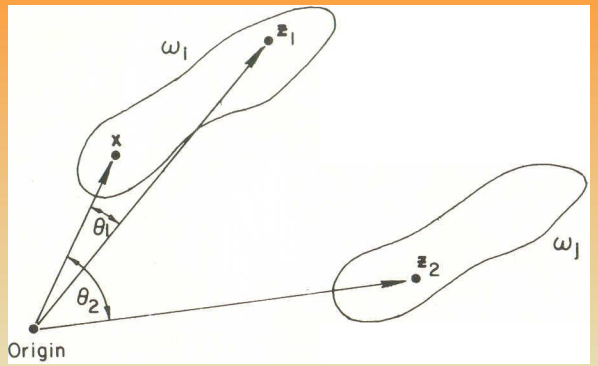
Measures of similarity need not be restricted to distance measures. For example, the non-metric similarity function:

$$S(X,Z) = \frac{X'Z}{\|X\|\|Z\|} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

which is the cosine of the angle between the vectors **x** and **z**, is maximum when **x** and **z** are oriented in the same direction with respect to the origin.

Contd..

This measure of similarity is useful when cluster regions tend to develop along principal axes, as shown below.



$$s(x,z_1) = \cos\theta_1 = \frac{x'z_1}{\|x\|\,\|z_2\|}$$

$$s(x,z_2) = \cos\theta_2 = \frac{x'z_2}{\|x\|\|z_2\|}$$

Figure 4. Illustration of a similarity measure

Contd..

Here the use of this similarity measure is governed by certain qualifications, such as sufficient separation of cluster regions with respect to each other as well as with respect to the coordinate system origin.

Contd..

When the patterns under consideration are binary valued with 0, 1 elements, the similarity function of Eq. (2) may be given an interesting non-geometrical interpretation.

Contd..

We say that a binary pattern **x** possesses the i_{th} attribute if x Then the term X^tZ in Eq. (2) is simply the number of attributes shared by **x** and **z**, while $\|X\|\|Z\| = \sqrt{(X'X)(Z'Z)}$ is the geometric mean of the number of attributes possessed by **x** and the number possessed by **z**. In this case the similarity function $s(X,Z)$ is, therefore, seen to be a measure of common attributes possessed by the binary vectors **x** and **z**.

Contd..

A binary variation of Eq. (2) which has been widely used in information retrieval, nosology (classification of diseases), and taxonomy (classification of plants and animals) is the so-called *Tanimoto measure,* which is given by

$$S(X,Z) = \frac{X'Z}{X'X + Z'Z - X'Z}$$

Clustering Criteria

After a measure of pattern similarity has been adopted, we are still faced with the problem of specifying a procedure for partitioning the given data into cluster domains.

Contd..

The clustering criterion used may represent a heuristic scheme, or it may be based on the minimization (or maximization) of a certain performance index. The heuristic approach is guided by intuition and experience.

Contd..

The Euclidean distance is generally used as measure of proximity. A suitable threshold is also necessary in order to define degrees of acceptable similarity in the cluster-seeking process.

Contd..

The performance-index (sometimes called objective function) approach is guided by the development of a procedure which will minimize or maximize the chosen performance index. One of the most often used indices is the sum of the squared errors index, given by..

Contd..

$$J = \sum_{j=1}^{N_C} \sum_{X \in S_j} \left\| X - m_j \right\|^2$$

where N_C is the number of clusters, S_j is the set of samples belonging to the *j*th cluster, and

$$m_j = \frac{1}{N_j} \sum_{X \in S_j} X$$

Contd..

is the sample mean vector of set S_j. N_j represents the number of samples in S_j. The index J represents the overall sum of the squared errors between the samples of a cluster domain and their corresponding mean. The K-means clustering algorithm is based on this performance index.

Contd..

Other common performance indices are :

- the average squared distances between samples in a cluster domain
- the average squared distances between samples in different cluster domains
- indices based on the scatter matrix concept,
- minimum and maximum-variance indices
- a score of other performance measures which have been used throughout the years.

Contd..

It is not uncommon to find a cluster-seeking algorithm that represents a combination of the heuristic and performance index approaches. The Isodata clustering algorithm is such a combination.

A Simple Cluster-Seeking Algorithm

Suppose that we have a set of *N* sample patterns $\{X_1, X_2, \ldots, X_N\}$. Let the first cluster center $Z_1 = X_1$ (or any other pattern) and select an arbitrary nonnegative threshold *T*.

Next, we compute the distance D_{21} from X_2 to Z_2 If this distance exceeds *T*, a new cluster center, $Z_2 = X_2$, is started. Otherwise, we assign X_2 to the domain of cluster center Z_1. Suppose that $D_{21} > T$ so that Z_2 is established. In the next step, the distances D_{31} and D_{32} from X_3 to Z_1 and Z_2 are computed.

Contd..

If both D_{31} and D_{32} are greater than *T*, a new cluster center, $Z_3 = X_3$, is created. Otherwise, we assign X_3 to the domain of the cluster center to which it is closest. This process is repeated for all patterns.

Contd..

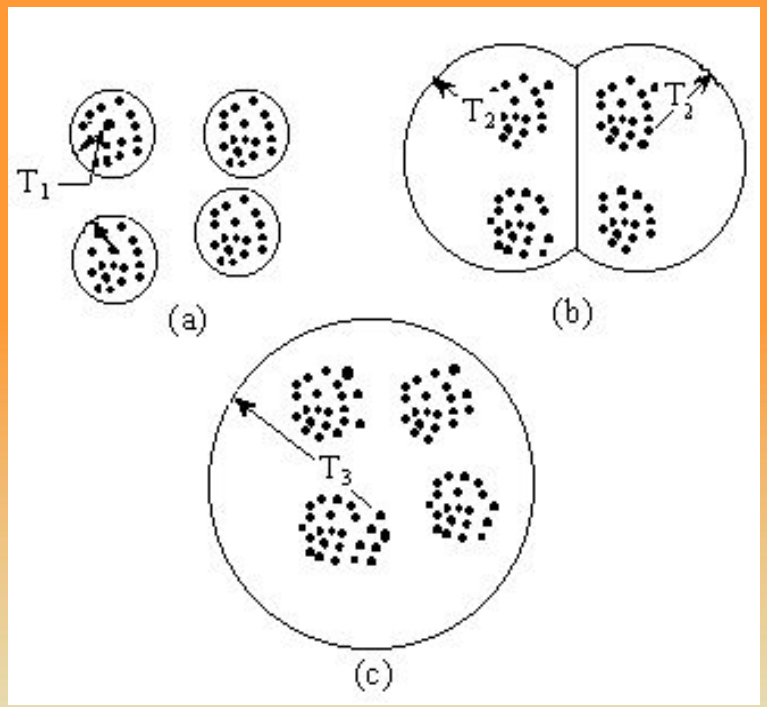


Figure 5. Effect of threshold in clustering

Contd..

The results of the foregoing procedure depend on

- the first cluster center chosen,
- the order in which the patterns are considered,
- the value of *T,* and,
- the geometrical properties of the data.

Contd..

Some intuitive idea for choosing a suitable value of T may be based on

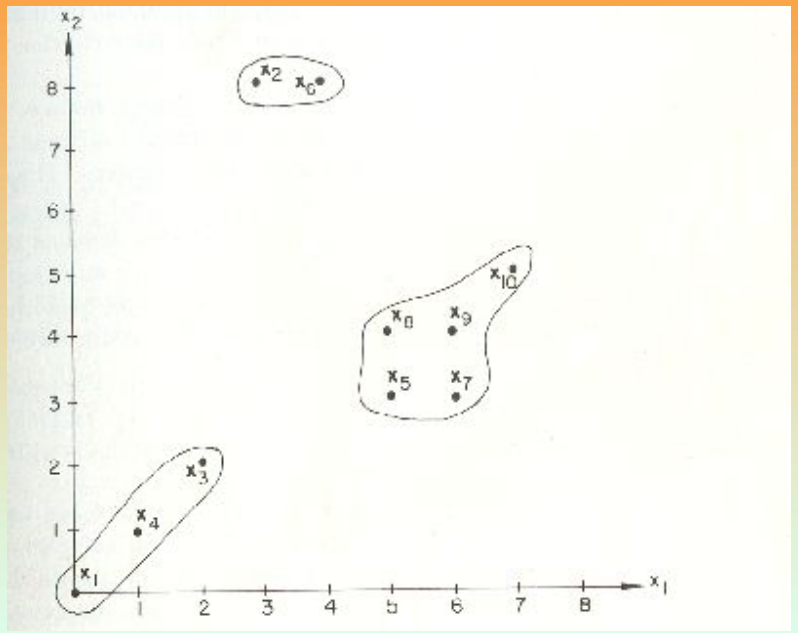
- MST of data points
- Intra-cluster distance
- Inter-cluster distance
- The closest and farthest points in cluster from the cluster center and
- the variance of these cluster domains

Contd..

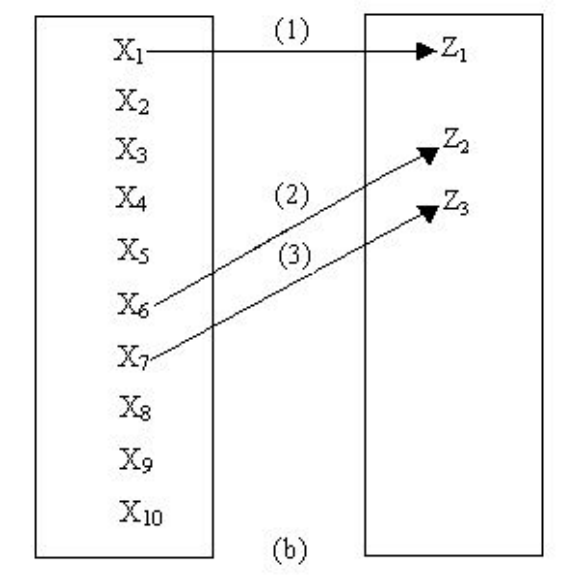
This procedure may be expected to yield useful results in situations where the data exhibit characteristic "pockets" which are reasonably well separated with respect to the chosen values of the threshold.

Maximin-Distance Algorithm

The *maximum-distance algorithm* is another simple heuristic procedure based on the Euclidean distance concept. Consider the ten two-dimensional samples shown in the Figure below.



Contd..



The table is initially empty. Then, in the first step, we arbitrarily let X_1 become the first cluster center, designated by Z_1.

Contd..

- Next, we determine the farthest sample from Z_1, which in this case is X_6, and call it cluster center Z_2
- In the third step we compute the distance from each remaining sample to Z_1 and Z_2. For every pair of these computations we save the minimum distance. Then, we select the maximum of these minimum distances. If this distance is an appreciable fraction (say, at least half or more), of the distance between cluster centers Z_1 and Z_2 we call the Z_3 corresponding sample cluster center . Otherwise the algorithm is terminated.

K-means Algorithm

Clustering is an unsupervised technique used in discovering inherent structure present in the set of patterns. In fact, clustering techniques aim to extract the groups present in a given data set and each such group is termed as a cluster.

Let the set of patterns be $S = \{x_1, x_2, ..., x_n\} \in \Re^m$, where x_i is the i_{th} pattern vector, n is the total number of patterns and m is the dimensionality of the feature space.

Contd..

Let the number of clusters be K. If the clusters are represented by $C_1, C_2, \ldots\ldots\ldots, C_K$ then we assume

P1. $C_i \neq \phi$, for $i = 1, 2, \ldots\ldots\ldots, K$

P2. $C_i \cap C_j = \phi$ for $i \neq j$ and

P3. $\cup_{i=1}^{K} C_i = S$ where ϕ represents null set.

Contd..

Clustering techniques may broadly be divided into two categories: Hierarchical and non-hierarchical. The non-hierarchical or partitional clustering problem deals with obtaining an optimal partition of into subsets such that some clustering criterion is satisfied.

Contd..

Among the partitional clustering techniques, the -Means algorithm has been one of the most widely used algorithms. Here the value of *K* needs to be known *a priori*. The principle used for clustering by K-means algorithm is to minimize the sum of intraclass distances to get the optimal clusters. Mathematically this principle has been stated below.

Contd..

1. Let $C_1, C_2, \ldots\ldots, C_K$ be a set of K clusters of S

2. Let $z_j = \dfrac{\left(\sum_{x \in C_j} x\right)}{\#C_j}$ for $j = 1,2,\ldots\ldots K.$ where $\#C_j$ represents the number of points in C_j

Contd..

3. Let $f(C_1, C_2, \ldots\ldots\ldots, C_K) = \sum_{j=1}^{K} \sum_{x \in C_j} \left\| x - z_j \right\|^2$

$f(C_1, C_2, \ldots\ldots\ldots, C_K)$ is referred as the objective function of the clustering $C_1, C_2, \ldots\ldots\ldots, C_K$

4. Minimize $f(C_1, C_2, \ldots\ldots\ldots, C_K)$ over all such $C_1, C_2, \ldots\ldots\ldots, C_K$ where $C_1, C_2, \ldots\ldots\ldots, C_K$ satisfy P1, P2 and P3 stated earlier.

Contd..

All possible clusterings of S are to be considered to get the optimal $C_1, C_2, \ldots\ldots, C_k$. So obtaining the exact solution of the problem is theoretically possible, yet not feasible in practice due to limitations of computer storage and time. One requires the evaluation of $S(n,k)$ partitions if exhaustive enumeration is used to solve the problem, where

$$S(n,k) = \frac{1}{k!}\sum_{j=1}^{k}(-1)^{k-j}\binom{k}{j}j^n.$$

This clearly indicates that exhaustive enumeration cannot lead to the required solution for most practical problems in reasonable computation time. For example, for the crude-oil data, the exact solution requires the examination of $S(56,3) > 10^{18}$ partitions.

Contd..

Thus, approximate heuristic techniques seeking a compromise or looking for an acceptable solution have usually been adopted. One such method is the Forgy's K-means algorithm.

Algorithm K-means

Step 1 **:** Select an initial cluster configuration.

Repeat

Step 2 **:** Calculate cluster centers $z_j, j = 1,2,.....K.$ of the existing groups.

Step 3 **:** Redistribute patterns among clusters utilizing the minimum squared Euclidean distance classifier concept :

$x_i \in C_j$ if $\left\|x_i - z_j\right\|^2 < \left\|x_i - z_l\right\|^2 \quad \forall l \in \{1,2,....K\}, l \neq j$

Until (there is no change in cluster centers)

End

Contd..

The behavior of the *K*-means algorithm is influenced by

- the number of cluster centers specified,
- the choice of initial cluster centers,
- the order in which the samples are taken, and,
- the geometrical properties of the data.

Contd..

Although no general proof of convergence exists for this algorithm, it can be expected to yield acceptable results when the data exhibit characteristic pockets which are relatively far from each other.

Isodata Algorithm

The *Isodata* (Iterative Self-Organizing Data Analysis) is similar in principle to the *K*-means procedure in the sense that cluster centers are iteratively determined sample means. Unlike the latter algorithm, however, Isodata represents a fairly comprehensive set of additional heuristic procedures which have been incorporated into an interactive scheme.

Before executing the algorithm it is necessary to specify a set N_C of initial cluster centers $Z_1, Z_2, \ldots, Z_{N_C}$. For a set of *N* samples, $\{X_1, X_2, \ldots, X_N\}$, Isodata consists of the following steps..

Contd..

Step 1. Specify the following process parameters:

K = number of cluster centers desired;

θ_N = a parameter against which the number of samples in a cluster domain is compared;

θ_s = standard deviation parameter;

θ_C = lumping parameter;

L = maximum number of pairs of cluster centers which can be lumped;

I = number of iterations allowed.

Contd..

Step 2. Distribute the *N* samples among the present cluster centers, using the relation

$$X \in S_j \quad if \quad \|X - Z_j\| < \|X - Z_i\|, \quad i = 1,2,...,N_C; \quad i =\neq j$$

for all **x** in the sample set. In this notation, S_j represents the subset of samples assigned to cluster center Z_j.

Step 3. Discard sample subsets with fewer than θ_N members; that is, if for any $j, N_j < \theta_N$, discard S_j and reduce N_C by 1.

Contd..

Step 4. Update each cluster center by $Z_j, j = 1,2,...,N_C$, setting it equal to the sample mean of its corresponding set S_j; that is,

$$Z_j = \frac{1}{N_j} \sum_{X \in S_j} X, \quad j = 1,2,...,N_C$$

where N_j is the number of samples in S_j

Step 5. Compute the average distance D_j of samples in cluster domain S_j from their corresponding cluster center, using the relation

$$D_j = \frac{1}{N_j} \sum_{X \in S_j} \left\| X - Z_j \right\|, \quad j = 1,2,...,N_C$$

Contd..

Step 6. Compute the overall average distance of the samples from their respective cluster centers, using the relation

$$D = \frac{1}{N} \sum_{j=1}^{N_C} N_j D_j$$

Contd..

Step 7.

(a) If this is the last iteration, set $\theta_C = 0$ and go to Step 11.

(b) If $N_C \leq K/2,$ go to Step 8.

(c) If this is an even-numbered iteration, or if $N_C \geq 2K,$ go to Step 11; otherwise, continue

Contd..

Step 8. Find the standard deviation vector $\sigma_j = (\sigma_{1j}, \sigma_{2j}, \ldots, \sigma_{nj})'$ for each sample subset, using the relation

$$\sigma_{ij} = \sqrt{\frac{1}{N}\sum_{X \in S_j}(X_{ik} - Z_{ij})^2}, \quad i = 1,2,\ldots,n; \qquad j = 1,2,\ldots,N_C$$

where *n* is the sample dimensionality, x_{ik} is the *i*th component of the *k*th sample in S_j, Z_{ij} is the *i*th component of Z_j, and N_j is the number of samples in S_j. Each component of σ_j represents the standard deviation of the samples in S_j along a principal coordinate axis.

Contd..

Step 9. Find the maximum component of each $\sigma_j, j = 1,2,...,N_C$ and denote it by $\sigma_{j\max}$.

Step 10. If for any $\sigma_{j\max}, j = 1,2,...,N_C$, we have $\sigma_{j\max} > \theta_S$ *and*

(a) $D_j > D$ and $N_j > 2(\theta_N + 1)$ *or*

(b) $N_C \leq K/2$

Contd..

then *split* Z_j into two new cluster centers Z_j^+ and Z_j^- , delete Z_j,and increase N_C by 1.

Cluster center Z_j^+ is formed by adding a given quantity y_j to the component of Z_j which corresponds to the maximum component of σ_j

Z_j^- is formed by subtracting y_j from the same component of Z_j. One way of specifying y_j is to let it be equal to some fraction of $\sigma_{j\max}$, that is, $y_j = k\sigma_{j\max}$, where $0 < k \leq 1.$

Contd..

The basic requirement in choosing y_j is that it be sufficient to provide a detectable difference in the distance from an arbitrary sample to the two new cluster centers, but not so large as to change the overall cluster domain arrangement appreciably.

If splitting took place in this step, go to Step 2; otherwise continue.

Contd..

Step 11. Compute the pairwise distances D_{ij} between all cluster centers :

$$D_{ij} = \left\| Z_i - Z_j \right\|, \quad i = 1,2,...,N_C - 1; \quad j = i+1,...,N_C$$

Contd..

Step 12. Compare the distances D_{ij} against the parameter θ_C. Arrange the *L* smallest distances which are less than θ_C in ascending order:

$$[D_{i_1 j_1}, D_{i_2 j_2}, \ldots, D_{iLjL}]$$

where $D_{i_1 j_1} < D_{i_2 j_2} < D_{iLjL}$ and *L* is the maximum number of pairs of cluster centers which can be lumped together. The lumping process is discussed in the next step.

Contd..

Step 13. With each distance D_{iljl} there is associated a pair of cluster centers Z_{il} and Z_{jl} Starting with the smallest of these distances, perform a pairwise lumping operation according to the following rule :

For $l = 1,2,\ldots,L,$ if neither Z_{il} nor Z_{jl} has been used in lumping in this iteration, merge these two cluster centers using the following relation:

$$Z_l^* = \frac{1}{N_{il}+N_{jl}}[N_{il}(Z_{il}) + N_{jl}(Z_{jl})]$$

Delete Z_{il} and Z_{jl} and reduce N_C by 1.

Contd..

It is noted that only pairwise lumping is allowed and that a lumped cluster center is obtained by weighting each old cluster center by the number of samples in its domain. Experimental evidence indicates that more complex lumping can produce unsatisfactory results. The above procedure makes the lumped cluster centers representative of the true average point of the combined subsets. It is also important to note that, since a cluster center can be lumped only once, this step will not always result in *L* lumped centers.

Contd..

Step 14. If this is the last iteration, the algorithm terminates. Otherwise go to Step 1 if any of the process parameters requires changing at the user's discretion, or go to Step 2 if the parameters are to remain the same for the next iteration. An iteration is counted every time the procedure returns to Step 1 or 2.

Example

Example: Let the patterns be {(0,0), (1,1), (2,2), (4,3), (4,4), (5,3), (5,4), (6,5)}

In this case $N = 8$ and $n = 2$. Suppose that we initially let $N_C = 1,\ Z_1 = (0,0)'$, and specify the following parameters:

Step 1. $K = 2,\quad \theta_N = 1,\quad \theta_S = 1,\quad \theta_C = 4,\quad L = 0,\quad I = 4$

Contd..

If no *a priori* information on the data being analyzed is available, these parameters are arbitrarily chosen and then adjusted during successive iterations through the algorithm.

Step 2. Since there is only one cluster center,

$$S_1 = \{X_1, X_2, \ldots, X_8\} \quad \text{and} \quad N_1 = 8$$

Step 3. Since $N_1 > \theta_N$, no subsets are discarded.

Contd..

***Step* 4.** Update the cluster centers

$$Z_1 = \frac{1}{N_1} \sum_{X \in S_1} X = \begin{pmatrix} 3.38 \\ 2.75 \end{pmatrix}$$

Step 5. Compute $\overline{D}_j$:

$$\overline{D}_1 = \frac{1}{N_1} \sum_{X \in S_1} \left\| X - Z_1 \right\| = 2.26$$

Contd..

Step 6. Compute $\overline{D}$ in this case

$$\overline{D} = \overline{D}_1 = 2.26$$

Step 7. Since this is not the last iteration and $N_c = K/2$ Go to step 8.

Step 8. Find the standard deviation vector for S_1

$$\sigma_1 = \begin{pmatrix} 1.99 \\ 1.56 \end{pmatrix}$$

Contd..

Step 9. The maximum component of σ_1 is 1.99 ; hence, $\sigma_{1\max} = 1.99.$

Step 10. Since $\sigma_{1\max} > \theta_S$ and $N_C = K/2,$ we split Z_1 into two new clusters. Following the procedure described in Step 10, suppose that we let $y_j = 0.5\sigma_{j\max} \approx 1.0.$ Then,

$$Z_1^+ = \begin{pmatrix} 4.38 \\ 2.75 \end{pmatrix}, \qquad Z_1^- = \begin{pmatrix} 2.38 \\ 2.75 \end{pmatrix}$$

Contd..

For convenience these two cluster centers are renamed Z_1 and Z_2 respectively. Also, N_C is increased by 1. Go to Step 2.

Step 2. The sample sets are now

$$S_1 = \{X_4, X_5, X_6, X_7, X_8\}, \qquad S_2 = \{X_1, X_2, X_3\}$$

and $N_1 = 5,\ N_2 = 3.$

***Step* 3.** Since both N_1 and N_2 are greater than θ_N, no subsets are discarded.

Contd..

***Step* 4.** Update the cluster centers :

$$Z_1 = \frac{1}{N_1}\sum_{X \in S_1} X = \begin{pmatrix} 4.80 \\ 3.80 \end{pmatrix}, \qquad Z_2 = \frac{1}{N_2}\sum_{X \in S_2} \begin{pmatrix} 1.00 \\ 1.00 \end{pmatrix}$$

***Step* 5.** Compute $\overline{D}_j, j = 1,2$:

$$\overline{D}_1 = \frac{1}{N_1}\sum_{X \in S_1} \|X - Z_1\| = 0.80, \qquad \overline{D}_2 = \frac{1}{N_2}\sum_{X \in S_2} \|X - Z_2\| = 0.94$$

Contd..

Step 6. Compute $\overline{D}$:

$$D = \frac{1}{N}\sum_{j=1}^{N_C} N_j \overline{D}_j = \frac{1}{8}\sum_{j=1}^{2} N_j \overline{D}_j = 0.85$$

Step 7. Since this is an even-numbered iteration, condition (c) of Step 7 is satisfied. Therefore, go to Step 11.

Contd..

***Step* 11.** Compute the pairwise distances:

$$D_{12} = \|Z_1 - Z_2\| = 4.72$$

***Step 12*.** Compare D_{12} to θ_C. In this case, $D_{12} > \theta_C$.

***Step 13*.** From the results of Step 12, we see that no lumping of cluster centers can take place.

***Step 14*.** Since this is not the last iteration, we are faced with the decision of whether or not to make an alteration in the parameters. Since, in this simple example,

Contd..

1. we have obtained the desired number of clusters,
2. their separation is greater than the average spread indicated by the standard deviations, and
3. each cluster subset contains a significant percentage of the total number of samples, we arrive at the conclusion that the cluster centers are representative of the data. Therefore, we return to Step 2.

Contd..

***Steps 2-6* :** yield the same results as in the previous iteration

***Step* 7.** None of the conditions in this step is satisfied. Therefore, we proceed to Step 8.

***Step* 8.** Compute the standard deviation of sets $S_1 = \{X_4, X_5, X_6, X_7, X_8\}$ and $S_2 = \{X_1, X_2, X_3\}$:

$$\sigma_1 = \begin{pmatrix} 0.75 \\ 0.75 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0.82 \\ 0.82 \end{pmatrix}$$

Contd..

***Step* 9.** In this case $\sigma_{1\max} = 0.75$ and $\sigma_{2\max} = 0.82.$

Step 10. The conditions for splitting are not satisfied. Therefore, we proceed to Step 11.

***Step 11*.** We obtain the same result as in the previous iteration:

$$D_{12} = \|Z_1 - Z_2\| = 4.72$$

***Step 12*.** We obtain the same result as in the Previous iteration.

Contd..

***Step* 13.** We obtain the same result as in the previous iteration.

Step 14. Nothing new has been added in this iteration, except the computation of the standard deviation vectors. Therefore, we return to Step 2.

Steps 2—6. yield the same results as in the previous iteration.

***Step 7*.** Since this is the last iteration, we set $\theta_C = 0$ and go to Step ll.

Contd..

Step 11. $D_{12} = \|Z_1 - Z_2\| = 5.72$ as before.

Step 12. We obtain the same result as in the previous iteration.

Step 13. From the results of Step 12, we see that no lumping can take place.

Step 14. Since this is the last iteration, the algorithm is terminated

Contd..

It should be clear, even from the above simple example, that the application of Isodata to a set of moderately complex data requires, in general, extensive experimentation before one can arrive at meaningful conclusions. However, by properly organizing the information obtained in each iteration, it is possible to gain considerable insight into the structure of the data.

Evaluation of Clustering Results

The principal difficulty in evaluating the results of clustering algorithms is inability to visualize the geometrical properties of a high-dimensional space. Several interpretation techniques exist which allow at least partial insight into the geometrical properties of the resulting cluster domains.

Contd..

A very useful interpretation tool is the distance between cluster centers This information is best presented in the form of a table.

Contd..

Cluster Center	Z_1	Z_2	Z_3	Z_4	Z_5
Z_1	0.0	4.8	14.7	2.1	50.6
Z_2		0.0	21.1	6.1	48.3
Z_3			0.0	15.0	36.7
Z_4				0.0	49.3
Z_5					0.0

Table 1. ***Distance Table for Interpreting Clustering Results***

Contd..

Cluster center Z_5 is far removed from the other cluster centers. If it is known that this cluster center is associated with numerous samples, we would accept it as a valid description *of* the data. otherwise we might dismiss this cluster center as representative of noise samples.

Contd..

If two cluster centers are relatively close, and one of the centers is associated with a much larger number of samples, it is often possible to merge the two cluster domains. The variances of a cluster domain about its mean can be used to infer the relative distribution of the samples in the domain.

Contd..

Cluster Domains	Variances			
	σ_1	σ_2	σ_3	σ_4
S_1	1.2	0.9	0.7	1.0
S_2	2.0	1.3	1.5	0.9
S_3	3.7	4.8	7.3	10.4
S_4	0.3	0.8	0.7	1.1
S_5	4.2	5.4	18.3	3.3

Table 2. Variance Table for Interpreting Clustering Results

Contd..

It is assumed that each variance component is along the direction of one of the coordinate axes.

Note :

- Since domain S_1 has very similar variances, it can be expected to be roughly spherical in nature.
- Cluster domain S_5 on the other hand, is significantly elongated about the third coordinate axis. A similar analysis can be carried out for the other domains.

Contd..

This information, coupled with the distance table and sample numbers, can be of significant value in interpreting clustering results.

Graph-Theoretic Approach

Till now, the clusters are determined in such a way that the intraset distance among each cluster is kept to a minimum, and the interset distances between two clusters are made as large as possible.

Contd..

An alternative approach to cluster seeking is to make use of some basic notion in graph theory. In this approach, a *pattern graph* is first constructed from the given sample patterns, which form the nodes of the graph. A node *j* is connected to a node *k* by an edge, if the patterns corresponding to these two nodes are similar or are related.

Contd..

Pattern X_j and pattern X_k are said to be similar if the similarity measure $S(X_j, X_k)$ is greater than a pre-specified threshold ***T.*** The similarity measure may be used to generate a similarity matrix **S**, whose elements are 0 or 1. The similarity matrix provides a systematic way to construct the pattern graph. Since cliques of a pattern graph form the clusters of the patterns, cluster seeking can be accomplished by detecting the cliques of the pattern graph.

Contd..

Several clique detection algorithms and programs have been introduced in the literature. Several such methods exist that utilize the MST of the data points to get the clustering.

Thank You